

Network Security – CSCI_6541_80

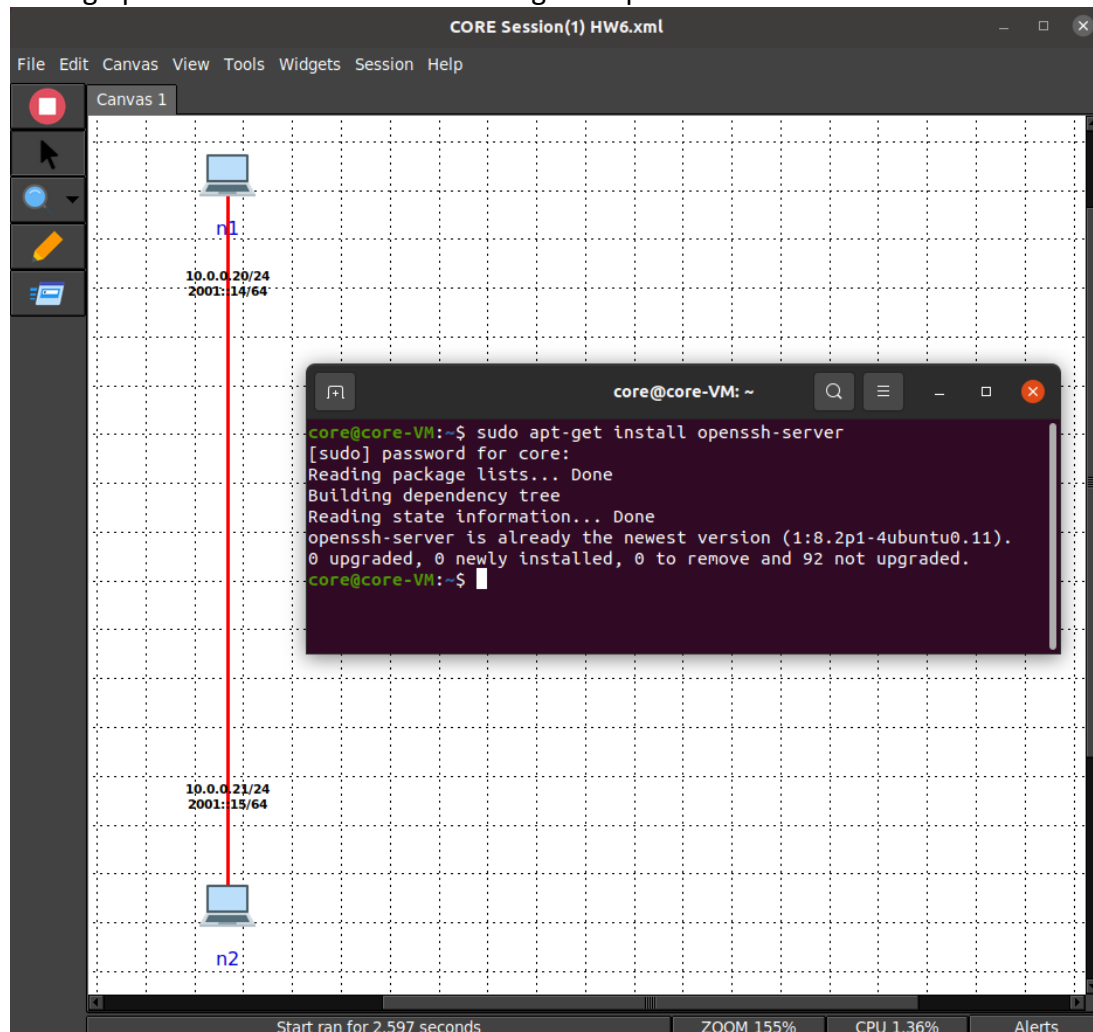
Namana Y Tarikere – G21372717

Homework Assignment – 6

SSH (10pts)

- 1) Use the CORE scenario below. Make sure OpenSSH Server is installed. Just in case it is not, run the following on Ubuntu: `sudo apt-get install openssh-server`.

