

Lab09/10- Network Address Translation

In this lab, we're going to create a network address translation engine. The key concept is that, via software, we will receive an IPv4 packet, manipulate it, then re-send it onto its destination.

Instructions

Perform the following tasks in an environment of your choosing. You are welcome to use your own computer, a network of computers is not necessary to complete this lab, however you're welcome to do so if you wish. The specific requirements of this lab are largely open, this allows you to choose an environment, programming language, and a packet manipulation scenario of your own.

1. In a language of your choosing, create a program that act as a network address translator.
 - a. You'll create an application that runs a listener or sorts. It'll listen for incoming network sockets
 - b. Your listener may bind to an entire network interface or a few ports, the option is yours (a true router would bind to an entire interface typically, but we can scope things down if you'd like, please grab a minimum of 80, 443, and 22 for the lab)
 - c. Once you've received the packet, manipulate the source IP address, keep track of the manipulation in a state table, then re-send it to its destination
 - d. You may choose to conduct the port manipulation symmetrically or statically
 - e. Feel free to use any language of your choice
 - f. Many freely-available development frameworks are available to assist with this task, feel free to use them
2. Create a short video demonstrating that you're successfully able to route a packet into your 'routing engine' and that it forwards the packet onward
 - a. Wireshark is a great tool to demonstrate your success
 - b. Camtasia offers a free demo. QuickTime will record on MacOS for free as well. The Windows App store has several options as well.

What to Submit

This lab has TWO separate submissions.

Submission01

Submission01 is to keep us on track. In submission01, the specific requirements are not specifically defined but should include the following accomplishments:

1. Show that you have created an environment to test/create your router. A few VMs networked together is fine. If you have other methods, that's fine too, just describe/show it.
2. Show that you can receive data over a TCP socket. You don't have to do anything with it, just prove that you're able to listen on a socket. Include the code for your current state.

Submission02

Submission02 will be the final wrap up of your router. It will have code, a video, and any additional documentation you wish to provide, showing that you've met the requirements.

At a minimum, provide the following documents to the D2L Dropbox:

1. Documentation on your environment and how it functions, including but not limited to operating system, binding type (only certain ports or entire interfaces), type of NAT, etc
2. A video showing the routing of a packet
3. Your code that you've written to complete the lab

Grading

Your grades will be based off of a rubric that includes the following items:

1. Proper documentation of your environment
2. A video showing the successful manipulation of the packet
3. Writing your own software to complete the task
4. Proper formatting, following good technology standards, and meeting submission guidelines