Final Project: Implementing a Simple Router

In the previous labs you implemented simple firewalls based on different rules. 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.

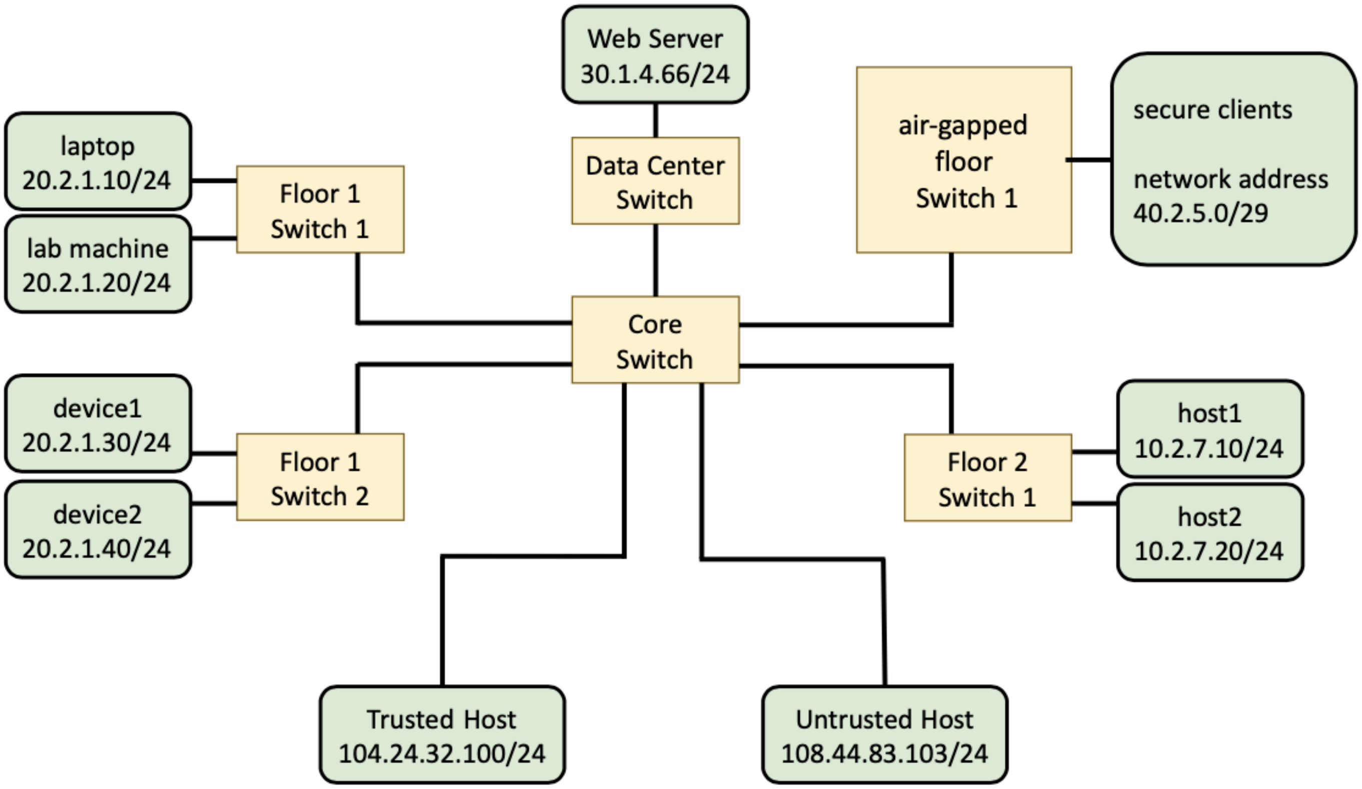
Assignment:

For this lab, we will be constructing a network for a small company. The company is in a 3-floor building, with each floor having its own subnet, switches and requirements. The third floor is an air-gapped floor. An air gap, air wall, air gapping or disconnected network is a network security measure employed on one or more computers to ensure that a secure computer network is physically isolated from unsecured networks, such as the public Internet or an unsecured LAN (Local Area Network). Additionally, we have a switch that connects us to the Web server in the data center, and a core switch connecting everything together (see topology).

