

TNE20003 – Internet and Cybersecurity for Engineering Applications

Portfolio Task – Lab 9 Pass Task

Aims:

- To observe and investigate tunnelling in a network.

Preparation:

- View ["Introduction to Cybersecurity" & "Cybersecurity"](#)

Due Date:

- All tasks in this lab are to be completed and demonstrated to your Lab instructor preferably during or at the end of the current lab, but if you do not complete the tasks you may demonstrate it at the beginning of your next lab class.

Task 1

Get an understanding of the lab.

The purpose of this lab is to investigate tunnelling, in particular to consider how this and similar software can be used to subvert a security program. We will be examining tunnelling of telnet but the general principles can be applied to other protocols.

Telnet is used because it is a simple protocol with messages transmitted without encryption ("in clear text"). Of course, this also makes it a very insecure protocol and you should never use telnet when there are alternatives. Do not under any circumstances interpret the use of telnet in this lab as a recommendation for it to be used anywhere else.

You will use the Wireshark packet sniffing system and open source http tunnel software. You will use Wireshark to examine the packets.

This work is to be carried out in the VMWARE environment. We will be running two Linux Ubuntu virtual machines.

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

This lab uses the GNU http tunnelling software. It provides two tunnelling programs: **hts** and **htc**. **hts** is installed on the server system to accept tunnels from a specified port number and transfer them to another port number. **htc** is used to communicate to **hts** to set up a tunnel.

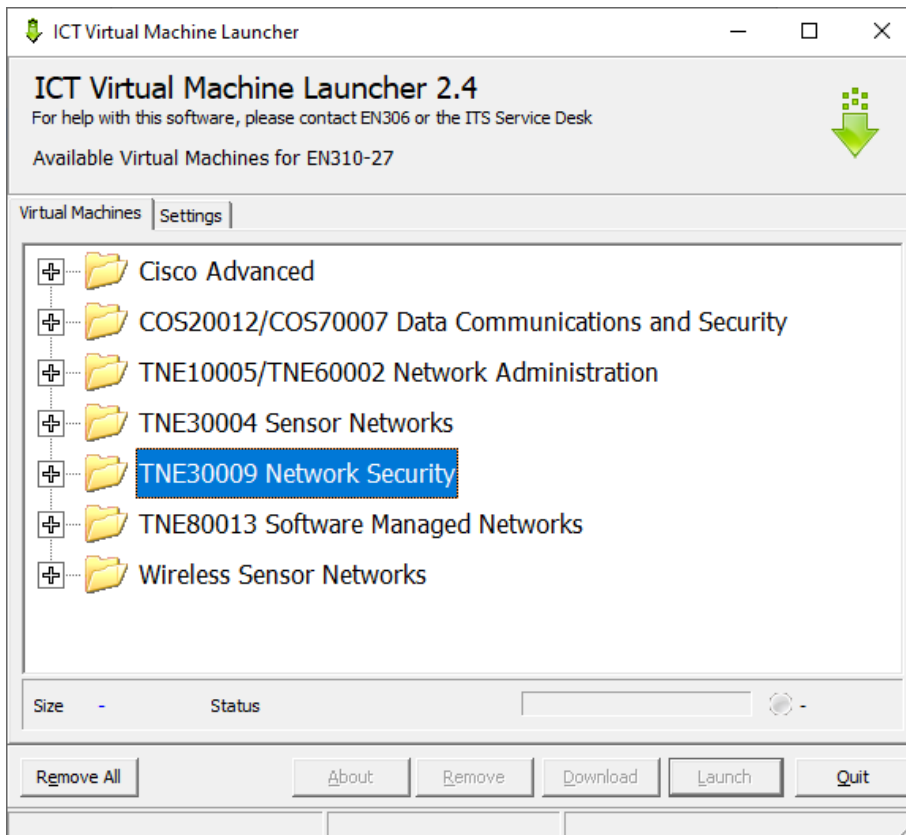
Assessment of this lab is through you demonstrating the working tunnel and wireshark output to the lab demonstrator and answering some questions (listed at the end of the lab) that the demonstrator will ask you.

