Virtual Private Network (VPN) Lab

Pavan Haravu Ramesh, Jeevan Venkataramana,
Vishwanath Shivananad Kaddi
Computer Engineering Department, College of Engineering
San Jose State University, San Jose, CA 94303

Abstract

A Virtual Private Network (VPN) is used for creating a private scope of computer communications or providing a secure extension of a private network into an insecure network such as internet. VPN is a widely used security technology. VPN can be built upon IPsec or Secure Socket Layer (SSL). These are two fundamentally different approaches for building VPN. This report provides details about implementation of SSL-based VPNs often referred to as SSL VPNs. Aim is to implement a simple SSL VPN for Ubuntu.

1. Introduction

An SSL VPN (Secure Sockets Layer Virtual Private Network) is a form of VPN that can be used with a standard web browser. In contrast to the traditional internet Protocol Security (IPsec) VPN, an SSL VPN does not require the installation of specialized client software on the end user's computer. SSL VPN is used to give remote users access to web applications, client/server applications and internal network connections. A Virtual Private Network (VPN) provides a secure communications mechanism for data and other information transmitted between two endpoints. An SSL VPN consists of one or more VPN devices to which users connects by using his web browser. The traffic between the web browser and SSL VPN device is encrypted with the SSL protocol.

2. Methodology

A Virtual Private Network (VPN) is used for creating a private scope of computer communications or providing a secure extension of a private network into an insecure network such as internet. VPN is a widely used security technology. VPN can be built upon IPsec or Secure Socket Layer (SSL). These are two fundamentally different approaches for building VPN. This report provides details about implementation of SSL-based VPNs often referred to as SSL VPNs. Aim is to implement a simple SSL VPN for Ubuntu. An SSL VPN (Secure Sockets Layer Virtual Private Network) is a form of VPN that can be used with a standard web browser. In contrast to the traditional internet Protocol Security (IPsec) VPN, an SSL VPN does not require the installation of specialized client software on the end user's computer. SSL VPN is used to give remote users access to web applications, client/server applications and internal network connections. A Virtual Private Network (VPN) provides a secure communications mechanism for data and other information transmitted between two endpoints. An SSL VPN consists of one or more VPN devices to which users connects by using his web browser The traffic between the web browser and SSL VPN device is encrypted with the SSL protocol.

2.1. Objectives and Technical Challenges

The learning objective is to master the network and security technologies underlying SSL VPNs.

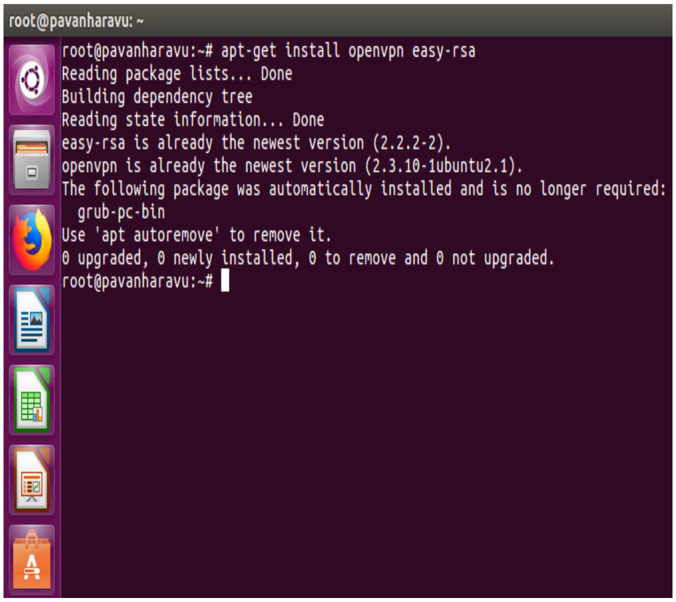
Other objectives are listed below:

1. Understanding the concepts and theory of Virtual Private Network
2. Create a Host-to-Host Tunnel using TUN/TAP
3. Create a Host-to-gateway Tunnel
4. Create a Gateway-to-Gateway Tunnel
5. Create a Virtual Private Network (VPN)
6. Authenticating VPN client with VPN server
7. Generate a Client Certificate

Establishing secure tun0 interface was a challenge. Also understanding Linux commands and working completely on terminal was a challenge.

3. Lab Environment

OpenSSL package is used. The package includes the header files, libraries and commands. OpenVPN is installed onto the server. OpenVPN is available in Ubuntu's default repositories. Also easy-rsa package is installed, which helps in setting up internal CA (certificate authority) that can be used with the VPN.

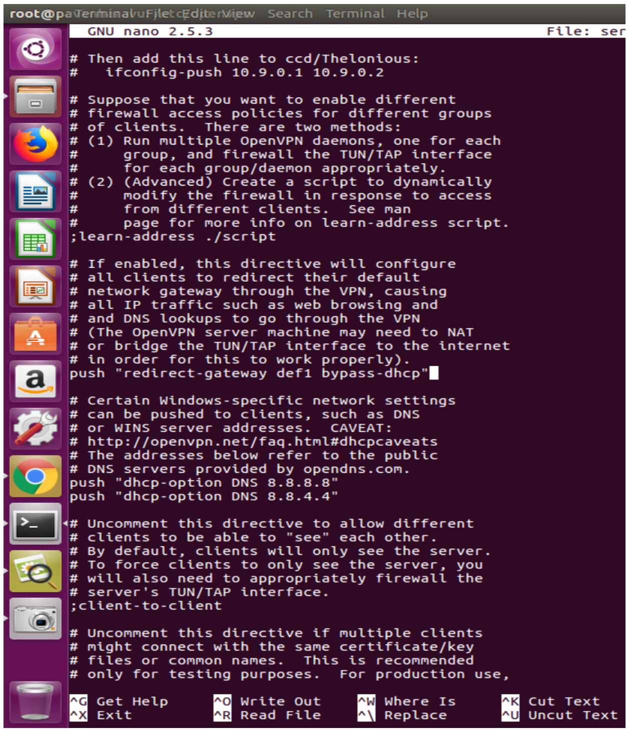


```
root@pavanharavu:~# apt-get install openvpn easy-rsa
Reading package lists... Done
Building dependency tree
Reading state information... Done
easy-rsa is already the newest version (2.2.2-2).
openvpn is already the newest version (2.3.10-1ubuntu2.1).
The following package was automatically installed and is no longer required:
  grub-pc-bin
Use 'apt autoremove' to remove it.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
root@pavanharavu:~# 
```

Figure 1. openvpn and easy-rsa package installation

3.1. Push DNS Changes through VPN

To force connections to use tunnel and to use VPN to route all traffic through the VPN, push DNS settings to the client computers.

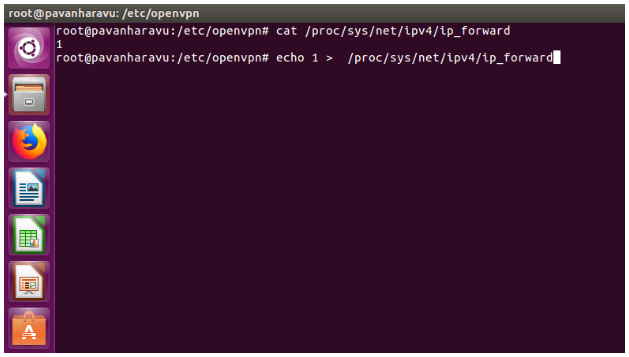


```
  GNU nano 2.5.3                                                  File: ser

# Then add this line to ccd/Thelonious:
#   ifconfig-push 10.9.0.1 10.9.0.2

# Suppose that you want to enable different
# firewall access policies for different groups
# of clients.  There are two methods:
# (1) Run multiple OpenVPN daemons, one for each
#     group, and firewall the TUN/TAP interface
#     for each group/daemon appropriately.
# (2) (Advanced) Create a script to dynamically
#     modify the firewall in response to access
#     from different clients.  See man
#     page for more info on learn-address script.
;learn-address ./script

# If enabled, this directive will configure
# all clients to redirect their default
# network gateway through the VPN, causing
# all IP traffic such as web browsing and
# and DNS lookups to go through the VPN
# (The OpenVPN server machine may need to NAT
# or bridge the TUN/TAP interface to the internet
# in order for this to work properly).
push "redirect-gateway def1 bypass-dhcp"

# Certain Windows-specific network settings
# can be pushed to clients, such as DNS
# or WINS server addresses.  CAVEAT:
# http://openvpn.net/faq.html#dhcpcaveats
# The addresses below refer to the public
# DNS servers provided by opendns.com.
push "dhcp-option DNS 8.8.8.8"
push "dhcp-option DNS 8.8.4.4"

# Uncomment this directive to allow different
# clients to be able to "see" each other.
# By default, clients will only see the server.
# To force clients to only see the server, you
# will also need to appropriately firewall the
# server's TUN/TAP interface.
;client-to-client

# Uncomment this directive if multiple clients
# might connect with the same certificate/key
# files or common names.  This is recommended
# only for testing purposes.  For production use,

^G Get Help     ^O Write Out    ^W Where Is     ^K Cut Text
^X Exit         ^R Read File    ^\ Replace      ^U Uncut Text
```

Figure 2. DNS changes to redirect all traffic through VPN

3.2. Setup Firewall

Basic firewall is built by enabling IP forwarding and echoing 1 into the ip_forward.

