**Denial of Service Basics**

**Lab Overview**

In this lab, you will implement a computer network Denial of Service (DoS) attack. This attack is designed to deprive services, such as web content, from legitimate users. To experience this type of attack, we will be using GENI Infrastructure to simulate a network of connected hosts, similar to hosts on the internet. We will use a variety of tools to measure performance and analyse network traffic. After creating the DoS condition, we will reuse the same tools to perform comparative analysis on the network traffic. After the analysis, we will write documentation detailing our findings and answer questions listed in the rubric.

**Preparation**

**Before you start, you will need to complete with the following setups (If you have complete some steps before, you can skip them):**

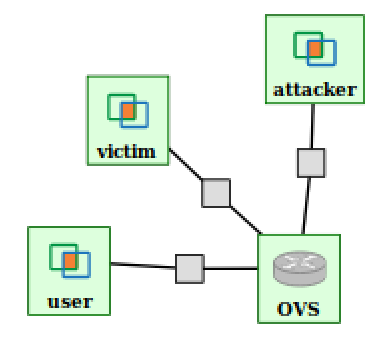
1. A working terminal program (see Setup-VM). In this lab, we suggest you set the memory size of your VM to 4GB, so that you can use wireshark on it.
2. Setup user account on GENI and join a Project. (see Setup-User)
3. Setup Lab in GENI (see Setup-Lab). The RSpec file location is shown as follows.
4. Establish SSH connections from host (VM) to each machine in the Lab (see Setup-Lab)

**RSpec File Location:**

<https://raw.githubusercontent.com/DrVoyager/EdGENI/master/Rspec-Files/DenialOfService-Rspec.txt>

**Network Topology**

Here is our network topology for this lab. Even though our machines are real, the DoS attack will be a simulation. We need to thoroughly understand the topology before we really understand our simulation. In this lab, we see 4 different devices: 1 network device and 3 end devices. The network device (ovs) functions as a router, but in our lab/simulation this device could be thought of as the internet or as the ISP these machines are all connected to. The 3 end devices will represent computers or servers connected to the internet (ovs).



In an overview, the Victim machine will be representing a server on the internet providing some service like hosting a website. Our Victim/server can provide its service to multiple users and multiple users can interact with it simultaneously. The User and Attacker machines both represent users on the internet. The User machine will be our control to represent normal use with the Victim/server. The Attacker will have more malicious intent than the User, and will be performing our DoS attack against our Victim/server. This attack will affect the performance of Victim/server and resulting in the Victim/server becoming unresponsive. If this simulation were to be real, the service (website) would become unavailable to all users.

**Task 1 – Measure Normal Network Traffic**

1. The ping command is a query, or a question, to another computer on a network. In this case a question is sent from the "user" machine to the "victim" machine to determine whether there is a connection. On the "user" machine. Type: **ping victim** and hit enter.

Type **Ctrl+C** to escape the ping command as we do not want this to run forever. **Screenshot the result and include in documentation.**

1. Tcpdump is a tool used to analyse traffic on a network. This tool prints out a description of the contents of packets on a network interface. We will use tcpdump to observe what is happening in Task 1-A. On the "victim" machine. Type: **sudo tcpdump -i eth1** and hit enter. Now go back to the "user" machine, type: **ping victim** and hit enter. After a few seconds, Type **Ctrl+C** on both terminals.

**Screenshot the tcpdump results and include in documentation with an explanation of what they mean.**

**Task 2 - Perform DoS Attack**

1. Iperf is software that tests the performance of a machine; how fast it serves network requests. The "victim" machine will run a iperf server and the "user" machine will run a iperf client with requests to the server on the "victim" machine. On the "victim" machine. Type: **iperf -s** and hit enter. Now go to the "user" machine. Type: **iperf -c** **victim** and hit enter. Wait a few minutes then **Screenshot** the lines that were printed on the "user" terminal. On “victim” machine, Type: Ctrl+C to stop the iperf server. **Include iperf client Screenshot in the documentation and explain what does that mean.**
2. On the “victim” machine, Type: **ping ovs** and hit enter. Note the IP address inside the parenthesis. Now go to the "ovs" machine. Type: **ifconfig** and hit enter. Look for the IP address noted earlier inside the output from the ifconfig command and remember the specific "eth" it is paired with. This will be a number that represents the ethernet interface that the “victim” machine is connected to on the “ovs” router, i.e. “eth1” or “eth2”.