Task 2

This lab uses Linux Virtual Machines (VMs) running in via VMWARE. If you are unfamiliar with Linux please spend some time before the lab learning some of its basic commands.

The VMs are both available via the start menu. To get them started do the following:

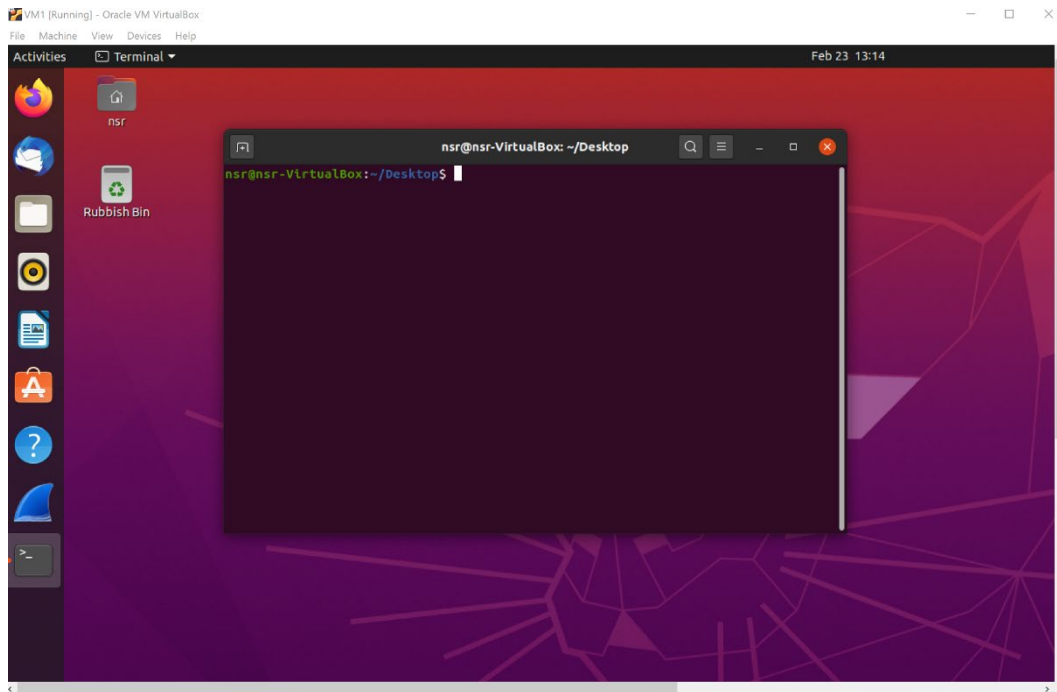
1. Go to Start menu > Virtual Machine Launcher. You will be presented with the following window:



Choose the TNE30009 Network Security tab. You will be presented with three VMs. Download Ubuntu 1 and Ubuntu 2. This may take a few minutes.

Launch the VMs. You may be asked to update the VMWARE software or the Ubuntu version. Ignore these.

2. You should now have two Ubuntu VMs running.



3. You will need to work within the terminal. Right click your mouse and choose “**Open in Terminal**”. You can determine your IP address with the command **ifconfig**. The interface you will be monitoring is **ens33**. It should have an IP address assigned from 192.168.0.0/16 with both hosts on the same /24 subnet.
4. Once both VMs are running determine their IP addresses and confirm they are on the same subnet. Use

Ifconfig

5. You should now be able to ping between the two VMs. Make sure you can.

ping <ip address>

Task 3

To set up the tunnel, choose one machine to be the tunnel server and the other to be the tunnel client. You will telnet via the tunnel from the tunnel client to the tunnel server. You are to determine the addresses and port numbers to use in the command below.

On the server machine use the following command to accept tunnelled messages from port 80. Messages are to be forwarded to the telnet server listening on port 23.

**sudo hts -F <IP address to forward to>:<forwarding port number>
<receiving port number>**

On the client machine use the following command to initiate a tunnel to the server machine over port 80.

Use 2323 as the receiving port number.

```
sudo htc -F <receiving port number> <IP address to forward to>:<forwarding port number>
```

Task 4

To monitor the traffic start Wireshark. Monitor the ethernet port which will usually be **ens3**.

