NSR/AS Lab 3 – VPNs – Dylan Rodwell – 105341089

Dylan Rodwell
(105341089)
105341089@student.swin.edu.au

Abstract

This lab session explored the implementation and behaviour of Virtual Private Networks (VPNs), using OpenVPN to connect two virtual Ubuntu machines. The goal of this lab was to establish a secure encrypted tunnel between both virtual machines using a shared key. The lab involved generating and securely transferring a key, installing ssh on the machines, and setting up the bi-directional VPN. Wireshark was used to monitor traffic on the tunnel and the ethernet interface, and did confirm that the VPN tunnels successfully encrypted traffic. This was demonstrated using pings and telnet.

Introduction to VPNs

What is a VPN?

Article [1] states that a Virtual Private Network (VPN) "is an encrypted connection that secures data transmission between devices over the internet". This is used to encrypt the data traffic and securely transfer data between devices.

VPN Capabilities and Vulnerabilities

The capabilities of VPNs include:

Secure encryption. To access the data on the VPN, a device needs an encryption key. This security feature prevents unauthorised access to your data while using a VPN, even when on a public network.

Data transfer. VPNs can also be used to transfer data via a secure connection. The encrypted connection stops people without the encryption key from accessing the data being transferred, making it suitable for transferring sensitive information.

Remote Access. The VPN tunnelling can also be used to remotely connect to devices via a secure connection where the traffic is encrypted. This is very useful for securely remotely logging on to a company network from home when working from home.

How VPNs can be used to implement organisational security policy

VPNs do help enforce organisational security policies in various ways. These include:

Data confidentiality Policies. By using VPNs to encrypt all communications and data traffic, it ensures that all traffic stays secure even when travelling through public networks. This keeps the data confidential and compliant with data protection policies like HIPAA and GDPR.

Access control. Access control can be used with VPNs as well, by using authenticated VPN connections. This allows organisations to apply role-based access control. This prevents users from accessing sensitive or confidential information that they shouldn't have access to via the VPN connection.

Auditing. VPN servers can log information about connections, which helps admins of the network to monitor connections for suspicious activity and aid in investigations.

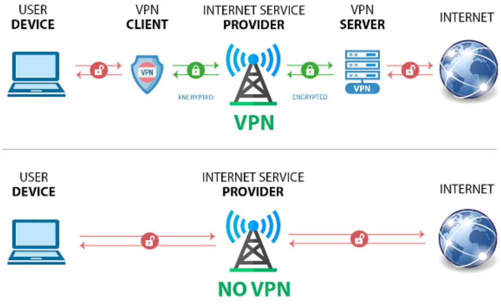


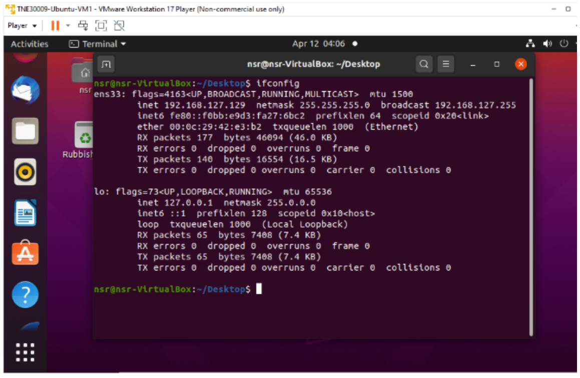
Figure 1. [4]. example diagram of VPN use in a business.

OpenVPN Behaviour

Host Configuration

The first step in this lab is to check the ip addresses on the virtual machines by using the command "ifconfig".

(ifconfig photo for VM1 was lost)


TNE30009-Ubuntu-VM1 - VMware Workstation 17 Player (Non-commercial use only)
Player
Activities
Terminal
Apr 12 04:06
nsr@nsr-VirtualBox: ~/Desktop
Rubbish

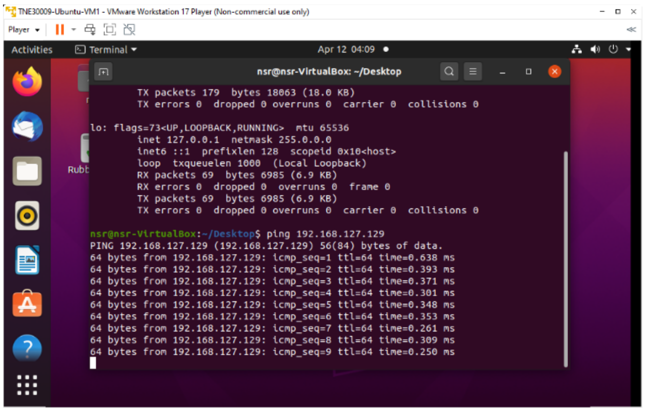
```
nsr@nsr-VirtualBox:~/Desktop$ ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.127.129  netmask 255.255.255.0  broadcast 192.168.127.255
        inet6 fe80::f0bb:e9d3:fa27:6bc2  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:42:e3:b2  txqueuelen 1000  (Ethernet)
        RX packets 177  bytes 46094 (46.0 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 140  bytes 16554 (16.5 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 65  bytes 7408 (7.4 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 65  bytes 7408 (7.4 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

nsr@nsr-VirtualBox:~/Desktop$ 
```

Figure 2. ifconfig used on VM2.

By running this command on both VMs, the addresses have been identified as:

VM1	192.168.127.128 255.255.255.0
VM2	192.168.127.129 255.255.255.0

To demonstrate connectivity between the two VMs, VM2 was pinged from VM1.

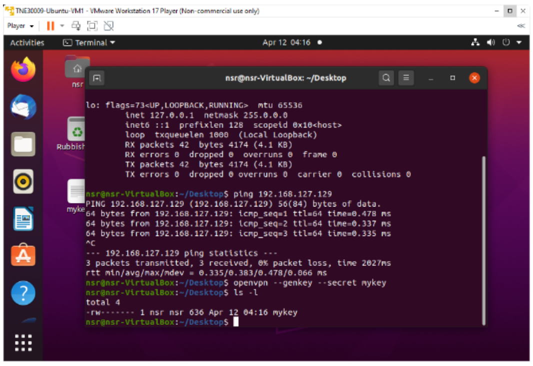

TNE30009-Ubuntu-VM1 - VMware Workstation 17 Player (Non-commercial use only)
Player
Activities
Terminal
Apr 12 04:09
nsr@nsr-VirtualBox: ~/Desktop
Rubb

```
        TX packets 179  bytes 18063 (18.0 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 69  bytes 6985 (6.9 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 69  bytes 6985 (6.9 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

nsr@nsr-VirtualBox:~/Desktop$ ping 192.168.127.129
PING 192.168.127.129 (192.168.127.129) 56(84) bytes of data.
64 bytes from 192.168.127.129: icmp_seq=1 ttl=64 time=0.638 ms
64 bytes from 192.168.127.129: icmp_seq=2 ttl=64 time=0.393 ms
64 bytes from 192.168.127.129: icmp_seq=3 ttl=64 time=0.371 ms
64 bytes from 192.168.127.129: icmp_seq=4 ttl=64 time=0.301 ms
64 bytes from 192.168.127.129: icmp_seq=5 ttl=64 time=0.348 ms
64 bytes from 192.168.127.129: icmp_seq=6 ttl=64 time=0.353 ms
64 bytes from 192.168.127.129: icmp_seq=7 ttl=64 time=0.261 ms
64 bytes from 192.168.127.129: icmp_seq=8 ttl=64 time=0.309 ms
64 bytes from 192.168.127.129: icmp_seq=9 ttl=64 time=0.250 ms
```