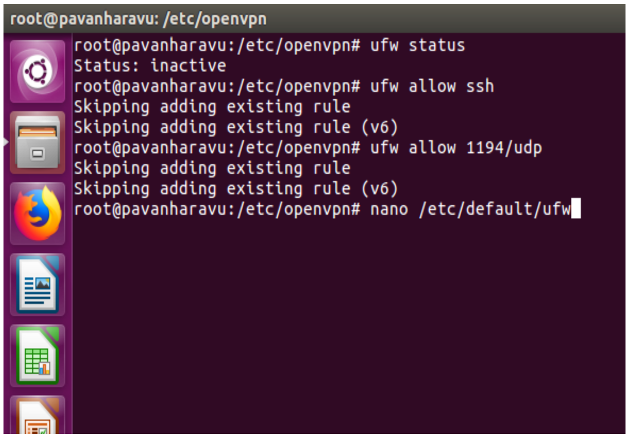


```
root@pavanharavu:/etc/openvpn# cat /proc/sys/net/ipv4/ip_forward
1
root@pavanharavu:/etc/openvpn# echo 1 >  /proc/sys/net/ipv4/ip_forward
```

Figure 3. Firewall setup

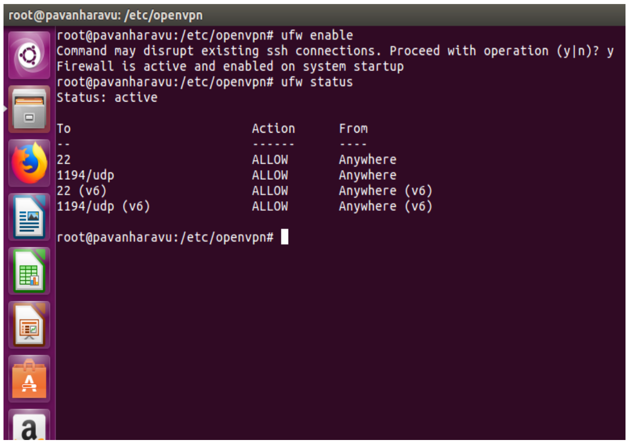
3.3. Configuration of firewall

Different applications can register their profiles with UFW upon installation. These profiles allow UFW to manage these applications by name. OpenSSH, the service will connect to server and has a profile registered with UFW. To make sure that the firewall allows SSH connections to log back in next time, allow OpenSSH, enable ufw and select UDP as the underlying protocol and choose port 1194 (can be any port).



```
root@pavanharavu:/etc/openvpn# ufw status
Status: inactive
root@pavanharavu:/etc/openvpn# ufw allow ssh
Skipping adding existing rule
Skipping adding existing rule (v6)
root@pavanharavu:/etc/openvpn# ufw allow 1194/udp
Skipping adding existing rule
Skipping adding existing rule (v6)
root@pavanharavu:/etc/openvpn# nano /etc/default/ufw
```

Figure 4. Setting upd port to 1194



```
root@pavanharavu:/etc/openvpn# ufw enable
Command may disrupt existing ssh connections. Proceed with operation (y|n)? y
Firewall is active and enabled on system startup
root@pavanharavu:/etc/openvpn# ufw status
Status: active

To                         Action      From
--                         ------      ----
22                         ALLOW       Anywhere
1194/udp                   ALLOW       Anywhere
22 (v6)                    ALLOW       Anywhere (v6)
1194/udp (v6)              ALLOW       Anywhere (v6)

root@pavanharavu:/etc/openvpn# 
```

Figure 5. ufw status

3.4. Enable NAT and IP masquerading for clients

IP Masquerade feature allows other internal computers connected to server to also reach internet. MASQS allows a set of machines to invisibly access the internet. To other machines on the internet, the outgoing traffic will appear to be from the IP MASQ Linux server itself. In addition to the added functionality, IP Masquerade provides the foundation to create a heavily secured networking environment.

Some NAT rules are added as shown below:

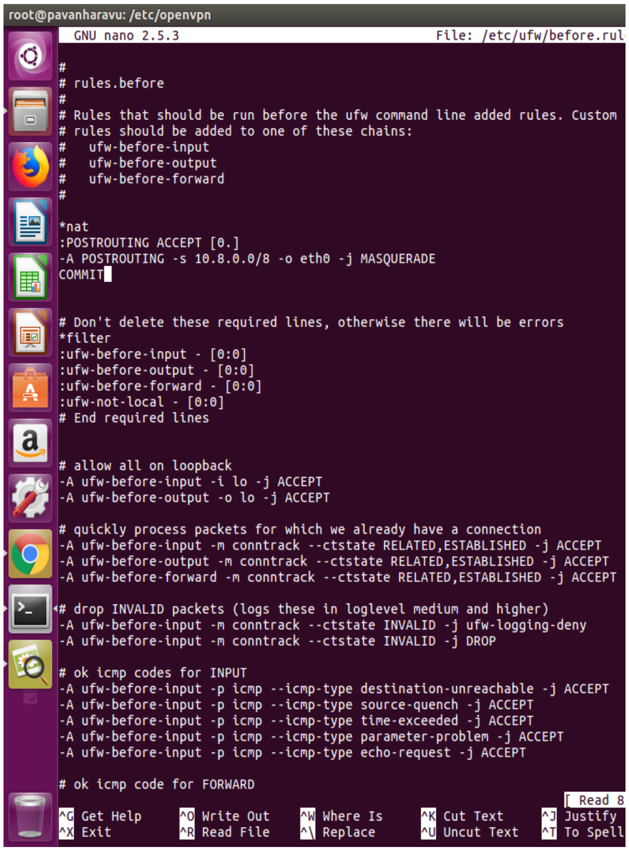
*nat

:POSTROUTING ACCEPT [0.0]

-A POSTROUTING -s 10.8.0.0/8 -o -j MASQUERADE

COMMIT

The rules uses the NAT packet and specifies the built in POSTROUTING chain for NAT (-A POSTROUTING) on the firewall's external networking device. The -j MASQUERADE target is specified to mask the private IP address of a node with the external IP address of the frewall/gateway. The incoming HTTP requests are forwarded to the listed destination IP address of 10.8.0.0/8 (tun0 interface). Once VPN service is up and running tun0 is created with the IP address of 10.8.0.0

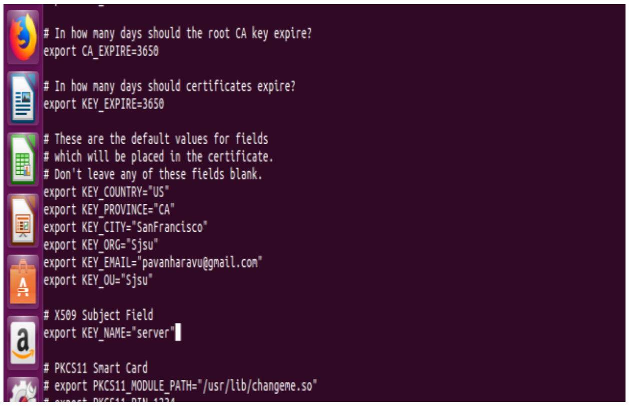


```
root@pavanharavu: /etc/openvpn
  GNU nano 2.5.3                                        File: /etc/ufw/before.rul

#
# rules.before
#
# Rules that should be run before the ufw command line added rules. Custom
# rules should be added to one of these chains:
#   ufw-before-input
#   ufw-before-output
#   ufw-before-forward
#

*nat
:POSTROUTING ACCEPT [0.]
-A POSTROUTING -s 10.8.0.0/8 -o eth0 -j MASQUERADE
COMMIT


# Don't delete these required lines, otherwise there will be errors
*filter
:ufw-before-input - [0:0]
:ufw-before-output - [0:0]
:ufw-before-forward - [0:0]
:ufw-not-local - [0:0]
# End required lines


# allow all on loopback
-A ufw-before-input -i lo -j ACCEPT
-A ufw-before-output -o lo -j ACCEPT

# quickly process packets for which we already have a connection
-A ufw-before-input -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
-A ufw-before-output -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
-A ufw-before-forward -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT

# drop INVALID packets (logs these in loglevel medium and higher)
-A ufw-before-input -m conntrack --ctstate INVALID -j ufw-logging-deny
-A ufw-before-input -m conntrack --ctstate INVALID -j DROP

# ok icmp codes for INPUT
-A ufw-before-input -p icmp --icmp-type destination-unreachable -j ACCEPT
-A ufw-before-input -p icmp --icmp-type source-quench -j ACCEPT
-A ufw-before-input -p icmp --icmp-type time-exceeded -j ACCEPT
-A ufw-before-input -p icmp --icmp-type parameter-problem -j ACCEPT
-A ufw-before-input -p icmp --icmp-type echo-request -j ACCEPT

# ok icmp code for FORWARD
                                                                      [ Read 8
^G Get Help     ^O Write Out    ^W Where Is     ^K Cut Text     ^J Justify
^X Exit         ^R Read File    ^\ Replace      ^U Uncut Text   ^T To Spell
```

Figure 6. updating NAT rules

3.5. Configure CA variables

By making changes in the vars files, CA values are configured. Vars file can be adjusted to determine how certificates will be created.

