TNE30009/TNE80009

Laboratory Session 3

1. Introduction

The purpose of this lab is to introduce you to VPN technology. We will set up a simple encrypted VPN between the two Ubuntu machines using Linux VPN software called OpenVPN. OpenVPN is a very flexible system enabling a full range of VPNs to be set up with similar capabilities to IPSec, however we will only be setting up a simple encrypted VPN tunnel between the two machines.

This work is to be carried out using the virtual machines used in the previous labs.

You will use OpenVPN software to generate a shared secret key, scp to copy the key to the other machine and OpenVPN to set up the tunnel. All software for this lab is already installed on the Virtual Machines you downloaded for lab 1.

You may be asked for a password. All passwords are `user`.

2. Host configuration

The two hosts should have been configured with IP addresses as in the previous labs.

Take note of the two IP addresses. On each host open a terminal by left clicking on the mouse and choosing "Open in Terminal". Then type `ifconfig`

"ifconfig" lists the network interfaces and optionally allows them to be configured. You should see something similar to:

```
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.64.19  netmask 255.255.255.0  broadcast
192.168.64.255
        inet6 fe80::39db:1cd5:a303:827f  prefixlen 64  scopeid
0x20<link>
        ether 08:00:27:0f:b2:41  txqueuelen 1000  (Ethernet)
        RX packets 495  bytes 99823 (99.8 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 77  bytes 10286 (10.2 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 261  bytes 25636 (25.6 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 261  bytes 25636 (25.6 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

On this machine there is an ethernet interface (enp0s3) and a loopback interface. The ethernet interface has an IP address of 192.168.64.19. Take note of the IP address on each machine

Now check connectivity between both hosts with a ping. In my system the two IP addresses are 192.168.64.19 and 192.168.64.20 but yours will probably be different.

From one host ping the other to ensure you have connectivity. You should see something similar to the following.

```
nsr@nsr-VirtualBox:~/Desktop$ ping 192.168.64.20
PING 192.168.64.20 (192.168.64.20) 56(84) bytes of data.
64 bytes from 192.168.64.20: icmp_seq=1 ttl=64 time=0.570 ms
```

```
64 bytes from 192.168.64.20: icmp_seq=2 ttl=64 time=1.25 ms
64 bytes from 192.168.64.20: icmp_seq=3 ttl=64 time=1.30 ms
64 bytes from 192.168.64.20: icmp_seq=4 ttl=64 time=1.30 ms
^C
```

Note that in Linux, unlike Windows, you will need to stop the ping with a ctrl-c.

3. Generate the shared password

Once connectivity is established generate the shared password on Ubuntu1.

Open a command prompt window and type:

```
openvpn --genkey --secret mykey
```

This will create a 256 byte key in a file called "mykey". Check to see that the file exists by typing the following (note that the parameter is the letter l, not the digit 1).

```
ls -l
```

You should see something similar to the following:

```
-rw------- 1 nsr nsr 636 Mar 28 11:36 mykey
```

"ls" is a command that lists files within a directory. "ls -l" instructs it to list them in long format. The first set of values (-rw) lists the permissions associated with the file. In this case only the owner can read or write the file. Next is the owner of the file, its group (both "nsr") its size, date and time created, and finally the name.

Now examine the file. Type:

```
cat mykey
```

"cat" is the "concatenate command which rather confusingly is most commonly used in Linux to list the contents of files. You should see something like the following:

```
#
# 2048 bit OpenVPN static key
#
-----BEGIN OpenVPN Static key V1-----
8c9e1947a331581e7f66b51041c48e6c
26da8d4bde50c98b80a6376cc55cb02b
a72cf3b7c2eb29c2cc5d7ac40c986bae
fa3fdaf533156199fe6a3caef06dabfd
06335281315905c14a635661c351b234
582840e9420981d614d31443018691f4
f3c1a9a12db96b234e2c7a724edbc02b
b14a96781051b046c2b3530a264c420f
aa902e35600993a581abe957710380d7
c9992985baf89a1c008e4c20ea2eb0ce
8892958ba6e14b2f8e4b35030ca7333f
71d457fd28f651dd3a8ace100c8e16d6
bdb6133c32b4760196f59937d0e3247d
647abe2ce16a3e2ca318476998b881aa
40d18ed6058e509d1544692a487d63e0
7c695a221810fbd0d452197002751e09
-----END OpenVPN Static key V1-----
```

4. Transfer the shared key to the other machine

We will use Secure Copy (scp) to copy the key from the machine it was generated on to the other machine. At the end of this process, both machines should have the same file “mykey” with the same content.

You need to install ssh (which includes scp) on each host. Type the following on both machines:

```
sudo apt update

sudo apt-get install openssh-server
```

You will be asked to confirm that you want to download and install it. Answer ‘yes’.

Once ssh is installed on both machines, go back to the machine you created mykey on and type the following, substituting the appropriate IP address:

```
scp mykey nsr@192.168.64.20:Desktop/mykey
```

Note the “:” at the end of the ip address. This tells the system that you are sending the file to another machine. It is very important you include it. You will be asked for a password which is “user” and may be asked to accept a “known host”. Type “yes”. You should see something similar to the following:

```
nsr@nsr-VirtualBox:~/Desktop$ scp mykey
nsr@192.168.64.20:Desktop/mykey
The authenticity of host '192.168.64.20 (192.168.64.20)' can't be
established.
ECDSA key fingerprint is
SHA256:nNQzZVsV131v2o6Xk12Dw4VLnShfheqSeRmqXyEFAJg.
Are you sure you want to continue connecting (yes/no/[fingerprint])?
yes
Warning: Permanently added '192.168.64.20' (ECDSA) to the list of
known hosts.
nsr@192.168.64.20's password:
mykey                                              100%  636
701.9KB/s   00:00
nsr@nsr-VirtualBox:~/Desktop$
```

Check that the file is on the other machine by using “ls” and “cat” as you did earlier on the first machine.

Once you have the shared secret key on both machines you can now set up an encrypted tunnel between them.

5. Set up encrypted tunnel

On Ubuntu1

In this example Ubuntu1 has IP address 192.168.64.19 and Ubuntu2 has IP address 192.168.64.20. Type the following from Ubuntu1 (substituting appropriate IP address). The remote IP address is that of Ubuntu 2:

```
sudo openvpn --remote 192.168.64.20 --dev tun1 --ifconfig 10.4.0.1
10.4.0.2 --verb 5 --secret mykey
```

This command sets up a tunnel from Ubuntu 1 into Ubuntu 2. The IP addresses of the tunnel end points are 10.4.0.1 on Ubuntu1 and 10.4.0.2 on Ubuntu2. Traffic going via the tunnel is encrypted using the key contained in the file “mykey”. “verb 5” means Verbose mode level 5. Level 10 gives a detailed dump of all traffic while level 0 means silent mode. Level 5 is somewhere in-between reporting on when it receives or sends messages.

On Ubuntu2

Set up the tunnel in the reverse direction so that traffic can travel in both directions.

```
sudo openvpn --remote 192.168.64.19 --dev tun1 --ifconfig 10.4.0.2
10.4.0.1 --verb 5 --secret mykey
```

You should now have an encrypted bi-directional tunnel between both machines. Open another terminal and do an ifconfig on Ubuntu2. You should see something similar to this:

```
nsr@nsr-VirtualBox:~/Desktop$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.64.20  netmask 255.255.255.0  broadcast
192.168.64.255
        inet6 fe80::6fed:ecc0:f656:630c  prefixlen 64  scopeid
0x20<link>
        ether 08:00:27:22:64:b1  txqueuelen 1000  (Ethernet)
        RX packets 1094  bytes 216457 (216.4 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 236  bytes 31870 (31.8 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 449  bytes 50497 (50.4 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 449  bytes 50497 (50.4 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::35d0:8969:9881:d058  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)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 6  bytes 288 (288.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Note that there is now a new interface on your machine called “tun1” which is the encrypted tunnel.

