AMRITA VISHWA VIDYAPEETHAM

B Tech -AIE

21AIE302 ADVANCED COMPUTER NETWORKS

SDN-DDOS-MONITOR:

A SIMPLE MACHINE-LEARNING TOOL

FOR DETECTING AND MITIGATING BOTNET ATTACKS

- Ganga Gowri

Submitted on

23 - 01 – 2023, Wednesday

Submitted by

Team - 06

Menta Sai Akshay	**- CB.EN. U4AIE20040**
Penaka Vishnu Reddy	**- CB.EN. U4AIE20048**
Sajja Bala Karthikeya	**- CB.EN. U4AIE20065**
Dharavathu Rohith	**- CB.EN. U4AIE20060**
Rachure Charith Sai	**- CB.EN. U4AIE20055**

TABLE OF CONTENTS

ABSTRACT

Title: Sdn-Ddos-Monitor: A Simple Machine Learning Tool For Detecting And Mitigating Botnet

Software defined networking is going to be an essential part of networking domain which moves the traditional networking domain to automation network. Data security is going to be an important factor in this new networking architecture. The packets which travel through the network are prune to attacks and this paper aims to classify the traffic into normal and malicious classes based on features given in dataset by using various machine learning techniques using RapidMiner tool. The generated dataset is processed and applied on various Machine learning models. Out of which best performing model will be chosen and used to detect attacks in real-time on our SDN.

This project aims to implement different Machine Learning (ML) algorithms in RAPIDMINER tool to analyze the detection performance for DDoS attacks using realtime dataset generated on our SDN network. This research has used four different types of ML algorithms which are K_Nearest_Neighbors (K-NN), Super Vector Machine (SVM), Deep Learning (DL) and Random Forest (RF). The best accuracy result in the presented evaluation was achieved when utilizing the Random Forest (RF) algorithms.

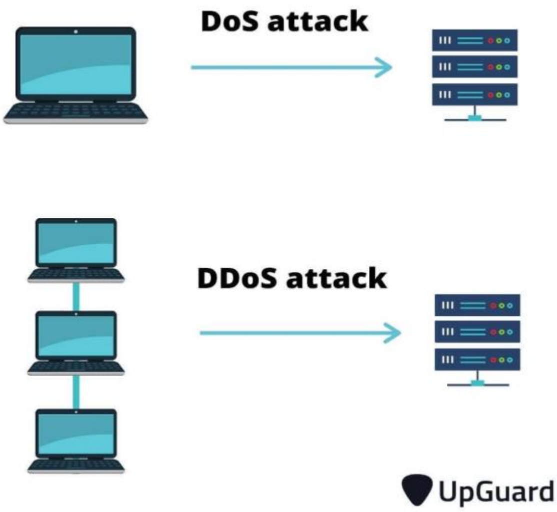
Introduction

SDN breaks the shackles of traditional network complexity and coupling and makes it possible for network architecture to satisfy flexibility, reliability and security at the same time. It separates the control plane from the data plane and separates the control function of the network from the data forwarding function. The control plane is only responsible for routing decisions, while the data plane realizes these decisions by forwarding packets and other behaviors. The separation of the two planes can improve the abstraction and programming ability of the network and makes the network structure less tedious and redundant. Although SDN architecture has many advantages compared with traditional network, it is often subjected to network threats and attacks. Network threats to SDN are mainly reflected in security device licensing and global view acquisition. Because packets are passed according to flow rules, physical security devices do not have the right to decide, and attackers can bypass security devices before deployment. The controller is the core of the entire network and can obtain various network status information. The attacker can use the controller to directly grasp the global view of the network to launch serious attacks. SDN has a clear plane structure, and the attack objects at different planes are different. At the control plane, because the controller manages the entire network, an attacker damaging the controller can cause serious damage. Controllers are the main targets of attacks in recent years.

DDoS attacks send a large amount of traffic to the network and consume network resources and cause network congestion. Many DDoS attacks are launched from distributed hosts. A DDoS attack is an aggressive and destructive network attack that causes the system to stop working by depleting system resources. It can destroy the user’s available network services, thus seriously threatening the network. When malicious data packets are sent by attackers on the network, normal traffic is processed or even cannot be processed due to the consumption of network resources. As a result, the network and servers become jammed and normal services are interrupted. Attackers who apply DDoS often target SDN mainly because of its unique characteristics

DDOS Botnet Attack

DDoS is short for distributed denial of service. A DDoS attack occurs when a threat actor uses resources from multiple, remote locations to attack an organization's online operations. Usually, DDoS attacks focus on generating attacks that manipulate the default, or even proper workings, of network equipment and services (e.g., routers, naming services or caching services). In fact, that's the main problem.When a DDoS attack takes place, the targeted organization experiences a crippling interruption in one or more of its services because the attack has flooded their resources with HTTP requests and traffic, denying access to legitimate users. DDoS attacks are ranked as one of the top four cybersecurity threats of our time, amongst social engineering, ransomware and supply chain attacks.

