DEVOPS INTERNSHIP

2021

Assignment

Virtual Private Network (VPN)

Name: Bijay Aashish Bhatta

Question Number 1:

i. At first, centos packages are updated using:

$ sudo yum update

ii. OpenVPN is not available in the default CentOS repositories. To install OpenVPN, Extra Packages for Enterprise Linux (EPEL) repository can be used. To install EPEL repository:

$ sudo yum install epel-release

iii. Update repository and install OpenVPN:

$ sudo yum update

$ sudo yum install openvpn

iv. Now we need to setup an internal CA to generate SSL key pairs to secure VPN connections. It can be achieved using following commands:

$ sudo yum install wget

$ wget https://github.com/OpenVPN/easy-rsa/archive/v3.0.8.tar.gz

$ sudo tar -xf v3.0.8.tar.gz

$ cd /etc/openvpn/

$ mkdir /etc/openvpn/easy-rsa

$ mv /home/psyphernix/easy-rsa-3.0.8 /etc/openvpn/easy-rsa

v. Certificates were created using EasyRSA and ca.crt, dh.pem, ca.key, crl.pem, server.crt, and server.key are copied to /etc/openvpn/server/ directory.

vi. Server.conf files was modified with only required following info:

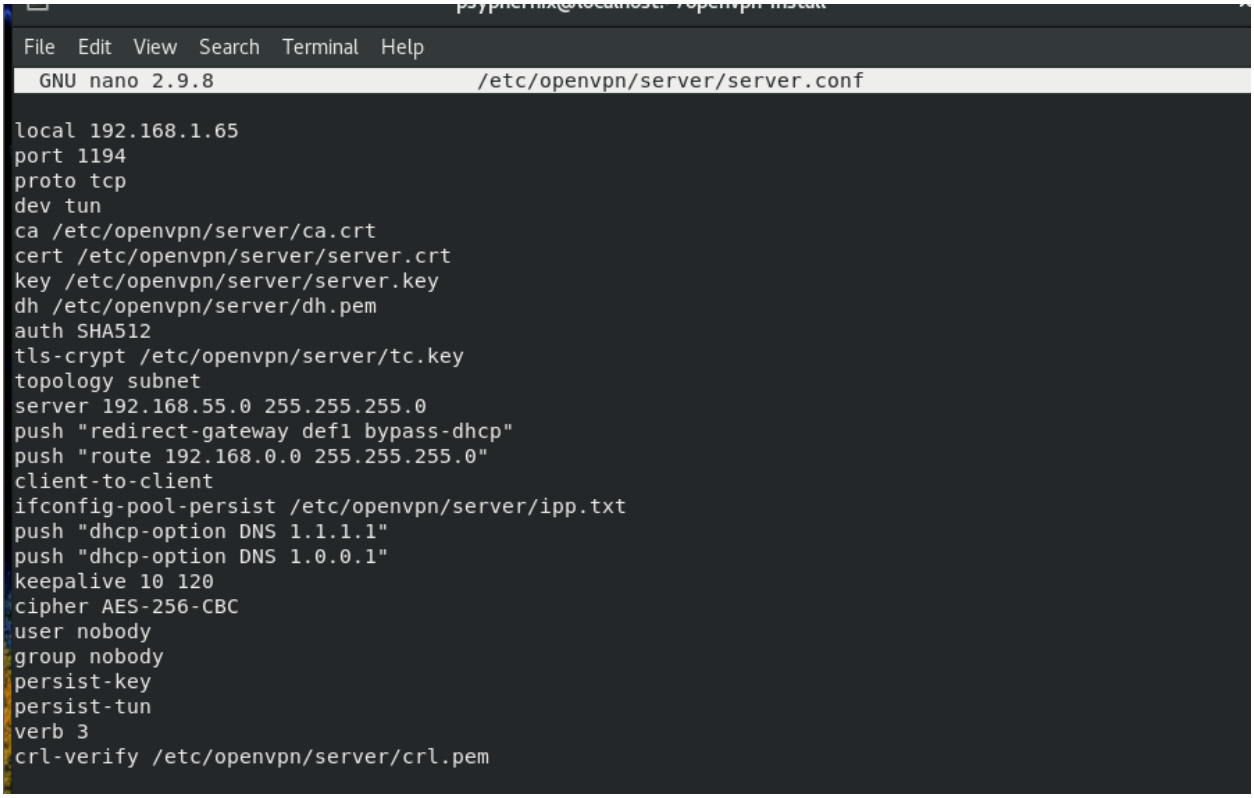

Figure 1 server.conf

vii. Following file client-common.txt is for common configuration for all clients.

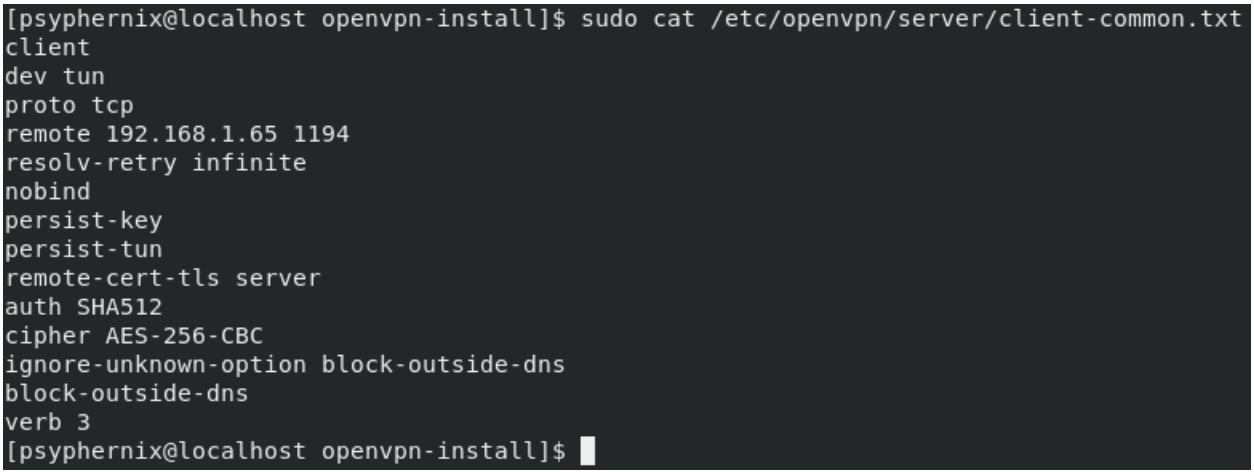

Figure 2 client-common.txt

viii. Client file with .ovpn format is created, then exported to another machine within a LAN as shown in Figure 3.

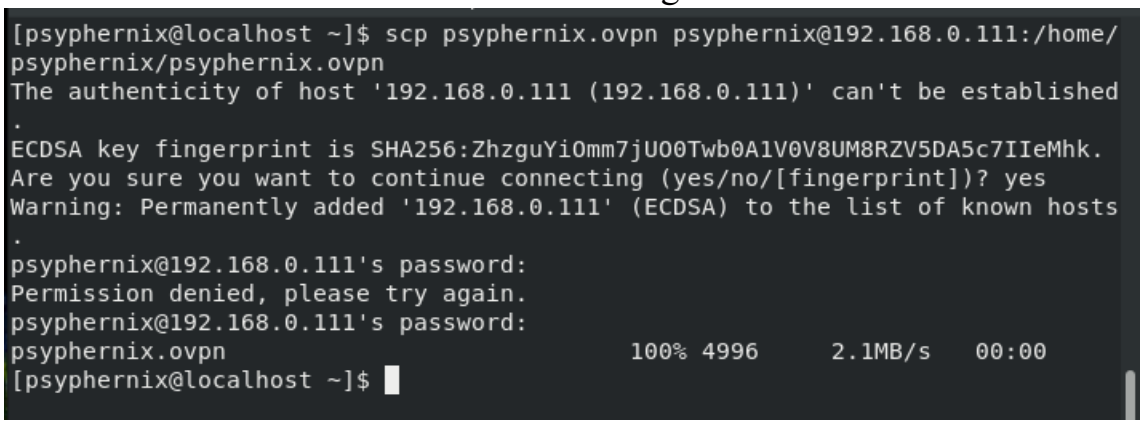

Figure 3 SCP to transfer file to another machine.