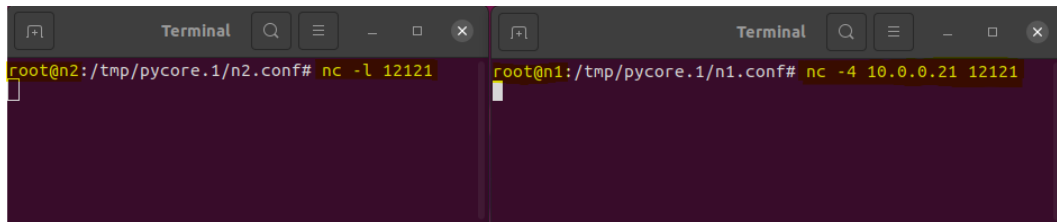
Setting up the core scenario and ensuring the OpenSSH server is installed as shown:



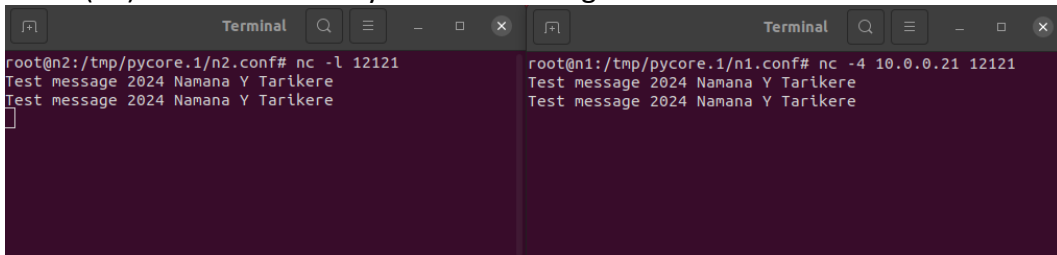
- 2) **(2pts)** Create a simple chat client using netcat:
 - a. nc manual is here: <https://www.commandlinux.com/man-page/man1/nc.1.html>
 - b. Run server on n2: `nc -l 12121`
 - c. Run client on n1: `nc -4 10.0.0.21 12121`
 - d. You can type "Test message 2024 {your name}" at the prompt of both ends and see message received on the other end.

e. Show a screenshot of your commands.

Running the server on n2 terminal and the client on the n1 terminal using the nc commands mentioned above:

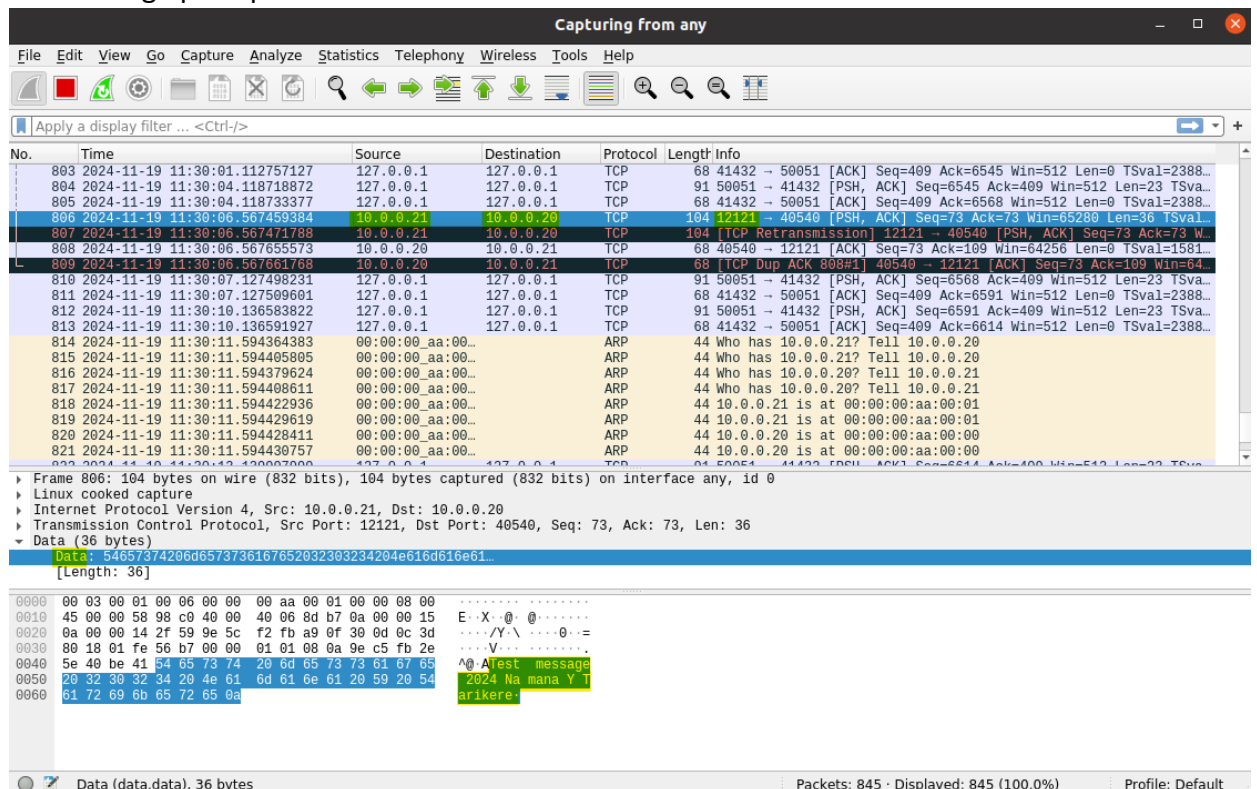


Typing “Test message 2024 Namana Y Tarikere” as prompt on both server (n2) and client (n1) terminals to verify that the message was received on the other end:

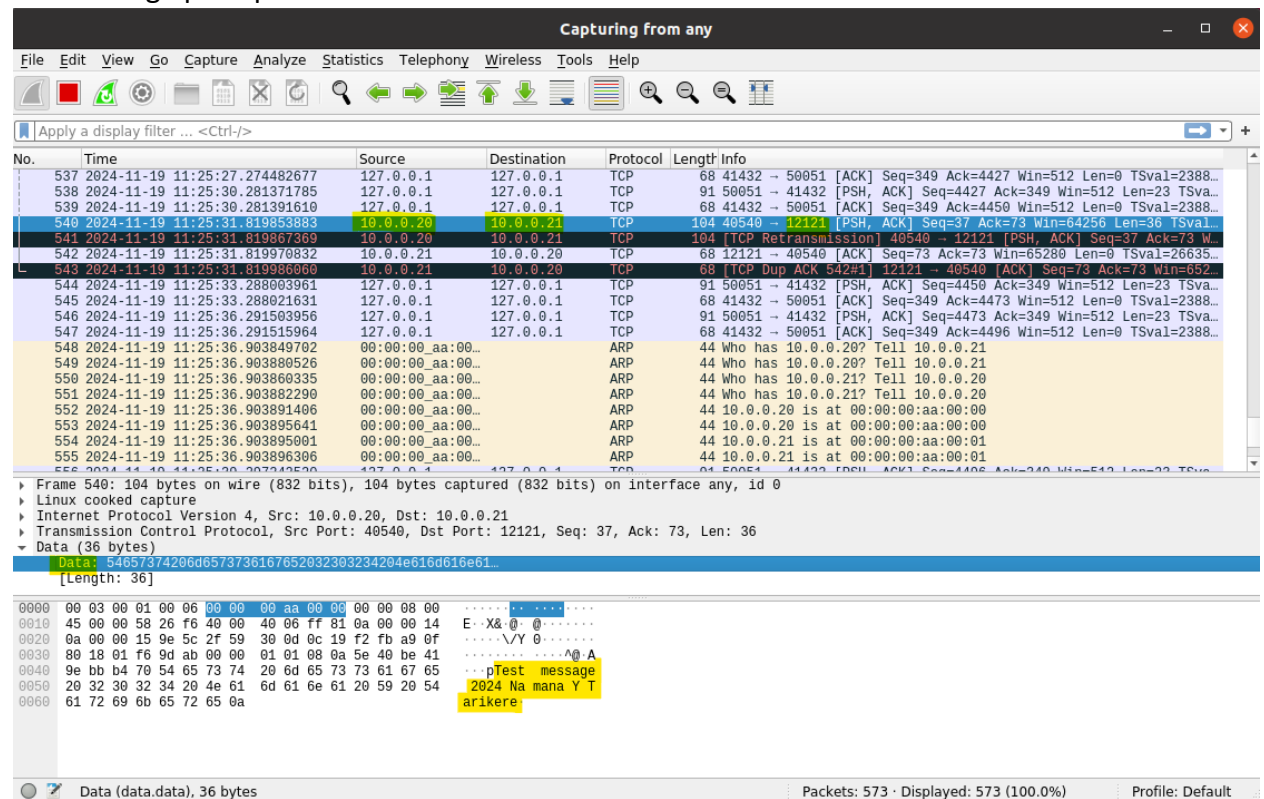


- 3) (1pts) Observe chat traffic being sent from n1 to n2 and vice versa by capturing traffic on the link between them. Use Wireshark to show the “Test message 2024 {your name}” captured.