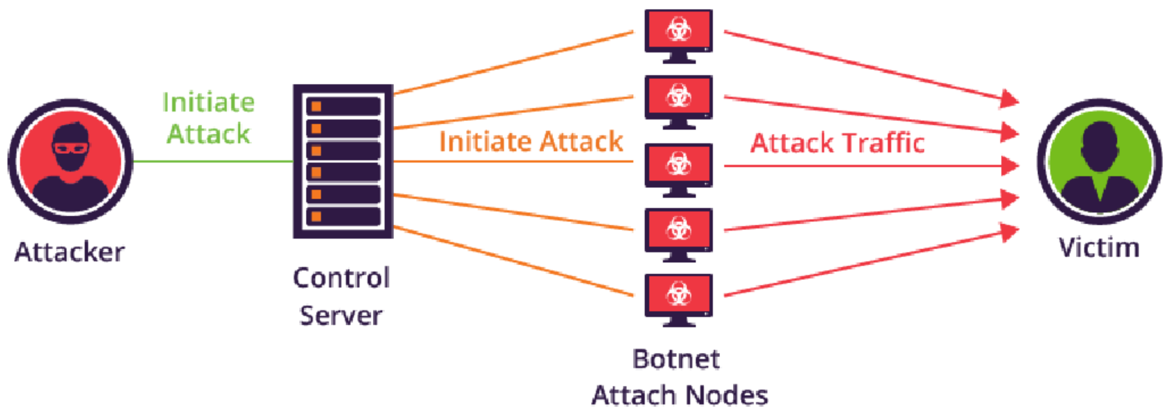


Denial of Service (DoS): Attacks using single compromised system to disrupt the authorized use of networks, systems, or applications

Distributed Denial of Service (DDoS): Employ multiple compromised computers to perform a coordinated and widely distributed DoS attack

Botnet Attack : Collection of compromised computers that are controlled for the purposes of carrying out DDoS attacks or other activities

- Can be large in number
- Systems join a botnet when they become infected by certain types of malware
- Like a virus, but instead of harming the system, it wants to take it over and control it
- Through email attachments, website links, or IM links
- Through unpatched operating system vulnerabilities



DDOS Attack Types

Flood Attack

- **TCP SYN flood :** Short for synchronize, SYN is a TCP packet sent to another computer requesting that a connection be established between them. If the SYN is received by the second machine, an SYN/ACK is sent back to the address requested by the SYN. Lastly, if the original computer receives the SYN/ACK, a final ACK is sent
- **UDP flood :** UDP flood is a type of denial-of-service (DoS) attack designed to render a system, server, bandwidth, or machine unavailable for legitimate users and requests. A session less protocol, UDP floods are highly effective and require few resources to execute.
- **ICMP flood :** A ping flood is a denial-of-service attack in which the attacker attempts to overwhelm a targeted device with ICMP echo-request packets, causing the target to become inaccessible to normal traffic. When the attack traffic comes from multiple devices, the attack becomes a DDoS or distributed denial-of-service attack.

DDOS Attack Detection Using Entropy

This entropy detection method is mainly used to calculate the distribution randomness of some attributes in the network packets' headers. These attributes could be the packet's source IP address, TTL value, or some other values indicating the packet's properties.The higher the randomness the higher is the entropy and vice versa. So, whenever the entropy is less than a threshold value we can say that a DDos attack is occurred.

Entropy Detection :

$$p_i = \frac{x_i}{\sum_{i=1}^{n} x_i}$$

x = Defines destination IP address ; n = No. of packets in window ;

p = Probability of each element in the window

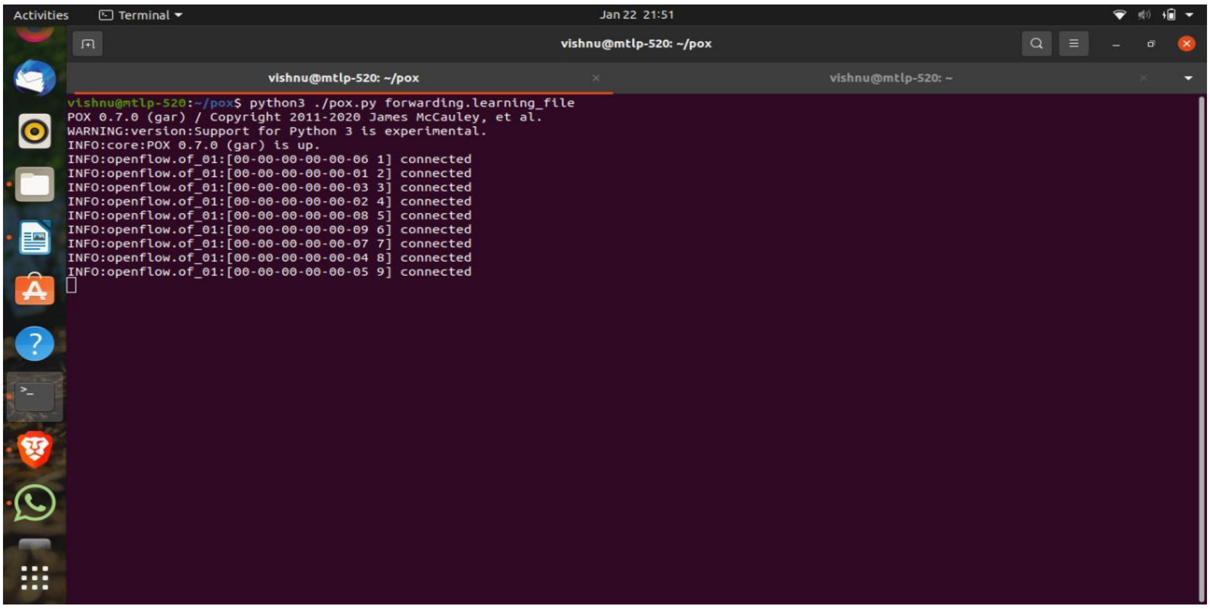
$$H1 = -\sum_{i=1}^{n} p_i log p_i$$

H1 = Information entropy od destination IP address Packets ;