ix. Client was connected to VPN using psyphernix.ovpn file that was exported from VPN server using:

$ sudo openvpn –config psyphernix.ovpn

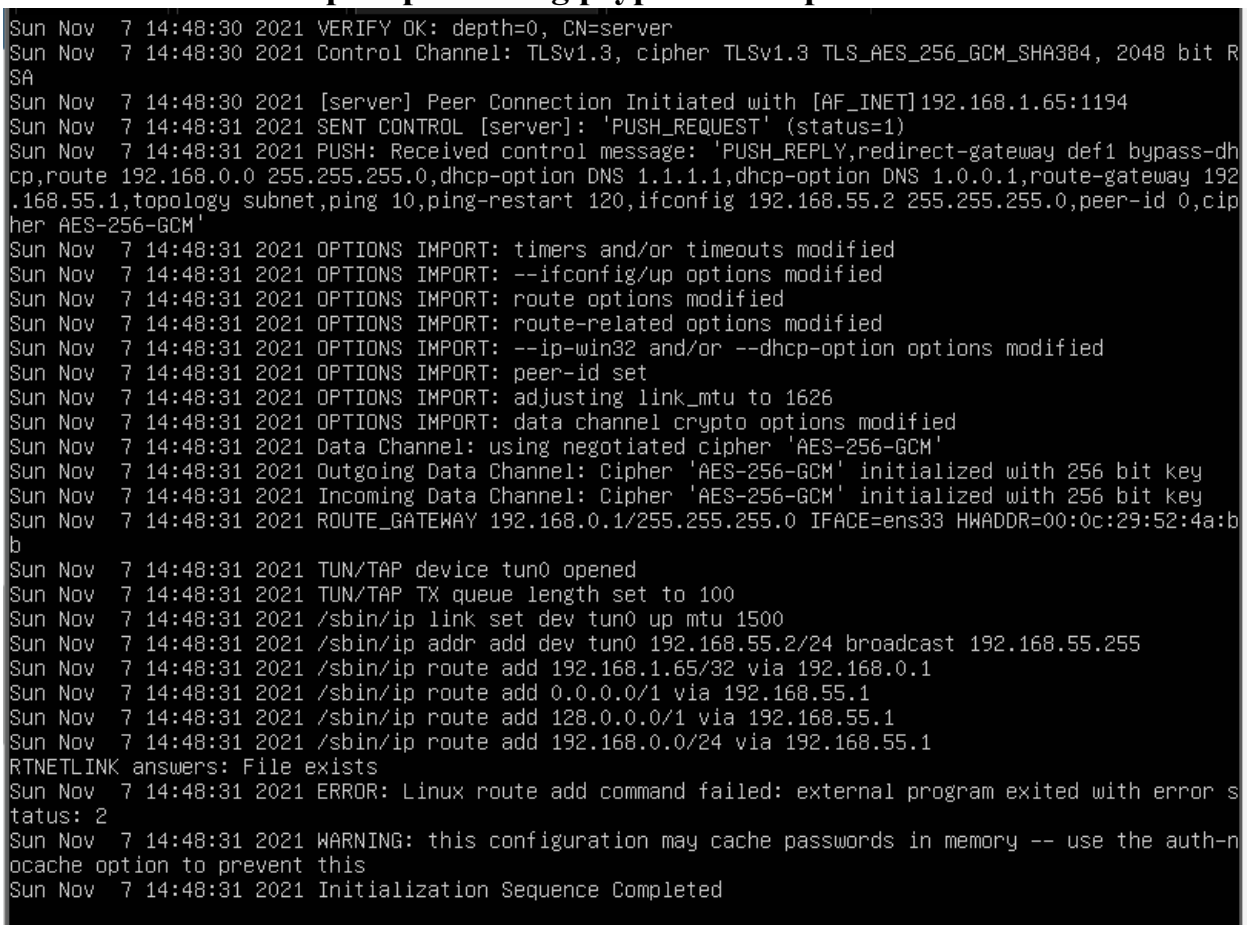

Figure 4 Linux client connected to VPN using openvpn.

