**Lab: Moving Target Defense**

*Dr. Prasad Calyam (Contact:* [*calyamp@missouri.edu*](mailto:calyamp@missouri.edu)*)*

Compiled by Ramya Payyavula

1. **Purpose of the Lab**

To create a cloud infrastructure that hosts a web server, attack the virtual machine hosting the service to make the service unavailable, and then migrate the status of the application running in the server to a different virtual machine to regain service.

1. **Reference to guide Lab Work**

* **Creating a LAMP stack**

<https://www.digitalocean.com/community/tutorials/how-to-install-linux-apache-mysql-php-lamp-stack-on-ubuntu-16-04>

* **SlowHTTPTest**

<https://blog.qualys.com/tag/slowhttptest>

* **Automate Root ssh**

<https://medium.com/@luck/setup-passwordless-ssh-on-ubuntu-16-04-7ac81592fee6>

1. **Lab steps and output collection guidelines:**

To complete this Lab, you will need to perform the following steps:

In the first step, you will launch a live video server using LAMP. Then you install monitoring scripts and all the dependencies to the respective virtual machines. Then, you will deploy attack scripts to a VM that behaves as an attacker. This attack is then launched to bring the server down. The controller monitoring the network senses the attack and moves the service statuses and application to an entirely new virtual machine. Then you switch the hosts to make the service available to the users once again.

* 1. **Setup Slice with servers, a controller and an attacker.**

Login to GENI portal and create slice for the lab. We are going to reserve the resources using Graphical Interface of the portal (Jacks).

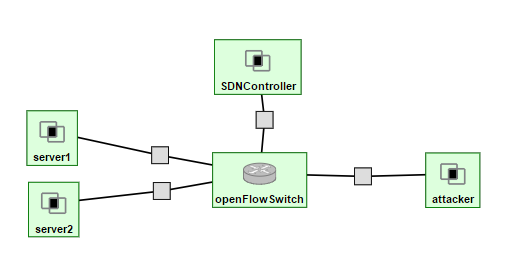
### Load a simple topology in Jacks

|  |
| --- |
| 1. In the Portal, open the Slice page for the slice you just created (Click on slices, select project and add name to the slice). Notice that you can create the slice with omni or Jacks. 2. Press the **Add Resources** button to launch Jacks for this slice. 3. From the **Choose RSpec** menu, select ‘Files’ option to load the provided Rspec file. 4. Once it is done, a few VMs and a switch appears. 5. Choose an aggregate to select where to reserve the resources and then click on “Reserve resources”. |

The topology you loaded has three VM types and a switch:

* 1. Server
  2. Controller
  3. Attacker

The sliver should look something like this:



**3.2 Install dependencies and initial setup**

* + 1. **Install LAMP stack to host web service**

Server1 and Server2 from the testbed setup are both web application hosting servers. These servers use Linux, Apache, MySQL (not mandatory), PHP, i.e., LAMP stack. For both servers, install LAMP. Since the controller is used to show the blacklisted IPs, LAMP stack should also be installed in the SDNController.

1. Update ubuntu and all installed software’s

|  |
| --- |
| *sudo apt-get update* |

1. Install Apache server

|  |
| --- |
| *sudo apt-get -y install apache2* |

1. Install PHP with Apache and mysql support

|  |
| --- |
| *sudo apt-get -y install php7.0 php-pear libapache2-mod-php7.0 php7.0-mysql php-mysql mysql-server python mysql.connector* |

**While creating MYSQL user give password as “root”**

1. Install wirehshark/Tshark for packet capture

|  |
| --- |
| *sudo apt-get install -y tshark* |

1. Override html with php (you can use vi, nano, vim or any editor of your choice)

|  |
| --- |
| *sudo vim /etc/apache2/mods-enabled/dir.conf* |



Restart apache server

|  |
| --- |
| *sudo service apache2 restart* |

1. Creating MySQL server on controller. Login to Controller console and execute following commands

|  |
| --- |
| *cd /etc/mysql/mysql.conf.d*  *sudo sed -i "s/.\*bind-address.\*/bind-address = 0.0.0.0/g"*  *mysqld.cnf*  *mysql -u root -p* |

enter your mysql password when prompted as “root”

Execute following commands in mysql console

|  |
| --- |
| *create database test;*  *create table blacklist(blacklist\_ip varchar(20));*  *CREATE USER 'monty'@'%' IDENTIFIED BY 'some\_pass';*  *GRANT ALL PRIVILEGES ON \*.\* TO 'monty'@'%';*  *CTRL+D;*  *sudo ufw allow 3306;*  *sudo service mysql restart;* |

