Deploy a Botnet and DDoS Attack Lab

Lab Description

Distributed Denial of Service (DDoS) attacks pose a significant threat to online services, leveraging multiple compromised devices to overwhelm targeted systems with traffic, rendering them inaccessible to legitimate users. This lab will simulate a Distributed Denial of Service (DDoS) attack orchestrated by multiple bots within a network, providing users with a real-time visual representation of the network traffic.

Learning Outcomes

- Understand what a Distributed Denial of Service attack is and visualize such attack.
- Understand Botnets and how they operate.
- Understand mitigation that could be placed in order to prevent a DDoS attack.

Lab Environment

- Use the SEED VM

Start-up Procedures

- Start-up Procedures will begin in Task 1.

Lab Instructions

Task 1: Set Up

1.1 Create & Move Files

To start this lab, you are going to create a new folder called **botnet-ddosLab**. We are going to copy the contents of B00-mini-interent folder and B05-botnet folder, you should be able to find these folders within the **lab-content/examples** directory.

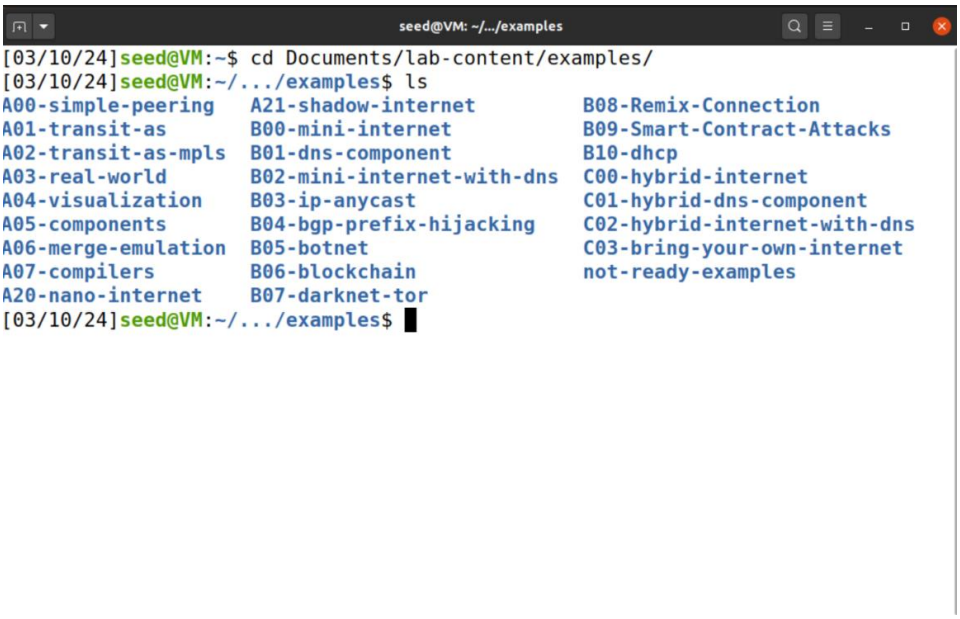


Figure 1: Contents of lab-content/examples directory.

Store the contents of these two folders in the new **botnet-ddosLab**. Also be sure to include **seedemu** folder in the **botnet-ddosLab** folder. Your folder should look something like this:

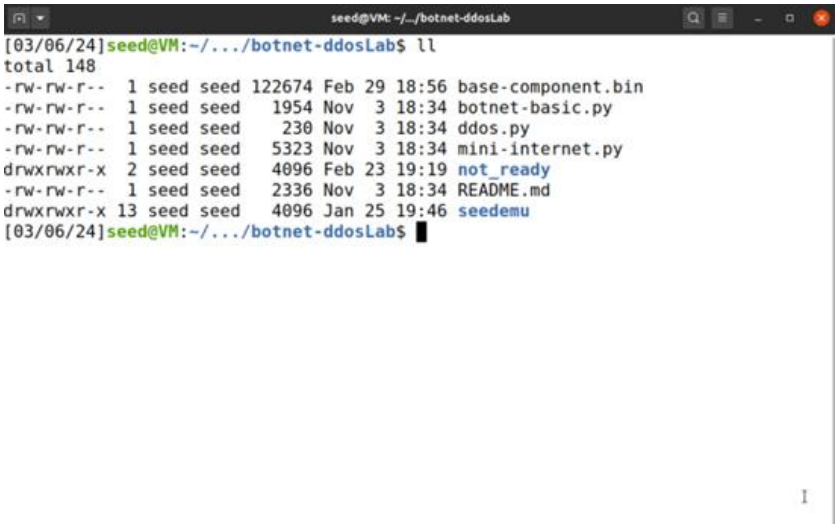


Figure 2: Contents of bot-ddosLab folder.

