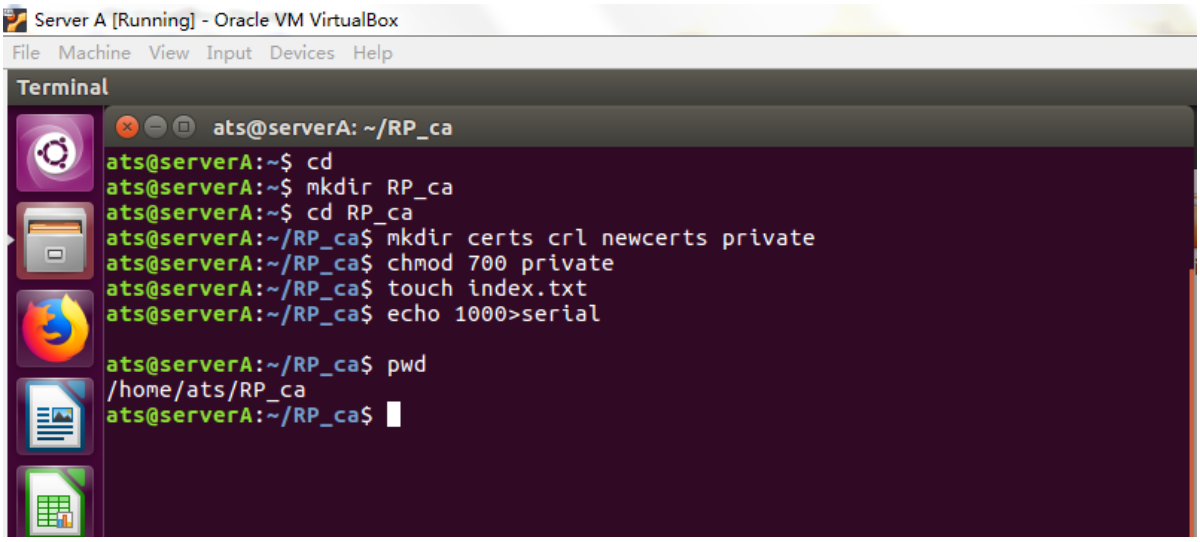
Certificates and VPN Report

PENG RONG

9511207004

921411446@qq.com



Newcerts:Certificates which issued by the root CA.
Certs: Copy of each certificates by the root CA.
Private: keeps private key.
Crl: the revoked certificates list.
Index.txt: OpenSSL use to keep track of all issued certificates. Initial value is 1000.

```
####################################################################
[ CA_default ]

dir             = /home/ats/RP_ca       # Where everything is kept
certs           = $dir/certs            # Where the issued certs are kept
crl_dir         = $dir/crl              # Where the issued crl are kept
database        = $dir/index.txt        # database index file.
#unique_subject = no                    # Set to 'no' to allow creation of
                                        # several ctificates with same subject.
new_certs_dir   = $dir/newcerts         # default place for new certs.

certificate     = $dir/certs/root.cert.pem      # The CA certificate
serial          = $dir/serial           # The current serial number
crlnumber       = $dir/crlnumber        # the current crl number
                                        # be commented out to leave a V1 CRL
crl             = $dir/crl.pem          # The current CRL
private_key     = $dir/private/root.key.pem     # The private key
RANDFILE        = $dir/private/.rand    # private random number file
```

截图(Alt + A)

```
default_crl_days= 30
default_md      = sha256

#x509_extensions        = usr_cert
```

```
[ req ]
default_bits            = 2048
default_keyfile         = privkey.pem
distinguished_name      = req_distinguished_name
attributes              = req_attributes
x509_extensions = v3_ca # The extentions to add to the self signed cert
default_md      = sha256
```

```
[ req_distinguished_name ]
countryName                     = Country Name (2 letter code)
countryName_default             = SE
countryName_min                 = 2
countryName_max                 = 2
stateOrProvinceName_default     = Blekinge
localityName_default            = Karlskrona

0.organizationName              = Organization Name (eg, company)
0.organizationName_default      = ET2540
```

```
# left out by default:
  keyUsage = critical,digitalSignature,cRLSign,keyCertSign
```

Task1:[v3_ca]

When we have the openssl.cnf, we can make certificate signing request. The assigned values are configure the OpenSSL, OpenSSL is used to sign certificate, so it needs the configuration files to coordinate.

In the [ca_default] section, certificate parameter is to point out the certificate's location, so I change to my root directory is /home/ats/RP_ca, you can see that by using pwd command. Private_key parameter is to point out the place where saving the private key. The

default_md parameter I think it is ask about which message digest to use,sha-1 (secure hash algorithm) can generate a 160-bit message digest, when receiving a message, this message digest can be used to verify the integrity of the data. But the Google and CWI Institute in Amsterdam have found it really existing hash collision, then somebody can fake, and sha-2 get its improvement, and sha-256 belongs to sha-2, so this lab we chose the sha256 as the value of default_md.

Then talk about keyUsage, this parameter is used to define the purpose of the keys, and have the following value: digitalSignature , nonRepudiation,keyEncipherment,dataEncipherment,keyAgreement,keyCertSign,cRLSign,encipherOnly,decipherOnly. The digital Signature value is used for verifying the digital signature such as integrity service. The nonRepudiation is besides the verification of digitalSignature, and it can protect against the signing entity falsely repudiating some action. The keyEncipherment bit, will be used when the subject public key or secret key is enciphering,it can act as key transport.

In [req_distinguished_name]section is recording some default information of DN, and DN is composed of attributes. Certificates user must prepare the issuer DN and subject DN.

The above setting is let the root openssl.cnf know the certificates information and can sign and issue after later.

Task2:[v3_intermediate_ca]

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

The subjectKeyIdentifier is the keyidentifier for your subject, and is a hash type. The authoritykeyidentifier contains two values ,keyid and issuer, keyid will present an attempt to copy the subject key identifier from the parent certificate ,its value is " always" so if there's any errors it will return. The issuer option only be done when the keyid fails or not included, it will copy the issuer and serial number from the issuer certificate. The basicConstraints is used for indicating whether a certificate is a CA certificate, and pathlen:0 means it can only be used to sign end users certificates and no further CAs.

Because root CA is vulnerable, so root CA would not directly sign for server end or client end, it's clever and has several intermediates to help itself sign for them. Therefore, the intermediate_ca configuration just

need some identifier and constraints. So [v3_ca] looks like complicated configuration, but [v3_intermediate_ca] is more like its shadow avatar.

Task3:[usr_cert]

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

# This is required for TSA certificates.
extendedKeyUsage = clientAuth,emailProtection
```

This section will be used for signing client certificates, such as e-mail. These extensions are added when 'ca' signs a request. As x509 extension requirement, we put such attributes is to clear user cert key is for what the purpose, and the extendedKeyUsage is required for client authentication and provide email protection.

Task4:[server_cert]

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

Also, these extensions are added when 'ca' signs a request. This goes against PKIX guidelines but some CAs do it and some software requires this to avoid interpreting an end user certificate as a CA.

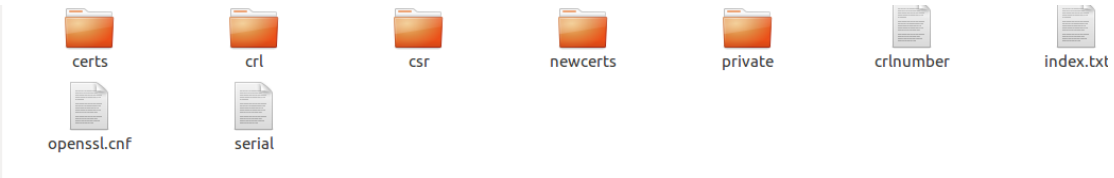
```
ats@serverA:~/RP_ca$ mkdir ca1
ats@serverA:~/RP_ca$ cd ca1
ats@serverA:~/RP_ca/ca1$ mkdir certs crl newcerts private csr
ats@serverA:~/RP_ca/ca1$ chmod 700 private
ats@serverA:~/RP_ca/ca1$ touch index.txt
ats@serverA:~/RP_ca/ca1$ echo 2000>serial
bash: 2000: Bad file descriptor
ats@serverA:~/RP_ca/ca1$ echo 2000 > serial
ats@serverA:~/RP_ca/ca1$
ats@serverA:~/RP_ca/ca1$ echo 2000 > crlnumber
```

```
[ CA_default ]

dir             = /home/ats/RP_ca/ca1   # Where everything is kept
certs           = $dir/certs            # Where the issued certs are kept
crl_dir         = $dir/crl              # Where the issued crl are kept
database        = $dir/index.txt        # database index file.
#unique_subject = no                    # Set to 'no' to allow creation of
                                        # several ctificates with same subject.
new_certs_dir   = $dir/newcerts         # default place for new certs.

certificate     = $dir/certs/ca1.cert.pem       # The CA certificate
serial          = $dir/serial           # The current serial number
crlnumber       = $dir/crlnumber        # the current crl number
                                        # must be commented out to leave a V1 CRL
crl             = $dir/crl.pem          # The current CRL
private_key     = $dir/private/ca1.key.pem     # The private key
RANDFILE        = $dir/private/.rand    # private random number file

x509_extensions = usr_cert              # The extentions to add to the cert
```

Task5:Policies

```
policy          = policy_anything
```

```
ats@serverA:~/RP_ca/ca1$ cp ../openssl.cnf .
```



Policy extensions = OID(object identifier)+ optional qualifiers, and policies defined under which the certificate has been issued, and defined the purpose which certificates will use.

Policy_match require the Country Name、State or Province Name、Organization Name should be same. The policy_anything would be flexible, and should change the value match as optional.

```
ats@serverA:~/RP_ca$ openssl genrsa -aes256 -out private/root.key.pem 4096
Generating RSA private key, 4096 bit long modulus
.....................................................................++
.............++
e is 65537 (0x10001)
Enter pass phrase for private/root.key.pem:
Verifying - Enter pass phrase for private/root.key.pem:
```

```
ats@serverA:~/RP_ca$ openssl genrsa -aes256 -out ca1/private/ca1.key.pem 4096
Generating RSA private key, 4096 bit long modulus
.................................++
.....................................................++
e is 65537 (0x10001)
Enter pass phrase for ca1/private/ca1.key.pem:
Verifying - Enter pass phrase for ca1/private/ca1.key.pem:
```

```
ats@serverA:~/RP_ca$ openssl rsa -in private/root.key.pem -pubout -out root.pub.pem
Enter pass phrase for private/root.key.pem:
writing RSA key
```

```
ats@serverA:~/RP_ca$ chmod 400 private/root.key.pem
ats@serverA:~/RP_ca$ chmod 400 ca1/private/ca1.key.pem
```

```
ats@serverA:~/RP_ca$ openssl req -config openssl.cnf -key private/root.key.pem -new -x509 -days 7300 -sha256 -extensions v3_ca -out certs/root
.cert.pem
Enter pass phrase for private/root.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.
-----
State or Province Name (full name) [Blekinge]:Blekinge
Organizational Unit Name (eg, section) []:BTH
Common Name (e.g. server FQDN or YOUR name) []:RONG PENG ROOT
Email Address []:921411446@qq.com
```

Task6:Options for the root certificate

req: create and processes certificate requests in PKCS#10 format, and it can additionally create self signed certificates for use as root CAs.

-config filename: allows an alternative configuration file to be specified.

-key filename: specifies the file to read the private key from.

-new :generates a new certificate request. If –key no use it will generate a new RSA.

-days n : specifies the number of days to certify the certificate for.

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

So the above command is to generate the self-signed certificate, firstly request is specifying the openssl.cnf configuration file, then find the root.key.pem file which under the private folder to read the private key. Secondly generate a new certificate request which is in line with x509 format, and can survive during 7300 days, secure hash algorithm using 256, its extensions section refers to v3_ca. Finally output the root.cert.pem file which under the cert folder, that's the self-signed certificate we will have.

Task7:Verify the root certificate

```
ats@serverA:~/RP_ca$ openssl x509 -noout -text -in certs/root.cert.pem
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 17618500443410988225 (0xf4817cce7492fcc1)
    Signature Algorithm: sha256WithRSAEncryption
        Issuer: ST=Blekinge, OU=BTH, CN=RONG PENG ROOT/emailAddress=921411446@qq.com
        Validity
            Not Before: Nov 27 21:49:40 2017 GMT
            Not After : Nov 22 21:49:40 2037 GMT
        Subject: ST=Blekinge, OU=BTH, CN=RONG PENG ROOT/emailAddress=921411446@qq.com
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (4096 bit)
                Modulus:
                    00:c3:3a:fc:a6:b6:96:3e:da:08:e7:98:00:95:b0:
                    58:1a:7b:c0:ce:e5:d3:70:31:ec:a7:d4:52:9a:cf:
                    db:42:9a:5a:b5:f5:1c:2e:a2:59:e7:99:a9:d2:4f:
                    37:b2:da:00:19:40:59:06:df:a3:8b:23:d3:b2:47:
                    be:c6:64:5f:97:49:d1:54:81:a5:b7:ad:55:a9:35:
                    de:5a:3a:43:51:55:aa:d5:f9:d9:cd:4b:0f:a4:10:
                    04:f8:8d:43:07:0a:33:cb:50:b8:bb:8c:cb:fe:7a:
                    ef:1a:69:39:8e:40:a9:ea:35:cc:e9:41:4b:42:56:
                    24:d4:92:f7:75:2f:ac:bc:4b:bd:e4:1e:94:a9:cc:
                    34:ad:9c:c2:15:f9:81:7c:7a:8a:64:3a:53:84:a4:
                    2f:8e:80:84:9a:54:c8:d2:37:74:08:46:47:48:ba:
                    68:07:a7:d6:fa:fa:11:70:94:2a:04:f5:7c:55:5c:
                    22:f4:3a:13:b3:05:5e:3c:a2:d4:46:07:d4:5f:0e:
                    51:fb:e2:6c:bb:0c:df:02:8b:8d:b1:79:27:6a:e2:
                    42:6f:e2:63:68:f5:32:c4:2b:31:35:75:e0:3f:06:
                    bb:8b:ee:fc:25:f3:3e:6b:8f:f0:b3:9b:cb:28:c4:
                    74:6c:e5:7d:64:4d:2d:82:3b:10:37:12:d9:d8:8b:
                    9a:da:df:94:e5:a1:18:cc:90:14:95:8e:4d:33:1f:
                    51:66:de:59:49:a5:12:13:f3:ea:2c:1f:0c:77:0a:
```

```
                    64:31:4f:d8:62:01:27:8f:f3:8d:f9:ff:5a:8d:59:
                    be:60:ee:81:b5:89:3e:76:46:eb:a2:d8:21:0c:bb:
                    14:fa:16:cc:9b:71:88:f3:ed:1c:4d:42:a9:c4:d5:
                    a6:a0:f2:39:cd:76:93:3f:03:19:9c:15:69:8f:80:
                    5d:31:86:87:98:b2:4b:b5:b1:e1:c2:80:c1:d2:d1:
                    22:c8:e5:cb:0d:92:63:09:4c:02:ad:37:30:34:c4:
                    46:0e:27:77:b8:8a:38:54:3a:b5:0e:ce:88:b0:c5:
                    8d:39:8d:f8:ca:ed:eb:c9:4c:aa:c2:7b:ab:e8:ff:
                    51:06:22:a2:36:95:70:ad:6c:4b:e9:c1:c1:9d:ae:
                    b4:01:ab:72:fb:c0:2e:32:81:cf:dd:2f:13:69:72:
                    e3:7a:42:fc:87:d0:82:44:e7:9d:7e:8d:3e:fc:80:
                    54:7d:65:ec:17:03:46:67:de:29:13:80:1b:f8:59:
                    b6:14:30:ef:cd:e2:9d:8d:41:a2:99:1f:d6:06:66:
                    2a:68:04:92:82:b3:32:98:c0:43:92:8a:1c:f9:25:
                    d8:f1:cd:d7:d1:42:ee:ea:ac:9f:28:1e:07:2e:d3:
                    ba:e7:d1
                Exponent: 65537 (0x10001)
    X509v3 extensions:
        X509v3 Subject Key Identifier:
            48:D4:11:C3:A2:EE:CB:14:97:35:38:38:F6:54:18:2F:EE:42:BF:C7
        X509v3 Authority Key Identifier:
            keyid:48:D4:11:C3:A2:EE:CB:14:97:35:38:38:F6:54:18:2F:EE:42:BF:C7

        X509v3 Basic Constraints:
            CA:TRUE
        X509v3 Key Usage: critical
            Digital Signature, Certificate Sign, CRL Sign
Signature Algorithm: sha256WithRSAEncryption
     be:23:f7:24:a2:17:0b:eb:10:95:91:77:cc:2f:a1:f4:9d:4b:
     14:c2:e1:45:5e:7d:4b:72:d0:83:af:06:17:f7:31:1d:10:f2:
     b4:49:16:d8:e5:71:6e:45:b8:8b:22:0a:67:a2:29:42:eb:0d:
     8c:49:c3:92:86:b1:83:22:98:fa:0a:ac:14:e6:11:bd:d5:f3:
```

```
Signature Algorithm: sha256WithRSAEncryption
     be:23:f7:24:a2:17:0b:eb:10:95:91:77:cc:2f:a1:f4:9d:4b:
     14:c2:e1:45:5e:7d:4b:72:d0:83:af:06:17:f7:31:1d:10:f2:
     b4:49:16:d8:e5:71:6e:45:b8:8b:22:0a:67:a2:29:42:eb:0d:
     8c:49:c3:92:86:b1:83:22:98:fa:0a:ac:14:e6:11:bd:d5:f3:
     cc:cd:8e:98:c6:31:1f:6c:7f:f8:7b:22:9b:d0:cc:56:ac:90:
     e4:89:cf:b9:ee:24:0d:ba:04:94:d5:c9:54:87:f9:5d:71:00:
     e7:4f:c5:6a:c0:e2:58:83:c2:6e:8c:42:e8:8d:bf:10:a1:d1:
     92:80:0f:5e:dc:79:02:2e:eb:01:60:78:82:38:e3:1f:7d:b4:
     fb:1e:93:f2:6e:ef:13:56:a4:1d:74:8a:c8:8e:d1:a8:7d:58:
     b0:c5:56:0e:89:38:c1:63:7d:f2:89:54:ab:66:0a:35:17:3f:
     96:08:f1:02:2e:92:52:02:97:8f:a5:93:c8:b4:de:4f:45:ca:
     85:62:8d:db:01:66:ad:f5:18:91:65:55:d2:f2:33:37:12:a8:
     ac:67:52:6f:3c:8b:7b:a2:e9:71:a2:8d:02:ee:8b:87:cb:30:
     db:fd:eb:62:77:af:5e:37:d5:43:7c:a2:3c:ef:2d:dd:34:22:
     78:33:8c:c0:87:87:c6:e1:e2:a9:25:2c:14:83:81:a7:36:e7:
     35:30:0f:fc:45:4e:4c:0c:ea:b7:26:a6:57:6d:bc:ec:6b:2d:
     5e:6f:27:34:14:71:0a:49:92:ac:04:66:ec:96:87:a1:9d:f8:
     ee:78:c6:a5:69:d6:99:e8:1d:bf:ee:9e:3b:02:2e:d2:24:82:
     8c:af:09:7c:78:cb:91:ea:dd:cf:77:0a:71:9e:fa:63:6c:39:
     90:f5:3d:59:e1:da:73:4f:9a:07:75:08:db:5f:ad:5b:28:b4:
     5c:ad:b2:10:64:37:0d:60:ab:8c:21:aa:cd:c0:2d:93:b8:ee:
     e3:9e:d8:93:8a:b2:6a:2a:31:b0:05:ce:22:ab:01:86:5c:ca:
     90:69:bc:16:30:7d:b6:2e:00:43:c4:b7:5d:9d:2e:c4:ca:43:
     3a:c7:45:f7:0a:ca:f6:a2:88:bb:ff:10:2e:f5:90:ae:7b:be:
     67:9d:15:4c:db:57:74:24:35:84:86:c9:7f:63:cd:ed:40:93:
     6e:d0:c9:8b:e1:f6:d0:10:82:7a:11:bf:4f:9c:a1:10:7f:b0:
     3c:db:b4:d5:ae:af:98:99:bb:d3:41:b8:be:99:8a:3f:85:ea:
     0a:50:54:b3:55:7d:09:75:9d:12:ac:2a:8b:40:17:4d:d2:78:
     26:be:69:a9:15:1b:bb:27
```

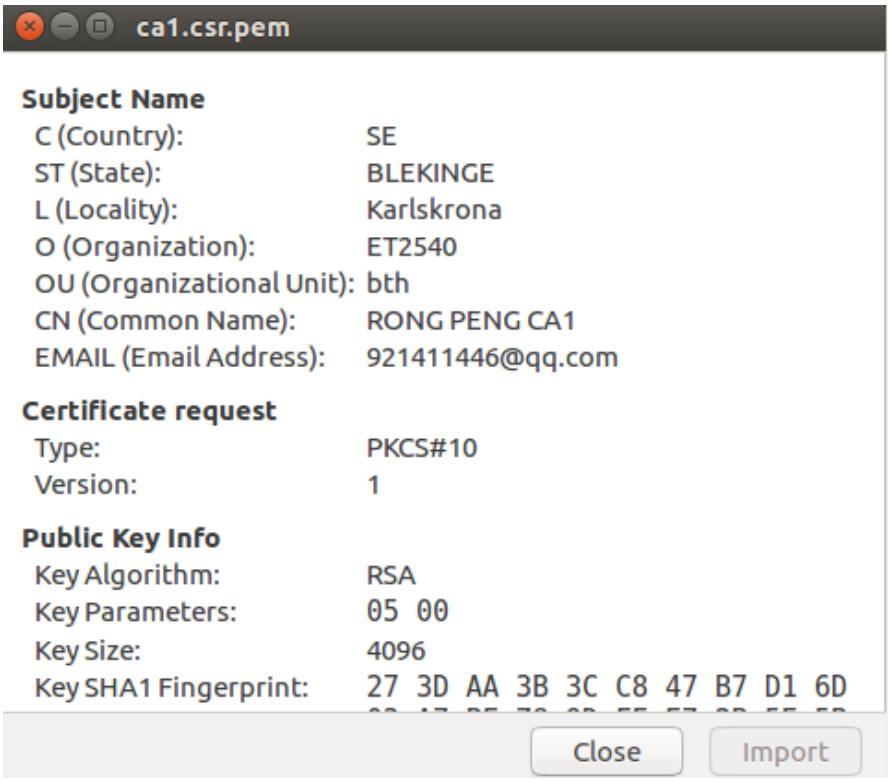
```
ats@serverA:~/RP_ca$ openssl req -config ca1/openssl.cnf -new -sha256 -key ca1/private/ca1.key.pem -out ca1/csr/ca1.csr.pem
Enter pass phrase for ca1/private/ca1.key.pem:
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to 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.
-----
State or Province Name (full name) [Blekinge]:Blekinge
Organizational Unit Name (eg, section) []:BTH
Common Name (e.g. server FQDN or YOUR name) []:RONG PENG CA1
Email Address []:,

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:atslabb00
An optional company name []:,
```

Task8:Verify the CSR

```
ats@serverA:~/RP_ca$ openssl req -text -noout -verify -in ca1/csr/ca1.csr.pem
verify OK
Certificate Request:
    Data:
        Version: 0 (0x0)
        Subject: ST=Blekinge, OU=BTH, CN=RONG PENG CA1/emailAddress=,
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (4096 bit)
                Modulus:
                    00:b0:c4:23:87:a5:9e:76:00:6b:7b:7d:b3:17:51:
                    88:8a:6f:c2:3f:b0:ab:98:2b:fe:d7:b2:86:f1:82:
                    12:1d:cd:bd:6c:42:97:5b:90:f5:d9:c3:20:78:66:
                    26:52:22:0c:81:12:42:90:9b:df:b1:05:45:c7:c3:
                    d9:b1:ea:78:42:01:88:08:28:22:0f:cb:ac:f2:39:
                    7b:69:db:97:ed:1d:9f:0a:4b:d7:10:ef:88:e9:02:
                    e8:d0:0b:fc:bc:d5:b0:d1:47:89:0d:0a:c5:24:18:
                    c4:08:60:4a:13:30:a2:49:36:b2:76:c6:e7:9d:74:
                    f6:d0:7e:53:03:6c:5e:df:8c:fd:f4:e8:a4:af:55:
                    b8:0c:7c:e3:7e:de:53:71:a9:48:0f:c9:2d:6e:42:
                    95:e3:a5:60:13:59:a1:c0:2f:a0:5d:70:d9:1c:ff:
                    34:41:e3:be:aa:e7:c7:63:8e:b6:fd:74:ee:85:8f:
                    2e:58:b2:d5:89:46:a8:65:10:a3:f7:b9:ff:90:18:
                    c4:88:cf:68:58:09:7f:82:78:78:6d:fb:28:32:ed:
                    2d:2a:47:10:5c:cf:d6:71:92:a2:4f:99:c3:7b:7e:
                    3e:79:ec:fc:e0:f9:e6:28:ff:71:9a:55:6e:d8:8a:
                    3f:83:23:e4:73:19:1d:83:d9:f4:b8:6f:70:66:de:
```

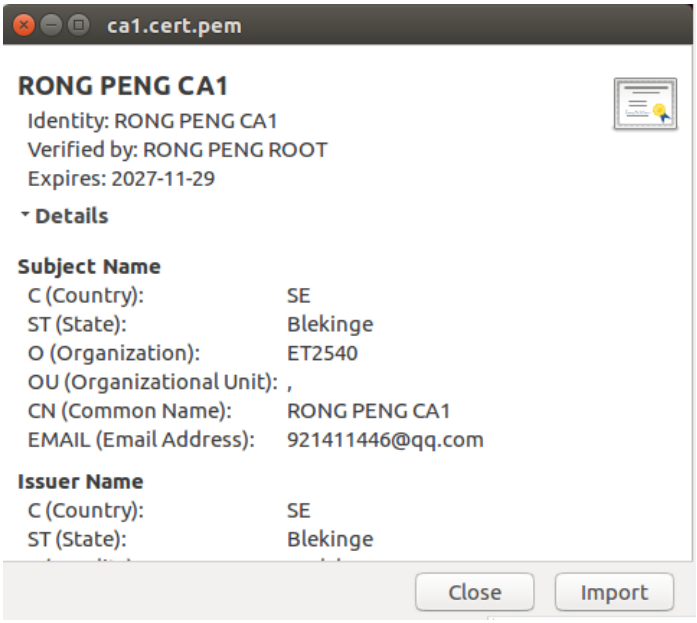
```
                    55:31:4e:63:17:1e:7c:b0:2b:f0:8f:3e:46:12:47:
                    43:30:95:52:41:ca:ad:18:04:46:ff:ea:b8:70:63:
                    7c:f6:12:41:ad:23:06:35:db:5d:1a:dc:d7:48:8c:
                    ab:dc:b7:55:09:8b:28:85:21:80:18:27:12:07:55:
                    ea:ba:e9:df:9b:48:98:c1:d3:3d:18:33:d1:d2:ef:
                    32:64:19:be:af:c3:02:6a:5a:81:5a:a0:c5:e6:57:
                    2b:08:4b:32:e0:b3:30:45:a6:6c:ee:43:95:b9:dd:
                    40:21:06:9e:26:cc:92:39:b8:fe:c6:94:88:ec:db:
                    55:b2:63:0e:37:d6:cf:e9:28:16:b3:cb:1d:71:f3:
                    70:5d:e9:12:20:bd:31:24:af:36:0a:24:ce:02:eb:
                    94:51:2e:fe:21:8d:c3:88:58:0b:49:b3:d3:d8:78:
                    fb:35:89:b7:d1:e9:a4:7a:34:85:ec:2c:fd:2f:08:
                    2c:5f:9d:b8:80:e3:2f:37:4d:33:96:3b:2b:86:bb:
                    68:d0:ec:0d:e9:f0:72:82:ca:0c:b9:e0:66:42:68:
                    ee:40:7f:76:b6:a8:59:61:eb:42:7d:6e:c9:1e:f0:
                    66:89:6a:64:99:df:cc:e7:cb:3f:dc:94:72:c0:bc:
                    4b:f0:ae:f0:12:a8:39:dc:36:a8:47:43:a8:dd:d9:
                    1c:66:79
                Exponent: 65537 (0x10001)
    Attributes:
        unstructuredName          :unable to print attribute
        challengePassword         :unable to print attribute
```

```
Signature Algorithm: sha256WithRSAEncryption
     79:63:94:69:29:dc:1f:d6:23:8a:4b:22:ed:e1:88:56:d7:94:
     a3:5b:d4:de:ed:75:95:47:2a:31:aa:74:aa:8e:b6:7e:29:08:
     20:c8:1e:8e:0c:60:ec:1d:c1:e4:2a:59:be:52:b3:19:9e:f4:
     d3:8a:49:1f:2e:84:cd:38:0a:c6:ba:f4:a5:d2:56:8c:bd:82:
     45:05:3d:e7:dd:85:f6:7f:b6:bc:eb:2a:06:04:4d:62:01:02:
     c4:87:c3:db:8a:ff:29:ed:33:d8:f7:17:26:b7:f1:43:c1:39:
     72:0d:8b:10:69:95:f0:dc:6f:af:fd:9f:a4:03:4e:4d:8d:f5:
     02:39:59:2e:97:26:cf:d6:c1:68:e3:0b:1d:4e:d9:cb:b4:5e:
     e8:e8:e8:43:33:cc:71:82:98:c0:5f:50:92:46:86:4d:07:82:
     5f:5f:c6:46:41:d2:67:f2:e4:bd:c2:e6:3d:6b:7a:4f:d6:f9:
     8c:3c:23:f7:38:f5:44:30:83:6b:d8:17:d2:e8:4b:75:89:fb:
     37:6b:fa:45:a7:86:7f:94:c4:ac:81:58:17:80:c7:57:42:a1:
     f2:a6:42:37:66:17:d0:e5:34:5c:7c:69:be:c4:6e:64:23:54:
     ee:d4:39:24:9a:62:05:fc:b9:da:67:ea:df:2c:8b:4c:ee:54:
     3c:53:ec:fb:48:1d:e9:10:33:9e:a0:2e:ac:96:cf:7c:1a:62:
     4c:82:1d:be:e1:fc:cf:d6:fd:36:d2:e1:d6:a3:5c:1a:c0:19:
     c2:aa:e6:23:1a:c5:fd:32:ae:e3:18:90:60:2c:76:3a:4e:0e:
     90:34:27:99:64:af:60:22:1e:d5:77:0b:84:9f:f3:2f:52:cb:
     d7:dc:8d:5a:35:15:6c:cd:a3:61:e5:ca:0a:93:04:9f:ef:42:
     51:95:ea:7d:87:10:c0:a2:57:83:75:dc:36:93:37:c2:4a:91:
     51:36:56:34:68:ff:1c:e3:2b:18:65:98:fb:23:44:2e:8e:2c:
     07:73:d8:ea:0e:56:77:3f:d4:37:a6:a5:e7:b9:2c:c6:bd:04:
     c0:52:5a:2e:b2:e0:a2:86:b7:c5:d9:a5:b4:55:6c:35:5f:4b:
     59:67:d9:cb:b7:9a:e3:bd:0a:57:e9:af:e6:02:b4:53:30:4f:
     a2:76:87:af:24:8b:8a:ea:c5:8e:c0:4a:24:81:73:b9:c8:a1:
     89:17:04:47:fa:5b:c7:c9:62:d3:3c:fe:19:95:a8:8d:68:a8:
     92:9c:84:85:34:bb:16:c3:c8:ee:2f:e0:cb:3f:b3:5d:d3:c2:
     fb:33:03:fd:31:64:09:55:23:84:4e:99:d1:a8:06:0e:2d:78:
     50:92:d8:65:31:a9:c1:a1
```



```
ats@serverA:~/RP_ca$ 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
```

```
Using configuration from openssl.cnf
Enter pass phrase for /home/ats/RP_ca/private/root.key.pem:
Check that the request matches the signature
Signature ok
Certificate Details:
        Serial Number: 4096 (0x1000)
        Validity
            Not Before: Dec  1 14:42:54 2017 GMT
            Not After : Nov 29 14:42:54 2027 GMT
        Subject:
            countryName               = SE
            stateOrProvinceName       = Blekinge
            organizationName          = ET2540
            organizationalUnitName    = ,
            commonName                = RONG PENG CA1
            emailAddress              = 921411446@qq.com
        X509v3 extensions:
            X509v3 Subject Key Identifier:
                54:86:E6:F0:8E:6F:88:02:1B:75:E6:2C:01:D3:BE:1A:D3:99:B5:8D
            X509v3 Authority Key Identifier:
                keyid:52:FA:0C:25:A7:CB:9E:97:1B:CF:CA:FC:74:14:23:91:15:61:7A:7F

            X509v3 Basic Constraints: critical
                CA:TRUE, pathlen:0
            X509v3 Key Usage: critical
                Digital Signature, Certificate Sign, CRL Sign
Certificate is to be certified until Nov 29 14:42:54 2027 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
```



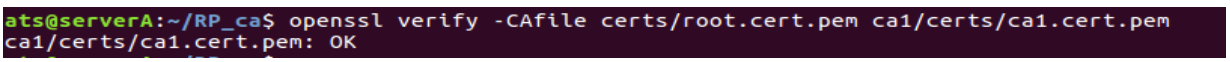
Task9:Options for intermediate CA certificate

```
CA(1SSL)                                    OpenSSL                                    CA(1SSL)

NAME
       ca - sample minimal CA application

SYNOPSIS
       openssl ca [-verbose] [-config filename] [-name section] [-gencrl] [-revoke file] [-status serial] [-updatedb] [-crl_reason reason]
       [-crl_hold instruction] [-crl_compromise time] [-crl_CA_compromise time] [-crldays days] [-crlhours hours] [-crlexts section]
       [-startdate date] [-enddate date] [-days arg] [-md arg] [-policy arg] [-keyfile arg] [-keyform PEM|DER] [-key arg] [-passin arg]
       [-cert file] [-selfsign] [-in file] [-out file] [-notext] [-outdir dir] [-infiles] [-spkac file] [-ss_cert file] [-preserveDN]
       [-noemailDN] [-batch] [-msie_hack] [-extensions section] [-extfile section] [-engine id] [-subj arg] [-utf8] [-multivalue-rdn]

DESCRIPTION
       The ca command is a minimal CA application. It can be used to sign certificate requests in a variety of forms and generate CRLs it
       also maintains a text database of issued certificates and their status.

