CS 261 Mini-Lab: Port Forwarding

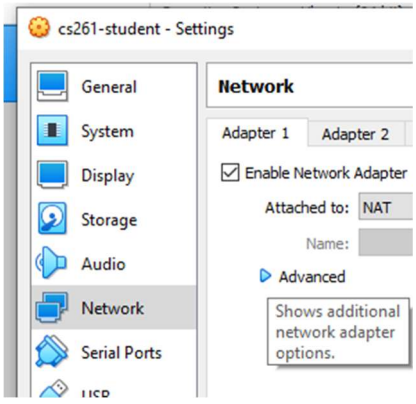
NOTE: Labs are designed to provide an opportunity to ask for help!

Setup

- Ensure that you're using the class VM in VirtualBox 6+ (or equivalent configuration). See the CS 261 Moodle for more details on the class configuration.
- You should not need to update your OS for this lab, but it's good practice to start each session with:
    1. sudo apt-get update
    2. sudo apt-get upgrade
    3. sudo reboot

Instructions

1. In your VM, create the directory **/srv/node/portForwarding**
2. Copy the helloWorld.js file that accompanied this document into the directory you just created.
    - If you downloaded it using the browser in the VM, just cp it.
    - If you downloaded it using a Windows browser, drag-and-drop it into the SVN or use the shared-folder feature.
3. In Terminal, go to the directory you created (if you're not already there).
4. Run **node helloWorld.js**
    - Note that npm init or install is not needed – there are no packages required for this lab.
5. Within the VM, open a browser at http://localhost:1337, and note that you get a "Hello World" response.
6. In the host OS (Windows), open a browser at http://localhost:1337 and note that it does not work – it eventually times out.
7. While the VM is still running, bring up the VirtualBox Manager screen.
8. Right click on the VM you're using, and choose **Settings**.
9. In the settings dialog, click on Network (on the left).
10. In the adapter shown, click on the arrow next to Advanced.



11. Click on the Port Forwarding button you can now see.
12. Click on the + button at the upper-right of the new dialog.

13. Click on the name of the rule (currently Rule 1), and replace the text with **CS261 Mini Lab - Port Forwarding**.
14. Click on the Host Port, and enter **1337**.
15. Click on the Guest Port, and enter **1337**.
16. Your setting should look like this:

    - Leave the other settings alone – we're using TCP (the underlying protocol for HTTP), and we want to leave the IP fields blank.
17. Click OK to dismiss the Port Forwarding dialog.
18. Click OK again to dismiss the Settings dialog.
19. In your Windows browser, hit refresh (or re-enter the same URL), and **note that you now get a Hello World result!**
20. Repeat steps 8 – 12 to bring back the port forwarding button.
21. Modify the Host Port to **1338** (leave the Guest Port alone!)
22. Repeat steps 18 – 20 – **note that the Windows browser now fails again**.
23. Replace the URL in the Windows browser with http://localhost:1338, and **note that you now get a Hello World result again.**

Notes

- In short, VirtualBox binds the Host IP::Host Port on the host OS (Windows). Incoming traffic is rewritten to replace the IP and Port with the Guest IP and Port, and then sent to the adapter within the VM.
- **Guest Port** is the port that will be written into the packets that are sent within the guest OS. Since we bound 1337 in our Node.js, that was the correct choice.
- **Host Port** is an arbitrary choice we made. VirtualBox is binding that port for us, and forwarding that traffic on to the VM. It's generally less confusing to use the same port for Guest and Host, with the following caveats:
    - If you're already using that port for something else in the host OS (Windows), then you can choose an arbitrary port range.
    - If you had multiple VMs running at once, all running the same code, you would need to set up port forwarding for each VM on unique Host Ports, with the same Guest Ports on each.
- **Host IP** is the IP that VirtualBox will bind on the host OS. Leaving it blank is the same as entering 0.0.0.0. Binding a TCP port on 0.0.0.0 is a special case, binding the port on every available network adapter.
    - You could have entered 127.0.0.1, and the lab would have worked as well.
- **Guest IP** is the new IP destination in the packet inside the guest OS.