Capturing chat traffic on Wireshark from n2 (10.0.0.21) to n1 (10.0.0.20) where packet has the message prompt in the data field as shown below:

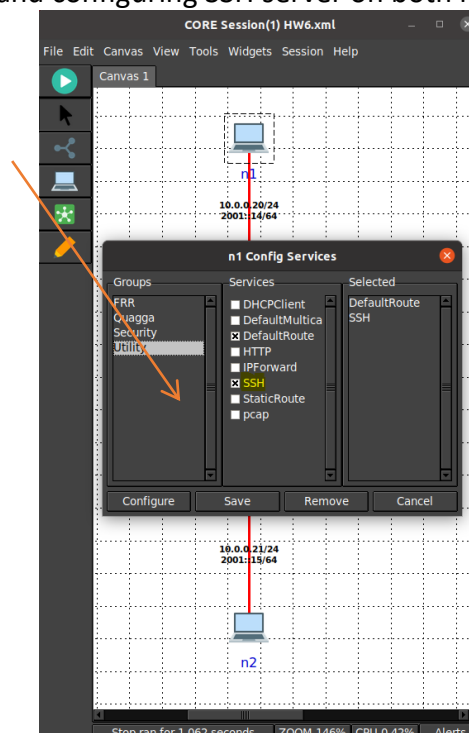


Capturing chat traffic on Wireshark from n1 (10.0.0.20) to n2 (10.0.0.21) where packet has the message prompt in the data field as shown below:



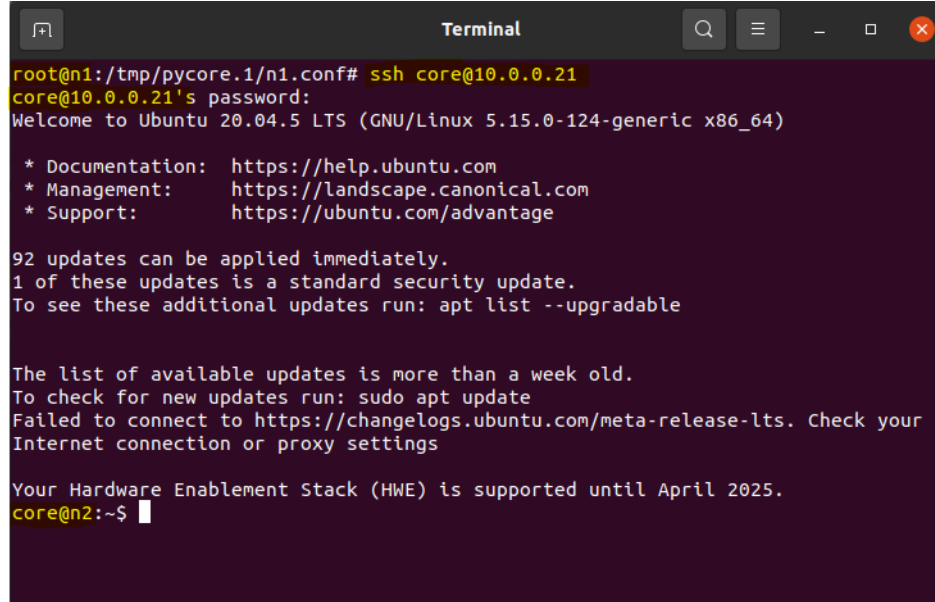
- 4) Stop the scenario and configure an SSH server on both n1 and n2. You can do that by right clicking on each, selecting "services", and enabling SSH utility as shown below.

Stopping the scenario and configuring SSH server on both n1 and n2 as shown below:



- 5) Re-run the scenario and make sure that n1 can SSH to n2 and vice versa. You will login with the same username and password you use for your Ubuntu host (username “core” password “core” if you use the VM I provided). Terminate the SSH session.