Note: If you would like to login to the database in any of the switches use following commands

|  |
| --- |
| *mysql -u monty -h '<controllerIP>' -p* |

Example: mysql -u monty -h '72.36.65.106' -p

enter password as 'some\_pass'

1. Check for index.html file in your /var/www/html folder. It is an html page set by default by apache. You should remove this file in Server1 and Server2 using

|  |
| --- |
| *cd /var/www/html*  *sudo rm index.html* |

1. Find the compressed file called BasicStreamingService.zip Secure copy it to your server1. You can only move to your respective pawprint folder as you cannot ssh into GENI VMs root from a local machine.

(**Note:** You should execute similar scripts in a lot of following steps. Make yourself familiar to the script below to have better understanding)

|  |
| --- |
| *scp -i <ssh\_key> /your/local/path/BasicStreamingService.zip pawprint@hostname:~* |

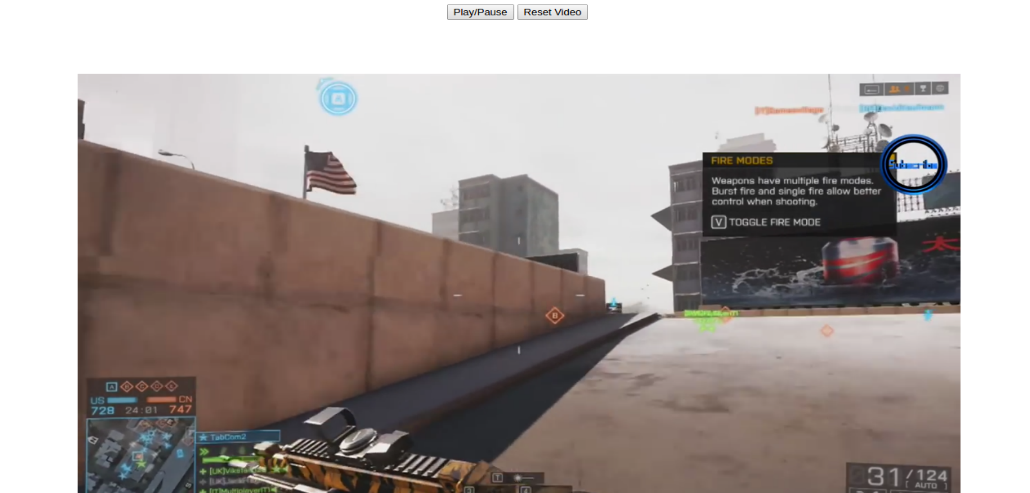
Move the zip file to /var/www/html folder

|  |
| --- |
| *sudo unzip BasicStreamingService.zip*  *cd BasicStreamingService*  *sudo mv \* /var/www/html* |

1. Change the permissions to application folder. Change directory to www and then change mode

|  |
| --- |
| *cd /var/www/*  *sudo chmod 777 html* |

1. Open the hostname of server1 or its corresponding IP in a browser to see a running application. The application looks like the following:



Similarly, copy the “index.php” and “blacklist.py” (You can find them in in your assignment folder) into the SDN controller’s /var/www/html

folder.

|  |
| --- |
| *scp -i <ssh\_key> /your/local/path/index.php pawprint@hostname:~*  *scp -i <ssh\_key> /your/local/path/blacklist.py pawprint@hostname:~*  *sudo mv \* /var/www/html* |

 In blacklist.py update your server1\_IP at

The file works as the index page for controller’s interface. The main purpose of this script and the administration at the controller side is to display the blacklisted IP.

**3.3 Implementation scripts**

**Note:** Most of these scripts need minor changes. Every VMs used hold a public IP and will differ from every resource allocation. Based on the data flow, the source and destination hosts should be configured.

1. SSH into attacker VM and copy the attack script:

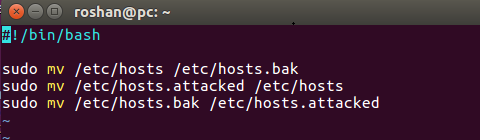
Install slowhttptest in this vm:

|  |
| --- |
| *sudo apt-get install slowhttptest* |

1. On your controller, maintain the **switchHosts.sh** script. Use an editor to change the host IPs. IP described as server1 will be your server1 IP address and the one as server2 will be your server2 IP address.