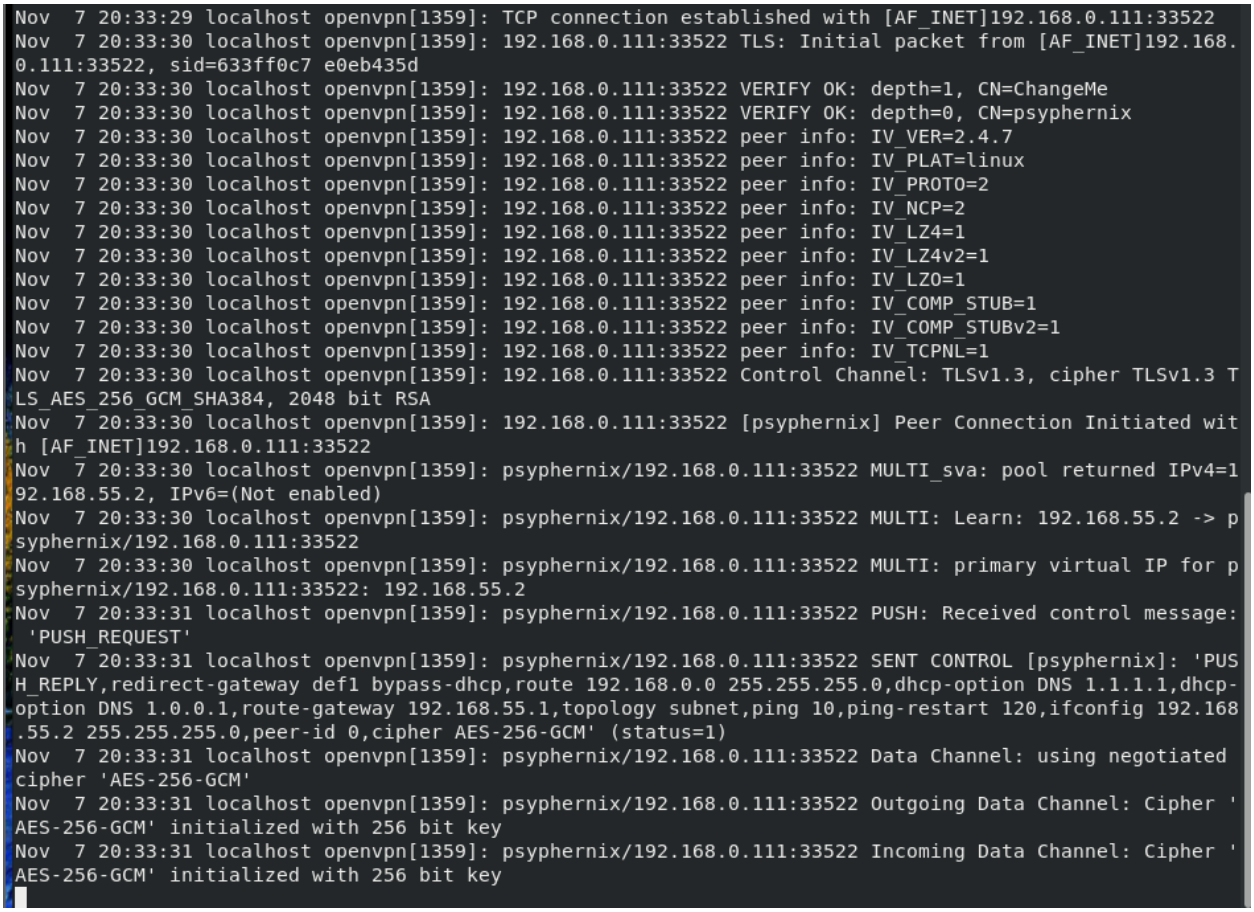

Figure 5 Server Connection log for psyphernix.ovpn

Question Number 2:

For this, the second client config file is created with the name LeapfrogHR.ovpn and the file was transferred to the windows 8 VM machine (connected in the bridged mode). LAN network has a network address of 192.168.0.0/24. And VPN network has the address 192.168.55.0/24. 192.168.1.65 is the server address to connect to the VPN, which is also the address provided by my home router to the VM machine in the bridged mode. OpenVPN connect is downloaded and the ovpn file is imported. The client was then connected to the VPN server as shown in figure 6.