SSH connection from n1 to n2 using the command `ssh core@10.0.0.21` as shown:



```
Terminal
root@n1:/tmp/pycore.1/n1.conf# ssh core@10.0.0.21
core@10.0.0.21's password:
Welcome to Ubuntu 20.04.5 LTS (GNU/Linux 5.15.0-124-generic x86_64)

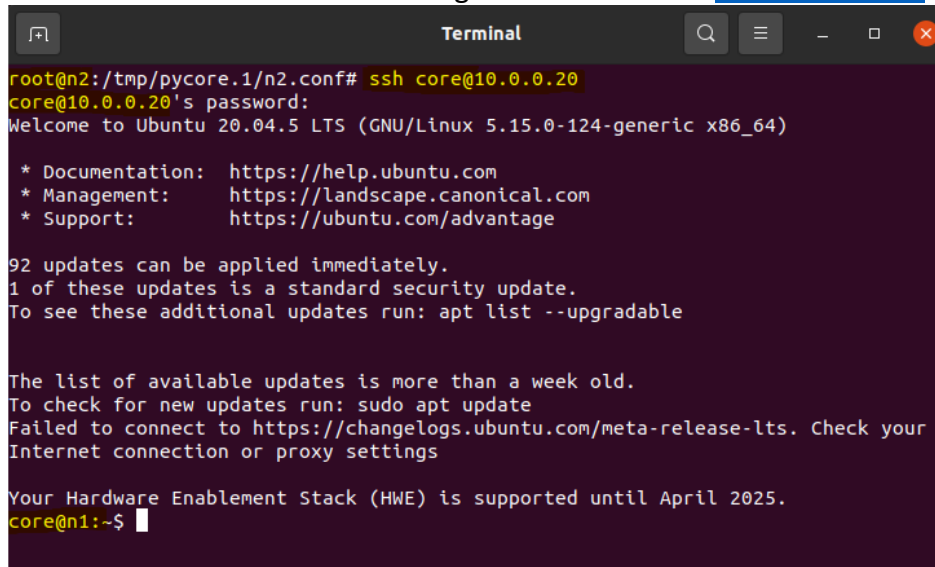
 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

92 updates can be applied immediately.
1 of these updates is a standard security update.
To see these additional updates run: apt list --upgradable

The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your
Internet connection or proxy settings

Your Hardware Enablement Stack (HWE) is supported until April 2025.
core@n2:~$
```

SSH connection from n2 to n1 using the command `ssh core@10.0.0.20` as shown:



```
Terminal
root@n2:/tmp/pycore.1/n2.conf# ssh core@10.0.0.20
core@10.0.0.20's password:
Welcome to Ubuntu 20.04.5 LTS (GNU/Linux 5.15.0-124-generic x86_64)

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

92 updates can be applied immediately.
1 of these updates is a standard security update.
To see these additional updates run: apt list --upgradable

The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your
Internet connection or proxy settings

Your Hardware Enablement Stack (HWE) is supported until April 2025.
core@n1:~$
```

- 6) (3pts) Now, establish the SSH session from n1 to n2 so you can tunnel the chat traffic inside of SSH (nc chat client and server should not be run from a terminal where you have an active SSH session). Show your exact steps and commands you used on all nodes, with screenshots.
- Read up on port forwarding options in SSH: <https://www.commandlinux.com/man-page/man1/ssh.1.html>
 - You will need two separate window terminals on n1, one to SSH to n2 and one to run nc. nc should not be run within an SSH session.

SSH connection from n1 (10.0.0.20) to n2 (10.0.0.21) to tunnel chat traffic using the core credentials and localhost (127.0.0.1) using the command as shown below:
Command -> `ssh -L 12121:127.0.0.1:12121 core@10.0.0.21`

```
Terminal
root@n1:/tmp/pycore.1/n1.conf# ssh -L 12121:127.0.0.1:12121 core@10.0.0.21
core@10.0.0.21's password:
Welcome to Ubuntu 20.04.5 LTS (GNU/Linux 5.15.0-124-generic x86_64)

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

92 updates can be applied immediately.
1 of these updates is a standard security update.
To see these additional updates run: apt list --upgradable

The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your
Internet connection or proxy settings

Your Hardware Enablement Stack (HWE) is supported until April 2025.
core@n2:~$
```

Next, run NC server on n2 using the command `nc -l 12121` as shown below:

```
Terminal
root@n2:/tmp/pycore.1/n2.conf# nc -l 12121
Test message 2024 Namana
Test message 2024 Namana Y Tarikere
```

Next, run NC client on n1 using the command `nc -4 127.0.0.1 12121` as shown:

```
Terminal
root@n1:/tmp/pycore.1/n1.conf# nc -4 127.0.0.1 12121
Test message 2024 Namana
Test message 2024 Namana Y Tarikere
```

- 7) (1pts) Show chat messages sent from client to server and show a Wireshark capture with the traffic between them (now tunnelled).