Use the font Courier Prime in 12 pt bold for terminal and code commands.

```
$ ls -l
```

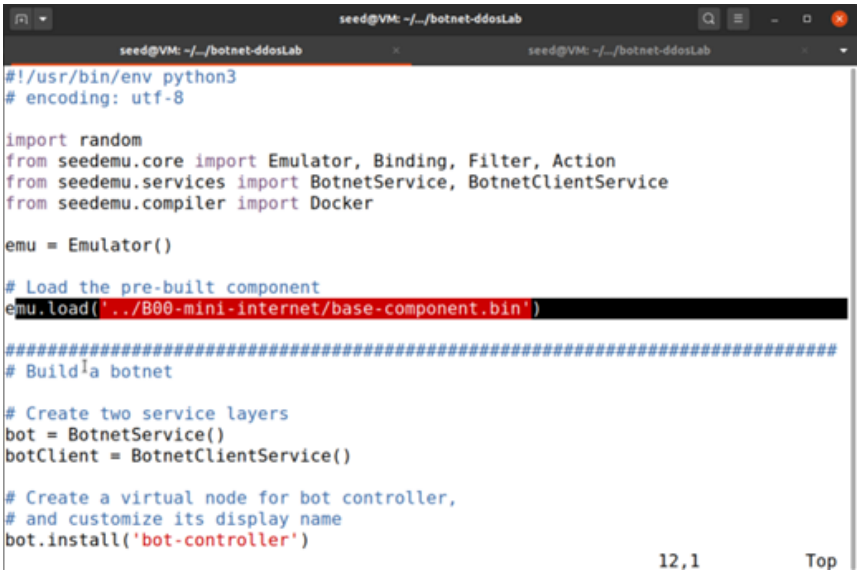
Make sure text and headers after the 1x1 table have a space before the paragraph in the paragraph settings.

1.2 Alter botnet-basic.py Script

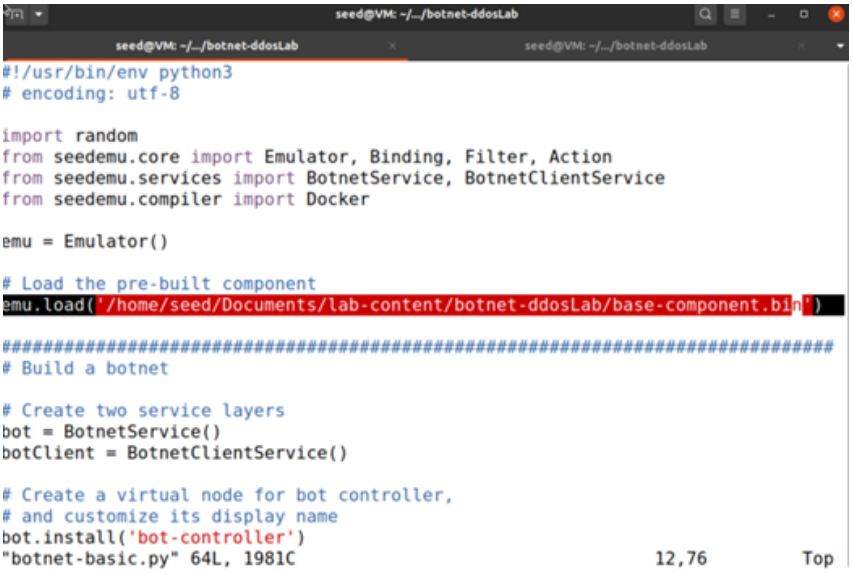
In order for this lab to run properly, you are going to need to change part of the botnet-basic.py code. Use a text editor of your preference to edit this part of the code to the current directory of the folder in which you are in (Edit line 12).