       The options descriptions will be divided into each purpose.

CA OPTIONS
       -config filename
           specifies the configuration file to use.

       -name section
           specifies the configuration file section to use (overrides default_ca in the ca section).

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

       -ss_cert filename
           a single self signed certificate to be signed by the CA.

       -spkac filename
           a file containing a single Netscape signed public key and challenge and additional field values to be signed by the CA. See the
           SPKAC FORMAT section for information on the required input and output format.
```

ca: sign certificate requests in a variety of forms and generate CRLs. It also maintains a text database of issued certificates and their status.
-config filename: specifies the configuration file to use.
-notext: don't output the text form of a certificate to the output file.
-md: the message digest to use.
-in : an input filename containing a single certificate request to be single by the CA.
-out filename: the output file to output certificates to. The certificate details will also be printed out to this file in PEM format.

The above command wants to sign certificate for CA1, using configuration file openssl.cnf and specifying the section v3_intermediate_ca as its intermediate CA, and this intermediate can survive during 3650 days, don't output this text form of a certificate to the output file. Its message digest uses sha256. The output file is ca1.cert.pem.

Task10:Verify the certificate for CA1



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

root.cert.pem is a self-signed CA, and *verify* command will go up the certificate chain all the way, up to a self-signed CA.

```
ats@serverA:~/RP_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, L=Karlskrona, O=ET2540, OU=BTH, CN=RONG PENG ROOT/emailAddress=921411446@qq.com
        Validity
            Not Before: Dec  1 14:42:54 2017 GMT
            Not After : Nov 29 14:42:54 2027 GMT
        Subject: C=SE, ST=Blekinge, O=ET2540, OU=,, CN=RONG PENG CA1/emailAddress=921411446@qq.com
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (4096 bit)
                Modulus:
                    00:d8:33:8d:c4:a3:02:f4:5f:6b:5a:8e:b5:77:13:
                    31:a0:19:e0:29:ba:9e:bf:3a:ce:0b:67:af:a3:57:
                    ab:fc:a0:10:be:7d:50:5d:72:a6:59:a6:97:4a:b3:
                    d8:ad:38:27:cb:d6:e7:64:c6:13:d6:de:70:6f:d7:
                    65:7e:4e:aa:54:1e:fb:f4:64:d3:21:cb:7a:42:e9:
                    be:f4:46:f8:e2:93:ea:d7:dc:23:d8:0f:e4:5f:8d:
                    05:19:c1:0a:06:f7:c6:45:4d:85:f4:e8:f9:28:f4:
                    66:f2:7b:37:d6:3d:c0:14:6d:d5:49:ab:96:4a:61:
                    92:70:f1:15:eb:b8:7b:fd:7a:ff:9d:f6:75:c9:2c:
                    d3:83:c0:8f:ef:a6:42:d7:50:c7:7e:d1:c9:80:83:
                    3f:55:c5:e1:a6:b8:25:a3:18:e6:0b:15:75:2f:2c:
                    7b:8e:e8:be:93:86:14:9a:0b:a7:66:e1:35:a4:cc:
                    68:1c:a1:4d:77:40:b6:e1:3c:8b:32:91:c0:9f:bc:
                    bd:dd:a9:cd:2c:74:e4:f0:17:a5:de:b9:41:d5:41:
                    09:d8:3c:b7:46:6a:87:57:ec:a5:2c:6f:b3:83:fe:
                    a7:f7:c7:33:a4:99:b7:3b:ff:6c:e7:80:de:09:63:
                    34:03:33:55:67:21:3e:ca:f0:66:8f:f0:5a:09:29:
                    95:32:6b:8b:8e:9a:09:5b:cc:c0:de:90:01:f5:76:
                    dc:87:1c:a1:40:c7:af:70:90:84:b7:1f:63:67:8b:
                    e8:e8:f1:66:cf:7b:15:9e:36:8c:a1:63:ec:a3:54:
                    ff:54:30:c5:c9:cc:ba:2d:de:07:03:e3:0d:4e:a0:
                    ce:5e:da:ff:18:4b:0d:5a:d7:b1:c6:68:23:87:ba:
```

```
                    f0:48:43:ce:b8:cd:22:1b:87:b0:2f:0f:ae:f4:79:
                    3e:88:f5:dd:c6:08:39:25:2b:32:7b:4a:3f:21:16:
                    b6:c0:27:f3:92:82:83:66:1a:7e:04:2b:c8:2c:ab:
                    ee:62:cb:19:09:3f:95:1f:f0:a4:8f:cc:75:12:0d:
                    1f:88:a5:82:6b:e0:d6:f9:8a:e4:1d:2c:a5:86:66:
                    26:35:1d
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Subject Key Identifier:
                54:86:E6:F0:8E:6F:88:02:1B:75:E6:2C:01:D3:BE:1A:D3:99:B5:8D
            X509v3 Authority Key Identifier:
                keyid:52:FA:0C:25:A7:CB:9E:97:1B:CF:CA:FC:74:14:23:91:15:61:7A:7F

            X509v3 Basic Constraints: critical
                CA:TRUE, pathlen:0
            X509v3 Key Usage: critical
                Digital Signature, Certificate Sign, CRL Sign
    Signature Algorithm: sha256WithRSAEncryption
         13:60:7e:8c:de:3c:5a:ae:41:33:2d:28:0f:46:e3:0b:66:16:
         fc:41:7e:3d:90:07:57:d7:1e:35:31:d8:50:ec:3f:0b:01:3a:
         dc:7b:3a:8c:49:91:7b:c8:2f:6f:a6:a7:95:62:13:2a:7d:fa:
         00:65:0e:05:d7:e5:35:d3:85:dc:81:90:f2:be:26:f8:ad:bc:
         88:35:71:9c:be:33:48:0c:5f:d4:ab:c3:0e:37:7c:f2:10:37:
         25:9e:f0:f4:f7:0c:4d:66:5c:e6:e7:9b:e8:1f:26:f7:d4:23:
         5a:c2:9a:e6:21:41:00:1c:f7:44:bc:39:43:a8:a2:0c:31:6c:
         bc:1b:55:99:9d:d3:db:5b:6d:b0:da:75:cb:08:63:c1:a3:66:
         62:10:e0:7c:e0:66:03:67:54:b2:e3:d5:cf:e0:a9:26:b0:0e:
         04:d1:a2:fa:30:f8:08:8f:21:69:57:2e:a5:98:f2:4d:12:40:
         31:7c:15:77:86:96:5d:5e:5b:8c:7e:ff:fc:7d:ba:d0:3e:81:
         d3:89:96:f9:d1:b1:ce:61:54:d1:92:f8:02:32:e2:17:ad:db:
         87:e6:85:60:56:c3:01:53:a9:bf:8e:c8:21:df:87:93:ed:37:
         51:29:42:9f:32:71:05:b4:5c:8e:57:14:e3:06:96:3d:e7:e3:
         0d:93:cf:32:b5:14:57:eb:15:5c:38:d6:f5:4e:bb:09:60:07:
         7e:74:5a:47:99:87:9d:99:56:ba:51:d3:7c:ef:45:51:9a:c1:
         a7:f6:24:04:59:e5:ec:60:81:89:cb:a4:93:cb:2e:58:4e:42:
         83:c3:d0:a2:fd:19:4c:ef:79:22:83:b7:ba:8a:ee:7b:eb:9d:
         c7:7f:6a:50:04:5c:dc:cb:93:b8:bf:b9:32:9b:e8:20:d1:ac:
```

The above command *x509 –noout –text –in* displays the ca1's information.

Task11: Create server certificate

Create the server certificate steps:

1.Create an RSA private key for the server (in this lab, I skip -ae256)

I went to the RP_ca/ca1/ directory, and coded:

openssl genrsa -out private/server.key.pem 2048

then generated the *server.key.pem*

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

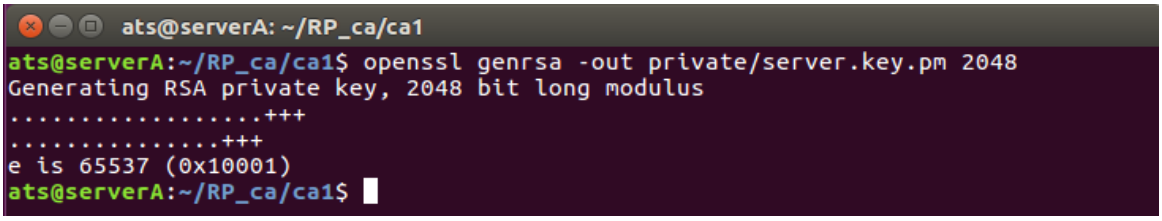
Also in RP_ca/ca1/ directory, wrote down :

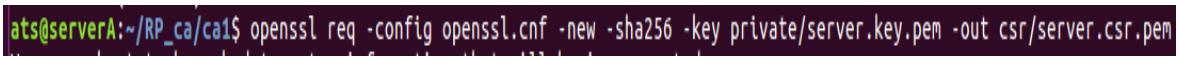
openssl req -config openssl.conf -new -sha256 -key private/server.key.pem -out csr/server.csr.pem

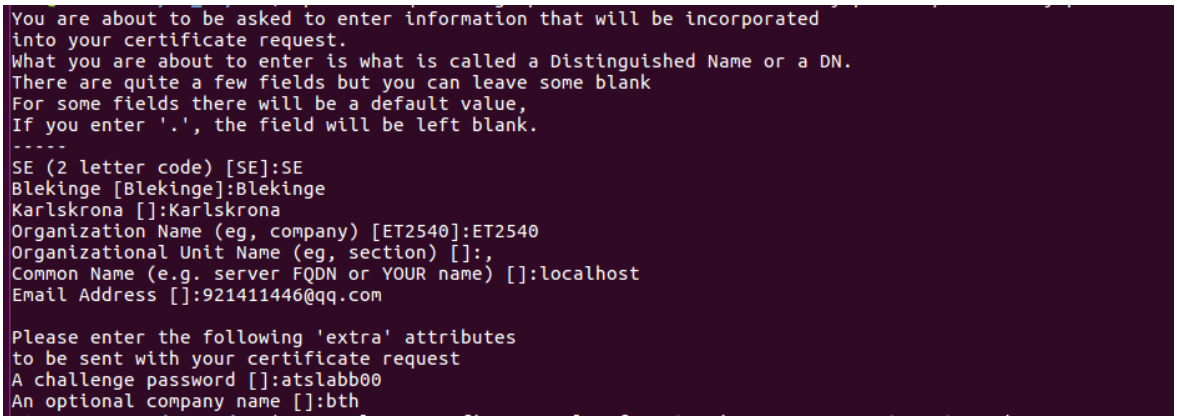
generated the *server.csr.pem*

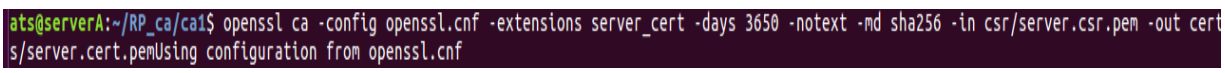
3.Use CA1's private key to sign the CSR and create a certificate : server.cert.pem

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









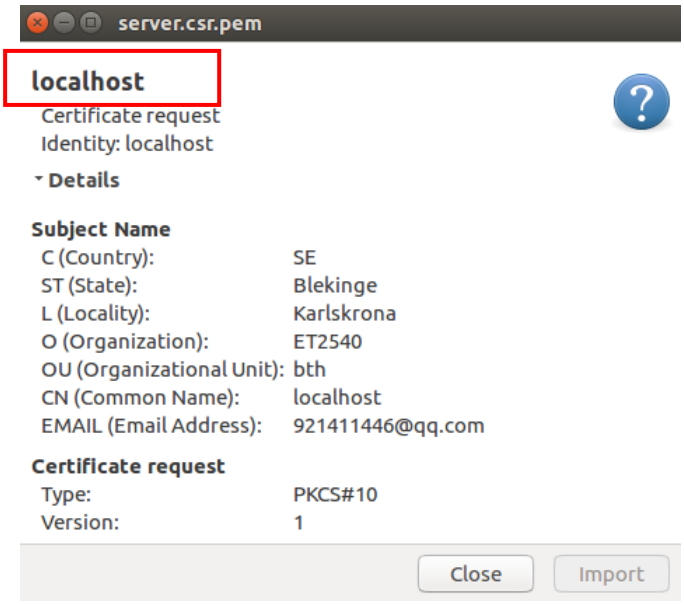
```
ats@serverA:~/RP_ca/ca1$ openssl ca -config openssl.cnf -extensions server_cert -days 3650 -notext -md sha256 -in csr/se
s/server.cert.pemUsing configuration from openssl.cnf
Enter pass phrase for /home/ats/RP_ca/ca1/private/ca1.key.pem:
Check that the request matches the signature
Signature ok
Certificate Details:
        Serial Number: 8193 (0x2001)
        Validity
            Not Before: Dec  1 16:38:55 2017 GMT
            Not After : Nov 29 16:38:55 2027 GMT
        Subject:
            countryName               = SE
            stateOrProvinceName       = Blekinge
            localityName              = Karlskrona
            organizationName          = ET2540
            organizationalUnitName    = bth
            commonName                = localhost
            emailAddress              = 921411446@qq.com
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            X509v3 Subject Key Identifier:
                A8:AE:A7:A5:75:10:A4:4D:7C:09:D3:D7:29:EB:BC:A0:9A:F7:74:D1
            X509v3 Authority Key Identifier:
                keyid:54:86:E6:F0:8E:6F:88:02:1B:75:E6:2C:01:D3:BE:1A:D3:99:B5:8D
                DirName:/C=SE/ST=Blekinge/L=Karlskrona/O=ET2540/OU=BTH/CN=RONG PENG ROOT/emailAddress=921411446@qq.com
                serial:10:00
```

```
                DirName:/C=SE/ST=Blekinge/L=Karlskrona/O=ET2540/OU=BTH/CN=RONG PENG ROOT/emailAddres
                serial:10:00

            X509v3 Key Usage: critical
                Digital Signature, Key Encipherment
            X509v3 Extended Key Usage:
                TLS Web Server Authentication
Certificate is to be certified until Nov 29 16:38:55 2027 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
```

Verify the server.csr.pem and the server.cert.pem:

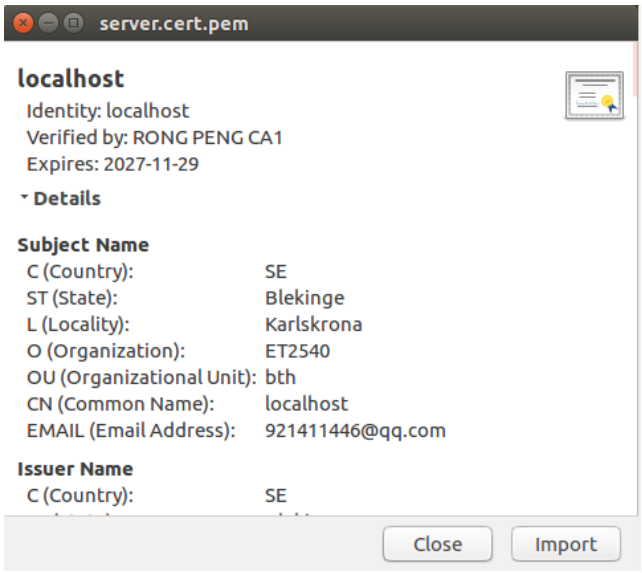
```
ats@serverA:~/RP_ca/ca1$ openssl req -text -noout -verify -in csr/server.csr.pem
verify OK
Certificate Request:
    Data:
        Version: 0 (0x0)
        Subject: C=SE, ST=Blekinge, L=Karlskrona, O=ET2540, OU=bth, CN=localhost/emailAddress=921411446@qq.com
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
                Modulus:
                    00:ca:b4:99:0e:e2:20:74:39:01:9e:34:5d:22:de:
                    40:9e:66:6d:e0:56:f1:7c:9b:17:92:db:c7:4f:1a:
                    77:19:4f:f8:8e:d2:30:47:36:91:a7:b1:14:dd:86:
                    5d:d9:56:d9:54:dd:15:69:15:a4:e6:40:2d:e9:bb:
                    8a:d7:5f:71:03:88:93:ac:ae:59:bc:58:14:dd:69:
                    eb:d2:e5:b9:93:6a:db:a2:e6:c1:ba:00:8b:00:99:
                    e3:7d:e2:b7:3b:f8:0f:6f:72:97:3c:f5:d6:05:2c:
                    33:29:bd:cb:4a:04:0a:8e:cc:e3:e3:c0:8b:be:ab:
                    82:a5:8f:1c:bb:41:aa:b2:a9:43:d4:84:ed:bc:37:
                    32:68:c9:79:25:49:40:ab:6f:f0:93:33:a1:2f:bd:
                    d0:f1:ae:31:8b:ee:ef:75:0c:d5:12:49:80:93:f7:
                    44:11:8b:75:82:ad:5d:b0:39:7d:7b:24:71:f7:dd:
                    2b:49:d1:eb:15:ce:d6:17:78:37:02:45:1b:5e:32:
                    06:b2:2d:be:1b:44:af:b0:a0:37:48:2b:a0:b8:ce:
                    62:f9:a5:70:0b:2f:fd:2c:23:99:07:13:cd:26:c6:
                    66:71:2a:f5:56:7a:14:d7:6a:77:4b:9a:19:2d:57:
                    fc:95:9d:3c:17:62:42:ca:77:ae:b7:3d:6c:93:5f:
                    c3:47
                Exponent: 65537 (0x10001)
        Attributes:
            unstructuredName         :unable to print attribute
            challengePassword        :unable to print attribute
    Signature Algorithm: sha256WithRSAEncryption
```

```
                Exponent: 65537 (0x10001)
        Attributes:
            unstructuredName         :unable to print attribute
            challengePassword        :unable to print attribute
    Signature Algorithm: sha256WithRSAEncryption
         5a:d0:d4:91:03:55:16:e0:fb:87:fb:e5:95:e9:03:63:b5:05:
         45:c4:08:a6:c7:b9:89:a6:a1:77:c6:5c:73:97:a1:1a:cc:b9:
         7d:8a:b8:5d:2b:c4:b6:2a:86:40:d1:b7:9f:5d:7c:d1:0b:a8:
         d8:f3:3b:12:28:15:96:e7:1d:cd:22:2c:24:dc:4d:eb:9e:01:
         7b:32:a9:75:fa:57:75:62:b8:2d:b5:84:99:b0:e7:d6:98:7b:
         af:a6:9e:3c:9f:b9:25:4e:df:fe:89:e4:1a:6d:5f:20:79:24:
         98:34:56:17:c2:de:e8:fc:d2:73:4b:5f:79:cb:46:9f:4c:90:
         16:cb:2d:64:15:07:64:02:48:56:4b:2f:f1:05:46:91:f8:f1:
         31:8c:1d:f4:d1:0f:af:bc:58:96:d7:71:4c:79:66:01:49:50:
         fc:59:71:39:cd:f1:97:44:41:ae:4a:b6:d1:4c:d8:f8:ba:be:
         08:ac:c1:2a:43:b3:5c:60:3b:96:bb:6a:bd:ec:79:a6:24:17:
         e7:06:4d:7c:40:ca:29:87:e9:fc:3b:65:11:61:24:c3:35:32:
         e3:b1:a8:0f:86:8d:bd:a1:75:18:e6:39:e0:98:bc:f4:54:27:
         01:be:7f:9e:ab:02:9b:07:6d:25:5e:85:f2:ec:92:7b:b2:5a:
         a4:a4:48:89
```




```
ats@serverA:~/RP_ca/ca1$ openssl x509 -noout -text -in certs/server.cert.pem
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 8193 (0x2001)
    Signature Algorithm: sha256WithRSAEncryption
        Issuer: C=SE, ST=Blekinge, O=ET2540, OU=,, CN=RONG PENG CA1/emailAddress=921411446@qq.com
        Validity
            Not Before: Dec  1 16:38:55 2017 GMT
            Not After : Nov 29 16:38:55 2027 GMT
        Subject: C=SE, ST=Blekinge, L=Karlskrona, O=ET2540, OU=bth, CN=localhost/emailAddress=921411446@qq.com
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
                Modulus:
                    00:ca:b4:99:0e:e2:20:74:39:01:9e:34:5d:22:de:
                    40:9e:66:6d:e0:56:f1:7c:9b:17:92:db:c7:4f:1a:
                    77:19:4f:f8:8e:d2:30:47:36:91:a7:b1:14:dd:86:
                    5d:d9:56:d9:54:dd:15:69:15:a4:e6:40:2d:e9:bb:
                    8a:d7:5f:71:03:88:93:ac:ae:59:bc:58:14:dd:69:
                    eb:d2:e5:b9:93:6a:db:a2:e6:c1:ba:00:8b:00:99:
                    e3:7d:e2:b7:3b:f8:0f:6f:72:97:3c:f5:d6:05:2c:
                    33:29:bd:cb:4a:04:0a:8e:cc:e3:e3:c0:8b:be:ab:
                    82:a5:8f:1c:bb:41:aa:b2:a9:43:d4:84:ed:bc:37:
                    32:68:c9:79:25:49:40:ab:6f:f0:93:33:a1:2f:bd:
                    d0:f1:ae:31:8b:ee:ef:75:0c:d5:12:49:80:93:f7:
                    44:11:8b:75:82:ad:5d:b0:39:7d:7b:24:71:f7:dd:
                    2b:49:d1:eb:15:ce:d6:17:78:37:02:45:1b:5e:32:
                    06:b2:2d:be:1b:44:af:b0:a0:37:48:2b:a0:b8:ce:
                    62:f9:a5:70:0b:2f:fd:2c:23:99:07:13:cd:26:c6:
                    66:71:2a:f5:56:7a:14:d7:6a:77:4b:9a:19:2d:57:
```

```
            Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            X509v3 Subject Key Identifier:
                A8:AE:A7:A5:75:10:A4:4D:7C:09:D3:D7:29:EB:BC:A0:9A:F7:74:D1
            X509v3 Authority Key Identifier:
                keyid:54:86:E6:F0:8E:6F:88:02:1B:75:E6:2C:01:D3:BE:1A:D3:99:B5:8D
                DirName:/C=SE/ST=Blekinge/L=Karlskrona/O=ET2540/OU=BTH/CN=RONG PENG ROOT/emailAddress=921411446@qq.com
                serial:10:00

            X509v3 Key Usage: critical
                Digital Signature, Key Encipherment
            X509v3 Extended Key Usage:
                TLS Web Server Authentication
    Signature Algorithm: sha256WithRSAEncryption
         d7:d0:29:f2:ce:42:00:b6:f0:1b:e8:3f:28:fb:f4:00:5b:02:
         97:2a:6b:c0:77:cb:d2:c1:ca:69:41:5c:50:50:a4:1d:33:ec:
         1d:e7:e1:18:c6:67:40:20:53:c2:3e:e3:a8:28:91:30:ce:d6:
         ac:e7:99:79:85:23:76:f8:8e:33:3c:75:6b:b3:6e:06:5c:ff:
         77:07:84:e9:38:1d:2b:c1:8f:30:95:e1:f3:ef:11:66:da:4f:
         f4:9b:20:aa:ad:b6:3a:29:80:3f:e4:75:33:cd:d5:19:c3:69:
         dc:42:a3:ac:e5:ee:38:0f:6c:2f:a9:f6:da:3c:14:67:0b:f8:
         79:d9:7f:e1:09:e7:2d:e3:06:72:4d:ab:83:d8:ed:24:a9:1f:
         12:0b:e5:22:df:14:62:d8:8a:47:12:9a:15:04:3c:22:88:fe:
         93:10:e3:ab:21:3f:df:c3:9e:9d:1c:31:0c:df:b7:f1:9d:2c:
         07:af:18:c5:b2:53:cb:14:2c:07:d4:96:cf:79:d9:be:4e:1e:
         7f:9d:6e:48:bc:55:62:bc:37:20:24:ed:1c:98:2a:f8:f9:54:
         a4:1c:ce:c5:0f:a9:2c:d5:00:51:3d:39:05:0d:5b:9f:2d:af:
         99:89:fc:74:7c:9d:07:3b:05:f8:e7:f9:b7:06:be:15:52:01:
```



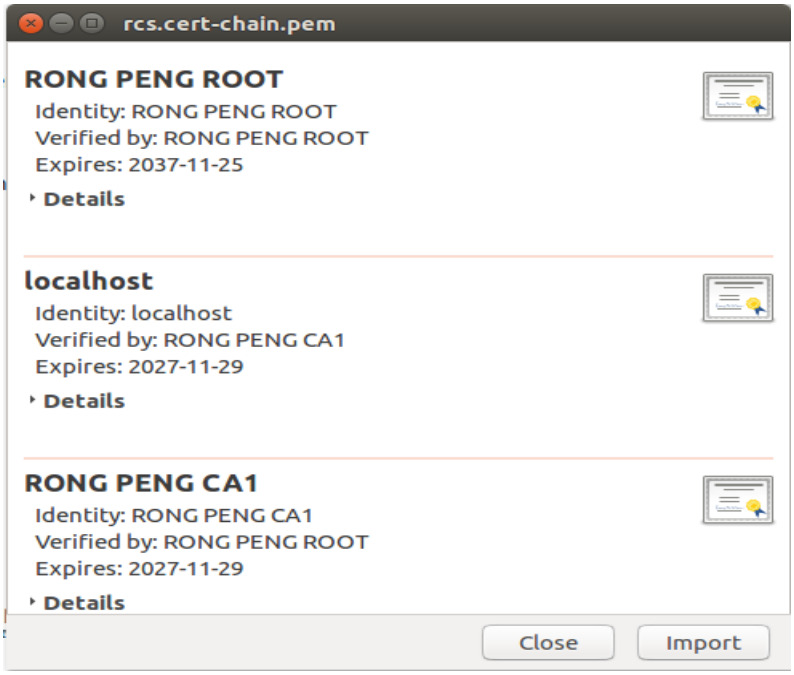
To be convenience, firstly I combined the server cert and the ca1 cert into one cert-chain file, then I went back to the RP_ca directory then combined the three certs and generated the cert-chain file, which is *rcs.cert-chain.pem*

Then I wanna verify server.cert.pem , so I using *openssl verify -CAfile rcs.cert-chain.pem server.cert.pem* to check whether the server cert is signed by ca1.cert.pem

```
ats@serverA:~/RP_ca/ca1/certs$ cat server.cert.pem  ca1.cert.pem > server.cert-c
hain.pem
```