The exchange of chat messages traffic between client (n1) and server (n2) after SSH tunnelling is shown below:

```
Terminal Terminal
root@n2:/tmp/pycore.1/n2.conf# nc -l 12121 root@n1:/tmp/pycore.1/n1.conf# nc -4 127.0.0.1 12121
Test message 2024 Namana Test message 2024 Namana
Test message 2024 Namana Y Tarikere Test message 2024 Namana Y Tarikere
Test message 2024 Namana
Test message 2024 Namana Y Tarikere
```

Wireshark screenshot where n2 (Server) is sending server encrypted data packets to n1(Client) after SSH tunnelling is shown below:

Wireshark interface showing a packet capture. The packet list displays a packet from 10.0.0.21 to 10.0.0.20, identified as "128 Server: Encrypted packet (len=60)". The packet details show "SSH Version 2" with encryption parameters: "encryption:chacha20-poly1305@openssh.com mac:<implicit> compression:none". The packet bytes show a hex dump of the encrypted data.

Encrypted Packet (ssh.encrypted_packet), 40 bytes

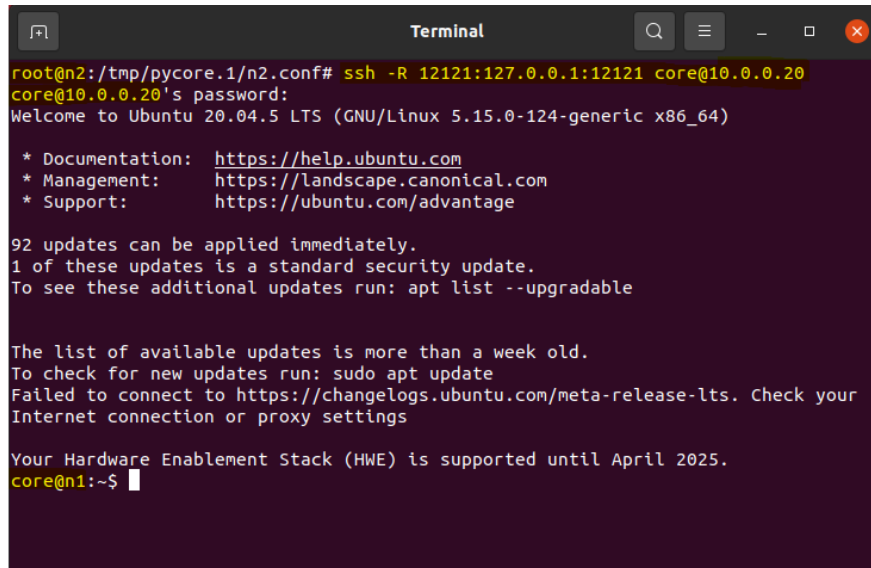
Wireshark screenshot where n1 (Client) is sending client encrypted data packets to n2(Server) after SSH tunnelling is shown below:

Wireshark interface showing a packet capture. The packet list displays a packet from 10.0.0.20 to 10.0.0.21, identified as "144 Client: Encrypted packet (len=76)". The packet details show "SSH Version 2" with encryption parameters: "encryption:chacha20-poly1305@openssh.com mac:<implicit> compression:none". The packet bytes show a hex dump of the encrypted data.

Encrypted Packet (ssh.encrypted_packet), 56 bytes

- 8) **(3pts)** Repeat 6 and 7 but this time establish the SSH session in step 6 from n2 to n1. Structure the SSH command so the nc connection from n1 to n2 is still tunnelled through the SSH session.

SSH connection from n2 (10.0.0.21) to n1 (10.0.0.20) to tunnel chat traffic using the command `ssh -R 12121:127.0.0.1:12121 core@10.0.0.20` to include port forwarding as shown below:



```
root@n2:/tmp/pycore.1/n2.conf# ssh -R 12121:127.0.0.1:12121 core@10.0.0.20
core@10.0.0.20's password:
Welcome to Ubuntu 20.04.5 LTS (GNU/Linux 5.15.0-124-generic x86_64)

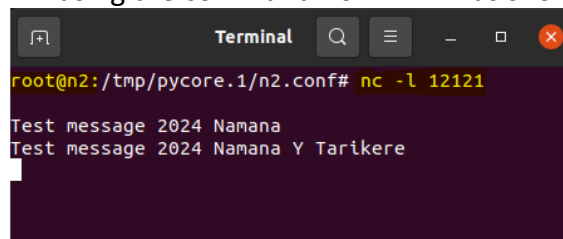
 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

92 updates can be applied immediately.
1 of these updates is a standard security update.
To see these additional updates run: apt list --upgradable

The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your
Internet connection or proxy settings

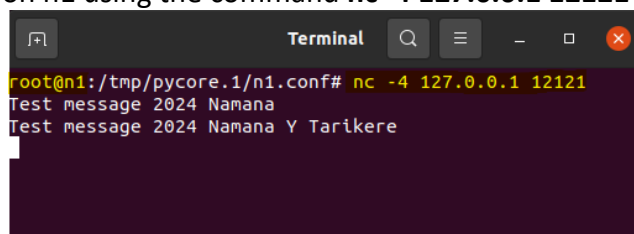
Your Hardware Enablement Stack (HWE) is supported until April 2025.
core@n1:~$
```

