Final Project: Implementing a Simple Router

In the previous lab you implemented a simple firewall that allowed ARP and ICMP packets but blocked some TCP packets. For your final project, you will be expanding on this to implement routing between devices on different subnets and implementing firewalls for certain subnets. The idea is to simulate an actual production network. You will be using ideas from Lab 1 to help construct the mininet topology, and ideas from Lab 3 to implement the rules allowing for traffic to flow through your network. Please refer back to those Labs for guidance on how to complete this assignment.

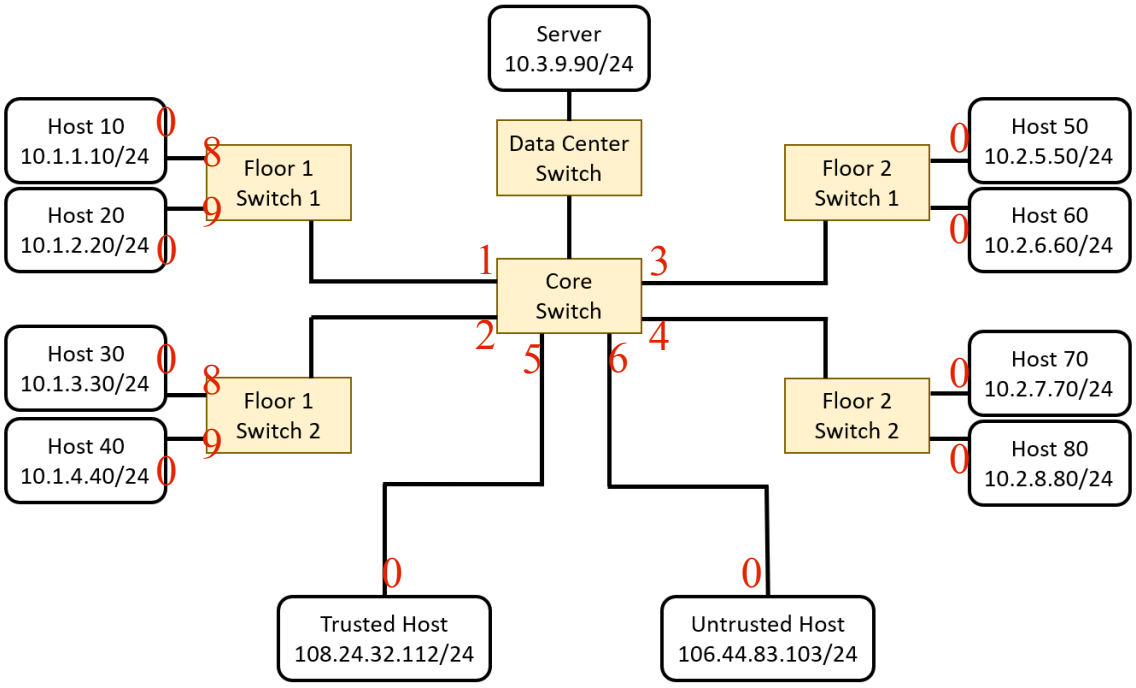
Assignment:

For this lab, we will be constructing a network for a small company. The company has a 2-floor building, with each floor having its own switches and subnets. Additionally, we have a switch and subnet for all the servers in the data center, and a core switch connecting everything together.

Your device's roles and IP addresses are as follows:

Device	Mininet Name	IP Address	Description
Floor 1 Hosts	h10, h20, h30, h40	10.1.1.10/24 10.1.2.20/24 10.1.3.30/24 10.1.4.40/24	Computers on floor 1 of the Department A in the company.
Floor 2 Hosts	h50, h60, h70, h80	10.2.5.50/24 10.2.6.60/24 10.2.7.70/24 10.2.8.80/24	Computers on floor 2 of the Department B in the company.
Trusted Host	h_trust	108.24.32.112/24	A trusted computer outside our network. This host is owned by certified employee from **Department B**.
Untrusted Host	h_untrust	106.44.83.103/24	An untrusted computer outside our network. We treat this computer as a potential hacker.
Server	h_server	10.3.9.90/24	A server used by our internal or trusted hosts.

The topology will look as follows:



Your goal will be to allow or block traffic between the hosts and servers. In this assignment, you will be allowed (and encouraged) to flood all non-IP traffic in the same method that you did in Lab 3 (using a destination port of of.OFPP_FLOOD). **However, you will need to specify specific ports for all IP traffic.** You may do this however you choose-- however, you may find it easiest to determine the correct destination port by using the destination IP address and source IP address, as well as the source port on the switch that the packet originated from. Additional information has been given to you in the do_final() function to allow you to make these decisions. Please see the comments in the provided code for guidance.