```
ats@serverA: ~/RP_ca
ats@serverA:~/RP_ca$ cat certs/root.cert.pem ca1/certs/server.cert-chain.pem > c
a1/certs/rcs.cert-chain.pem
ats@serverA:~/RP_ca$
```

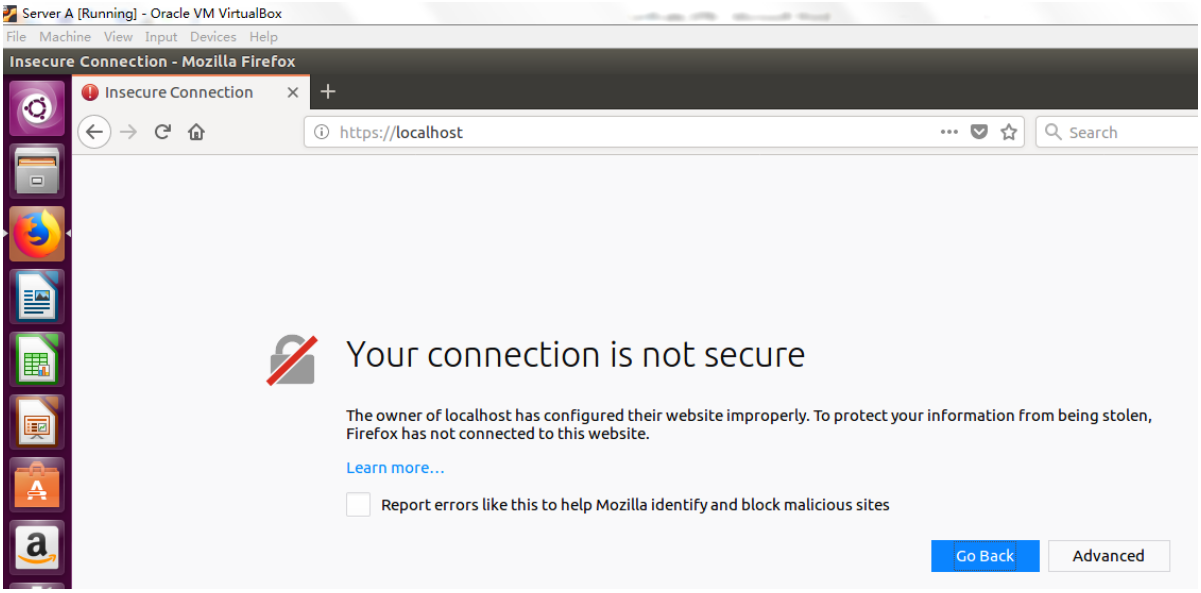
```
ats@serverA: ~/RP_ca/ca1/certs
ats@serverA:~/RP_ca/ca1/certs$ openssl verify -CAfile rcs.cert-chain.pem server.
cert.pem
server.cert.pem: OK
ats@serverA:~/RP_ca/ca1/certs$
```

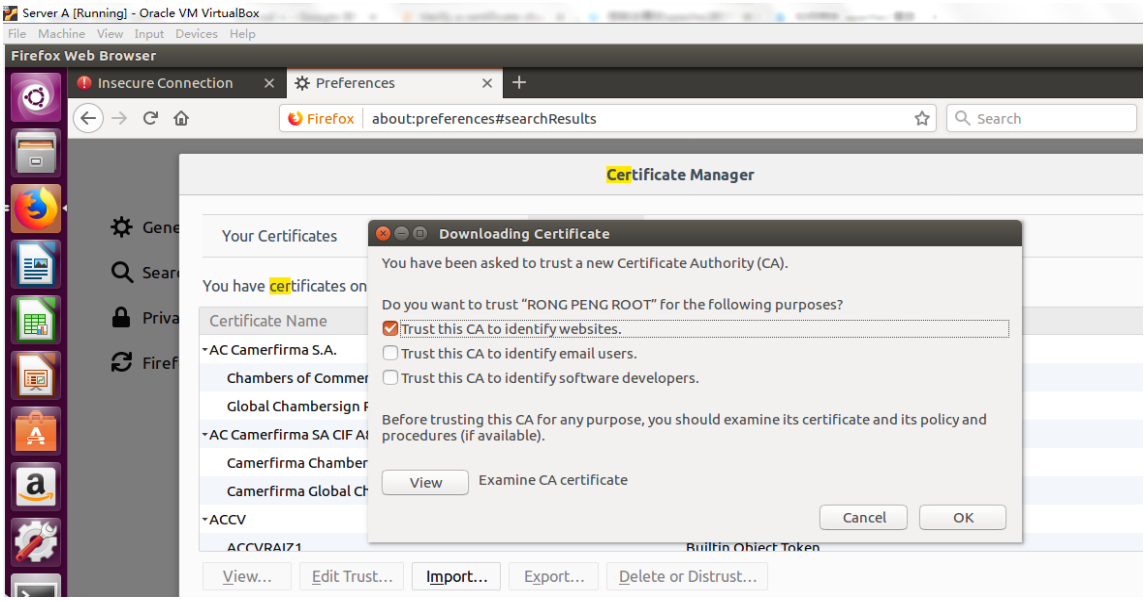


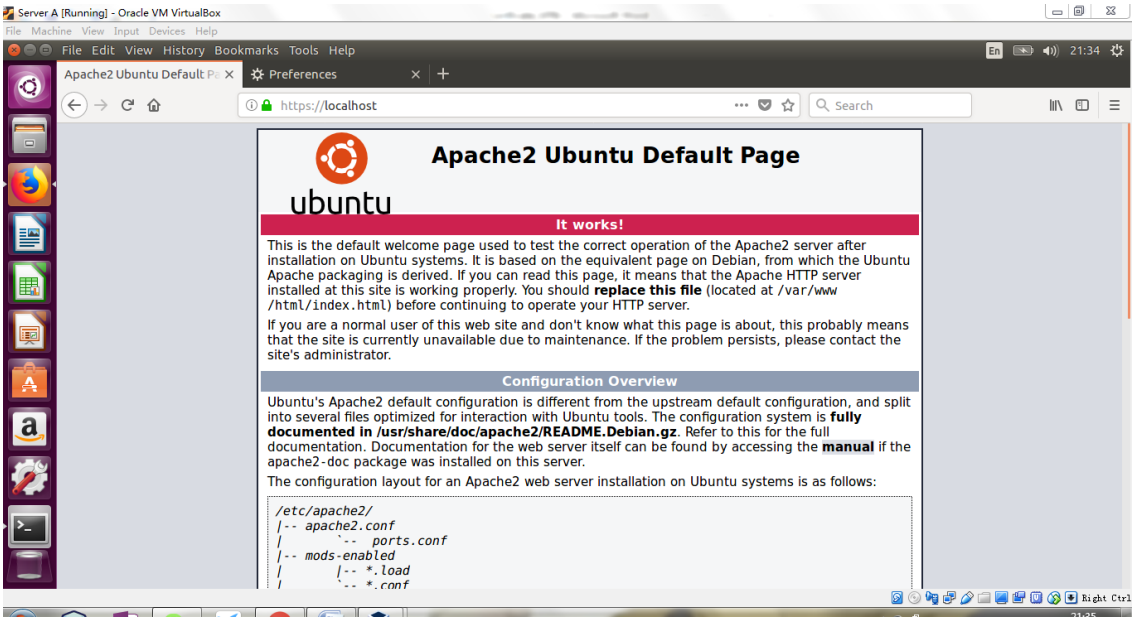
Task12:Show your certificate in Firefox

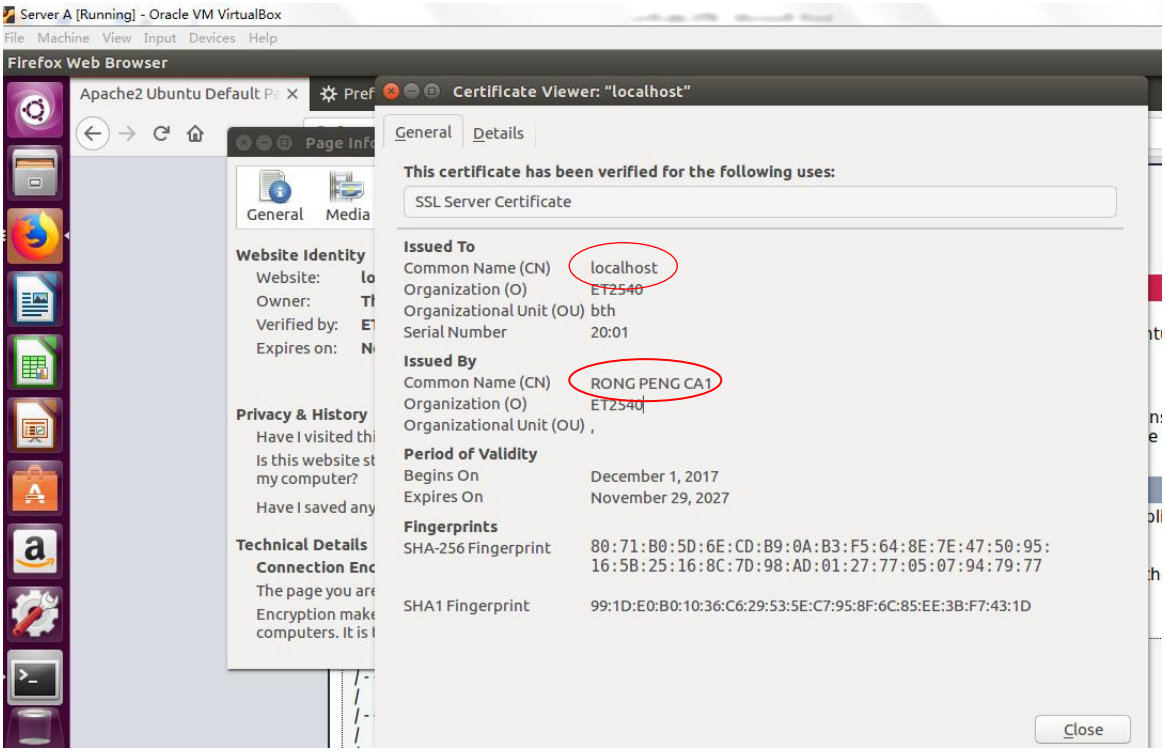
```
ats@serverA: ~
ats@serverA:~$ cp /home/ats/RP_ca/ca1/private/server.key.pem /etc/ssl/private/
cp: cannot stat '/etc/ssl/private/server.key.pem': Permission denied
ats@serverA:~$ sudo cp /home/ats/RP_ca/ca1/private/server.key.pem /etc/ssl/priva
te/
[sudo] password for ats:
ats@serverA:~$ sudo cp /home/ats/RP_ca/ca1/certs/server.cert.pem /etc/ssl/certs/
ats@serverA:~$ sudo gedit /etc/apache2/sites-enabled/default-ssl.conf

(gedit:3847): Gtk-WARNING **: Calling Inhibit failed: GDBus.Error:org.freedeskto
p.DBus.Error.ServiceUnknown: The name org.gnome.SessionManager was not provided
by any .service files

** (gedit:3847): WARNING **: Set document metadata failed: Setting attribute met
adata::gedit-spell-enabled not supported

** (gedit:3847): WARNING **: Set document metadata failed: Setting attribute met
adata::gedit-encoding not supported

** (gedit:3847): WARNING **: Set document metadata failed: Setting attribute met
adata::gedit-position not supported
ats@serverA:~$ sudo service apache2 restart
ats@serverA:~$
```

```
SSLCertificateFile    /home/ats/RP_ca/ca1/certs/server.cert.pem
SSLCertificateKeyFile /home/ats/RP_ca/ca1/private/server.key.pem
SSLCertificateChainFile /home/ats/RP_ca/ca1/certs/rcs.cert-chain.pem
```


Server A [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Insecure Connection - Mozilla Firefox
Insecure Connection
https://localhost
Search
Your connection is not secure
The owner of localhost has configured their website improperly. To protect your information from being stolen, Firefox has not connected to this website.
Learn more...
Report errors like this to help Mozilla identify and block malicious sites
Go Back
Advanced


Server A [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Firefox Web Browser
Insecure Connection
Preferences
Firefox
about:preferences#searchResults
Search
Certificate Manager
Your Certificates
You have certificates on
Certificate Name
AC Camerfirma S.A.
Chambers of Commer
Global Chambersign F
AC Camerfirma SA CIF A
Camerfirma Chamber
Camerfirma Global Ch
ACCV
Downloading Certificate
You have been asked to trust a new Certificate Authority (CA).
Do you want to trust "RONG PENG ROOT" for the following purposes?
Trust this CA to identify websites.
Trust this CA to identify email users.
Trust this CA to identify software developers.
Before trusting this CA for any purpose, you should examine its certificate and its policy and procedures (if available).
View
Examine CA certificate
Cancel
OK
View...
Edit Trust...
Import...
Export...
Delete or Distrust...





Now this page is using my cert-chain .

Task13:Create a CRL for CA1

CRL: certification revocation list.

To Enable CRL Distribution Lists extension, I edit the ca1 openssl.conf ,add the crlDistributionPoints.

```
[ server_cert ]

basicConstraints =CA:FALSE
subjectKeyIdentifier= hash
authorityKeyIdentifier =keyid,issuer:always
keyUsage= critical, digitalSignature,keyEncipherment
extendedKeyUsage =serverAuth
crlDistributionPoints= URI:https://localhost/ca1.crl.pem
```

Create a CRL for CA1:

```
ats@serverA:~/RP_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/ats/RP_ca/ca1/private/ca1.key.pem:
```

```
ats@serverA:~/RP_ca$ openssl crl -in ca1/crl/ca1.crl.pem -noout -text
Certificate Revocation List (CRL):
        Version 2 (0x1)
    Signature Algorithm: sha256WithRSAEncryption
        Issuer: /C=SE/ST=Blekinge/O=ET2540/OU=,/CN=RONG PENG CA1/emailAddress=921411446@qq.com
        Last Update: Dec  1 21:27:57 2017 GMT
        Next Update: Dec 31 21:27:57 2017 GMT
        CRL extensions:
            X509v3 CRL Number:
                8192
No Revoked Certificates.
    Signature Algorithm: sha256WithRSAEncryption
         88:27:20:93:4d:8a:0d:20:0c:08:53:61:04:42:06:e2:9d:06:
         8c:16:c0:09:ce:53:42:63:69:ef:bd:63:5c:48:03:13:34:a1:
         53:44:cf:25:04:0c:88:84:1d:29:c4:56:6e:36:04:38:bb:21:
         cc:cf:a6:7a:01:45:91:76:36:e0:f4:bf:75:18:09:97:fc:75:
         4b:8a:30:3f:d7:3f:43:cd:75:0e:e8:6c:a9:25:4d:01:d5:46:
         72:73:51:b6:2a:b1:7e:9c:0a:c4:b2:21:b9:fc:0b:34:94:32:
         6e:ac:63:db:d1:f3:08:7c:df:cb:2d:19:50:6e:ec:04:0b:be:
         bb:72:58:8e:00:69:02:78:fa:4e:ca:b4:71:b4:83:5b:c3:0d:
         13:c7:56:a2:08:b3:80:47:fe:62:7f:7d:cd:da:34:e8:bd:ac:
         08:d2:a4:75:9b:b1:a7:bc:76:a3:ca:91:3b:d0:53:a7:d0:26:
         2d:c5:9e:73:3e:3e:40:d6:10:e9:52:b0:c6:8b:36:7f:ec:7d:
         82:cc:28:5e:71:d5:b8:d0:fe:f2:73:da:b7:d5:0e:aa:a8:4e:
         5f:ac:38:96:0f:bc:ec:92:77:43:8d:19:8b:31:fe:ae:ca:b5:
         c7:44:bb:63:2b:8a:e6:ab:f6:a1:d1:b0:9d:e7:12:bc:7b:88:
         72:f0:0f:81:8a:35:df:bc:47:af:7b:28:67:37:c4:44:1a:ac:
         c6:3c:28:ed:23:3b:c4:76:a1:c5:41:ec:c2:38:bd:e5:9f:99:
         e2:2f:09:ae:a9:7c:41:3f:c1:bc:cc:f5:eb:58:86:f4:2b:e2:
         21:6d:09:b7:66:87:22:ce:0e:a9:1c:6e:66:3d:cc:da:0e:a9:
         ef:39:7e:f5:12:22:10:42:22:09:f1:35:c2:13:13:07:76:85:
         0e:8f:c8:28:e7:08:6c:37:d6:2c:6a:1f:b4:da:db:ed:df:9c:
         2c:5c:08:2f:87:16:b7:49:23:c0:51:b7:fa:b9:e9:c8:fa:97:
         7e:b3:9d:ab:f6:40:5e:4f:84:03:aa:17:52:f7:a1:6e:be:53:
         45:c1:39:db:d2:86:85:07:be:ea:ee:a3:df:ae:7d:58:fa:74:
         fe:78:8a:19:f7:c1:cf:95:70:96:2c:2c:97:65:13:f0:a7:f9:
```

Task14:Revoke a certificate

Create a user certificate for dragos.ilie@bth.se, verify the usr certification , then revoke this certificate.

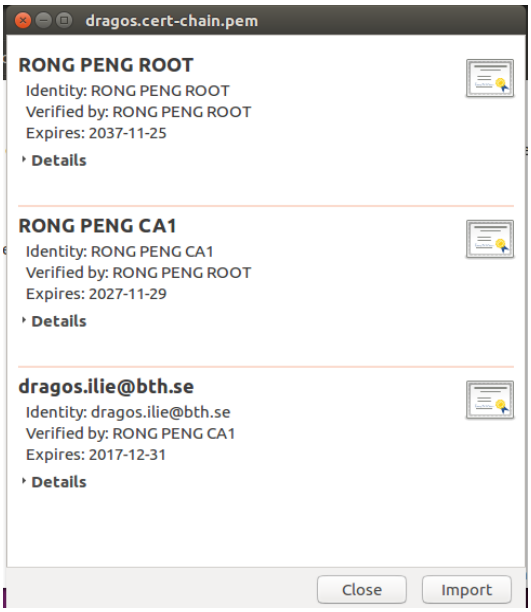
```
ats@serverA:~/RP_ca/ca1$ openssl genrsa -out private/usr.key.pem 2048
Generating RSA private key, 2048 bit long modulus
.....................................+++
......+++
e is 65537 (0x10001)
ats@serverA:~/RP_ca/ca1$ openssl req -config openssl.cnf -new -key private/usr.key.pem -out csr/usr.csr.pem
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.
-----
SE (2 letter code) [SE]:SE
Blekinge [Blekinge]:Blekinge
Karlskrona []:Karlskrona
Organization Name (eg, company) [ET2540]:ET2540
Organizational Unit Name (eg, section) []:,
Common Name (e.g. server FQDN or YOUR name) []:dragos.ilie@bth.se
Email Address []:dragos.ilie@bth.se

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:atslabb00
An optional company name []:bth
ats@serverA:~/RP_ca/ca1$ openssl ca -config openssl.cnf -extensions usr_cert -days 30 -notext -md sha256 -in csr/usr.csr.pem -out certs/usr.ce
rt.pem
Using configuration from openssl.cnf
Enter pass phrase for /home/ats/RP_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  1 22:57:29 2017 GMT
            Not After : Dec 31 22:57:29 2017 GMT
        Subject:
```

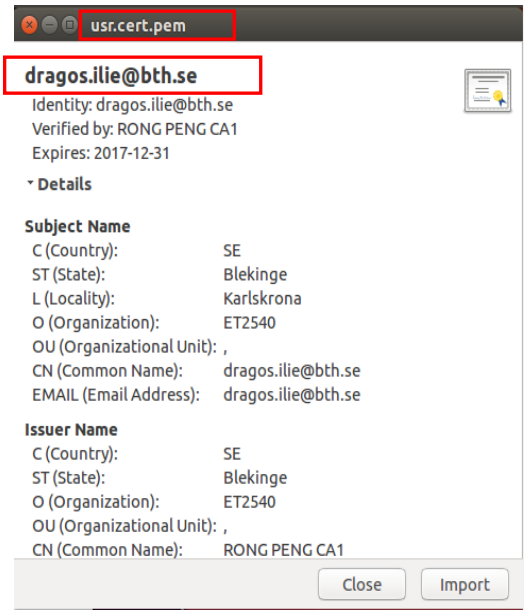
```
Certificate Details:
        Serial Number: 8194 (0x2002)
        Validity
            Not Before: Dec  1 22:57:29 2017 GMT
            Not After : Dec 31 22:57:29 2017 GMT
        Subject:
            countryName               = SE
            stateOrProvinceName       = Blekinge
            localityName              = Karlskrona
            organizationName          = ET2540
            organizationalUnitName    = ,
            commonName                = dragos.ilie@bth.se
            emailAddress              = dragos.ilie@bth.se
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            X509v3 Key Usage: critical
                Digital Signature, Non Repudiation, Key Encipherment
            Netscape Comment:
                OpenSSL Generated Certificate
            X509v3 Subject Key Identifier:
                6F:9A:CB:15:39:E2:F2:07:3F:8A:02:0E:E9:7D:5C:D3:7C:7D:3B:64
            X509v3 Authority Key Identifier:
                keyid:54:86:E6:F0:8E:6F:88:02:1B:75:E6:2C:01:D3:BE:1A:D3:99:B5:8D

            X509v3 Extended Key Usage:
                TLS Web Client Authentication, E-mail Protection
Certificate is to be certified until Dec 31 22:57:29 2017 GMT (30 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
```

create the cert-chain, then verify:

```
ats@serverA:~/RP_ca$ cat certs/root.cert.pem ca1/certs/ca1.cert.pem ca1/certs/us
r.cert.pem > ca1/certs/dragos.cert-chain.pem
ats@serverA:~/RP_ca$
```

```
ats@serverA:~/RP_ca$ openssl verify -CAfile ca1/certs/dragos.cert-chain.pem ca1/
certs/usr.cert.pem
ca1/certs/usr.cert.pem: OK
ats@serverA:~/RP_ca$
```





Revoke the usr.cert.pem (CN:dragos.ilie@bth.se):

```
ats@serverA: ~/RP_ca/ca1
ats@serverA:~/RP_ca/ca1$ openssl ca -config openssl.cnf -revoke certs/usr.cert.p
em
Using configuration from openssl.cnf
Enter pass phrase for /home/ats/RP_ca/ca1/private/ca1.key.pem:
Revoking Certificate 2002.
Data Base Updated
ats@serverA:~/RP_ca/ca1$
```

The following content was displaying in the index.txt:

```
V       211291160939Z           2000    unknown /C=SE/ST=Blekinge/L=Karlskrona/O=ET2540/OU=,/CN=localhost/emailAddress=921411446@qq.com
V       211291163855Z           2001    unknown /C=SE/ST=Blekinge/L=Karlskrona/O=ET2540/OU=bth/CN=localhost/emailAddress=921411446@qq.com
R       171231225729Z   171201231206Z   2002    unknown /C=SE/ST=Blekinge/L=Karlskrona/O=ET2540/OU=,/CN=dragos.ilie@bth.se/
emailAddress=dragos.ilie@bth.se
```

Recreate the CRL:

```
ats@serverA:~/RP_ca/ca1$ openssl ca -config openssl.cnf -gencrl -out crl/ca1.crl
.pem
Using configuration from openssl.cnf
Enter pass phrase for /home/ats/RP_ca/ca1/private/ca1.key.pem:
```

```
ats@serverA:~/RP_ca/ca1$ openssl crl -in crl/ca1.crl.pem -noout -text
Certificate Revocation List (CRL):
        Version 2 (0x1)
    Signature Algorithm: sha256WithRSAEncryption
        Issuer: /C=SE/ST=Blekinge/O=ET2540/OU=,/CN=RONG PENG CA1/emailAddress=921411446@qq.com
        Last Update: Dec  1 23:17:16 2017 GMT
        Next Update: Dec 31 23:17:16 2017 GMT
        CRL extensions:
            X509v3 CRL Number:
                8193
Revoked Certificates:
    Serial Number: 2002
        Revocation Date: Dec  1 23:12:06 2017 GMT
    Signature Algorithm: sha256WithRSAEncryption
         24:e3:ed:1b:35:c2:85:c7:a2:0b:15:4e:63:1f:9d:d2:bc:3b:
         ab:e2:ab:23:82:db:40:61:a6:cf:2f:8b:c4:4c:1a:05:1f:be:
         43:8e:ed:68:fb:33:1c:86:f0:38:e9:0c:6d:06:3a:d4:1b:6f:
         8b:b8:88:88:18:9a:b3:6d:70:7b:a3:a0:a9:6a:58:e5:2f:f3:
         39:6a:64:14:56:c9:65:d0:85:73:d5:bc:68:92:41:ed:7e:06:
         e8:7f:05:dc:f7:48:6e:d7:24:08:2b:5d:2c:40:c6:45:e4:2a:
         d2:1a:ae:bc:aa:9c:2b:53:09:99:9d:2c:51:56:da:0a:a0:a2:
         5b:9d:3e:2e:48:6c:4b:86:a5:b2:89:12:80:c4:92:b2:8b:ab:
         41:32:73:0a:0e:2e:de:42:50:53:07:eb:00:e2:75:98:1e:f6:
         c8:0e:e1:06:ce:cd:c5:4e:44:0d:e6:da:47:b2:f5:e3:90:ef:
         9f:44:d9:92:72:9b:4b:56:c4:58:b6:f3:6a:80:c7:8a:2a:41:
```

Task15:Host-to-host transport mode VPN with PSK authentication

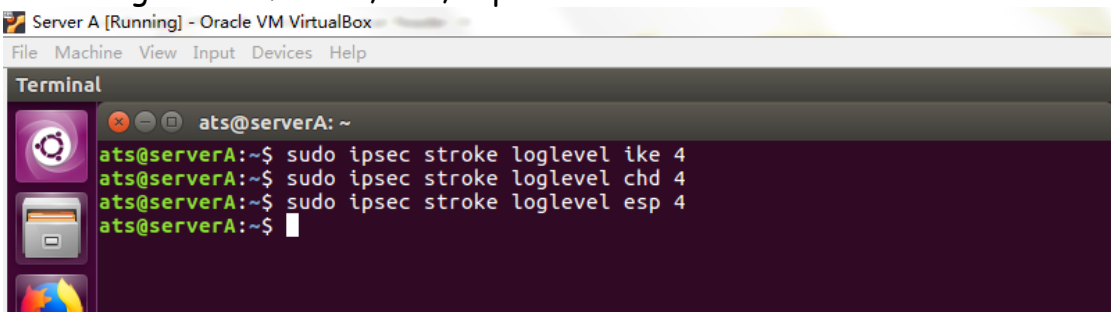
There happens something like dependencies in my serverA, so firstly I using *sudo apt-get –f install* to install some dependencies on serverA.

