

## **LAB – 2 ASSIGNMENT**

ET 2595 Network and System Security

Jan 2024

Name : NIKHIL CHALIKONDA

Id No:

Mail: nich20@student.bth.se

### **I. Tasks Performing**

#### **Objective 1: [v3\_ca]**

This section enumerates the parameters linked to the [v3\_ca] extension.:

```
subjectKeyIdentifier      = hash
authorityKeyIdentifier   = keyid:always, issuer
basicConstraints         = critical, CA:true
keyUsage                 = critical, digitalSignature, cRLSign, keyCertSign
```

- 1. **\*\*SubjectKeyIdentifier Extension\*\*:** This identifies certificates with a specific public key using a 'hash' value or a hexadecimal string. OpenSSL calculates this hash as outlined in RFC 5280.
- 2. **\*\*AuthorityKeyIdentifier Extension\*\*:** This extension, when set as 'keyid:always, issuer', links a private key to its corresponding public key. The process involves the utilization of two distinct values: 'keyid' to replicate the subject key identifier from the issuer certificate, and 'issuer' to duplicate the distinguished name (DN) and serial number of the issuer from their respective certificate.
- 3. **\*\*BasicConstraints Extension\*\*:** When set to 'critical, CA:true', this extension helps determine if the certificate is a certificate authority (CA). Setting it as 'CA:true' means the certificate is a certificate authority. Marking it as 'critical' is essential as it marks the extension as critical.

- 4. \*\*KeyUsage Extension\*\*: This extension, set as 'critical, digitalSignature, cRLSign, keyCertSign', defines how the key associated with the certificate can be used. Specifically, it permits the key to verify signatures on both public key certificates and Certificate Revocation Lists (CRLs)..

## Objective 2: [ v3\_intermediate\_ca ]

Within this section, you'll locate the parameters associated with the [v3\_intermediate\_ca] extension.

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

In Task 2, the parameters remain consistent with those detailed in Task 1, save for a distinctive alteration involving the assignment of 'pathlen:0'. This specific modification exclusively applies to the endorsement of concluding user certificates. The imperative criterion stipulates that the 'pathlen' value necessitates being either precisely 0 or a numerical value surpassing 0 to ensure compliance.

## Performing of Tasks 3: [ usr\_cert ]

This part details the settings associated with the [usr\_cert] extension:

```
basicConstraints        = CA:FALSE
subjectKeyIdentifier    = hash
authorityKeyIdentifier  = keyid, issuer
extendedKeyUsage        = clientAuth, emailProtection
keyUsage                = critical, nonRepudiation, digitalSignature,
                           keyEncipherment
```

- **basicConstraints:** When given the value CA:FALSE, it indicates that the certificate is not a Certificate Authority (CA) certificate.
- **nscomment = "OpenSSL Generated Certificate":** This is assigned so that it appears in Netscape's comment listbox.
- **subjectKeyIdentifier=hash:** This is a way of identifying certificates using their public keys.
- **authorityKeyIdentifier=keyid,issuer:** This helps in identifying the public key that can be traced back to the private key used to sign the certificate.

## Performing of Task 4: [ server\_cert ]

Within this segment, you'll discover specific information outlining the parameters attributed to the [server\_cert] extension.

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

- **basicConstraints=CA:FALSE:** This indicates it's not a certificate used for being a Certificate Authority (CA).
- **subjectKeyIdentifier=hash:** This is a way to spot certificates based on their public keys.
- **authorityKeyIdentifier=keyid,issuer:** It helps locate the public key connected to the private key used for signing the certificate.
- **keyUsage:** This section lists the allowed uses for the key. It can include types like digitalSignature, nonRepudiation, keyEncipherment, dataEncipherment, keyAgreement, keyCertSign, cRLSign, encipherOnly, and decipherOnly.
  - **keyEncipherment:** Used when a public key carries private keys for transport.
  - **digitalSignature:** Used to check digital signatures on things other than certificates and lists. These bits confirm things like entity authentication and integrity services.

### Performing of Task 5: [policy\_match] and [policyAnything]

```
[ policy_match ]
countryName          = match
stateOrProvinceName = match
organizationName     = match
organizationalUnitName = optional
commonName           = supplied
emailAddress         = optional

[ policyAnything ]
countryName          = optional
stateOrProvinceName = optional
organizationName     = optional
organizationalUnitName = optional
commonName           = supplied
emailAddress         = optional
```

Certainly! In the "policy\_match" policy, if a field is labeled as "match," it has to contain the exact information as that field in the CA's details. Fields marked as "supplied" must be present, while those labeled as "optional" don't have to be there, but they're allowed. Essentially, this policy mandates that any certificate issued by it must have the same country, state, and organization name as the CA.

## Performing of Task 6:

Explore the method for generating a self-signed certificate through a command.

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

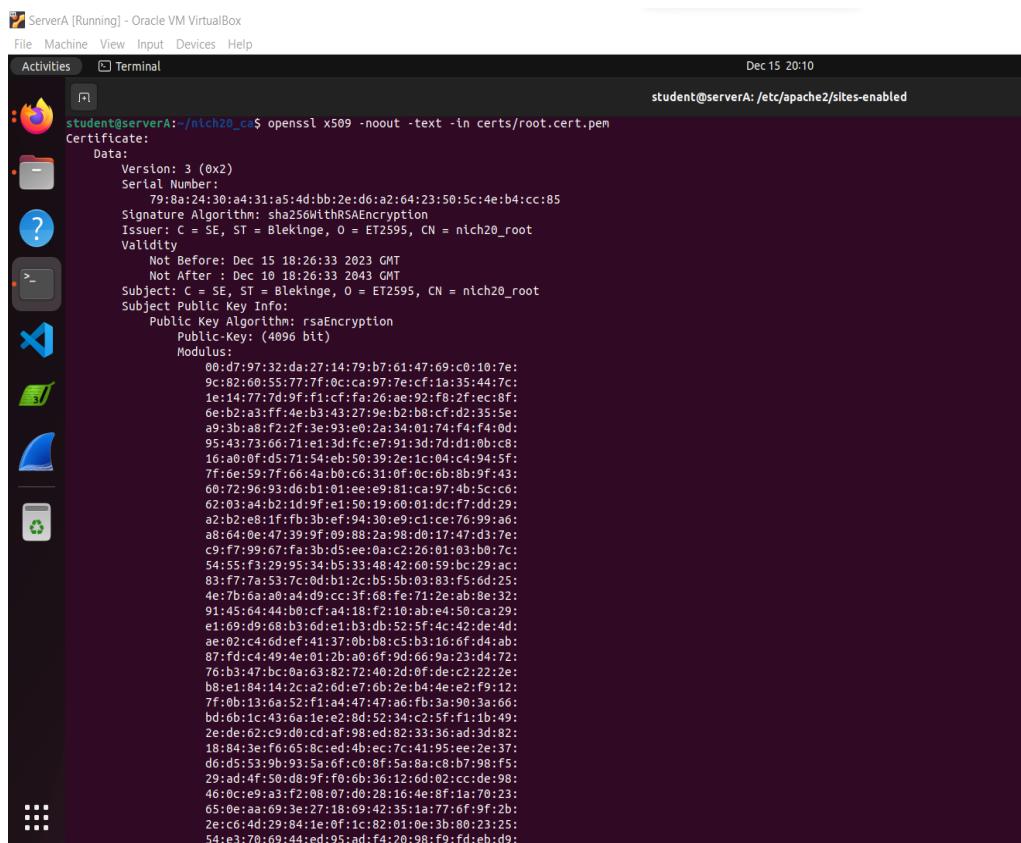
These directives serve distinct functions:

- 'config' designates a configuration file, such as openssl.cnf.
- 'key' denotes the file containing the private key.
- 'new' signals the creation of a new certificate request.
- 'x509' is used for self-signed certificate generation.
- 'days' permits setting the certificate's validity duration.
- 'extensions' indicates the inclusion of extensions in the certificate.
- 'out' specifies the file path for storing the resulting certificate.

## Performing of Task 7:

The instruction employed to authenticate the root certificate.

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



```
student@serverA:~/nich20_ca$ openssl x509 -noout -text -in certs/root.cert.pem
Certificate:
Data:
Version: 3 (0x2)
Serial Number:
    79:8a:24:30:a4:31:a5:4d:bb:2e:d6:a2:64:23:50:5c:4e:b4:cc:85
Signature Algorithm: sha256WithRSAEncryption
Issuer: C = SE, ST = Blekinge, O = ET2595, CN = nich20_root
Validity
    Not Before: Dec 15 18:26:33 2023 GMT
    Not After : Dec 16 18:26:33 2043 GMT
Subject: C = SE, ST = Blekinge, O = ET2595, CN = nich20_root
Subject Public Key Info:
    Public Key Algorithm: rsaEncryption
        Public-Key: (4096 bit)
            Modulus:
                00:d7:97:32:da:27:14:79:b7:61:47:69:c0:10:7e:
                9c:82:60:55:77:7f:0c:ca:97:7e:c1:a3:35:44:7c:
                1e:14:77:7d:9f:f1:cf:fa:26:ae:92:f8:2f:ec:8f:
                6e:b2:a3:ff:4e:b3:43:27:9e:b2:b8:cf:d2:35:5e:
                a9:3b:a8:f2:2f:3e:93:eo:2a:34:61:74:f4:f4:8d:
                95:43:73:66:71:el:3d:fc:e7:91:3d:7d:di:0b:c8:
                16:a0:0f:d5:71:54:eb:50:39:2e:1c:04:c4:94:5f:
                7f:ee:59:7f:66:4a:b6:co:31:0f:0c:6b:8b:9f:43:
                60:72:96:93:d6:b1:01:ee:e9:81:ca:97:4b:5c:c6:
                62:03:a4:b2:1d:9f:el:50:19:60:01:dc:f7:dd:29:
                a2:b2:e8:1f:fb:3b:ef:94:30:ee:c1:ce:76:99:a6:
                a8:64:0e:47:39:9f:09:88:2a:98:d0:17:47:d3:7e:
                c9:f7:99:67:fa:3b:d5:ee:0a:c2:26:01:03:bc:07:c:
                54:55:f3:29:95:34:b5:33:48:42:60:59:bc:29:ac:
                83:f7:7a:53:7c:0d:b1:2c:b5:b8:03:83:f5:6d:25:
                4e:7b:6a:a0:a9:d9:cc:3f:68:fe:71:2e:ab:8e:32:
                91:45:64:44:b6:cf:a4:18:f2:10:ab:ed:56:c8:29:
                e1:69:d9:60:b3:6d:el:b3:db:52:5f:4c:42:de:4d:
                ae:02:c4:6d:ef:41:37:0b:08:c5:b3:16:6f:d4:ab:
                87:fd:c4:49:4e:01:2b:a0:6f:9d:66:9a:23:d4:72:
                76:b3:47:bc:0a:63:82:72:40:2d:0f:de:c2:22:2e:
                b8:el:84:14:2c:a2:6d:e7:ob:2e:b4:4e:e2:f9:12:
                7f:0b:13:6a:52:f1:a4:47:47:a6:fb:3a:90:3a:66:
                bd:6b:1c:43:6a:1e:e2:8d:52:34:c2:5f:f1:1b:49:
                2e:de:62:c9:d0:cd:af:98:ed:82:33:36:ad:3d:82:
                18:84:3e:f6:65:8c:ed:4b:ec:7c:41:95:ee:2e:37:
                d6:d5:53:9b:93:5a:6f:c0:8f:5a:8a:c8:b7:98:f5:
                29:ad:4f:50:d8:9f:f0:6b:36:12:6d:02:cc:de:98:
                46:0c:e9:a3:f2:08:07:d0:28:10:4e:8f:la:70:23:
                65:0e:aa:69:3e:27:18:69:42:35:la:77:0f:9f:2b:
                2e:c6:4d:29:84:1e:0f:1c:82:01:0e:3b:80:23:25:
                54:e3:70:69:44:ed:95:ad:f4:20:98:f9:fd:eb:d9:
```

```

student@serverA:/etc/apache2/sites-enabled
46:0c:e9:a3:f2:08:07:d0:28:16:4e:8f:a1:70:23:
05:0e:0a:0d:3e:27:0f:1c:b2:01:0e:0d:3e:27:0f:9f:2b:
2e:0c:02:09:03:02:09:03:02:09:03:02:09:03:02:09:03:
54:c5:70:69:44:cd:05:ad:f4:20:98:f0:fd:eb:d0:
1f:fe:82:18:c2:9a:3b:10:08:07:e1:e1:48:d2:bc:
ec:23:0a:91:15:1b:45:58:c9:9b:82:39:e5:5a:c8:
d6:13:81
Exponent: 65537 (0x10001)
X509v3 extensions:
X509v3 Subject Key Identifier:
X509v3 Authority Key Identifier:
45:64:6A:F6:D8:0C:44:A2:0C:A3:76:36:79:9B:E0:83:A2:5C:35:FD
X509v3 Basic Constraints: critical
CA:TRUE
X509v3 Key Usage: critical
Signature Algorithm: sha256WithRSAEncryption
Signature Value:
9e:be:02:de:0e:81:a7:74:10:65:08:96:c3:18:d0:f6:57:8a:
e4:bc:1a:23:83:f8:cd:09:e0:0d:e0:11:b6:b9:6c:2f:f8:c2:fa:
bd:20:27:92:14:0f:41:a8:0e:0b:bb:ca:2a:4c:db:38:f1:c8:
27:ed:8e:7a:47:f6:3c:5c:cd:21:50:90:cc:23:26:5b:20:
c6:fd:ea:78:29:78:1e:4d:87:0a:30:f5:af:65:d0:f9:93:4c:e2:11:
10:0f:ce:0:cd:55:9a:06:85:c8:05:a3:9e:1e:d0:c0:0e:47:
7b:88:fb:df:7a:46:72:1d:0a:0d:0e:0d:0e:0d:0e:0d:0e:0d:
91:5d:4d:3c:c4:ae:06:43:a9:49:0d:f6:bc:00:03:b6:c0:35:a8:
03:ca:fd:1b:2a:9a:cc:0:5b:59:32:5a:ba:e4:c2:9e:07:50:48:
bf:81:4e:rd:1:20:5e:f5:03:9f:5d:50:06:83:a8:28:82:85:ec:
cb:7b:9f:11:0b:08:00:aab:f7:64:09:0e:7:37:db:54:9:6b:
ca:4d:4a:2a:02:f2:c9:08:bb:ad:bd:9d:77:f1:c5:14:7e:30:08:
61:70:c9:60:b9:00:62:2d:69:00:cc:16:93:bd:b3:a5:53:58:
bf:9e:9c:23:f0:03:19:d0:7e:74:be:c5:41:06:bd:0d:01:7a:
75:ce:c8:0e:02:bb:a2:14:2b:b3:f6:ba:e4:b5:08:41:f3:a7:
3b:0d:0d:0d:0d:0d:0d:0d:0d:0d:0d:0d:0d:0d:0d:0d:0d:0d:
a9:23:35:12:81:f9:03:92:30:fe:0f:f5:ad:76:3b:f8:72:b9:
8c:55:14:a5:33:12:30:f5:f8:46:44:fd:2a:68:fd:c7:56:0e:
dc:d1:48:5b:64:14:83:1e:9a:08:2e:4d:b1:79:c1:48:01:37:
54:ab:36:f0:78:9c:39:97:8c:0d:e1:e2:c8:fd:02:e6:68:cf:
49:0c:f2:64:db:30:67:dd
```

## Performing of Task 8:

The command is utilized to verify the Certificate Signing Request (CSR).

```
openssl req -text -noout -verify -in ca1/csr/ca1.csr.pem
```

The Certificate Signing Request (CSR) is created using the RSA private key, and it appears as follows:

```

student@serverA:~/nich20_cai$ openssl req -text -noout -verify -in ca1/csr/ca1.csr.pem
Certificate request self-signature verify OK
Certificate Request:
Data:
Version: 1 (0x0)
Subject: C = SE, ST = Blekinge, O = ET2595, CN = nich20_cai
Subject Public Key Info:
    Public Key Algorithm: rsaEncryption
        Public-Key: (4096 bit)
            Modulus:
                00:ed:6d:e2:28:de:45:20:58:87:ad:b0:6e:c1:aa:
                e7:ea:0:6e:fa:9f:3a:32:cb:31:f6:ba:97:a1:66:
                f9:45:0:0:6f:2c:f3:24:0d:69:43:f6:72:2f:97:87:
                8f:58:78:0e:65:a6:e6:2c:d1:95:f9:9e:51:bb:cc:
                d3:f8:b9:a9:8a:7d:fa:89:b0:e1:ef:f8:2e:2e:40:
                d8:de:4d:5f:c7:b3:fa:e9:bd:b5:87:d7:32:f5:c9:
                e9:5d:3c:34:a6:3e:67:11:32:69:76:f9:4f:82:03:
                10:d8:45:e5:50:43:49:c0:c7:83:a8:b6:c2:c9:df:
                77:d4:1b:6a:f1:b7:fd:43:9c:3a:a5:c5:79:c0:e0:
                49:07:53:8c:ba:ca:c0:22:71:b0:53:10:dc:c5:db:
                91:c8:4f:2c:7c:8e:3c:bd:74:c7:10:a6:20:df:c3:
                fd:26:15:0b:27:71:c9:9d:a0:f0:b7:c2:46:96:58:
                92:52:e0:99:a3:99:20:64:e1:a0:c9:88:df:d9:4c:
                6f:de:d6:93:fa:38:e3:3d:e8:8d:46:96:e3:08:88:
                3b:19:da:75:46:ba:0e:83:0d:93:76:a0:07:e0:6d:
                92:7c:f5:dd:1:dc:be:b5:c7:aa:56:95:df:5d:ee:
                cf:07:f4:6e:0d:6e:30:77:53:a3:33:f8:89:04:a4:
                48:bc:c7:50:86:2c:0d:e1:b6:56:21:1:84:f7:ee:
                b7:a6:38:ad:5d:fa:b1:60:1f:85:89:54:c2:6c:cd:
                6c:ac:90:91:ec:a7:90:50:b8:25:e7:02:11:ed:2e:
                1d:9f:94:5c:1a:55:70:7b:93:36:de:22:93:4a:2f:
                b9:2d:63:33:03:00:7b:16:22:fd:b9:be:a1:37:5b:
                2e:3e:0b:c0:33:b6:ef:f2:7d:ff:94:08:0c:54:f7:
                31:14:b4:61:49:74:53:0f:8c:44:94:02:8c:18:4f:
                98:14:b2:34:86:78:08:68:22:b3:23:24:22:56:c8:
                04:de:a4:ad:40:01:37:e9:41:b5:b8:50:51:ec:79:
                32:fb:45:d0:64:0f:4c:2d:49:d8:4c:0f:8f:a2:36:
                36:1e:56:0c:04:a2:7d:4f:b3:98:06:55:d7:0d:2f:
                a9:1a:93:d7:5e:96:08:c4:85:53:f8:3c:34:98:da:
                97:0f:56:a2:0c:a7:f7:64:d1:70:29:da:da:76:59:
                ac:7b:4:38:23:d6:f1:94:2e:9f:11:be:41:7e:ea:
                42:44:72:25:f4:e5:aa:74:73:f3:bf:26:d0:14:0e:
                79:f6:33:6f:77:27:98:42:71:84:41:ca:f2:f8:96:
                79:b6:03:17:4e:e2:fc:09:b7:bc:be:a0:02:2f:4d:
```

```

ServerA [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal Dec 15 20:09
student@serverA: /etc/apache2/sites-enabled

2e:5e:00:c0:35:00:ef:12:70:11:94:00:00:34:17:
31:14:b4:61:49:74:53:0f:8c:44:94:d2:8c:18:4f:
98:14:b2:34:86:78:08:68:22:3b:23:24:22:56:c8:
04:de:a4:ad:40:01:37:e9:41:b5:b8:50:51:ec:79:
32:fb:45:d0:64:0f:4c:d2:49:d8:4c:0f:8f:a2:36:
36:1e:56:0c:04:a2:7d:4f:b3:98:06:55:d7:0d:2f:
a9:1a:93:d7:5e:96:08:c4:85:p3:f8:c3:34:98:da:
97:0f:56:a2:6c:af:76:64:d1:70:29:da:da:76:59:
ac:b7:b4:38:23:d6:1f:94:2e:9f:11:8e:41:7e:ea:
42:44:72:25:f4:e5:aa:74:73:f3:bf:26:d6:14:de:
79:6c:33:6f:77:27:98:42:71:84:41:ca:f2:f8:96:
79:b6:03:17:4e:ee:fc:09:b7:bc:be:a0:02:2f:4d:
47:83:75

Exponent: 65537 (0x10001)
Attributes:
  (none)
Requested Extensions:
Signature Algorithm: sha256WithRSAEncryption
Signature Value:
04:79:25:a5:20:6d:8f:3b:0c:13:11:43:49:14:2d:e4:a2:f3:
cc:53:a5:c4:91:c5:f7:a3:38:b5:c5:2a:b8:5a:1e:b1:78:ff:bf:
e1:c5:4b:00:1a:66:93:e4:c2:c3:f9:a8:18:99:0e:cd:ad:ed:6:d7:
f8:ac:bc:6e:7f:d7:b2:9e:ab:58:be:c9:df:20:d4:80:49:af:
ee:76:af:96:95:9f:a3:e2:dc:e7:2e:de:f7:b7:83:03:5a:48:
eb:03:96:16:c6:0b:f5:c7:a2:e6:ab:60:03:0c:6e:bc:fe:78:
3e:64:2b:36:e7:ba:50:be:c7:1f:d3:0b:07:bc:25:da:de:22:
b0:d3:19:db:9e:7a:bc:fb:7c:d2:0e:f1:b4:df:65:73:21:a8:
bf:4e:1f:cb:48:6b:a1:9d:d4:41:ac:d6:88:05:b2:83:98:de:
ed:45:bc:12:f4:a4:ac:a3:d5:1e:bf:70:b9:11:e0:5f:c9:8d:
6a:e2:9f:6e:ac:2c:a2:38:10:72:d1:20:c8:64:81:6f:5d:5a:
b3:be:8b:cf:77:17:1e:d0:e7:4e:cd:85:cd:9e:43:14:4d:d5:
2d:d2:fa:01:e7:68:db:0f:e4:f7:fe:c3:3e:87:3f:63:50:a9:
0a:4d:8f:e5:f5:b7:63:f5:1a:48:08:1c:3c:33:3a:e0:f6:a7:
cd:c9:3d:88:3e:cf:65:6f:ba:12:e9:d3:d5:aa:12:e4:8e:51:
c2:b9:0d:6f:e2:98:1f:11:d1:e8:3e:73:09:66:5c:60:35:dc:
82:db:2a:f4:ea:8e:8d:24:21:a5:77:9b:58:9a:98:a3:22:d4:
77:1e:38:7a:8a:68:6f:45:1d:f9:02:da:1d:de:df:37:dd:fe:
9f:bf:59:08:96:df:39:4c:d3:c6:1d:95:db:4e:79:b5:ds:e9:
a1:90:3d:a9:53:ec:42:9c:9a:a0:91:50:fe:25:8d:f2:28:
af:35:bf:11:bc:e9:77:28:2c:27:b5:10:45:81:1e:f5:9c:c8:
f7:a6:69:bd:b6:11:66:91:91:92:65:75:d4:1c:28:d7:d7:40:
e6:7b:9e:93:de:22:67:4a:39:57:ad:97:b7:42:9d:4c:09:07:8a:
cf:f3:a8:f5:62:dd:1a:c7:e6:5d:43:06:a3:e0:a5:aa:7c:0a:
d2:55:a5:59:4d:26:b3:77:7c:a4:18:bc:79:50:62:f1:2d:9a:
55:de:6a:bb:a3:0a:0d:80:ef:cd:7b:88:39:b7:03:23:30:94:
72:67:02:51:b7:c0:e4:34:c9:3e:3d:61:36:15:e5:33:fe:94:
a4:dd:0e:56:d1:c2:73:18:83:b3:1c:48:8a:f8:6c:f3:3c:25:
fe:f0:5f:05:9a:cf:25:02

```

## Performing of Task 9:

This command enables the generation of the CA1 certificate utilizing a Certificate Signing Request (CSR):

```
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
```

- 'config': It points to the 'openssl.cnf' configuration file necessary for the CA1 certificate signing process.
- 'extensions': Specifies the 'v3\_intermediate\_ca' extension, enabling the CA1 certificate to function as an intermediate certificate authority with essential capabilities.
- 'days 3650': Sets the certificate's validity period to 3650 days, equivalent to a 10-year duration.
- 'notext': Instructs the command to abstain from generating text-based output.
- 'md sha256': Specifies the utilization of the SHA-256 encryption algorithm for the certificate's signature generation.
- 'in': Identifies the input file.
- 'out': Designates the destination path for the resulting output file storage.

