Systems Security
Homework #2
Due: 1pm on 2/13/2019

In this homework you will become familiar with the Open Networking Laboratory (ONL); this is the environment that we work with when we require machines to be connected. While there is a way to accomplish this with virtual machines, it is more interesting when bits are actually moving across the wire. Also, it will be easier to see how the feats we accomplish in class could be performed in real networks.

You have been given read-access to the hw2-notes document, which you will need in order to complete this assignment.

A few key points:
- ONL has a finite amount of resources. It would be wise to start the homework early, in an attempt to spread out the usage.
- ONL account applications are fielded by a real person. He will not be awake at 3AM the day before the assignment is due to accept your application. **For your own benefit, get the signup done ASAP.**
- While it is possible to complete this assignment on a variety of platforms, please work inside of the class VM you created in Homework 1. This will aid with any troubleshooting, and also help ensure fair grading standards among students.

A few remarks from your TAs:
- Because the ONL is hosted at WashU, connection while on the campus network is going to be the best possible way to log in. Last semester, there were myriad problems with less than optimal internet connections interrupting the work that students were doing with ONL. If you suffer these same issues, please make a post on Piazza, and consider coming to campus as a potential way to mitigate these issues as we go forwards this semester.
- We very much encourage snapshotting your virtual machines before and after homework assignments, if you have the luxury of that much space. Snapshots have, more than once, saved both of the TAs from many hours of re-creating work that otherwise would have been lost thanks to a broken virtual machine.
- Finally, if you're finding a particular tool, like less, grep, gdb, etc. cumbersome to use, please make a post on Piazza! Your peers and your TAs, not to mention your professor, are great resources that are available to you there, and we can almost certainly help make your life a little easier on the command line. :)

1. Understand the purpose of ONL

To provide further context of why we use ONL and what ONL provides us, visit https://onl.wustl.edu and read the first five paragraphs on the homepage. To be explicit, the last sentence you are required to read is "An example of such a chart is shown to the right." The information on the routers is out of date and is in the process of being updated, but it is sufficient to understand the main idea.

Here are some other important details. ONL is essentially a closed network, which means that traffic originating inside the testbed cannot reach the rest of the Internet. Additionally, the only machines that users can access are the ones they have been allocated for their experiment.

Once you have read the paragraphs, answer the following questions in your hw2-notes document:

1. In your own words, what does ONL provide to end users?
2. What tool do users work with to create a new topology?
3. User configurations are sent to what entity in ONL for processing?
4. In your own words, why might monitoring traffic counters be useful?

2. Sign up for an ONL account

In order to use ONL ourselves, we must sign up for an account. (If you already have an account from a previous use of ONL, you can use that account. Try logging in to make sure you remember the password.) Since you are already on the ONL homepage, click "Get an account," fill out the necessary information, and submit the application. If a prompt does not apply to you, you should be able to leave it blank. For the reason why you want an account, please say something to the effect of "I'm a student in cse523 this semester."

Again, remember that these applications are processed by a real person (who will likely not be up at 2AM just to approve your ONL account request). However, within 24 hours of applying you should have been granted access to ONL.

3. Ensure the proper working environment

We need to make sure some things are in working order before we can use ONL. Once you are logged into the ONL homepage, click the "Getting Started" item on the menu.

1. Click the "Get RLI.jar" hyperlink to download the Remote Laboratory Interface (RLI).
2. We need to download the Java Runtime Environment (provided by Sun specifically) to

run the RLI jar file. Open a terminal, enter the following commands, and go through their respective prompts:

```
sudo add-apt-repository ppa:webupd8team/java
sudo apt-get update
sudo apt-get install oracle-java8-installer
```

At this point, java version 1.8 should be installed. Verify this by navigating to the directory your RLI jar file is in and running it with the command "java -jar RLI.jar". A graphical menu should load, which you can immediately close.

3. In your terminal window, enter the command "ssh {username}@onl.wustl.edu" to ensure that you can reach the ONL base server. The symbol {username} should be filled in with the ONL username that you registered under. Please include the contents of your user directory on the ONL base server (via ls) in your hw2-notes.

4. Set up the experiment

You will need three terminal windows for this section. The first will run the RLI.jar file, which we saw in the previous section. The second will be an SSH tunnel so that the RLI application can communicate with the ONL controller (otherwise it will not work). The third will be an SSH connection to the ONL base server, which we saw in the previous section.