Figure 3. pinging VM2 from VM1.

Generate Shared Password

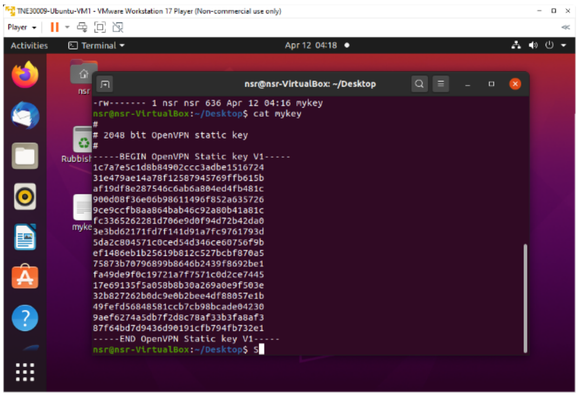
The next step in the lab is to generate a new key that will be used as the password.



```
lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 42  bytes 4174 (4.1 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 42  bytes 4174 (4.1 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

nsr@nsr-VirtualBox:~/Desktop$ ping 192.168.127.129
PING 192.168.127.129 (192.168.127.129) 56(84) bytes of data.
64 bytes from 192.168.127.129: icmp_seq=1 ttl=64 time=0.478 ms
64 bytes from 192.168.127.129: icmp_seq=2 ttl=64 time=0.337 ms
64 bytes from 192.168.127.129: icmp_seq=3 ttl=64 time=0.335 ms
^C
--- 192.168.127.129 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2027ms
rtt min/avg/max/mdev = 0.335/0.383/0.478/0.066 ms
nsr@nsr-VirtualBox:~/Desktop$ openvpn --genkey --secret mykey
nsr@nsr-VirtualBox:~/Desktop$ ls -l
total 4
-rw------- 1 nsr nsr 636 Apr 12 04:16 mykey
nsr@nsr-VirtualBox:~/Desktop$
```

Figure 4. generate shared password on VM1.

Now to examine the key using the cat command.

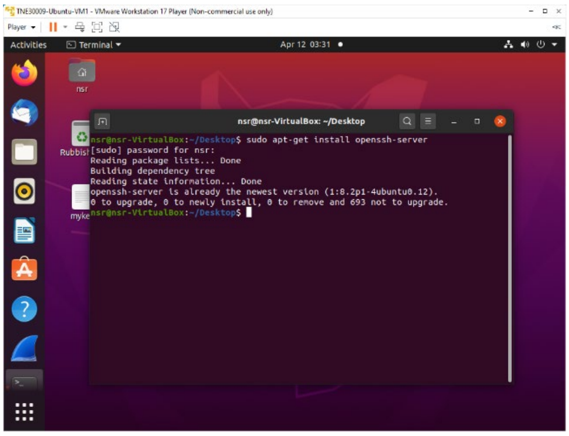


```
-rw------- 1 nsr nsr 636 Apr 12 04:16 mykey
nsr@nsr-VirtualBox:~/Desktop$ cat mykey
#
# 2048 bit OpenVPN static key
#
-----BEGIN OpenVPN Static key V1-----
1c7a7e5c1d8b84902ccc3adbe1516724
31e479ae14a78f12587945769ffb615b
af19df8e287546c6ab6a804ed4fb481c
900d08f36e06b98611496f852a635726
9ce9ccfb8aa864bab46c92a80b41a81c
fc3365262281d706e9d0f94d72b42da0
3e3bd62171fd7f141d91a7fc9761793d
5da2c804571c0ced54d346ce60756f9b
ef1486eb1b25619b812c527bcbf870a5
75873b70796899b8646b2439f8692be1
fa49de9f0c19721a7f7571c0d2ce7445
17e69135f5a058b8b30a269a0e9f503e
32b827262b0dc9e0b2bee4df88057e1b
49fefd56848581ccb7cb98bcade04230
9aef6274a5db7f2d8c78af33b3fa8af3
87f64bd7d9436d90191cfb794fb732e1
-----END OpenVPN Static key V1-----
nsr@nsr-VirtualBox:~/Desktop$ S
```

Figure 5. examine OpenVPN key on VM1.

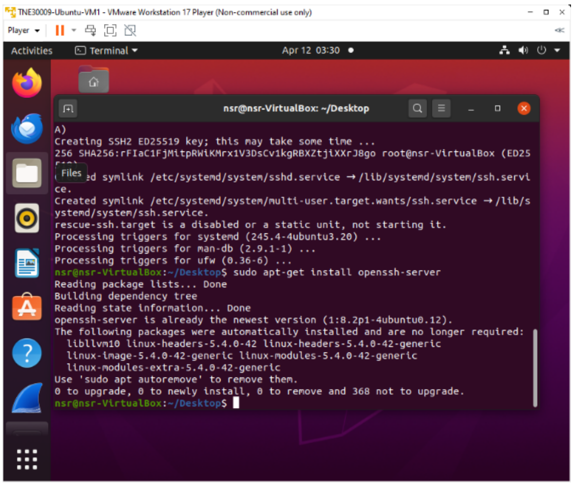
Transfer Shared Key To Other Machine

The next task is to install openssh on each of the machines.



```
nsr@nsr-VirtualBox:~/Desktop$ sudo apt-get install openssh-server
[sudo] password for nsr:
Reading package lists... Done
Building dependency tree
Reading state information... Done
openssh-server is already the newest version (1:8.2p1-4ubuntu0.12).
0 to upgrade, 0 to newly install, 0 to remove and 693 not to upgrade.
nsr@nsr-VirtualBox:~/Desktop$
```