## Performing of Task 10:

A sequence of commands is utilized to validate the authenticity and legitimacy of the CA1 certificate.

```
openssl x509 -noout -text -in ca1/certs/ca1.cert.pem
openssl verify -CAfile certs/root.cert.pem ca1/certs/ca1.cert.pem
```

```
student@serverA:~/nich20_ca$ openssl x509 -noout -text -in ca1/certs/ca1.cert.pem
Certificate:
Data:
    Version: 3 (0x2)
    Serial Number: 4096 (0x1000)
    Signature Algorithm: sha256WithRSAEncryption
    Issuer: C = SE, ST = Blekinge, O = ET2595, CN = nich20_root
    Validity
        Not Before: Dec 15 18:28:49 2023 GMT
        Not After : Dec 12 18:28:49 2033 GMT
    Subject: C = SE, ST = Blekinge, O = ET2595, CN = nich20_ca1
Subject Public Key Info:
    Public Key Algorithm: rsaEncryption
        Public-Key: (4096 bit)
            Modulus:
                00:ed:6d:e2:28:d8:f4:5c:20:58:87:ad:b0:6e:c1:aa:
                e7:ea:a6:fe:fa:9f:3a:32:cb:31:f6:ba:97:a1:66:
                f9:45:c8:6f:2c:cf:24:d0:69:43:f6:72:f9:87:87:
                8f:58:78:0e:05:a6:e6:2c:d1:95:f9:9e:s1:1b:c:
                d3:f8:b9:a9:8a:7d:fa:89:b0:e1:f8:02:e4:40:
                d8:de:40:f7:c7:b3:fa:e0:bd:5b:87:d7:32:f5:9:
                e9:5d:3c:34:a6:3e:67:11:32:69:76:f9:4f:82:03:
                10:da:84:5e:50:43:49:c0:c7:83:a8:b6:c2:c9:df:
                77:da:41:ba:f1:b7:fd:43:9c:3e:a5:c5:79:c0:e0:
                49:07:53:8c:ba:ca:c0:22:71:b0:53:10:dc:c5:db:
                91:c8:4f:cc:7c:8c:3c:bd:74:c7:10:ad:20:df:c3:
                fd:26:15:0b:27:71:c9:9d:fb:67:c2:46:96:58:
                92:52:e0:99:a3:99:20:64:e1:a9:c9:88:df:d9:4c:
                6f:de:d0:93:fa:38:3e:3d:8d:46:96:3:08:bb:
                3b:19:da:75:46:ba:6e:83:9d:93:76:0a:07:e0:6d:
                92:7c:f9:dd:dd:11:dc:ze:b5:9d:a9:56:95:df:d5:ee:
                cf:0e:4d:5d:5d:6e:53:53:53:53:53:53:53:53:53:
                40:b6:c7:50:50:52:0d:61:b0:56:21:21:44:77:ee:
                b7:a6:38:ad:5d:5d:5b:60:1f:85:89:54:c2:66:cd:
                66:c1:98:91:91:91:7:99:50:b7:25:c7:82:11:ed:2e:
                1d:9f:94:5c:1a:55:70:7b:93:36:de:22:93:4a:2f:
                b9:2d:63:33:03:08:7b:16:22:fd:b9:be:a1:37:5b:
                2e:3e:0b:0:33:b6:ef:f2:7d:ff:94:ba:0c:54:f7:
                31:14:b4:61:49:74:53:0f:144:94:d2:8:18:4f:
                98:14:b2:34:86:78:08:69:22:3b:23:24:22:56:c8:
                04:de:ad:ad:40:01:37:e9:41:b5:b8:50:51:ec:79:
                32:fb:45:0d:64:0f:4c:2d:49:db:4c:0f:8f:a2:36:
                36:1e:56:0c:04:02:7d:4f:b3:98:06:55:7d:0d:2f:
                a9:1a:93:07:5e:96:08:c4:85:53:f8:3c:34:98:da:
                97:0f:56:a2:6c:af:76:64:d1:70:29:da:76:59:
                ac:b7:b4:38:23:d6:01:94:2e:9f:11:8e:41:7e:ea:
                42:44:72:25:f4:e5:aa:74:73:f3:bf:26:de:14:de:
                79:6c:33:6f:77:27:98:42:71:84:41:ca:f2:f8:96:
```

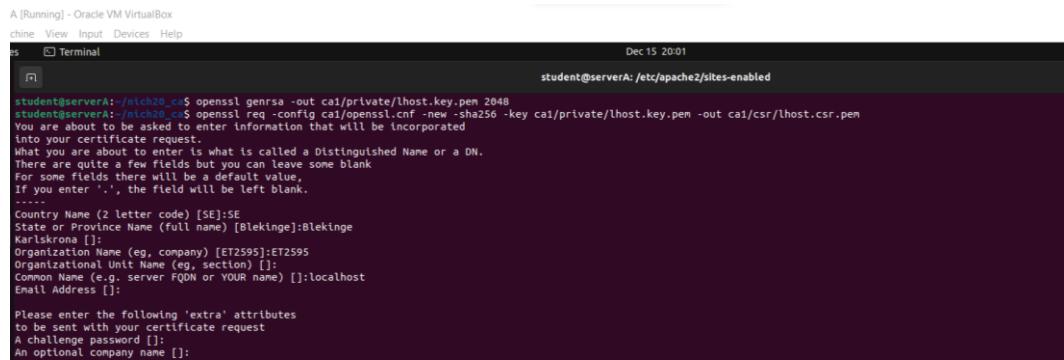
```
student@serverA:~/nich20_ca$ openssl verify -CAfile certs/root.cert.pem ca1/certs/ca1.cert.pem
ca1/certs/ca1.cert.pem: OK
```

## Performing of Task 11:

Below is the process used to generate a server certificate:

- Initially, an RSA private key was produced using a particular command. Following that, a Certificate Signing Request (CSR) was generated by employing the previously created RSA private key named lhost.key.pem.

```
openssl genrsa -out ca1/private/lhost.key.pem 2048
openssl req -config ca1/openssl.cnf -new -sha256 -key ca1/private/lhost.key.pem -
out ca1/csr/lhost.csr.pem
```

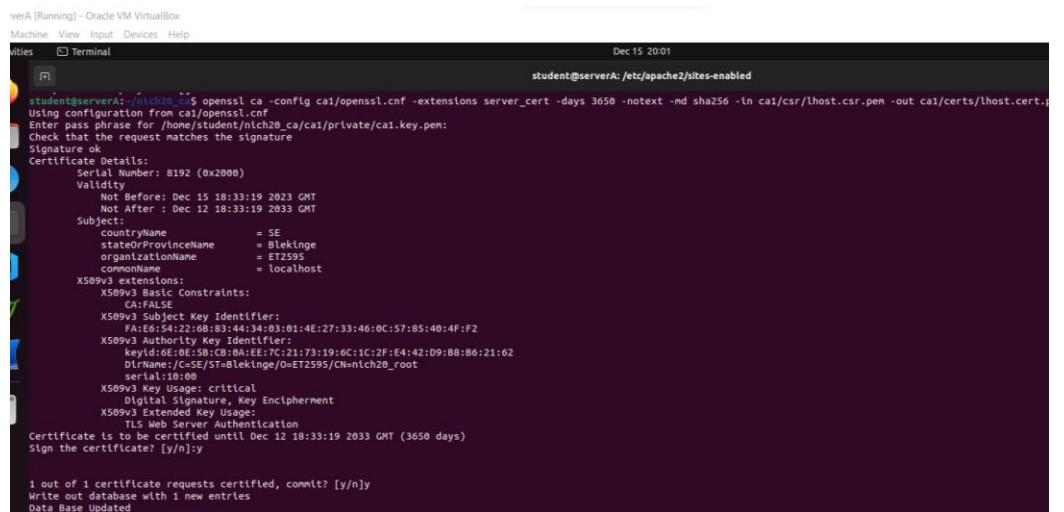


A [Running] - Oracle VM VirtualBox  
File View Input Devices Help  
student@serverA: ~\$ openssl genrsa -out ca1/private/lhost.key.pem 2048
student@serverA: ~\$ openssl req -config ca1/openssl.cnf -new -sha256 -key ca1/private/lhost.key.pem -out ca1/csr/lhost.csr.pem

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) [SE]:SE  
State or Province Name (full name) [Blekinge]:Blekinge  
Locality Name []:Karlskrona  
Organization Name (eg, company) [ET2595]:ET2595  
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 []:

- Ultimately, we crafted a CA1 certificate using the command provided below:

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



verA [Running] - Oracle VM VirtualBox  
File View Input Devices Help  
Activities Terminal Dec 15 20:01  
student@serverA: /etc/apache2/sites-enabled

student@serverA: ~\$ openssl ca -config ca1/openssl.cnf -extensions server\_cert -days 3650 -notext -md sha256 -in ca1/csr/lhost.csr.pem -out ca1/certs/lhost.cert.pem

Using configuration from ca1/openssl.cnf  
Enter pass phrase for /home/student/nich20\_ca/ca1/private/ca1.key.pem:  
Check that the request matches the signature  
Signature ok  
Certificate Details:  
 Serial Number: 8192 (0x2000)  
 Validity  
 Not Before: Dec 15 18:33:19 2023 GMT  
 Not After : Dec 12 18:33:19 2033 GMT  
 Subject:  
 countryName = SE  
 stateOrProvinceName = Blekinge  
 organizationName = ET2595  
 commonName = localhost  
X509v3 extensions:  
 X509v3 Basic Constraints:  
 CA:FALSE  
 X509v3 Subject Key Identifier:  
 FA:6:54:22:06:83:44:34:03:01:4E:27:33:46:0C:57:85:40:4F:F2  
 X509v3 Authority Key Identifier:  
 keyId:0E:01:98:C8:0A:EE:C7:21:73:19:6C:1C:2F:E4:42:D9:B8:86:21:62  
 Dir:/etc/pki/tls/certs/ET2595/CN=nich20\_root  
 serial:1000  
 X509v3 Key Usage: critical  
 Digital Signature, Key Encipherment  
 X509v3 Extended Key Usage:  
 TLS Web Server Authentication  
Certificate is to be certified until Dec 12 18:33:19 2033 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

- During this phase, we proceed with the validation of the Certificate Signing Request (CSR) by employing a specific command dedicated to this verification process.

```
openssl req -text -noout -verify -in ca1/csr/lhost.csr.pem
```

```

student@serverA:~/nich20_cai$ openssl req -text -noout -verify -in ca1/csr/lhost.csr.pem
Certificate request self-signature verify OK
Certificate Request:
Data:
Version: 1 (0x0)
Subject: C = SE, ST = Blekinge, O = ET2595, CN = localhost
Subject Public Key Info:
    Public Key Algorithm: rsaEncryption
        Public-Key: (2048 bit)
            Modulus:
                00:b9:c7:f4:08:23:78:21:90:ca:1e:0a:17:64:59:
                87:a4:97:62:f4:5d:87:10:4e:f5:64:4a:a4:85:99:
                a5:df:63:aa:48:f8:94:0a:01:7a:45:13:51:db:44:
                ac:34:99:6e:42:28:b2:d7:cdb:2:53:a8:0e:63:1d:
                57:4e:24:09:1e:5a:f3:75:5c:bb:2e:4e:bf:71:26:
                9d:86:61:46:6a:e7:b9:08:1e:74:7d:70:bb:71:2b:
                d2:8e:46:7f:9e:e9:ed:f8:00:6a:9f:09:98:54:d0:
                4d:5c:0f:0d:52:46:52:9b:8b:cfa:0:65:a:c:0:b6:
                a4:30:92:51:f6:9b:cb:b3:a9:c0:de:a7:d0:13:9f:
                15:92:63:15:3f:be:54:ad:be:cc:0c:2b:28:5a:bf:
                16:a6:64:02:ad:bc:3c:6e:96:86:b7:f3:11:8a:69:
                39:47:54:c1:29:5d:c2:26:c2:e3:97:91:c7:e2:ec:
                f1:c4:4e:3b:a0:34:b3:ab:c2:0a:89:27:84:45:60:
                e3:db:db:b0:25:a8:9b:82:1b:6a:7c:1f:32:2f:86:
                37:30:dc:aa:6c:75:42:af:00:0a:f7:c7:ec:55:a5:
                e3:e4:42:d5:8a:54:id:28:b5:89:63:11:ac:19:e4:
                11:65:7a:51:de:9a:18:1d:cc:06:cb:fc:f8:38:d9:
                15:99
            Exponent: 65537 (0x10001)
        Attributes:
            (none)
        Requested Extensions:
Signature Algorithm: sha256WithRSAEncryption
Signature Value:
03:5c:4e:09:af:1c:f4:b6:ad:ad:cb:b3:31:c0:69:8d:16:34:
id:0b:56:5f:06:fb:cd:43:01:e7:1f:d8:57:f5:e2:c2:2b:2b:
f7:05:ed:f8:71:17:04:14:19:2e:56:28:74:0a:B2:72:47:43:
50:64:99:9b:25:c7:e8:a1:3d:35:32:b8:1e:52:43:9d:33:55:
6c:c3:6:3b:4d:75:6:f:e8:9a:27:74:f6:4f:04:f0:0:cb:
d2:aa:76:74:6b:b7:e0:11:47:b5:93:82:95:1c:f4:c0:c6:
ef:9f:e1:6d:2d:00:09:5e:d6:5f:d7:1b:5a:c1:ea:f6:46:2c:
fe:ef:a9:02:bd:41:53:a1:61:83:f2:92:85:71:24:04:63:72:
9a:1c:e4:dd:0e:99:64:f6:a3:06:e3:ab:12:84:87:ce:22:51:
19:72:9e:dc:46:0c:ad:de:53:98:3d:ae:08:6a:91:22:

```

- We confirm the validation of the server certificate (localhost) generated by employing a dedicated command designed for this verification process.

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

```

student@serverA:~/nich20_cai$ openssl x509 -text -noout -in ca1/certs/lhost.cert.pem
Certificate:
Data:
Version: 3 (0x2)
Serial Number: 8192 (0x2000)
Signature Algorithm: sha256WithRSAEncryption
Issuer: C = SE, ST = Blekinge, O = ET2595, CN = nich20_cai
Subject: C = SE, ST = Blekinge, O = ET2595, CN = localhost
Subject Public Key Info:
    Public Key Algorithm: rsaEncryption
        Public-Key: (2048 bit)
            Modulus:
                00:b9:c7:f4:08:23:78:21:90:ca:1e:0a:17:64:59:
                87:a4:97:62:f4:5d:87:10:4e:f5:64:4a:a4:85:99:
                a5:df:63:aa:48:f8:94:0a:01:7a:45:13:51:db:44:
                ac:34:99:6e:42:28:b2:d7:cdb:2:53:a8:0e:63:1d:
                57:4e:24:09:1e:5a:f3:75:5c:bb:2e:4e:bf:71:26:
                9d:86:61:46:6a:e7:b9:08:1e:74:7d:70:bb:71:2b:
                d2:8e:46:7f:9e:e9:ed:f8:00:6a:9f:09:98:54:d0:
                4d:5c:0f:0d:52:46:52:9b:8b:cfa:0:65:a:c:0:b6:
                a4:30:92:51:f6:9b:cb:b3:a9:c0:de:a7:d0:13:9f:
                15:92:63:15:3f:be:54:ad:be:cc:0c:2b:28:5a:bf:
                16:a6:64:02:ad:bc:3c:6e:96:86:b7:f3:11:8a:69:
                39:47:54:c1:29:5d:c2:26:c2:e3:97:91:c7:e2:ec:
                f1:c4:4e:3b:a0:34:b3:ab:c2:0a:89:27:84:45:60:
                e3:db:db:b0:25:a8:9b:82:1b:6a:7c:1f:32:2f:86:
                37:30:dc:aa:6c:75:42:af:00:0a:f7:c7:ec:55:a5:
                e3:e4:42:d5:8a:54:id:28:b5:89:63:11:ac:19:e4:
                11:65:7a:51:de:9a:18:1d:cc:06:cb:fc:f8:38:d9:
                15:99
            Exponent: 65537 (0x10001)
X509v3 extensions:
X509v3 Basic Constraints:
    CA:FALSE
X509v3 Subject Key Identifier:
    FA:E6:54:22:0B:83:44:34:03:01:4E:27:33:46:0C:57:85:40:4FF2
X509v3 Authority Key Identifier:
    keyId:0E:0E:5B:C8:0A:EE:7C:21:73:19:6C:1C:2F:E4:42:D9:88:86:21:62
    DirName:/C=SE/ST=Blekinge/O=ET2595/CN=nich20_root
    serial:10:00
X509v3 Key Usage: critical
    Digital Signature, Key Encipherment
X509v3 Extended Key Usage:
    TLS Web Server Authentication
Signature Algorithm: sha256WithRSAEncryption

```

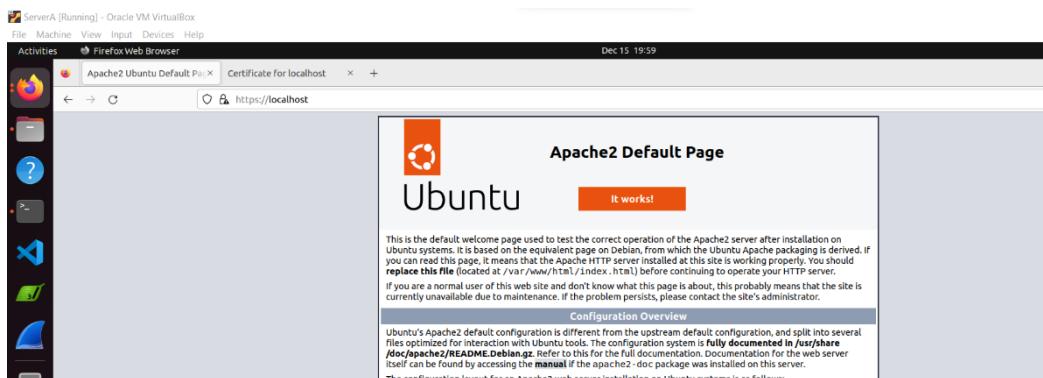
- In this stage, we verify the server certificate against the certificate chain using the specified command, as depicted in the accompanying figure.

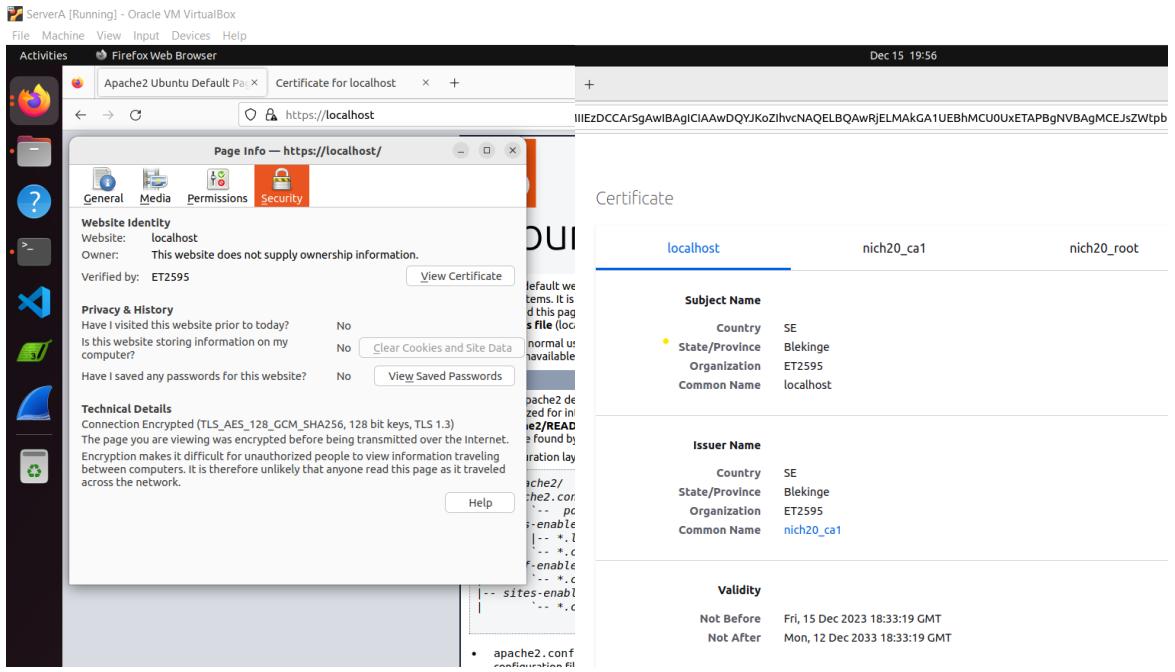
```
openssl verify -CAfile ca1/certs/ca1.cert-chain.pem ca1/certs/lhost.cert.pem
```

```
student@serverA: /etc/apache2/sites-enabled
student@serverA: /nich20_ca$ openssl verify -CAfile ca1/certs/ca1.cert-chain.pem ca1/certs/lhost.cert.pem
ca1/certs/lhost.cert.pem: OK
```

## Performing Task 12:

Once the suggested settings for the firewall browser preferences, as described in the lab manual, have been applied, the subsequent images illustrate the sequential process. These visuals serve as a guide to follow the provided steps meticulously for a comprehensive examination.