```
s@serverA: ~
Reading state information... Done
You might want to run 'apt-get -f install' to correct these:
The following packages have unmet dependencies:
 strongswan : Depends: strongswan-charon but it is not going to be installed
              Depends: strongswan-starter but it is not going to be installed
 thunderbird-gnome-support : Depends: thunderbird (= 1:52.4.0+build1-0ubuntu0.16
E: Unmet dependencies. Try 'apt-get -f install' with no packages (or specify a s
ats@serverA:~$ sudo apt-get -f install
Reading package lists... Done
Building dependency tree
Reading state information... Done
Correcting dependencies... Done
The following packages were automatically installed and are no longer required:
  linux-headers-4.4.0-31 linux-headers-4.4.0-31-generic linux-headers-4.4.0-47
  linux-headers-4.4.0-47-generic linux-image-4.4.0-31-generic
  linux-image-4.4.0-47-generic linux-image-extra-4.4.0-31-generic
```

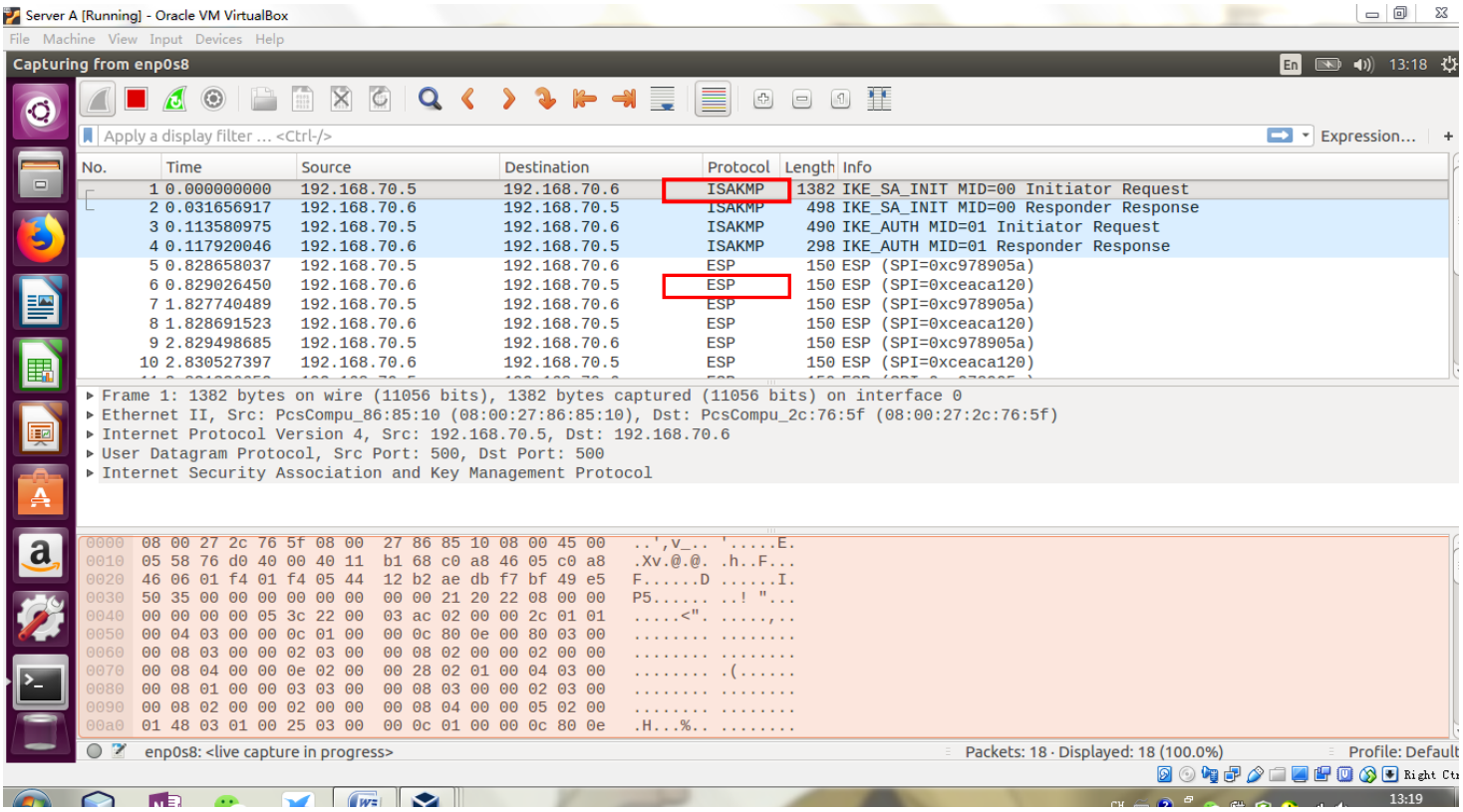
Later, I began to install strongSwan.

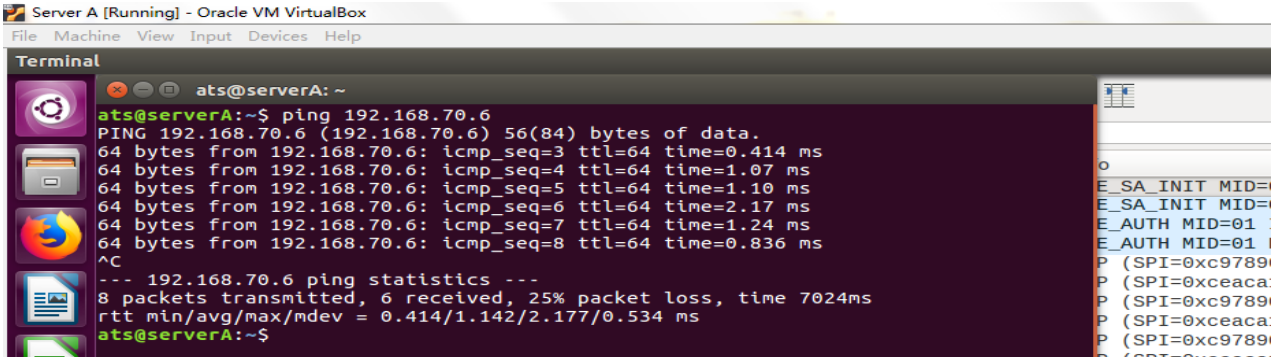
```
ats@serverA:~$ sudo apt-get install strongswan
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  linux-headers-4.4.0-31 linux-headers-4.4.0-31-generic linux-headers-4.4.0-47 linux-headers-4.4.0-47-generic linux-image-4.4.0-31-generic
  linux-image-4.4.0-47-generic linux-image-extra-4.4.0-31-generic linux-image-extra-4.4.0-47-generic
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  libstrongswan libstrongswan-standard-plugins strongswan-charon strongswan-libcharon strongswan-starter
Suggested packages:
  libstrongswan-extra-plugins libcharon-extra-plugins
The following NEW packages will be installed:
  libstrongswan libstrongswan-standard-plugins strongswan strongswan-charon strongswan-libcharon strongswan-starter
0 upgraded, 6 newly installed, 0 to remove and 280 not upgraded.
Need to get 3 731 kB of archives.
After this operation, 16,1 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://se.archive.ubuntu.com/ubuntu xenial-updates/main amd64 libstrongswan amd64 5.3.5-1ubuntu3.4 [1 398 kB]
Get:2 http://se.archive.ubuntu.com/ubuntu xenial-updates/main amd64 strongswan-libcharon amd64 5.3.5-1ubuntu3.4 [1 241 kB]
Get:3 http://se.archive.ubuntu.com/ubuntu xenial-updates/main amd64 strongswan-starter amd64 5.3.5-1ubuntu3.4 [742 kB]
Get:4 http://se.archive.ubuntu.com/ubuntu xenial-updates/main amd64 strongswan-charon amd64 5.3.5-1ubuntu3.4 [55,6 kB]
Get:5 http://se.archive.ubuntu.com/ubuntu xenial-updates/main amd64 libstrongswan-standard-plugins amd64 5.3.5-1ubuntu3.4 [267 kB]
Get:6 http://se.archive.ubuntu.com/ubuntu xenial-updates/main amd64 strongswan all 5.3.5-1ubuntu3.4 [27,1 kB]
Fetched 3 731 kB in 3s (1 121 kB/s)
```

Enable log level 4 for ike, chd, esp



At first time, I configured my ipsec.conf and ipsec.secrets well, but also cannot work, I also discussed with teacher, we have checked the /etc/network/interfaces, and restarted network-manager, networking services but still no works, stucking in ping each .I went back home and thinking about where the problem lies in, finally I cleared out ,because I set several host-only network cards, then serverA cannot recognize which one is for the 192.168.70.5. The following screenshot is about after configuring well of the network interfaces and ping successfully then checking in the wireshark:





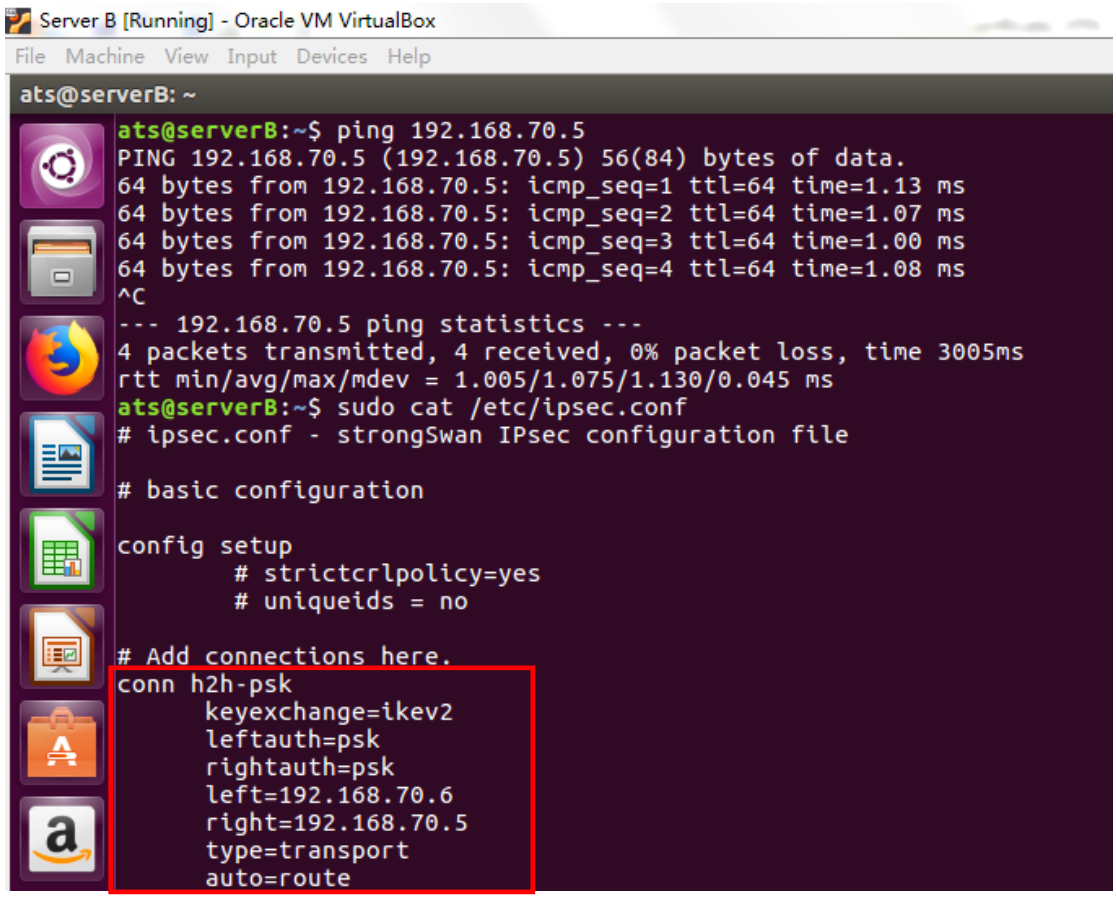
ISAKMP: Internet Security Association Key Management Protocol.
ESP: Encapsulate Security Payload.
After the key managing then beginning the payload exchange.
And I check the ipsec statusall.

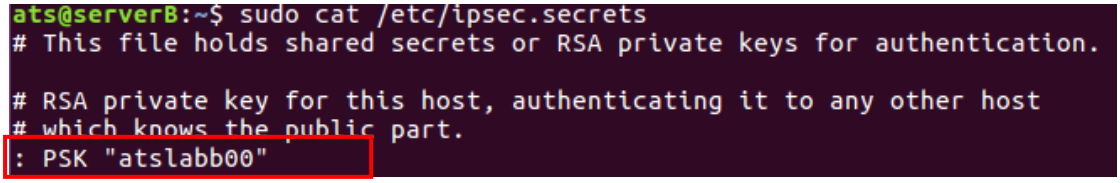
```
ats@serverA:~$ sudo ipsec statusall
[sudo] password for ats:
Status of IKE charon daemon (strongSwan 5.3.5, Linux 4.4.0-103-generic, x86_64):
  uptime: 7 minutes, since Dec 09 13:14:28 2017
  malloc: sbrk 1486848, mmap 0, used 349392, free 1137456
  worker threads: 11 of 16 idle, 5/0/0/0 working, job queue: 0/0/0/0, scheduled: 3
  loaded plugins: charon test-vectors aes rc2 sha1 sha2 md4 md5 random nonce x509 revocation constraints pubkey pkcs1 pkcs7 pkcs8 pkcs12 p
nskey sshkey pem openssl fips-prf gmp agent xcbc hmac ccm gcm attr kernel-netlink resolve socket-default connmark stroke updown
Listening IP addresses:
  192.168.60.100
  192.168.70.5
  10.0.98.100
Connections:
     h2h-psk:  192.168.70.5...192.168.70.6  IKEv2
     h2h-psk:   local:  [192.168.70.5] uses pre-shared key authentication
     h2h-psk:   remote: [192.168.70.6] uses pre-shared key authentication
     h2h-psk:   child:  dynamic === dynamic TRANSPORT
Routed Connections:
     h2h-psk{1}:  ROUTED, TRANSPORT, reqid 1
     h2h-psk{1}:   192.168.70.5/32 === 192.168.70.6/32
Security Associations (1 up, 0 connecting):
     h2h-psk[1]: ESTABLISHED 3 minutes ago, 192.168.70.5[192.168.70.5]...192.168.70.6[192.168.70.6]
     h2h-psk[1]: IKEv2 SPIs: 3550e549bff7dbae_i* e63614ae9cc1c8f7_r, pre-shared key reauthentication in 2 hours
     h2h-psk[1]: IKE proposal: AES_CBC_128/HMAC_SHA1_96/PRF_HMAC_SHA1/MODP_2048
     h2h-psk{2}:  INSTALLED, TRANSPORT, reqid 1, ESP SPIs: ceaca120_i c978905a_o
     h2h-psk{2}:  AES_CBC_128/HMAC_SHA1_96, 384 bytes_i (6 pkts, 209s ago), 384 bytes_o (6 pkts, 209s ago), rekeying in 39 minutes
     h2h-psk{2}:   192.168.70.5/32 === 192.168.70.6/32
ats@serverA:~$
```

The following screenshot is how I configured ipsec.conf and ipsec.secrets. In ipsec.conf I set the internet keys exchange using IKEV2 format and their authentication is using pre-shared key, let both side know each public network address, then the type is transport mode, auto is set to route.

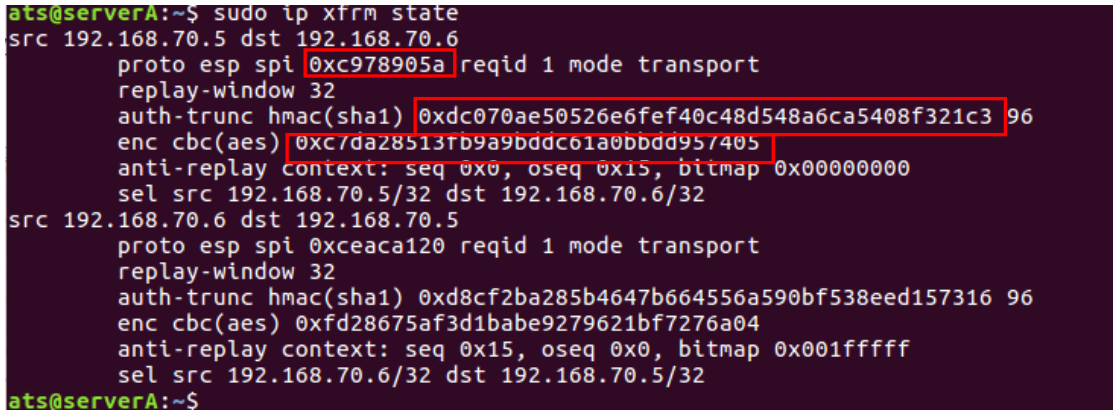
```
ats@serverA:~$ sudo cat /etc/ipsec.conf
# ipsec.conf - strongSwan IPsec configuration file

# basic configuration

config setup
        # strictcrlpolicy=yes
        # uniqueids = no

# Add connections here.
conn h2h-psk
      keyexchange=ikev2
      leftauth=psk
      rightauth=psk
      left=192.168.70.5
      right=192.168.70.6
      type=transport
      auto=route
```

```
ats@serverA:~$ sudo cat /etc/ipsec.secrets
# 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.
: PSK "atslabb00"
```



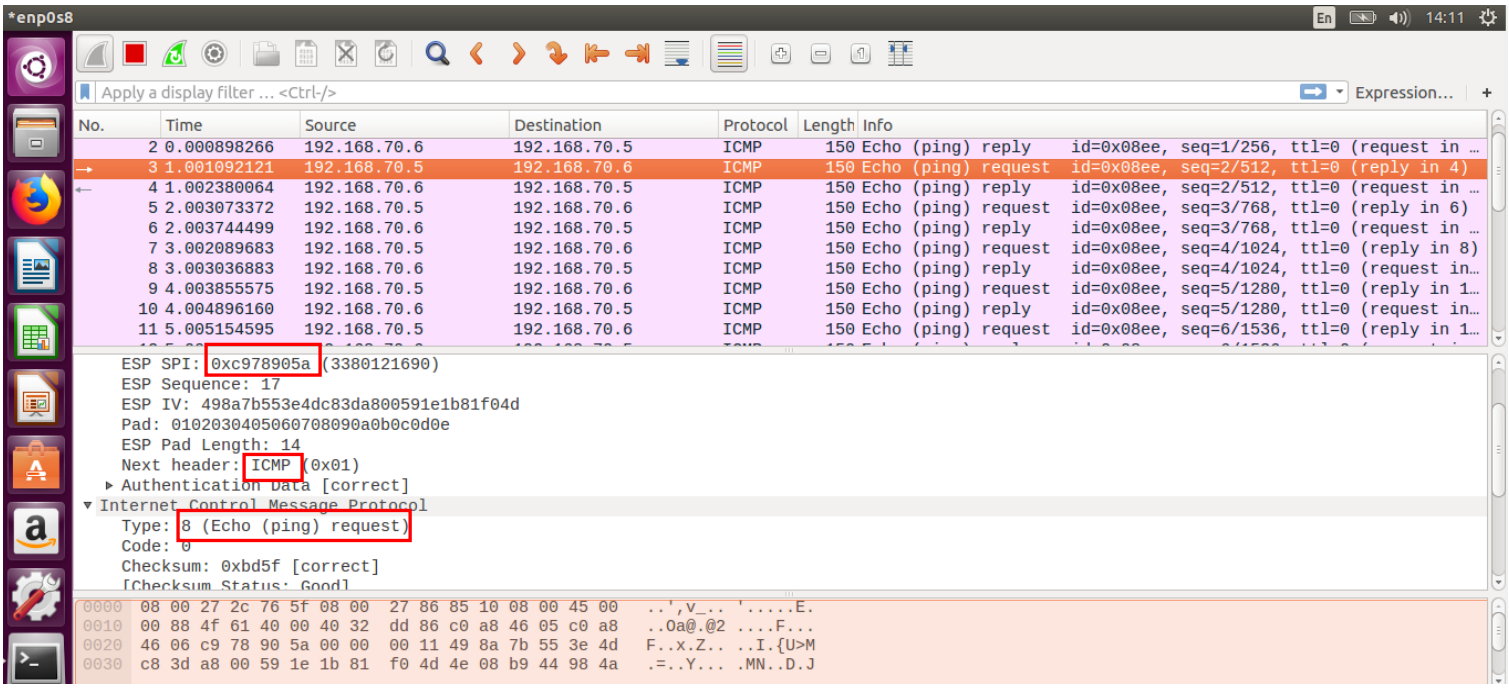


Task16:Decrypt traffic with Wireshark



ESP SAs

Protocol	Src IP	Dest IP	SPI	Encryption	Encryption Key	Authentication	Authentication Key
IPv4	192.168.70.5	192.168.70.6	0xc978905a	AES-CBC [RFC3602]	0xc7da28513fb9a9bddc61a0bbdd957405	HMAC-SHA-1-96 [RFC2404]	0xdc070ae50526e6fef40c48d548a6ca54...
IPv4	192.168.70.6	192.168.70.5	0xceaca120	AES-CBC [RFC3602]	0xfd28675af3d1babe9279621bf7276a04	HMAC-SHA-1-96 [RFC2404]	0xd8cf2ba285b4647b664556a590bf538...



Task17:List the entries in the SPD

```
ats@serverA:~$ sudo ip xfrm policy
src 192.168.70.6/32 dst 192.168.70.5/32
        dir in priority 2819
        tmpl src 0.0.0.0 dst 0.0.0.0
                proto esp reqid 1 mode transport
src 192.168.70.5/32 dst 192.168.70.6/32
        dir out priority 2819
        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
```

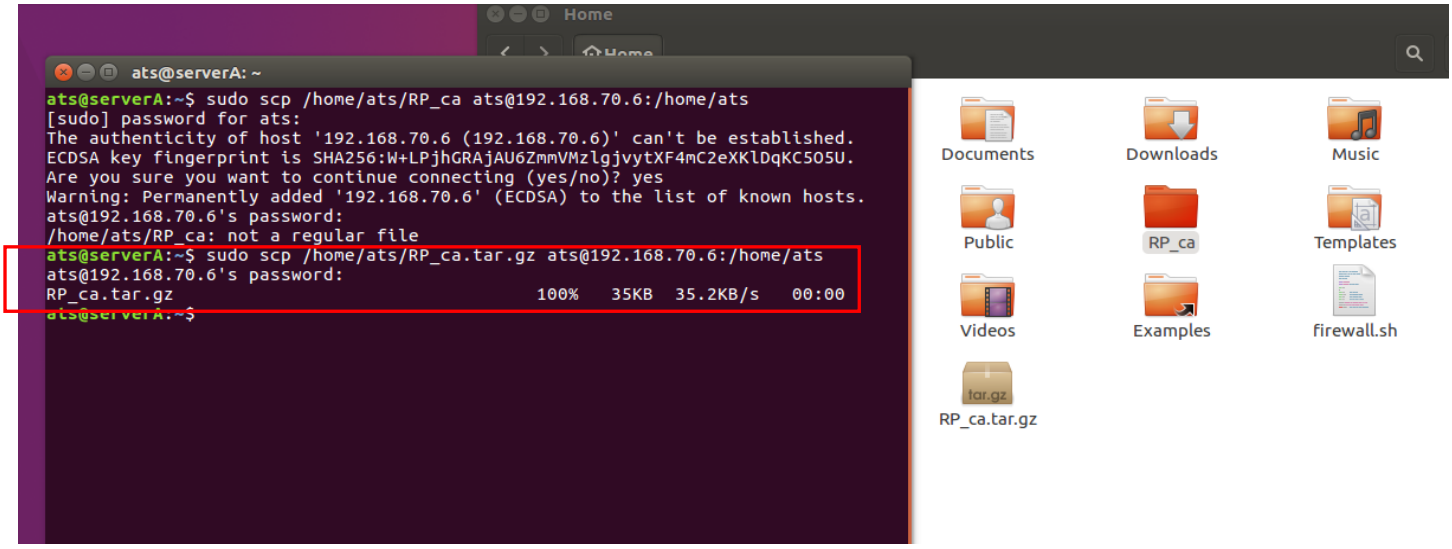
The major two important security databases:
SPD: Security Policy Database
SAD: Security Association Database
The policy direction is *in*, *tmpl* is a template list specified reqid , mode and so on. The 192.168.70.5 's requestment ID (*reqid*)is 1, and its mode is *transport*, using protocol is IPsec *ESP*(Encapsulating Security Payload)

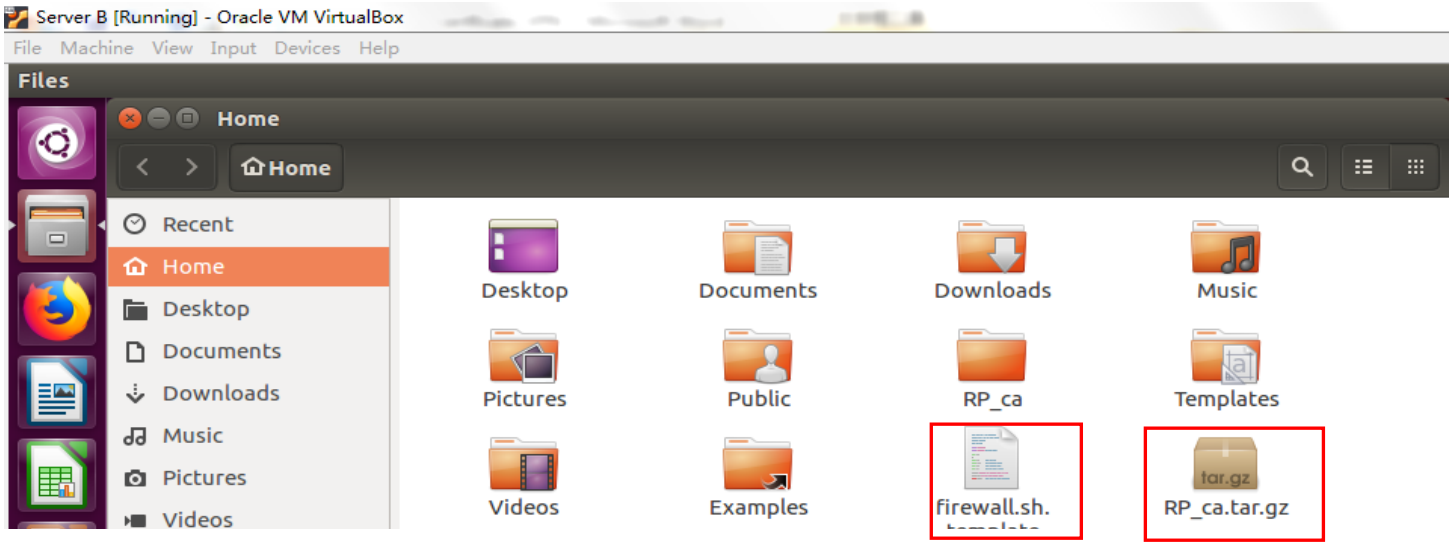
Task18:Host-to-host transport mode VPN with cert authentication

I have generated 192.168.70.5.cert.pem which done like Task11, then copied to the ipsec.d folder's corresponding places. And following I will use serverB to present the 192.168.70.6.cert.pem creation steps, which is quite like the 192.168.70.5.cert.pem generation.

```
ats@serverA:~$ sudo cp /home/ats/RP_ca/ca1/certs/192.168.70.5.cert.pem /etc/ipsec.d/certs
ats@serverA:~$ sudo cp /home/ats/RP_ca/ca1/private/192.168.70.5.key.pem /etc/ipsec.d/private
ats@serverA:~$ sudo cp /home/ats/RP_ca/certs/root.cert.pem /etc/ipsec.d/cacerts
ats@serverA:~$ sudo cp /home/ats/RP_ca/ca1/certs/ca1.cert.pem /etc/ipsec.d/cacer
ts
```

Remotely copying the certificates folder to serverB, then can modify part of the files for serverB and can save time.





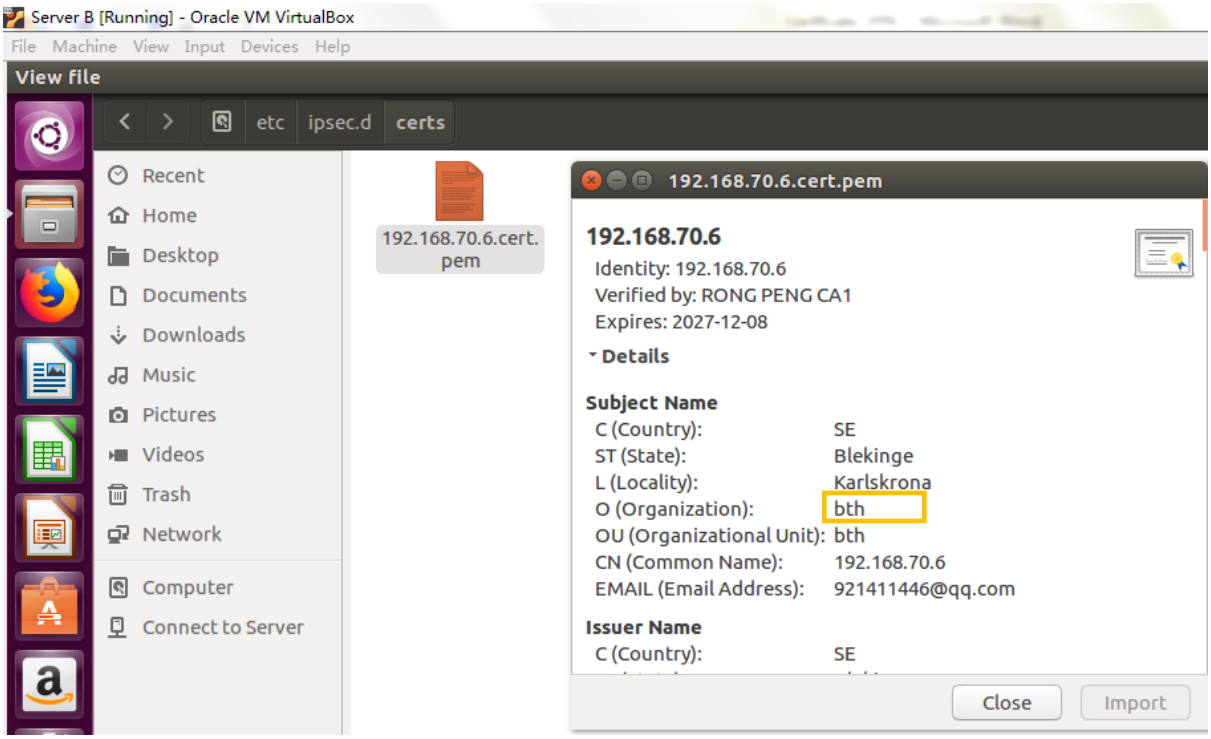
The following screenshots are about creating the 192.168.70.6.key.pem-> 192.168.70.6.csr.pem-> 192.168.70.6.cert.pem , then copy them into ipsec.d right places, and don't forget putting CA1 and root CA into cacerts folder.

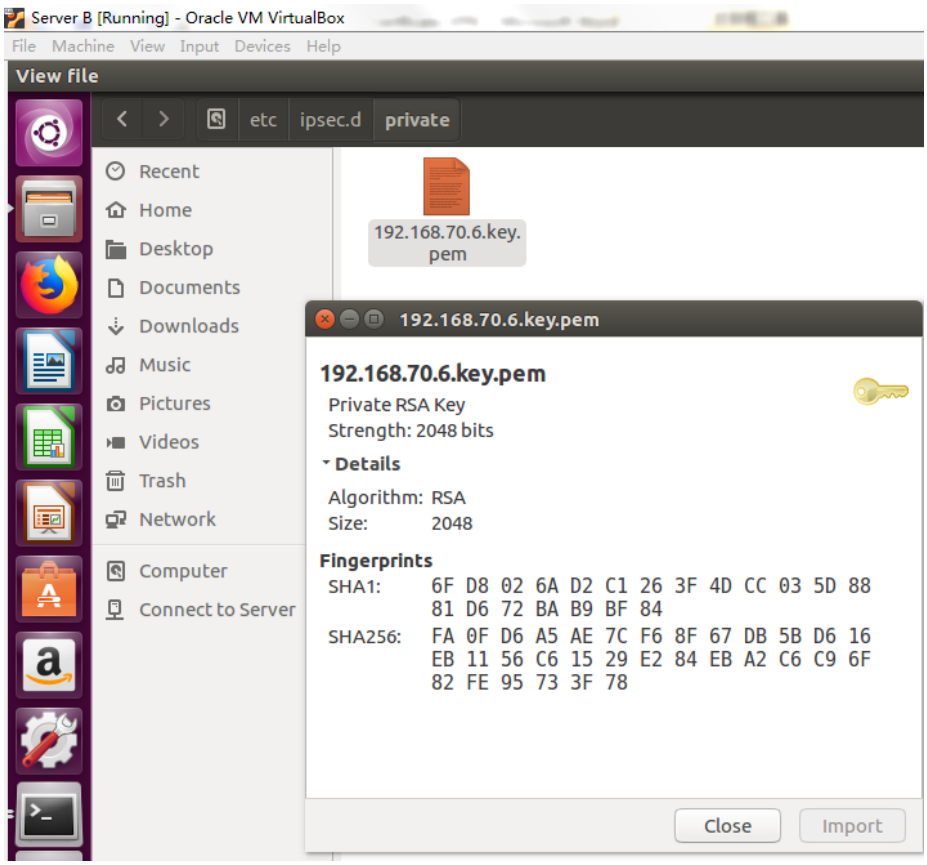
```
ats@serverB:~/RP_ca$ sudo openssl genrsa -out private/192.168.70.6.key.pem
[sudo] password for ats:
Generating RSA private key, 2048 bit long modulus
......+++
.............................+++
e is 65537 (0x10001)
```

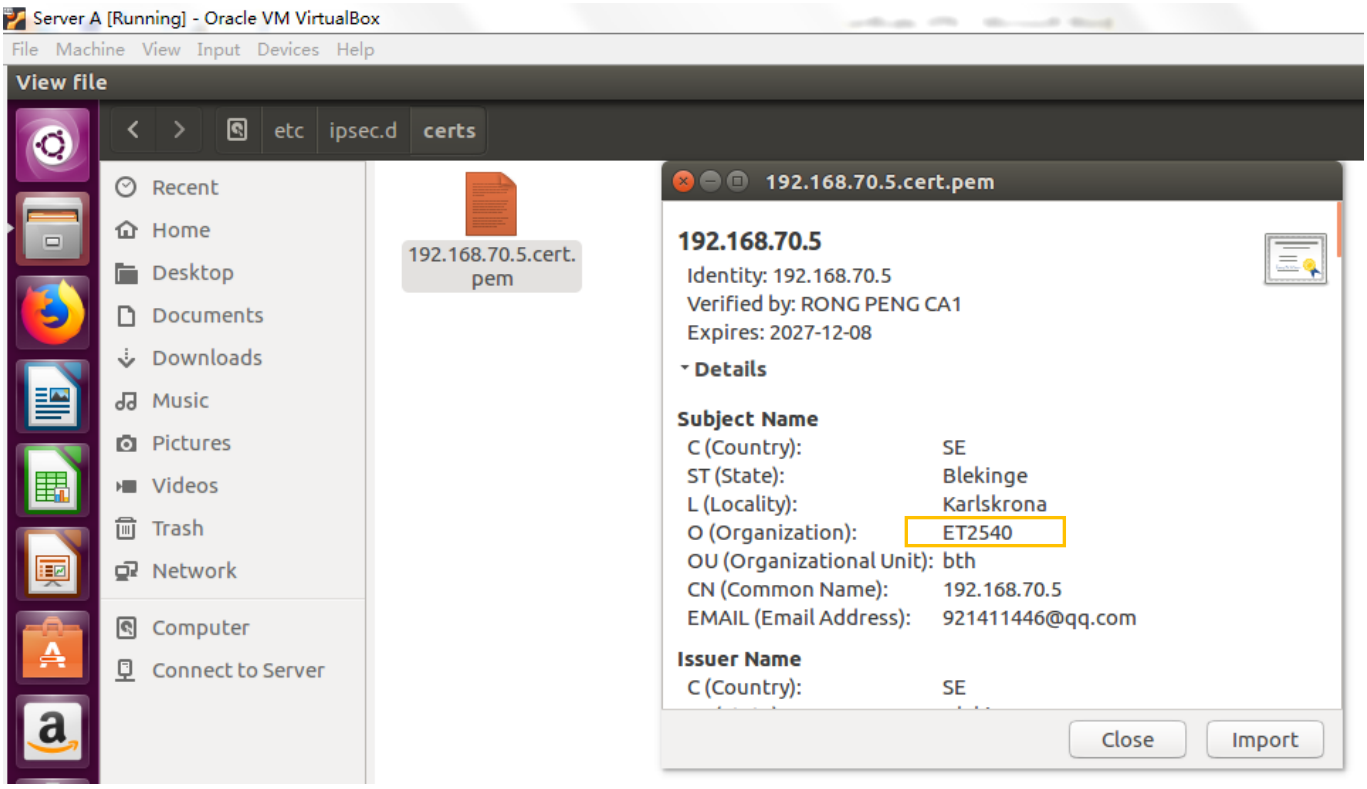
```
ats@serverB:~/RP_ca$ sudo openssl req -config openssl.cnf -new -sha256 -key ca1/private/192.168.70.6.key.pem -out ca1/csr/192.168.70.6.csr.pem