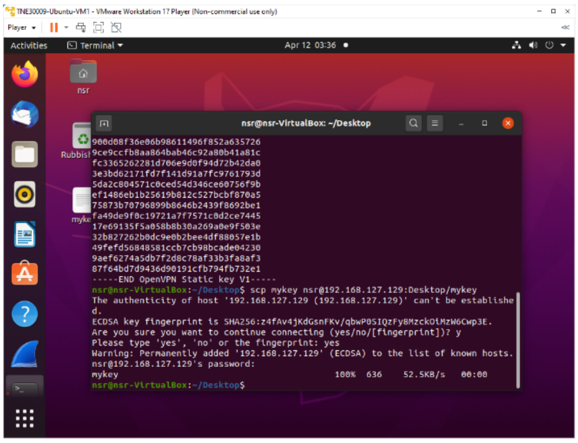
Figure 6. install openssh onto VM1.



```
A)
Creating SSH2 ED25519 key; this may take some time ...
256 SHA256:rFIaC1FjMitpRWiKMrx1V3DsCv1kgRBXZtjiXXrJ8go root@nsr-VirtualBox (ED25
519)
Created symlink /etc/systemd/system/sshd.service → /lib/systemd/system/ssh.servi
ce.
Created symlink /etc/systemd/system/multi-user.target.wants/ssh.service → /lib/s
ystemd/system/ssh.service.
rescue-ssh.target is a disabled or a static unit, not starting it.
Processing triggers for systemd (245.4-4ubuntu3.20) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for ufw (0.36-6) ...
nsr@nsr-VirtualBox:~/Desktop$ sudo apt-get install openssh-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
openssh-server is already the newest version (1:8.2p1-4ubuntu0.12).
The following packages were automatically installed and are no longer required:
  libllvm10 linux-headers-5.4.0-42 linux-headers-5.4.0-42-generic
  linux-image-5.4.0-42-generic linux-modules-5.4.0-42-generic
  linux-modules-extra-5.4.0-42-generic
Use 'sudo apt autoremove' to remove them.
0 to upgrade, 0 to newly install, 0 to remove and 368 not to upgrade.
nsr@nsr-VirtualBox:~/Desktop$
```

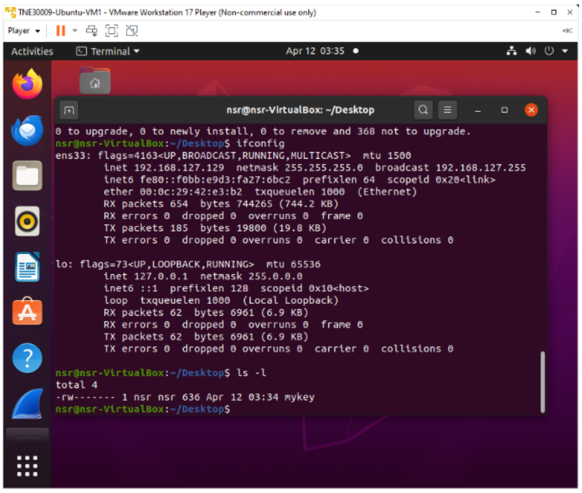
Figure 7. install openssh onto VM2.

Now the key is going to be sent to VM2 via ssh.



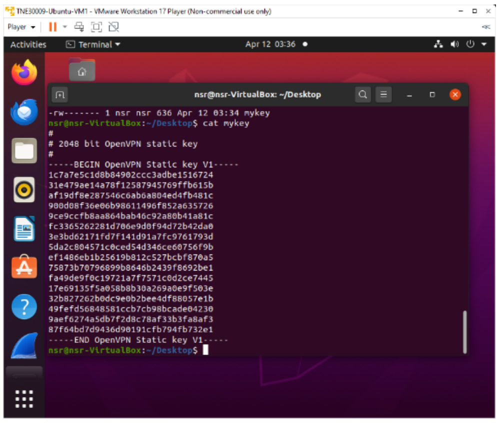
```
900d08f36e06b98611496f852a635726
9ce9ccfb8aa864bab46c92a80b41a81c
fc3365262281d706e9d0f94d72b42da0
3e3bd62171fd7f141d91a7fc9761793d
5da2c804571c0ced54d346ce60756f9b
ef1486eb1b25619b812c527bcbf870a5
75873b70796899b8646b2439f8692be1
fa49de9f0c19721a7f7571c0d2ce7445
17e69135f5a058b8b30a269a0e9f503e
32b827262b0dc9e0b2bee4df88057e1b
49fefd56848581ccb7cb98bcade04230
9aef6274a5db7f2d8c78af33b3fa8af3
87f64bd7d9436d90191cfb794fb732e1
-----END OpenVPN Static key V1-----
nsr@nsr-VirtualBox:~/Desktop$ scp mykey nsr@192.168.127.129:Desktop/mykey
The authenticity of host '192.168.127.129 (192.168.127.129)' can't be establishe
d.
ECDSA key fingerprint is SHA256:z4fAv4jKdGsnFKv/qbwP0SIQzFy8MzckOiMzW6Cwp3E.
Are you sure you want to continue connecting (yes/no/[fingerprint])? y
Please type 'yes', 'no' or the fingerprint: yes
Warning: Permanently added '192.168.127.129' (ECDSA) to the list of known hosts.
nsr@192.168.127.129's password:
mykey                                         100%  636    52.5KB/s   00:00
nsr@nsr-VirtualBox:~/Desktop$
```

Figure 8. sending key to VM2 via ssh.



```
0 to upgrade, 0 to newly install, 0 to remove and 368 not to upgrade.
nsr@nsr-VirtualBox:~/Desktop$ ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.127.129  netmask 255.255.255.0  broadcast 192.168.127.255
        inet6 fe80::f0bb:e9d3:fa27:6bc2  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:42:e3:b2  txqueuelen 1000  (Ethernet)
        RX packets 654  bytes 744265 (744.2 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 185  bytes 19800 (19.8 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 62  bytes 6961 (6.9 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 62  bytes 6961 (6.9 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

nsr@nsr-VirtualBox:~/Desktop$ ls -l
total 4
-rw------- 1 nsr nsr 636 Apr 12 03:34 mykey
nsr@nsr-VirtualBox:~/Desktop$
```

Figure 9. confirm the key file has been transferred to VM2.