## Performing of Task 13:

The steps we executed involved the use of a sequence of specific commands to generate the Certificate Revocation List (CRL) tailored for CA1.

```
openssl ca -config ca1/openssl.cnf -gencrl -out ca1/crl/ca1.crl.pem
openssl crl -in ca1/crl/ca1.crl.pem -noout -text
```

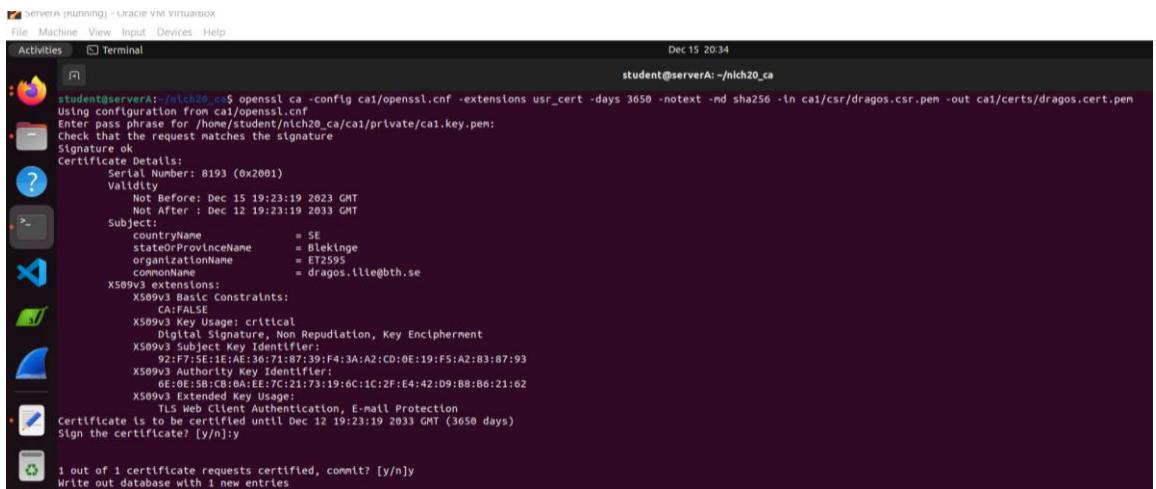
```
student@serverA:~/nich20_ca$ openssl ca -config ca1/openssl.cnf -gencrl -out ca1/crl/ca1.crl.pem
Using configuration from ca1/openssl.cnf
Enter pass phrase for /home/student/nich20_ca/ca1/private/ca1.key.pem:
student@serverA:~/nich20_ca$ openssl crl -noout -text -in ca1/crl/ca1.crl.pem
Certificate Revocation List (CRL):
Version 2 (0x1)
Signature Algorithm: sha256WithRSAEncryption
Issuer: C = SE, ST = Blekinge, O = ET2595, CN = nich20_ca1
Last Update: Dec 15 19:14:55 2023 GMT
Next Update: Jan 14 19:14:55 2024 GMT
CRL extensions:
X509v3 CRL Number:
8192
No Revoked Certificates.
Signature Algorithm: sha256WithRSAEncryption
Signature Value:
11:5c:1d:f4:cd:f2:0a:14:b3:a6:a5:7f:bc:79:d7:97:ba:45:
2a:e9:52:4a:62:74:0b:da:be:ca:2b:3d:29:c0:50:53:7c:62:
3c:3c:0:eb:05:b0:af:94:d6:fd:79:0f:c2:0:96:b8:ad:87:
20:96:a9:fa:1f:1e:a0:04:93:08:00:94:c3:42:29:2b:f0:3a:
7f:79:84:70:92:f1:be:76:4e:a3:9a:f5:db:b1:9b:d1:5e:c3:
c7:7b:94:00:a9:7b:08:49:ce:a1:c3:8c:a1:f3:1b:c0:4f:
47:07:43:90:71:7d:95:1b:be:08:f5:71:7b:00:be:1f:d9:bf:
31:37:35:fb:01:10:28:af:a1:01:04:bd:88:72:56:57:9b:
59:87:c4:b9:dc:83:1b:16:45:65:71:6d:06:0f:3c:49:68:e7:
39:03:58:db:4e:7c:90:27:7d:fa:16:cc:2f:57:ac:84:6b:
2b:b1:04:9d:54:e4:aa:ca:8d:8b:5a:ed:e4:a1:6a:d1:8f:04:
e0:85:7a:b9:05:ce:50:09:95:28:bf:f0:4e:i1:91:fb:c1:d5:
a4:90:b5:b0:9b:f7:54:24:08:0e:bb:b5:90:47:dd:5:19:02:e6:
2d:37:28:6a:85:61:30:36:fb:0d:ab:20:18:72:97:7e:3f:be:
10:78:71:f6:e3:ec:e1:7f:ba:27:46:b4:0c:6d:6a:d5:c0:94:
28:03:97:ab:f9:84:16:da:a5:9a:41:f7:75:3b:49:ff:f6:
de:4e:0c:63:78:46:68:b3:72:9b:ef:ec:7c:b6:31:da:b7:a4:
ab:cb:78:a1:c6:0d:17:ad:5a:fa:0e:ab:c8:a5:5b:bd:ed:96:
56:18:08:84:23:41:77:a2:e3:4a:cd:21:ds:8a:0f:45:12:86:
78:7b:ab:a9:0d:e6:99:a2:16:de:67:42:62:2d:7b:89:5c:
3f:5e:14:01:20:8a:4e:98:9b:df:75:cb:ee:77:68:b1:b9:16:
68:c0:e5:c2:b1:f5:4e:bd:20:c9:21:11:de:9d:c0:86:cb:b0:
b6:5f:b4:bf:d4:e8:96:fe:b9:27:ba:d2:31:41:1e:ef:c8:d2:
a5:c1:f8:ce:4c:re:1:4b:47:35:7d:43:03:fb:f1:0c:30:4a:88:
cf:f5:83:c2:f7:4e:1d:81:93:57:26:44:23:3a:90:d1:8a:1d:
38:73:f9:99:0d:1rcf:88:05:95:fa:83:b5:d8:e3:be:cf:2b:60:
82:e8:7a:85:94:84:d2:20:65:af:24:79:24:65:47:b8:24:1b:
b6:01:4f:96:64:01:23:09:e7:8c:c1:59:16:94:50:14:bc:f9:
18:5a:58:99:b4:4c:63:89
```

## Performing of Task 14 :

We employed the subsequent commands to revoke a certificate. Initially, a user certificate was generated under the name 'dragos,' following which it was revoked and subsequently removed.

```
openssl genrsa -out ca1/private/usrkey.key.pem 2048
```

```
openssl req -config ca1/openssl.cnf -new -sha256 -key ca1/private/dragos.key.pem -out ca1/csr/dragos.csr.pem
```



The screenshot shows a terminal window titled "serverA (running) - Oracle VM VirtualBox". The command entered is:

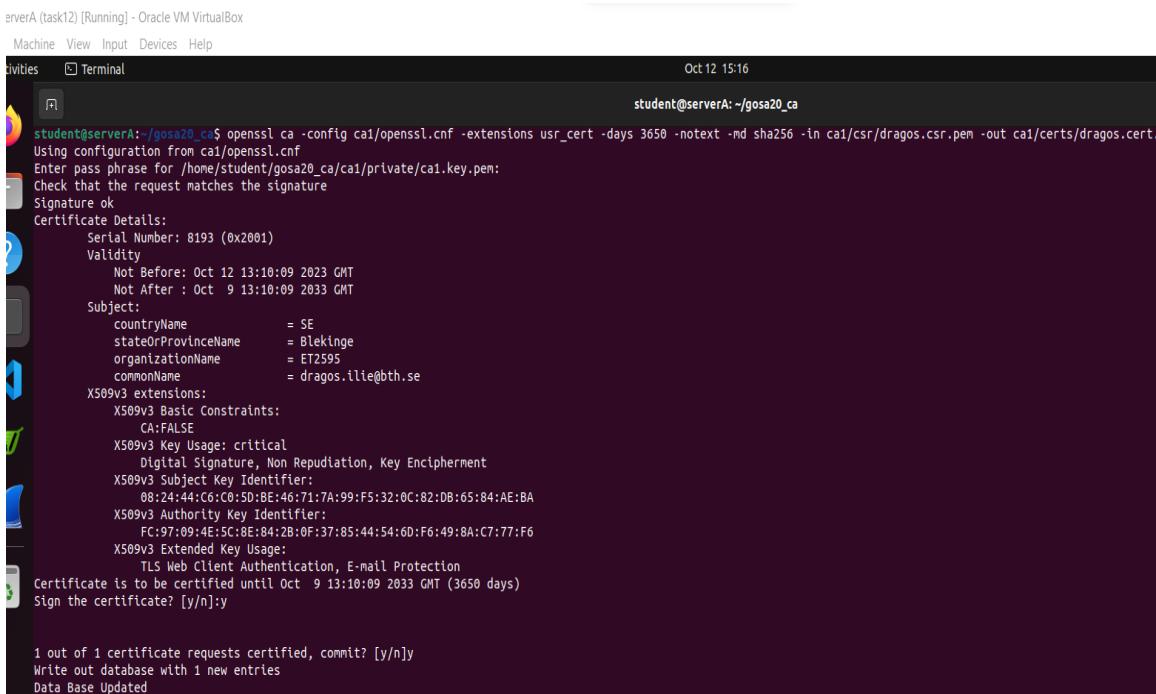
```
student@serverA:~/nich20_ca$ openssl ca -config ca1/openssl.cnf -extensions usr_cert -days 3650 -notext -md sha256 -in ca1/csr/dragos.csr.pem -out ca1/certs/dragos.cert.pem
```

The output displays the certificate details, including the serial number (8193), validity period (Dec 15 19:23:19 2023 GMT to Dec 12 19:23:19 2033 GMT), subject information (countryName=SE, stateOrProvinceName=Blekinge, organizationName=ET2595, commonName=dragos.ilie@bth.se), and various X509v3 extensions and constraints.

At the end of the process, the terminal asks:

```
Sign the certificate? [y/n]:y  
1 out of 1 certificate requests certified, commit? [y/n]  
Write out database with 1 new entries
```

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



The screenshot shows a terminal window titled "serverA (task12) [Running] - Oracle VM VirtualBox". The command entered is:

```
student@serverA:~/gosa20_ca$ openssl ca -config ca1/openssl.cnf -extensions usr_cert -days 3650 -notext -md sha256 -in ca1/csr/dragos.csr.pem -out ca1/certs/dragos.cert
```

The output displays the certificate details, identical to the previous one, but with a different timestamp (Oct 12 13:10:09 2023 GMT to Oct 9 13:10:09 2033 GMT). The subject information remains the same.

At the end of the process, the terminal asks:

```
Sign the certificate? [y/n]:y  
1 out of 1 certificate requests certified, commit? [y/n]  
Write out database with 1 new entries  
Data Base Updated
```

```
openssl req -text -noout -verify -in ca1/csr/usrkey.csr.pem
```

```

File Machine View Input Devices Help
Activities Terminal Dec 15 20:35
student@serverA:~/nich20_ca$ openssl req -text -noout -verify -in ca1/csr/dragos.csr.pem
Certificate request self-signature verify OK
Certificate Request:
Data:
    Version: 1 (0x0)
    Subject: C = SE, ST = Blekinge, O = ET2595, CN = dragos.ilie@bth.se
    Subject Public Key Info:
        Public Key Algorithm: rsaEncryption
            Public-Key: (2048 bit)
                Modulus:
                    00:ed:8f:82:f2:b4:eb:c8:ef:8c:16:95:7c:c6:df:
                    b2:3c:61:2f:11:65:5d:cd:91:61:bd:1d:c3:75:37:
                    19:ea:a5:eb:cb:64:b6:85:e3:4e:74:b6:29:73:b4:
                    67:71:8f:5c:12:80:a4:b6:6f:90:20:b1:e5:5c:b9:
                    c9:34:91:65:40:5b:c1:c0:6b:13:4f:56:87:5b:82:
                    55:ab:23:e6:43:77:f0:51:68:eb:6b:80:51:54:f5:
                    26:2e:3b:ae:58:ca:13:51:55:e5:ef:fb:fc:d1:6d:
                    86:a7:57:7b:5c:f9:35:0a:ea:84:fb:1c:a0:82:76:
                    85:f9:02:b3:9d:3e:59:d6:4f:0a:dc:fc:8a:32:9f:
                    fc:7d:7b:28:05:3e:e2:b1:ea:a5:f3:b3:a8:34:4a:
                    be:89:9e:fd:f1:82:92:cb:f8:ea:07:54:c2:53:12:
                    8e:28:a9:a4:d7:4b:04:16:cc:ff:d4:44:bf:b7:38:
                    52:04:54:24:50:29:10:58:63:40:c8:7f:71:id:ed:
                    8c:a5:ea:48:68:ac:c9:18:b2:33:47:fd:53:78:99:
                    8d:a1:c3:4f:ad:2f:35:46:7d:7e:4:39:ce:08:52:
                    d5:4e:b7:1f:4d:ad:f1:4d:e7:3f:30:f4:0e:ac:d6:
                    49:03:58:bd:db:3c:1e:47:9a:d5:47:73:6f:2b:a4:
                    fe:2f
                Exponent: 65537 (0x10001)
        Attributes:
            (none)
        Requested Extensions:
    Signature Algorithm: sha256WithRSAEncryption
Signature Value:
    c7:dd:c0:85:1d:9f:66:4d:53:97:40:a4:22:a0:35:52:63:4c:
    7c:6c:69:f1:a2:a9:7e:eb:68:65:23:69:14:a0:02:69:ea:2e:
    c1:df:fe:7:98:33:2f:ec:0d:b5:3e:65:3b:10:a3:c3:03:12:8f:
    96:4d:55:1e:d1:d0:5a:86:93:1f:02:af:5e:c6:8f:7d:53:c2:
    db:81:f2:a9:6f:fa:23:09:4f:0d:rc:03:a4:fc:90:8e:25:be:
    dd:c2:f1:85:40:aa:69:e2:ed:29:71:42:1:a:f0:68:cc:c8:81:
    7b:6e:d2:58:7e:db:cb:d1:e4:2f:39:7e:2b:59:09:0c:9a:9f:
    2a:1b:52:8a:e4:11:94:18:39:04:74:6b:ff:4d:89:61:fd:d2:
    cc:3f:c8:b3:f4:c4:a5:62:0c:13:d3:a4:41:a0:4b:ff:e9:60:
    aa:39:6a:d3:48:1d:c2:bc:50:19:71:11:83:2e:bc:a9:53:03:
    58:3b:60:b6:bc:f5:eb:a0:c7:d8:41:81:94:d6:a1:73:f7:27:
    38:1f:1f:93:11:06:35:01:eb:f2:7a:c3:70:23:04:a8:35:3e:

```

`openssl x509 -text -noout -in ca1/certs/usrkey.cert.pem`  
`openssl verify -CAfile ca1/certs/ca1.cert-chain.pem ca1/certs/usrkey.cert.pem`

To revoke we used:

`openssl ca -config ca1/openssl.cnf -revoke ca1/certs/usrkey.cert.pem`

```

ServerA [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal Dec 15 20:36
student@serverA:~/nich20_ca$ openssl x509 -text -noout -in ca1/certs/dragos.cert.pem
Certificate:
Data:
    Version: 3 (0x2)
    Serial Number: 8193 (0x2001)
    Signature Algorithm: sha256WithRSAEncryption
    Issuer: C = SE, ST = Blekinge, O = ET2595, CN = nich20_ca1
    Validity
        Not Before: Dec 15 19:23:19 2023 GMT
        Not After : Dec 12 19:23:19 2033 GMT
    Subject: C = SE, ST = Blekinge, O = ET2595, CN = dragos.ilie@bth.se
    Subject Public Key Info:
        Public Key Algorithm: rsaEncryption
            Public-Key: (2048 bit)
                Modulus:
                    00:ed:8f:82:f2:b4:eb:c8:ef:8c:16:95:7c:c6:df:
                    b2:3c:61:2f:11:65:5d:cd:91:61:bd:1d:c3:75:37:
                    19:ea:a5:eb:cb:64:b6:85:e3:4e:74:b6:29:73:b4:
                    67:71:8f:5c:12:80:a4:b6:6f:90:20:b1:e5:5c:b9:
                    c9:34:91:65:40:5b:c1:c0:6b:13:4f:56:87:5b:82:
                    55:ab:23:e6:43:77:f0:51:68:eb:6b:80:51:54:f5:
                    26:2e:3b:ae:58:ca:13:51:55:e5:ef:fb:fc:d1:6d:
                    86:a7:57:7b:5c:f9:35:0a:ea:84:fb:1c:a0:82:76:
                    85:f9:02:b3:9d:3e:59:d6:4f:0a:dc:fc:8a:32:9f:
                    fc:7d:7b:28:05:3e:e2:b1:ea:a5:f3:b3:a8:34:4a:
                    be:89:9e:fd:f1:82:92:cb:f8:ea:07:54:c3:53:12:
                    8e:28:a9:ad:7d:4b:b4:16:cc:ff:d4:34:bf:b7:38:
                    52:04:54:24:50:29:10:58:63:40:c8:7f:71:id:ed:
                    8c:a5:ea:48:68:ac:c9:18:b2:33:47:fd:53:78:99:
                    8d:a1:c3:4f:ad:2f:35:46:7d:7e:ed:39:ce:08:52:
                    d5:4e:b7:1f:4d:ad:f1:4d:e7:3f:30:f4:0e:ac:d6:
                    49:03:58:bd:db:3c:1e:47:9a:d5:47:73:6f:2b:a4:
                    fe:2f
                Exponent: 65537 (0x10001)
X509v3 extensions:
X509v3 Basic Constraints:
    CA:FALSE
    X509v3 Key Usage: critical
        Digital Signature, Non Repudiation, Key Encipherment
    X509v3 Subject Key Identifier:
        92:F7:5E:1E:AE:3E:71:87:39:F4:3A:2:CD:0E:19:F5:A2:B3:87:93
    X509v3 Authority Key Identifier:
        6E:0E:5B:CB:0A:EE:7C:21:73:19:6C:1C:2F:E4:42:D9:BB:B6:21:62
    X509v3 Extended Key Usage:
        TLS Web Client Authentication, E-mail Protection
    Signature Algorithm: sha256WithRSAEncryption
Signature Value:

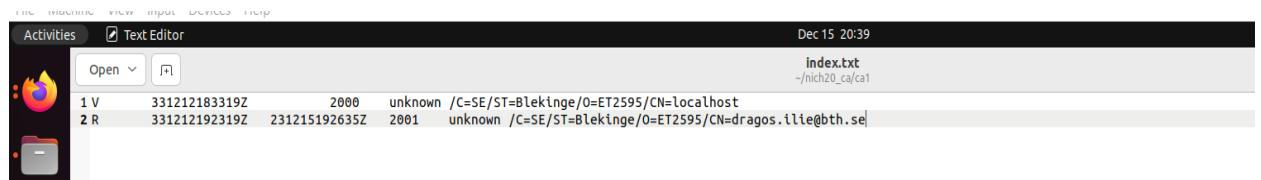
```

```

0E:0E:5B:CB:0A:E7C:21:73:19:6C:1C:2F:E4:42:D9:B8:B6:21:62
X509v3 Extended Key Usage:
    TLS Web Client Authentication, E-mail Protection
Signature Algorithm: sha256WithRSAEncryption
Signature Value:
db:74:02:10:a4:fa:5e:3a:7e:ad:cd:8e:d1:3d:bb:03:0f:5d:
6b:7c:1c:95:80:be:0c:3a:f5:dd:84:e9:ce:b1:2a:e7:f5:8a:
e8:7a:51:60:9d:c5:bf:55:6f:6b:ec:7b:59:7b:b5:08:85:8a:
a9:6e:48:0d:64:b3:ae:0d:7c:e5:6e:0f:08:51:0e:3f:02:
9a:d3:d6:81:5a:38:e0:4d:9a:7e:6f:57:2d:65:e4:d6:0e:02:
fc:bi:1a:el:32:52:72:5e:59:dd:34:f4:62:23:fe:e7:8d:35:
e2:d3:f9:4c:40:41:de:94:82:01:4a:1a:b2:40:b3:67:1d:d2:
90:9f:ee:33:30:8c:ff:19:d7:d4:12:ce:21:be:9c:c4:f1:17:27:
94:93:ef:0e:42:57:18:35:53:09:73:bf:00:b1:9:b6:el:e0:
f7:65:33:23:T5:9a:ad:f3:c0:b8:a9:4a:00:c0:b0:43:2a:d8:
68:22:25:73:6b:4d:ae:de:21:2f:3d:7a:f4:35:61:96:51:da:
2b:4a:74:de:99:4a:9e:a5:f4:42:2e:c3:39:f7:1b:e9:df:82:
c1:af:83:04:bf:22:12:73:58:23:a0:57:e7:d3:7a:a0:da:3b:
7f:ea:08:5c:d3:d3:f4:86:85:bb:8e:7e:60:a8:7d:cc:09:08:
e0:30:d3:3b:23:ed:36:9d:61:c9:93:f4:07:40:f4:64:f3:39:
b5:31:ef:74:4e:eb:2d:de:0:d9:e2:f1:b1:30:95:c2:23:63:aa:
8b:10:61:49:az:69:1e:34:94:4b:76:ad:77:44:83:45:82:d7:
12:ea:0d:f5:eb:42:3d:fe:68:1:cd:f3:3a:46:64:9c:9d:ab:
9a:ba:a6:05:d3:6e:eb:fd:3:e8:44:9c:59:4b:0f:1a:9b:57:
e6:06:1e:23:cb:83:6a:21:81:ef:43:ef:f3:2c:14:47:35:45:
5e:29:fd:ds:ae:f5:11:dd:dc:ca:6e:44:68:ff:63:cd:72:7a:
bd:05:c:f:c:dd:70:89:al:8a:74:2c:4b:80:30:9d:aa:15:
c3:81:50:05:46:10:44:1f:21:c0:35:c0:0e:fb:6b:85:dec:0:
a1:c:f:ob:5f:29:c5:57:9c:10:b6:7f:d5:2e:a5:ee:c2:c1:8:af:
83:5c:63:bb:62:bb:7f:c:25:71:66:1:db:81:c5:27:3b:70:
1b:33:48:48:3c:49:d9:df:40:0:3c:3:2a:66:dd:db:56:db:c:f:df:
85:a0:ef:9:22:47:d2:c7:f1:d0:0c:55:6d:08:dd:4c:78:32:c3:
f3:94:ef:ef:48:14:2:7c:67:dd:36:e3:0b:ao:cc:40:51:21:b9:
60:68:f1:76:49:d4:7c:40
student@serverA: ~/nich20_ca$ openssl verify -CAfile cai/certs/ca1.cert-chain.pem cai/certs/dragos.cert.pem
ca1/certs/dragos.cert.pem: OK
student@serverA: ~/nich20_ca$ openssl ca -config cai/openssl.cnf -revoke cai/certs/dragos.cert.pem
Using configuration from cai/openssl.cnf
Enter pass phrase for /home/student/nich20_ca/ca1/private/ca1.key.pem:
Revoking Certificate 2001.
Data Base Updated

```

In the 'index.txt' file, as you navigate through its contents, you'll readily discern the distinct marker denoting the revoked certificate—specifically represented by the letter 'R.' This symbol serves as a clear indication within the file structure, signifying the certificate's revocation status.