Next, run NC server on n2 using the command `nc -l 12121` as shown below:



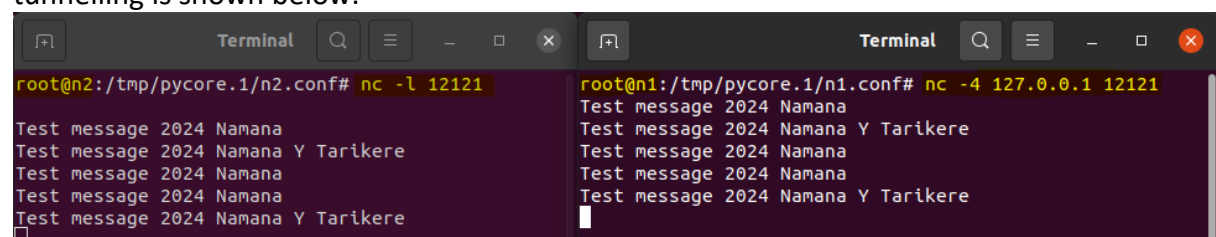
```
root@n2:/tmp/pycore.1/n2.conf# nc -l 12121
Test message 2024 Namana
Test message 2024 Namana Y Tarikere
```

Next, run NC client on n1 using the command `nc -4 127.0.0.1 12121` as shown:



```
root@n1:/tmp/pycore.1/n1.conf# nc -4 127.0.0.1 12121
Test message 2024 Namana
Test message 2024 Namana Y Tarikere
```

The exchange of chat messages traffic between client (n1) and server (n2) after SSH tunnelling is shown below:



```
root@n2:/tmp/pycore.1/n2.conf# nc -l 12121
Test message 2024 Namana
Test message 2024 Namana Y Tarikere
Test message 2024 Namana
Test message 2024 Namana
Test message 2024 Namana Y Tarikere

root@n1:/tmp/pycore.1/n1.conf# nc -4 127.0.0.1 12121
Test message 2024 Namana
Test message 2024 Namana Y Tarikere
Test message 2024 Namana
Test message 2024 Namana
Test message 2024 Namana Y Tarikere
```

Wireshark screenshot where n2 (Server) is sending encrypted data packets to n1(Client) after SSH tunnelling is shown below:

The screenshot shows a Wireshark capture of an SSH tunnel. The packet list at the top shows a packet from 10.0.0.21 to 10.0.0.20, identified as 'Client: Encrypted packet (len=68)'. The packet details pane shows the following information:

- Frame 9866: 136 bytes on wire (1088 bits), 136 bytes captured (1088 bits) on interface any, id 0
- Linux cooked capture
- Internet Protocol Version 4, Src: 10.0.0.21, Dst: 10.0.0.20
- Transmission Control Protocol, Src Port: 45990, Dst Port: 22, Seq: 2319, Ack: 3703, Len: 68
- SSH Protocol
 - SSH Version 2 (encryption:chacha20-poly1305@openssh.com mac:<implicit> compression:none)
 - Packet Length (encrypted): bfcf7fa6
 - Encrypted Packet: 3f203668db6e362b4f85094e9352365e00a441f822ce77e2...
 - MAC: 6e02bb9e3fa4e92172864fde77ffec6d
 - [Direction: client-to-server]