6. Using the VPN tunnel

Still on Ubuntu2 ping the destination address of the tunnel. (In this example the command is ping 10.4.0.1). You should see something similar to the following:

```
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=1.04 ms
64 bytes from 10.4.0.1: icmp_seq=2 ttl=64 time=2.37 ms
64 bytes from 10.4.0.1: icmp_seq=3 ttl=64 time=1.49 ms
64 bytes from 10.4.0.1: icmp_seq=4 ttl=64 time=1.74 ms
64 bytes from 10.4.0.1: icmp_seq=5 ttl=64 time=1.48 ms
64 bytes from 10.4.0.1: icmp_seq=6 ttl=64 time=2.18 ms
64 bytes from 10.4.0.1: icmp_seq=7 ttl=64 time=0.868 ms
^C
--- 10.4.0.1 ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6009ms
```

Wireshark

Still on Ubuntu2 start Wireshark and capture traffic from tun1. As before ping the destination address of the tunnel.

Is the traffic encrypted? Why?

Now use Wireshark to capture traffic from the ethernet interface on Ubuntu2. As before ping the destination address of the tunnel.

Is the traffic encrypted when monitoring this interface? How do you explain what you see?

Telnet from Ubuntu2 to Ubuntu1

Again remaining on Ubuntu2, telnet into the destination address of tun1. You should see something similar to this:

```
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: user
Password:

Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.8.0-50-generic x86_64)
```

Do an ifconfig to ensure you have actually logged into the remote machine. You should see something similar to the following:

```
nsr@nsr-VirtualBox:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.64.19  netmask 255.255.255.0  broadcast
192.168.64.255
        inet6 fe80::39db:1cd5:a303:827f  prefixlen 64  scopeid
0x20<link>
        ether 08:00:27:0f:b2:41  txqueuelen 1000  (Ethernet)
        RX packets 1619  bytes 311200 (311.2 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 358  bytes 51696 (51.6 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 468  bytes 53063 (53.0 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 468  bytes 53063 (53.0 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::f55b:fd4b:cc07:b632  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)
        RX packets 141  bytes 8647 (8.6 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 116  bytes 10136 (10.1 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Once again monitor the tun1 interface using Wireshark while you do some simple commands (try ifconfig again or ls). Then monitor the ethernet interface. Take screen grabs for your report.

On which interface is traffic encrypted and which interface not encrypted? Why?

7. Assessment of this lab

Assessment of this lab is from a short report explaining what you did, what you observed and explanation as to what you saw. Grade is either pass / not pass. If the report is not passed you will be asked to resubmit an improved version of it.

Title

This is to be “NSR/AS Lab 3 - VPNs” followed by the student’s name and student id.

Abstract

No more than 200 words summarising the report

Introduction to VPNs

This is to be no more than 2 A4 pages (including diagrams). (NOTE: Two pages is a maximum, not a recommendation.) It is to discuss:

- VPN characteristics and capabilities
- How VPNs can be used to implement organisational security policy

OpenVPN behaviour

This section is to be addressed using the data collected above. It is to be no more than 5 pages.

- Discuss the VPN tunnels you set up in this lab.
- Answer the questions asked in Section 6.
- Answer the question “What is a VPN tunnel?” using the tunnel in this lab as an example.

Conclusion

No more than half a page summarising the main points of the report.

References

TNE30009/TNE80009

Laboratory Session 3

All sources are to be properly referenced. Use IEEE referencing. DO NOT JUST PROVIDE A LIST OF WEBPAGES.

Diagrams

All diagrams are to be numbered and captioned. If they are not the student's original work, they are to be referenced.

You may use any standard formatting you like, although IEEE is preferred.