NOTE: Be sure to include "**/base-component.bin**" at the end.

Before:



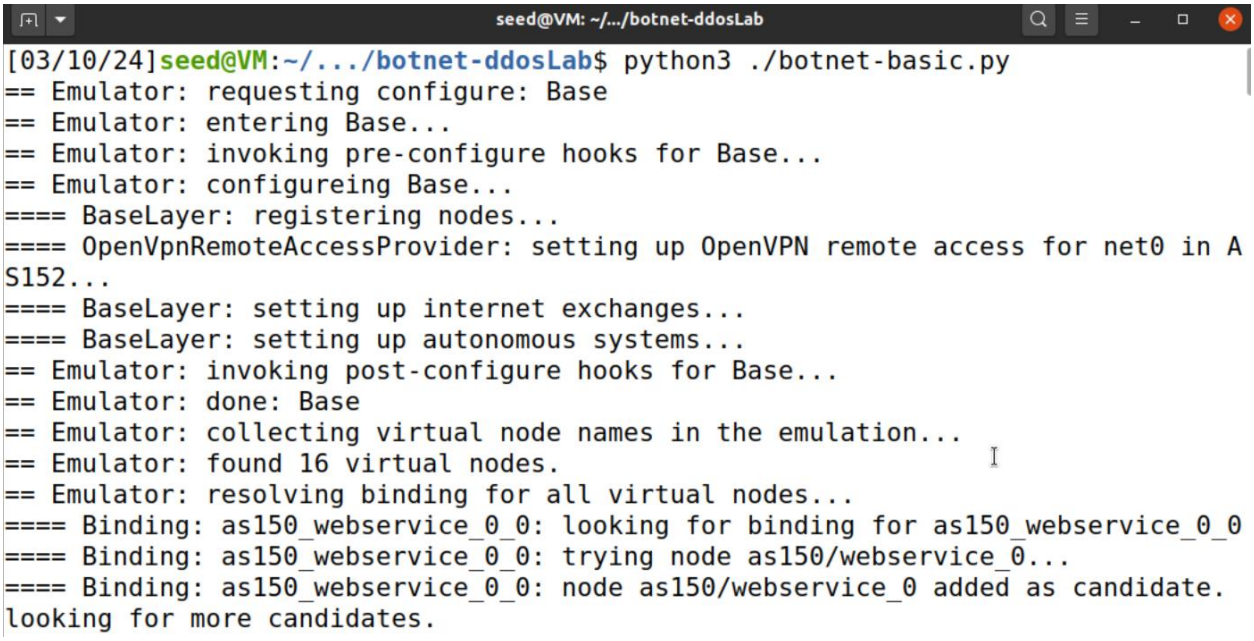
After:



1.3 Bring Up the Network

We are going to have to set up the network by running the python script.

> Run this command in the **/botnet-ddosLab** directory.
>
> `$ python3 ./botnet-basic.py`



An output folder should be created after entering this command.

Figure 3: output folder created.

```
Run these commands within the newly created /output directory.

$ docker-compose build

$ docker-compose up

$ yes | docker network prune

$ docker-compose up
```

1.4 Debugging Task (1.3)

There is a high chance that you may run into errors.

```
If unsuccessful in bringing up the network with the commands above, try these commands:

$ docker stop $(docker ps -aq)

$ docker rm $(docker ps -aq)
```

Then start again from task 1.3 and skip task 1.4.

1.5 Bring up Client Side of the Network

Now that the network is up in the **'output'** directory, we must now bring the network up on the client side. Go into the **/client** directory.

```
Run these commands withing the botnet-ddosLab directory.

$ docker-compose build

$ docker-compose down

$ docker-compose up
```