Requirements:
To protect our servers from the untrusted Internet, we will be blocking all IP traffic from the Untrusted Host to the Server. To block the Internet from discovering our internal IP addresses, we will also block all ICMP traffic from the Untrusted Host to anywhere internally (i.e., Host 10-80 and the Server). For the trusted host, it can send any traffic to the hosts in the Department B (Host 50, 60, 70, 80). Meanwhile, similar as the untrusted host, the trusted host cannot send any ICMP and IP traffic to the server, and it cannot send ICMP traffic to the hosts in the Department A (Host 10, 20, 30, 40).
Additionally, all ICMP traffic from the hosts in Department A (Host 10, 20, 30, 40) to the hosts in Department B (Host 50, 60, 70, 80) should be blocked and vice versa.

You need to prepare a 5-10 min demo presentation with a TA or Grader. In your demo presentation, you will be asked to explain how you implemented the various requirements and show that they work properly to the TAs. If you need help figuring out how to do this, look back to previous assignments and see how you tested them.

Provided Code:

Available in a ZIP file here.
https://users.soe.ucsc.edu/~qian/code/final_project.zip

We have provided you with starter code (skeleton files) to get you started on this assignment. The controller file (final_controller_skel.py) needs to be placed in ~/pox/pox/misc, and the mininet file (final.py) should be placed in your home directory (~). This time, you will need to modify both files to meet the lab requirements.

You will be using slightly different commands to create the Hosts and Links in the Mininet file to give you more information to make decisions within the Controller file. Additionally, you will notice that you have additional information provided in the do_final function. This is documented in the comments within the files.

Summary of Goals:

- Create a Mininet Topology (See Lab 1 for help) to represent the above topology.
- Create a Pox controller (See Lab 3 for help) with the following features:
    - All hosts are able to communicate, EXCEPT:
        - Untrusted Host cannot send ICMP traffic to Host 10 to 80, or the Server.
        - Untrusted Host cannot send any IP traffic to the Server.
        - Trusted Host cannot send ICMP traffic to Host 10 to 40 in Department A, or the Server.
        - Trusted Host cannot send any IP traffic to the Server.
        - Hosts in Department A (Host 10 to 40) cannot send any ICMP traffic to the hosts in Department B (Host 50 to 80), and vice versa.

Testing:

You may test with mininet commands and observing packets with Wireshark inside your VM. We will **not** be telling you what commands to run to verify the assignment goals are met in this document. Figuring out how to prove your work is a part of this assignment. Please test your code comprehensively before coming to the demo sessions.

Grading Rubric:

Total: 100 points
You have submit the code before the demo presentation.

20 points: Mininet Topology (tested with mininet commands in the demo)
- 10: Devices are successfully created**. Please name your hosts and servers using the names specified in the "Mininet Name" column in above table.**
- 10: Links are successfully created, and the topology is correct.
- 10: IP addresses are correct.

60 points: Pox Controller (tested with mininet commands in the demo)
- 20: Hosts can communicate.
    - 10 point deduction if rules not installed in flow table.
    - 10 point deduction if IP traffic is implemented using OFPP_FLOOD.
- 10: Untrusted Host cannot send ICMP traffic to Host 10 to 80
    - 5 point deduction if Untrusted Host cannot send ANY traffic to the hosts.
- 10: Untrusted/Trust Host cannot send any traffic to Server
- 10: Trusted Host cannot send ICMP traffic to Host 10 to 40
    - 5 point deduction if Trusted Host cannot send ICMP traffic to Host 50 to 80

10: Host 10 to 40 cannot send ICMP traffic to Host 50 to 80

20 points: Quality of your demo presentation with the TAs.
You will need to attend the demo sessions and explain your code and results to the TAs/Tutors. Credits are given based on the clarity of your illustration and result justification in the demo presentation.
Before attending the demo sessions, **you have to submit your code files on Canvas**.
Partial credit may be awarded for incomplete assignments based upon the submitted code and explanations in the demo as to why something may not be functioning properly. If you are not able to get the expected results, you could explain what you think is going on (for partial credit).

10 extra points for Early Birds: You can choose to submit your code and present the demo earlier than the project due date (Mar 7), then you can gain **10 extra points** for the project. To do that, you need to commit your code on Canvas **by 11:59:59 pm on March 1st** and sign up the early bird demo section slots in **Week 9**.

The arrangement of the demo session schedules will be announced in class.

Deliverables on Canvas:

1. **final_skel.py**: Your topology code.
2. **finalcontroller_skel.py**: Your controller code.
3. **README.txt**: A readme file explaining your submission.