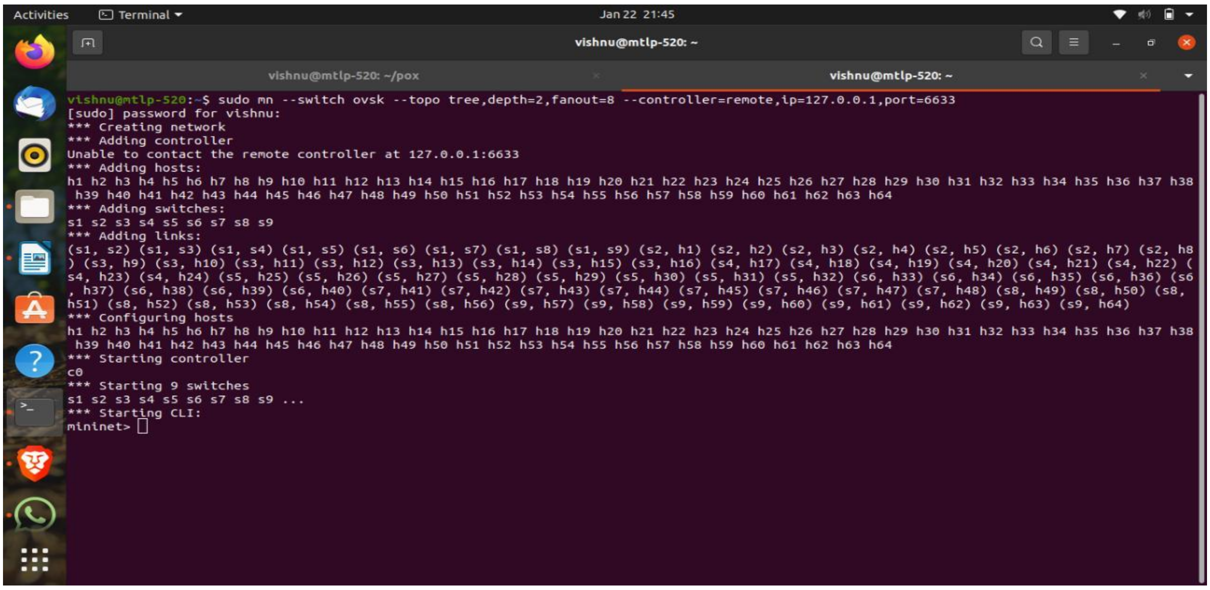
P = Probability of each element in the Window ; n = No. of Packets in window

POX Controller : The Pox controller is used in this experiment. It is widely used in experiments with highspeed and light weight. It is designed as a platform, so a user-defined controller can be built on it. Pox provides OpenFlow interface for topology and path selection. The kernel is the assembly point for all components, and components can interact with each other through the kernel. Pox provides Openflow interface for topology discovery and path selection and can customize components to realize specific functions. Pox controller supports rapid development of controller prototype functions and it can produce superior performance applications, which decreases the burden of developers and improves development efficiency

Connecting to POX Controller :

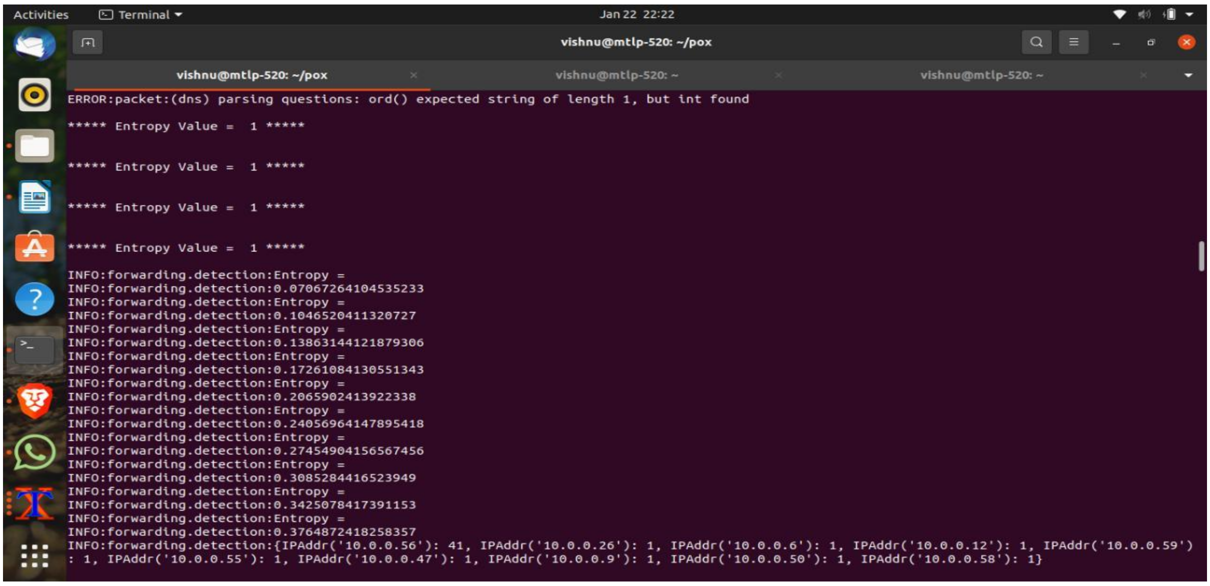


Creating a Topology :