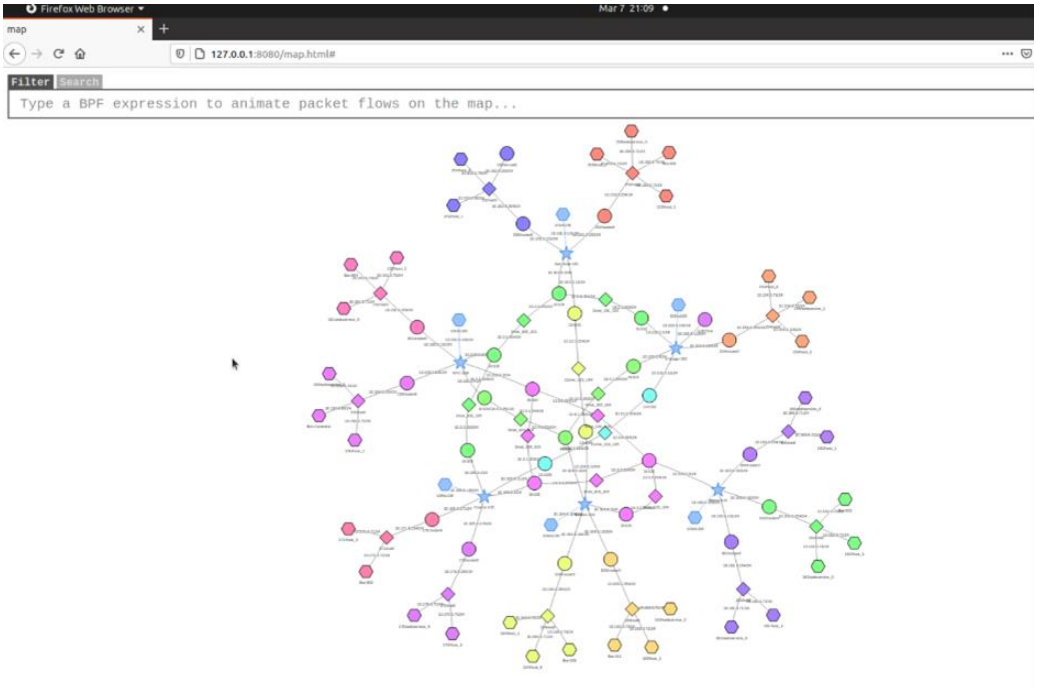
```
Successfully built 6d16e6dccaf4
Successfully tagged client_seedemu-client:latest
[03/07/24]seed@VM:~/.../client$ docker-compose down
Removing network client_default
WARNING: Network client_default not found.
[03/07/24]seed@VM:~/.../client$ docker-compose up
Creating network "client_default" with the default driver
Creating seedemu_client ... done
Attaching to seedemu_client
```

Figure 4: Client successfully launched.

1.6 Opening up the Visual Map

Open up Firefox web browser and go to this link:

- **http://127.0.0.1:8080/map.html**



Task 2 – Locating Bots and Commencing DDoS Attack

2.1 Locate Bots

There should be 6 bots located on this map, locate all 6 of them, take a screenshot and place them here (Hint: use the search bar to locate them):

Find the IP address of each bot and include a screenshot of each individual location:

Bot-00:

Bot-01:

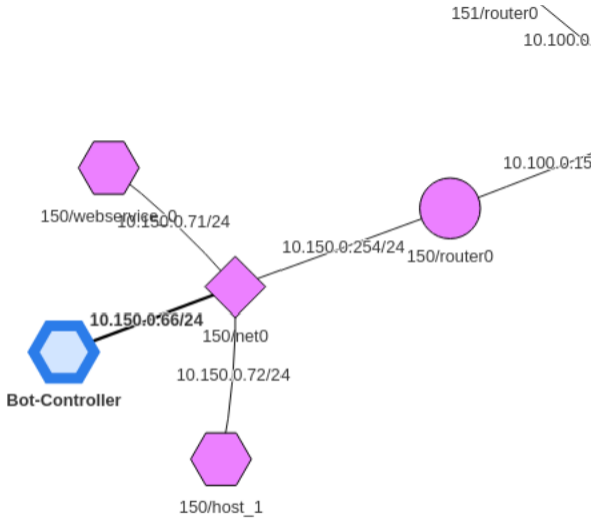
Bot-02:

Bot-03:

Bot-04:

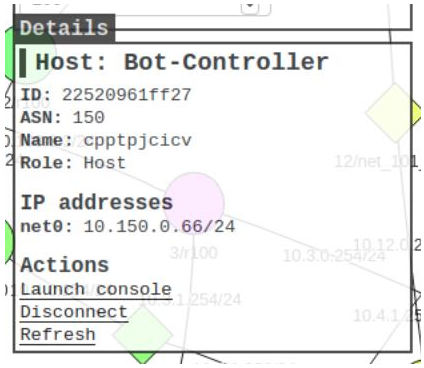
Bot-05:

Example:



2.2 Launch Console Controller

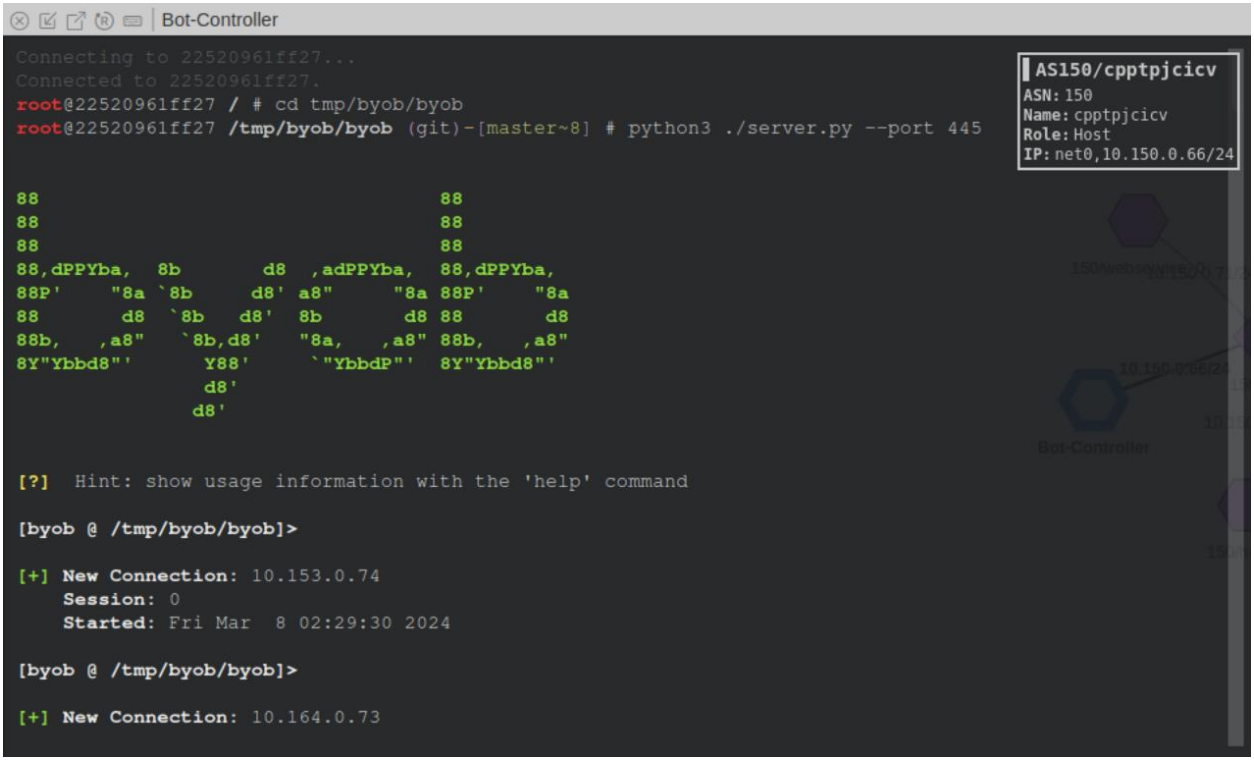
Click on the Bot-controller that is located on the network diagram and you should be able to see the details of that specific host on the right of the web page. Under 'Actions' you should be able to see 3 options, click on '**Launch Console**'.