1. Set up the SSH tunnel. The command is
   ```
   ssh -L 7070:onlsrv:7070 {username}@onl.wustl.edu
   ```
   Do not work in the SSH window created by this connection. Sometimes weird things can happen, and you don't want your RLI to get messed up halfway through a reservation.
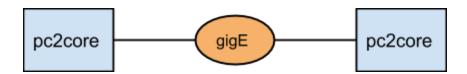2. Navigate to the RLI.jar file and launch it.
3. SSH to ONL.

There are three main stages of running an ONL experiment: making the reservation, committing the experiment, and closing the experiment. However, before we can start this sequence, we must have a topology to commit.

In the RLI graphical interface, go to "Topology" --> "Add PC2core" --> "Add Host2core" and specify 2 instances rather than 1. It looks like only one image pops up in the RLI, but click and drag the machines apart and you will find they were placed one on top of the other. You should see two separate machines.

Now go to "Topology" --> "Add GigE Switch", select "Enter", and move the switch in between the two hosts. Click the "Links Off" button on the topology to be able to draw links, and draw links between the switch and each of the hosts. This should create a connection between the switch and the two machines. Click "Links On" to go back to the click and drag mouse. At this point we have a specific topology to request. The topology should look something like the following

picture, probably with slightly different labels:

1. Go to "File" --> "Make a reservation". Go through the menu. Fill in the amount of time you would like the reservation for. Even if you want to poke around, 30 minutes should be a generous reservation length. Before your reservation is over, the RLI will prompt you if you'd like to extend your reservation; assuming there is no conflicting reservation ahead of you, it will actually extend them.

   Once your reservation is granted, you have real hardware allocated to you during that time interval, and that hardware will be setup in such a way that it logically fulfills your topology. On the pop-up that confirms your reservation, notice the time when it starts. It may be now, or if ONL is busy it may be in the future. If it is in the future you will have to wait until that reservation time comes to take the next steps.
2. Go to "File" --> "Commit" to actually initiate the setup of the hardware. When this completes, the machines will be ready for you to use.

See the "Appendix" section of the document if there are any issues in the commit process or while the experiment is running.

5. Do the experiment

The hosts are given logical names which correspond to their IP addresses. For example, the host at 192.168.1.1 in your experiment can be called h1x1. In order to use these logical names, in your SSH connection to the ONL base server type

        source /users/onl/.topology

There is a slight subtlety. When SSHing into the host machines, you precede this logical name by a dollar sign. In this case, the logical names are shell variables. However, when using it elsewhere, i.e. as the target of a ping, we use the logical name without the dollar sign. In this case, the logical name is an actual hostname. For the purposes of this section, we will use the logical names h1x1 and h1x2 for the two hosts.

1. From the base server, SSH into h1x1 by doing the command "ssh $h1x1".
2. Once logged into h1x1, type "uname -a" and include this output in your hw2-notes.
3. Type "ping -c 10 h1x2" and include this output in your hw2-notes.

Lastly, we will do something interesting with our reservation. The gigE switch that we have in our topology should ensure that there is a link between our two hosts rated at 1 gigabit per

second (Gbps); gigE is short for Gigabit Ethernet. Networking researchers and practitioners often need to measure the throughput of connections. In this case, we will measure the TCP throughput between the two hosts.

1. While still logged into h1x1, type "iperf3 -s -D". iperf3 is a traffic generator with a client / server architecture. We just created the server on h1x1.
2. Exit h1x1 and log onto h1x2.
3. Type "iperf3 -c h1x1" and hit enter. We just created the client, which initiates a 10 second TCP transfer between the two hosts. Paste your results in your hw2-notes document.

Now we can close the experiment. Go to "File" --> "Close" on the RLI graphical interface. Any open SSH connections to the host machines will be closed automatically, because the machines are being rebooted. Now select "File" --> "Exit", and when prompted to cancel your reservation, select yes. You can close your remaining terminal windows at this point.

6. Finalize your hw2-notes file in Google Docs.

You should know how to do this after Homework 1. You are strongly encouraged to revisit the instructions if you are unsure how the notes should be written.

Appendix

If there is a problem and you have already committed, wait until you can select the "File" --> "Close" option successfully. When you close the topology, the ONL control system restarts those machines to wipe their slate clean. It is not uncommon for this to fix whatever problems were being encountered. If you have waited a fair amount of time (5-10 minutes) and the option still does not present itself, you can close the RLI manually (by closing the terminal window, pkill, etc) and then canceling your reservation from the web interface. At this point you can start from scratch.