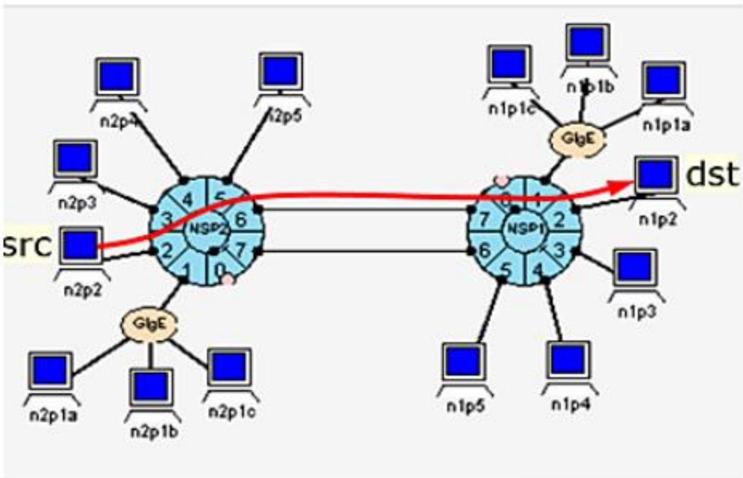
Anamoly Detection Using Entropy :

Entropy = 1 Normal Traffic , Cha nge in Entropy = DDOS Attack



Generating Normal Traffic

IPERF is a traffic generation tool that allows the user to experiment with different TCP and UDP parameters to see how they affect network performance. The typical way that IPERF is used is to first start one IPERF process running in server mode as the traffic receiver, and then start another IPERF process running in client mode on another host as the traffic sender.



The typical way that IPERF is used is to first start one IPERF process running in server mode as the traffic receiver, and then start another IPERF process running in client mode on another host as the traffic sender. In order to send a single UDP stream from n2p2 to n1p2 as shown in Fig we would run IPERF in server mode on n1p2 and IPERF in client mode on n2p2.

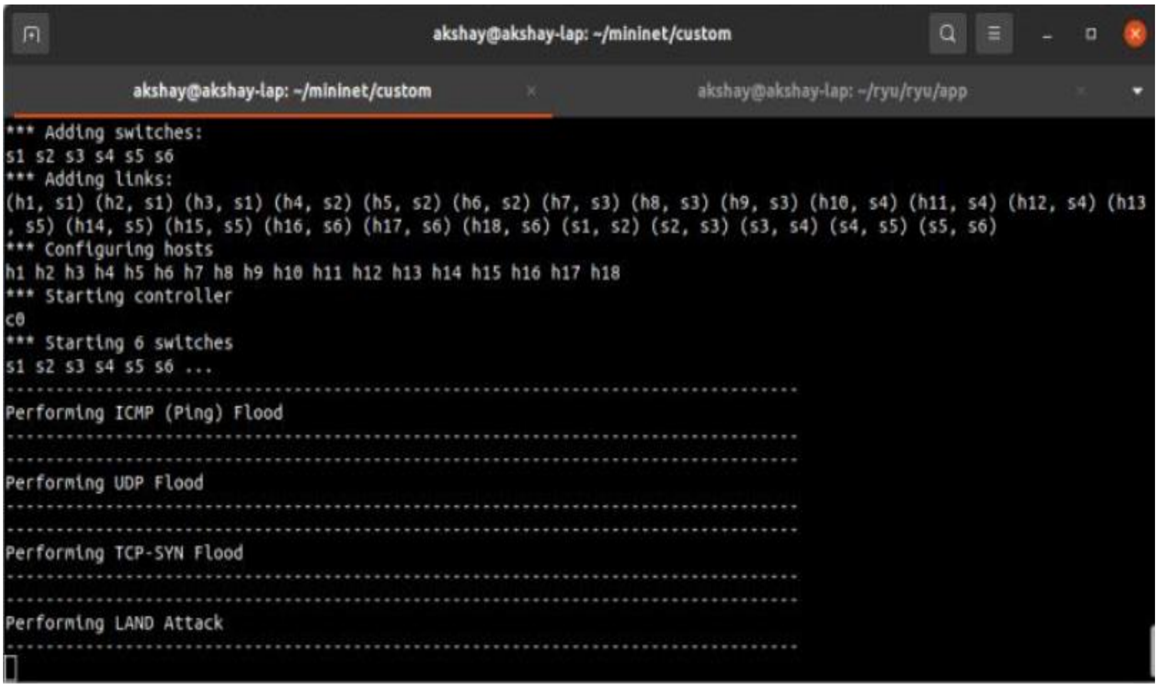
```
src.cmd("iperf -p 5050 -c 10.0.0.1")
src.cmd("iperf -p 5051 -u -c 10.0.0.1")
```

```
akshay@akshay-lap: ~/mininet/custom
akshay@akshay-lap: ~/mininet/custom        akshay@akshay-lap: ~/ryu/ryu/app
--------------------------------------------------------------------------------
Generating traffic ...
--------------------------------------------------------------------------------
Iteration n 1 ...
--------------------------------------------------------------------------------
generating ICMP traffic between h12 and h16 and TCP/UDP traffic between h12 and h1
h12 Downloading index.html from h1
h12 Downloading test.zip from h1
generating ICMP traffic between h10 and h5 and TCP/UDP traffic between h10 and h1
h10 Downloading index.html from h1
h10 Downloading test.zip from h1
generating ICMP traffic between h18 and h8 and TCP/UDP traffic between h18 and h1
h18 Downloading index.html from h1
h18 Downloading test.zip from h1
generating ICMP traffic between h12 and h11 and TCP/UDP traffic between h12 and h1
h12 Downloading index.html from h1
h12 Downloading test.zip from h1
generating ICMP traffic between h8 and h8 and TCP/UDP traffic between h8 and h1
h8 Downloading index.html from h1
h8 Downloading test.zip from h1
generating ICMP traffic between h12 and h4 and TCP/UDP traffic between h12 and h1
h12 Downloading index.html from h1
h12 Downloading test.zip from h1
```