TNE30009-Ubuntu-VM1 - VMware Workstation 17 Player (Non-commercial use only)
Activities
Terminal
Apr 12 03:36
nsr@nsr-VirtualBox: ~/Desktop

```
-rw------- 1 nsr nsr 636 Apr 12 03:34 mykey
nsr@nsr-VirtualBox:~/Desktop$ cat mykey
#
# 2048 bit OpenVPN static key
#
-----BEGIN OpenVPN Static key V1-----
1c7a7e5c1d8b84902ccc3adbe1516724
31e479ae14a78f12587945769ffb615b
af19df8e287546c6ab6a804ed4fb481c
900d08f36e06b98611496f852a635726
9ce9ccfb8aa864bab46c92a80b41a81c
fc3365262281d706e9d0f94d72b42da0
3e3bd62171fd7f141d91a7fc9761793d
5da2c804571c0ced54d346ce60756f9b
ef1486eb1b25619b812c527bcbf870a5
75873b70796899b8646b2439f8692be1
fa49de9f0c19721a7f7571c0d2ce7445
17e69135f5a058b8b30a269a0e9f503e
32b827262b0dc9e0b2bee4df88057e1b
49fefd56848581ccb7cb98bcade04230
9aef6274a5db7f2d8c78af33b3fa8af3
87f64bd7d9436d90191cfb794fb732e1
-----END OpenVPN Static key V1-----
nsr@nsr-VirtualBox:~/Desktop$ 
```

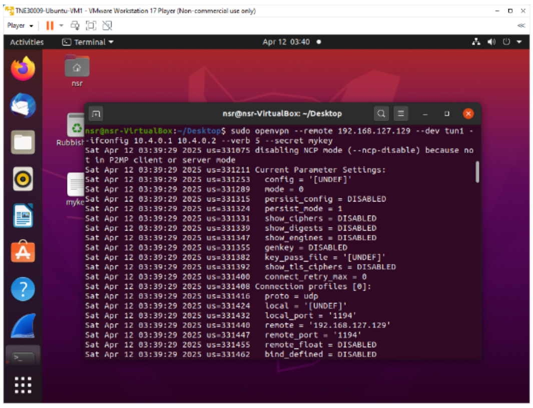
Figure 10. examine the key to confirm it is the same one.

Set Up Encrypted Tunnel

The next step in the lab is to finally configure the tunnels between the machines using OpenVPN.

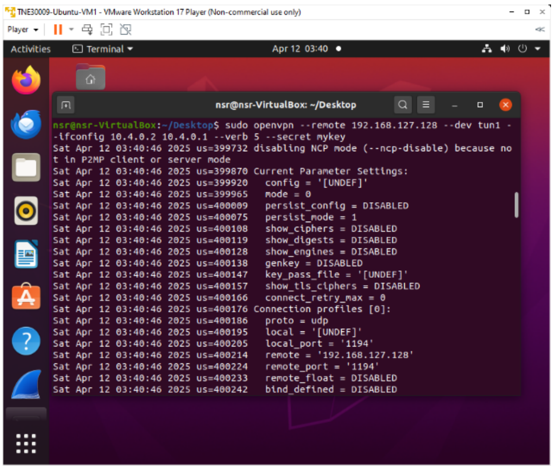
The ip addresses the tunnels will be using are:

VM1 Tunnel IP	10.4.0.1
VM2 Tunnel IP	10.4.0.2


TNE30009-Ubuntu-VM1 - VMware Workstation 17 Player (Non-commercial use only)
Activities
Terminal
Apr 12 03:40
nsr
Rubbish
myke
nsr@nsr-VirtualBox: ~/Desktop

```
nsr@nsr-VirtualBox:~/Desktop$ sudo openvpn --remote 192.168.127.129 --dev tun1 -
-ifconfig 10.4.0.1 10.4.0.2 --verb 5 --secret mykey
Sat Apr 12 03:39:29 2025 us=331075 disabling NCP mode (--ncp-disable) because no
t in P2MP client or server mode
Sat Apr 12 03:39:29 2025 us=331211 Current Parameter Settings:
Sat Apr 12 03:39:29 2025 us=331253   config = '[UNDEF]'
Sat Apr 12 03:39:29 2025 us=331289   mode = 0
Sat Apr 12 03:39:29 2025 us=331315   persist_config = DISABLED
Sat Apr 12 03:39:29 2025 us=331324   persist_mode = 1
Sat Apr 12 03:39:29 2025 us=331331   show_ciphers = DISABLED
Sat Apr 12 03:39:29 2025 us=331339   show_digests = DISABLED
Sat Apr 12 03:39:29 2025 us=331347   show_engines = DISABLED
Sat Apr 12 03:39:29 2025 us=331355   genkey = DISABLED
Sat Apr 12 03:39:29 2025 us=331382   key_pass_file = '[UNDEF]'
Sat Apr 12 03:39:29 2025 us=331392   show_tls_ciphers = DISABLED
Sat Apr 12 03:39:29 2025 us=331400   connect_retry_max = 0
Sat Apr 12 03:39:29 2025 us=331408 Connection profiles [0]:
Sat Apr 12 03:39:29 2025 us=331416   proto = udp
Sat Apr 12 03:39:29 2025 us=331424   local = '[UNDEF]'
Sat Apr 12 03:39:29 2025 us=331432   local_port = '1194'
Sat Apr 12 03:39:29 2025 us=331440   remote = '192.168.127.129'
Sat Apr 12 03:39:29 2025 us=331447   remote_port = '1194'
Sat Apr 12 03:39:29 2025 us=331455   remote_float = DISABLED
Sat Apr 12 03:39:29 2025 us=331462   bind_defined = DISABLED
```

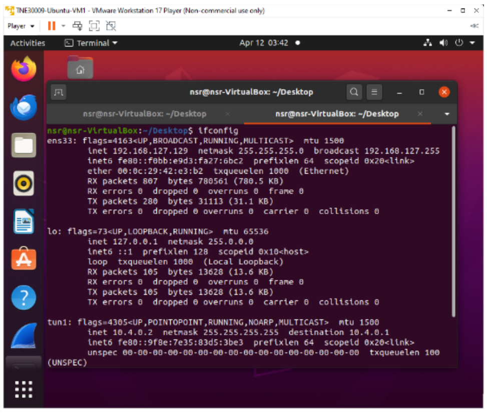
Figure 11. setup tunnel from VM1.