By regenerating the Certificate Revocation List (CRL), it becomes possible to access information about the revoked certificate.

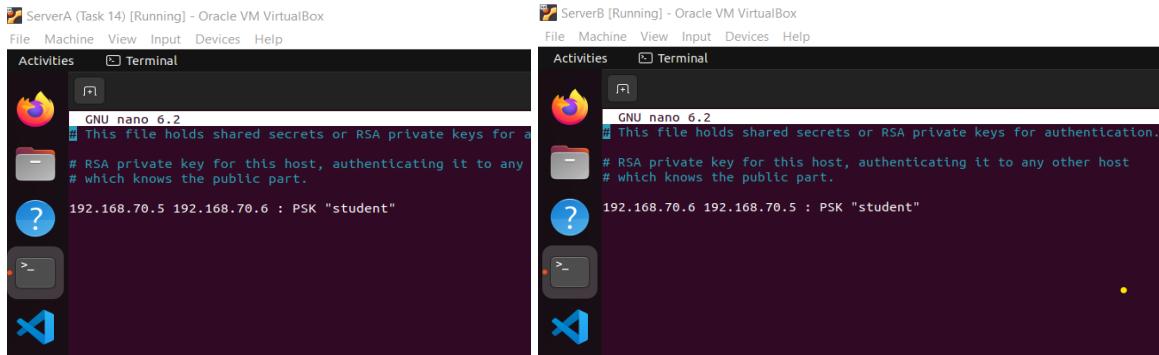
```

Activities Terminal Dec 15 20:38
student@serverA: ~/nich20_ca
student@serverA: ~/nich20_ca$ openssl ca -config cai/openssl.cnf -gencrl -out cai/crl/ca1.crl.pem
Using configuration from cai/openssl.cnf
Enter pass phrase for /home/student/nich20_ca/ca1/private/ca1.key.pem:
student@serverA: ~/nich20_ca$ openssl crl -noout -text -in cai/crl/ca1.crl.pem
Certificate Revocation List (CRL):
Version 2 (0x1)
Signature Algorithm: sha256WithRSAEncryption
Issuer: C = SE, ST = Blekinge, O = ET2595, CN = nich20_ca1
Last Update: Dec 15 19:28:01 2023 GMT
Next Update: Jan 14 19:28:01 2024 GMT
CRL extensions:
X509v3 CRL Number:
8193
Revoked Certificates:
Serial Number: 2001
    Revocation Date: Dec 15 19:26:35 2023 GMT
    Signature Algorithm: sha256WithRSAEncryption
    Signature Value:
        64:e6:44:07:0a:ab:f6:f5:29:2f:72:62:14:f3:be:df:5c:6d:
        39:d6:45:23:d5:04:30:c0:de:09:5e:48:01:ee:5:bc:cf:04:
        0a:t6:69:7f:f8:80:f1:ab:de:fd:ff:04:f5:0a:40:fd:c2:12:
        21:61:2f:ef:42:f1:20:9a:b5:92:c5:f6:ad:ba:0e:37:fa:49:
        2f:71:bb:c2:fe:66:6f:94:22:46:3d:df:b5:la:eb:2a:92:66:
        a2:3b:44:72:48:df:ba:df:58:ee:92:6e:76:56:3a:62:c0:b3:
        64:a8:99:fb:26:a2:54:72:cc:df:0b:11:dd:07:7b:8a:19:7a:
        bc:d3:59:bd:5c:b2:96:ee:1b:bb:8f:0d:52:f7:1a:77:1a:fb:
        b4:b4:3f:9b:60:f9:90:0a:1b:64:77:ds:46:15:a3:63:98:8d:
        99:82:bb:99:ee:7f:ad:53:ea:fc:02:ee:43:0f:91:45:16:91:
        f6:db:31:83:49:4f:06:29:ba:46:a3:dc:97:58:62:86:75:70:
        15:a9:96:7c:a8:91:31:48:2e:30:4d:4e:7a:2:ca:5f:fa:f0:1e:
        a0:ea:89:f1:80:ba:ae:1:1d:48:9b:ba:c1:dc:6f:92:6f:30:
        2c:fa:b7:71:71:d0:08:87:16:00:dd:a9:df:c9:46:0c:86:66:
        80:b7:3d:17:4f:9b:e2:0f:78:f9:09:58:cd:e2:f4:2e:1c:97:
        0f:48:56:15:33:d1:bf:6d:8a:a4:0e:cb:ds:12:c1:f4:de:e0:
        d1:bf:bf:d9:3b:96:4f:9a:68:be:1a:5c:32:e4:62:71:18:0e:
        d0:82:9e:92:7e:d4:2a:9b:47:bd:f3:fa:0f:5c:06:e3:32:08:
        b8:1e:18:b2:14:79:bd:74:9b:1e:53:21:cd:eb:ab:7d:5b:77:
        f3:f3:cd:04:fe:72:d2:38:f5:60:ee:84:49:fe:f9:fa:b4:26:
        50:c7:2c:01:9d:5c:25:fe:94:54:67:a2:44:99:13:e5:3e:8d:
        4f:23:52:77:ee:08:57:42:3b:57:92:08:1a:1f:3d:9e:24:0b:
        7c:09:96:68:63:f1:35:0d:7b:9f:e5:df:d7:a3:cf:35:6d:29:
```

## Performing of Task 15:

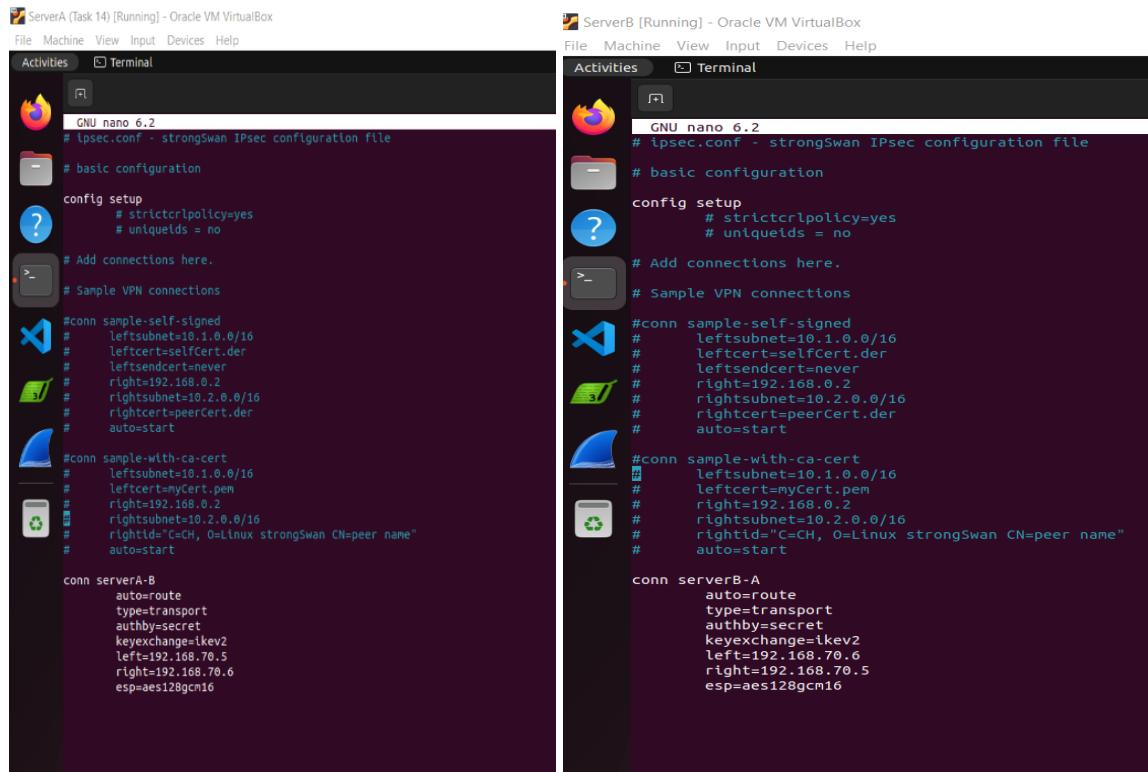
To facilitate this specific task, we configured Server A and Server B. And implementing specific adjustments within the `ipsec.conf` and `ipsec.secrets` files. These modifications were aimed at establishing a host-to-host transport VPN utilizing (PSK) and IKEv2 protocols. The details of these configurations are visually depicted in the accompanying figures for reference and verification.

The instructions listed below are part of the configuration in the Server A `ipsec.conf` file, The instructions listed below are part of the configuration in the Server B `ipsec.conf` file:



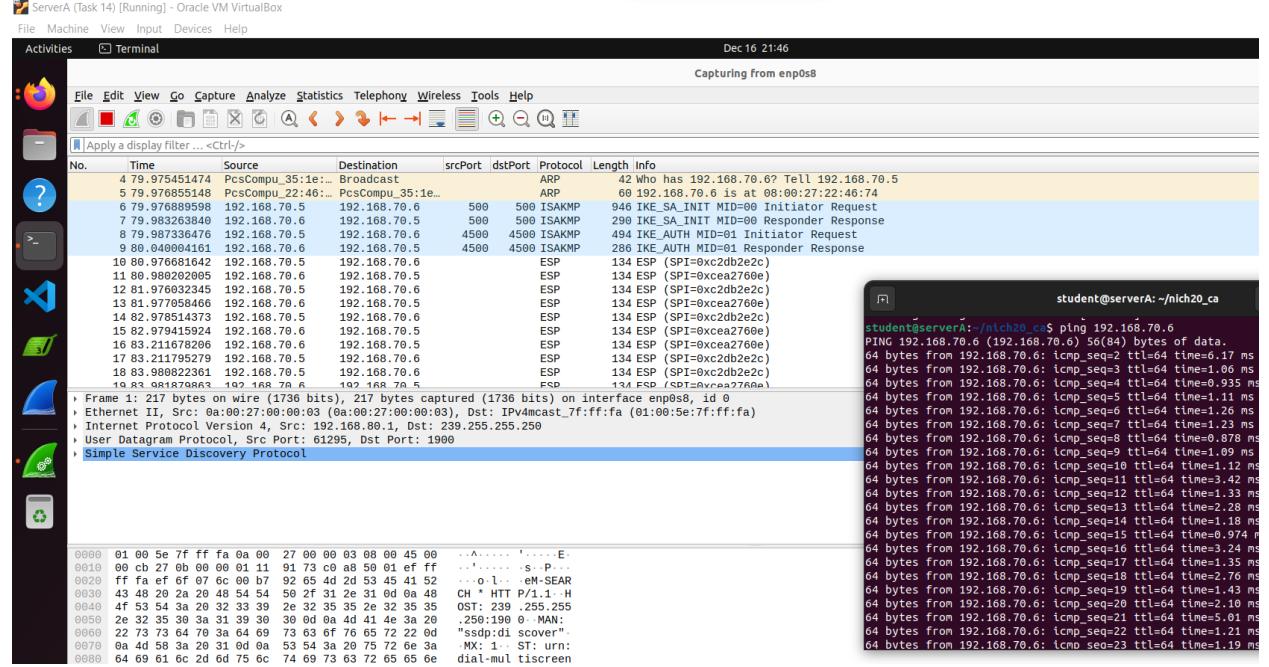
ServerA (Task 14) [Running] - Oracle VM VirtualBox  
File Machine View Input Devices Help  
Activities Terminal  
GNU nano 6.2  
# This file holds shared secrets or RSA private keys for authentication.  
# RSA private key for this host, authenticating it to any other host  
# which knows the public part.  
192.168.70.5 192.168.70.6 : PSK "student"  
  
ServerB [Running] - Oracle VM VirtualBox  
File Machine View Input Devices Help  
Activities Terminal  
GNU nano 6.2  
# This file holds shared secrets or RSA private keys for authentication.  
# RSA private key for this host, authenticating it to any other host  
# which knows the public part.  
192.168.70.6 192.168.70.5 : PSK "student"

The instructions provided below represent the content incorporated within the configuration of the Server A `ipsec.secrets` file. The instructions provided below represent the content incorporated within the configuration of the Server B `ipsec.secrets` file.



ServerA (Task 14) [Running] - Oracle VM VirtualBox  
File Machine View Input Devices Help  
Activities Terminal  
GNU nano 6.2  
# ipsec.conf - strongSwan IPsec configuration file  
# basic configuration  
config setup  
# strictcrlpolicy=yes  
# uniqueids = no  
# Add connections here.  
# Sample VPN connections  
#conn sample-self-signed  
# leftsubnet=10.1.0.0/16  
# lefcert= selfcert.der  
# leftsendcert=never  
# right=192.168.0.2  
# rightsubnet=10.2.0.0/16  
# rightcert= peerCert.der  
# auto=start  
#conn sample-wth-ca-cert  
# leftsubnet=10.1.0.0/16  
# leftcert= myCert.pem  
# right=192.168.0.2  
# rightsubnet=10.2.0.0/16  
# rightcert= peerCert.der  
# auto=start  
conn serverA-B  
auto=route  
type=transport  
authby=secret  
keyexchange=ikev2  
left=192.168.70.5  
right=192.168.70.6  
esp=aes128gcm16  
  
ServerB [Running] - Oracle VM VirtualBox  
File Machine View Input Devices Help  
Activities Terminal  
GNU nano 6.2  
# ipsec.conf - strongSwan IPsec configuration file  
# basic configuration  
config setup  
# strictcrlpolicy=yes  
# uniqueids = no  
# Add connections here.  
# Sample VPN connections  
#conn sample-self-signed  
# leftsubnet=10.1.0.0/16  
# lefcert= selfcert.der  
# leftsendcert=never  
# right=192.168.0.2  
# rightsubnet=10.2.0.0/16  
# rightcert= peerCert.der  
# auto=start  
#conn sample-wth-ca-cert  
# leftsubnet=10.1.0.0/16  
# leftcert= myCert.pem  
# right=192.168.0.2  
# rightsubnet=10.2.0.0/16  
# rightcert= peerCert.der  
# auto=start  
conn serverB-A  
auto=route  
type=transport  
authby=secret  
keyexchange=ikev2  
left=192.168.70.6  
right=192.168.70.5  
esp=aes128gcm16

The recorded network traffic, encompassing ISKAMP packets, has been meticulously captured and documented through the Wireshark application specifically on Server A. This comprehensive data provides insight into the exchanged ISKAMP packets within the network environment, facilitating detailed analysis and troubleshooting, if necessary.



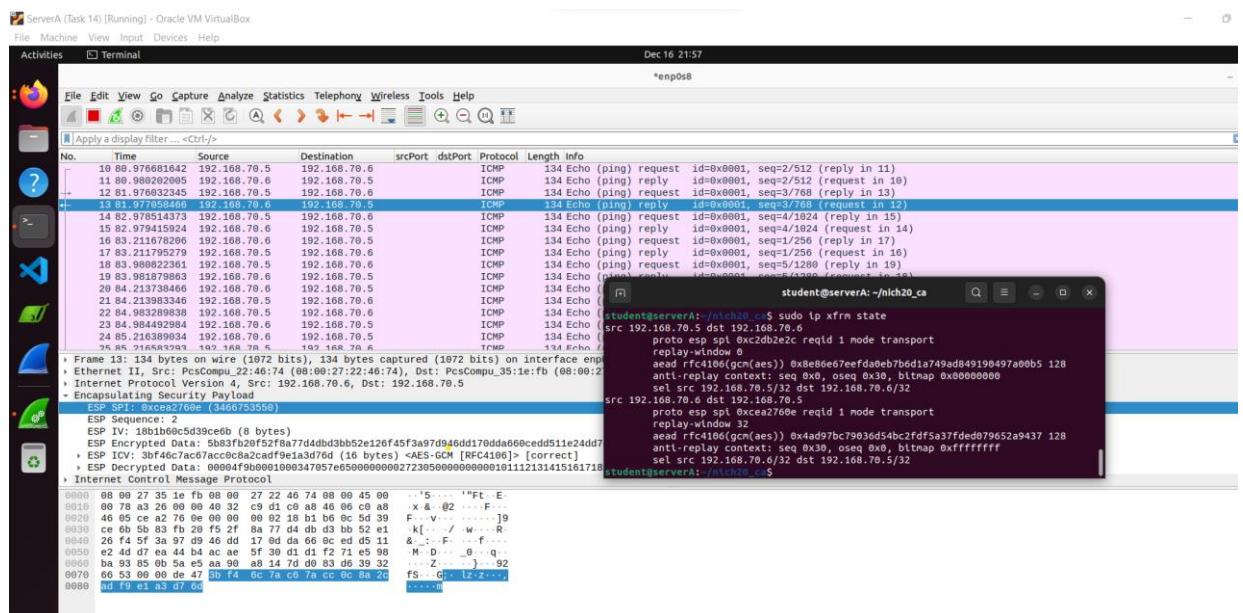
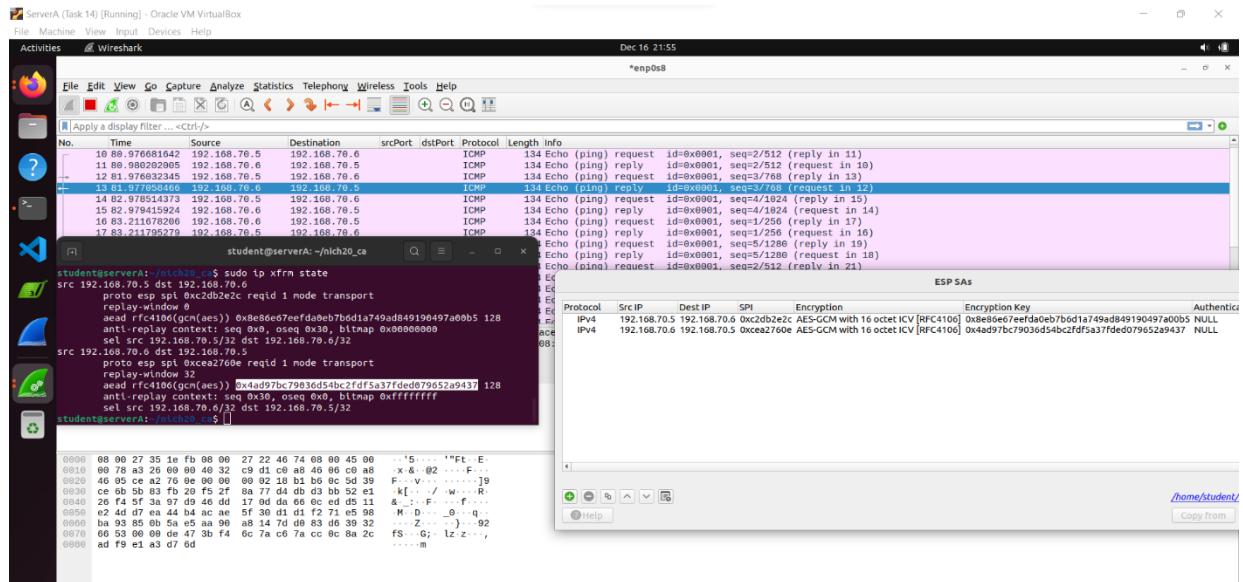
```
student@serverA: ~/nich20_ca
[1]+ Stopped ping 192.168.70.6
student@serverA: ~/nich20_ca$ sudo ipsec statusall
Status of IKE charon daemon (strongSwan 5.9.5, Linux 5.15.0-56-generic, x86_64):
  uptime: 32 minutes, since Dec 16 21:12:55 2023
  malloc: sbrk 2912256, mmap 0, used 1298208, free 1614048
  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 sha1 sha2 md5 mgf1 random nonce x509 revocation constraints pubkey pkcs1 pkcs7 pkcs8 pkcs12 ppg dnskey ss
  hkey pem openssl fips-prf gmp agent xcbc hmac gcm drbg attr kernel-netlink resolve socket-default connmark stroke updown eap-mschapv2 xauth-gene
  ric counters
  Listening IP addresses:
    192.168.60.100
    192.168.70.5
    10.0.4.15
  Connections:
    serverA-to-serverB: 192.168.70.5...192.168.70.6 IKEv2
    serverA-to-serverB: local: [192.168.70.5] uses pre-shared key authentication
    serverA-to-serverB: remote: [192.168.70.6] uses pre-shared key authentication
    serverA-to-serverB: child: dynamic == dynamic TRANSPORT
  Routed Connections:
    serverA-to-serverB[1]: ROUTED, TRANSPORT, reqid 1
    serverA-to-serverB[1]: 192.168.70.5/32 === 192.168.70.6/32
  Security Associations (1 up, 0 connecting):
    serverA-to-serverB[1]: ESTABLISHED 30 minutes ago, 192.168.70.5[192.168.70.5]...192.168.70.6[192.168.70.6]
    serverA-to-serverB[1]: IKEv2 SPIs: 17aa3e2cea3bf2d1* 8249adf265d95a90_r, pre-shared key reauthentication in 2 hours
    serverA-to-serverB[1]: IKE proposal: AES_CBC_128/HMAC_SHA2_256_128/PRF_AES128_XCB/CURVE_25519
    serverA-to-serverB[2]: INSTALLED, TRANSPORT, reqid 1, ESP SPIs: cea2760e_i c2db2e2c_o
    serverA-to-serverB[2]: AES GCM 16 128, 3072 bytes_i (48 pkts, 1783s ago), 3072 bytes_o (48 pkts, 1783s ago), rekeying in 14 minutes
    serverA-to-serverB[2]: 192.168.70.5/32 === 192.168.70.6/32
```