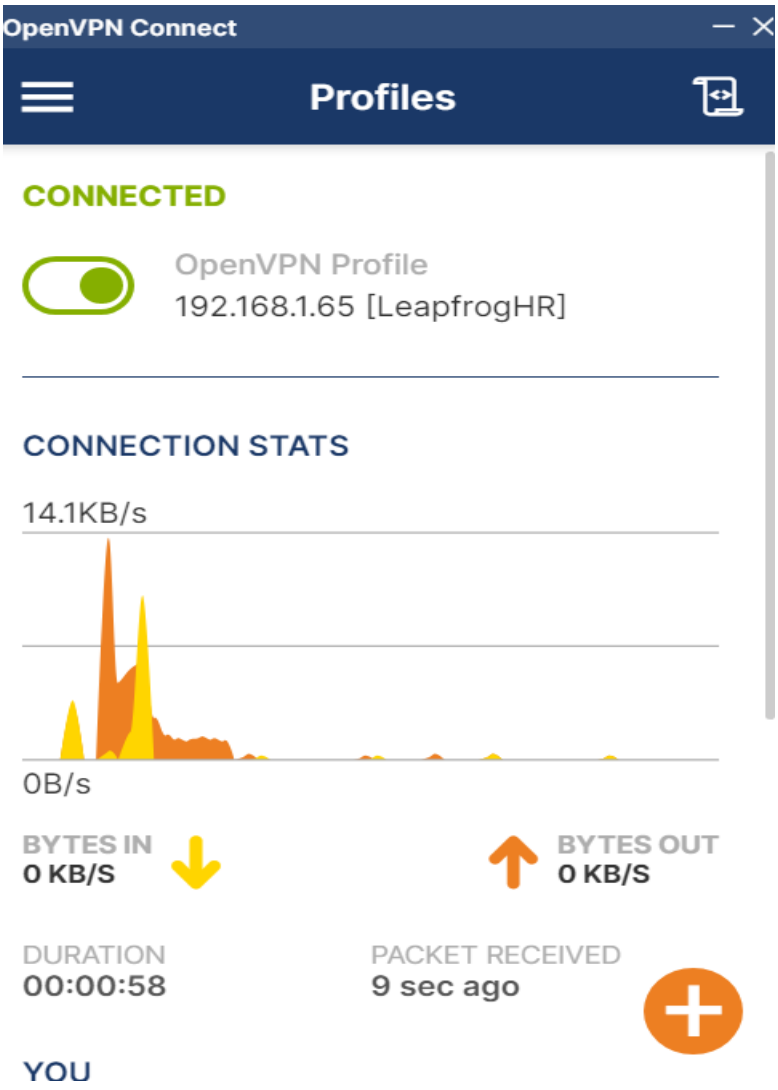

Figure 6 OpenVPN connect app showing client connected to VM

```
Nov  7 23:12:52 localhost openvpn[1359]: TCP connection established with [AF_INET]192.168.1.72:1363
Nov  7 23:12:52 localhost openvpn[1359]: 192.168.1.72:1363 TLS: Initial packet from [AF_INET]192.168.1
.72:1363, sid=8d41d167 c754e1b3
Nov  7 23:12:52 localhost openvpn[1359]: 192.168.1.72:1363 VERIFY OK: depth=1, CN=ChangeMe
Nov  7 23:12:52 localhost openvpn[1359]: 192.168.1.72:1363 VERIFY OK: depth=0, CN=LeapfrogHR
Nov  7 23:12:52 localhost openvpn[1359]: 192.168.1.72:1363 peer info: IV_VER=3.git::c2153df1
Nov  7 23:12:52 localhost openvpn[1359]: 192.168.1.72:1363 peer info: IV_PLAT=win
Nov  7 23:12:52 localhost openvpn[1359]: 192.168.1.72:1363 peer info: IV_NCP=2
Nov  7 23:12:52 localhost openvpn[1359]: 192.168.1.72:1363 peer info: IV_TCPNL=1
Nov  7 23:12:52 localhost openvpn[1359]: 192.168.1.72:1363 peer info: IV_PROTO=30
Nov  7 23:12:52 localhost openvpn[1359]: 192.168.1.72:1363 peer info: IV_CIPHERS=AES-256-GCM:AES-128-G
CM:CHACHA20-POLY1305:AES-256-CBC
Nov  7 23:12:52 localhost openvpn[1359]: 192.168.1.72:1363 peer info: IV_AUTO_SESS=1
Nov  7 23:12:52 localhost openvpn[1359]: 192.168.1.72:1363 peer info: IV_GUI_VER=OCWindows_3.3.2-2475
Nov  7 23:12:52 localhost openvpn[1359]: 192.168.1.72:1363 peer info: IV_SSO=openurl,crtext
Nov  7 23:12:52 localhost openvpn[1359]: 192.168.1.72:1363 Control Channel: TLSv1.3, cipher TLSv1.3 TL