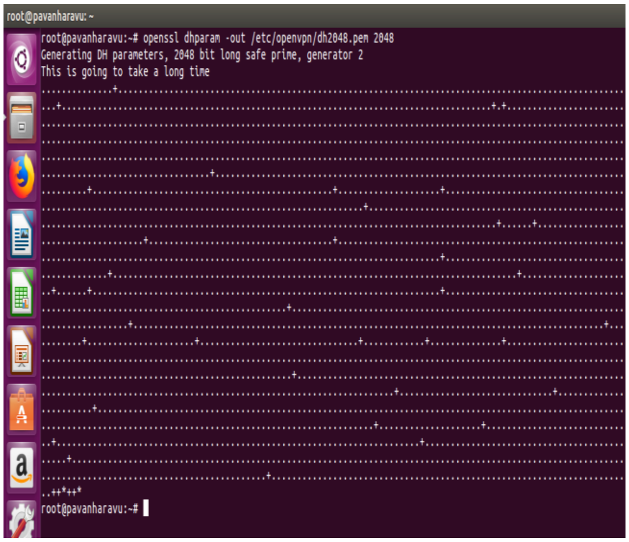


```
# In how many days should the root CA key expire?
export CA_EXPIRE=3650

# In how many days should certificates expire?
export KEY_EXPIRE=3650

# These are the default values for fields
# which will be placed in the certificate.
# Don't leave any of these fields blank.
export KEY_COUNTRY="US"
export KEY_PROVINCE="CA"
export KEY_CITY="SanFrancisco"
export KEY_ORG="Sjsu"
export KEY_EMAIL="pavanharavu@gmail.com"
export KEY_OU="Sjsu"

# X509 Subject Field
export KEY_NAME="server"

# PKCS11 Smart Card
# export PKCS11_MODULE_PATH="/usr/lib/changeme.so"
```

Figure 7. variable file

3.6. Generate Diffie Hellman Key

Before a VPN is established, the VPN client must authenticate the VPN server, making sure that the server is not a fraudulent one. On the other hand, the VPN server must authenticate the client (i.e. user), making sure that the user has the permission to create such a VPN tunnel. After the authentication is done, the client and the server will agree upon a session key for the VPN tunnel. This session key is only known to the client and the server. The process of deriving this session key is called key exchange.

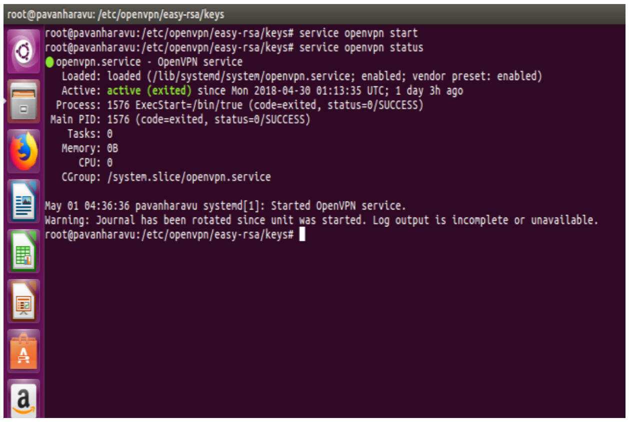


```
root@pavanharavu: ~
root@pavanharavu:~# openssl dhparam -out /etc/openvpn/dh2048.pem 2048
Generating DH parameters, 2048 bit long safe prime, generator 2
This is going to take a long time
..........+.................................................................
..+.......................................................................+.+.
.............................................................................
.............................................................................
........................+....................................................
......+..................................+.................+.................
.............................................+...............................
...........................................................+.....+...........
..............+..............................................................
.............................................................................
......................................................+......................
..........+.........................................................+........
..+.....+..............................................+.....................
....................................+........................................
............+..................................................................+..
.......+...............+.....................................................
.............................................................................
....................................+........................................
.........................................................+...................+....
........+....................................................................
.............................................................................
..............................................................+..............
..+........................................................+.................
....+........................................................................
.............................+...............................................
..++*++*
root@pavanharavu:~#
```

Figure 8. Diffie-Hellman key

3.7. Initiate VPN service

VPN service can be started by inputting the command service start VPN.

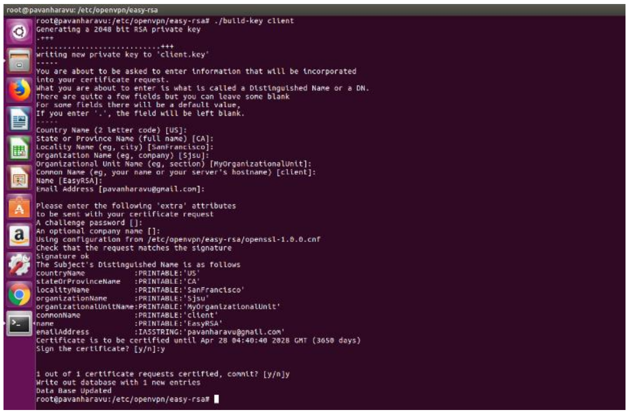


```
root@pavanharavu: /etc/openvpn/easy-rsa/keys
root@pavanharavu:/etc/openvpn/easy-rsa/keys# service openvpn start
root@pavanharavu:/etc/openvpn/easy-rsa/keys# service openvpn status
● openvpn.service - OpenVPN service
   Loaded: loaded (/lib/systemd/system/openvpn.service; enabled; vendor preset: enabled)
   Active: active (exited) since Mon 2018-04-30 01:13:35 UTC; 1 day 3h ago
  Process: 1576 ExecStart=/bin/true (code=exited, status=0/SUCCESS)
 Main PID: 1576 (code=exited, status=0/SUCCESS)
    Tasks: 0
   Memory: 0B
      CPU: 0
   CGroup: /system.slice/openvpn.service

May 01 04:36:36 pavanharavu systemd[1]: Started OpenVPN service.
Warning: Journal has been rotated since unit was started. Log output is incomplete or unavailable.
root@pavanharavu:/etc/openvpn/easy-rsa/keys#
```

Figure 9. VPN service

3.8. Build certificate authority

To initiate the process of creating the root certificate authority key and certificate input the command ./build-ca in vars directory. All values should be populated automatically.

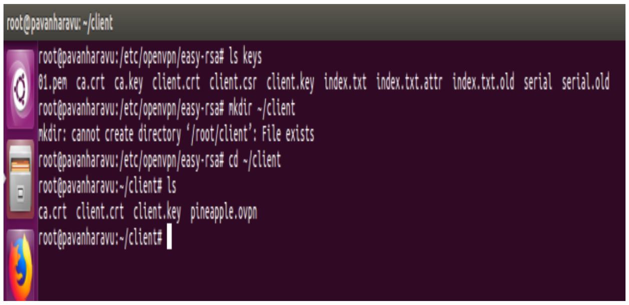


```
root@pavanharavu: /etc/openvpn/easy-rsa
root@pavanharavu:/etc/openvpn/easy-rsa# ./build-key client
Generating a 2048 bit RSA private key
.+++
.............................+++
writing new private key to 'client.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [US]:
State or Province Name (full name) [CA]:
Locality Name (eg, city) [SanFrancisco]:
Organization Name (eg, company) [Sjsu]:
Organizational Unit Name (eg, section) [MyOrganizationalUnit]:
Common Name (eg, your name or your server's hostname) [client]:
Name [EasyRSA]:
Email Address [pavanharavu@gmail.com]:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
Using configuration from /etc/openvpn/easy-rsa/openssl-1.0.0.cnf
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
countryName           :PRINTABLE:'US'
stateOrProvinceName   :PRINTABLE:'CA'
localityName          :PRINTABLE:'SanFrancisco'
organizationName      :PRINTABLE:'Sjsu'
organizationalUnitName:PRINTABLE:'MyOrganizationalUnit'
commonName            :PRINTABLE:'client'
name                  :PRINTABLE:'EasyRSA'
emailAddress          :IA5STRING:'pavanharavu@gmail.com'
Certificate is to be certified until Apr 28 04:40:40 2028 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
root@pavanharavu:/etc/openvpn/easy-rsa#
```

Figure 10. Certificate Authority

3.9. Creating Certificates

Client certificate is created.