You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
SE (2 letter code) [SE]:SE
Blekinge [Blekinge]:Blekinge
Karlskrona []:Karlskrona
Organization Name (eg, company) [bth]:bth
Organizational Unit Name (eg, section) []:bth
Common Name (e.g. server FQDN or YOUR name) []:192.168.70.6
Email Address []:921411446@qq.com

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:atslabb00
An optional company name []:bth
```

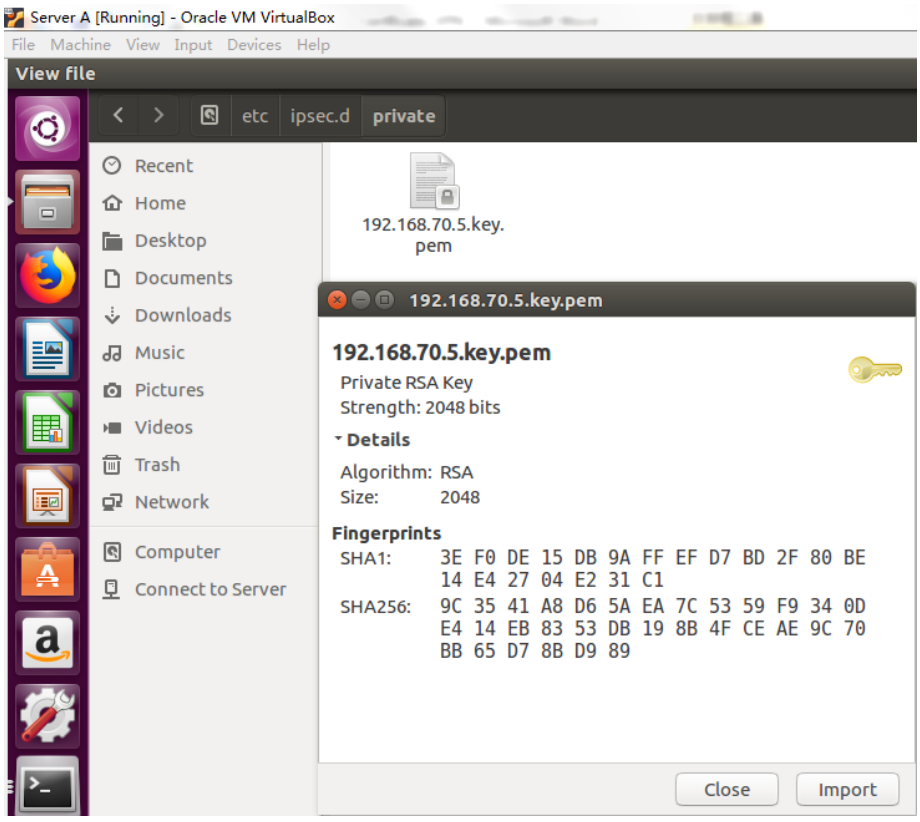
```
ats@serverB: ~/RP_ca/ca1                                                    En  15:54
ats@serverB:~/RP_ca/ca1$ sudo openssl ca -config openssl.cnf -extensions server_cert -days 3650 -notext -md sha256 -in csr/192.168.70.6.csr.pe
m -out certs/192.168.70.6.cert
[sudo] password for ats:
Using configuration from openssl.cnf
Enter pass phrase for /home/ats/RP_ca/ca1/private/ca1.key.pem:
Check that the request matches the signature
Signature ok
Certificate Details:
        Serial Number: 8197 (0x2005)
        Validity
            Not Before: Dec 10 14:25:39 2017 GMT
            Not After : Dec  8 14:25:39 2027 GMT
        Subject:
            countryName               = SE
            stateOrProvinceName       = Blekinge
            localityName              = Karlskrona
            organizationName          = bth
            organizationalUnitName    = bth
            commonName                = 192.168.70.6
            emailAddress              = 921411446@qq.com
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            X509v3 Subject Key Identifier:
                19:AB:58:63:88:B8:73:97:51:9D:F4:00:41:AB:56:EC:68:4A:E5:6E
            X509v3 Authority Key Identifier:
                keyid:54:86:E6:F0:8E:6F:88:02:1B:75:E6:2C:01:D3:BE:1A:D3:99:B5:8D
                DirName:/C=SE/ST=Blekinge/L=Karlskrona/O=ET2540/OU=BTH/CN=RONG PENG ROOT/emailAddress=921411446@qq.com
                serial:10:00

            X509v3 Key Usage: critical
                Digital Signature, Key Encipherment
            X509v3 Extended Key Usage:
```

Little differences for 192.168.70.6.cert.pem and 192.168.70.5.cert.pem, the organization in 192.168.70.6.cert.pem is bth, in 192.168.70.5.cert.pem is ET2540.






Server A [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
View file
etc ipsec.d certs
Recent
Home
Desktop
Documents
Downloads
Music
Pictures
Videos
Trash
Network
Computer
Connect to Server
192.168.70.5.cert.pem
192.168.70.5.cert.pem
192.168.70.5
Identity: 192.168.70.5
Verified by: RONG PENG CA1
Expires: 2027-12-08
Details
Subject Name
C (Country): SE
ST (State): Blekinge
L (Locality): Karlskrona
O (Organization): ET2540
OU (Organizational Unit): bth
CN (Common Name): 192.168.70.5
EMAIL (Email Address): 921411446@qq.com
Issuer Name
C (Country): SE
Close
Import

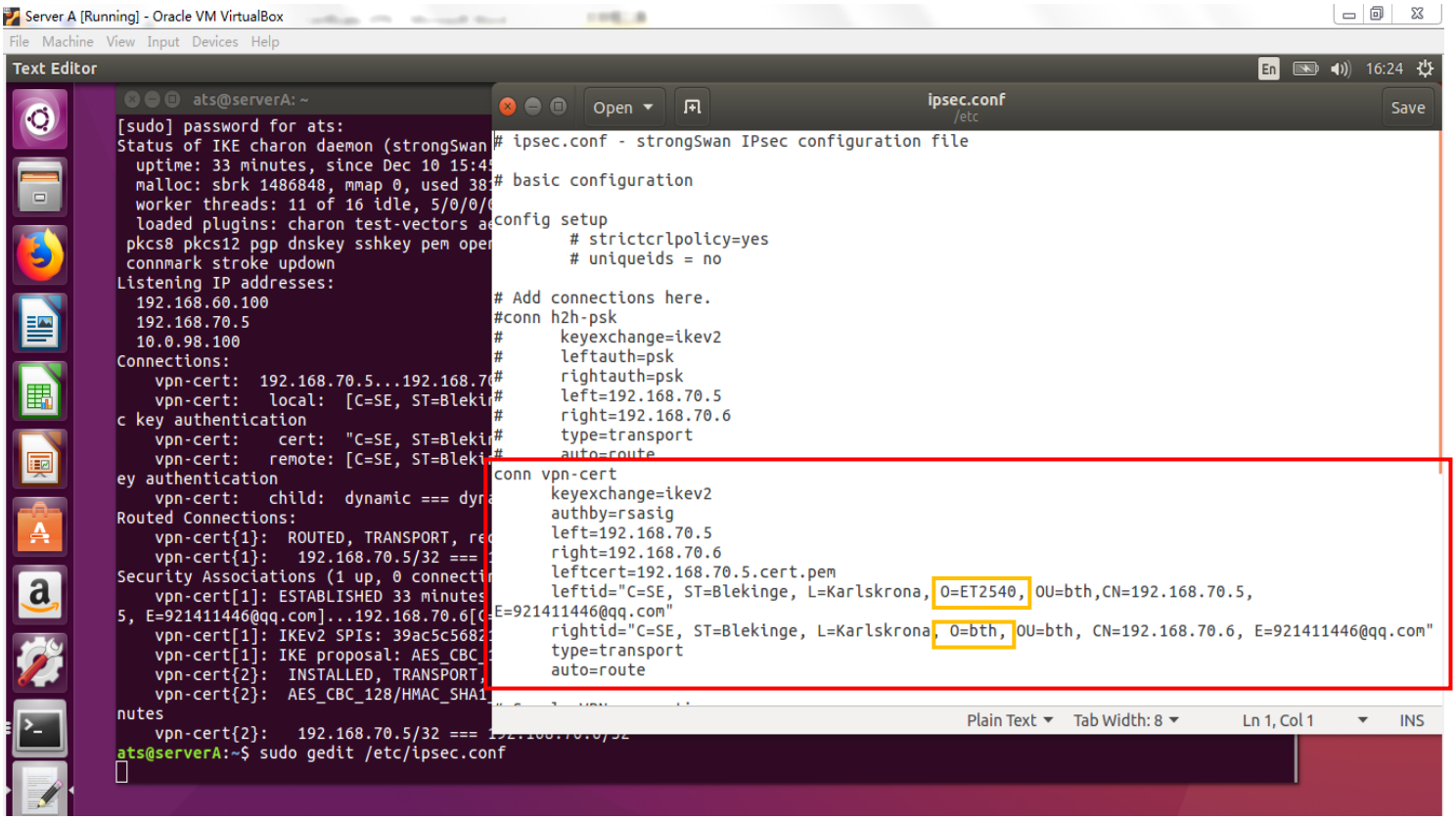

Server A [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
View file
etc ipsec.d private
Recent
Home
Desktop
Documents
Downloads
Music
Pictures
Videos
Trash
Network
Computer
Connect to Server
192.168.70.5.key.pem
192.168.70.5.key.pem
192.168.70.5.key.pem
Private RSA Key
Strength: 2048 bits
Details
Algorithm: RSA
Size: 2048
Fingerprints
SHA1: 3E F0 DE 15 DB 9A FF EF D7 BD 2F 80 BE 14 E4 27 04 E2 31 C1
SHA256: 9C 35 41 A8 D6 5A EA 7C 53 59 F9 34 0D E4 14 EB 83 53 DB 19 8B 4F CE AE 9C 70 BB 65 D7 8B D9 89
Close
Import

Checking the ROOT and CA1:

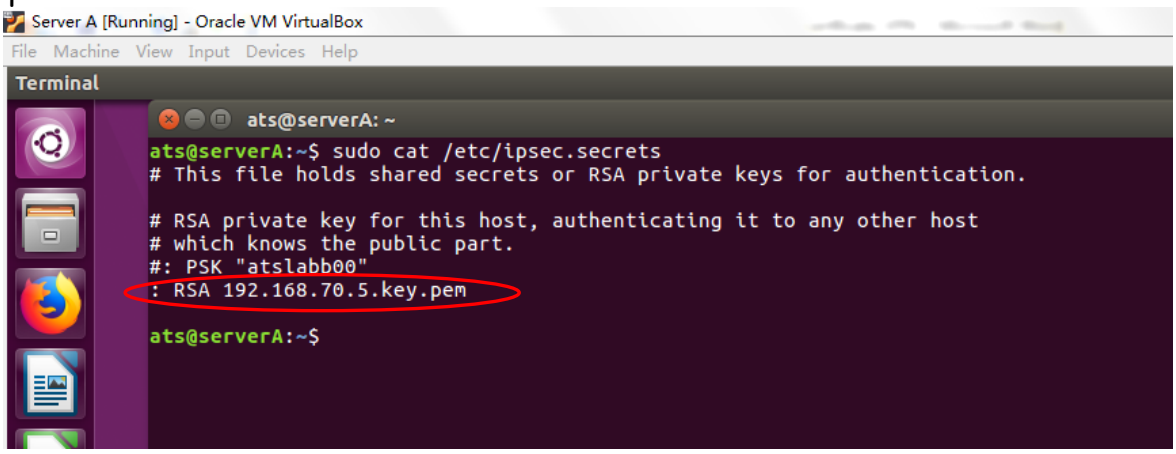
```
ats@serverA:~$ sudo ipsec rereadcacerts
ats@serverA:~$ sudo ipsec listcacerts
```

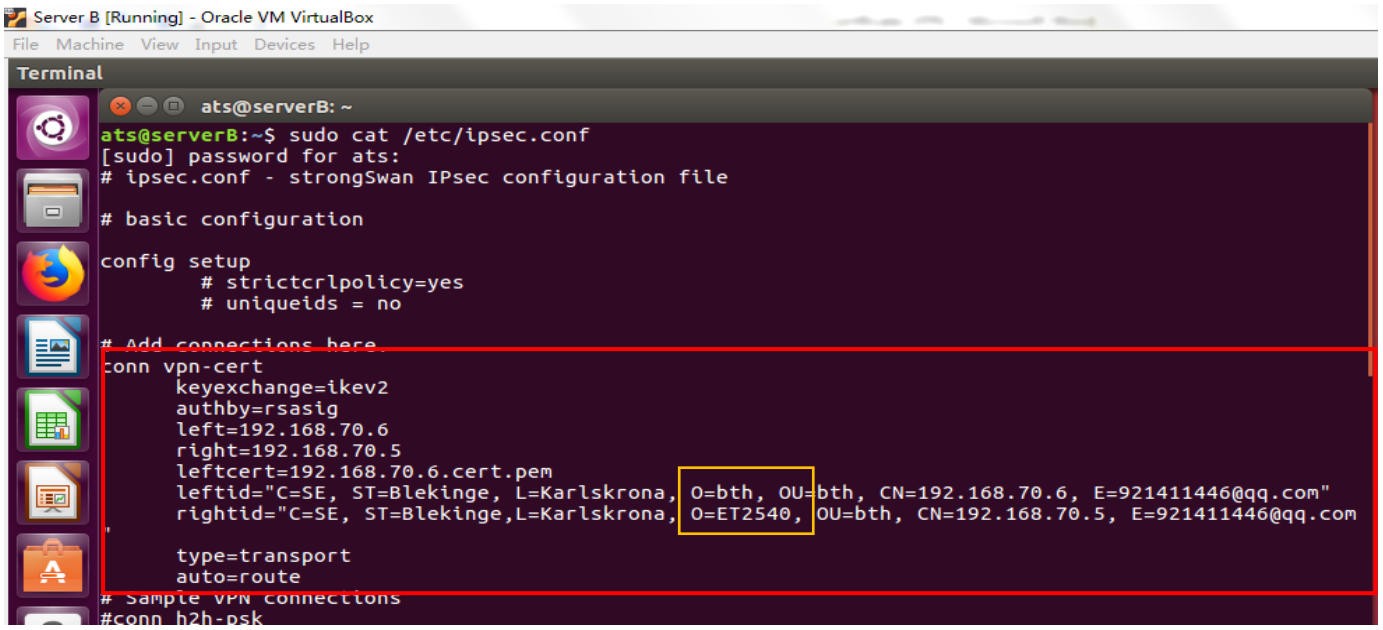
```
ats@serverA:~$ sudo ipsec listcacerts
[sudo] password for ats:

List of X.509 CA Certificates:

  subject:  "C=SE, ST=Blekinge, O=ET2540, OU=,, CN=RONG PENG CA1, E=921411446@qq.com"
  issuer:   "C=SE, ST=Blekinge, L=Karlskrona, O=ET2540, OU=BTH, CN=RONG PENG ROOT, E=921411446@qq.com"
  serial:    10:00
  validity:  not before Dec 01 15:42:54 2017, ok
             not after  Nov 29 15:42:54 2027, ok
  pubkey:    RSA 4096 bits
  keyid:     27:3d:aa:3b:3c:c8:47:b7:d1:6d:03:a7:be:78:9d:fe:e7:2b:5e:5b
  subjkey:   54:86:e6:f0:8e:6f:88:02:1b:75:e6:2c:01:d3:be:1a:d3:99:b5:8d
  authkey:   52:fa:0c:25:a7:cb:9e:97:1b:cf:ca:fc:74:14:23:91:15:61:7a:7f
  pathlen:   0