S_AES_256_GCM_SHA384, 2048 bit RSA
Nov  7 23:12:52 localhost openvpn[1359]: 192.168.1.72:1363 [LeapfrogHR] Peer Connection Initiated with
 [AF_INET]192.168.1.72:1363
Nov  7 23:12:52 localhost openvpn[1359]: LeapfrogHR/192.168.1.72:1363 MULTI_sva: pool returned IPv4=19
2.168.55.3, IPv6=(Not enabled)
Nov  7 23:12:52 localhost openvpn[1359]: LeapfrogHR/192.168.1.72:1363 MULTI: Learn: 192.168.55.3 -> Le
apfrogHR/192.168.1.72:1363
Nov  7 23:12:52 localhost openvpn[1359]: LeapfrogHR/192.168.1.72:1363 MULTI: primary virtual IP for Le
apfrogHR/192.168.1.72:1363: 192.168.55.3
Nov  7 23:12:52 localhost openvpn[1359]: LeapfrogHR/192.168.1.72:1363 PUSH: Received control message:
'PUSH_REQUEST'
Nov  7 23:12:52 localhost openvpn[1359]: LeapfrogHR/192.168.1.72:1363 SENT CONTROL [LeapfrogHR]: 'PUSH
_REPLY,redirect-gateway def1 bypass-dhcp,route 192.168.0.0 255.255.255.0,dhcp-option DNS 1.1.1.1,dhcp-
option DNS 1.0.0.1,route-gateway 192.168.55.1,topology subnet,ping 10,ping-restart 120,ifconfig 192.16
8.55.3 255.255.255.0,peer-id 0,cipher AES-256-GCM' (status=1)
Nov  7 23:12:52 localhost openvpn[1359]: LeapfrogHR/192.168.1.72:1363 Data Channel: using negotiated c
ipher 'AES-256-GCM'
Nov  7 23:12:52 localhost openvpn[1359]: LeapfrogHR/192.168.1.72:1363 Outgoing Data Channel: Cipher 'A
ES-256-GCM' initialized with 256 bit key
Nov  7 23:12:52 localhost openvpn[1359]: LeapfrogHR/192.168.1.72:1363 Incoming Data Channel: Cipher 'A
ES-256-GCM' initialized with 256 bit key
```