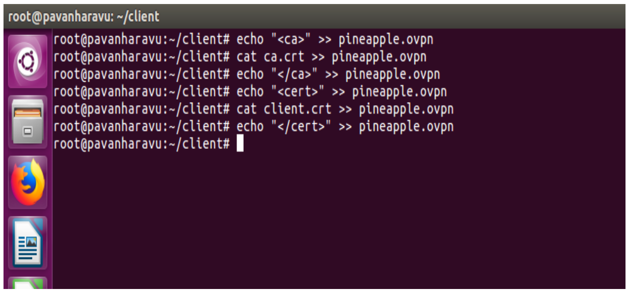


```
root@pavanharavu:/etc/openvpn/easy-rsa# ls keys
01.pem  ca.crt  ca.key  client.crt  client.csr  client.key  index.txt  index.txt.attr  index.txt.old  serial  serial.old
root@pavanharavu:/etc/openvpn/easy-rsa# mkdir ~/client
mkdir: cannot create directory '/root/client': File exists
root@pavanharavu:/etc/openvpn/easy-rsa# cd ~/client
root@pavanharavu:~/client# ls
ca.crt  client.crt  client.key  pineapple.ovpn
root@pavanharavu:~/client#
```

Figure 11. client certificate

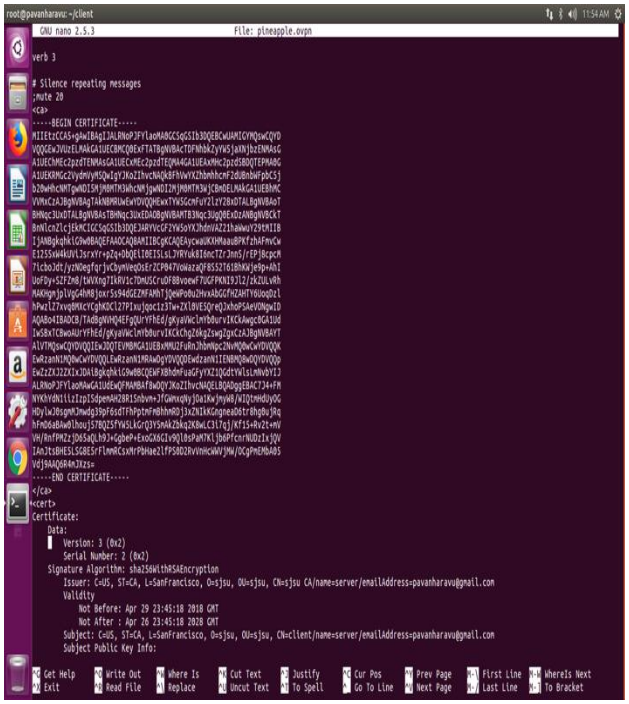
4.0. Unified file pineapple.ovpn

Move all certificates and keys to pineapple.ovpn file. Client only needs pineapple.ovpn to connect to server and start using VPN service.



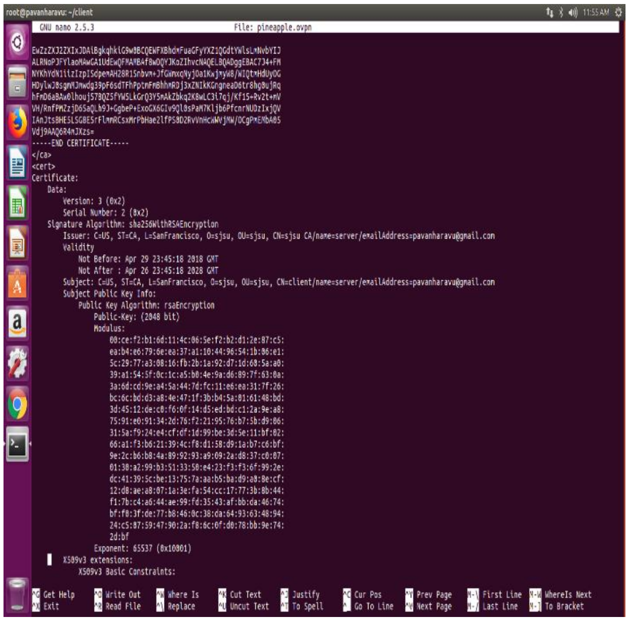
```
root@pavanharavu:~/client# echo "<ca>" >> pineapple.ovpn
root@pavanharavu:~/client# cat ca.crt >> pineapple.ovpn
root@pavanharavu:~/client# echo "</ca>" >> pineapple.ovpn
root@pavanharavu:~/client# echo "<cert>" >> pineapple.ovpn
root@pavanharavu:~/client# cat client.crt >> pineapple.ovpn
root@pavanharavu:~/client# echo "</cert>" >> pineapple.ovpn
root@pavanharavu:~/client#
```

Figure 12. certificates and keys moved to pineapple.ovpn



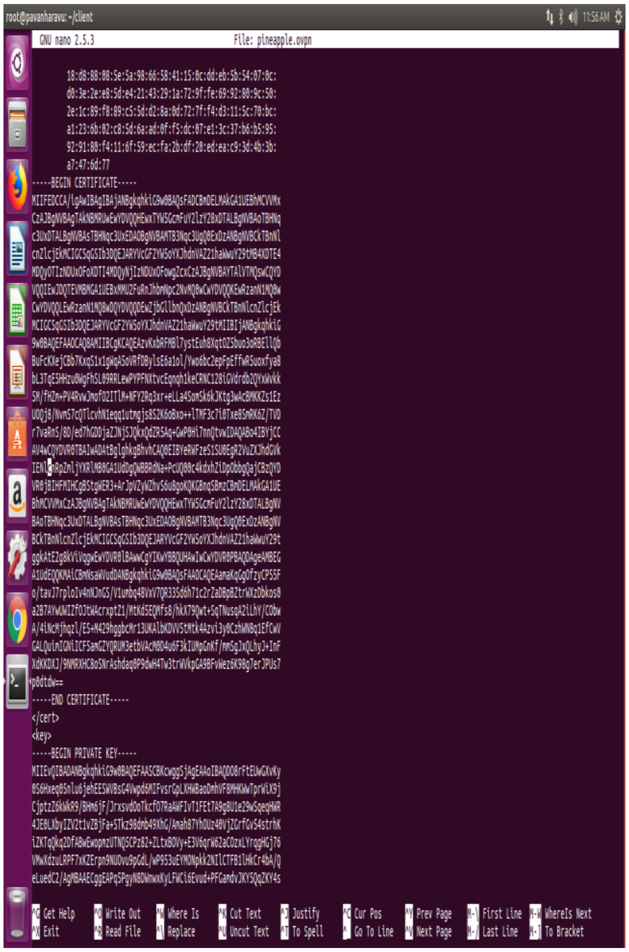
```
verb 3

# Silence repeating messages
;mute 20
<ca>
-----BEGIN CERTIFICATE-----
MIIEtzCCA5+gAwIBAgIJALRNoPJFYlaoMA0GCSqGSIb3DQEBCwUAMIGYMQswCQYD
VQQGEwJVUzELMAkGA1UECBMCQ0ExFTATBgNVBAcTDFNhbkZyYW5jaXNjbzENMAsG
A1UEChMEc2pzdTENMAsGA1UECxMEc2pzdTEQMA4GA1UEAxMHc2pzdSBDQTEPMA0G
A1UEKRMGc2VydmVyMSQwIgYJKoZIhvcNAQkBFhVwYXZhbmhhcmF2dUBnbWFpbC5j
b20wHhcNMTgwNDI5MjM0MTM3WhcNMjgwNDI2MjM0MTM3WjCBmDELMAkGA1UEBhMC
VVMxCzAJBgNVBAgTAkNBMRUwEwYDVQQHEwxTYW5GcmFuY2lzY28xDTALBgNVBAoT
BHNqc3UxDTALBgNVBAsTBHNqc3UxEDAOBgNVBAMTB3Nqc3UgQ0ExDzANBgNVBCkT
BnNlcnZlcjEkMCIGCSqGSIb3DQEJARYVcGF2YW5oYXJhdnVAZ21haWwuY29tMIIB
IjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAycwaUKXHMaauBPKfzhAFmvCw
E125SxW4kUVlJsrxYr+pZq+DbQEiI0EISLsLJYRYuk8I6ncTZrJnnS/rEPj8cpcM
7icboJdt/yzNOegfqrjvCbymVeqOsErZCP047VoWazaQF8S52T61BhKWje9p+AhI
UoFDy+SZFZm8/tWVXng7IkRV1c7DmUSCruDF8BvoewF7UGFPKNI9Jl2/zkZULvRh
MAKHgmjplVgG4hM8joxr5s94dGEZMFAMhTjQeWPo0u2HvxAbGGfHZAHTY6UoqDzl
hPwzlZ7xvq0MXcYCghKDCl27PIxujqoc1z3Tw+ZXl0VESQreQJxhoPSAeVONgwID
AQABo4IBADCB/TAdBgNVHQ4EFgQUrYFhEd/gKyaVWclmYb0urvIKCkAwgc0GA1Ud
IwSBxTCBwoAUrYFhEd/gKyaVWclmYb0urvIKCkChgZ6kgZswgZgxCzAJBgNVBAYT
AlVTMQswCQYDVQQIEwJDQTEVMBMGA1UEBxMMU2FuRnJhbmNpc2NvMQ0wCwYDVQQK
EwRzanN1MQ0wCwYDVQQLEwRzanN1MRAwDgYDVQQDEwdzanN1IENBMQ8wDQYDVQQp
EwZzZXJ2ZXIxJDAiBgkqhkiG9w0BCQEWFXBhdmFuaGFyYXZ1QGdtYWlsLmNvbYIJ
ALRNoPJFYlaoMAwGA1UdEwQFMAMBAf8wDQYJKoZIhvcNAQELBQADggEBAC7J4+FM
NYKhYdN1iizIzpISdpemAH28R1Snbvm+JfGWmxqNyjOa1KwjmyW8/WIQtmHdUyOG
HDylwJ0sgmMJmwdg39pF6sdTFhPptmFmBhhmRDj3xZNIkKGngneaD6tr8hg0ujRq
hFmD6aBAw0lhouj57BQZ5fYW5LkGrQ3Y5mAkZbkq2K8wLC3i7qj/Kf15+Rv2t+mV
VH/RnfPMZzjD65aQLh9J+GgbeP+ExoGX6GIv9Ql0sPaM7Kljb6PfcnrNUDzIxjQV
IAnJtsBHESLSG8ESrFlmmRCsxMrPbHae2lfPS0D2RvVnHcWWVjMW/OCgPmEMbA05
Vdj9AAQ6R4mJXzs=
-----END CERTIFICATE-----
</ca>
<cert>
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 2 (0x2)
    Signature Algorithm: sha256WithRSAEncryption
        Issuer: C=US, ST=CA, L=SanFrancisco, O=sjsu, OU=sjsu, CN=sjsu CA/name=server/emailAddress=pavanharavu@gmail.com
        Validity
            Not Before: Apr 29 23:45:18 2018 GMT
            Not After : Apr 26 23:45:18 2028 GMT
        Subject: C=US, ST=CA, L=SanFrancisco, O=sjsu, OU=sjsu, CN=client/name=server/emailAddress=pavanharavu@gmail.com
        Subject Public Key Info:
```