## Performing of Task 16:

For accessing details regarding established connections, such as Security Parameter Indexes (SPIs), encryption and authentication keys, and related information, we employ the following command:

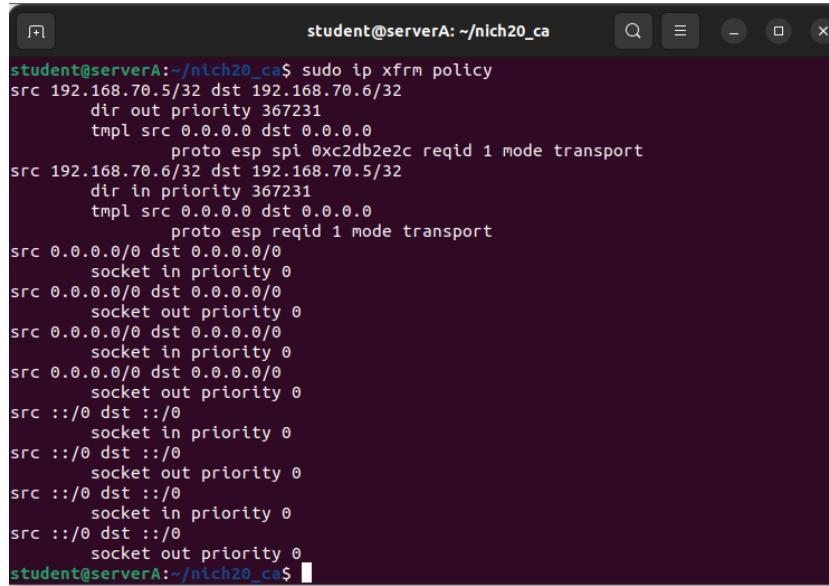
*sudo ip xfrm state*

```
student@serverA:~/nich20_ca$ sudo ip xfrm state
src 192.168.70.5 dst 192.168.70.6
    proto esp spi 0xc2db2e2c reqid 1 mode transport
    replay-window 32
    aead rfc4106(gcm(aes)) 0x8e86e67eefda0eb7b6d1a749ad849190497a00b5 128
    anti-replay context: seq 0x30, oseq 0x30, 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 0xcea2760e reqid 1 mode transport
    replay-window 32
    aead rfc4106(gcm(aes)) 0x4ad97bc79036d54bc2fdf5a37fde079652a9437 128
    anti-replay context: seq 0x30, oseq 0x0, bitmap 0xffffffff
    sel src 192.168.70.6/32 dst 192.168.70.5/32
student@serverA:~/nich20_ca$
```



## Performing of Task 17:

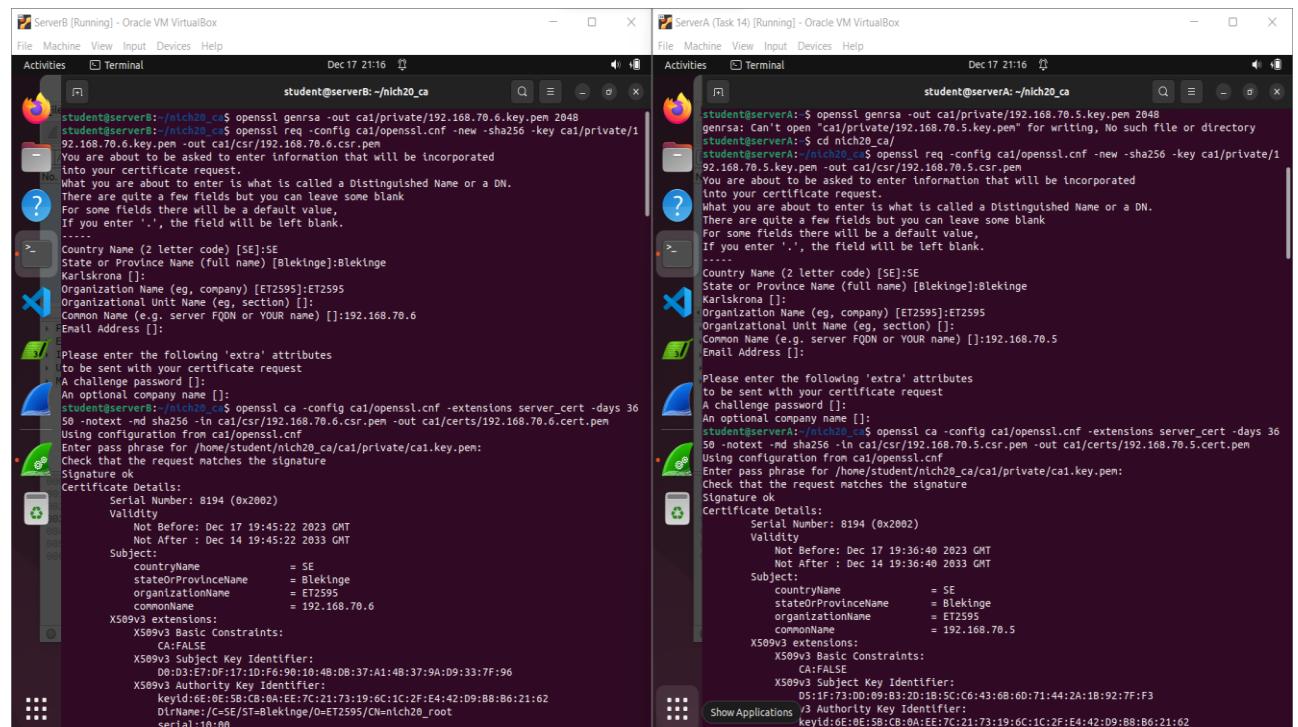
When executing the command 'sudo ip xfrm policy,' it generates the subsequent output:



```
student@serverA:~/nich20_ca$ 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 0xc2db2e2c 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:~/nich20_ca$
```

## Performing of Task 18:

Certificates have been crafted for Server A and Server B employing a methodology reminiscent of the techniques employed in previous tasks, drawing parallels with the procedures outlined in both Task 11 and Task 14. The generation of these certificates for the respective servers was conducted through a similar approach, ensuring compliance with established protocols and standards.



The image shows two side-by-side terminal windows from Oracle VM VirtualBox. Both windows are titled 'Activities' and 'Terminal'. The left window is for 'ServerB [Running]' and the right is for 'ServerA (Task 14) [Running]'. Both windows show the same sequence of commands being run by the user 'student'.

```
student@serverB:~/nich20_ca$ openssl genrsa -out ca1/private/192.168.70.6.key.pem 2048
student@serverB:~/nich20_ca$ openssl req -config ca1openssl.cnf -new -sha256 -key ca1/private/192.168.70.6.key.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) [SE]:SE
State or Province Name (full name) [Blekinge]:Blekinge
Karlskrona []
Organization Name (eg, company) [ET2595]:ET2595
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@serverB:~/nich20_ca$ openssl ca -config ca1openssl.cnf -extensions server_cert -days 3650 -notext -md sha256 -in ca1/csr/192.168.70.6.csr.pem -out ca1/certs/192.168.70.6.cert.pem
Using configuration from ca1openssl.cnf
Enter pass phrase for /home/student/nich20_ca/ca1/private/ca1.key.pem:
Check that the request matches the signature
Signature ok
Certificate Details:
    Serial Number: 8194 (0x2002)
    Validity
        Not Before: Dec 17 19:45:22 2023 GMT
        Not After : Dec 14 19:45:22 2033 GMT
    Subject:
        countryName            = SE
        stateOrProvinceName    = Blekinge
        organizationName       = ET2595
        commonName             = 192.168.70.6
X509v3 extensions:
    X509v3 Basic Constraints:
        CA:FALSE
    X509v3 Subject Key Identifier:
        00:D3:E7:DF:17:10:F6:90:10:4B:0B:37:A1:4B:37:9A:D9:33:7F:96
    X509v3 Authority Key Identifier:
        keyid:0E:0E:5B:C8:0A:EE:7C:21:73:19:6C:1C:2F:E4:42:D9:B8:B6:21:62
        DirName:/C=SE/ST=Blekinge/O=ET2595/CN=nich20_rroot
        serial:10:00

student@serverA:~/nich20_ca$ openssl genrsa -out ca1/private/192.168.70.5.key.pem 2048
genrsa: Can't open "ca1/private/192.168.70.5.key.pem" for writing, No such file or directory
student@serverA:~/nich20_ca$ cd nich20_ca/
student@serverA:~/nich20_ca$ openssl req -config ca1openssl.cnf -new -sha256 -key ca1/private/192.168.70.5.key.pem -out ca1/csr/192.168.70.5.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) [SE]:SE
State or Province Name (full name) [Blekinge]:Blekinge
Karlskrona []
Organization Name (eg, company) [ET2595]:ET2595
Organizational Unit Name (eg, section) []
Common Name (e.g. server FQDN or YOUR name) []:192.168.70.5
Email Address []

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []
An optional company name []
student@serverA:~/nich20_ca$ openssl ca -config ca1openssl.cnf -extensions server_cert -days 3650 -notext -md sha256 -in ca1/csr/192.168.70.5.csr.pem -out ca1/certs/192.168.70.5.cert.pem
Using configuration from ca1openssl.cnf
Enter pass phrase for /home/student/nich20_ca/ca1/private/ca1.key.pem:
Check that the request matches the signature
Signature ok
Certificate Details:
    Serial Number: 8194 (0x2002)
    Validity
        Not Before: Dec 17 19:36:40 2023 GMT
        Not After : Dec 14 19:36:40 2033 GMT
    Subject:
        countryName            = SE
        stateOrProvinceName    = Blekinge
        organizationName       = ET2595
        commonName             = 192.168.70.5
X509v3 extensions:
    X509v3 Basic Constraints:
        CA:FALSE
    X509v3 Subject Key Identifier:
        D5:1F:73:0D:09:B3:20:1B:5C:C6:43:6B:6D:71:44:2A:1B:92:7F:F3
    X509v3 Authority Key Identifier:
        keyid:0E:0E:5B:C8:0A:EE:7C:21:73:19:6C:1C:2F:E4:42:D9:B8:B6:21:62
        DirName:/C=SE/ST=Blekinge/O=ET2595/CN=nich20_rroot
        serial:10:00
```

```
student@serverB:~/nich20_ca$ openssl ca -config ca/OpenSSL.cnf -extensions server_cert -days 3650 -notext -md sha256 -in ca1/csr/192.168.70.6.csrf.pem -out ca1/certs/192.168.70.6.cert.pem
Using configuration from ca/OpenSSL.cnf
Enter pass phrase for /home/student/nich20_ca/ca1/private/ca1.key.pem:
Check that the request matches the signature
Signature ok
Certificate Details:
    Serial Number: 8194 (0x2002)
    Validity
        Not Before: Dec 17 19:45:22 2023 GMT
        Not After : Dec 14 19:45:22 2033 GMT
    Subject:
        countryName      = SE
        stateOrProvinceName = Blekinge
        organizationName   = ET2595
        commonName        = 192.168.70.6
X509v3 extensions:
    X509v3 Basic Constraints:
        CA:FALSE
        SubjectKeyIdentifier:
            00:31:E7:DF:17:1D:F6:90:10:4B:37:A1:B4:37:9A:D9:33:9E
    X509v3 Authority Key Identifier:
        Keyid:00:E5:8C:CB:0A:EE:7C:21:73:19:6C:1C:2F:E4:D2:D9:BB:B6:21:62
        DirName:/O=St*Blekinge/O=Et2595/CN=nich20_root
        serial:10:00
    X509v3 Key Usage: critical
        Digital Signature, Encipherment
    X509v3 Extended Key Usage:
        TLS Web Server Authentication
Certificate is to be certified until Dec 14 19:45:22 2033 GMT (3650 days)
Sign the certificate? [y/n]:y

1 out of 1 certificate requests certified, commit? [y/n]
Write out database with 1 new entries
Data Base Updated
student@serverB:~/nich20_ca$ openssl req -text -noout -verify -in ca1/csr/192.168.70.6.csrf.pem
Certificate request self-signature verify OK
Certificate Request:
    Data:
        Version: 1 (0x0)
        Subject: C = SE, ST = Blekinge, O = ET2595, CN = 192.168.70.6
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public Key: (2048 bit)
                    Modulus:
                        00:a0:be:f1:a6:10:c0:e5:55:de:cd:1f:46:e4:03:
                    Exponent:
                        01:00:01:00:00:00:00:00:00:00:00:00:00:00:00:00:
        Subject Key Identifier:
            Public Key Algorithm: rsaEncryption
            Public Key: (2048 bit)
                Modulus:
                    00:b2:19:d1:d3:86:b7:fe:3a:d3:d7:e9:92:81:ef:
```

```
student@serverB:~/nich20_ca$ openssl ca -config ca/OpenSSL.cnf -extensions server_cert -days 3650 -notext -md sha256 -in ca1/csr/192.168.70.5.csrf.pem -out ca1/certs/192.168.70.5.cert.pem
Using configuration from ca/OpenSSL.cnf
Enter pass phrase for /home/student/nich20_ca/ca1/private/ca1.key.pem:
Check that the request matches the signature
Signature ok
Certificate Details:
    Serial Number: 8194 (0x2002)
    Validity
        Not Before: Dec 17 19:36:40 2023 GMT
        Not After : Dec 14 19:36:40 2033 GMT
    Subject:
        countryName      = SE
        stateOrProvinceName = Blekinge
        organizationName   = ET2595
        commonName        = 192.168.70.5
X509v3 extensions:
    X509v3 Basic Constraints:
        CA:FALSE
        SubjectKeyIdentifier:
            05:73:0D:00:9B:3D:0B:15:C0:d3:6B:6D:71:44:2A:18:92:7F:F3
    X509v3 Authority Key Identifier:
        Keyid:00:E5:8C:CB:0A:EE:7C:21:73:19:6C:1C:2F:E4:D2:D9:BB:B6:21:62
        DirName:/O=St*Blekinge/O=Et2595/CN=nich20_root
        serial:10:00
    X509v3 Key Usage: critical
        Digital Signature, Encipherment
    X509v3 Extended Key Usage:
        TLS Web Server Authentication
Certificate is to be certified until Dec 14 19:36:40 2033 GMT (3650 days)
Sign the certificate? [y/n]:y

1 out of 1 certificate requests certified, commit? [y/n]
Write out database with 1 new entries
Data Base Updated
student@serverB:~/nich20_ca$ openssl req -text -noout -verify -in ca1/csr/192.168.70.5.csrf.pem
Certificate request self-signature verify OK
Certificate Request:
    Data:
        Version: 1 (0x0)
        Subject: C = SE, ST = Blekinge, O = ET2595, CN = 192.168.70.5
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public Key: (2048 bit)
                    Modulus:
                        00:b2:19:d1:d3:86:b7:fe:3a:d3:d7:e9:92:81:ef:
```

```
student@serverB:~/nich20_ca$ openssl req -text -noout -verify -in ca1/csr/192.168.70.6.csr.pem
Certificate request self-signature verify OK
Certificate Request:
Data:
Version: 1 (0x0)
Subject: C = SE, ST = Blekinge, O = ET2595, CN = 192.168.70.6
Subject Public Key Info:
    Public Key Algorithm: rsaEncryption
        Public-Key: (2048 bit)
            Modulus:
                00:ab:ef:a6:66:10:c0:eb:55:de:c:d1:f7:e9:40:03:
            d3:d5:45:2d:c7:71:79:f9:1a:b5:c2:c2:b2:63:9b:
            cf:08:01:90:9c:20:2d:4f:60:3d:35:3a:30:30:30:
            2d:01:3c:92:c3:2b:d6:1c:92:05:39:39:39:39:bf:
            0a:9c:d1:c6:fa:02:30:a9:b9:5b:91:0f:95:35:91:
            77:9d:f5:cd:7:93:5b:5a:c5:53:d5:5e:b3:b2:1f:6:
            e2:24:be:60:81:fe:ab:5a:5a:05:47:47:b0:b0:f1:12:
            96:77:9b:be:9a:4c:1c:1e:7b:b3:39:ac:67:37:24:
            9f:59:de:dc:b1:f9:0c:bc:c0:2f:a8:4e:c5:99:09:
            35:df:ec:21:61:b8:89:49:00:3f:3a:71:7d:02:
            ac:ea:ea:ea:ea:ea:ea:ea:ea:ea:ea:ea:ea:ea:ea:
            0b:5b:0f:53:0b:0f:53:0b:0f:53:0b:0f:53:0b:0f:
            7b:fa:b1:d6:51:0f:89:07:dc:cc:ee:76:10:85:5f:
            01:8c:76:02:89:57:2d:c5:23:4b:0d:5a:01:77:02:
            3a:69:7a:ee:bb:24:ba:b5:a5:ad:93:82:c3:76:53:f:
            c2:fb:19:56:c1:08:aa:3:11:ea:78:00:1d:bb:99:
            67:4f:69:2b:b3:59:25:51:20:64:f1:17:8a:dc:5d:
            36:09
        Exponent: 65537 (0x10001)
Attributes:
    (none)
Requested Extensions:
Signature Algorithm: sha256WithRSAEncryption
Signature Value:
30:da:49:ad:ec:32:b8:68:54:c7:94:f1:5b:e2:97:0e:2e:0f:
5b:fd:5d:81:b4:39:bc:c3:92:c8:f9:78:f7:cc:5f:cb:80:0e:
ab:00:94:ea:56:64:3a:50:55:f4:08:22:26:3c:0d:84:c3:50:
56:01:45:3a:50:55:f4:08:22:26:3c:0d:84:c3:50:56:01:
f6:68:f3:08:44:25:63:b2:b1:e5:9a:9b:1c:27:b2:39:re:4:
5a:14:29:20:13:bc:c2:97:57:00:c5:02:00:10:01:1c:6a:
13:74:60:2c:28:0b:02:e5:39:80:04:2a:4b:27:27:84:a8:
4b:5a:8c:13:42:61:ca:76:6d:03:47:f9:dcb:f2:f5:c1:a4:d9:
e2:28:b5:c9:01:90:09:65:40:2d:eb:0b:6d:36:f4:7c:3d:80:
81:72:a3:ad:31:04:5a:2f:3c:c1:18:99:da:9e:0e:31:51:19:
78:fb:7c:03:2d:2b:0b:2b:0b:2b:0b:2b:0b:2b:0b:2b:0b:2b:
d1:01:01:01:01:01:01:01:01:01:01:01:01:01:01:01:01:01:
9e:96:f6:c3:17:67:f6:f7:9c:a5:14:9e:63:82:00:79:43:c6:
34:5f:77:27:89:1h:f7:d6:56:ah:c3:df:61:9b:99:41:c3:16:
```

```
student@serverB:~/atch20_ca$ openssl x509 -text -noout -in ca1/certs/192.168.70.6.cert.pem
Certificate:
Data:
    Version: 3 (0x2)
    Serial Number: 8194 (0x2002)
    Signature Algorithm: sha256WithRSAEncryption
    Issuer: C = SE, ST = Blekinge, O = Et2595, CN = nich20_ca1
    Validity
        Not Before: Dec 17 19:45:22 2023 GMT
        Not After : Dec 14 19:45:22 2033 GMT
    Subject: C = SE, ST = Blekinge, O = Et2595, CN = 192.168.70.6
    Subject Public Key Info:
        Public Key Algorithm: rsaEncryption
            Public-Key: (2048 bit)
                Modulus:
                    00:ab:ef:a0:66:10:c0:eb:55:de:cd:1f:46:e4:03:
                    d3:d5:2d:2d:c7:71:79:ef:91:a8:c5:2c:b2:03:9b:
                    cf:08:4b:90:e7:0b:2d:1a:34:b9:96:6e:65:da:73:
                2:d1:ef:1f:59:92:3f:2c:3a:0d:1e:96:95:j7:39:bf:
                0a:dc:02:0c:f9:2a:30:ab:b9:5b:91:0a:95:35:91:
                77:9d:f5:30:1a:0b:0a:0a:0a:0a:0a:0a:0a:0a:0a:
                02:02:02:02:01:fe:ab:a5:5a:01:47:47:b0:b1:f1:
                96:77:9b:be:9a:4c:1c:1e:b7:bb:39:ac:67:37:24:
                9f:59:de:d9:b1:f9:0c:b4:c0:2f:a8:4e:c5:99:09:
                35:df:fe:21:61:b8:89:a9:0e:0d:3f:3a:17:02:
                ac:ae:aa:50:ce:08:08:00:16:16:3f:2a:ab:95:0f:0b:
                0b:5e:ef:0f:52:dd:fe:f3:00:65:53:48:2e:c4:
                7b:10:03:0d:05:01:09:09:09:09:09:09:09:09:09:
                09:09:09:09:09:09:09:09:09:09:09:09:09:09:09:
                3a:69:7a:ee:bb:24:b5:65:ad:93:82:c3:76:7d:5f:
                c2:fb:19:56:c1:08:aa:ee:11:ee:78:00:1d:bb:09:
                67:af:09:2b:b3:59:25:51:20:64:3f:17:8a:dc:5d:
                36:09
                Exponent: 65537 (0x10001)
X509v3 extensions:
X509v3 extensions Constraints:
CA:FALSE
X509v3 Subject Key Identifier:
D0:D3:E7:DF:17:10:F6:90:10:4B:DB:37:A1:4B:37:9A:D9:33:7F:96
X509v3 Authority Key Identifier:
X509v3 extensions:
X509v3 extensions Constraints:
CA:FALSE
X509v3 Subject Key Identifier:
D5:1F:73:00:09:B3:D0:1B:5C:g6:43:6B:6D:71:44:2A:1B:92:7F:F3
X509v3 Authority Key Identifier:
```

```

student@serverB: ~$ openssl verify -CAfile ca1/certs/ca1.cert-chain.pem ca1/certs/192.168.70.6.cert.pem
OK

student@serverA: ~$ openssl verify -CAfile ca1/certs/ca1.cert-chain.pem ca1/certs/192.168.70.5.cert.pem
OK

```

Subsequently, we proceeded to place the Root CA and CA1 certificates into the "/etc/ipsec.d/cacerts/" directory, while situating the newly generated certificates in the "/etc/ipsec.d/certs/" directory for both Server A and Server B. Furthermore, the private keys pertinent to these servers were securely stored in the "/etc/ipsec.d/private/" directory, ensuring their accessibility when needed for secure communication.