2.3 Start the byob Server

To start the byob server, you must:

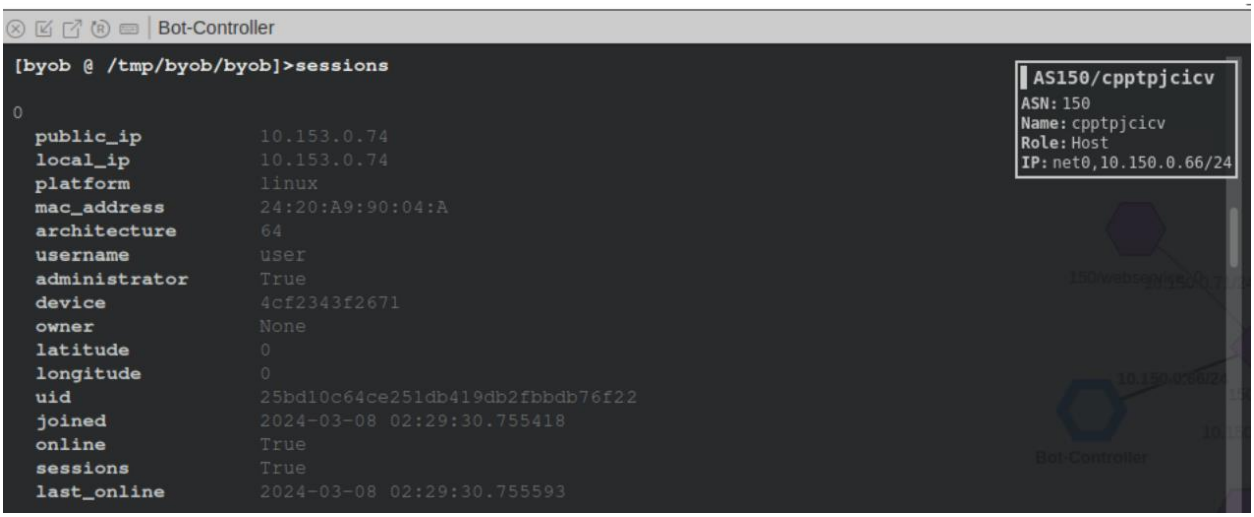
- cd into the **temp/byob/byob** folder
- run the server.py script on port 445 (hint: **python3 ./server.py --port 445**)



Enter Screenshot of your own byob server launching:

2.4 Wait for the all the Bot Nodes to Connect

There should be 6 bot nodes (takes about 15-20 seconds). From this we can type in ‘sessions’ in the console and gather information about the bot nodes.



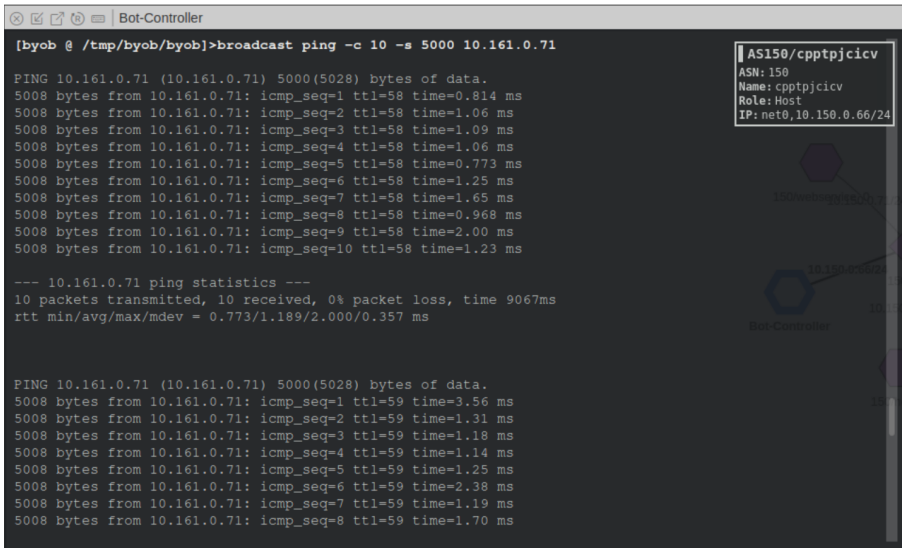
Place screenshot here of all 6 bot nodes sessions:

2.5 Launch DDoS Attack

Now it’s time to launch the attack.