Generating DDOS Traffic

A Distributed Denial of Service attack (DDOS) is similar to a DOS attack but carried out from different nodes (or different attackers) simultaneously. Commonly DDOS attacks are carried out by botnets. Botnets are automated scripts or programs which infect computers to carry out an automated task (in this case a DDOS attack). A hacker can create a botnet and infect many computers from which botnets will launch DOS attacks, the fact many botnets are shooting simultaneously turn the DOS attack into a DDOS attack (that's why it is called "distributed"). The tool hping3 allows you to send manipulated packets. This tool allows you to control the size, quantity and fragmentation of packets in order to overload the target and bypass or attack firewalls. Hping3 can be useful for security or capability testing purposes, using it you can test firewalls effectivity and if a server can handle a big amount of packets .

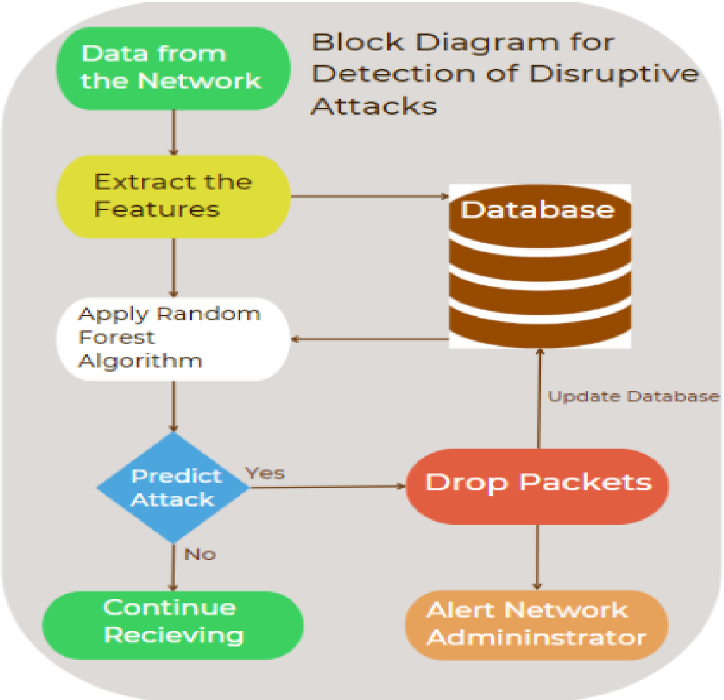
```
print("--------------------------------------------------------------------------------")
print("Performing ICMP (Ping) Flood")
print("--------------------------------------------------------------------------------")
src.cmd("timeout 20s hping3 -1 -V -d 120 -w 64 -p 80 --rand-source --flood {}".format(dst)
sleep(100)

src = choice(hosts)
dst = ip_generator()
print("--------------------------------------------------------------------------------")
print("Performing UDP Flood")
print("--------------------------------------------------------------------------------")
src.cmd("timeout 20s hping3 -2 -V -d 120 -w 64 --rand-source --flood {}".format(dst))
sleep(100)
```



DDOS Attack Detection Using ML Algorithms

Depending on the source of the data to be studied or the method used to identify anomalous occurrences, there are several categories that might be used to classify intrusion detection. Flow-based or packet-based detection is employed depending on the data to be processed, and depending on the detection method, signature-based or anomaly-based detection can be utilized.Flow-based detection was selected based on the source of the data to be evaluated since it would be more suited for high-speed networks and more effective than packet-based detection in terms of processing and memory overhead. Anomaly-based intrusion detection is implemented as a detection method, and more specifically, detection with ML Algorithms (KNN, Naïve Bayes, Logistic Regression, Decision Tree, Random Forest)

Process Flow



Mitigating DDOS Traffic : Once a malicious flow has been verified by the anomaly detection module, the Anomaly Mitigation module is in charge of implementing mitigation measures to prevent network disruption or performance degradation. When label is 1 It will block the traffic generated port for some time and after mitigation done it will unblock the port.

Commands

Generating Normal Traffic:

run command ryu-manager start_traffic_collection.py in ryu/ryu/app/

run command sudo python2 generate_normal_traffic.py in mininet/custom/

Generating DDos Traffic:

run command ryu-manager collect_ddos_traffic.py in ryu/ryu/app/

run command sudo python2 generate_ddos_traffic.py in mininet/custom/

Detecting DDos and Normal Traffic:

run command ryu-manager mitigation_module.py in ryu/ryu/app/

run command sudo python2 topology.py in mininet/custom/

after the above commands, run command xterm h1 h2 h3 h6 h17 in mininet

Normal Traffic:

ping 10.0.0.15 on node h17 and then ping 10.0.0.4 on h6

DDoS Traffic:

on node h17 ping 10.0.0.18

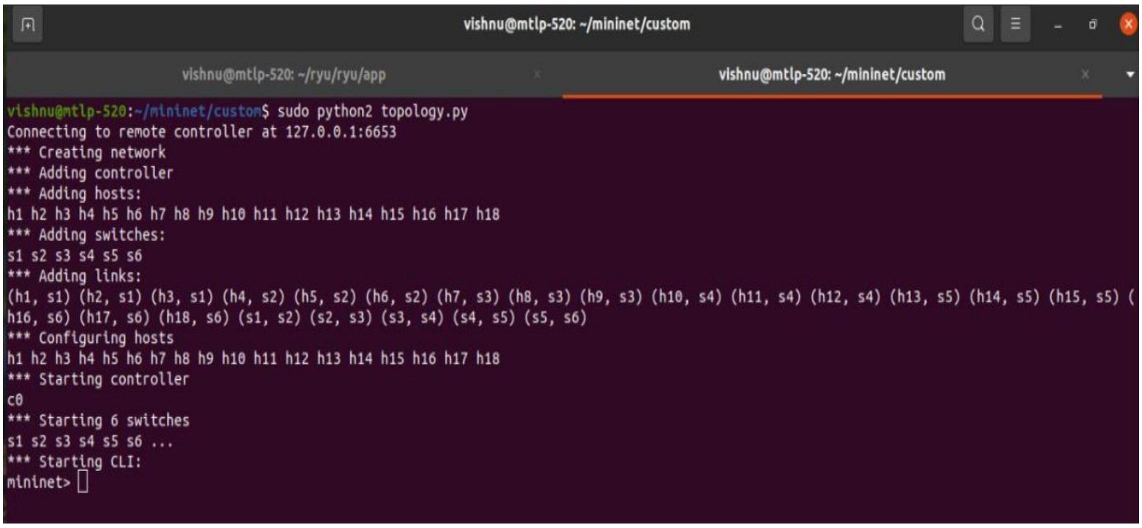
on node h1 hping3 10.0.0.12 --icmp --rand-source --flood

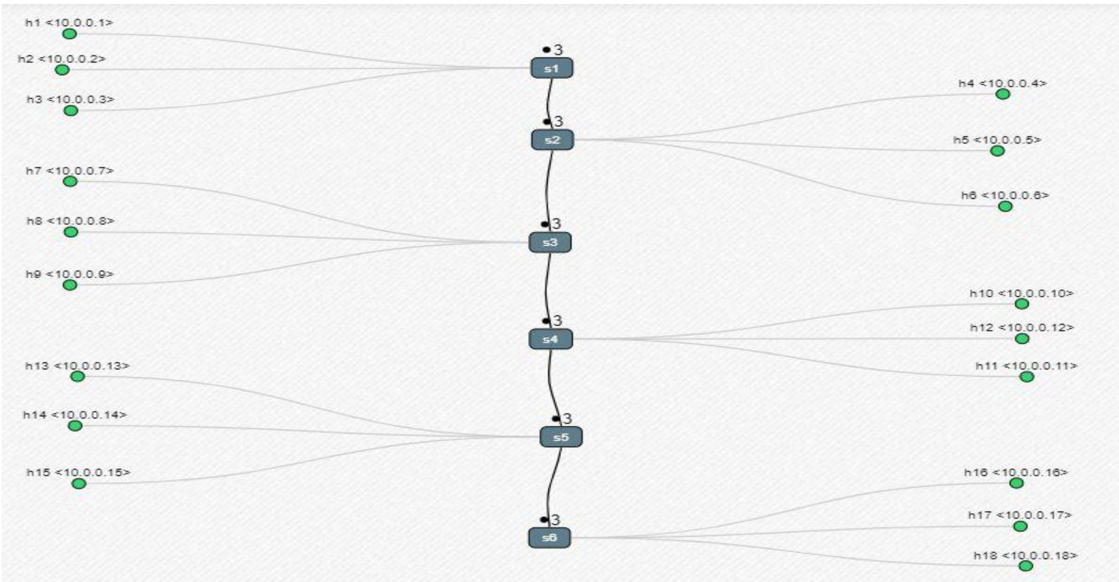
on node h2 hping3 10.0.0.12 --udp --rand-source --flood