TNE30009-Ubuntu-VM1 - VMware Workstation 17 Player (Non-commercial use only)
Activities
Terminal
Apr 12 03:40
nsr@nsr-VirtualBox: ~/Desktop

```
nsr@nsr-VirtualBox:~/Desktop$ sudo openvpn --remote 192.168.127.128 --dev tun1 -
-ifconfig 10.4.0.2 10.4.0.1 --verb 5 --secret mykey
Sat Apr 12 03:40:46 2025 us=399732 disabling NCP mode (--ncp-disable) because no
t in P2MP client or server mode
Sat Apr 12 03:40:46 2025 us=399870 Current Parameter Settings:
Sat Apr 12 03:40:46 2025 us=399920   config = '[UNDEF]'
Sat Apr 12 03:40:46 2025 us=399965   mode = 0
Sat Apr 12 03:40:46 2025 us=400009   persist_config = DISABLED
Sat Apr 12 03:40:46 2025 us=400075   persist_mode = 1
Sat Apr 12 03:40:46 2025 us=400108   show_ciphers = DISABLED
Sat Apr 12 03:40:46 2025 us=400119   show_digests = DISABLED
Sat Apr 12 03:40:46 2025 us=400128   show_engines = DISABLED
Sat Apr 12 03:40:46 2025 us=400138   genkey = DISABLED
Sat Apr 12 03:40:46 2025 us=400147   key_pass_file = '[UNDEF]'
Sat Apr 12 03:40:46 2025 us=400157   show_tls_ciphers = DISABLED
Sat Apr 12 03:40:46 2025 us=400166   connect_retry_max = 0
Sat Apr 12 03:40:46 2025 us=400176 Connection profiles [0]:
Sat Apr 12 03:40:46 2025 us=400186   proto = udp
Sat Apr 12 03:40:46 2025 us=400195   local = '[UNDEF]'
Sat Apr 12 03:40:46 2025 us=400205   local_port = '1194'
Sat Apr 12 03:40:46 2025 us=400214   remote = '192.168.127.128'
Sat Apr 12 03:40:46 2025 us=400224   remote_port = '1194'
Sat Apr 12 03:40:46 2025 us=400233   remote_float = DISABLED
Sat Apr 12 03:40:46 2025 us=400242   bind_defined = DISABLED
```

Figure 12. setup tunnel from VM2.

Using ifconfig to check that there is a tunnel interface (tun1).

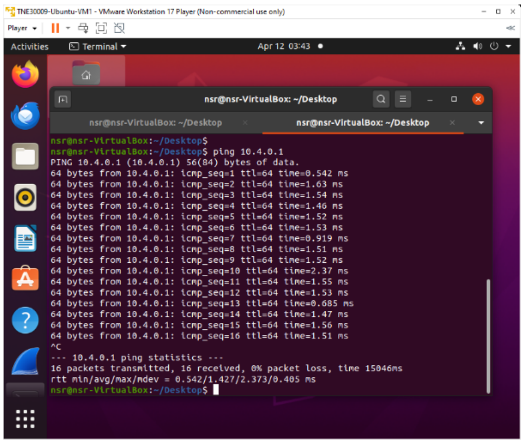

TNE30009-Ubuntu-VM1 - VMware Workstation 17 Player (Non-commercial use only)
Activities
Terminal
Apr 12 03:42
nsr@nsr-VirtualBox: ~/Desktop
nsr@nsr-VirtualBox: ~/Desktop
nsr@nsr-VirtualBox: ~/Desktop

```
nsr@nsr-VirtualBox:~/Desktop$ ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.127.129  netmask 255.255.255.0  broadcast 192.168.127.255
        inet6 fe80::f0bb:e9d3:fa27:6bc2  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:42:e3:b2  txqueuelen 1000  (Ethernet)
        RX packets 807  bytes 780561 (780.5 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 280  bytes 31113 (31.1 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 105  bytes 13628 (13.6 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 105  bytes 13628 (13.6 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

tun1: flags=4305<UP,POINTOPOINT,RUNNING,NOARP,MULTICAST>  mtu 1500
        inet 10.4.0.2  netmask 255.255.255.255  destination 10.4.0.1
        inet6 fe80::9f8e:7e35:83d5:3be3  prefixlen 64  scopeid 0x20<link>
        unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00  txqueuelen 100
(UNSPEC)
```

Figure 13. ifconfig on VM2.

Using the VPN Tunnel

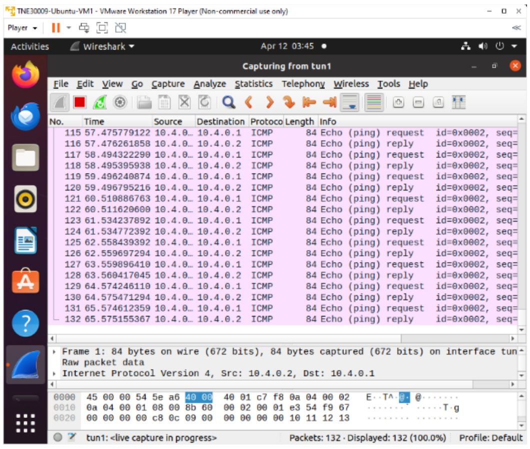
To check the connectivity between the two machines using the tunnel, VM2 pings tunnel destination on VM1.


TNE30009-Ubuntu-VM1 - VMware Workstation 17 Player (Non-commercial use only)
Activities
Terminal
Apr 12 03:43
nsr@nsr-VirtualBox: ~/Desktop
nsr@nsr-VirtualBox: ~/Desktop
nsr@nsr-VirtualBox: ~/Desktop

```
nsr@nsr-VirtualBox:~/Desktop$
nsr@nsr-VirtualBox:~/Desktop$ ping 10.4.0.1
PING 10.4.0.1 (10.4.0.1) 56(84) bytes of data.
64 bytes from 10.4.0.1: icmp_seq=1 ttl=64 time=0.542 ms
64 bytes from 10.4.0.1: icmp_seq=2 ttl=64 time=1.63 ms
64 bytes from 10.4.0.1: icmp_seq=3 ttl=64 time=1.54 ms
64 bytes from 10.4.0.1: icmp_seq=4 ttl=64 time=1.46 ms
64 bytes from 10.4.0.1: icmp_seq=5 ttl=64 time=1.52 ms
64 bytes from 10.4.0.1: icmp_seq=6 ttl=64 time=1.53 ms
64 bytes from 10.4.0.1: icmp_seq=7 ttl=64 time=0.919 ms
64 bytes from 10.4.0.1: icmp_seq=8 ttl=64 time=1.51 ms
64 bytes from 10.4.0.1: icmp_seq=9 ttl=64 time=1.52 ms
64 bytes from 10.4.0.1: icmp_seq=10 ttl=64 time=2.37 ms
64 bytes from 10.4.0.1: icmp_seq=11 ttl=64 time=1.55 ms
64 bytes from 10.4.0.1: icmp_seq=12 ttl=64 time=1.53 ms
64 bytes from 10.4.0.1: icmp_seq=13 ttl=64 time=0.685 ms
64 bytes from 10.4.0.1: icmp_seq=14 ttl=64 time=1.47 ms
64 bytes from 10.4.0.1: icmp_seq=15 ttl=64 time=1.56 ms
64 bytes from 10.4.0.1: icmp_seq=16 ttl=64 time=1.51 ms
^C
--- 10.4.0.1 ping statistics ---
16 packets transmitted, 16 received, 0% packet loss, time 15046ms
rtt min/avg/max/mdev = 0.542/1.427/2.373/0.405 ms
nsr@nsr-VirtualBox:~/Desktop$ 
```

Figure 14. VM2 ping tunnel destination address.