Figure 13. Pineapple.ovpn file



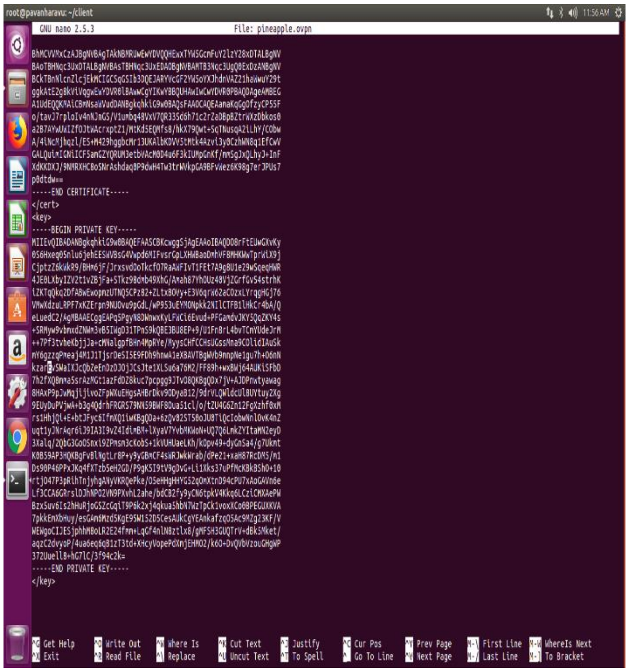
```
EwZzZXJ2ZXIxJDAiBgkqhkiG9w0BCQEWFXBhdmFuaGFyYXZ1QGdtYWlsLmNvbYIJ
ALRNoPJFYlaoMAwGA1UdEwQFMAMBAf8wDQYJKoZIhvcNAQELBQADggEBAC7J4+FM
NYKhYdN1iizIzpISdpemAH28R1Snbvm+JfGWmxqNyjOa1KwjmyW8/WIQtmHdUyOG
HDylwJ0sgmMJmwdg39pF6sdTFhPptmFmBhhmRDj3xZNIkKGngneaD6tr8hg0ujRq
hFmD6aBAw0lhouj57BQZ5fYW5LkGrQ3Y5mAkZbkq2K8wLC3i7qj/Kf15+Rv2t+mV
VH/RnfPMZzjD65aQLh9J+GgbeP+ExoGX6GIv9Ql0sPaM7Kljb6PfcnrNUDzIxjQV
IAnJtsBHESLSG8ESrFlmmRCsxMrPbHae2lfPS0D2RvVnHcWWVjMW/OCgPmEMbA05
Vdj9AAQ6R4mJXzs=
-----END CERTIFICATE-----
</ca>
<cert>
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 2 (0x2)
    Signature Algorithm: sha256WithRSAEncryption
        Issuer: C=US, ST=CA, L=SanFrancisco, O=sjsu, OU=sjsu, CN=sjsu CA/name=server/emailAddress=pavanharavu@gmail.com
        Validity
            Not Before: Apr 29 23:45:18 2018 GMT
            Not After : Apr 26 23:45:18 2028 GMT
        Subject: C=US, ST=CA, L=SanFrancisco, O=sjsu, OU=sjsu, CN=client/name=server/emailAddress=pavanharavu@gmail.com
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
                Modulus:
                    00:ce:f2:b1:6d:11:4c:06:5e:f2:b2:d1:2e:87:c5:
                    ea:b4:e6:79:6e:ea:37:a1:10:44:96:54:1b:06:e1:
                    5c:29:77:a3:08:16:fb:2b:1a:92:d7:1d:68:5a:a0:
                    39:a1:54:5f:0c:1c:a5:b0:4e:9a:d6:89:7f:63:0a:
                    3a:6d:cd:9e:a4:5a:44:7d:fc:11:e6:ea:31:7f:26:
                    bc:6c:bd:d3:a8:4e:47:1f:3b:b4:5a:01:61:48:bd:
                    3d:45:12:de:c0:f6:0f:14:d5:ed:bd:c1:2a:9e:a8:
                    75:91:e0:91:34:2d:76:f2:21:95:76:b7:5b:d9:06:
                    31:5a:f9:24:e4:cf:df:1d:99:be:3d:5e:11:bf:02:
                    66:a1:f3:b6:21:39:4c:f8:d1:58:d9:1a:b7:c6:bf:
                    9e:2c:b6:b8:4a:89:92:93:a9:09:2a:d8:37:c0:07:
                    01:30:a2:99:b3:51:33:50:e4:23:f3:f3:6f:99:2e:
                    dc:41:39:5c:be:13:75:7a:aa:b5:ba:d9:a8:8e:cf:
                    12:d8:ae:a8:07:1a:3e:fa:54:cc:17:77:3b:8b:44:
                    f1:7b:c4:a6:44:ae:99:fd:35:43:af:bb:da:46:74:
                    bf:f0:3f:de:77:b8:46:0c:38:da:64:93:63:48:94:
                    24:c5:87:59:47:90:2a:f8:6c:0f:d0:78:bb:9e:74:
                    2d:bf
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Basic Constraints:
```