on node h3 hping3 10.0.0.12 --syn --rand-source --flood

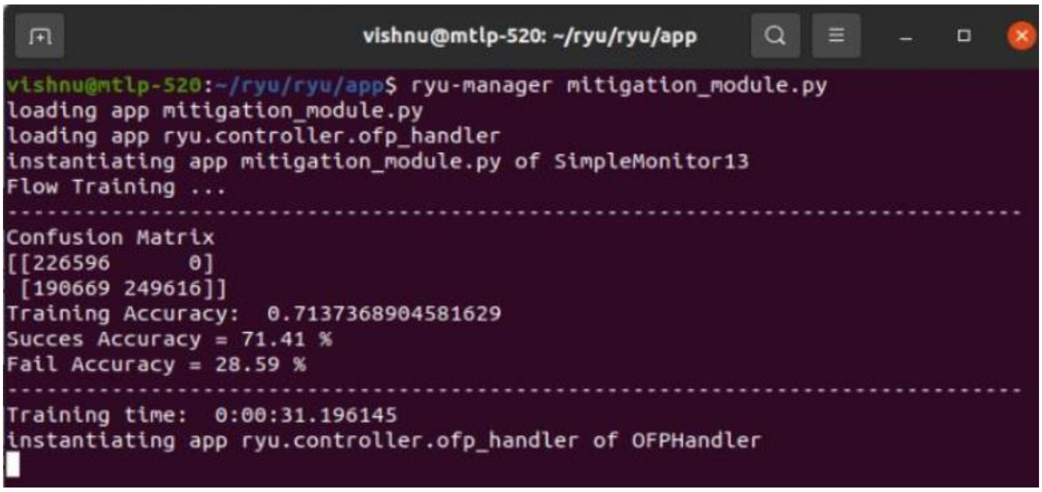
Output & results

TOPOLOGY GENERATION

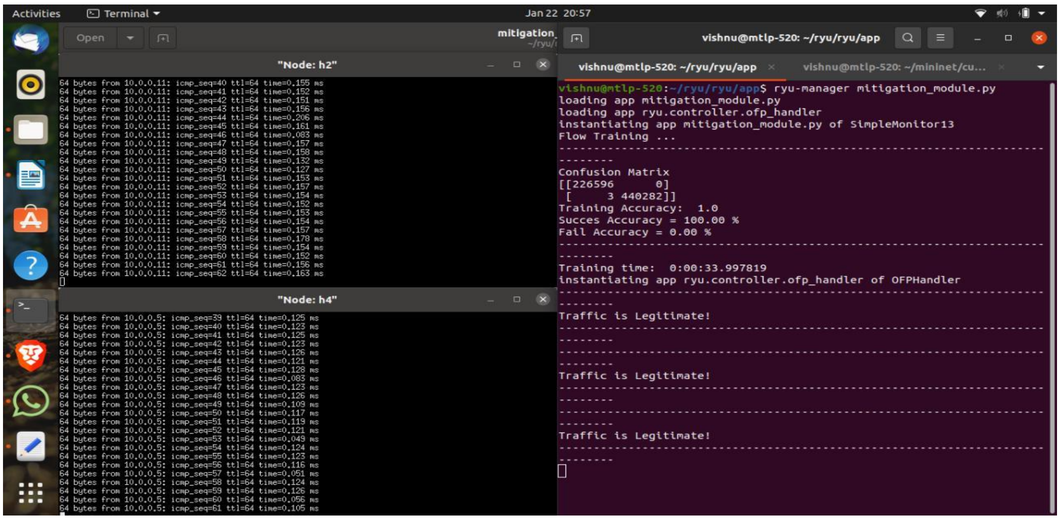




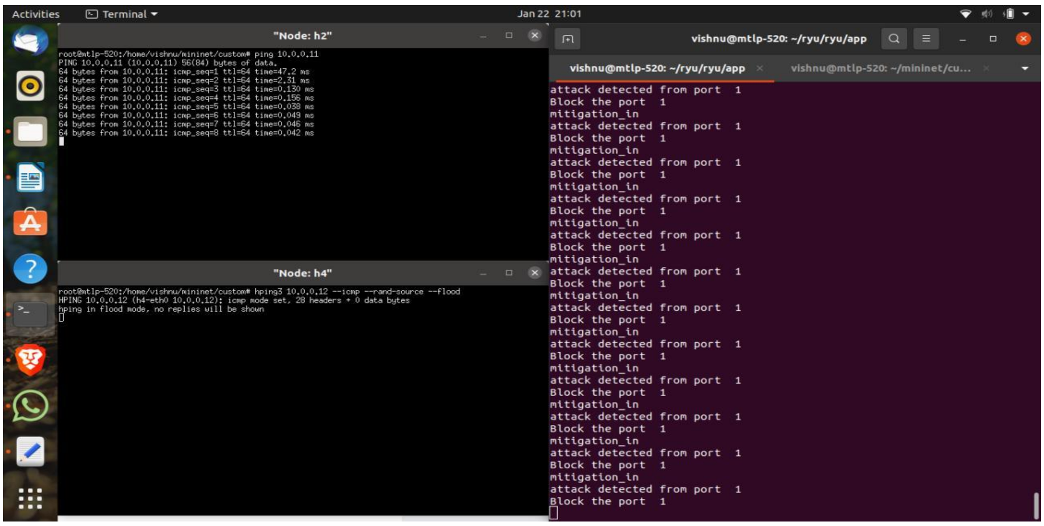
LOADING CONTROLLER



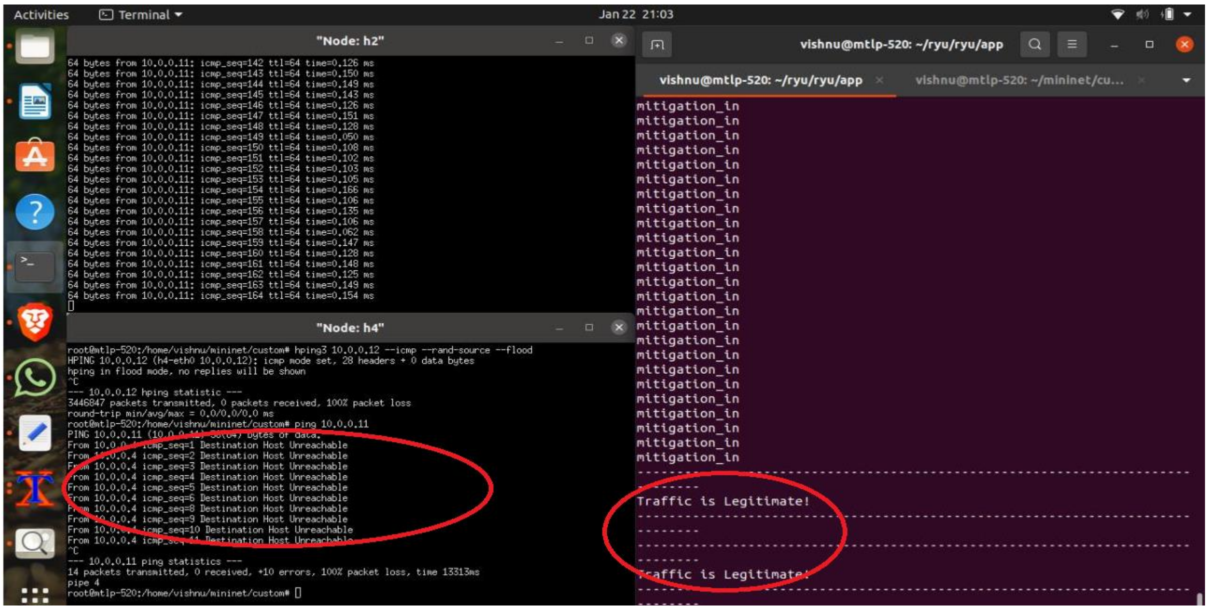
WHEN TRAFFIC IS NORMAL



WHEN DDOS TRAFFIC IS DETECTED



AFTER MITIGATION

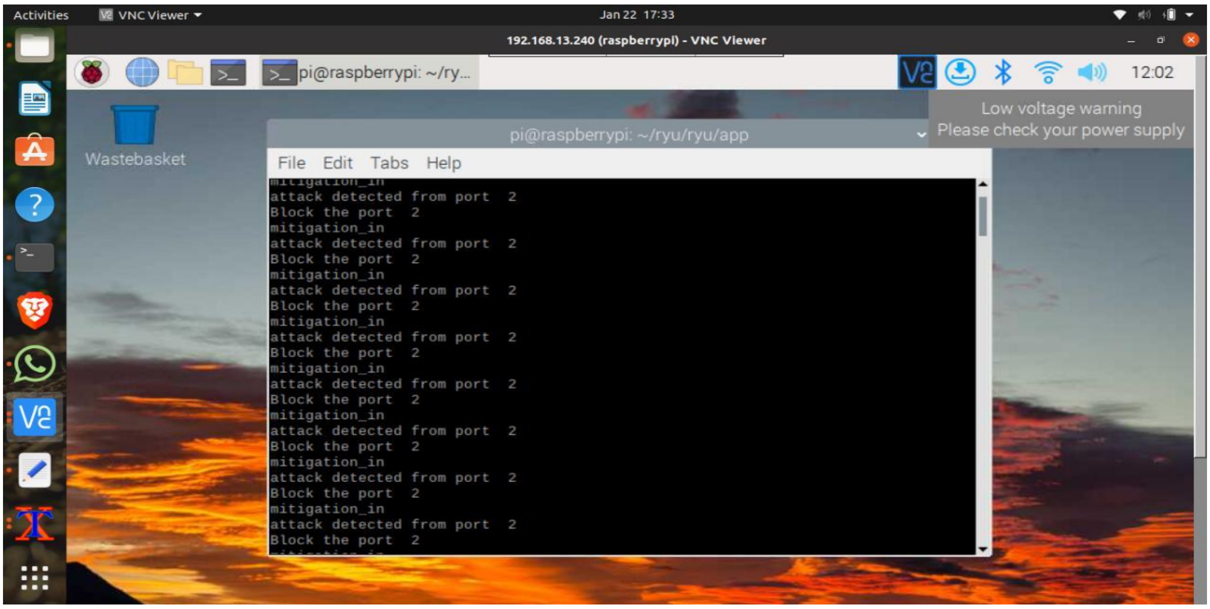


IMPLEMENTATION IN RASPBERRYPI

Connect the Raspberry Pi to your Computer.

Change the IP Address in topology file to RaspberryPi's IP Address.

Run Ryu Controller in RaspberryPi environment.



RESULTS

K Nearest Neighbors = 100%

Logistic Regression = 66.6%

Naïve Bayes = 71.41%

Decision Tree = 100%

Random Forest = 100 %

Future Works

It may fail to detect the attack traffic in a multi-controller environment. So, in the future, these models are also evaluated to detect attacks in a multi-controller context. The presented model acquired findings from a single dataset, which serves as a limitation of the model. Consequently, a distributed dataset can be examined in order to provide directions for future enhancements. Memory and other limited resources and computing abilities, as well as a diversity of standards and protocols, characterize the Internet of Things. These variables add significantly to the difficulties in researching IoT security issues, including anomaly mitigation utilizing IDS. In spite of the extensive study on anomaly detection in IoT networks, there are numerous key outstanding challenges that require additional investigation. The following are a few of these issues:.. There are no publicly available IoT network traffic datasets. Because assessing and validating anomaly prevention strategies on a real network will be difficult, efforts to create an IoT dataset are essential. This will make evaluating and validating suggested anomaly mitigation techniques