Wireshark is now used to monitor the traffic and verify encryption.



```
No.  Time         Source    Destination Protoco Length Info
 115 57.475779122 10.4.0… 10.4.0.1   ICMP        84 Echo (ping) request  id=0x0002, seq=
 116 57.476261858 10.4.0… 10.4.0.2   ICMP        84 Echo (ping) reply    id=0x0002, seq=
 117 58.494322290 10.4.0… 10.4.0.1   ICMP        84 Echo (ping) request  id=0x0002, seq=
 118 58.495395938 10.4.0… 10.4.0.2   ICMP        84 Echo (ping) reply    id=0x0002, seq=
 119 59.496240874 10.4.0… 10.4.0.1   ICMP        84 Echo (ping) request  id=0x0002, seq=
 120 59.496795216 10.4.0… 10.4.0.2   ICMP        84 Echo (ping) reply    id=0x0002, seq=
 121 60.510886763 10.4.0… 10.4.0.1   ICMP        84 Echo (ping) request  id=0x0002, seq=
 122 60.511620600 10.4.0… 10.4.0.2   ICMP        84 Echo (ping) reply    id=0x0002, seq=
 123 61.534237892 10.4.0… 10.4.0.1   ICMP        84 Echo (ping) request  id=0x0002, seq=
 124 61.534772392 10.4.0… 10.4.0.2   ICMP        84 Echo (ping) reply    id=0x0002, seq=
 125 62.558439392 10.4.0… 10.4.0.1   ICMP        84 Echo (ping) request  id=0x0002, seq=
 126 62.559697294 10.4.0… 10.4.0.2   ICMP        84 Echo (ping) reply    id=0x0002, seq=
 127 63.559896410 10.4.0… 10.4.0.1   ICMP        84 Echo (ping) request  id=0x0002, seq=
 128 63.560417045 10.4.0… 10.4.0.2   ICMP        84 Echo (ping) reply    id=0x0002, seq=
 129 64.574246110 10.4.0… 10.4.0.1   ICMP        84 Echo (ping) request  id=0x0002, seq=
 130 64.575471294 10.4.0… 10.4.0.2   ICMP        84 Echo (ping) reply    id=0x0002, seq=
 131 65.574612359 10.4.0… 10.4.0.1   ICMP        84 Echo (ping) request  id=0x0002, seq=
 132 65.575155367 10.4.0… 10.4.0.2   ICMP        84 Echo (ping) reply    id=0x0002, seq=

Frame 1: 84 bytes on wire (672 bits), 84 bytes captured (672 bits) on interface tun
Raw packet data
Internet Protocol Version 4, Src: 10.4.0.2, Dst: 10.4.0.1

0000  45 00 00 54 5e a6 40 00  40 01 c7 f8 0a 04 00 02   E··T^·@· @·······
0010  0a 04 00 01 08 00 8b 60  00 02 00 01 e3 54 f9 67   ········ ·····T·g
0020  00 00 00 00 c8 0c 09 00  00 00 00 00 10 11 12 13   ········ ········

tun1: <live capture in progress>      Packets: 132 · Displayed: 132 (100.0%)      Profile: Default
```

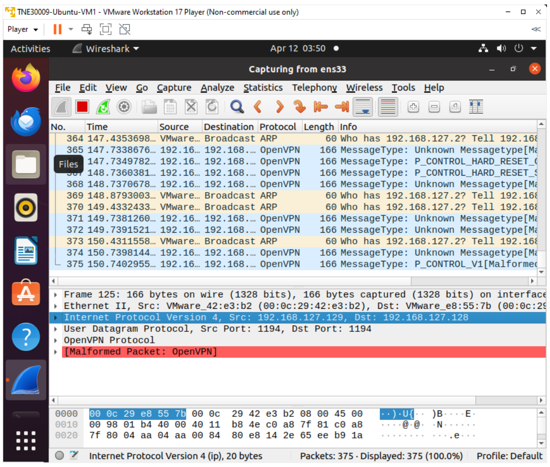
TNE30009-Ubuntu-VM1 - VMware Workstation 17 Player (Non-commercial use only)
Player
Activities
Wireshark
Apr 12 03:45
Capturing from tun1
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Figure 15. wireshark monitoring tun1 on VM2.

a) Is the traffic encrypted? Why?

No, the traffic is not encrypted as we can see the ICMP protocol.

Now wireshark is used to monitor the ethernet interface on VM2.



```
No.  Time         Source    Destination Protocol Length Info
 364 147.4353698… VMware… Broadcast ARP           60 Who has 192.168.127.2? Tell 192.168
 365 147.7338676… 192.16… 192.168.… OpenVPN      166 MessageType: Unknown Messagetype[Ma
 366 147.7349782… 192.16… 192.168.… OpenVPN      166 MessageType: P_CONTROL_HARD_RESET_C
 367 148.7360381… 192.16… 192.168.… OpenVPN      166 MessageType: P_CONTROL_HARD_RESET_S
 368 148.7370678… 192.16… 192.168.… OpenVPN      166 MessageType: Unknown Messagetype[Ma
 369 148.8793003… VMware… Broadcast ARP           60 Who has 192.168.127.2? Tell 192.168
 370 149.4332433… VMware… Broadcast ARP           60 Who has 192.168.127.2? Tell 192.168
 371 149.7381260… 192.16… 192.168.… OpenVPN      166 MessageType: Unknown Messagetype[Ma
 372 149.7391521… 192.16… 192.168.… OpenVPN      166 MessageType: Unknown Messagetype[Ma
 373 150.4311558… VMware… Broadcast ARP           60 Who has 192.168.127.2? Tell 192.168
 374 150.7398144… 192.16… 192.168.… OpenVPN      166 MessageType: Unknown Messagetype[Ma
 375 150.7402955… 192.16… 192.168.… OpenVPN      166 MessageType: P_CONTROL_V1[Malformed

Frame 125: 166 bytes on wire (1328 bits), 166 bytes captured (1328 bits) on interface
Ethernet II, Src: VMware_42:e3:b2 (00:0c:29:42:e3:b2), Dst: VMware_e8:55:7b (00:0c:29
Internet Protocol Version 4, Src: 192.168.127.129, Dst: 192.168.127.128
User Datagram Protocol, Src Port: 1194, Dst Port: 1194
OpenVPN Protocol
[Malformed Packet: OpenVPN]

0000  00 0c 29 e8 55 7b 00 0c  29 42 e3 b2 08 00 45 00   ··)·U{·· )B····E·
0010  00 98 01 b4 40 00 40 11  b8 4e c0 a8 7f 81 c0 a8   ····@·@· ·N······
0020  7f 80 04 aa 04 aa 00 84  80 e8 14 2e 65 ee b9 1a   ········ ···.e···

Internet Protocol Version 4 (ip), 20 bytes      Packets: 375 · Displayed: 375 (100.0%)      Profile: Default
```

TNE30009-Ubuntu-VM1 - VMware Workstation 17 Player (Non-commercial use only)
Player
Activities
Wireshark
Apr 12 03:50
Capturing from ens33
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Files

Figure 16. wireshark monitoring ethernet interface on VM2.