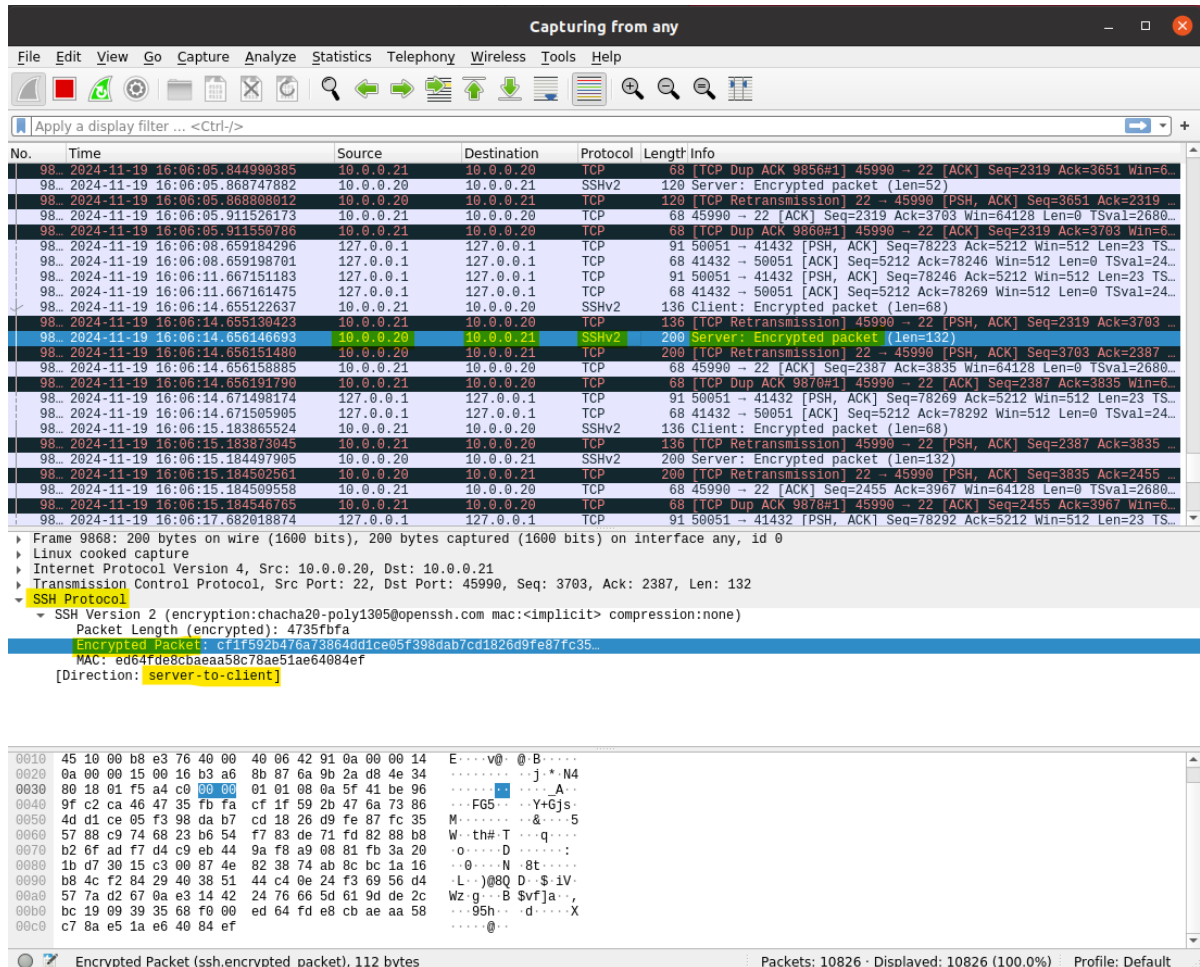
The packet bytes pane shows the raw data in hexadecimal and ASCII. The ASCII column shows the following characters:

```

E...S...@...a...
...M...j...
...u...F...
...A-B...? 6h.n6+
...N-R6A...A...w...
...B...jXv...y...
SA...U...n...?...!
r-O-w-m
  
```

Here, Wireshark is capturing the **Client Encrypted packets** with the direction as client to server even though the message is **transferred from Server (n2) to Client (n1) because of port forwarding** in the ssh tunnel connection (-R mentioned in the command for ssh connection from n2 to n1).

Wireshark screenshot where n1 (Client) is sending encrypted data packets to n2(Server) after SSH tunnelling is shown below:



Here, Wireshark is capturing the **Server Encrypted packets** with the direction as server to client even though the message is **transferred from Client (n1) to Server (n2) because of port forwarding** in the ssh tunnel connection (-R mentioned in the command for ssh connection from n2 to n1).