Before this, we took the necessary steps to validate and incorporate the updated CA certificates by executing the command "sudo ipsec rereadcacerts." Following this, we confirmed the successful recognition and integration of these certificates into the system by listing the available CA certificates through the command "sudo ipsec listcacerts." These meticulous steps were instrumental in verifying the seamless integration and recognition of the certificates within the system configuration.

```

student@serverB: ~$ sudo ipsec rereadcacerts
student@serverB: ~$ sudo ipsec listcacerts

List of X.509 CA Certificates

subject: "C=SE, ST=Blekinge, O=ET2595, CN=nich20_ca"
issuer: "C=SE, ST=Blekinge, O=ET2595, CN=nich20_root"
validity: not before Dec 15 19:28:49 2023, ok
not after Dec 12 19:28:49 2033, ok (expires in 3647 days)
serial: 10:00
flags: CA CRLSign
pathlen: 0
authKeyId: 45:64:6a:f6:d8:0c:44:a2:8c:a3:76:36:79:9b:e0:83:a2:5c:35:fd
subjKeyId: 6e:0e:5b:c0:b0:ee:7c:21:73:19:6c:c1:2f:e4:42:d9:b6:b2:21:62
pubkey: RSA 4096 bits
keyId: 15:eb:c3:4:ccc8:80:3f:1:ee:9d:d3:d9:fc:ee:60:7f:f9:a1:6d
subjKey: 6e:0e:5b:c0:b0:ee:7c:21:73:19:6c:c1:2f:e4:42:d9:b6:b2:21:62

subject: "C=SE, ST=Blekinge, O=ET2595, CN=nich20_root"
issuer: "C=SE, ST=Blekinge, O=ET2595, CN=nich20_root"
validity: not before Dec 15 19:26:33 2023, ok
not after Dec 10 19:26:33 2043, ok (expires in 7297 days)
serial: 79:8a:24:30:a4:31:a5:4d:bb:2e:d6:a2:64:23:50:5c:4e:b4:c8:85
flags: CA CRLSign self-signed
authKeyId: 45:64:6a:f6:d8:0c:44:a2:8c:a3:76:36:79:9b:e0:83:a2:5c:35:fd
subjKeyId: 45:64:6a:f6:d8:0c:44:a2:8c:a3:76:36:79:9b:e0:83:a2:5c:35:fd
pubkey: RSA 4096 bits
keyId: 7a:88:7c:07:4e:25:76:4b:2c:99:8f:e0:c7:51:f5:c4:7f:af:c3
subjKey: 45:64:6a:f6:d8:0c:44:a2:8c:a3:76:36:79:9b:e0:83:a2:5c:35:fd

student@serverA: ~$ sudo ipsec rereadcacerts
student@serverA: ~$ sudo ipsec listcacerts

List of X.509 CA Certificates

subject: "C=SE, ST=Blekinge, O=ET2595, CN=nich20_ca"
issuer: "C=SE, ST=Blekinge, O=ET2595, CN=nich20_root"
validity: not before Dec 15 19:28:49 2023, ok
not after Dec 12 19:28:49 2033, ok (expires in 3647 days)
serial: 10:00
flags: CA CRLSign
pathlen: 0
authKeyId: 45:64:6a:f6:d8:0c:44:a2:8c:a3:76:36:79:9b:e0:83:a2:5c:35:fd
subjKeyId: 6e:0e:5b:c0:b0:ee:7c:21:73:19:6c:c1:2f:e4:42:d9:b6:b2:21:62
pubkey: RSA 4096 bits
keyId: 15:eb:c3:4:ccc8:80:3f:1:ee:9d:d3:d9:fc:ee:60:7f:f9:a1:6d
subjKey: 6e:0e:5b:c0:b0:ee:7c:21:73:19:6c:c1:2f:e4:42:d9:b6:b2:21:62

subject: "C=SE, ST=Blekinge, O=ET2595, CN=nich20_root"
issuer: "C=SE, ST=Blekinge, O=ET2595, CN=nich20_root"
validity: not before Dec 15 19:26:33 2023, ok
not after Dec 10 19:26:33 2043, ok (expires in 7297 days)
serial: 79:8a:24:30:a4:31:a5:4d:bb:2e:d6:a2:64:23:50:5c:4e:b4:c8:85
flags: CA CRLSign self-signed
authKeyId: 45:64:6a:f6:d8:0c:44:a2:8c:a3:76:36:79:9b:e0:83:a2:5c:35:fd
subjKeyId: 45:64:6a:f6:d8:0c:44:a2:8c:a3:76:36:79:9b:e0:83:a2:5c:35:fd
pubkey: RSA 4096 bits
keyId: 7a:88:7c:07:4e:25:76:4b:2c:99:8f:e0:c7:51:f5:c4:7f:af:c3
subjKey: 45:64:6a:f6:d8:0c:44:a2:8c:a3:76:36:79:9b:e0:83:a2:5c:35:fd

```

Below, the customization and configuration of the ipsec.secrets and ipsec.conf files are demonstrated, showcasing the adjustments made to suit specific requirements:

The image contains two side-by-side screenshots of Oracle VM VirtualBox. Both screenshots show a terminal window titled 'Terminal' with the command 'student@server[AB]: ~/nich20\_ca'. The left screenshot (ServerB) shows the contents of /etc/ipsec.conf, which includes sections for basic configuration, sample VPN connections, and a server connection (conn server-A). The right screenshot (ServerA) shows the same file with minor differences, such as a different IP address for the right subnet. Below the terminals are standard Linux desktop toolbars.

The image contains two side-by-side screenshots of Oracle VM VirtualBox. Both screenshots show a terminal window titled 'Terminal' with the command 'student@server[AB]: ~/nich20\_ca'. The left screenshot (ServerB) shows the contents of /etc/ipsec.secrets, which contains the RSA private key for the host. The right screenshot (ServerA) shows the same file with a different RSA private key. Below the terminals are standard Linux desktop toolbars.

The connectivity between Server A and Server B was established by executing ping commands on both servers. Server A initiated the connection by pinging the IP address 192.168.70.6, while on Server B, the connection was initiated by pinging the IP address 192.168.70.5. These commands were instrumental in verifying the connectivity and successful establishment of communication between the two servers. This is where we witness the successful pinging formed between Server A and Server B.

The image contains two side-by-side screenshots of Oracle VM VirtualBox. Both screenshots show a terminal window titled 'Terminal' with the command 'student@server[AB]: ~/nich20\_ca'. The left screenshot (ServerB) shows the command 'sudo ipsec restart' being run, followed by a ping command to 192.168.70.6. The right screenshot (ServerA) shows the command 'ping 192.168.70.6' being run. Both terminals show the output of the ping command, indicating successful connectivity. Below the terminals are standard Linux desktop toolbars.

```

student@serverB:~/nich20_ca$ sudo ipsec statusall
Status of IKE charon daemon (strongSwan 5.9.5, Linux 5.15.0-56-generic, x86_64):
    uptime: 19 minutes, since Dec 17 21:09:39 2023
    malloc: sbrk 2355200, mmap 0, used 1333536, free 1021664
    worker threads: 11 of 16 idle, 5/0/0/0 working, job queue: 0/0/0/0, scheduled: 2
    loaded plugins: charon aesnt aes rc2 sha2 sha1 md5 mgf1 random nonce x509 revocation constrain
    ts pubkey pkcs1 pkcs7 pkcs8 pkcs12 ppp dnskey sshkey pem openssl fips-prf gmp agent xcbc hmac gc
    m dbg attr kernel-netlink resolve socket-default connmark stroke updown esp-mschapv2 xauth-gene
    ric counters
    Listening IP addresses:
        192.168.80.100
        192.168.70.6
        10.0.4.15
    Connections:
        serverB-A: 192.168.70.6...192.168.70.5 IKEV2
        serverB-A: local: [C=SE, ST=Blekinge, O=ET2595, CN=192.168.70.6] uses public key authentic
        ation
        serverB-A: cert: "C=SE, ST=Blekinge, O=ET2595, CN=192.168.70.6"
        serverB-A: remote: [C=SE, ST=Blekinge, O=ET2595, CN=192.168.70.5] uses public key authentic
        ation
        serverB-A: child: dynamic === dynamic TRANSPORT
    Routed Connections:
        serverB-A(1): ROUTED, TRANSPORT, reqid 1
        serverB-A(1): 192.168.70.6/32 === 192.168.70.5/32
    Security Associations (1 up, 0 connecting)
    serverB-A[1]: ESTABLISHED 19 minutes ago, 192.168.70.6[C=SE, ST=Blekinge, O=ET2595, CN=192.16
    8.70.6]...192.168.70.5[C=SE, ST=Blekinge, O=ET2595, CN=192.168.70.5]
    servers-B[1]: IKEV2 SPIs: 570186715f73a525_1* 62022b5543f3972c_r*, public key reauthentication
    in 2 hours
    serverB-A[1]: IKE proposal: AES_CBC_128/HMAC_SHA2_256_128/PRF AES128 XCB/CURVE_25519
    serverB-A(2): INSTALLED, TRANSPORT, reqid 1, ESP SPIs: cbf2b21b_o
    serverB-A(2): AES_CBC_128/HMAC_SHA2_256_128, 2368 bytes_l (37 pkts, 1153s ago), 2368 bytes_o
    (37 pkts, 1153s ago), rekeying in 26 minutes
    serverB-A(2): 192.168.70.6/32 === 192.168.70.5/32

student@serverA:~/nich20_ca$ sudo ipsec statusall
Status of IKE charon daemon (strongSwan 5.9.5, Linux 5.15.0-56-generic, x86_64):
    uptime: 19 minutes, since Dec 17 21:09:45 2023
    malloc: sbrk 3031040, mmap 0, used 1370912, free 1660128
    worker threads: 11 of 16 idle, 5/0/0/0 working, job queue: 0/0/0/0, scheduled: 2
    loaded plugins: charon aesnt aes rc2 sha2 sha1 md5 mgf1 random nonce x509 revocation constrain
    ts pubkey pkcs1 pkcs7 pkcs8 pkcs12 ppp dnskey sshkey pem openssl fips-prf gmp agent xcbc hmac gc
    m dbg attr kernel-netlink resolve socket-default connmark stroke updown esp-mschapv2 xauth-gene
    ric counters
    Listening IP addresses:
        192.168.80.100
        192.168.70.5
        10.0.4.15
    Connections:
        serverA-B: 192.168.70.5...192.168.70.6 IKEV2
        serverA-B: local: [C=SE, ST=Blekinge, O=ET2595, CN=192.168.70.5] uses public key authentic
        ation
        serverA-B: cert: "C=SE, ST=Blekinge, O=ET2595, CN=192.168.70.5"
        serverA-B: remote: [C=SE, ST=Blekinge, O=ET2595, CN=192.168.70.6] uses public key authentic
        ation
        serverA-B: child: dynamic === dynamic TRANSPORT
    Routed Connections:
        serverA-B(1): ROUTED, TRANSPORT, reqid 1
        serverA-B(1): 192.168.70.5/32 === 192.168.70.6/32
    Security Associations (1 up, 0 connecting)
    serverA-B[1]: ESTABLISHED 19 minutes ago, 192.168.70.5[C=SE, ST=Blekinge, O=ET2595, CN=192.16
    8.70.5]...192.168.70.6[C=SE, ST=Blekinge, O=ET2595, CN=192.168.70.6]
    servers-B[1]: IKEV2 SPIs: 570186715f73a525_1* 62022b5543f3972c_r*, public key reauthentication
    in 2 hours
    serverA-B[1]: IKE proposal: AES_CBC_128/HMAC_SHA2_256_128/PRF AES128 XCB/CURVE_25519
    serverA-B(2): INSTALLED, TRANSPORT, reqid 1, ESP SPIs: cbf2b21b_l ce45466a_o
    serverA-B(2): AES_CBC_128/HMAC_SHA2_256_128, 2368 bytes_l (37 pkts, 1156s ago), 2368 bytes_o
    (37 pkts, 1156s ago), rekeying in 24 minutes
    serverA-B(2): 192.168.70.5/32 === 192.168.70.6/32

```

The presence of encryption is clear in ensuring the security of the traffic at this location. Measures have been taken to safeguard the transmitted data through encryption.

```

student@serverA:~/nich20_ca$ sudo ip xfrm state
src 192.168.70.5 dst 192.168.70.6
proto esp spi 0xce45466a reqid 1 mode transport
replay-window 0
auth-trunc hmac(sha256) 0xe1ba97d074774b3d08f175745ce456c103863e907875ac
152921e9d1060b2465 128
enc cbc(aes) 0xe2c2c497d73abfc6af44d3e0686dd06c
anti-replay context: seq 0x0, oseq 0x25, 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 0xcbf2b21b reqid 1 mode transport
replay-window 32
auth-trunc hmac(sha256) 0xcfcade67fb0fd728966dd1b5b9fa9db4ff3ae0eeb83b37
enc cbc(aes) 0xac149769b8fcdb83f3f3e0e74832caa3
anti-replay context: seq 0x25, oseq 0x0, bitmap 0xffffffff
sel src 192.168.70.6/32 dst 192.168.70.5/32
student@serverA:~/nich20_ca$ 

```

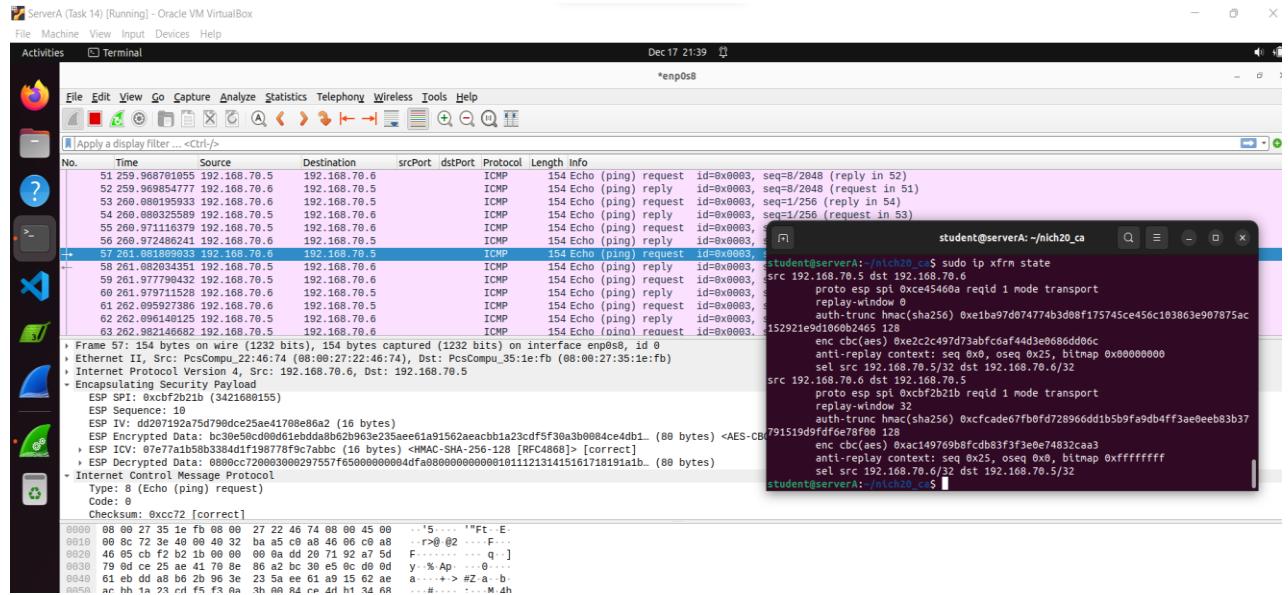
```

student@serverA:~/nich20_ca$ sudo ip xfrm state
src 192.168.70.5 dst 192.168.70.6
proto esp spi 0xce45466a reqid 1 mode transport
replay-window 0
auth-trunc hmac(sha256) 0xe1ba97d074774b3d08f175745ce456c103863e907875ac
152921e9d1060b2465 128
enc cbc(aes) 0xe2c2c497d73abfc6af44d3e0686dd06c
anti-replay context: seq 0x0, oseq 0x25, 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 0xcbf2b21b reqid 1 mode transport
replay-window 32
auth-trunc hmac(sha256) 0xcfcade67fb0fd728966dd1b5b9fa9db4ff3ae0eeb83b37
enc cbc(aes) 0xac149769b8fcdb83f3f3e0e74832caa3
anti-replay context: seq 0x25, oseq 0x0, bitmap 0xffffffff
sel src 192.168.70.6/32 dst 192.168.70.5/32
student@serverA:~/nich20_ca$ 

student@serverA:~/nich20_ca$ sudo ipsec statusall
Status of IKE charon daemon (strongSwan 5.9.5, Linux 5.15.0-56-generic, x86_64):
    uptime: 19 minutes, since Dec 17 21:09:45 2023
    malloc: sbrk 3031040, mmap 0, used 1370912, free 1660128
    worker threads: 11 of 16 idle, 5/0/0/0 working, job queue: 0/0/0/0, scheduled: 2
    loaded plugins: charon aesnt aes rc2 sha2 sha1 md5 mgf1 random nonce x509 revocation constrain
    ts pubkey pkcs1 pkcs7 pkcs8 pkcs12 ppp dnskey sshkey pem openssl fips-prf gmp agent xcbc hmac gc
    m dbg attr kernel-netlink resolve socket-default connmark stroke updown esp-mschapv2 xauth-gene
    ric counters
    Listening IP addresses:
        192.168.80.100
        192.168.70.5
        10.0.4.15
    Connections:
        serverA-B: 192.168.70.5...192.168.70.6 IKEV2
        serverA-B: local: [C=SE, ST=Blekinge, O=ET2595, CN=192.168.70.5] uses public key authentic
        ation
        serverA-B: cert: "C=SE, ST=Blekinge, O=ET2595, CN=192.168.70.5"
        serverA-B: remote: [C=SE, ST=Blekinge, O=ET2595, CN=192.168.70.6] uses public key authentic
        ation
        serverA-B: child: dynamic === dynamic TRANSPORT
    Routed Connections:
        serverA-B(1): ROUTED, TRANSPORT, reqid 1
        serverA-B(1): 192.168.70.5/32 === 192.168.70.6/32
    Security Associations (1 up, 0 connecting)
    serverA-B[1]: ESTABLISHED 19 minutes ago, 192.168.70.5[C=SE, ST=Blekinge, O=ET2595, CN=192.16
    8.70.5]...192.168.70.6[C=SE, ST=Blekinge, O=ET2595, CN=192.168.70.6]
    servers-B[1]: IKEV2 SPIs: 570186715f73a525_1* 62022b5543f3972c_r*, public key reauthentication
    in 2 hours
    serverA-B[1]: IKE proposal: AES_CBC_128/HMAC_SHA2_256_128/PRF AES128 XCB/CURVE_25519
    serverA-B(2): INSTALLED, TRANSPORT, reqid 1, ESP SPIs: cbf2b21b_l ce45466a_o
    serverA-B(2): AES_CBC_128/HMAC_SHA2_256_128, 2368 bytes_l (37 pkts, 1156s ago), 2368 bytes_o
    (37 pkts, 1156s ago), rekeying in 24 minutes
    serverA-B(2): 192.168.70.5/32 === 192.168.70.6/32

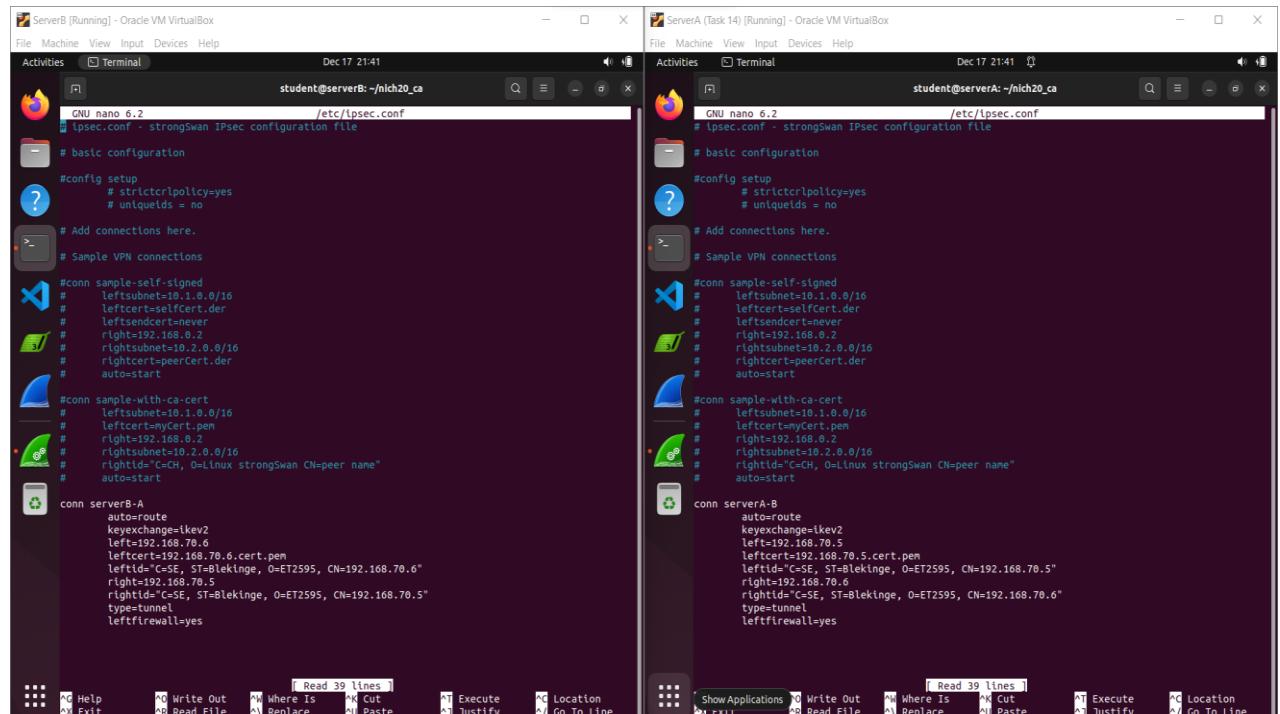
```

The method used to decrypt the traffic appears to align with the process executed in Task 16. It seems the decryption of the traffic follows a similar approach as the one employed in our previous Task 16.