Figure 7 Server Connection log for LeapfrogHR.ovpn

```
C:\Users\BijayAashish>tracert ekantipur.com

Tracing route to ekantipur.com [103.48.88.30]
over a maximum of 30 hops:

  1     2 ms     2 ms     2 ms  192.168.55.1
  2    11 ms     5 ms     7 ms  192.168.1.254
  3    12 ms    15 ms    10 ms  27.34.52.1
  4    14 ms    24 ms     8 ms  rp-pppoe-04.wlink.com.np [202.79.40.15]
  5    10 ms     9 ms    12 ms  ae-7-137.40.gw-jwl-stc-01.wlink.com.np [202.79.4
0.137]
  6    13 ms     9 ms     4 ms  12.34.79.202.wlink.com.np [202.79.34.12]
  7     7 ms     6 ms     6 ms  icc-streaming-241.ekantipur.com [103.48.88.241]

  8    11 ms     7 ms    10 ms  ip-103-48-88-30.ekantipur.com [103.48.88.30]

Trace complete.

C:\Users\BijayAashish>
```

Figure 8 Trace route showing connection to ekantipur.com being routed via VPN server machine.

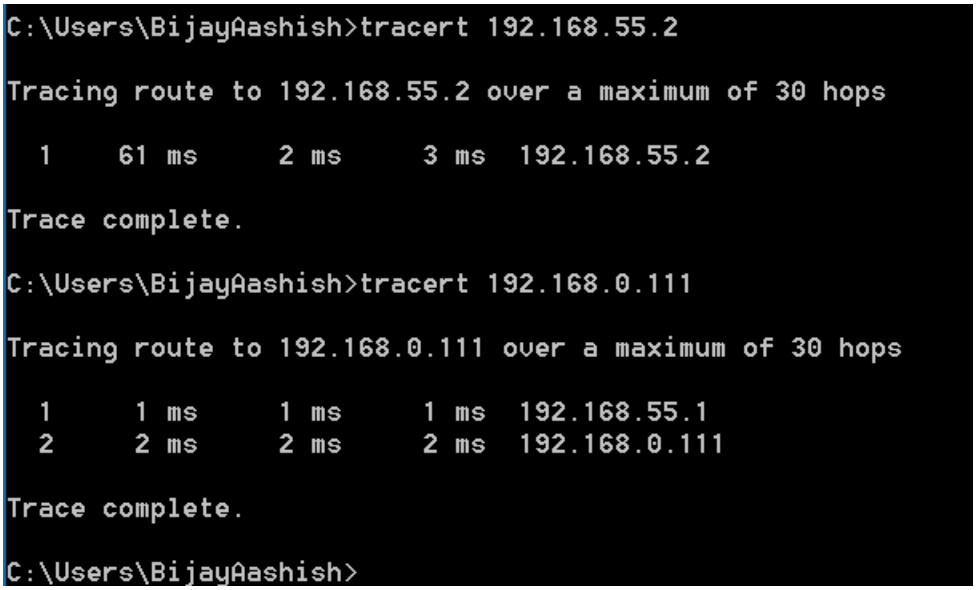

Figure 9 Trace route to LAN client is successful.

