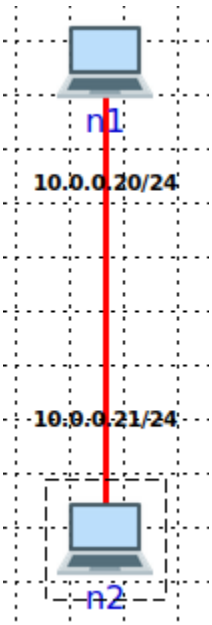
SSH (10pts)

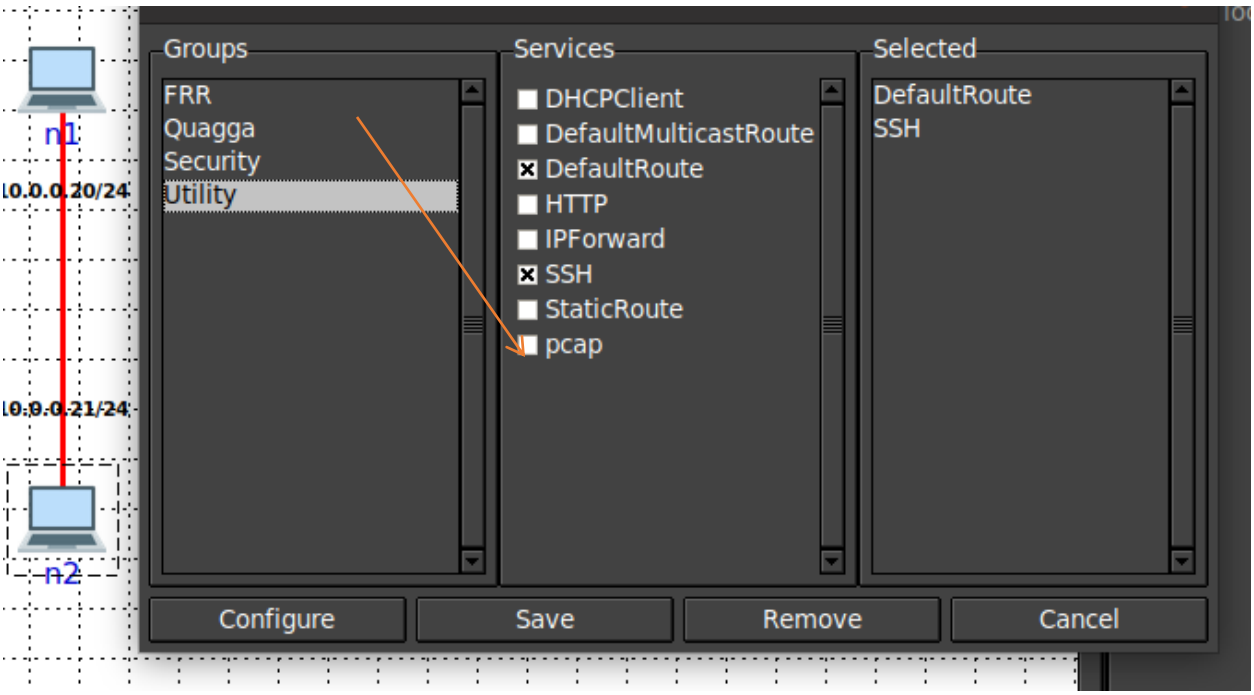
1) Use the CORE scenario below. Make sure OpenSSH Server is installed. Just in case it is not, run the following on Ubuntu: sudo apt-get install openssh-server.



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.

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.

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.



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.
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.
    a. Read up on port forwarding options in SSH: https://www.commandlinux.com/man-page/man1/ssh.1.html
    b. 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.
7) (1pts) Show chat messages sent from client to server and show a Wireshark capture with the traffic between them (now tunneled).
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 tunneled through the SSH session.