  subject:  "C=SE, ST=Blekinge, L=Karlskrona, O=ET2540, OU=BTH, CN=RONG PENG ROOT, E=921411446@qq.com"
  issuer:   "C=SE, ST=Blekinge, L=Karlskrona, O=ET2540, OU=BTH, CN=RONG PENG ROOT, E=921411446@qq.com"
  serial:    bc:fe:ba:67:3b:62:eb:4b
  validity:  not before Nov 30 17:03:05 2017, ok
             not after  Nov 25 17:03:05 2037, ok
  pubkey:    RSA 4096 bits
  keyid:     fc:89:15:51:a3:32:d9:85:eb:be:b6:9b:da:dc:c3:0c:41:a9:07:cf
  subjkey:   52:fa:0c:25:a7:cb:9e:97:1b:cf:ca:fc:74:14:23:91:15:61:7a:7f
  authkey:   52:fa:0c:25:a7:cb:9e:97:1b:cf:ca:fc:74:14:23:91:15:61:7a:7f
```

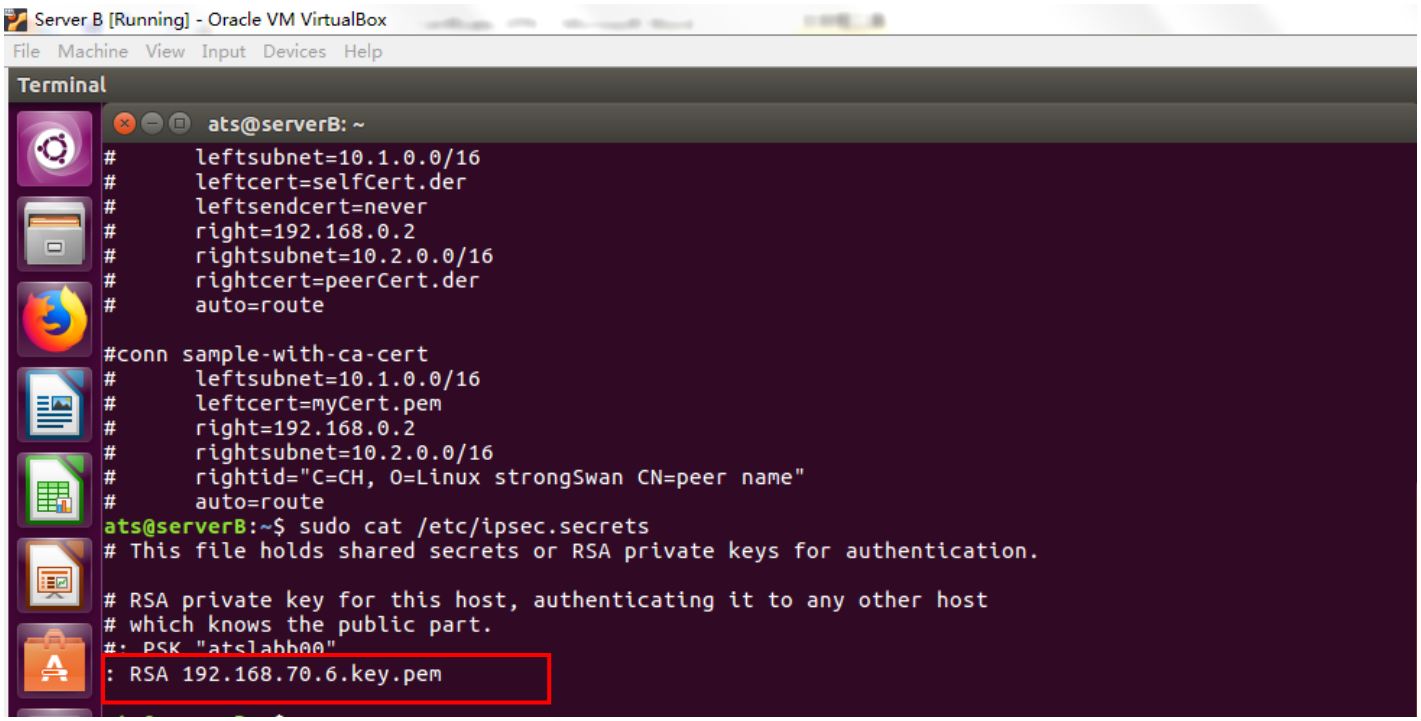
ipsec.conf:



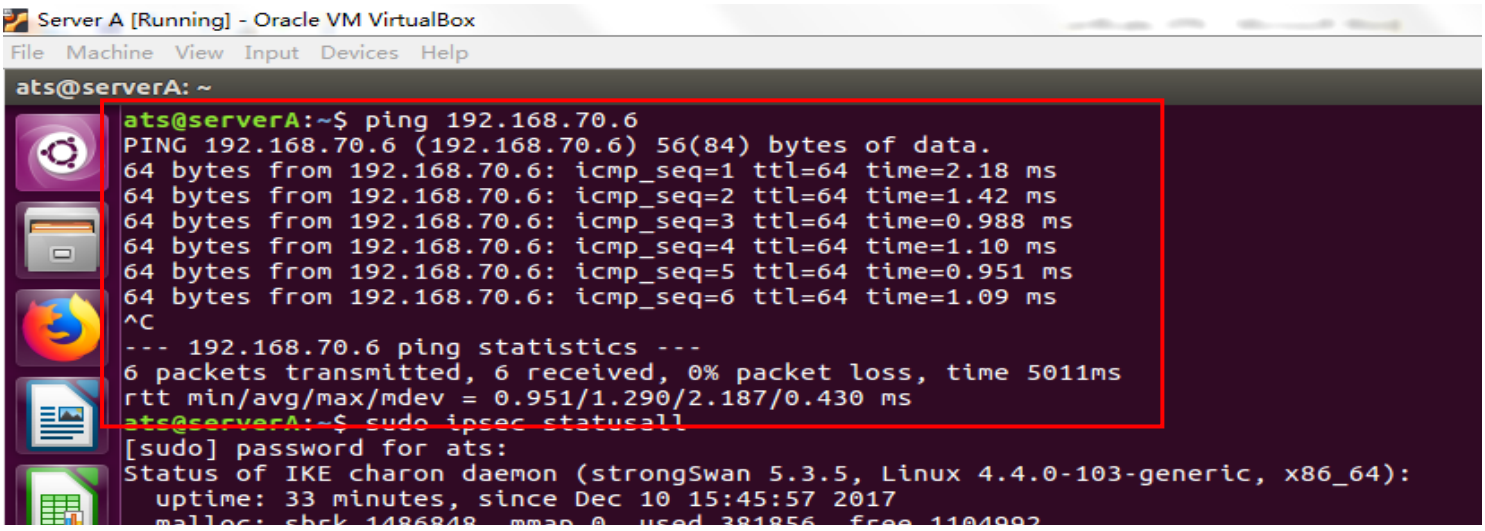
ipsec.secrets:

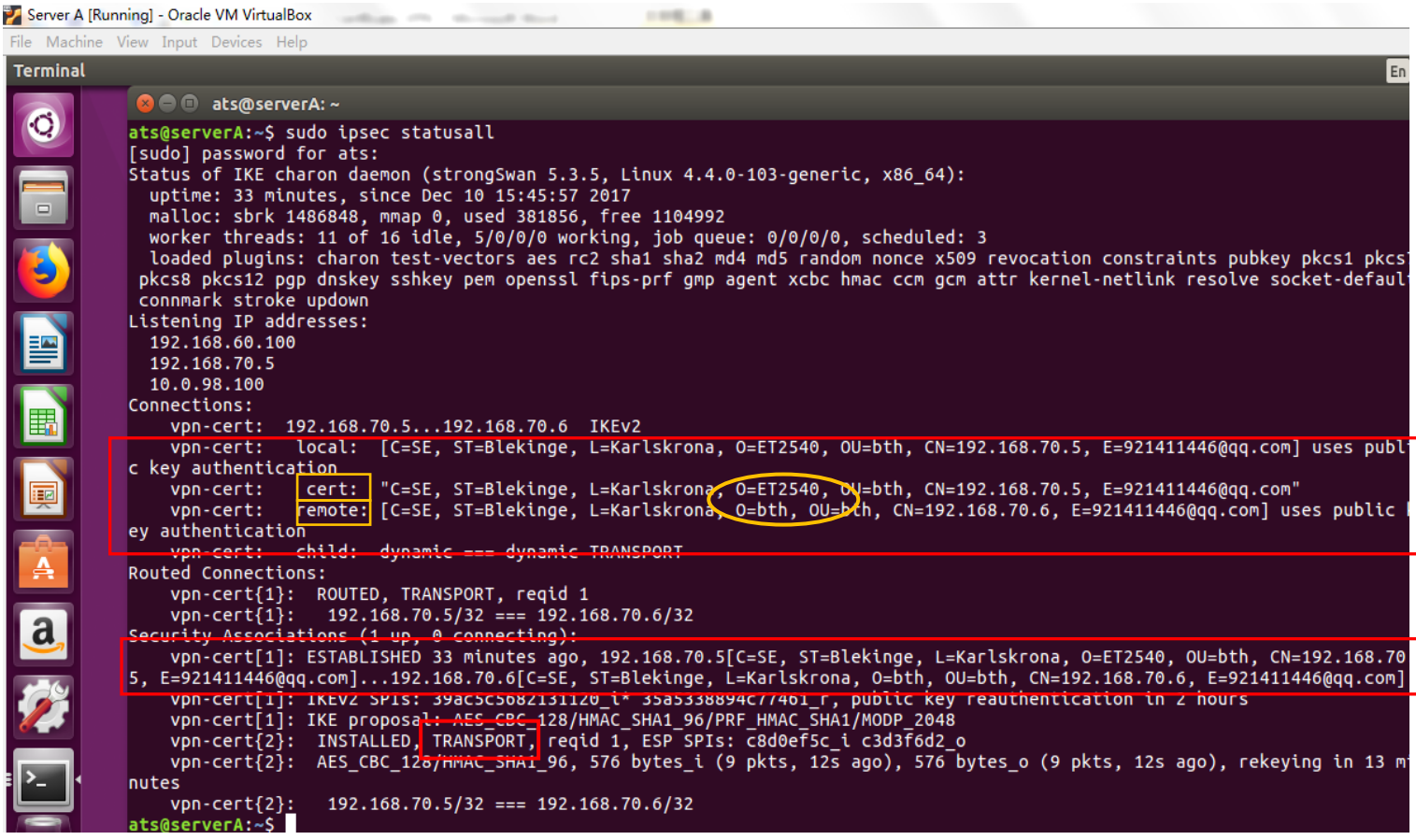


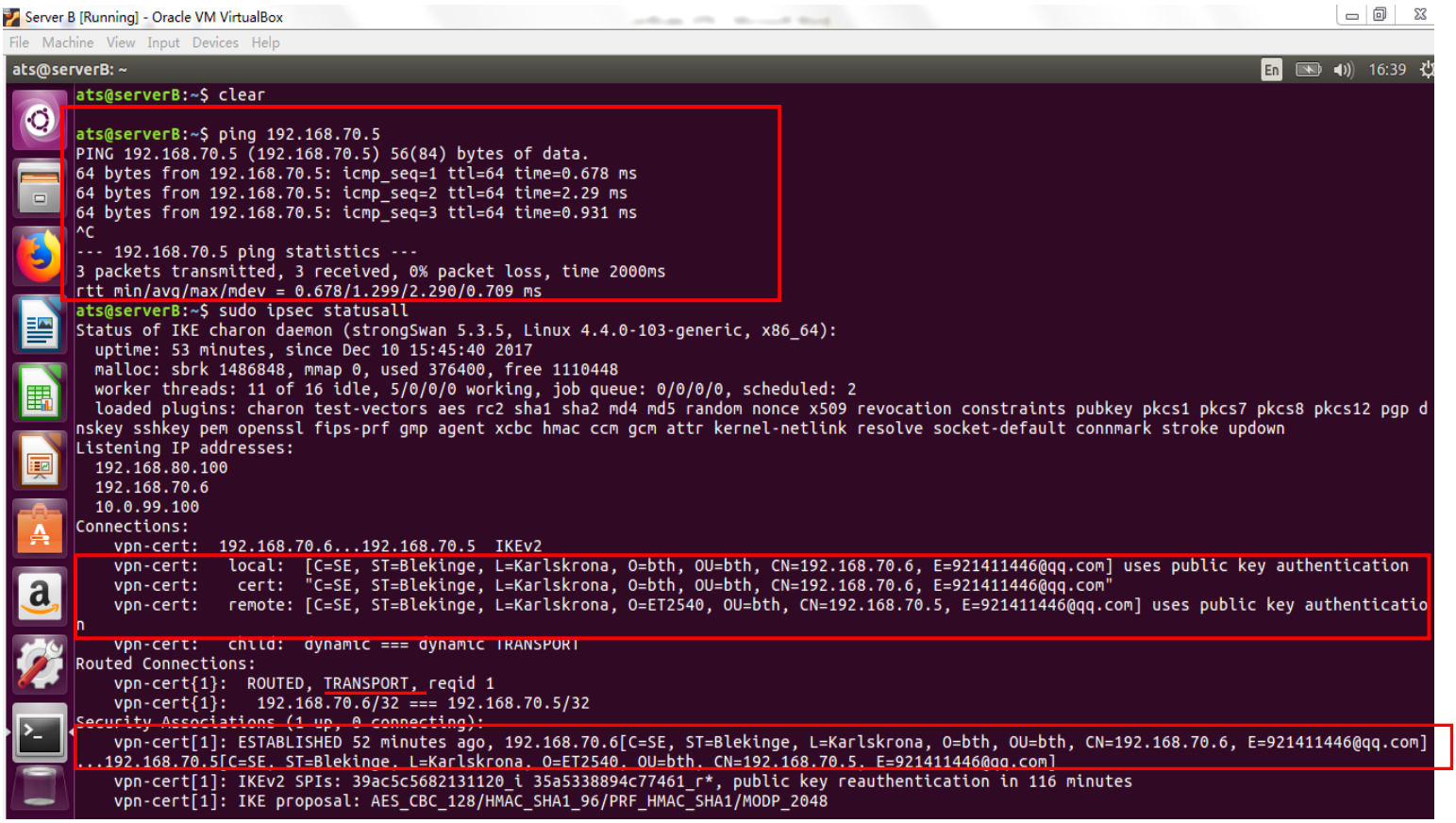




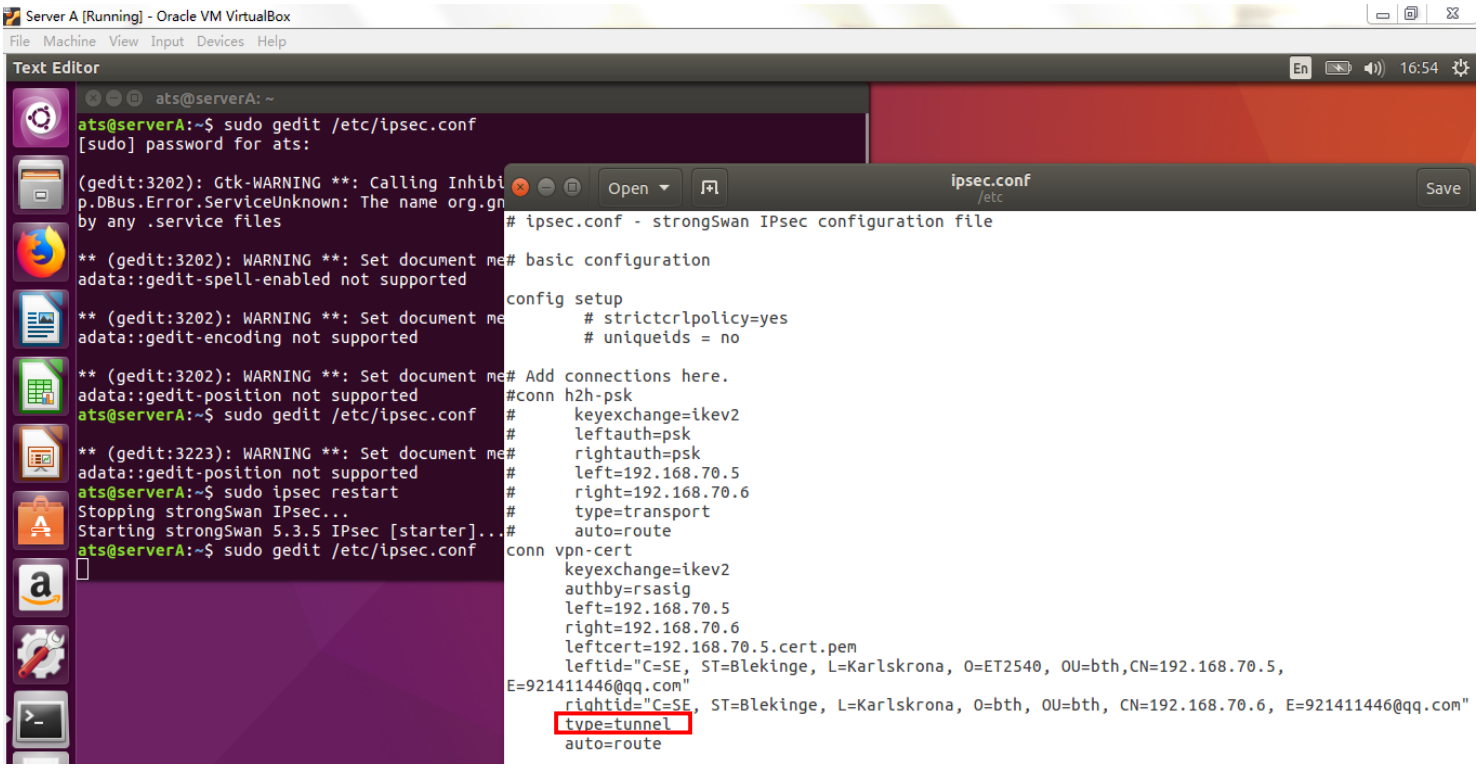
Ping 192.168.70.6 from serverA:

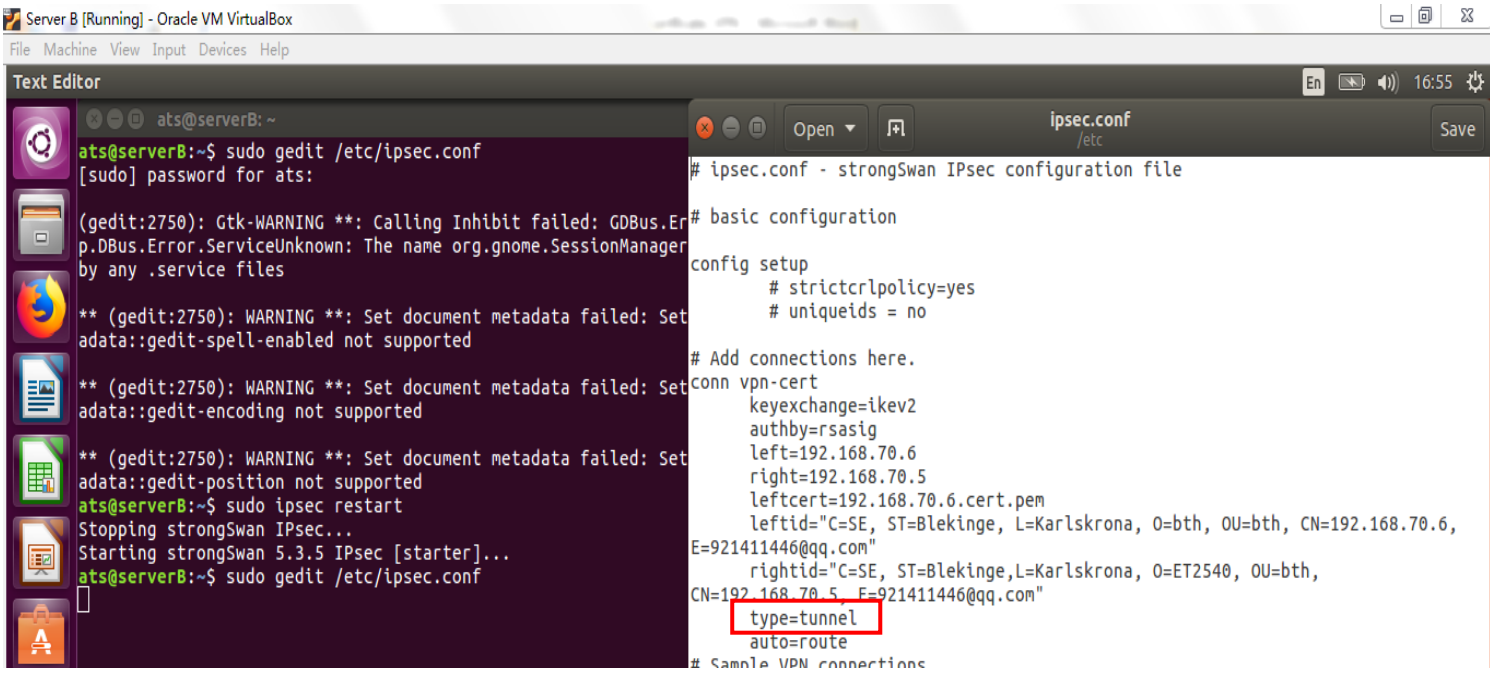


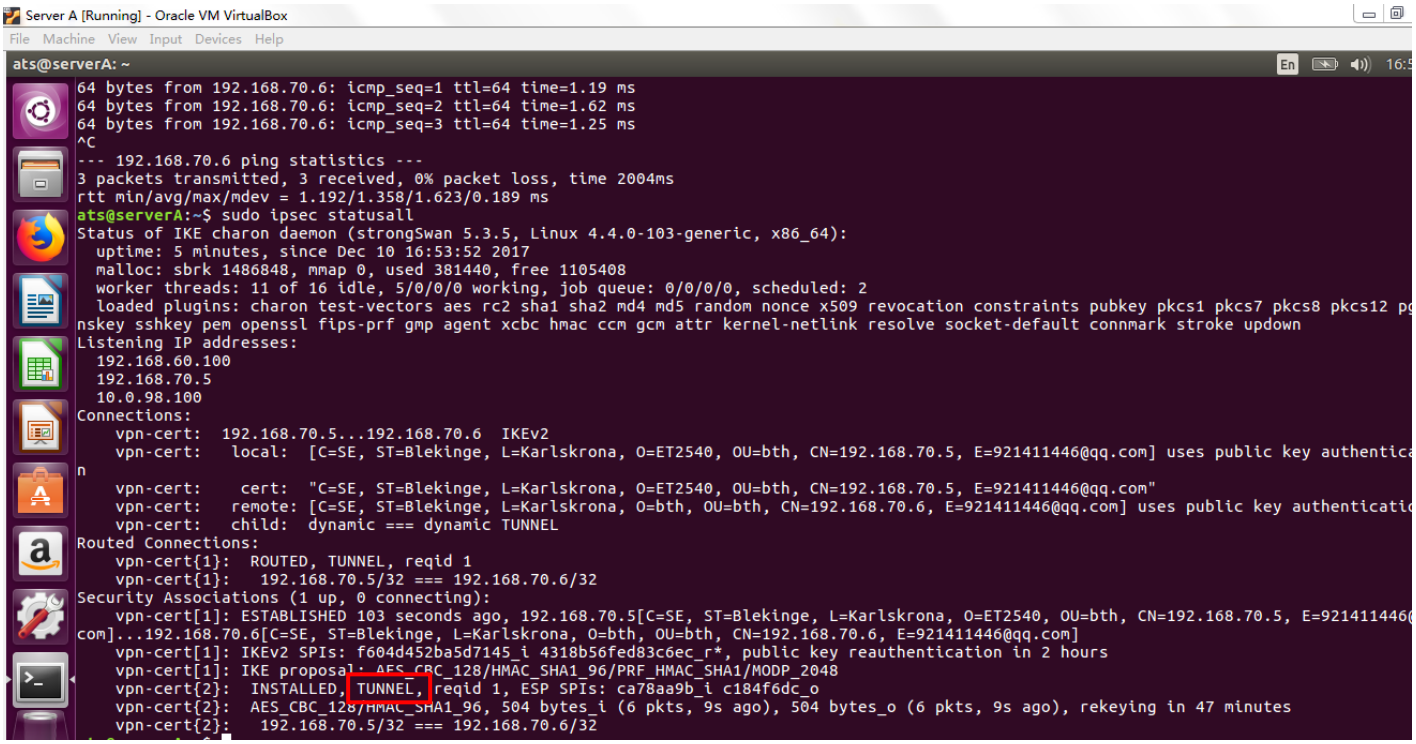


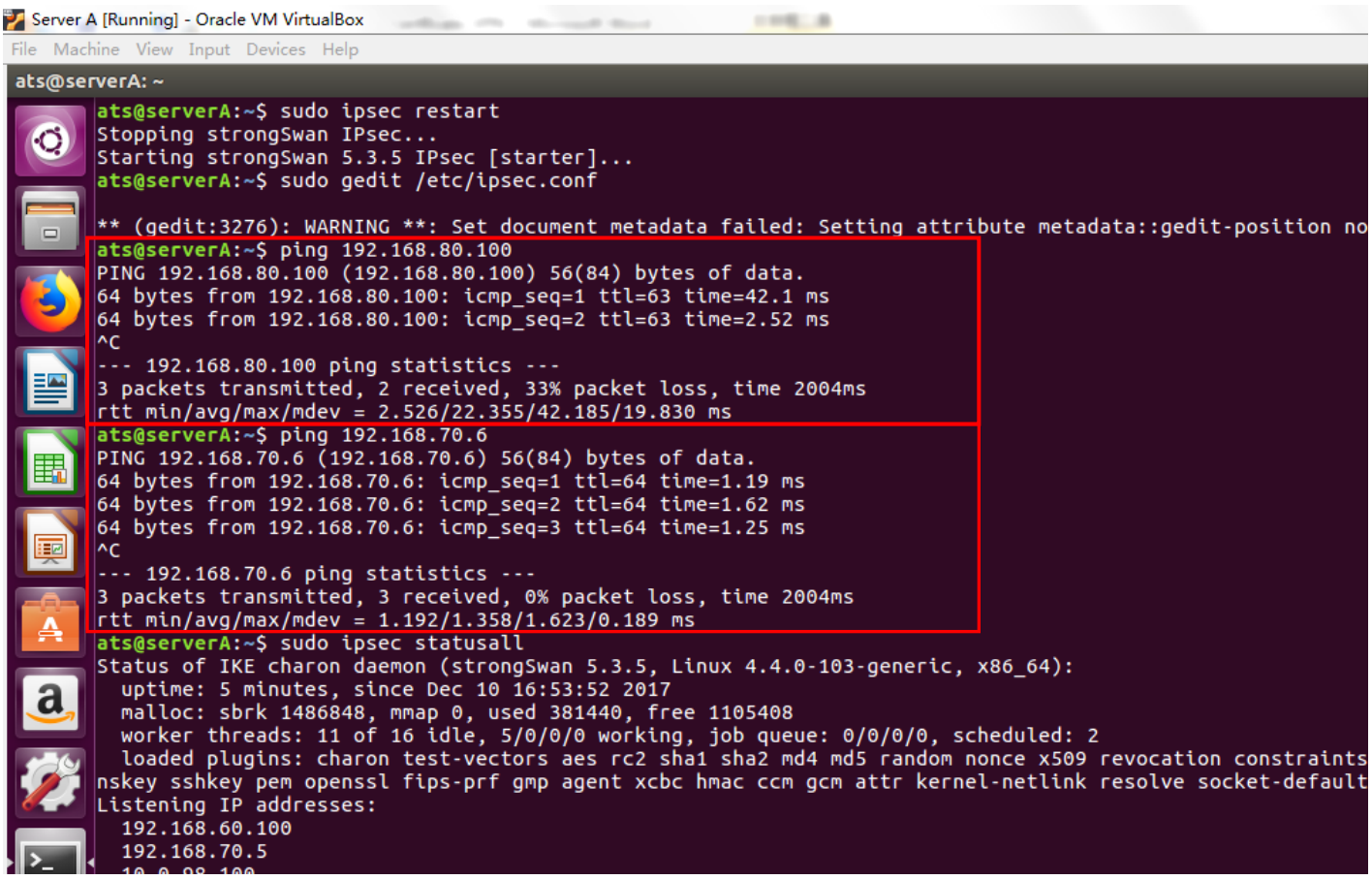

Server B [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
ats@serverB: ~
16:39
ats@serverB:~$ clear
ats@serverB:~$ ping 192.168.70.5
PING 192.168.70.5 (192.168.70.5) 56(84) bytes of data.
64 bytes from 192.168.70.5: icmp_seq=1 ttl=64 time=0.678 ms
64 bytes from 192.168.70.5: icmp_seq=2 ttl=64 time=2.29 ms
64 bytes from 192.168.70.5: icmp_seq=3 ttl=64 time=0.931 ms
^C
--- 192.168.70.5 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2000ms
rtt min/avg/max/mdev = 0.678/1.299/2.290/0.709 ms
ats@serverB:~$ sudo ipsec statusall
Status of IKE charon daemon (strongSwan 5.3.5, Linux 4.4.0-103-generic, x86_64):
uptime: 53 minutes, since Dec 10 15:45:40 2017
malloc: sbrk 1486848, mmap 0, used 376400, free 1110448
worker threads: 11 of 16 idle, 5/0/0/0 working, job queue: 0/0/0/0, scheduled: 2
loaded plugins: charon test-vectors aes rc2 sha1 sha2 md4 md5 random nonce x509 revocation constraints pubkey pkcs1 pkcs7 pkcs8 pkcs12 pgp d
nskey sshkey pem openssl fips-prf gmp agent xcbc hmac ccm gcm attr kernel-netlink resolve socket-default connmark stroke updown
Listening IP addresses:
192.168.80.100
192.168.70.6
10.0.99.100
Connections:
vpn-cert: 192.168.70.6...192.168.70.5 IKEv2
vpn-cert: local: [C=SE, ST=Blekinge, L=Karlskrona, O=bth, OU=bth, CN=192.168.70.6, E=921411446@qq.com] uses public key authentication
vpn-cert: cert: "C=SE, ST=Blekinge, L=Karlskrona, O=bth, OU=bth, CN=192.168.70.6, E=921411446@qq.com"
vpn-cert: remote: [C=SE, ST=Blekinge, L=Karlskrona, O=ET2540, OU=bth, CN=192.168.70.5, E=921411446@qq.com] uses public key authenticatio
n
vpn-cert: child: dynamic === dynamic TRANSPORT
Routed Connections:
vpn-cert{1}: ROUTED, TRANSPORT, reqid 1
vpn-cert{1}: 192.168.70.6/32 === 192.168.70.5/32
Security Associations (1 up, 0 connecting):
vpn-cert[1]: ESTABLISHED 52 minutes ago, 192.168.70.6[C=SE, ST=Blekinge, L=Karlskrona, O=bth, OU=bth, CN=192.168.70.6, E=921411446@qq.com]
...192.168.70.5[C=SE, ST=Blekinge, L=Karlskrona, O=ET2540, OU=bth, CN=192.168.70.5, E=921411446@qq.com]
vpn-cert[1]: IKEv2 SPIs: 39ac5c5682131120_i 35a5338894c77461_r*, public key reauthentication in 116 minutes
vpn-cert[1]: IKE proposal: AES_CBC_128/HMAC_SHA1_96/PRF_HMAC_SHA1/MODP_2048

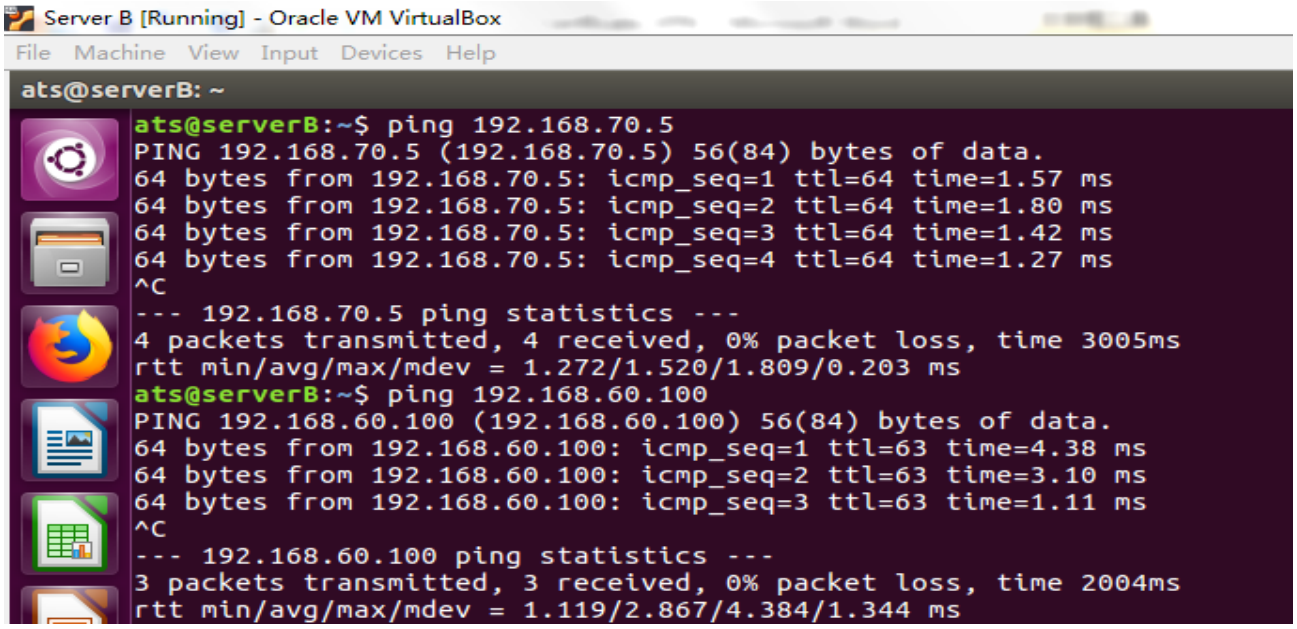
Task19:Tunnel mode VPN with cert authentication between Server A and Server B



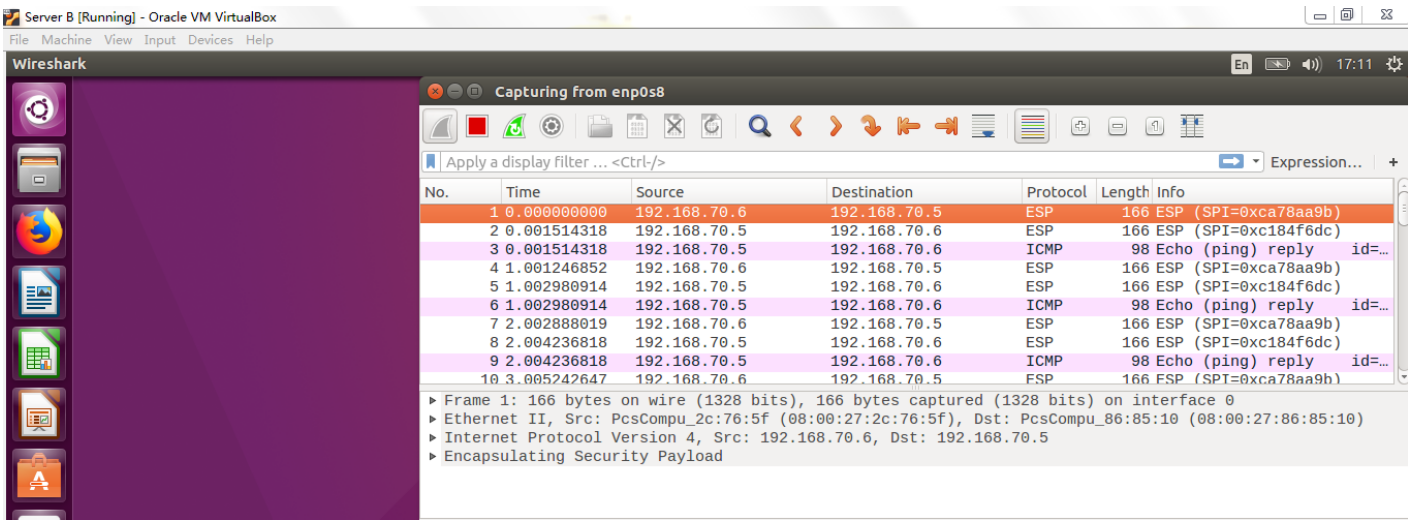


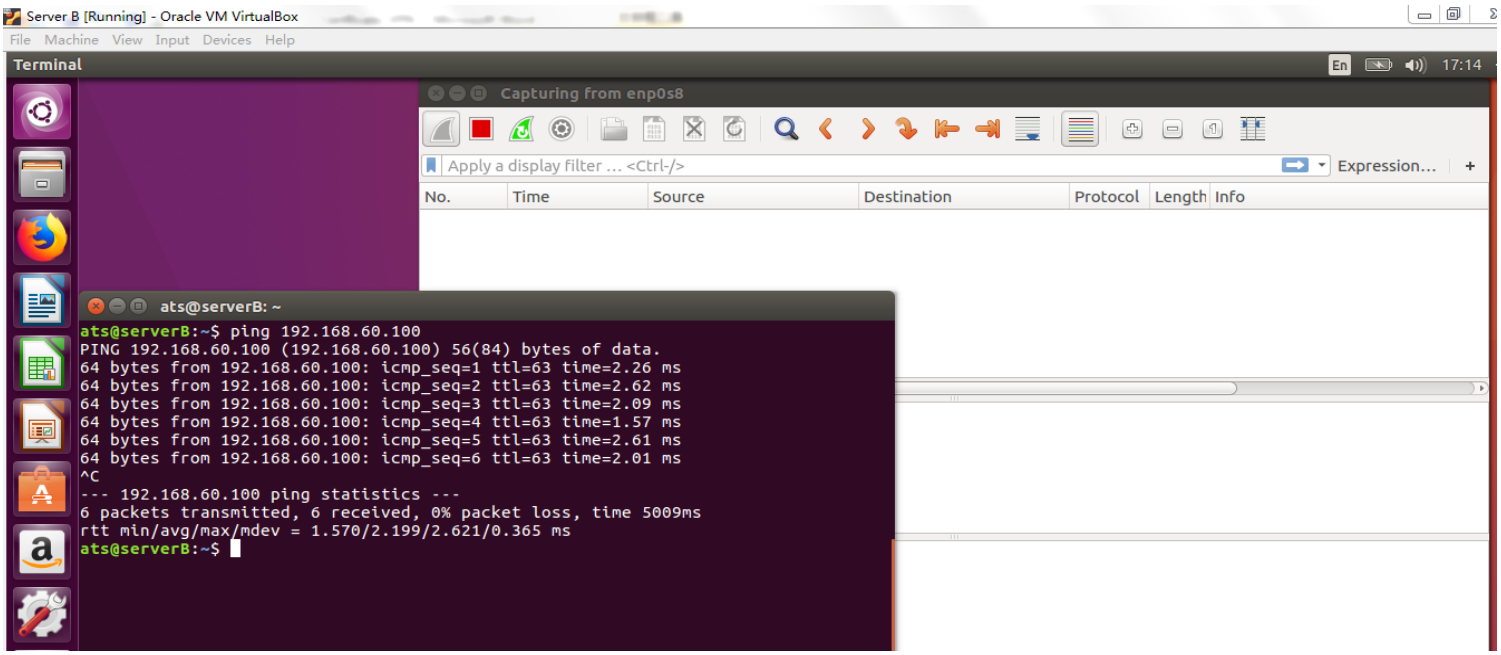

Server A [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
ats@serverA: ~
64 bytes from 192.168.70.6: icmp_seq=1 ttl=64 time=1.19 ms
64 bytes from 192.168.70.6: icmp_seq=2 ttl=64 time=1.62 ms
64 bytes from 192.168.70.6: icmp_seq=3 ttl=64 time=1.25 ms
^C
--- 192.168.70.6 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2004ms
rtt min/avg/max/mdev = 1.192/1.358/1.623/0.189 ms
ats@serverA:~$ sudo ipsec statusall
Status of IKE charon daemon (strongSwan 5.3.5, Linux 4.4.0-103-generic, x86_64):
uptime: 5 minutes, since Dec 10 16:53:52 2017
malloc: sbrk 1486848, mmap 0, used 381440, free 1105408
worker threads: 11 of 16 idle, 5/0/0/0 working, job queue: 0/0/0/0, scheduled: 2
loaded plugins: charon test-vectors aes rc2 sha1 sha2 md4 md5 random nonce x509 revocation constraints pubkey pkcs1 pkcs7 pkcs8 pkcs12 pg
nskey sshkey pem openssl fips-prf gmp agent xcbc hmac ccm gcm attr kernel-netlink resolve socket-default connmark stroke updown
Listening IP addresses:
192.168.60.100
192.168.70.5
10.0.98.100
Connections:
vpn-cert: 192.168.70.5...192.168.70.6 IKEv2
vpn-cert: local: [C=SE, ST=Blekinge, L=Karlskrona, O=ET2540, OU=bth, CN=192.168.70.5, E=921411446@qq.com] uses public key authentica
n
vpn-cert: cert: "C=SE, ST=Blekinge, L=Karlskrona, O=ET2540, OU=bth, CN=192.168.70.5, E=921411446@qq.com"
vpn-cert: remote: [C=SE, ST=Blekinge, L=Karlskrona, O=bth, OU=bth, CN=192.168.70.6, E=921411446@qq.com] uses public key authenticati
vpn-cert: child: dynamic === dynamic TUNNEL
Routed Connections:
vpn-cert{1}: ROUTED, TUNNEL, reqid 1
vpn-cert{1}: 192.168.70.5/32 === 192.168.70.6/32
Security Associations (1 up, 0 connecting):
vpn-cert[1]: ESTABLISHED 103 seconds ago, 192.168.70.5[C=SE, ST=Blekinge, L=Karlskrona, O=ET2540, OU=bth, CN=192.168.70.5, E=921411446(
com]...192.168.70.6[C=SE, ST=Blekinge, L=Karlskrona, O=bth, OU=bth, CN=192.168.70.6, E=921411446@qq.com]
vpn-cert[1]: IKEv2 SPIs: f604d452ba5d7145_i 4318b56fed83c6ec_r*, public key reauthentication in 2 hours
vpn-cert[1]: IKE proposal: AES_CBC_128/HMAC_SHA1_96/PRF_HMAC_SHA1/MODP_2048
vpn-cert{2}: INSTALLED, TUNNEL, reqid 1, ESP SPIs: ca78aa9b_i c184f6dc_o
vpn-cert{2}: AES_CBC_128/HMAC_SHA1_96, 504 bytes_i (6 pkts, 9s ago), 504 bytes_o (6 pkts, 9s ago), rekeying in 47 minutes
vpn-cert{2}: 192.168.70.5/32 === 192.168.70.6/32


Server A [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
ats@serverA: ~
ats@serverA:~$ sudo ipsec restart
Stopping strongSwan IPsec...
Starting strongSwan 5.3.5 IPsec [starter]...
ats@serverA:~$ sudo gedit /etc/ipsec.conf
** (gedit:3276): WARNING **: Set document metadata failed: Setting attribute metadata::gedit-position no
ats@serverA:~$ ping 192.168.80.100
PING 192.168.80.100 (192.168.80.100) 56(84) bytes of data.
64 bytes from 192.168.80.100: icmp_seq=1 ttl=63 time=42.1 ms
64 bytes from 192.168.80.100: icmp_seq=2 ttl=63 time=2.52 ms
^C
--- 192.168.80.100 ping statistics ---
3 packets transmitted, 2 received, 33% packet loss, time 2004ms
rtt min/avg/max/mdev = 2.526/22.355/42.185/19.830 ms
ats@serverA:~$ ping 192.168.70.6
PING 192.168.70.6 (192.168.70.6) 56(84) bytes of data.
64 bytes from 192.168.70.6: icmp_seq=1 ttl=64 time=1.19 ms
64 bytes from 192.168.70.6: icmp_seq=2 ttl=64 time=1.62 ms
64 bytes from 192.168.70.6: icmp_seq=3 ttl=64 time=1.25 ms
^C
--- 192.168.70.6 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2004ms
rtt min/avg/max/mdev = 1.192/1.358/1.623/0.189 ms
ats@serverA:~$ sudo ipsec statusall
Status of IKE charon daemon (strongSwan 5.3.5, Linux 4.4.0-103-generic, x86_64):
uptime: 5 minutes, since Dec 10 16:53:52 2017
malloc: sbrk 1486848, mmap 0, used 381440, free 1105408
worker threads: 11 of 16 idle, 5/0/0/0 working, job queue: 0/0/0/0, scheduled: 2
loaded plugins: charon test-vectors aes rc2 sha1 sha2 md4 md5 random nonce x509 revocation constraints
nskey sshkey pem openssl fips-prf gmp agent xcbc hmac ccm gcm attr kernel-netlink resolve socket-default
Listening IP addresses:
192.168.60.100
192.168.70.5




```
ats@serverB:~$ sudo ipsec statusall
[sudo] password for ats:
Status of IKE charon daemon (strongSwan 5.3.5, Linux 4.4.0-103-generic, x86_64):
  uptime: 12 minutes, since Dec 10 16:55:11 2017
  malloc: sbrk 1486848, mmap 0, used 384416, free 1102432
  worker threads: 11 of 16 idle, 5/0/0/0 working, job queue: 0/0/0/0, scheduled: 3
  loaded plugins: charon test-vectors aes rc2 sha1 sha2 md4 md5 random nonce x509 revocation constraints pubkey pkcs1 pkcs7 pkcs8 pkcs12 pgp d
nskey sshkey pem openssl fips-prf gmp agent xcbc hmac ccm gcm attr kernel-netlink resolve socket-default connmark stroke updown
Listening IP addresses:
  192.168.80.100
  192.168.70.6
  10.0.99.100
Connections:
    vpn-cert:  192.168.70.6...192.168.70.5  IKEv2
    vpn-cert:   local:  [C=SE, ST=Blekinge, L=Karlskrona, O=bth, OU=bth, CN=192.168.70.6, E=921411446@qq.com] uses public key authentication
    vpn-cert:    cert:  "C=SE, ST=Blekinge, L=Karlskrona, O=bth, OU=bth, CN=192.168.70.6, E=921411446@qq.com"
    vpn-cert:   remote: [C=SE, ST=Blekinge, L=Karlskrona, O=ET2540, OU=bth, CN=192.168.70.5, E=921411446@qq.com] uses public key authenticatio
n
    vpn-cert:   child:  dynamic === dynamic TUNNEL
Routed Connections:
    vpn-cert{1}:  ROUTED, TUNNEL, reqid 1
    vpn-cert{1}:   192.168.70.6/32 === 192.168.70.5/32
Security Associations (1 up, 0 connecting):
    vpn-cert[1]: ESTABLISHED 9 minutes ago, 192.168.70.6[C=SE, ST=Blekinge, L=Karlskrona, O=bth, OU=bth, CN=192.168.70.6, E=921411446@qq.com].
..192.168.70.5[C=SE, ST=Blekinge, L=Karlskrona, O=ET2540, OU=bth, CN=192.168.70.5, E=921411446@qq.com]
    vpn-cert[1]: IKEv2 SPIs: f604d452ba5d7145_i* 4318b56fed83c6ec_r, public key reauthentication in 2 hours
    vpn-cert[1]: IKE proposal: AES_CBC_128/HMAC_SHA1_96/PRF_HMAC_SHA1/MODP_2048
    vpn-cert{2}:  INSTALLED, TUNNEL, reqid 1, ESP SPIs: c184f6dc_i ca78aa9b_o
    vpn-cert{2}:  AES_CBC_128/HMAC_SHA1_96, 840 bytes_i (10 pkts, 277s ago), 840 bytes_o (10 pkts, 277s ago), rekeying in 39 minutes
    vpn-cert{2}:   192.168.70.6/32 === 192.168.70.5/32
```





Look, only encrypted outbound traffic.

Task20:Tunnel mode VPN with IP forwarding for client A an client B

Because some parts are same as serverA, so I copied the firewall.sh to serverB. This is prepare for the last Task need.