Figure 14. DSA SHA256 with RSA encryption



```
         18:d8:88:08:5e:5a:98:66:58:41:15:0c:dd:eb:5b:54:07:0c:
         d0:3e:2e:e8:5d:e4:21:43:29:1a:72:9f:fe:69:92:80:9c:50:
         2e:1c:89:f8:89:c5:5d:d2:8a:0d:72:7f:f4:d3:11:5c:70:bc:
         a1:23:6b:02:c8:5d:6a:ad:0f:f5:dc:07:e1:3c:37:b6:b5:95:
         92:91:80:f4:11:6f:59:ec:fa:2b:df:20:ed:ea:c9:3d:4b:3b:
         a7:47:6d:77
-----BEGIN CERTIFICATE-----
MIIFEDCCA/igAwIBAgIBAjANBgkqhkiG9w0BAQsFADCBmDELMAkGA1UEBhMCVVMx
CzAJBgNVBAgTAkNBMRUwEwYDVQQHEwxTYW5GcmFuY2lzY28xDTALBgNVBAoTBHNq
c3UxDTALBgNVBAsTBHNqc3UxEDAOBgNVBAMTB3Nqc3UgQ0ExDzANBgNVBCkTBnNl
cnZlcjEkMCIGCSqGSIb3DQEJARYVcGF2YW5oYXJhdnVAZ21haWwuY29tMB4XDTE4
MDQyOTIzNDUxOFoXDTI4MDQyNjIzNDUxOFowgZcxCzAJBgNVBAYTAlVTMQswCQYD
VQQIEwJDQTEVMBMGA1UEBxMMU2FuRnJhbmNpc2NvMQ0wCwYDVQQKEwRzanN1MQ0w
CwYDVQQLEwRzanN1MQ8wDQYDVQQDEwZjbGllbnQxDzANBgNVBCkTBnNlcnZlcjEk
MCIGCSqGSIb3DQEJARYVcGF2YW5oYXJhdnVAZ21haWwuY29tMIIBIjANBgkqhkiG
9w0BAQEFAAOCAQ8AMIIBCgKCAQEAzvKxbRFMBl7ystEuh8XqtOZ5buo3oRBEllQb
BuFcKXejCBb7KxqS1x1gWqA5oVRfDBylsE6a1ol/Ywo6bc2epFpEffwR5uoxfya8
bL3TqE5HHzu0WgFhSL09RRLewPYPFNXtvcEqnqh1keCRNC128iGVdrdb2QYxWvkk
5M/fHZm+PV4RvwJmofO2ITlM+NFY2Rq3xr+eLLa4SomSk6kJKtg3wAcBMKKZs1Ez
UOQj8/NvmS7cQTlcvhN1eqq1utmgjs8S2K6oBxo++lTMF3c7i0Txe8SmRK6Z/TVD
r7vaRnS/8D/ed7hGDDjaZJNjSJQkxQdZR5Aq+GwP0Hi7nnQtvwIDAQABo4IBYjCC
AV4wCQYDVR0TBAIwADAtBglghkgBhvhCAQ0EIBYeRWFzeS1SU0EgR2VuZXJhdGVk
IENlcnRpZmljYXRlMB0GA1UdDgQWBBRdNa+PcUQ00c4kdxhZiDpObbgQajCBzQYD
VR0jBIHFMIHCgBStgWER3+ArJpVZyWZhvS6u8goKQKGBnqSBmzCBmDELMAkGA1UE
BhMCVVMxCzAJBgNVBAgTAkNBMRUwEwYDVQQHEwxTYW5GcmFuY2lzY28xDTALBgNV
BAoTBHNqc3UxDTALBgNVBAsTBHNqc3UxEDAOBgNVBAMTB3Nqc3UgQ0ExDzANBgNV
BCkTBnNlcnZlcjEkMCIGCSqGSIb3DQEJARYVcGF2YW5oYXJhdnVAZ21haWwuY29t
ggkAtE2g8kViVqgwEwYDVR0lBAwwCgYIKwYBBQUHAwIwCwYDVR0PBAQDAgeAMBEG
A1UdEQQKMAiCBmNsaWVudDANBgkqhkiG9w0BAQsFAAOCAQEAamaKqGgOfzyCP55F
o/tavJ7rploIv4nNJnGS/V1umbq48VxV7QR33Sd6h71c2rZaDBpBZtrWXzDbkos0
a2B7AYwUWIZfOJtWAcrxptZ1/MtKd5EQMfs8/hkX79Qwt+SqTNusqA2iLhY/CObw
A/4iNcMjhqzl/ES+M429hggbcMr13UKAlbKDVV5tMtk4Azvi3y0CzhWN8q1EfCwV
GALQuimIGNiICF5amGZYQRUM3etbVAcM0D4u6F3kIUMpGnKf/mmSgJxQLhyJ+InF
XdKKDXJ/9NMRXHC8oSNrAshdaq0P9dwH4Tw3trWVkpGA9BFvWez6K98g7erJPUs7
p0dtdw==
-----END CERTIFICATE-----
</cert>
<key>
-----BEGIN PRIVATE KEY-----
MIIEvQIBADANBgkqhkiG9w0BAQEFAASCBKcwggSjAgEAAoIBAQDO8rFtEUwGXvKy
0S6Hxeq05nlu6jehEESWVBsG4Vwpd6MIFvsrGpLXHWBaoDmhVF8MHKWwTprWiX9j
CjptzZ6kWkR9/BHm6jF/JrxsvdOoTkcfO7RaAWFIvT1FEt7A9g8U1e29wSqeqHWR
4JE0LXbyIZV2t1vZBjFa+STkz98dmb49XhG/Amah87YhOUz40VjZGrfGv54strhK
iZKTqQkq2DfABwEwopmzUTNQ5CPz82+ZLtxBOVy+E3V6qrW62aCOzxLYrqgHGj76
VMwXdzuLRPF7xKZErpn9NUOvu9pGdL/wP953uEYMONpkk2NIlCTFB1lHkCr4bA/Q
eLuedC2/AgMBAAECggEAPq5PgyN8DWnwxKyLFWCi6Evud+PFGamdvJKY5QqZKY4s
```

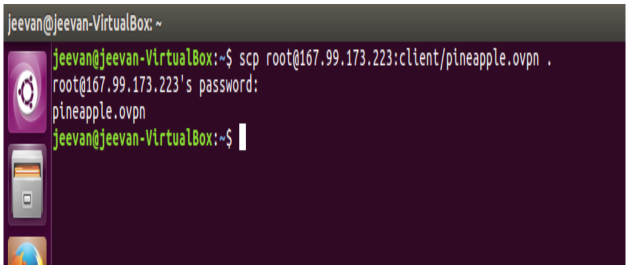
Figure 15. client certificate



```
root@pavanharavu: ~/client
  GNU nano 2.5.3                    File: pineapple.ovpn

BhMCVVMxCzAJBgNVBAgTAkNBMRUwEwYDVQQHEwxTYW5GcmFuY2lzY28xDTALBgNV
BAoTBHNqc3UxDTALBgNVBAsTBHNqc3UxEDAOBgNVBAMTB3Nqc3UgQ0ExDzANBgNV
BCkTBnNlcnZlcjEkMCIGCSqGSIb3DQEJARYVcGF2YW5oYXJhdnVAZ21haWwuY29t
ggkAtE2g8kViVqgwEwYDVR0lBAwwCgYIKwYBBQUHAwIwCwYDVR0PBAQDAgeAMBEG
A1UdEQQKMAiCBmNsaWVudDANBgkqhkiG9w0BAQsFAAOCAQEAanaKqGgOfzyCP55F
o/tavJ7rploIv4nNJnGS/V1umbq48VxV7QR33Sd6h71c2rZaDBpBZtrWXzDbkos0
a2B7AYwUWIZfOJtWAcrxptZ1/MtKd5EQMfs8/hkX79Qwt+SqTNusqA2iLhY/CObw
A/4iNcMjhqzl/ES+M429hggbcMr13UKAlbKDVV5tMtk4Azvi3y0CzhWN8q1EfCwV
GALQuimIGNiICF5amGZYQRUM3etbVAcM0D4u6F3kIUMpGnKf/mmSgJxQLhyJ+InF
XdKKDXJ/9NMRXHCBoSNrAshdaq8P9dwH4Tw3trWVkpGA9BFvWez6K98g7erJPUs7
p0dtdw==
-----END CERTIFICATE-----
</cert>
<key>
-----BEGIN PRIVATE KEY-----
MIIEvQIBADANBgkqhkiG9w0BAQEFAASCBKcwggSjAgEAAoIBAQDO8rFtEUwGXvKy
0S6Hxeq05nlu6jehEESWVBsG4Vwpd6MIFvsrGpLXHWBaoDmhVF8MHKWwTprWiX9j
CjptzZ6kWkR9/BHm6jF/JrxsvdOoTkcfO7RaAWFIvT1FEt7A9g8U1e29wSqeqHWR
4JE0LXbyIZV2t1vZBjFa+STkz98dmb49XhG/Amah87YhOUz48VjZGrfGvS4strhK
iZKTqQkq2DfABwEwopnzUTNQ5CPzB2+ZLtxBOVy+E3V6qrW62aCOzxLYrqgHGj76
VMwXdzuLRPF7xKZErpn9NUOvu9pGdL/wP953uEYNONpkk2NIlCTFB1lHkCr4bA/Q
eLuedC2/AgMBAAECggEAPq5PgyN8DWnwxKyLFWCi6Evud+PFGamdvJKY5QqZKY4s
+SRMyw9vbnxdZNWm3vB5IWgD31TPnS9kQBE3BU8EP+9/U1FnBrL4bvTCnYUdeJrM
++7Pf3tvheKbjjJa+cMNalgpfBHn4MpRYe/MyysCHfCCHsUGssMna9COlidIAuSk
mY6gzzqPmeaj4M1J1TjsrDeSI5E9FDh9hnwA1eXBAVTBgWVb9mnpNe1gu7h+O6nN
kzarEvSWaIXJcQbZeEnDzDJOjJCsJte1XLSu6a76M2/FF89h+wxBWj64AUKiSFbD
7h2fXQ8mma5srAzMGt1azFdDZ8kuc7pcpgg9JTvO8QKBgQDx7jV+AJDPmwtyawag
8HAxP9pJwMqjijlvoZFpWXuEHgsAHBrDkv9ODyaB12/9drVLQWldcUl8UYtuy2Xg
9EUyDuPVjwA+b3g4QdrhFRGRS79NN59BWF8Oua51cl/o/tZU4G6Zn12FgXzhf0xM
rs1HhjQl+E+btJFyc6IfmXQ1iwKBgQDa+6zQv82ST58oJU8TiQcIobwNnlOvK4nZ
uqt1yJNrAqr6iJ9IA3I9vZ4IdimBM+lXyaV7YvbMKWoN+UQ7Q6LmkZYItaMN2eyD
3Xalq/2QbG3GoOSnxi9ZPmsm3cKobS+1kVUHUaeLKh/kOpv49+dyGnSa4/g7Ukmt
K0B59AP3HQKBgFvBlNgtLr8P+y9yGBmCF4sWRJwkWrab/dPe21+xaH87RcDM5/m1
Ds90P46PPxJKq4fXTzb5eH2GD/P9gK5I9tV9gDvG+Li1Xks37uPfMcKBkBShO+10
rtjO47P3pRihTnjyhgANyVKRQePke/O5eHHgHHYG52qOmXtnD94cPU7xAoG4Vn6e
Lf3CCA6GRrslDJhNPO2VN9PXvhL2ahe/bdCB2fy9yCN6tpkV4Kkq6LCziCMXAePW
Bzx5uv6Is2hHuRjoGSZcGqiT9P6k2xj4qkua5hbN7WzTpCk1voxXCo0BPEGUKKVA
7pkkEmXbHuy/esGAm6Mzd5KgE95M152D5CesAUkCgYEAnkafzqO5Ac9MZg23KF/V
WEWgoCIJESjphhMBoLR2E24fmn+LqGf4nlN8ztlx8/gMFSH3GUQTrV+dBk5Mket/
aqzC2dvyoP/4ua6eq6qB1zT3td+XHcyVopePdXnjEHMO2/k6O+DvQVbVzouGHgWP
372Uuell8+hG7lC/3f94c2k=
-----END PRIVATE KEY-----
</key>

^G Get Help    ^O Write Out   ^W Where Is    ^K Cut Text    ^J Justify     ^C Cur Pos     ^Y Prev Page   M-\ First Line  M-W WhereIs Next
^X Exit        ^R Read File   ^\ Replace     ^U Uncut Text  ^T To Spell    ^_ Go To Line  ^V Next Page   M-/ Last Line   M-] To Bracket
```