Question Number 3:

The intrusion detection system is a monitoring system that uses signatures to detect and analyse both inbound and outbound network traffic. It can detect abnormal activities in the network. IDS do not take any measure on its own. IDS don't modify the network packets by any means, while IPS avoids the packet from transfer based on the contents of the packet, in the same way how a firewall stops traffic by IP address. IDS require another system or human to act. It just notifies about abnormal traffic or attack in the network. Some advantages of installing an intrusion detection system instead of a prevention system are:

i. It uses signature-based detection methods to recognize possible threats. Using the signature database, IDS assure the immediate and positive discovery of known anomalies, normally safe from giving false alerts. Also, it doesn't alter packets as IPS does.
ii. It uses its comprehensive attack signature database, raises caution, and shows proper warnings on spotting an attack or abnormal traffic.
iii. Various kinds of attacks on the network help to discriminate malicious traffic and non-malicious traffic, aiding the administrator to tune, comprise, and implement practical controls.
iv. It aids the organization to keep up organizational consistency and meet security strategies as it provides more visibility over the whole system.
v. Host-based intrusion detection system provides detection of any alteration or tries to modify system files or any anomalous activity originating from inside the organization.

IDS cannot prevent an atomic attack as it has an off-site implementation, and it can only notify about abnormal traffic and attack. To prevent an attack, the organization will need to implement the Intrusion Prevention System (IPS) which has in-line implementation and cannot only detect abnormal traffic and attacks but also prevent attacks on the network.

VPN service is usually available in routers, but it can be implemented in servers using the software. There are many open software available for different systems like Suricata and Snort. Here, Suricata is going to be implemented on the centos 8 servers.

Suricata was installed using:

$ dnf install suricata

Then all the required changes were made on Suricata.yaml config file which is available on /etc/suricata/suricata.yaml and a rule have been created to notify the system of ongoing Denial of Service (DoS) attack on the system.

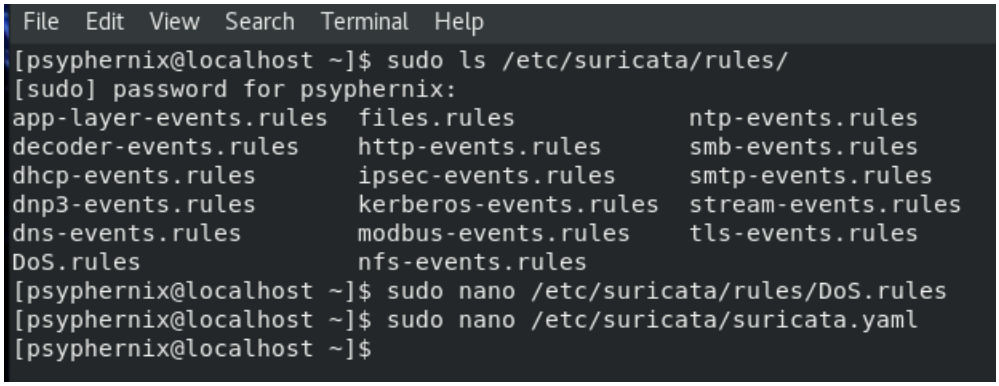

Figure 10 Suricata Rules

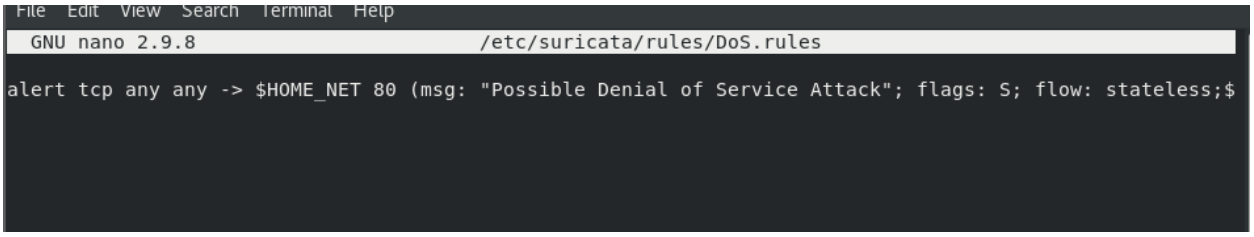

Figure 11 DoS rule