The switchHosts.sh executable file consists of a few other host files. Edit these hosts files in accordance to the comments/hints provided.

For reference, use the following screenshots.



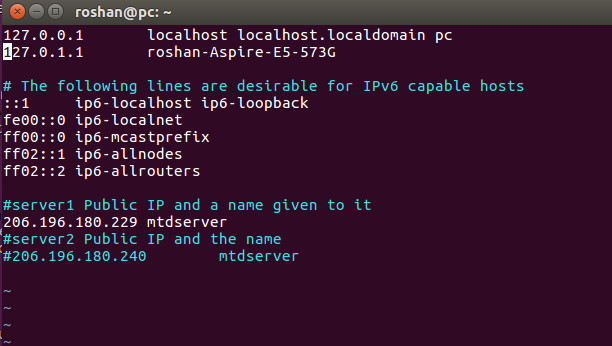
This is what the switchHosts.sh file holds. Note that hosts file in /etc folder consists of the local host and other manually assigned hosts. To tell the system which the current active server is, we alternate between the server1 and server2 using a temporary host file called **hosts.bak**. ‘hosts’ file in /etc is manually changed to make the server1 as the default server. However, **hosts.attacked** file in the /etc activates the server2. When we switch the files the servers change.

**Note:** After every implementation, the switchHosts.sh file should be executed to revert the changes.

hosts and **hosts.attacked** files should contain the public IPs of the server1 and server2 VMs. You can check their respective IPs in the GENI portal by clicking on the VMs.

The folder /etc should by default contain the ‘hosts’ file. Just add the IP addresses at the end of contents of the file. You should however create new **hosts.bak** and **hosts.attacked** files in the /etc folder of your **local** machine /etc/hosts file should look like:

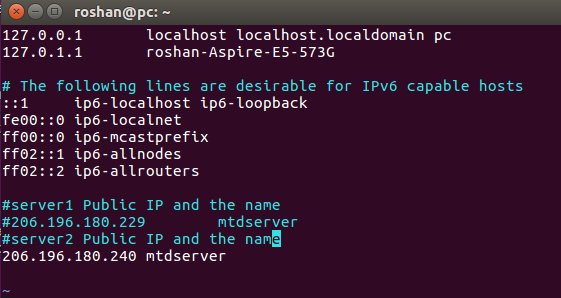
|  |
| --- |
| *cd /etc*  *sudo vim hosts* |



Change the IPs with respect to your server IPs.

The hosts.attacked file looks like:

|  |
| --- |
| *sudo vim hosts.attacked* |



Change the IPs with respect to your server IPs.

|  |
| --- |
| *sudo chmod +x switchHosts.sh* |

**3.4 Attack launch and mitigation implementation**

You will need 2 instances of Controller Console (root access), 1 instance of Attacker console, 2 instances of server1 console (root access) and 1 instance of your local machine terminal.

1. Run the watcher.sh script in one of the controller terminals. (You can find watcher.sh script in the Lab documents, you can transfer it using scp or copy manually).

|  |
| --- |
| *python watcher.py* |

This script will now listen to attacks in server1.

1. Open your browser and go to your live application. Just use the IP address of server1 on your browser. Start playing the video or maybe the video starts automatically.
2. There are multiple ways we can keep track of packets that are getting through the network. Using Wireshark or Tshark (terminal shark), we can see details of every packets that get through the network. Execute the following Tshark command on your Server1 console

|  |
| --- |
| *rm -rf capture.csv*  *sudo tshark -Tfields -e ip.src -e ip.dst -E header=y -E separator=, -E occurrence=f > ~/capture.csv* |

leave it running while the attack is going on.

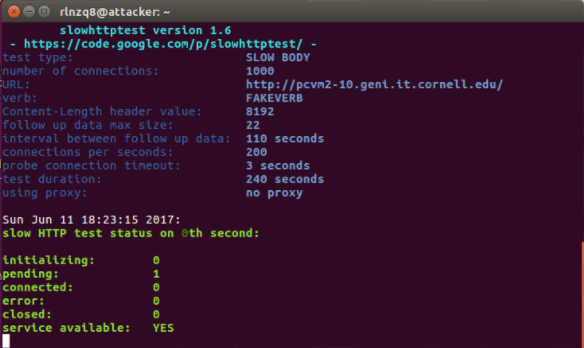
This tshark command runs tshark to check the ICMP packets getting through the network. The script also creates a log file called ping.log which holds all the packet information from using tshark.