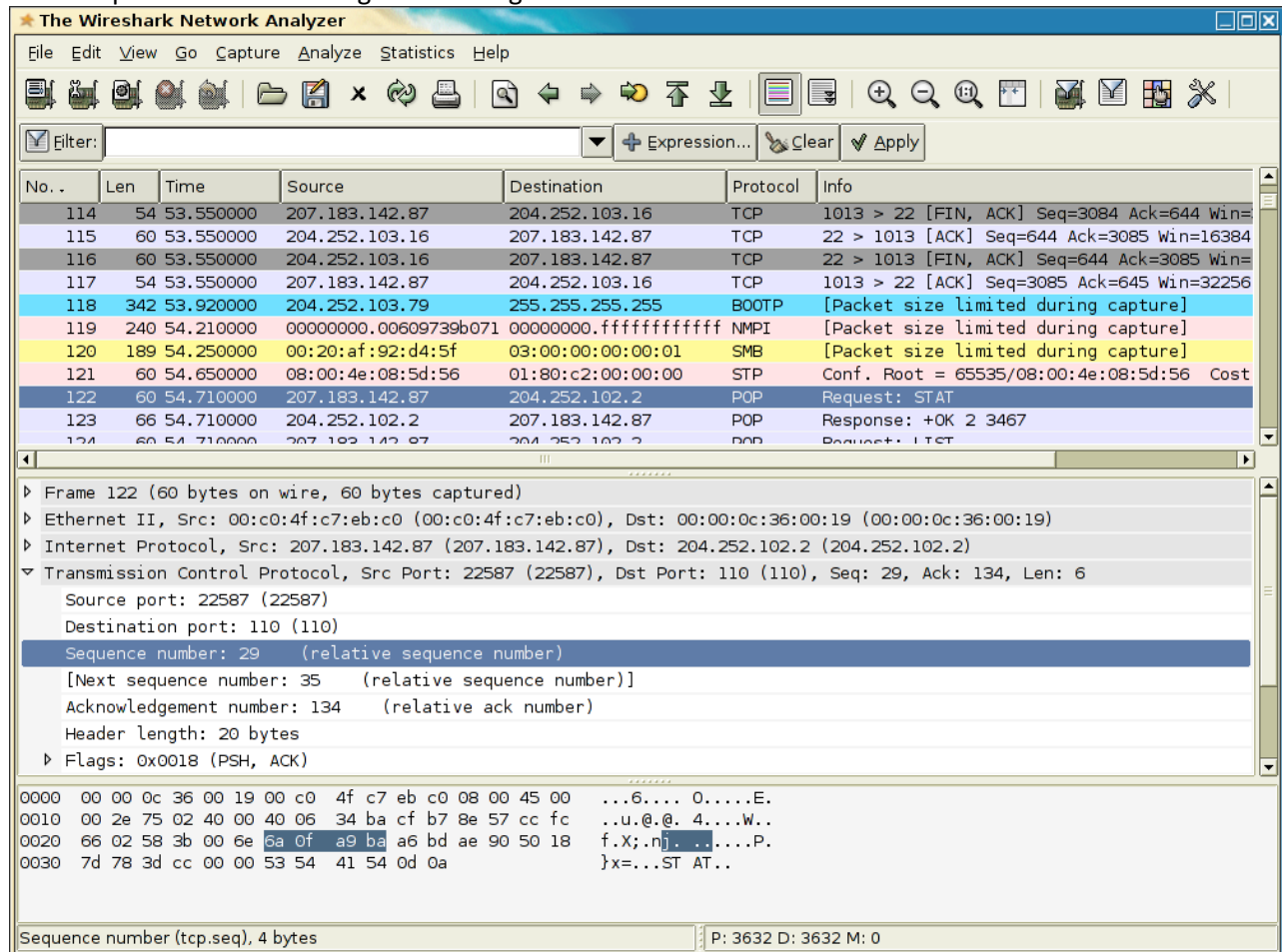
Initiate a telnet session via the tunnel from the client. The format of the telnet command is:

```
telnet <IP address of tunnel client> <receiving port number>
```

You should receive a telnet login request. Login with user **nsr** and password **user**.

Using Wireshark you should be able to see the transmitted traffic. Note how the telnet traffic is tunnelled.