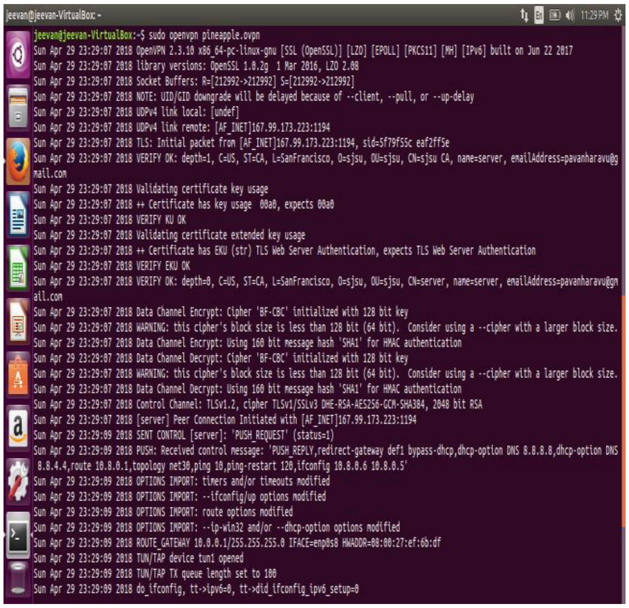
Figure 16. Private Key

4. Results



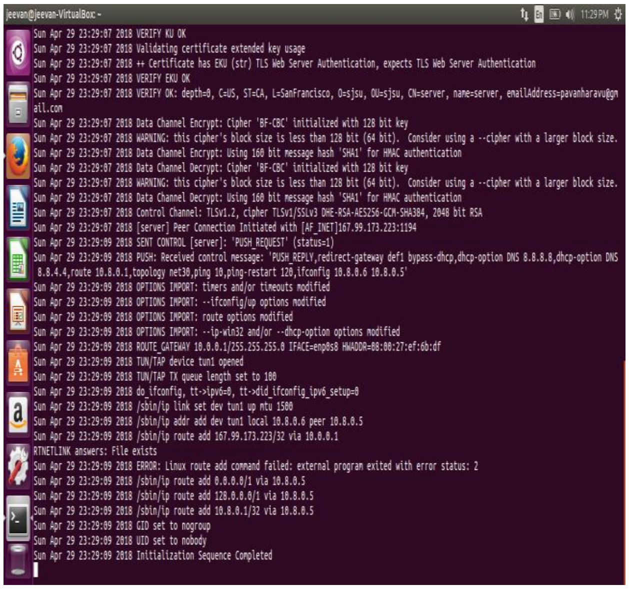
```
jeevan@jeevan-VirtualBox: ~
jeevan@jeevan-VirtualBox:~$ scp root@167.99.173.223:client/pineapple.ovpn .
root@167.99.173.223's password:
pineapple.ovpn
jeevan@jeevan-VirtualBox:~$ 
```

Figure 17. Securely copying file to client computer



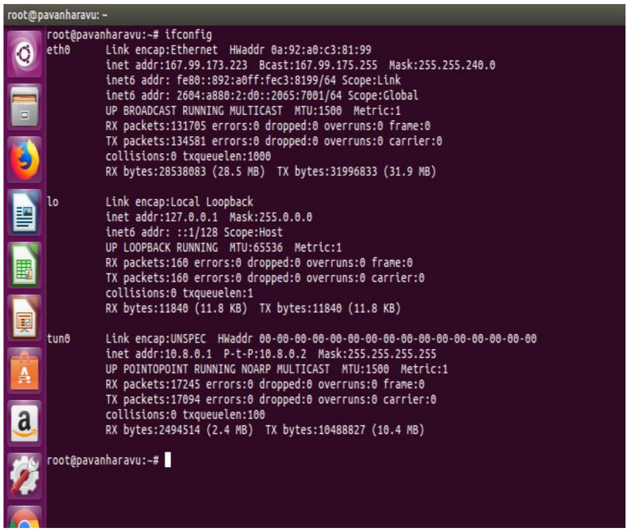
```
jeevan@jeevan-VirtualBox: ~
jeevan@jeevan-VirtualBox:~$ sudo openvpn pineapple.ovpn
Sun Apr 29 23:29:07 2018 OpenVPN 2.3.10 x86_64-pc-linux-gnu [SSL (OpenSSL)] [LZO] [EPOLL] [PKCS11] [MH] [IPv6] built on Jun 22 2017
Sun Apr 29 23:29:07 2018 library versions: OpenSSL 1.0.2g  1 Mar 2016, LZO 2.08
Sun Apr 29 23:29:07 2018 Socket Buffers: R=[212992->212992] S=[212992->212992]
Sun Apr 29 23:29:07 2018 NOTE: UID/GID downgrade will be delayed because of --client, --pull, or --up-delay
Sun Apr 29 23:29:07 2018 UDPv4 link local: [undef]
Sun Apr 29 23:29:07 2018 UDPv4 link remote: [AF_INET]167.99.173.223:1194
Sun Apr 29 23:29:07 2018 TLS: Initial packet from [AF_INET]167.99.173.223:1194, sid=5f79f55c eaf2ff5e
Sun Apr 29 23:29:07 2018 VERIFY OK: depth=1, C=US, ST=CA, L=SanFrancisco, O=sjsu, OU=sjsu, CN=sjsu CA, name=server, emailAddress=pavanharavu@g
mail.com
Sun Apr 29 23:29:07 2018 Validating certificate key usage
Sun Apr 29 23:29:07 2018 ++ Certificate has key usage  00a0, expects 00a0
Sun Apr 29 23:29:07 2018 VERIFY KU OK
Sun Apr 29 23:29:07 2018 Validating certificate extended key usage
Sun Apr 29 23:29:07 2018 ++ Certificate has EKU (str) TLS Web Server Authentication, expects TLS Web Server Authentication
Sun Apr 29 23:29:07 2018 VERIFY EKU OK
Sun Apr 29 23:29:07 2018 VERIFY OK: depth=0, C=US, ST=CA, L=SanFrancisco, O=sjsu, OU=sjsu, CN=server, name=server, emailAddress=pavanharavu@gm
ail.com
Sun Apr 29 23:29:07 2018 Data Channel Encrypt: Cipher 'BF-CBC' initialized with 128 bit key
Sun Apr 29 23:29:07 2018 WARNING: this cipher's block size is less than 128 bit (64 bit).  Consider using a --cipher with a larger block size.
Sun Apr 29 23:29:07 2018 Data Channel Encrypt: Using 160 bit message hash 'SHA1' for HMAC authentication
Sun Apr 29 23:29:07 2018 Data Channel Decrypt: Cipher 'BF-CBC' initialized with 128 bit key
Sun Apr 29 23:29:07 2018 WARNING: this cipher's block size is less than 128 bit (64 bit).  Consider using a --cipher with a larger block size.
Sun Apr 29 23:29:07 2018 Data Channel Decrypt: Using 160 bit message hash 'SHA1' for HMAC authentication
Sun Apr 29 23:29:07 2018 Control Channel: TLSv1.2, cipher TLSv1/SSLv3 DHE-RSA-AES256-GCM-SHA384, 2048 bit RSA
Sun Apr 29 23:29:07 2018 [server] Peer Connection Initiated with [AF_INET]167.99.173.223:1194
Sun Apr 29 23:29:09 2018 SENT CONTROL [server]: 'PUSH_REQUEST' (status=1)
Sun Apr 29 23:29:09 2018 PUSH: Received control message: 'PUSH_REPLY,redirect-gateway def1 bypass-dhcp,dhcp-option DNS 8.8.8.8,dhcp-option DNS
 8.8.4.4,route 10.8.0.1,topology net30,ping 10,ping-restart 120,ifconfig 10.8.0.6 10.8.0.5'
Sun Apr 29 23:29:09 2018 OPTIONS IMPORT: timers and/or timeouts modified
Sun Apr 29 23:29:09 2018 OPTIONS IMPORT: --ifconfig/up options modified
Sun Apr 29 23:29:09 2018 OPTIONS IMPORT: route options modified
Sun Apr 29 23:29:09 2018 OPTIONS IMPORT: --ip-win32 and/or --dhcp-option options modified
Sun Apr 29 23:29:09 2018 ROUTE_GATEWAY 10.0.0.1/255.255.255.0 IFACE=enp0s8 HWADDR=08:00:27:ef:6b:df
Sun Apr 29 23:29:09 2018 TUN/TAP device tun1 opened
Sun Apr 29 23:29:09 2018 TUN/TAP TX queue length set to 100
Sun Apr 29 23:29:09 2018 do_ifconfig, tt->ipv6=0, tt->did_ifconfig_ipv6_setup=0
```