We want to use tcpdump again, but this time run it specifically for the interface of the “victim” machine. On the “osv” machine, (using the “eth#” that was discovered earlier) Type: **sudo tcpdump -i eth# -vv** and hit enter. Now go to the "victim" machine. Type: **iperf -s** and hit enter. Next go to the "user" machine. Type: **iperf -c** victim and hit enter. **Screenshot a few lines of the result of tcpdump and include/explain what it means in the documentation.**

1. Now let’s perform the DoS attack. On the "attacker" machine. Type: **sudo hping3 -S --flood victim** and hit enter. After a few seconds, Type: **Ctrl+C** to stop the attack. **Include Screenshot of ovs’s tcpdump results in the documentation with an explanation of what is happening regarding the attack.**

**\*\*\***If hping3 does not work, it might need to be installed on the “attacker” machine. These two commands should download/install hping3:

**wget ftp.us.debian.org/debian/pool/main/h/hping3/hping3\_3.a2.ds2-7\_amd64.deb**

**sudo dpkg -i ~/hping3\_3.a2.ds2-7\_amd64.deb**

**Task 3 - Measure DoS Network Traffic**

1. We will now collect normal traffic and DoS traffic data and compare them. First, go to the "osv" machine and hit: **Ctrl+C**. Then type: **sudo tcpdump -i eth# -s0 -w capture1.pcap** (where eth# corresponds to the same "eth" you used in Task 2-B) and hit enter. Now in the "osv" machine terminal, This command takes the tcpdump monitored traffic and writes it in a file called capture1.pcap.

Go to the "victim" machine and press **Ctrl+C**. Then Type: **iperf -s** and hit enter.

Go to the "user" machine. Type: **iperf -c victim** and hit enter.

Go to the "attacker" machine. Type: **sudo hping3 -S --flood victim** and hit enter.

After 5 seconds, press Ctrl+C on your attacker machine. It will stop the hping3 command. Press **Ctrl+C** on all other terminal windows, including victim, user, and osv.

**NOTE**: Stop the hping3 command immediately when 5 seconds elapse. Don’t wait for too long. Otherwise, you will end up with a large pcap file. The wireshark may crash when processing a large pcap on our VM.

**Describe what happened and take Screenshots.**

**Task 4 - Analyze DoS Network Traffic with Wireshark**

1. Now we will analyze the information gathered after generating an attack. To do this we need to transfer the capture1.pcap file from the “osv” machine back to our host computer i.e. the terminal used to ssh into “osv” machine.

To do that we will use sftp. We will open this connection just like we did for ssh. Look at the details section of your slice on the GENI portal. Find the details entry for the node with client ID of “osv”. Copy the ssh line associated with your username into a new terminal. The code should look like this format:

**ssh <username>@<domain> -p <port\_number>**

We need to change the syntax of this command to work for sftp. Like this:

**sftp -i ~/path/to/private/ssh-key -oPort=<port\_number> <username>@<domain>**

My sftp login looks like this :

sftp -i ~/.ssh/calvin\_geni\_ssh\_rsa -oPort=26098 calvinmc@pc3.instageni.ku.gpeni.net

Yours will be different, but follow the same format as above. If you specified a passphrase when creating SSH keys you will be asked to provide it. When a sftp connection has been established, your prompt will change to look like this: **sftp>**

Now type: **get capture1.pcap**

**NOTE:** this could take awhile to transfer depending on the size of the .pcap file. If you waited more than 5 seconds while capturing traffic, this file can become very large. When this transfer is complete, the .pcap file will be on your host computer. To verify this, open a new terminal and type: **ls -al**

1. Now we will look at this .pcap file using a tool called wireshark. Open a new terminal and execute the following two commands:

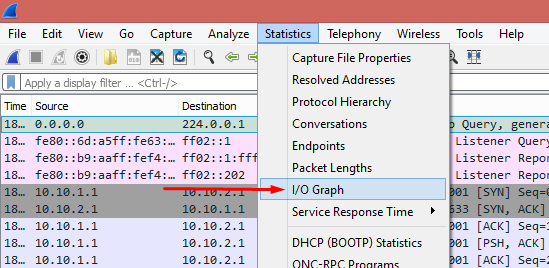
**export LIBOVERLAY\_SCROLLBAR=0**

**/usr/bin/wireshark**

**NOTE:** You can also install a wireshark in a real machine instead of vpark-ubuntu and analyze on it. It will be more reliable. If you want to use wireshark on vpark-ubuntu, to make the wireshark function normally, you are recommended to set the memory size of the vpark-ubuntu to 4GB.

Go to File →Open, and select the pcap file that you have downloaded.

Use the Statistics→I/O graph function to analyze the traffic, as shown below.



* 1. **Copy the graph. This is the graph of the traffic you ran in the previous task.**
  2. **Describe the I/O graph that is generated by your capture file. (To understand the graph, you need to read the graph legend, x-axis and y-axis values. The graph should be self-explanatory. If you have difficulty in reading the graph, you may need to read the Wireshark manuals or documentations).**
  3. **Do you see at which time you started the flooding attack? Why is it very distinctive?**
  4. **Did the attack end at some point? What do you think happened at this point?**