Your output should something like the diagram below:



The screenshot shows the Wireshark Network Analyzer interface. The packet list pane displays a list of captured packets. The packet details pane shows the structure of a BOOTP packet, including source and destination ports, sequence number, and flags. The packet bytes pane shows the raw data of the packet.

No.	Len	Time	Source	Destination	Protocol	Info
114	54	53.550000	207.183.142.87	204.252.103.16	TCP	1013 > 22 [FIN, ACK] Seq=3084 Ack=644 Win=
115	60	53.550000	204.252.103.16	207.183.142.87	TCP	22 > 1013 [ACK] Seq=644 Ack=3085 Win=16384
116	60	53.550000	204.252.103.16	207.183.142.87	TCP	22 > 1013 [FIN, ACK] Seq=644 Ack=3085 Win=
117	54	53.550000	207.183.142.87	204.252.103.16	TCP	1013 > 22 [ACK] Seq=3085 Ack=645 Win=32256
118	342	53.920000	204.252.103.79	255.255.255.255	BOOTP	[Packet size limited during capture]
119	240	54.210000	00000000.00609739b071	00000000.ffffffffffff	NMPI	[Packet size limited during capture]
120	189	54.250000	00:20:af:92:d4:5f	03:00:00:00:00:01	SMB	[Packet size limited during capture]
121	60	54.650000	08:00:4e:08:5d:56	01:80:c2:00:00:00	STP	Conf. Root = 65535/08:00:4e:08:5d:56 Cost
122	60	54.710000	207.183.142.87	204.252.102.2	POP	Request: STAT
123	66	54.710000	204.252.102.2	207.183.142.87	POP	Response: +OK 2 3467
124	60	54.710000	207.183.142.87	204.252.102.2	POP	Request: LIST

Frame 122 (60 bytes on wire, 60 bytes captured)

Ethernet II, Src: 00:c0:4f:c7:eb:c0 (00:c0:4f:c7:eb:c0), Dst: 00:00:0c:36:00:19 (00:00:0c:36:00:19)

Internet Protocol, Src: 207.183.142.87 (207.183.142.87), Dst: 204.252.102.2 (204.252.102.2)

Transmission Control Protocol, Src Port: 22587 (22587), Dst Port: 110 (110), Seq: 29, Ack: 134, Len: 6

Source port: 22587 (22587)

Destination port: 110 (110)

Sequence number: 29 (relative sequence number)

[Next sequence number: 35 (relative sequence number)]

Acknowledgement number: 134 (relative ack number)

Header length: 20 bytes

Flags: 0x0018 (PSH, ACK)

0000 00 00 0c 36 00 19 00 c0 4f c7 eb c0 08 00 45 00 ...6.... 0....E.

0010 00 2e 75 02 40 00 40 06 34 ba cf b7 8e 57 cc fc ...u.@.@. 4....W..

0020 66 02 58 3b 00 6e 6a 0f a9 ba a6 bd ae 90 50 18 f.X;.nj.P.

0030 7d 78 3d cc 00 00 53 54 41 54 0d 0a }x=...ST AT..

Sequence number (tcp.seq), 4 bytes P: 3632 D: 3632 M: 0

Task 5

Assessment of this lab

This lab will be assessed with a discussion between you and the lab supervisor. You will need to demonstrate that you have successfully completed the lab. He or she will also ask you the following:

1. What command is needed on the tunnel server to accept tunnelled messages on port 80 and send them to port 23 on the local host? Include the IP addresses that you used. `sudo hts -F 192.168.157.129:23 80`
2. What command is needed on the tunnel client to tunnel messages to and from port 2323 to and from port 80? `sudo htc -F 2323 192.168.157.128:80`
3. What did Wireshark interpret the tunnelled traffic as? TCP and Telnet protocol, but since we using port 80 so HTTP could be seen
4. How did the tunnelling software attempt to hide the traffic? It encapsulates data (could be malicious) with another protocol like HTTP in this case
5. How might tunnelling of this kind be used to subvert security policy?

Tunneling could help to transport traffic under the security firewalls, since it could encapsulates data with another protocol and pass it through, for instance malicious code could be wrapped in HTTP header.