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

Device	Mininet Name	Subnet address	IP Addresses	Description
Floor 1 Hosts	laptop, lab machine, device1, device2	20.2.1.0/24	20.2.1.10/24 20.2.1.20/24 20.2.1.30/24 20.2.1.40/24	End devices on floor 1 of Department A in the company.
Floor 2 Hosts	host1, host2	10.2.7.0/24	10.2.7.10/24 10.2.7.20/24	End devices on floor 2 of Department B in the company.
Air-gap floor Secure clients	Define max number of clients possible for the assigned subnet	40.2.5.0/29	Assign IP addresses based on the provided subnet address	Secure client devices on the air-gapped floor in the company. Named from client1 to clientX. (X = max possible number)
Trusted Host	h_trust	104.24.32.0/24	104.24.32.100/24	A trusted computer outside our network. This host is owned by a certified employee from Department B.
Untrusted Host	h_untrust	108.44.83.0/24	108.44.83.103/24	An untrusted computer outside our network. We treat this computer as a potential hacker.
Web Server	h_server	30.1.4.0/24	30.1.4.66/24	A web server used to serve HTTP requests

The topology will look as follows:



Your goal will be to allow or block traffic between the different devices, based on the rules given below. 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 -- although as a suggestion, 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.

Network setup and rules to be implemented

- Floor3 consists of secure devices that are completely isolated from the network, as it is the air-gap floor. The devices can only communicate amongst themselves.
- Untrusted Host cannot send ICMP traffic to any of the devices on floor1 and floor2, or the Web Server.
- Untrusted Host cannot send any IP traffic to the Web Server.
- Trusted Host cannot send ICMP traffic to end devices on floor1 of the Department A, or the Web Server.
- Trusted Host cannot send any TCP to the Web Server.
- Hosts in Department A cannot send any ICMP traffic to the hosts in Department B, and vice versa.

Provided Code:

Available in a ZIP file here.

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

In this assignment, because there are multiple switches, you will be using slightly different commands than previous labs to create the Hosts and Links in the Mininet file. Additionally, you will notice that you have additional information provided in the Controller file do_final function. (This is documented in the comments within the files.)

Testing:

You may test with Mininet commands and observe 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.

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 required rules as mentioned above.

Prepare for a 5-10 min demo presentation with a TA. In your demo presentation, you will be asked to explain how you implemented the various requirements and expected to show that they work properly. If you need help figuring out how to do this, carefully recheck your previous assignments and see how you tested them.

Grading Rubric:

Total: 100 points
Your code must be submitted by the submission deadline.
If you do not attend your demo, there is no credit given for the project.
There will be no rescheduling of the demo section that you signed up for.

20 points: Mininet Topology (preprocessing - we test submitted code before your demo)

10: Devices are successfully created**. Please name your hosts and servers using the names specified in the "Mininet Name" column in the above table.**
5: Links are successfully created, and the topology is correct.
5: IP addresses are correct.

60 points: Pox Controller (preprocessing - we test submitted code before your demo)

- 20: Hosts can communicate.
  - 10-point deduction if rules are not installed in the flow table.
  - 10-point deduction if IP traffic is implemented using OFPP_FLOOD.
- 10: Untrusted Host cannot send ICMP traffic to any of the devices on floor1 and floor2
  - 5-point deduction if Untrusted Host cannot send ANY traffic to the devices.
- 10: Trusted Host cannot send ICMP traffic to end devices on floor1 of the Department A, or the Web Server
  - 5-point deduction if Trusted Host cannot send ICMP traffic to devices on floor2
- 10: Hosts in Department A cannot send any ICMP traffic to the hosts in Department B
- 5: Untrusted Host cannot send IP traffic to Web Server
- 5: Trusted Host cannot send TCP traffic to Web Server

20 points: Quality of your demo presentation with the TAs.
You must attend the demo session and explain your code and results to the TAs. Credit is given based on the clarity of your illustration and result justification in the demo presentation.

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: If you submit your code early by **November 21st at 11:59pm** and do your demo during **Week 9**, then you can gain **10 extra points** for the project.

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.