To launch the DDoS, use this command:

```
> broadcast ping -c 10 -s 5000 10.161.0.71
```



In your own words, explain each step of this command here:

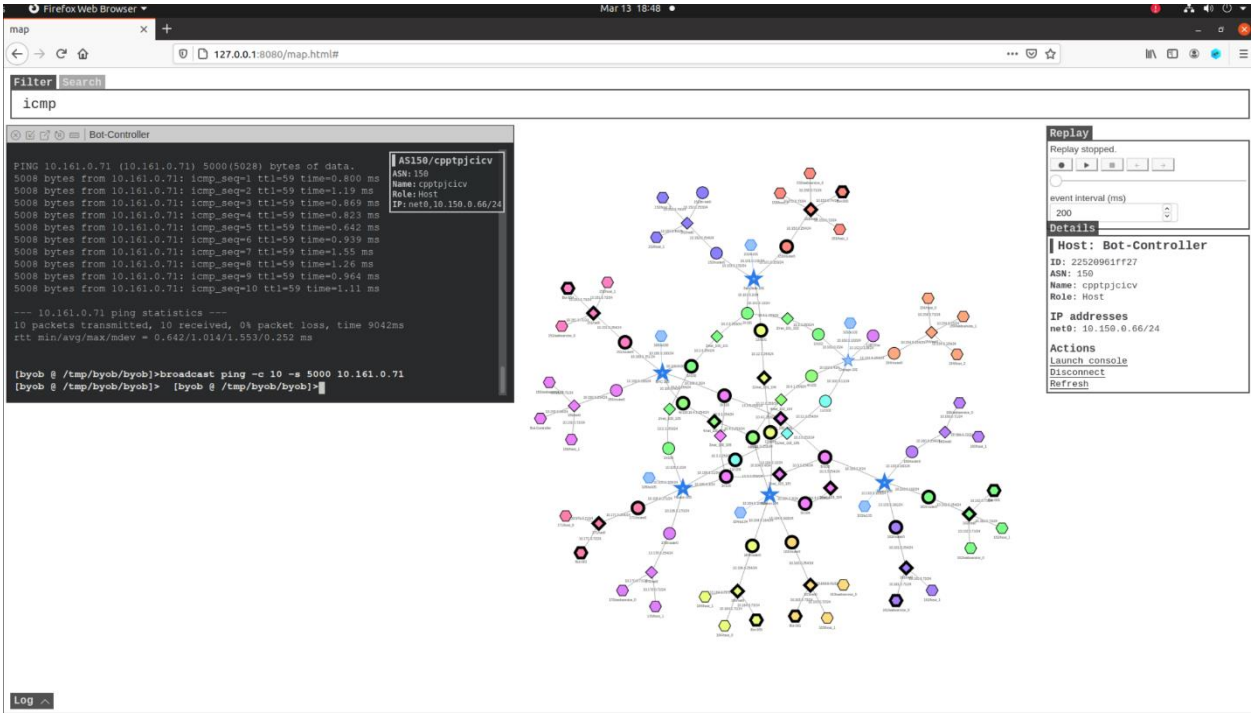
-Broadcast ping:

-c 10:

-s 5000 10.161.0.71:

2.6 Turn on ICMP Traffic

In order to visualize the attack, go back into the filter portion of the search bar and input ICMP. This will turn on ICMP traffic so it can give you a better visual of the pings happening in real time. Repeat step 2.6 to visualize traffic.



Submit your own screenshot of DDoS Visual:

2.6 Mitigations

Traffic Monitoring and Analysis: Continuously monitor network traffic for anomalies and suspicious patterns. Implement intrusion detection systems (IDS) and intrusion prevention systems (IPS) to automatically detect and block malicious traffic.

CAPTCHA and Challenge-Response Mechanisms: Implement CAPTCHA challenges or other challenge-response mechanisms to differentiate between human users and bots. This can help filter out malicious bot traffic while allowing legitimate users to access the service.

Rate Limiting and Traffic Shaping: Implement rate limiting and traffic shaping mechanisms to throttle or prioritize traffic based on predefined rules. This can help mitigate the impact of the DDoS attack by limiting the rate of incoming requests from suspicious sources.

Conclusion

The deployment of a botnet and the execution of a DDoS attack in this lab have provided valuable insights into the complexities of cyber threats. Participants have learned to control a botnet and visualize the impact of a DDoS attack in real-time. This practical experience emphasizes the importance of understanding cyber threats to develop effective security strategies. As cyber threats continue to evolve, ongoing education and hands-on training remain essential in preparing for and mitigating future attacks.