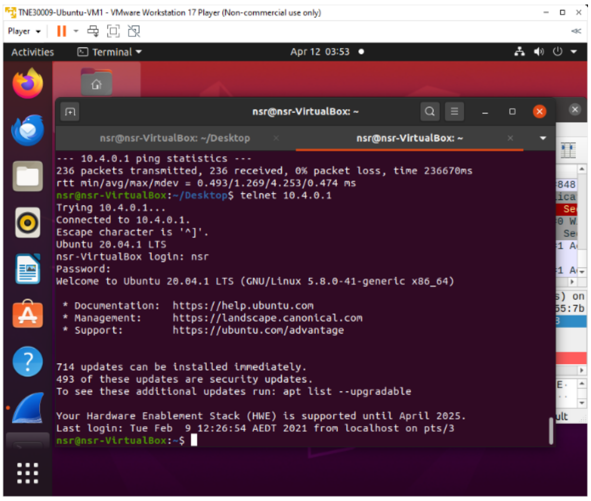
b) Is the traffic encrypted?

Yes, the traffic is encrypted on this interface.

c) How do you explain what you see?

The protocol of the traffic is OpenVPN and the packets have “Malformed Packet: OpenVPN” on them, indicating that it is encrypted.

Now the lab says to telnet from VM2 into the tunnel address of VM1.

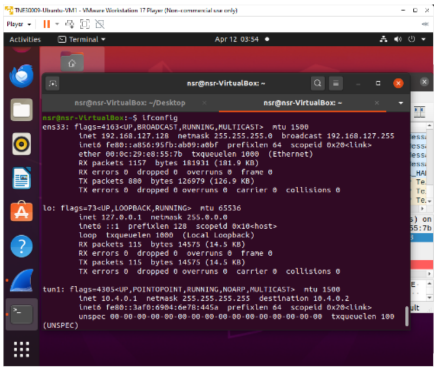


```
--- 10.4.0.1 ping statistics ---
236 packets transmitted, 236 received, 0% packet loss, time 236670ms
rtt min/avg/max/mdev = 0.493/1.269/4.253/0.474 ms
nsr@nsr-VirtualBox:~/Desktop$ telnet 10.4.0.1
Trying 10.4.0.1...
Connected to 10.4.0.1.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
nsr-VirtualBox login: nsr
Password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.8.0-41-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage


714 updates can be installed immediately.
493 of these updates are security updates.
To see these additional updates run: apt list --upgradable

Your Hardware Enablement Stack (HWE) is supported until April 2025.
Last login: Tue Feb  9 12:26:54 AEDT 2021 from localhost on pts/3
nsr@nsr-VirtualBox:~$
```

TNE30009-Ubuntu-VM1 - VMware Workstation 17 Player (Non-commercial use only)
Player
Activities
Terminal
Apr 12 03:53
nsr@nsr-VirtualBox: ~
nsr@nsr-VirtualBox: ~/Desktop
nsr@nsr-VirtualBox: ~

Figure 17. VM2 telnetting into tunnel address on VM1.

To verify that the telnet worked, use ifconfig to check the ip address.



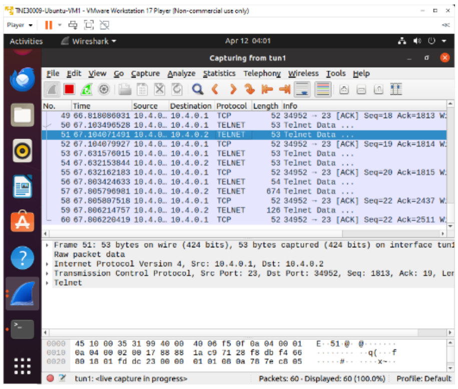
```
nsr@nsr-VirtualBox:~$ ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.127.128  netmask 255.255.255.0  broadcast 192.168.127.255
        inet6 fe80::a856:95fb:ab09:a0bf  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:e8:55:7b  txqueuelen 1000  (Ethernet)
        RX packets 1157  bytes 181931 (181.9 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 880  bytes 126979 (126.9 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 115  bytes 14575 (14.5 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 115  bytes 14575 (14.5 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

tun1: flags=4305<UP,POINTOPOINT,RUNNING,NOARP,MULTICAST>  mtu 1500
        inet 10.4.0.1  netmask 255.255.255.255  destination 10.4.0.2
        inet6 fe80::3af0:6904:6e78:445a  prefixlen 64  scopeid 0x20<link>
        unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00  txqueuelen 100
(UNSPEC)
```

TNE30009-Ubuntu-VM1 - VMware Workstation 17 Player (Non-commercial use only)
Player
Activities
Terminal
Apr 12 03:54
nsr@nsr-VirtualBox: ~
nsr@nsr-VirtualBox: ~/Desktop
nsr@nsr-VirtualBox: ~

Figure 18. ifconfig on VM2 showing successful telnet.

d) Which interface is traffic encrypted and which interface is not encrypted?

The tun1 interface appears unencrypted when monitored with wireshark.

