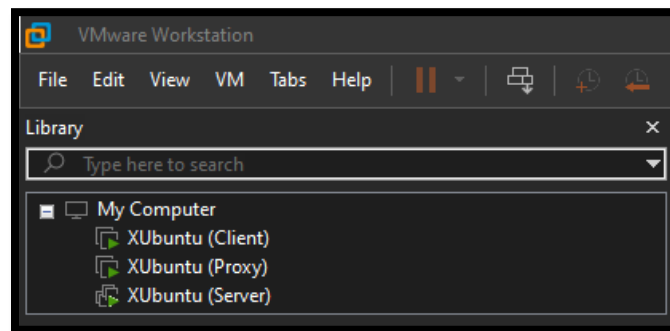


Network Address Translation

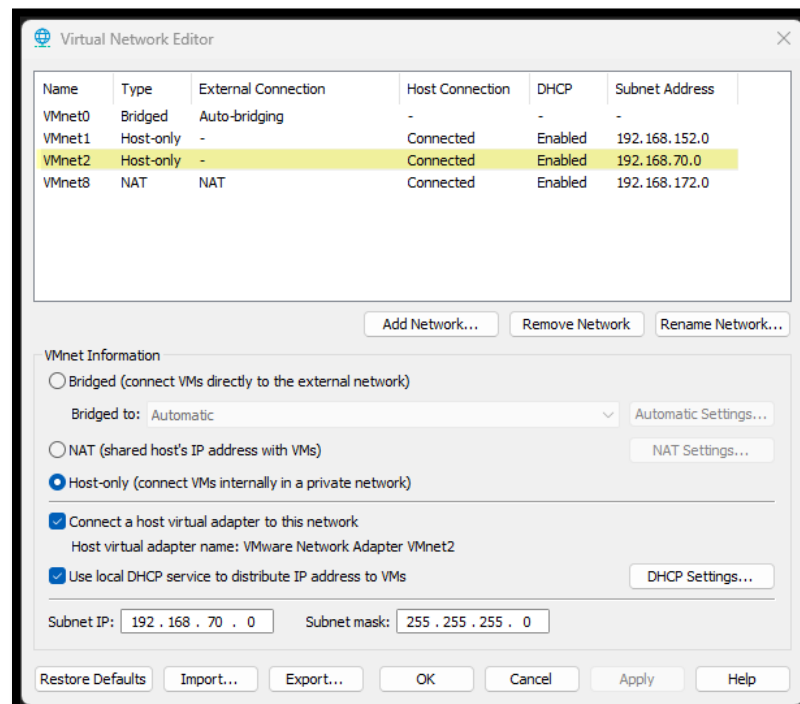
1 CONFIGURE ENVIRONMENT

To create and test the router functionality for submission 2, I have set up a virtual environment using VMware Workstation. This environment includes three virtual machines (VMs) networked together to simulate a real-world network environment. This VM configuration includes:

- Client VM
- Server VM
- Router/Proxy VM



I have created a host-only network by using the 'Virtual Network Editor' tool within VMware. While this restricts external connections and creates a closed network environment, this setup will allow me to control all communication and data exchanged within the environment.



2 TCP SOCKET COMMUNICATION

Using the python library `scapy`, I have created a proxy to monitor and intercept data transmitted via a TCP socket within the closed network. After initiating the script, the proxy passively listens for incoming TCP connections. Whenever a packet is received, it verifies the presence of IP and TCP layers and then extracts relevant data such as source and destination IP addresses and port numbers. The proxy then inspects the packet's payload to ensure it contains a length greater than zero. If this condition is met, the proxy extracts and displays the payload length, the payload contents in hexadecimal format, and the contents in human-readable format. This script is designed to accept incoming connections from both the client and server within the closed network.

```
root@ubuntu:/home/se/Desktop# python3 proxy.py
[*] Packet received from Client
  src: 192.168.70.130:20
  dst: 192.168.70.129:8080
  len: 34
  payload (hex): 48656c6c6f2066726f6d20636c69656e74
  payload (decoded): Hello from client

[*] Packet received from Server
  src: 192.168.70.129:8080
  dst: 192.168.70.130:20
  len: 12
  payload (hex): 000000000000
  payload (decoded):

[*] Packet received from Server
  src: 192.168.70.129:20
  dst: 192.168.70.130:9876
  len: 34
  payload (hex): 48656c6c6f2066726f6d20736572766572
  payload (decoded): Hello from server

[*] Packet received from Client
  src: 192.168.70.130:9876
  dst: 192.168.70.129:20
  len: 12
  payload (hex): 000000000000
  payload (decoded):
```

All relevant scripts are included in the submission folder, including `client.py`, `proxy.py`, and `server.py`. The server and client scripts are identical, with their only distinction being their destination addresses.

While the IP addresses are currently hard coded for the purpose of this submission, they will be dynamically read and identified as either internal (client) or external (server) for submission 2. As all IP addresses in this closed network adhere to the 192.X.X.X pattern, any conversions will be purely used for demonstrating the functionality of the proxy in a simulated environment. As such, submission 2 will extend this proxy to manipulate the source IP address as necessary, maintain a state table to track these modifications, and then forward the data to its target destination.