1. Right after you do so, run the following command on Attacker Console to launch an attack on server1. Replace ServerURL with your server1 URL opened in browser.

|  |
| --- |
| *slowhttptest -c 1000 -H -g -o slowhttp -r 200 -t GET -u <ServerURL>* |

Example:

*slowhttptest -c 1000 -H -g -o slowhttp -r 200 -t GET -u* [*http://192.122.236.125/index.html*](http://192.122.236.125/index.html)



Right when the attack starts, the attacker console should look like the screenshot above.

These are two different attack tools we are using to launch the attacks. One is the SLOWHTTPTEST (see reference link) and the other one is hping3.

1. Run analyzer.sh script on Server1 Console, which will analyze the network continuously, And update the Attacker IP in blacklist table automatically in case of an attack. (You can find analyzer.sh script in the Lab documents, you can transfer it using scp or copy manually). **Replace your** **controllerIP** where you see --host=”” in analyzer.sh

|  |
| --- |
| *sudo chmod +x analyzer.sh*  *sh analyzer.sh* |

**(Note: If you face any error try retyping the contents of the file a new file and execute it)**

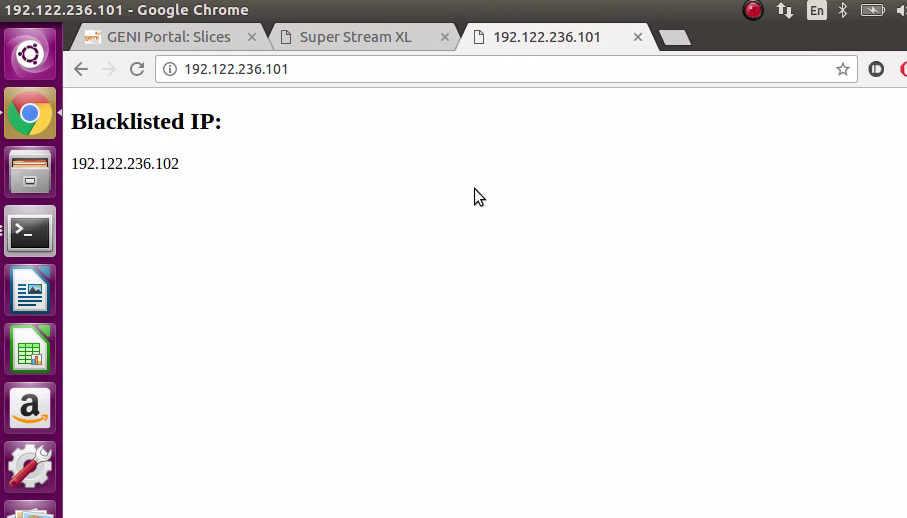
1. When attack is launched, the video streaming should stop as the server is brought down. When this happens, the number of established HTTP connections change at a greater magnitude. We use low-scale connection flooding, i.e. around 300 connections.
2. The watcher.sh script senses the attack in a few seconds and starts automatically migrating the entire application with the timestamp to server2. Do not cancel this script until you see two complete migrations (Just to make sure the timestamp is a correct one). Then the script can be stopped by pressing Ctrl + C.
3. Run the switchHosts.sh script on your controller for migration of targeted Server to a new Server.

|  |
| --- |
| **Note: Can you do “ls” and see what’s your file names are? Some users might face weird issue, where their file names are saved as filenames and question marks at the end(hosts? ,hosts.attacked?). In that case we need to change the filenames back to original filenames**  *sudo mv hosts? hosts*  *sudo mv hosts.attacked? hosts.attacked*  *sudo mv hosts.bak? hosts.bak* |

|  |
| --- |
| *sudo chmod +x switchHosts.sh*  *sh switchHosts.sh* |

1. You must have notice that the video still doesn’t work. This is because we are still on the same server. Refresh the page to see that the video starts from where it had stopped after the attack. The switchHosts.sh file changes IP address to use as the new MTDserver. Again, execute switchHosts.sh after every successful or unsuccessful implementation so that /etc/hosts contains the default contents.

After the successful completion of this phase, we can look for the blacklisted IP using the controller’s UI. To do this, just get on a browser and put the hostname of the controller or its IP on the URL section. On success, the web server on the controller should look somewhat similar to the image below.



Attacker’s IP

Controller’s IP