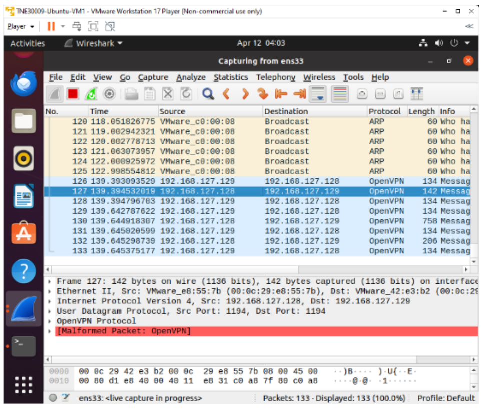

TNE30009-Ubuntu-VM1 - VMware Workstation 17 Player (Non-commercial use only)
Player
Activities
Wireshark
Apr 12 04:01
Capturing from tun1
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

```
No.   Time         Source     Destination Protocol Length Info
   49 66.818086031 10.4.0…    10.4.0.1    TCP         52 34952 → 23 [ACK] Seq=18 Ack=1813 W:
   50 67.103496528 10.4.0…    10.4.0.1    TELNET      53 Telnet Data ...
   51 67.104071491 10.4.0…    10.4.0.2    TELNET      53 Telnet Data ...
   52 67.104079927 10.4.0…    10.4.0.1    TCP         52 34952 → 23 [ACK] Seq=19 Ack=1814 W:
   53 67.631576015 10.4.0…    10.4.0.1    TELNET      53 Telnet Data ...
   54 67.632153844 10.4.0…    10.4.0.2    TELNET      53 Telnet Data ...
   55 67.632162183 10.4.0…    10.4.0.1    TCP         52 34952 → 23 [ACK] Seq=20 Ack=1815 W:
   56 67.803424633 10.4.0…    10.4.0.1    TELNET      54 Telnet Data ...
   57 67.805796981 10.4.0…    10.4.0.2    TELNET     674 Telnet Data ...
   58 67.805807518 10.4.0…    10.4.0.1    TCP         52 34952 → 23 [ACK] Seq=22 Ack=2437 W:
   59 67.806214757 10.4.0…    10.4.0.2    TELNET     126 Telnet Data ...
   60 67.806220419 10.4.0…    10.4.0.1    TCP         52 34952 → 23 [ACK] Seq=22 Ack=2511 W:

› Frame 51: 53 bytes on wire (424 bits), 53 bytes captured (424 bits) on interface tun1
  Raw packet data
› Internet Protocol Version 4, Src: 10.4.0.1, Dst: 10.4.0.2
› Transmission Control Protocol, Src Port: 23, Dst Port: 34952, Seq: 1813, Ack: 19, Len
› Telnet

0000  45 10 00 35 31 99 40 00  40 06 f5 0f 0a 04 00 01   E··51·@· @·······
0010  0a 04 00 02 00 17 88 88  1a c9 71 28 f8 db f4 66   ········ ··q(···f
0020  80 18 01 fd dc 23 00 00  01 01 08 0a 78 7e c8 05   ·····#·· ····x~··

tun1: <live capture in progress>        Packets: 60 · Displayed: 60 (100.0%)   Profile: Default
```

Figure 19. wireshark monitoring interface tun1.

This is because wireshark is monitoring traffic within the tunnel itself, allowing visibility into the unencrypted packets.

Interface ens33 appears encrypted when monitored in wireshark.


TNE30009-Ubuntu-VM1 - VMware Workstation 17 Player (Non-commercial use only)
Player
Activities
Wireshark
Apr 12 04:03
Capturing from ens33
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

```
No.   Time          Source            Destination       Protocol Length Info
  120 118.051826775 VMware_c0:00:08   Broadcast         ARP          60 Who ha
  121 119.002942321 VMware_c0:00:08   Broadcast         ARP          60 Who ha
  122 120.002778713 VMware_c0:00:08   Broadcast         ARP          60 Who ha
  123 121.063073957 VMware_c0:00:08   Broadcast         ARP          60 Who ha
  124 122.000925972 VMware_c0:00:08   Broadcast         ARP          60 Who ha
  125 122.998554812 VMware_c0:00:08   Broadcast         ARP          60 Who ha
  126 139.393093529 192.168.127.129   192.168.127.128   OpenVPN     134 Messag
  127 139.394532019 192.168.127.128   192.168.127.129   OpenVPN     142 Messag
  128 139.394796703 192.168.127.129   192.168.127.128   OpenVPN     134 Messag
  129 139.642787622 192.168.127.129   192.168.127.128   OpenVPN     134 Messag
  130 139.644918307 192.168.127.128   192.168.127.129   OpenVPN     758 Messag
  131 139.645020599 192.168.127.129   192.168.127.128   OpenVPN     134 Messag
  132 139.645298739 192.168.127.128   192.168.127.129   OpenVPN     206 Messag
  133 139.645375177 192.168.127.129   192.168.127.128   OpenVPN     134 Messag

› Frame 127: 142 bytes on wire (1136 bits), 142 bytes captured (1136 bits) on interface
› Ethernet II, Src: VMware_e8:55:7b (00:0c:29:e8:55:7b), Dst: VMware_42:e3:b2 (00:0c:29
› Internet Protocol Version 4, Src: 192.168.127.128, Dst: 192.168.127.129
› User Datagram Protocol, Src Port: 1194, Dst Port: 1194
› OpenVPN Protocol
› [Malformed Packet: OpenVPN]

0000  00 0c 29 42 e3 b2 00 0c  29 e8 55 7b 08 00 45 00   ··)B···· )·U{··E·
0010  00 80 d1 e8 40 00 40 11  e8 31 c0 a8 7f 80 c0 a8   ····@·@· ·1······

ens33: <live capture in progress>      Packets: 133 · Displayed: 133 (100.0%)   Profile: Default
```

Figure 20. wireshark monitoring interface ens33.

This is because wireshark is monitoring the traffic outside the tunnel where the data is encrypted.

What Is A VPN Tunnel?

A VPN is a secure, encrypted connection established over public networks between two or more

Conclusion

This lab successfully demonstrates the implementation of a VPN tunnel using OpenVPN between two virtual Ubuntu machines. By using a shared key and configuring the tunnel interfaces, an encrypted connection was created. By performing this experiment, it demonstrated the differences between the tunnel traffic and the physical interface traffic. Wireshark, telnet and pinging was used to verify that the connection was encrypted and validate connectivity between the two virtual machines.

References

[1] “What is a VPN? A Complete Guide to Virtual Private Networks,” *Palo Alto Networks*, 2015. https://www.paloaltonetworks.com.au/cyberpedia/what-is-a-vpn#What-is-the-difference-between-personal-and-business-VPNs.

[2] Kaspersky, “What is a VPN and how does it work?,” *Kaspersky*, Nov. 03. 2020. https://www.kaspersky.com/resource-center/definitions/what-is-a-vpn.

[3] Cloudflare, “VPN security: How VPNs help secure data and control access.” *Cloudflare.com*, 2024. https://www.cloudflare.com/learning/access-management/vpn-security/.

[4] R. Mendenhall, “Steps for Selecting and Setting up a Small Business Vpn,” *Yarro.org*, Oct. 05, 2022. https://yarro.org/steps-for-selecting-and-setting-up-a-small-business-vpn/.