## Performing of Task 19:

Tunnel mode maintains consistency with the majority of the settings established in task 18. With a notable alteration being made to the "type" parameter within the ipsec.conf files. Here for both Server A and Server B, the setting of the "type" to "tunnel" is pivotal as it enables the IPsec connection to function in a tunneling configuration, primarily utilized to secure communication between distinct networks or subnetworks.



Sure, the communication link between Server A and Server B was established by using the 'ping' command. On Server A, the connection was started by pinging the IP address 192.168.70.6, while on Server B, the initiation of the connection was made by pinging the IP address 192.168.70.5. This process allowed both servers to check for connectivity and response time between them by sending test packets back and forth.

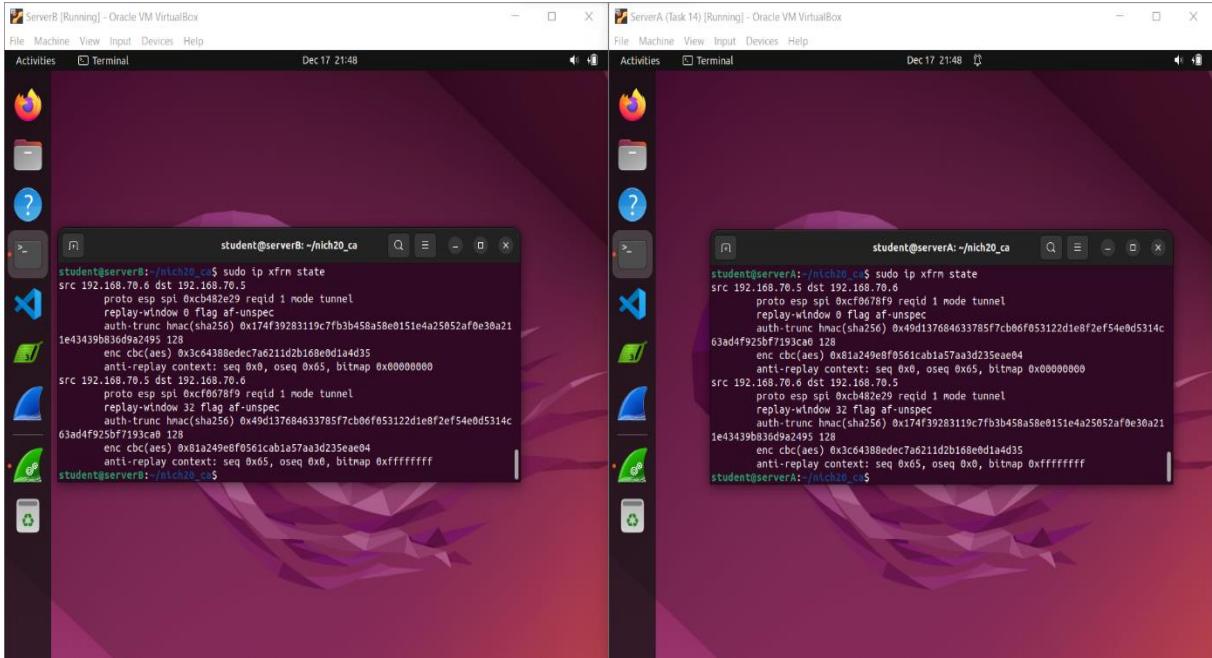
The screenshot shows two Oracle VM VirtualBox windows. The left window is titled "ServerB [Running] - Oracle VM VirtualBox" and the right window is titled "ServerA (Task 14) [Running] - Oracle VM VirtualBox". Both windows have "Activities" and "Terminal" tabs open. In the Terminal tab of ServerB, a user named "student" runs the command "ping 192.168.70.5". In the Terminal tab of ServerA, another user named "student" runs the command "ping 192.168.70.6". Both terminals show the ping results with various ICMP sequence numbers and timestamps.

This screenshot shows the same two Oracle VM VirtualBox windows. In the Terminal tab of ServerB, the user runs "sudo ipsec statusall". In the Terminal tab of ServerA, the user runs "student@serverA:~/nich20\_ca\$ sudo ipsec statusall". Both terminals display detailed status information about IKE daemon processes, security associations, and tunnel configurations.

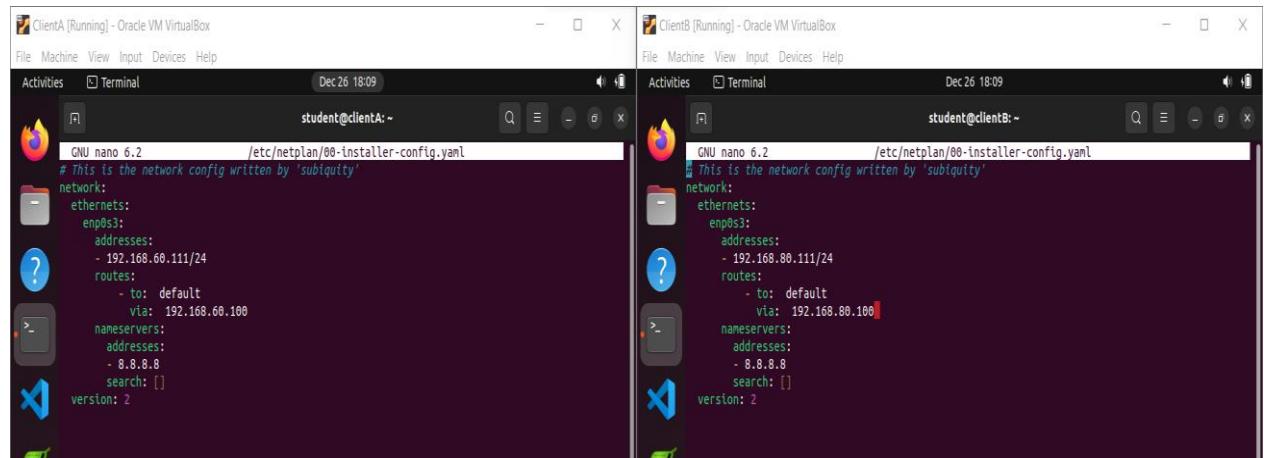
  

This screenshot shows the same two Oracle VM VirtualBox windows. Both have Wireshark running in the foreground. The left window (ServerB) has a capture titled "Capturing from enp0s8" and the right window (ServerA) has a capture titled "Capturing from enp0s8". Both Wireshark interfaces show a list of captured network frames, primarily showing ICMP and ESP protocol traffic between the two hosts.



## Performing of Task 20:

Certainly! A ping test was carried out to confirm the tunneling link between Client A and Client B. This connection was set up by configuring specific gateways for each client. In the "/etc/netplan/00-installer-config.yaml" file, Client A's gateway was set as 192.168.60.100, while Client B's gateway was configured as 192.168.80.100. Through this configuration, both clients were able to communicate with each other using the designated gateways, ensuring a functional and connected tunnel between them.



In the ipsec.conf files of Server A and Server B, specific subnet values were defined for the IPsec connection. Server A's configuration sets the left-subnet value as 192.168.60.0 and the right-subnet value as 192.168.80.0. Conversely, Server B's configuration reverses these values, setting the left subnet to 192.168.80.0 and the right subnet to 192.168.60.0. This setup ensures that each server accurately recognizes and directs traffic to the corresponding subnets at the opposite end of the established tunnel. Meanwhile, the ipsec.secrets file retains its previous settings, maintaining the necessary security credentials for the IPsec connection without any alterations.

```

ServerA (task-19-nikhil) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal Dec 26 18:51
student@serverA: ~
GNU nano 6.2 /etc/ipseconfig
# ipsec.conf - strongSwan IPsec configuration file

# basic configuration

#config setup
#       # strictcrlpolicy=yes
#       # uniqueids = no

# Add connections here.

# Sample VPN connections

#conn sample-self-signed
#       leftsubnet=192.168.0.0/16
#       leftcert=selfCert.der
#       leftsencert=never
#       right=192.168.0.2
#       rightsubnet=10.2.0.0/16
#       rightcert=peerCert.der
#       autostart

#conn sample-with-ca-cert
#       leftsubnet=192.168.0.0/16
#       leftcert=caCert.pem
#       rightsubnet=10.2.0.0/16
#       rightid="C=CH, O=Linux strongSwan CN=peer name"
#       rightcert=peerCert.der
#       autostart

conn serverA-B
        auto-route
        keyexchange=lkev2
        left=192.168.70.5
        leftcert=caCert.pem
        rightsubnet=192.168.68.0/24
        rightcert=caSE, ST=Blekinge, O=ET2595, CN=192.168.70.5"
        right=192.168.70.6
        rightsubnet=192.168.68.0/24
        rightid="caSE, ST=Blekinge, O=ET2595, CN=192.168.70.6"
        type=tunnel
        leftfirewall=yes

ServerB (Task 19-nikhil) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal Dec 26 18:51
student@serverB: ~
GNU nano 6.2 /etc/ipseconfig
# ipsec.conf - strongSwan IPsec configuration file

# basic configuration

#config setup
#       # strictcrlpolicy=yes
#       # uniqueids = no

# Add connections here.

# Sample VPN connections

#conn sample-self-signed
#       leftsubnet=10.1.0.0/16
#       leftcert=selfCert.der
#       leftsencert=never
#       right=192.168.0.2
#       rightsubnet=10.2.0.0/16
#       rightcert=peerCert.der
#       autostart

#conn sample-with-ca-cert
#       leftsubnet=10.1.0.0/16
#       leftcert=caCert.pem
#       rightsubnet=10.2.0.0/16
#       rightid="C=CH, O=Linux strongSwan CN=peer name"
#       rightcert=peerCert.der
#       autostart

conn serverB-A
        auto-route
        keyexchange=lkev2
        left=192.168.70.6
        leftcert=caCert.pem
        rightsubnet=192.168.68.0/24
        rightcert=caSE, ST=Blekinge, O=ET2595, CN=192.168.70.6"
        right=192.168.70.5
        rightsubnet=192.168.68.0/24
        rightid="caSE, ST=Blekinge, O=ET2595, CN=192.168.70.5"
        type=tunnel
        leftfirewall=yes

```

to enable IP forwarding on both servers, two commands were utilized. The first command, "sudo sysctl -w net.ipv4.ip\_forward=1", modifies a system parameter to allow the forwarding of IP packets between network interfaces. Following that, the second command, "sudo sysctl -p", is executed to apply and activate these system changes effectively. These commands work together to enable dynamic adjustments of system settings, ensuring improved network routing capabilities on the servers.

```

ClientA [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal Dec 26 18:54
student@clientA: ~
student@clientA:~$ ping 192.168.80.111
PING 192.168.80.111 (192.168.80.111) 56(84) bytes of data.
64 bytes from 192.168.80.111: icmp_seq=1 ttl=62 time=3.18 ms
64 bytes from 192.168.80.111: icmp_seq=2 ttl=62 time=4.08 ms
64 bytes from 192.168.80.111: icmp_seq=3 ttl=62 time=3.68 ms
64 bytes from 192.168.80.111: icmp_seq=4 ttl=62 time=3.96 ms
64 bytes from 192.168.80.111: icmp_seq=5 ttl=62 time=22.6 ms

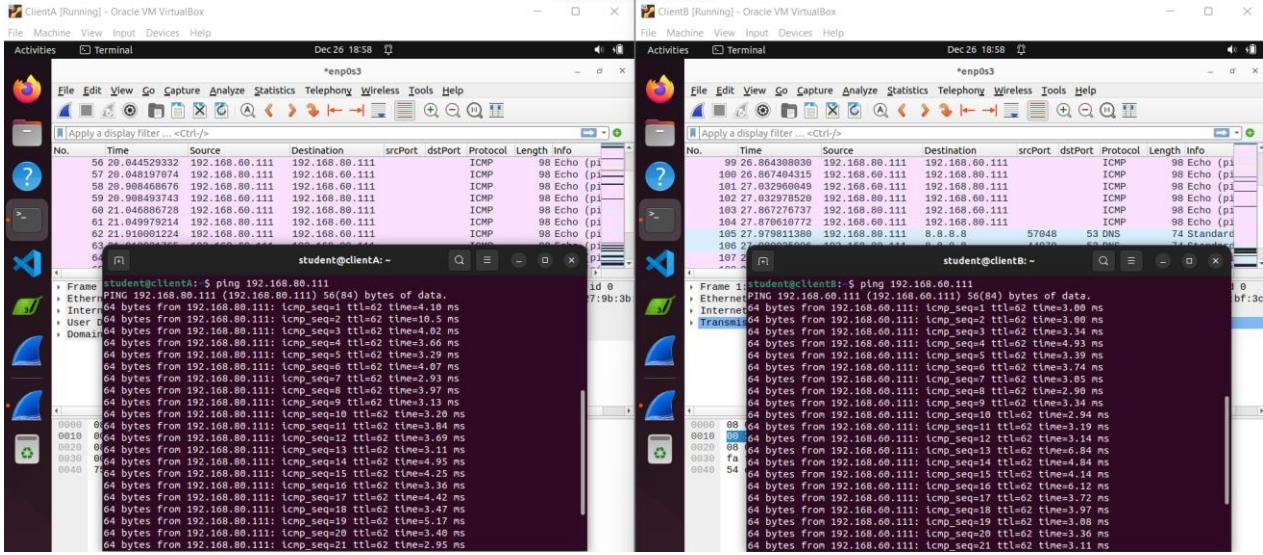
ClientB [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal Dec 26 18:54
student@clientB: ~
student@clientB:~$ ping 192.168.60.111
PING 192.168.60.111 (192.168.60.111) 56(84) bytes of data.
64 bytes from 192.168.60.111: icmp_seq=1 ttl=62 time=3.19 ms
64 bytes from 192.168.60.111: icmp_seq=2 ttl=62 time=4.99 ms
64 bytes from 192.168.60.111: icmp_seq=3 ttl=62 time=4.05 ms
64 bytes from 192.168.60.111: icmp_seq=4 ttl=62 time=3.37 ms
64 bytes from 192.168.60.111: icmp_seq=5 ttl=62 time=3.25 ms
64 bytes from 192.168.60.111: icmp_seq=6 ttl=62 time=6.45 ms
64 bytes from 192.168.60.111: icmp_seq=7 ttl=62 time=3.84 ms

Wireshark [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal Dec 26 18:54
student@clientB: ~

ServerA (task-19-nikhil) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal Dec 26 18:54
student@serverA: ~
student@serverA:~$ sudo nano /etc/ipseconfig
student@serverA:~$ sudo ipsec restart
Stopping strongSwan IPsec...
[[ASTarting strongSwan 5.9.5 IPsec [starter]...
student@serverA:~$ sudo sysctl -w net.ipv4.ip_forward=1 & sudo sysctl -p
[1] 3746
net.ipv4.ip_forward = 1
[1]+ Done                  sudo sysctl -w net.ipv4.ip_forward=1
student@serverA:~$ 

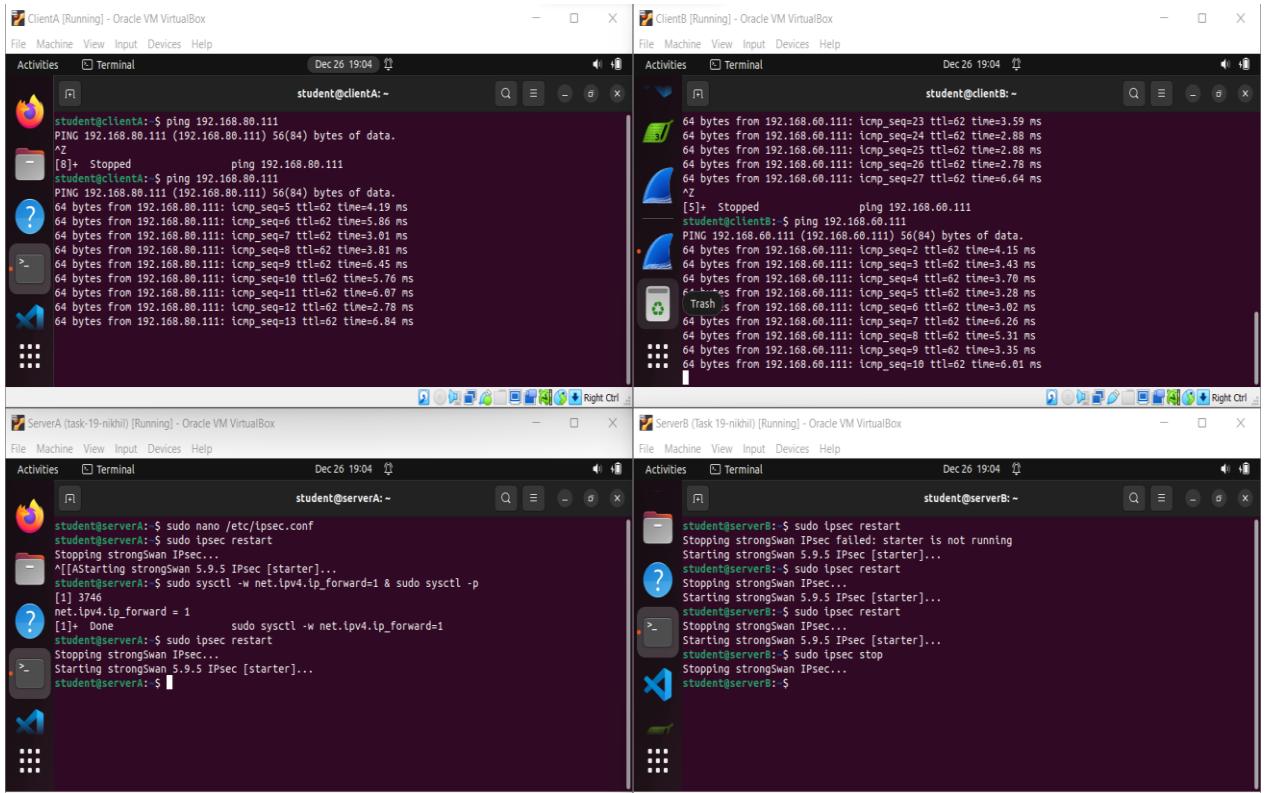
ServerB (Task 19-nikhil) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal Dec 26 18:54
student@serverB: ~
student@serverB:~$ sudo nano /etc/ipseconfig
student@serverB:~$ sudo ipsec restart
Stopping strongSwan IPsec...
Starting strongSwan 5.9.5 IPsec [starter]...
student@serverB:~$ sudo sysctl -w net.ipv4.ip_forward=1 & sudo sysctl -p
[1] 4028
net.ipv4.ip_forward = 1
[1]+ Done                  sudo sysctl -w net.ipv4.ip_forward=1
student@serverB:~$ 

```

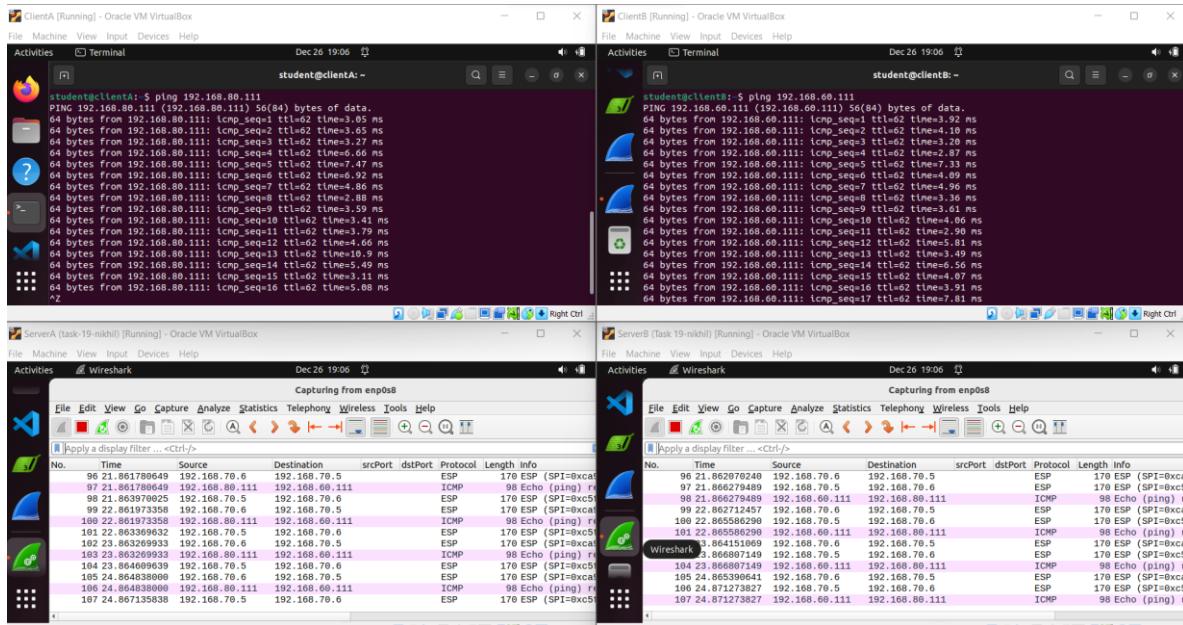


To make sure that the established tunnel was being used for communication, initially trying to spot ESP (Encapsulating Security Payload) packets didn't give us a clear answer and was quite tricky, especially when dealing with complex technical documentation. However, seeking help from the Stack Overflow led us to a simpler and more reliable method.