in the IoT much easier. There are not any standard authentication apps for IoT. The validation of implemented structures is critical since it guarantees that they are developed acceptably. The implemented structures are put to the test in a variety of ways, including simulations and tests. However, because of a lack of standard authentication applications, most of implemented IDS structures in the IoT are not evaluated in contrast to other IDS structures in the IoT. As a result, efforts must be made to produce standard authentication, which will assure duplication, reproducibility, and research continuity. RNN and CNN are examples of supervised and unsupervised ML techniques, and both can be discovered using the CICDDoS2019 dataset.. It is possible to gather and examine real-time packets against the classified training dataset. It is possible to use a technique for splitting the data and comparing it with the performance of the classifiers utilized fold cross authentication

Conclusion

In this work, 5 machine learning algorithms are used to develop a model that can automatically identify and mitigate DDoS assaults in SDN networks. All of the traffic flow entries are regularly collected by the model, which then extracts the native flow features and expands them by including additional features. A detection module uses five criteria to categorize each flow as normal or anomalous. When an attack is discovered, its source is prevented. Three ML algorithms were assessed with regard to the classification ML method utilized in the detection module, including random forest, Decision Tree, Logistic Regression, Naïve Bayes, and KNN. The outcomes of the experiment demonstrated that Random Forest is the best classifier for the generated network. Without disrupting regular traffic, the implemented methodology proved effective at swiftly and correctly identifying and thwarting threats.