```
ats@serverA:~$ sudo scp firewall.sh.tar.gz ats@192.168.70.6:/home/ats
ats@192.168.70.6's password:
firewall.sh.tar.gz                            100% 1214     1.2KB/s   00:00
ats@serverA:~$
```

I have used the iptables lab's command which I have learned, firstly accept the forward traffic ,and let allow the NAT interface to build ESTABLISHED,RELATED state, and keep track on it. And let the POSTROUTING chain 's outbound traffic instead their IP of the SNAT IP. Also the necessary one is temporarily change the net.ipv4.ip_forward=1, that will allow forward traffic.

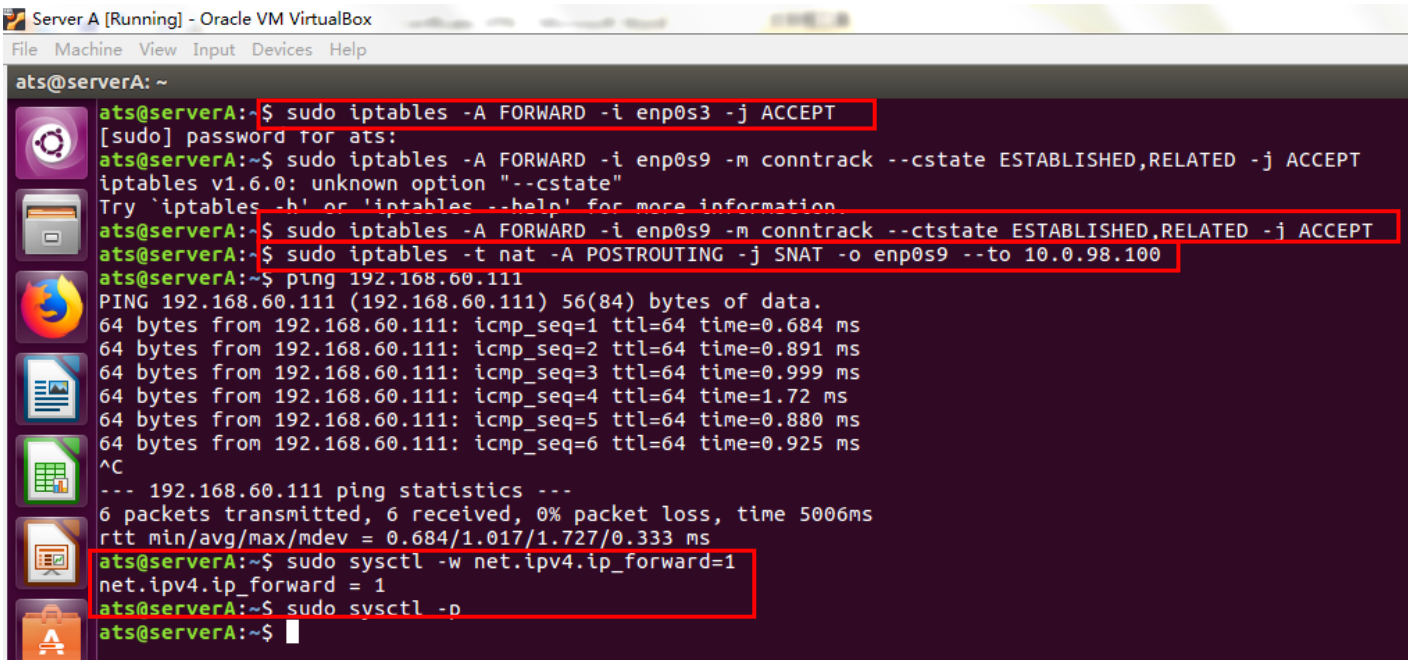
sudo iptables -A FORWARD -i enp0s3 -j ACCEPT

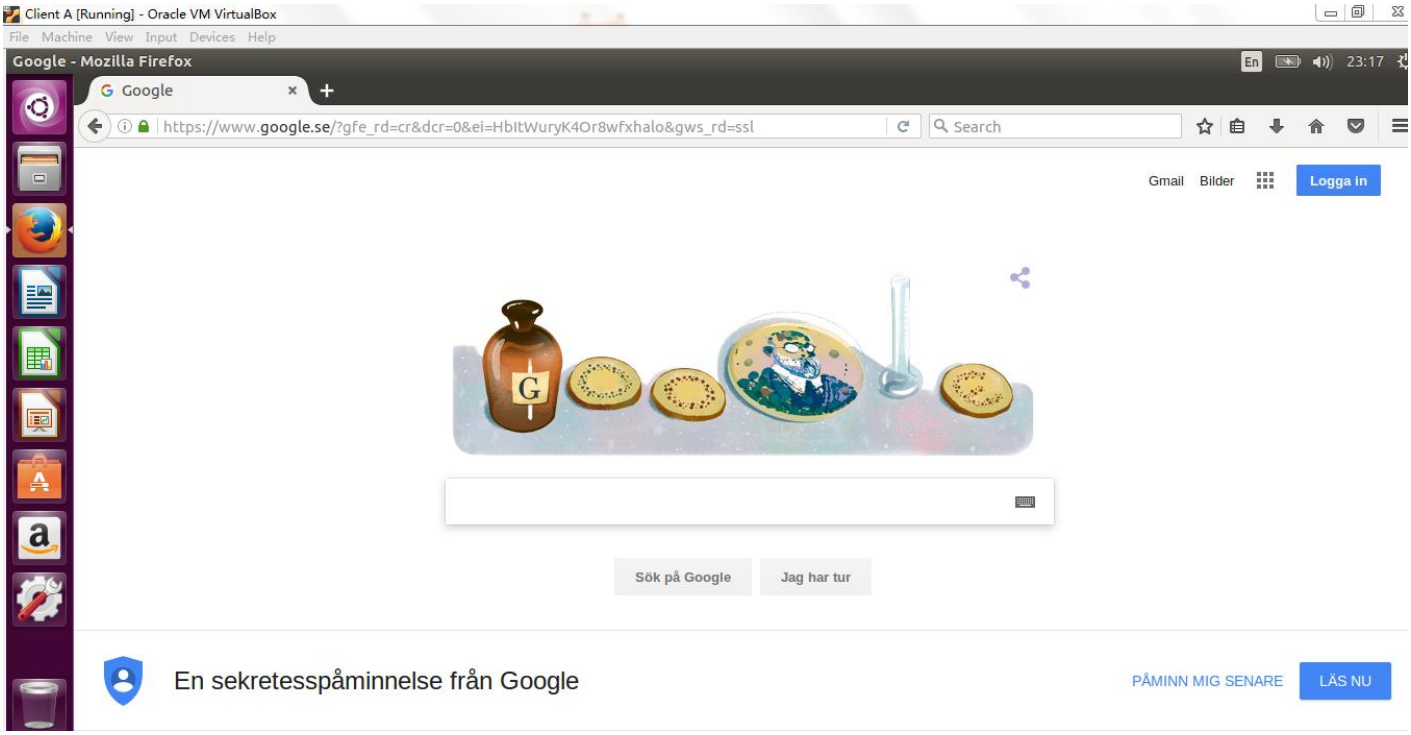
sudo iptables -A FORWARD -i enp0s9 -m conntrack -ctstate ESTABLISHED,RELATED -j ACCEPT

sudo iptables -t nat -A POSTROUTING -j SNAT -o enp0s9 -to 10.0.98.100 (ServerB :10.0.99.100)

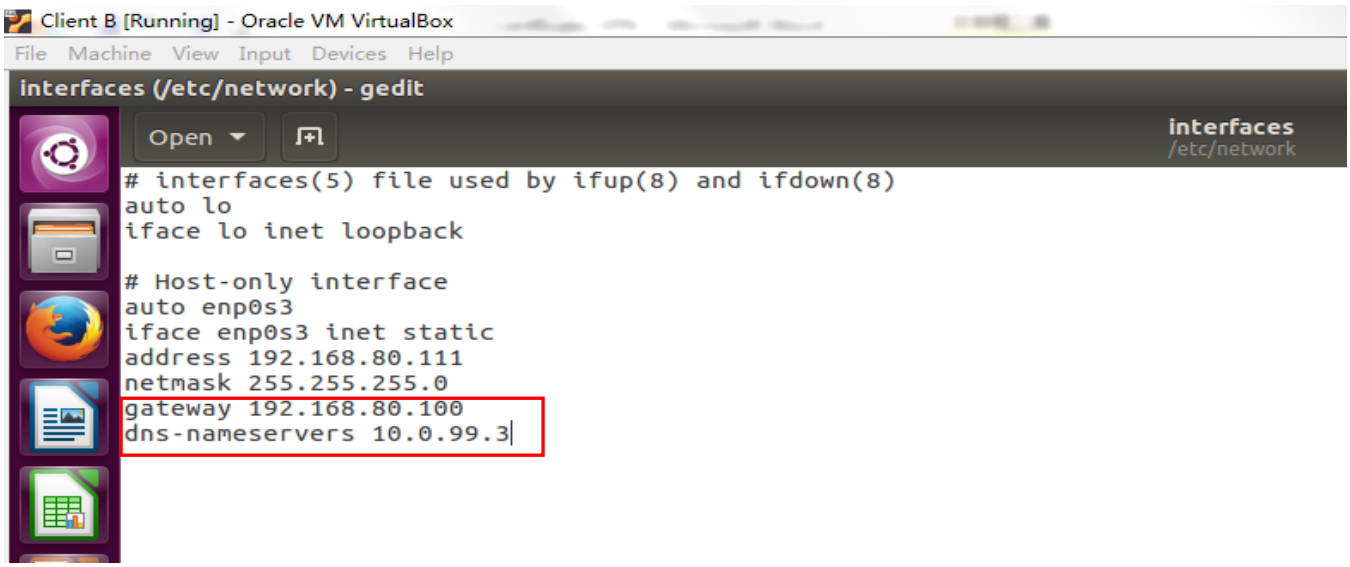
sudo sysctl -w net.ipv4.ip_forward=1

sudo sysctl -p

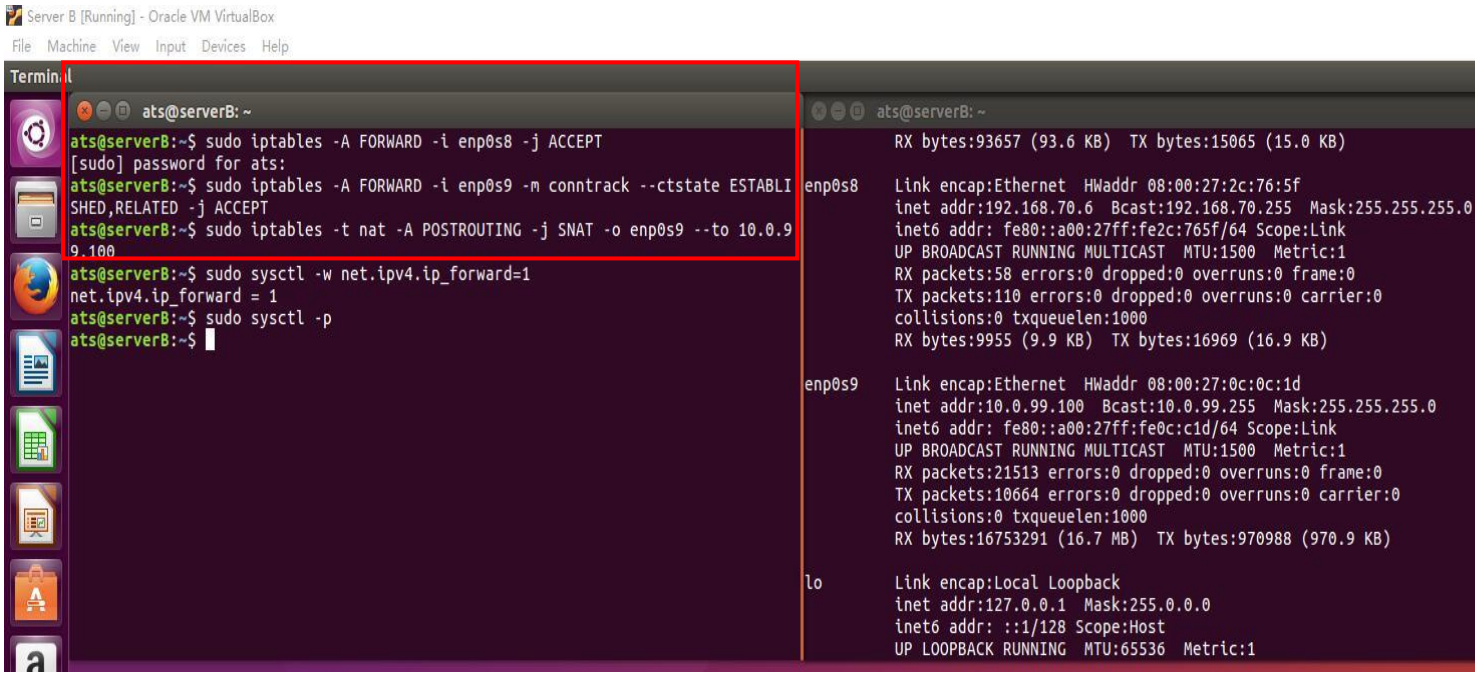




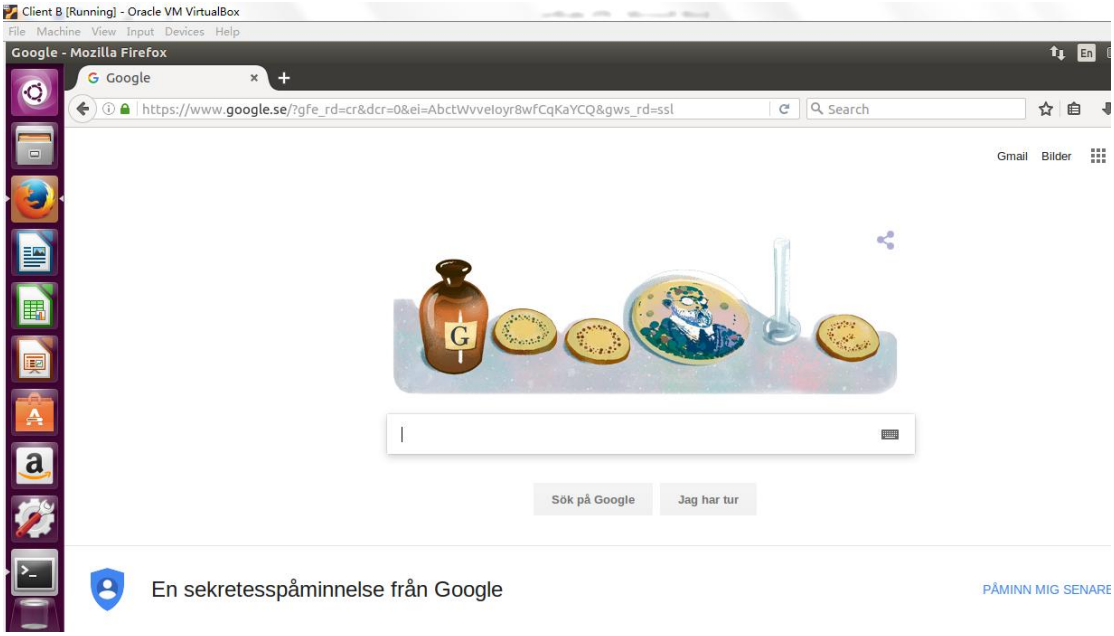
Configuring the clientB ,add the gateway and dns-nameservers ,let the traffic between with serverB.



Same like serverA &clientA 's configuration:



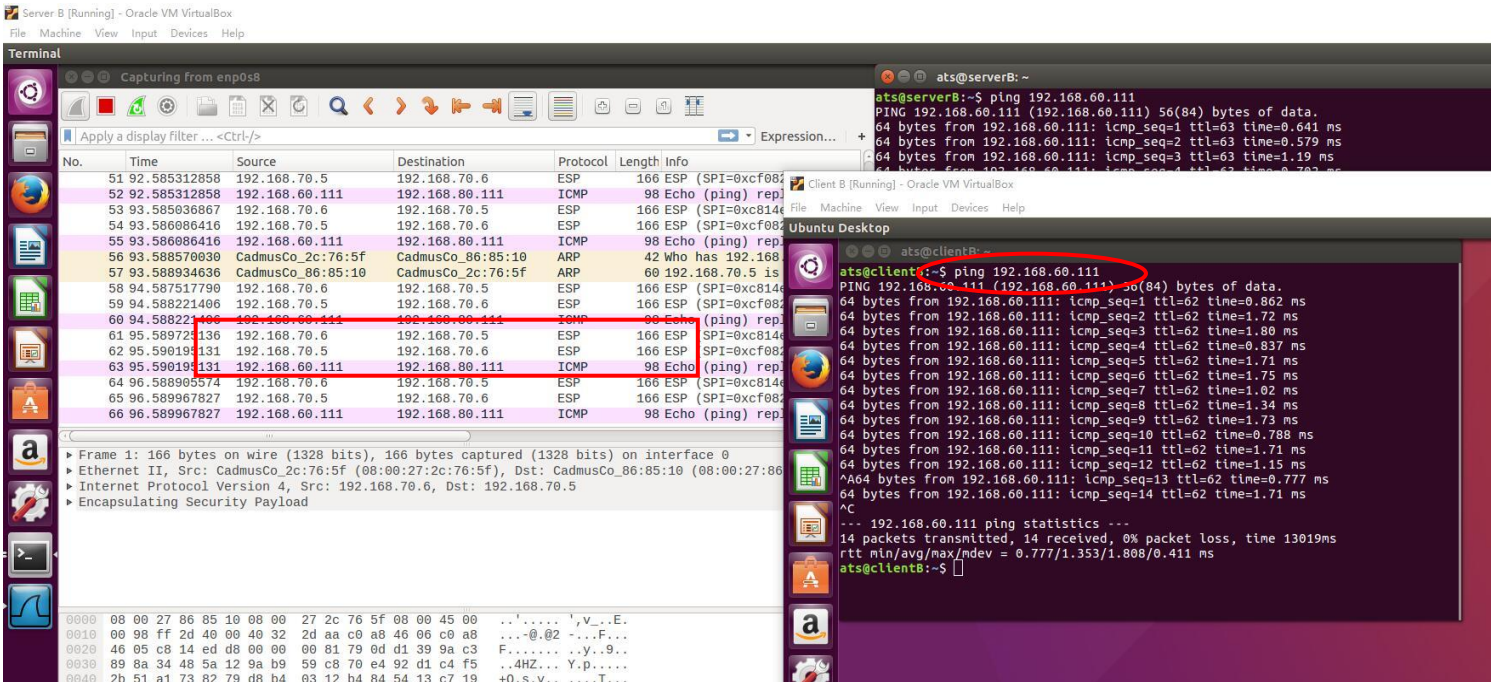
Look, it can access the Google.



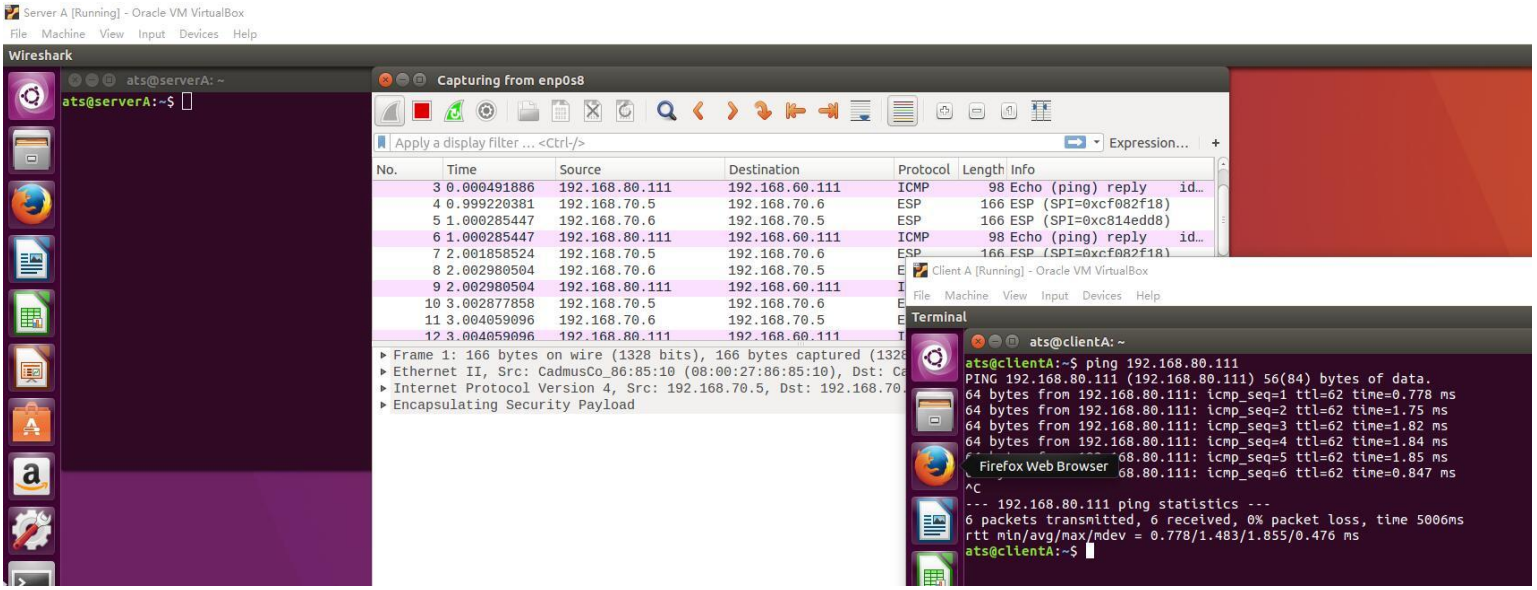

```
conn vpn-cert
      keyexchange=ikev2
      authby=rsasig
      left=192.168.70.6
      right=192.168.70.5
      leftcert=192.168.70.6.cert.pem
      leftid="C=SE, ST=Blekinge, L=Karlskrona, O=bth, OU=bth, CN=192.168.70.6, E=921411446@qq.com"
      rightid="C=SE, ST=Blekinge, L=Karlskrona, O=ET2540, OU=bth, CN=192.168.70.5, E=921411446@qq.com"
      leftsubnet=192.168.80.0/24
      rightsubnet=192.168.60.0/24
      type=tunnel
      auto=route
```

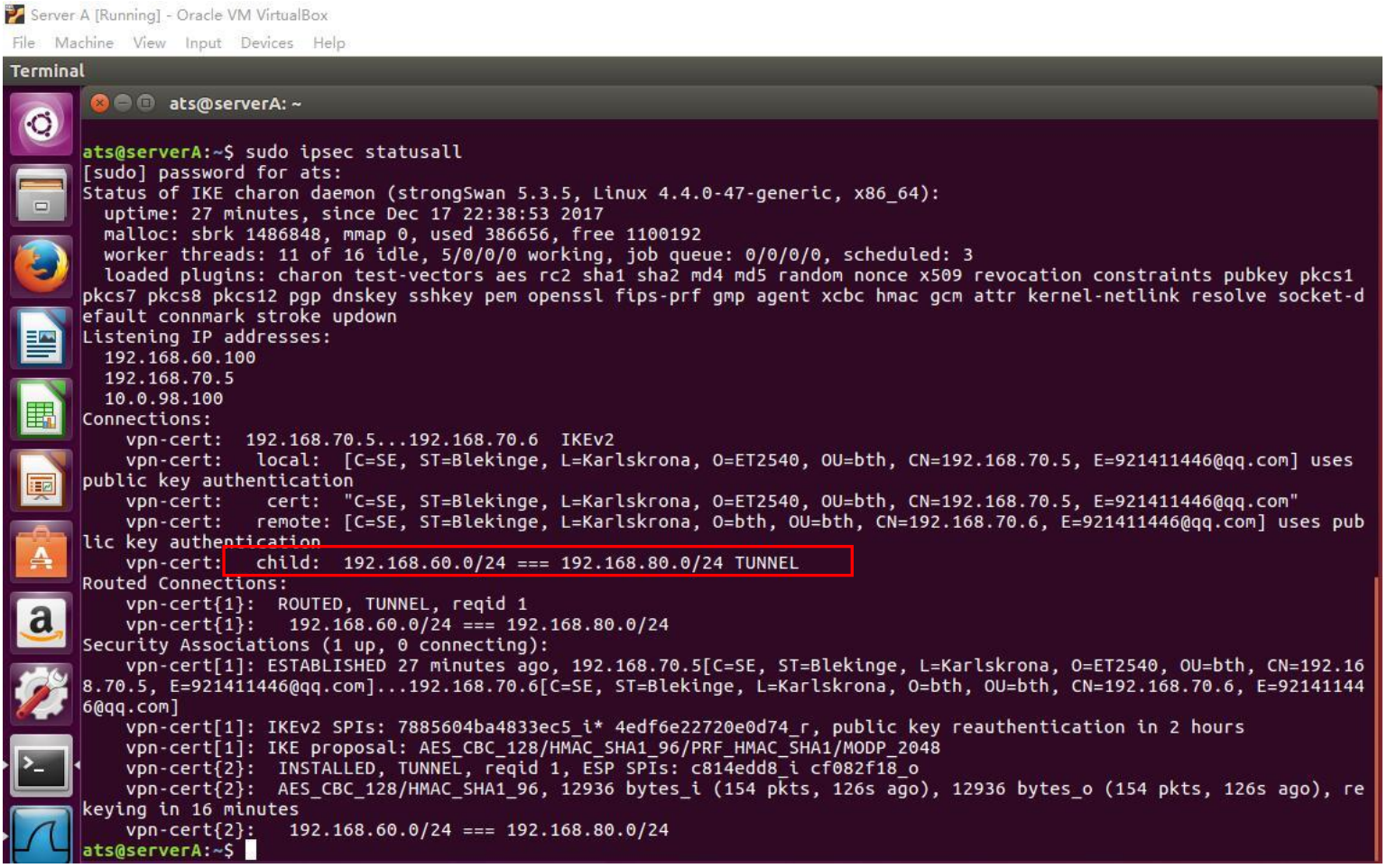
This is server B 's ipsec.conf, server A just reverses all the tag about left and right. In this task I add this subnet pairs to make sure the private network subnet.

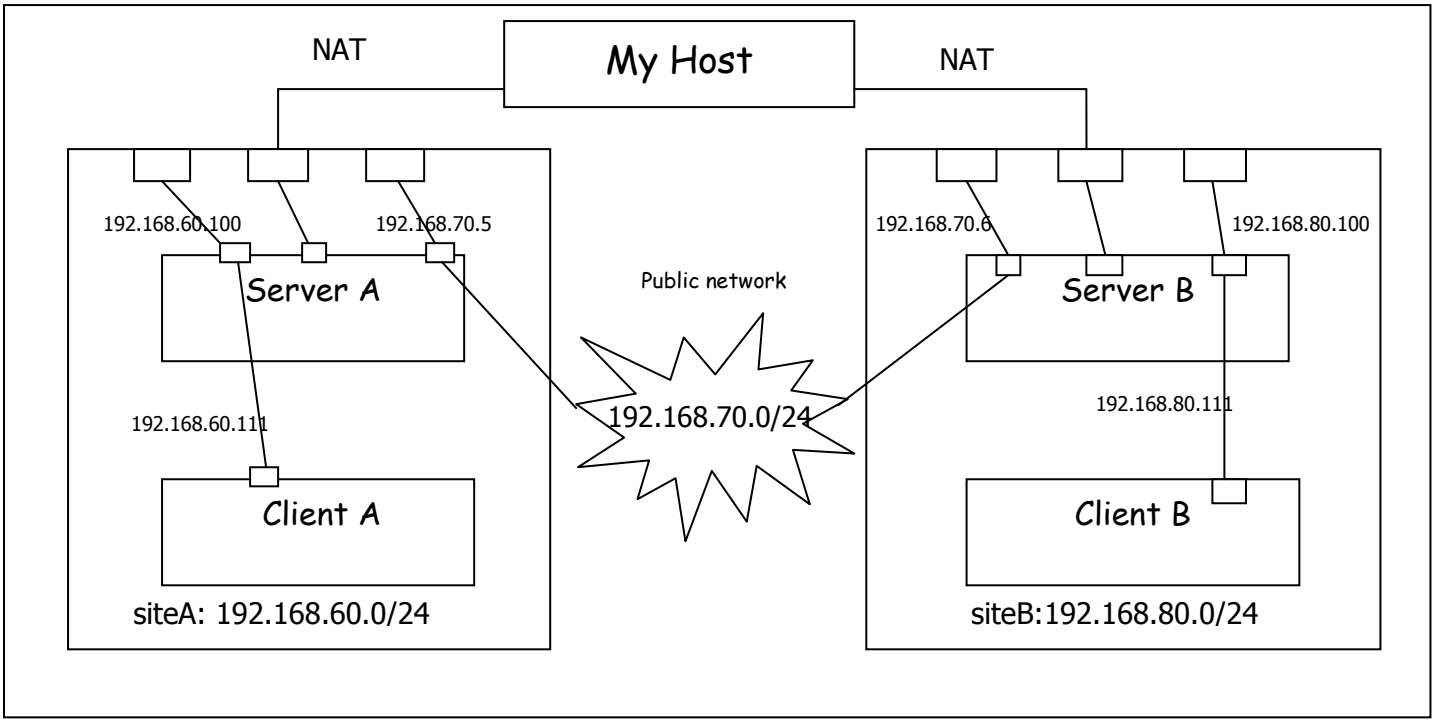
Then I used client B ping 192.168.60.111, and open the WireShark to monitor the enp0s8(192.168.70.6) to see the protocols.



The above screenshot and the following pictures are the evidences to prove CilentB communicating with Client A over ServerA and ServerB IPsec tunnel.







About this Task I remembered professor have drew such picture for me, that is better to understand the whole lab2 network interfaces' relation. That is why I configured my ipsec.conf like that way, add left and right subnet.

Task21:Site A to Site B VPN with default DROP firewall rules

In this task , I used lab1's firewall.sh, because in the last Task of the lab1 , clientA can surf net, so firstly , I add the lab1 's firewall.sh, then I wrote my this task firewall rules in the terminal. Besides the lab1's firewall rules, I added :