While we had IP forwarding facilitated, we operated a test. Whenever we deliberately stopped the IPsec service on either of the servers, the ongoing ping communication also stopped. This clear observation, shown in the provided figure, strongly suggested that the communication between Server A and Server B was actually relying on the set-up tunnel. This test effectively showed us that the established tunnel was indeed crucial for the communication between the two servers.

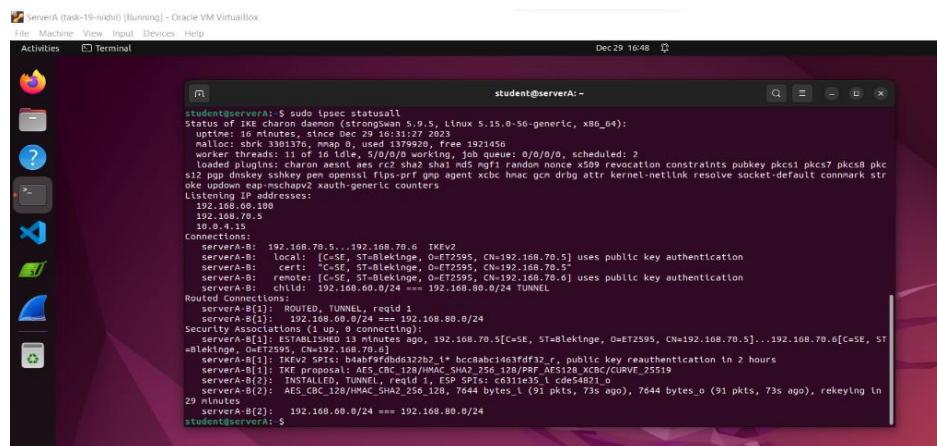


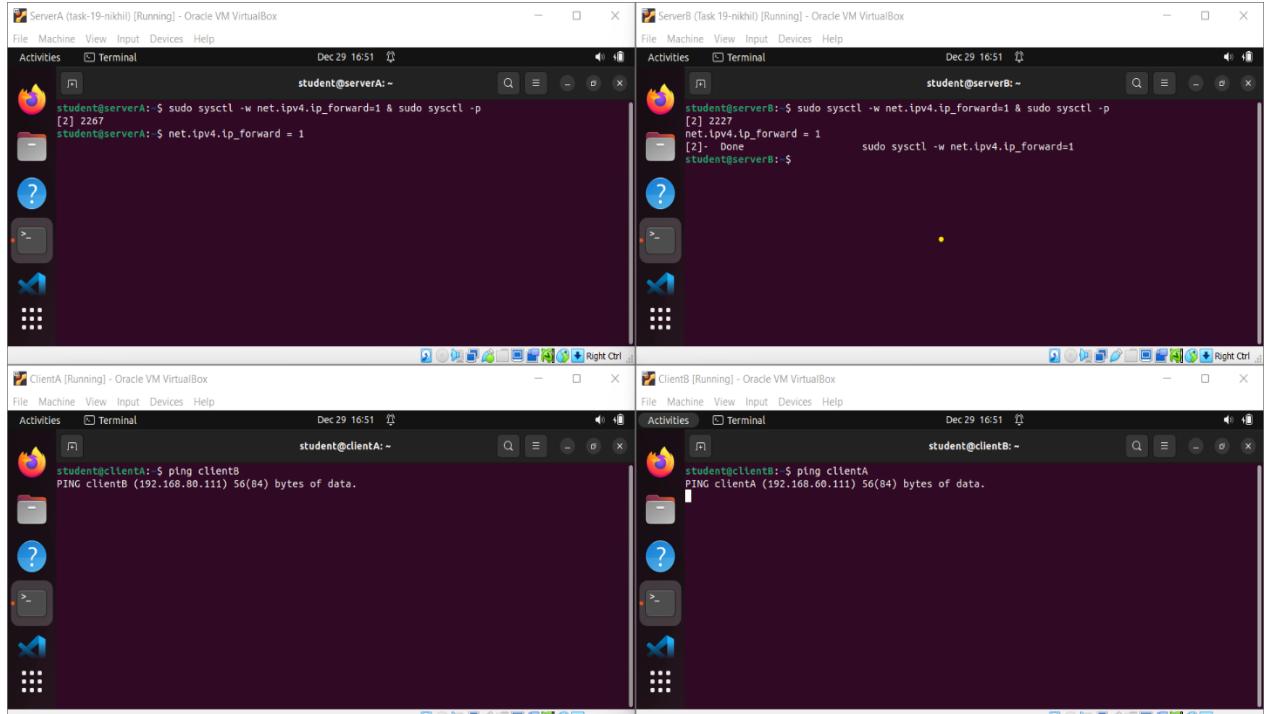
The ESP (Encapsulating Security Payload) traffic observed originating from both servers is a strong indicator of the active involvement of the IPsec tunnel in securing the communication flow between them. This ESP traffic presence signifies that data being transmitted between the servers is undergoing encryption within the established tunnel. This encryption ensures that the exchanged information remains confidential and maintains its integrity throughout the communication process.



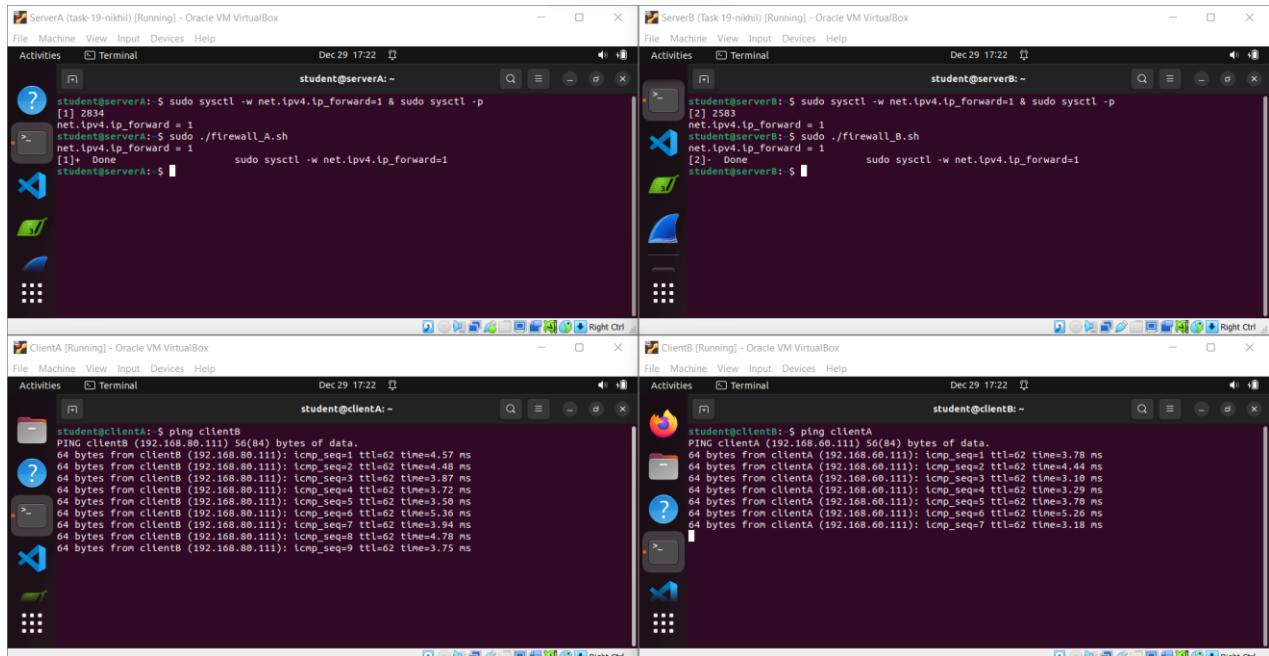
## **Performing of Task21:**

In this particular task, while keeping the configurations in 'ipsec.conf' and 'ipsec.secrets' unchanged, the focus was on adjusting the firewall rules on both servers. These modifications aimed to enable site-to-site communication and ensure that both clients could access the internet through their respective servers. These adjustments were intended to facilitate communication between the sites in a secure manner while allowing Client A and Client B to access the internet via their designated servers. Each screenshot showcased the required alterations made to the firewall settings, ensuring seamless communication and internet access for both clients through their respective server setups.

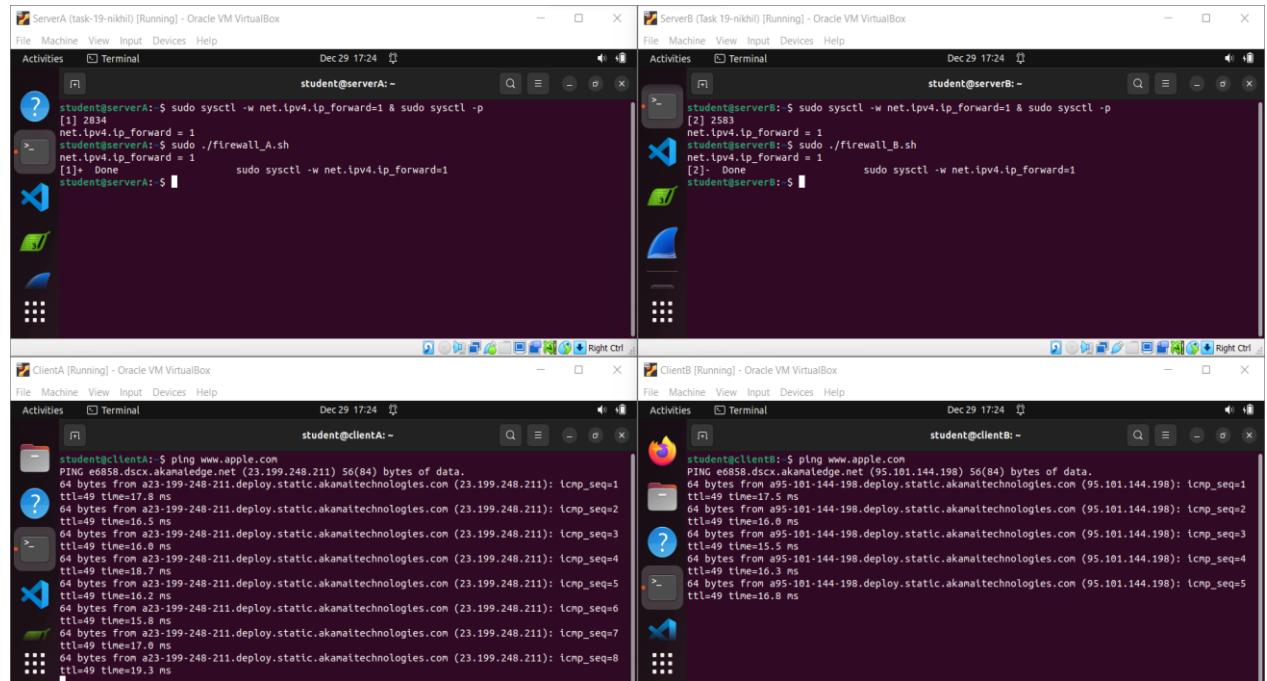




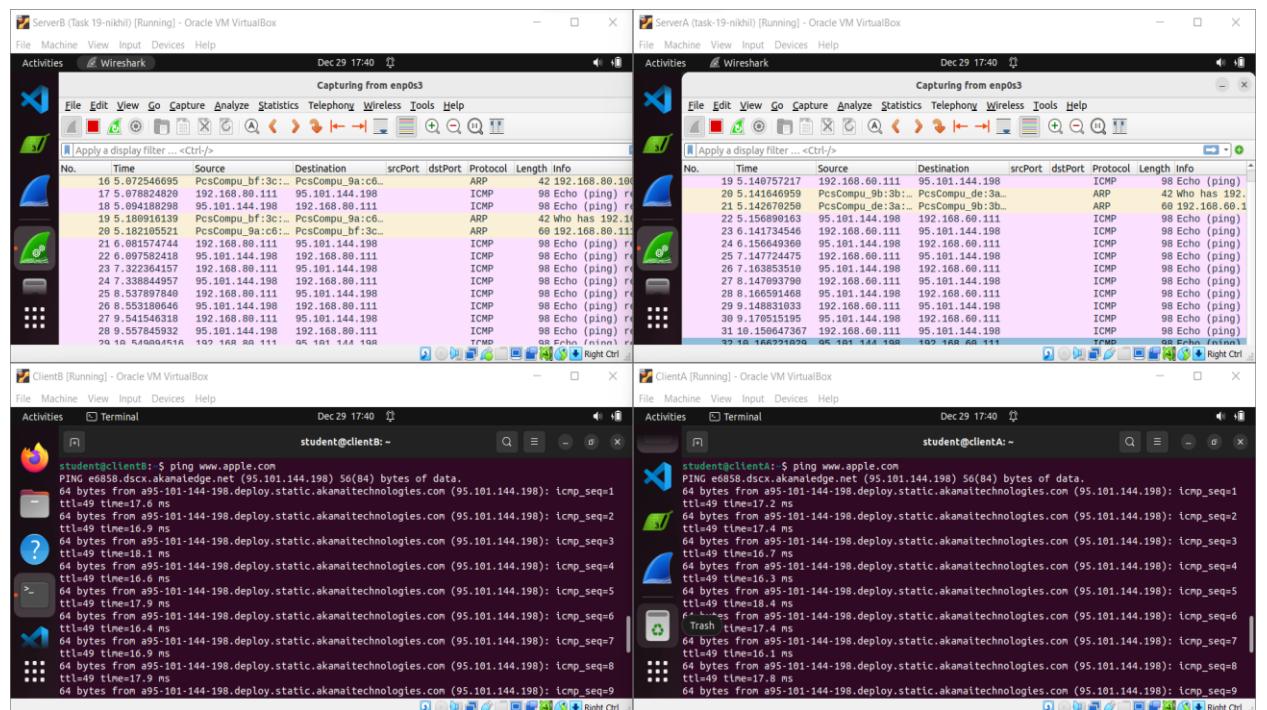
At this point, despite setting up appropriate rules in IPsec and enabling IPv4 forwarding on the servers, the clients were unable to ping each other. To resolve this, specific firewall rules were crafted and implemented on both Server A and Server B. Once these firewall rules were applied, an immediate change was observed: the clients regained the ability to ping each other successfully. These adjustments in the firewall settings directly facilitated the establishment of communication between the clients, resolving the issue of ping connectivity that existed previously.



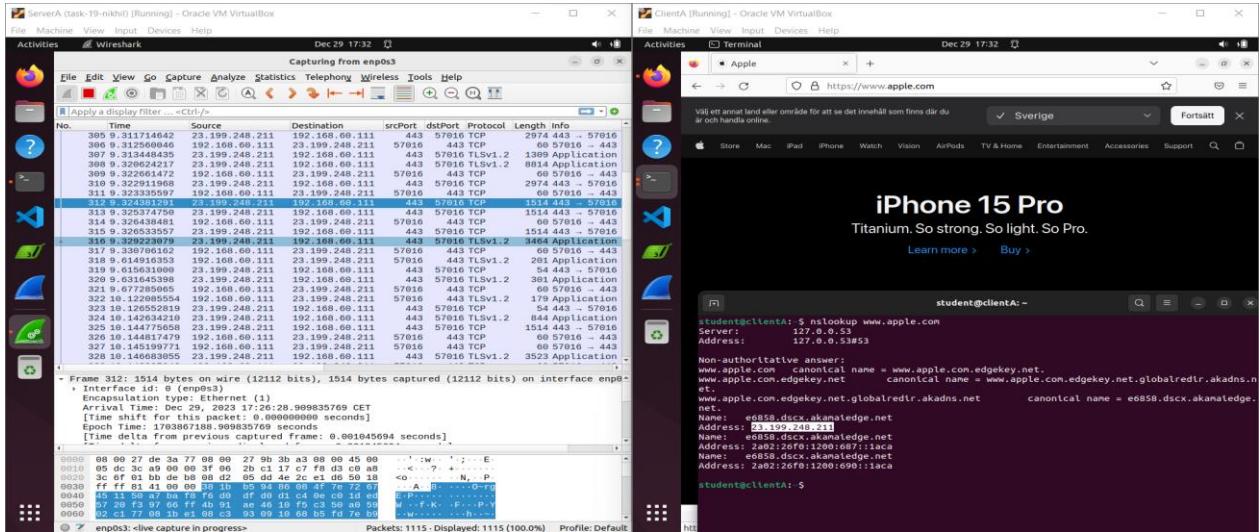
After setting up firewall rules on both Server A and Server B, the clients regained their capability to ping each other effectively. This positive outcome clearly indicates that modifying and implementing the firewall rules played a crucial role in resolving the communication problems that existed earlier. These adjustments directly facilitated the restoration of connectivity between the clients, addressing the previous issues they encountered while attempting to communicate.



Demonstrate the network connection between Client A and Client B through Server B to Server A using Wireshark captures on the respective machines.

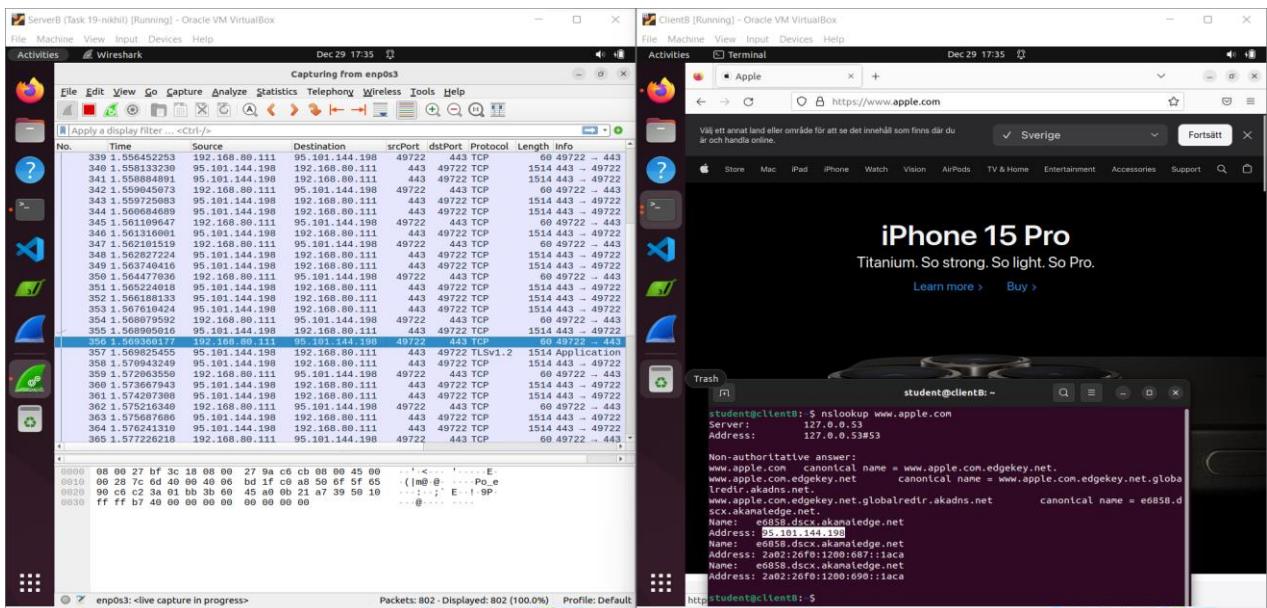


After confirming the ping connectivity, it was ensured that both Client A and Client B had internet access through their respective servers. This was validated by successfully accessing facebook.com from both clients, affirming their ability to utilize their servers for internet connectivity. This verification was supported by visual evidence.

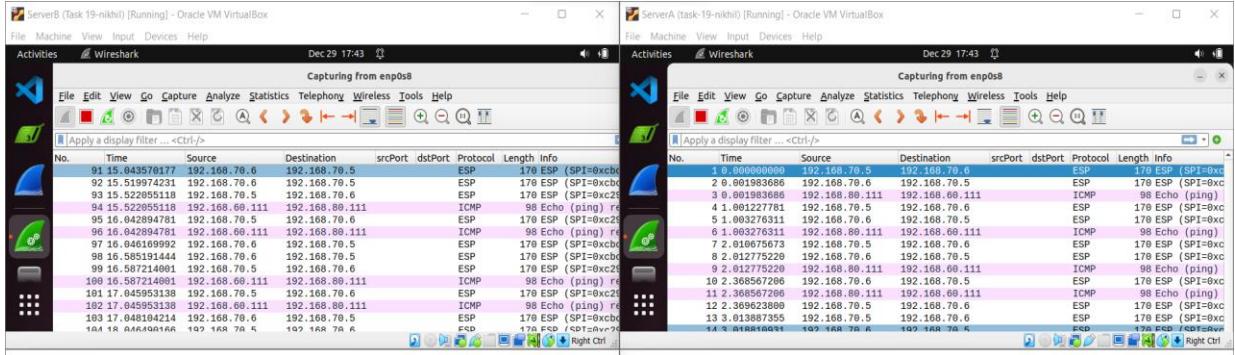


To ensure that both clients could access the internet via their assigned servers, we utilized Apple's IP address, 23.199.248.211, to confirm connectivity. The validation was confirmed by examining Wireshark session logs, which indicated successful internet connections for both clients routed through their respective servers.

We compared the recorded web page reload on Server A when Client A accessed the internet with the *nslookup* displayed in Client A, revealing the identical IP address, 23.199.248.211. Similarly, we conducted the same comparison for Client B's internet access, analyzing the data captured on Server B. This comparison highlighted the consistency between the web page reloads and the corresponding ping requests, reaffirming that both clients were effectively accessing the internet through their designated servers.



Certainly! Encapsulating Security Payload (ESP) traffic was captured separately on both Server A and Server B. This examination involved observing and analyzing the ESP traffic present on each server's network interface. The goal was to comprehensively study the encrypted data flow within the communication channels of both servers.



On Server B, a detailed capture of ESP (Encapsulating Security Payload) traffic was conducted. This detailed analysis aimed to examine the ESP traffic more closely, allowing for a comprehensive understanding of the encrypted data flow within the network. This deeper examination on Server B provided a more in-depth insight into how the Encapsulating Security Payload operates within the communication framework.