Figure 18. server and client authentication



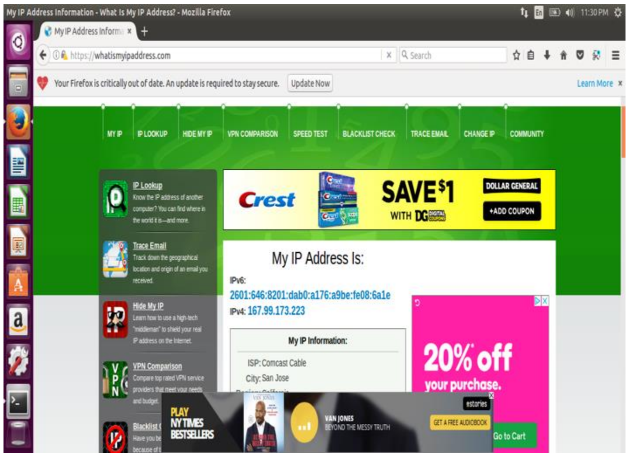
```
jeevan@jeevan-VirtualBox: ~
Sun Apr 29 23:29:07 2018 VERIFY KU OK
Sun Apr 29 23:29:07 2018 Validating certificate extended key usage
Sun Apr 29 23:29:07 2018 ++ Certificate has EKU (str) TLS Web Server Authentication, expects TLS Web Server Authentication
Sun Apr 29 23:29:07 2018 VERIFY EKU OK
Sun Apr 29 23:29:07 2018 VERIFY OK: depth=0, C=US, ST=CA, L=SanFrancisco, O=sjsu, OU=sjsu, CN=server, name=server, emailAddress=pavanharavu@gm
ail.com
Sun Apr 29 23:29:07 2018 Data Channel Encrypt: Cipher 'BF-CBC' initialized with 128 bit key
Sun Apr 29 23:29:07 2018 WARNING: this cipher's block size is less than 128 bit (64 bit).  Consider using a --cipher with a larger block size.
Sun Apr 29 23:29:07 2018 Data Channel Encrypt: Using 160 bit message hash 'SHA1' for HMAC authentication
Sun Apr 29 23:29:07 2018 Data Channel Decrypt: Cipher 'BF-CBC' initialized with 128 bit key
Sun Apr 29 23:29:07 2018 WARNING: this cipher's block size is less than 128 bit (64 bit).  Consider using a --cipher with a larger block size.
Sun Apr 29 23:29:07 2018 Data Channel Decrypt: Using 160 bit message hash 'SHA1' for HMAC authentication
Sun Apr 29 23:29:07 2018 Control Channel: TLSv1.2, cipher TLSv1/SSLv3 DHE-RSA-AES256-GCM-SHA384, 2048 bit RSA
Sun Apr 29 23:29:07 2018 [server] Peer Connection Initiated with [AF_INET]167.99.173.223:1194
Sun Apr 29 23:29:09 2018 SENT CONTROL [server]: 'PUSH_REQUEST' (status=1)
Sun Apr 29 23:29:09 2018 PUSH: Received control message: 'PUSH_REPLY,redirect-gateway def1 bypass-dhcp,dhcp-option DNS 8.8.8.8,dhcp-option DNS
 8.8.4.4,route 10.8.0.1,topology net30,ping 10,ping-restart 120,ifconfig 10.8.0.6 10.8.0.5'
Sun Apr 29 23:29:09 2018 OPTIONS IMPORT: timers and/or timeouts modified
Sun Apr 29 23:29:09 2018 OPTIONS IMPORT: --ifconfig/up options modified
Sun Apr 29 23:29:09 2018 OPTIONS IMPORT: route options modified
Sun Apr 29 23:29:09 2018 OPTIONS IMPORT: --ip-win32 and/or --dhcp-option options modified
Sun Apr 29 23:29:09 2018 ROUTE_GATEWAY 10.0.0.1/255.255.255.0 IFACE=enp0s8 HWADDR=08:00:27:ef:6b:df
Sun Apr 29 23:29:09 2018 TUN/TAP device tun1 opened
Sun Apr 29 23:29:09 2018 TUN/TAP TX queue length set to 100
Sun Apr 29 23:29:09 2018 do_ifconfig, tt->ipv6=0, tt->did_ifconfig_ipv6_setup=0
Sun Apr 29 23:29:09 2018 /sbin/ip link set dev tun1 up mtu 1500
Sun Apr 29 23:29:09 2018 /sbin/ip addr add dev tun1 local 10.8.0.6 peer 10.8.0.5
Sun Apr 29 23:29:09 2018 /sbin/ip route add 167.99.173.223/32 via 10.0.0.1
RTNETLINK answers: File exists
Sun Apr 29 23:29:09 2018 ERROR: Linux route add command failed: external program exited with error status: 2
Sun Apr 29 23:29:09 2018 /sbin/ip route add 0.0.0.0/1 via 10.8.0.5
Sun Apr 29 23:29:09 2018 /sbin/ip route add 128.0.0.0/1 via 10.8.0.5
Sun Apr 29 23:29:09 2018 /sbin/ip route add 10.8.0.1/32 via 10.8.0.5
Sun Apr 29 23:29:09 2018 GID set to nogroup
Sun Apr 29 23:29:09 2018 UID set to nobody
Sun Apr 29 23:29:09 2018 Initialization Sequence Completed
```

Figure 19. server and client authentication completed



```
root@pavanharavu: ~
root@pavanharavu:~# ifconfig
eth0      Link encap:Ethernet  HWaddr 0a:92:a0:c3:81:99
          inet addr:167.99.173.223  Bcast:167.99.175.255  Mask:255.255.240.0
          inet6 addr: fe80::892:a0ff:fec3:8199/64 Scope:Link
          inet6 addr: 2604:a880:2:d0::2065:7001/64 Scope:Global
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:131705 errors:0 dropped:0 overruns:0 frame:0
          TX packets:134581 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:28538083 (28.5 MB)  TX bytes:31996833 (31.9 MB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:160 errors:0 dropped:0 overruns:0 frame:0
          TX packets:160 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:11840 (11.8 KB)  TX bytes:11840 (11.8 KB)

tun0      Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
          inet addr:10.8.0.1  P-t-P:10.8.0.2  Mask:255.255.255.255
          UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500  Metric:1
          RX packets:17245 errors:0 dropped:0 overruns:0 frame:0
          TX packets:17094 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          RX bytes:2494514 (2.4 MB)  TX bytes:10488827 (10.4 MB)

root@pavanharavu:~# 
```

Figure 20. server IP and tun0 interface



My IP Address Information - What Is My IP Address? - Mozilla Firefox

https://whatismyipaddress.com

Your Firefox is critically out of date. An update is required to stay secure.

My IP Address Is:

IPv6: 2601:646:8201:dab0:a176:a9be:fe08:6a1e

IPv4: 167.99.173.223

My IP Information:

ISP: Comcast Cable

City: San Jose

My IP Address Inform
Search
Update Now
Learn More
MY IP
IP LOOKUP
HIDE MY IP
VPN COMPARISON
SPEED TEST
BLACKLIST CHECK
TRACE EMAIL
CHANGE IP
COMMUNITY
IP Lookup
Know the IP address of another computer? You can find where in the world it is—and more.
Trace Email
Track down the geographical location and origin of an email you received.
Hide My IP
Learn how to use a high-tech "middleman" to shield your real IP address on the Internet.
VPN Comparison
Compare top rated VPN service providers that meet your needs and budget.
Crest
SAVE $1
WITH DG
DOLLAR GENERAL
+ADD COUPON
20% off
your purchase.
Go to Cart
PLAY NY TIMES BESTSELLERS
VAN JONES
BEYOND THE MESSY TRUTH
GET A FREE AUDIOBOOK
estories

Figure 21. Client IP same as server IP

5. Conclusion

SSL VPN is successfully build and verified in client computer by checking its IP address.

6. Acknowledgment

We would like to thank Juzi Zhao for providing technical knowledge and guidance for implementing the lab.

7. References

[1] Syracuse seed labs
http://www.cis.syr.edu/~wedu/seed/Labs_12.04/Networking/VPN/
[2] Openvpn community
https://openvpn.net/