```
ats@serverA:~$ sudo iptables -A INPUT -s 192.168.70.6 -d 192.168.70.5 -j ACCEPT
ats@serverA:~$ sudo iptables -A OUTPUT -s 192.168.70.5 -d 192.168.70.6 -j ACCEPT
```

```
@serverA:~$ sudo iptables -A FORWARD -p icmp --icmp-type 8 -j ACCEPT
@serverA:~$ sudo iptables -A FORWARD -p icmp --icmp-type 0 -j ACCEPT
```

Firstly, I let allow the two public network addresses (192.168.70.5 and 192.168.70.6)can get traffic in and out, then I check the WireShark, I found the packets have exchanged between 70.5/.6, but in the terminal the ping 192.168.80.111 just stuck up, so I thought I need to allow the ICMP protocol.

My iptables rules:

1. sudo ./firewall.sh (Same as Lab1 and scp for serverB)
2. sudo sysctl –w net.ipv4.ip_forward=1
3. sudo sysctl -p
4. check client A and client B can access Internet through NAT
5. sudo iptables –A INPUT –s 192.168.70.6 (serverB: 192.168.70.5) –d 192.168.70.5 (serverB: 192.168.70.6) –j ACCEPT
6. sudo iptables –A OUTPUT –s 192.168.70.5(serverB: 192.168.70.6) –d 192.168.70.6 (serverB: 192.168.70.5)
7. sudo iptables –A FORWARD –p icmp --icmp-type 8 –j ACCEPT
8. sudo iptables –A FORWARD –p icmp --icmp-type 0 –j ACCEPT

About my ipsec.conf I modified a little:

```
conn vpn-cert-tunnel-firewall
        keyexchange=ikev2
        authby=rsasig
        leftfirewall=no
        rightfirewall=no
        left=192.168.70.5
        right=192.168.70.6
        leftcert=192.168.70.5.cert.pem
        leftid="C=SE, ST=Blekinge, L=Karlskrona, O=ET2540, OU=bth, CN=192.168.70.5, E=921411446@qq.com"
        rightid="C=SE, ST=Blekinge, L=Karlskrona, O=bth, OU=bth, CN=192.168.70.6, E=921411446@qq.com"
        leftsubnet=192.168.60.0/24
        rightsubnet=192.168.80.0/24
        type=tunnel
        auto=route
```

r B [Running] - Oracle VM VirtualBox

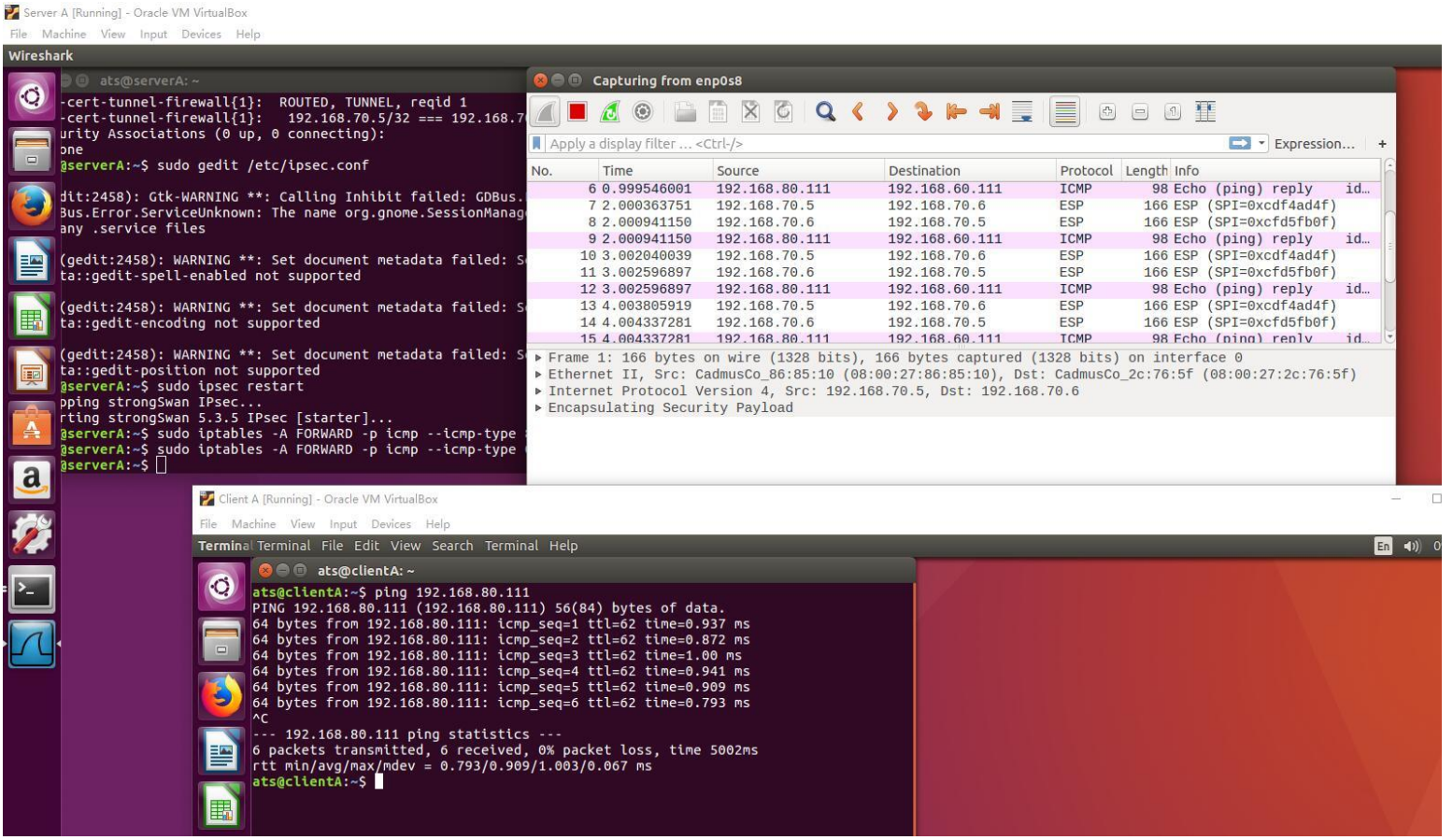
achine View Input Devices Help

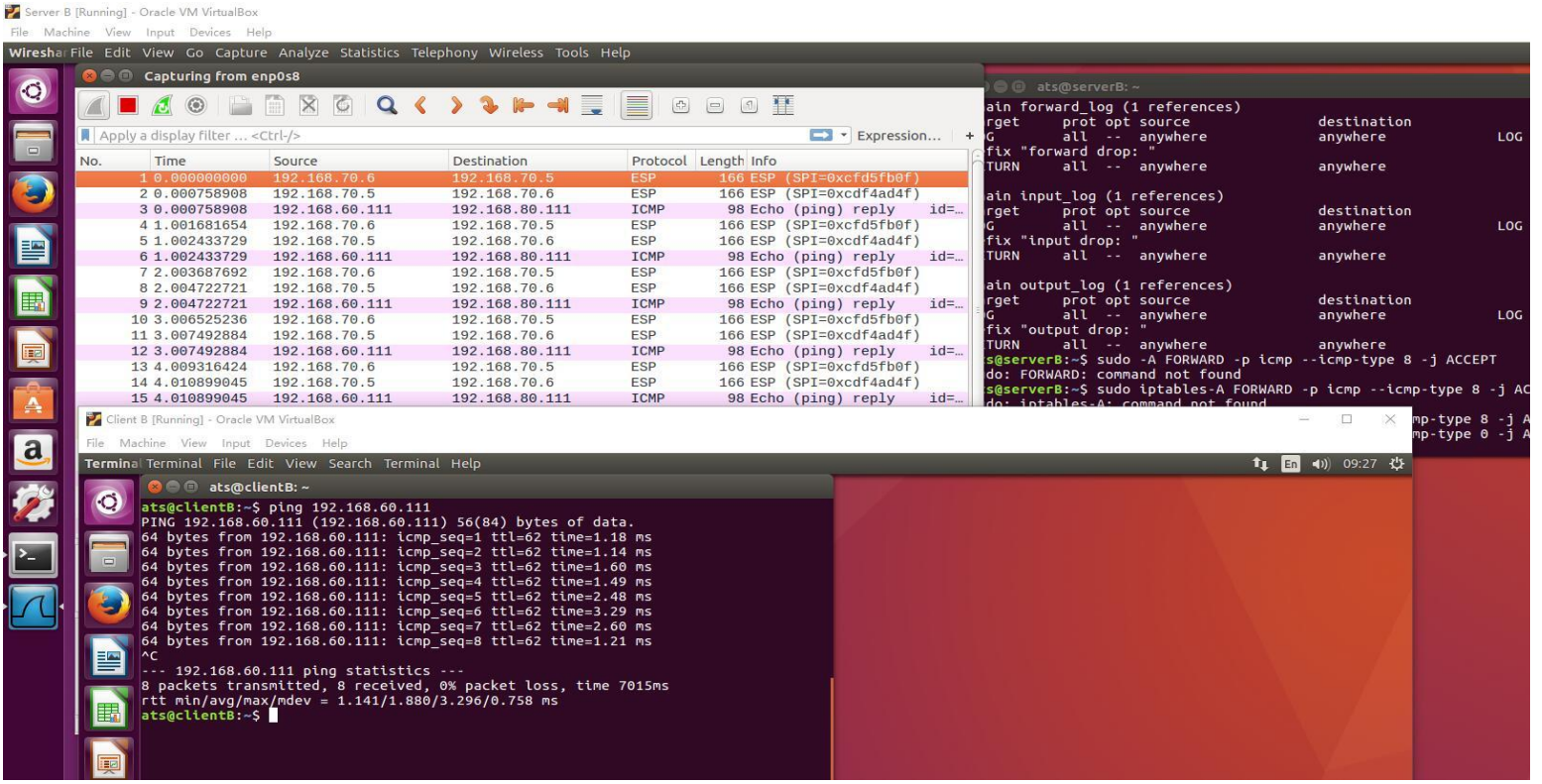
onf (/etc) - gedit

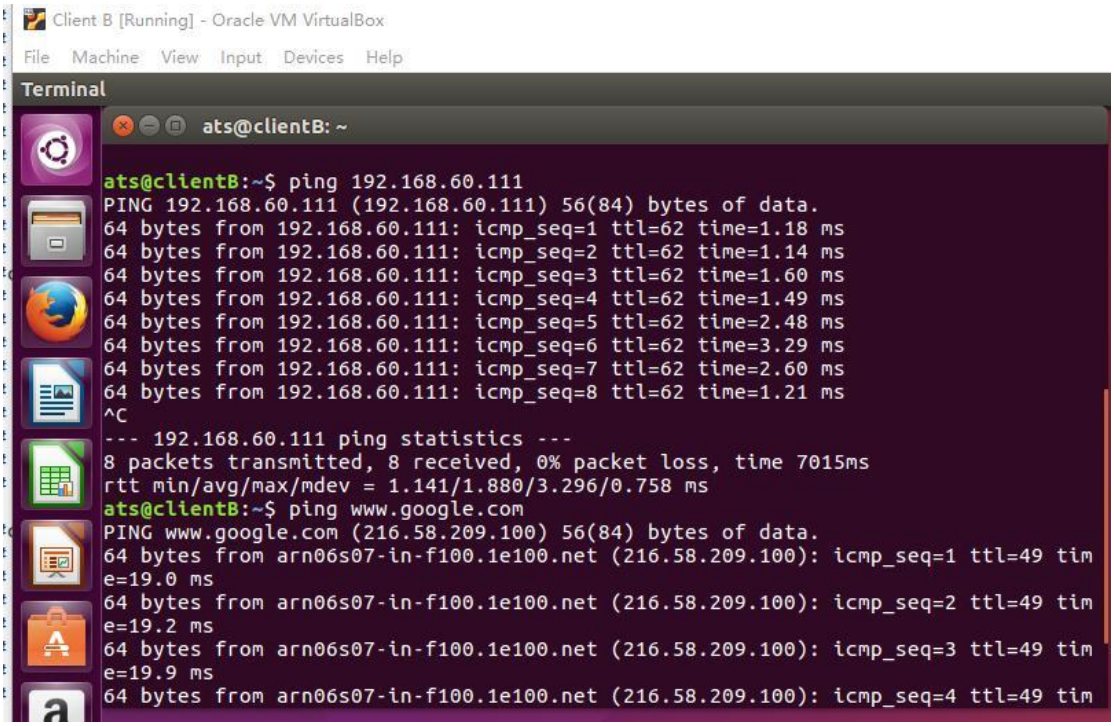
Open ipsec.conf /etc

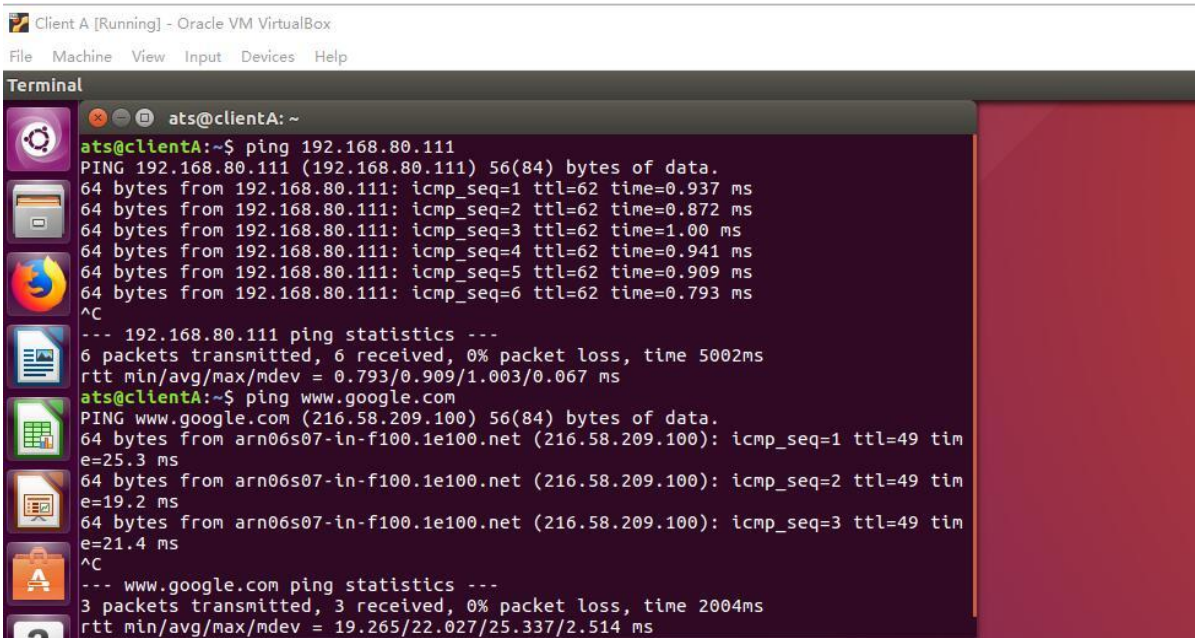
```
        # strictcrlpolicy=yes
        # uniqueids = no

# Add connections here.

# Sample VPN connections
conn vpn-cert-tunnel-firewall
        keyexchange=ikev2
        authby=rsasig
        leftfirewall=no
        rightfirewall=no
        left=192.168.70.6
        right=192.168.70.5
        leftsubnet=192.168.80.0/24
        rightsubnet=192.168.60.0/24
        leftcert=192.168.70.6.cert.pem
        leftid="C=SE, ST=Blekinge, L=Karlskrona, O=bth, OU=bth, CN=192.168.70.6, E=921411446@qq.com"
        rightid="C=SE, ST=Blekinge, L=Karlskrona, O=ET2540, OU=bth, CN=192.168.70.5, E=921411446@qq.com"
        type=tunnel
        auto=route
```

The following screenshots are as my evidences of under the DROP firewall rules condition, client A can communicate with client B over IPsec tunnel mode:


Server A [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Wireshark
ats@serverA: ~
-cert-tunnel-firewall{1}: ROUTED, TUNNEL, reqid 1
-cert-tunnel-firewall{1}: 192.168.70.5/32 === 192.168.7
urity Associations (0 up, 0 connecting):
one
@serverA:~$ sudo gedit /etc/ipsec.conf
dit:2458): Gtk-WARNING **: Calling Inhibit failed: GDBus.
Bus.Error.ServiceUnknown: The name org.gnome.SessionManag
any .service files
(gedit:2458): WARNING **: Set document metadata failed: S
ta::gedit-spell-enabled not supported
(gedit:2458): WARNING **: Set document metadata failed: S
ta::gedit-encoding not supported
(gedit:2458): WARNING **: Set document metadata failed: S
ta::gedit-position not supported
@serverA:~$ sudo ipsec restart
pping strongSwan IPsec...
rting strongSwan 5.3.5 IPsec [starter]...
@serverA:~$ sudo iptables -A FORWARD -p icmp --icmp-type
@serverA:~$ sudo iptables -A FORWARD -p icmp --icmp-type
@serverA:~$
Capturing from enp0s8
Apply a display filter ... <Ctrl-/>
Expression...
No. Time Source Destination Protocol Length Info
6 0.999546001 192.168.80.111 192.168.60.111 ICMP 98 Echo (ping) reply id...
7 2.000363751 192.168.70.5 192.168.70.6 ESP 166 ESP (SPI=0xcdf4ad4f)
8 2.000941150 192.168.70.6 192.168.70.5 ESP 166 ESP (SPI=0xcfd5fb0f)
9 2.000941150 192.168.80.111 192.168.60.111 ICMP 98 Echo (ping) reply id...
10 3.002040039 192.168.70.5 192.168.70.6 ESP 166 ESP (SPI=0xcdf4ad4f)
11 3.002596897 192.168.70.6 192.168.70.5 ESP 166 ESP (SPI=0xcfd5fb0f)
12 3.002596897 192.168.80.111 192.168.60.111 ICMP 98 Echo (ping) reply id...
13 4.003805919 192.168.70.5 192.168.70.6 ESP 166 ESP (SPI=0xcdf4ad4f)
14 4.004337281 192.168.70.6 192.168.70.5 ESP 166 ESP (SPI=0xcfd5fb0f)
15 4.004337281 192.168.80.111 192.168.60.111 ICMP 98 Echo (ping) reply id...
Frame 1: 166 bytes on wire (1328 bits), 166 bytes captured (1328 bits) on interface 0
Ethernet II, Src: CadmusCo_86:85:10 (08:00:27:86:85:10), Dst: CadmusCo_2c:76:5f (08:00:27:2c:76:5f)
Internet Protocol Version 4, Src: 192.168.70.5, Dst: 192.168.70.6
Encapsulating Security Payload
Client A [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Terminal Terminal File Edit View Search Terminal Help
ats@clientA: ~
ats@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=0.937 ms
64 bytes from 192.168.80.111: icmp_seq=2 ttl=62 time=0.872 ms
64 bytes from 192.168.80.111: icmp_seq=3 ttl=62 time=1.00 ms
64 bytes from 192.168.80.111: icmp_seq=4 ttl=62 time=0.941 ms
64 bytes from 192.168.80.111: icmp_seq=5 ttl=62 time=0.909 ms
64 bytes from 192.168.80.111: icmp_seq=6 ttl=62 time=0.793 ms
^C
--- 192.168.80.111 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5002ms
rtt min/avg/max/mdev = 0.793/0.909/1.003/0.067 ms
ats@clientA:~$


Server B [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Wireshark File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Capturing from enp0s8
Apply a display filter ... <Ctrl-/>
Expression...
No. Time Source Destination Protocol Length Info
1 0.000000000 192.168.70.6 192.168.70.5 ESP 166 ESP (SPI=0xcfd5fb0f)
2 0.000758908 192.168.70.5 192.168.70.6 ESP 166 ESP (SPI=0xcdf4ad4f)
3 0.000758908 192.168.60.111 192.168.80.111 ICMP 98 Echo (ping) reply id=...
4 1.001681654 192.168.70.6 192.168.70.5 ESP 166 ESP (SPI=0xcfd5fb0f)
5 1.002433729 192.168.70.5 192.168.70.6 ESP 166 ESP (SPI=0xcdf4ad4f)
6 1.002433729 192.168.60.111 192.168.80.111 ICMP 98 Echo (ping) reply id=...
7 2.003687692 192.168.70.6 192.168.70.5 ESP 166 ESP (SPI=0xcfd5fb0f)
8 2.004722721 192.168.70.5 192.168.70.6 ESP 166 ESP (SPI=0xcdf4ad4f)
9 2.004722721 192.168.60.111 192.168.80.111 ICMP 98 Echo (ping) reply id=...
10 3.006525236 192.168.70.6 192.168.70.5 ESP 166 ESP (SPI=0xcfd5fb0f)
11 3.007492884 192.168.70.5 192.168.70.6 ESP 166 ESP (SPI=0xcdf4ad4f)
12 3.007492884 192.168.60.111 192.168.80.111 ICMP 98 Echo (ping) reply id=...
13 4.009316424 192.168.70.6 192.168.70.5 ESP 166 ESP (SPI=0xcfd5fb0f)
14 4.010899045 192.168.70.5 192.168.70.6 ESP 166 ESP (SPI=0xcdf4ad4f)
15 4.010899045 192.168.60.111 192.168.80.111 ICMP 98 Echo (ping) reply id=...
ats@serverB: ~
ain forward_log (1 references)
rget prot opt source destination
G all -- anywhere anywhere LOG
fix "forward drop: "
TURN all -- anywhere anywhere
ain input_log (1 references)
rget prot opt source destination
G all -- anywhere anywhere LOG
fix "input drop: "
TURN all -- anywhere anywhere
ain output_log (1 references)
rget prot opt source destination
G all -- anywhere anywhere LOG
fix "output drop: "
TURN all -- anywhere anywhere
s@serverB:~$ sudo -A FORWARD -p icmp --icmp-type 8 -j ACCEPT
do: FORWARD: command not found
s@serverB:~$ sudo iptables-A FORWARD -p icmp --icmp-type 8 -j AC
do: iptables-A: command not found
Client B [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Terminal Terminal File Edit View Search Terminal Help
ats@clientB: ~
ats@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=1.18 ms
64 bytes from 192.168.60.111: icmp_seq=2 ttl=62 time=1.14 ms
64 bytes from 192.168.60.111: icmp_seq=3 ttl=62 time=1.60 ms
64 bytes from 192.168.60.111: icmp_seq=4 ttl=62 time=1.49 ms
64 bytes from 192.168.60.111: icmp_seq=5 ttl=62 time=2.48 ms
64 bytes from 192.168.60.111: icmp_seq=6 ttl=62 time=3.29 ms
64 bytes from 192.168.60.111: icmp_seq=7 ttl=62 time=2.60 ms
64 bytes from 192.168.60.111: icmp_seq=8 ttl=62 time=1.21 ms
^C
--- 192.168.60.111 ping statistics ---
8 packets transmitted, 8 received, 0% packet loss, time 7015ms
rtt min/avg/max/mdev = 1.141/1.880/3.296/0.758 ms
ats@clientB:~$





That is the my serverA and serverB ipsec.conf :
Contains all my conn configurations:

Server B [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

ipsec.conf (/etc) - gedit

Open ipsec.conf /etc

```
        # strictcrlpolicy=yes
        # uniqueids = no

# Add connections here.

# Sample VPN connections
conn vpn-cert-tunnel-firewall
      keyexchange=ikev2
      authby=rsasig
      leftfirewall=no
      rightfirewall=no
      left=192.168.70.6
      right=192.168.70.5
      leftsubnet=192.168.80.0/24
      rightsubnet=192.168.60.0/24
      leftcert=192.168.70.6.cert.pem
      leftid="C=SE, ST=Blekinge, L=Karlskrona, O=bth, OU=bth, CN=192.168.70.6, E=921411446@qq.com"
      rightid="C=SE, ST=Blekinge, L=Karlskrona, O=ET2540, OU=bth, CN=192.168.70.5, E=921411446@qq.com"
      type=tunnel
      auto=route
#conn vpn-cert
#       keyexchange=ikev2
#       authby=rsasig
#       left=192.168.70.6
#       right=192.168.70.5
#       leftcert=192.168.70.6.cert.pem
#       leftid="C=SE, ST=Blekinge, L=Karlskrona, O=bth, OU=bth, CN=192.168.70.6, E=921411446@qq.com"
#       rightid="C=SE, ST=Blekinge, L=Karlskrona, O=ET2540, OU=bth, CN=192.168.70.5, E=921411446@qq.com"
#       leftsubnet=192.168.80.0/24
#       rightsubnet=192.168.60.0/24
#       type=tunnel
#       auto=route
#conn vpn-cert
#       keyexchange=ikev2
#       authby=rsasig
#       left=192.168.70.6
#       right=192.168.70.5
#       leftcert=192.168.70.6.cert.pem
#       leftid="C=SE, ST=Blekinge, L=Karlskrona, O=bth, OU=bth, CN=192.168.70.6, E=921411446@qq.com"
#       rightid="C=SE, ST=Blekinge, L=Karlskrona, O=ET2540, OU=bth, CN=192.168.70.5, E=921411446@qq.com"
#       type=transport
#       auto=route

#conn h2h-psk
#       keyexchange=ikev2
#       leftauth=psk
#       rightauth=psk
#       left=192.168.70.6
#       right=192.168.70.5
#       type=transport
#       auto=route
```

Server A [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

ipsec.conf (/etc) - gedit

Open ipsec.conf /etc

```
        # strictcrlpolicy=yes
        # uniqueids = no

# Add connections here.

# Sample VPN connections
conn vpn-cert-tunnel-firewall
      keyexchange=ikev2
      authby=rsasig
      leftfirewall=no
      rightfirewall=no
      left=192.168.70.5
      right=192.168.70.6
      leftsubnet=192.168.60.0/24
      rightsubnet=192.168.80.0/24
      leftcert=192.168.70.5.cert.pem
      leftid="C=SE, ST=Blekinge, L=Karlskrona, O=ET2540, OU=bth, CN=192.168.70.5, E=921411446@qq.com"
      rightid="C=SE, ST=Blekinge, L=Karlskrona, O=bth, OU=bth, CN=192.168.70.6, E=921411446@qq.com"
      type=tunnel
      auto=route

#conn vpn-cert
#       keyexchange=ikev2
#       authby=rsasig
#       left=192.168.70.5
#       right=192.168.70.6
#       leftcert=192.168.70.5.cert.pem
#       leftid="C=SE, ST=Blekinge, L=Karlskrona, O=ET2540, OU=bth, CN=192.168.70.5, E=921411446@qq.com"
#       rightid="C=SE, ST=Blekinge, L=Karlskrona, O=bth, OU=bth, CN=192.168.70.6, E=921411446@qq.com"
#       leftsubnet=192.168.60.0/24
#       rightsubnet=192.168.80.0/24
#       type=tunnel
#       auto=route
#conn vpn-cert
#       keyexchange=ikev2
#       authby=rsasig
#       left=192.168.70.5
#       right=192.168.70.6
#       leftcert=192.168.70.5.cert.pem
#       leftid="C=SE, ST=Blekinge, L=Karlskrona, O=ET2540, OU=bth, CN=192.168.70.5, E=921411446@qq.com"
#       rightid="C=SE, ST=Blekinge, L=Karlskrona, O=bth, OU=bth, CN=192.168.70.6, E=921411446@qq.com"
#       type=transport
#       auto=route
#conn h2h-psk
#       keyexchange=ikev2
#       leftauth=psk
#       rightauth=psk
#       left=192.168.70.5
#       right=192.168.70.6
#       type=transport
#      auto=route
```