International Journal of Scientific Engineering and Applied Science (IJSEAS) – Volume-8, Issue-9, September 2022
ISSN: 2395-3470
www.ijseas.com

Machine Learning to Detect and Mitigate DDoS Attacks on SDN

D G Tejas , T N R Kumar

Department of Computer Science, M S Ramaiah Institute of Technology,India

e-mail: tejasdg966@gmail.com,tnrkumar@msrit.edu

Abstract

Software defined networking (SDN) has many advantages, including flexibility, monitoring, and innovation. However, SDNs are vulnerable to many security threats. One of the main types of attacks that disrupt SDN networks is Distributed Denial of Service (DDoS) attacks. There are multiple ways of forestalling DDoS assaults on SDN networks.Machine learning strategies such as the Naive Bayes, Support Vector Machines (SVMs), and MLP classifier are ways to identify and prevent DDoS attacks. This process involves training the RYU controller and creating a record of normal and attack traffic. When the controller is in a detection mode, a sample of the traffic is provided as input from one of the hosts, and the controller calls the machine learning algorithm to determine the type of traffic. With attack traffic, blocking the host MAC address reduces the attack. The outcomes showed that MLP classifier performs better than the other evaluated algorithms.

Keywords - Distributed Denial of Service (DDoS), Machine learning, MLP classifier, Naive Bayes, Support Vector Machines (SVMs), Software defined networking (SDN)

1. Introduction

Software defined networking (SDN) is very accepted today because of its scalability, flexibility, and monitoring benefits[1]. The main difference between traditional networks and SDNs is that traditional network devices are a combination of control planes and data planes, whereas SDNs have a separate control plane from the data planes[2]. The data plane contains network elements such as switches and routers supervised by the controller in the control plane[3]. Also the controller handles the configuration and management , which simplifies network management[1][2]. Overseers don't have to access and reset huge number of gadgets on the network to perform network overhauls and fixes.Easily integrate real-time policy applications and network applications from the controller[4]. Fig 1 shows the distinction between a conventional network and SDN.

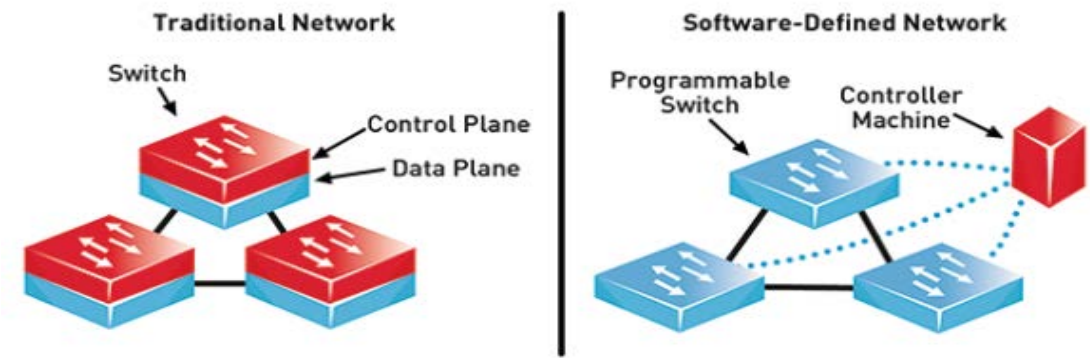

Fig 1: Difference between traditional network and Software Defined Network

The controller requires certain essential services to operate the data plane. It can exchange data with application layer services to provide network functions such as routing , load balancing, and access detection[1]. Every one of the administrations and applications utilized in the application layer are planned across the network with the working framework introduced on the controller to give the most elevated level of network control, computerization, and proficiency[1]. Applications utilize the application programming point of interaction (APIs), which incorporate Java API for nearby correspondence with the controller, or the Representational State Transfer (REST) API for far off correspondence with the controller[3]. Hence, the design of the SDN is displayed in Fig 2.

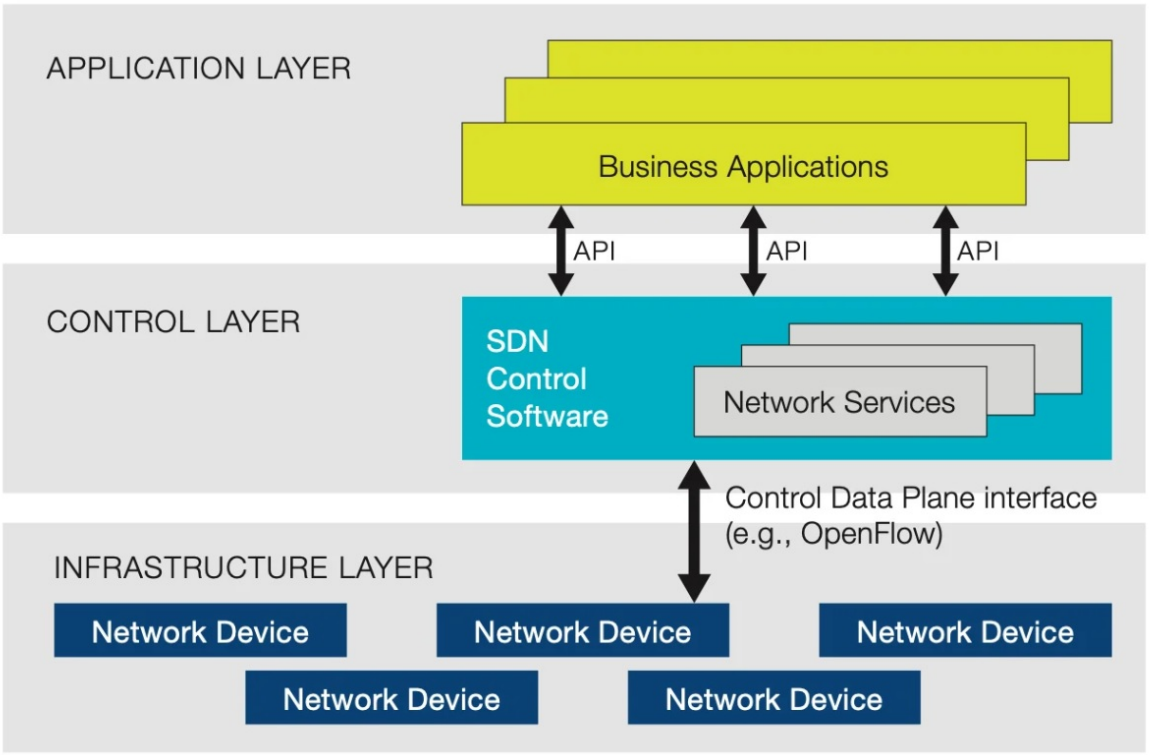

Fig 2: SDN Architecture

In any case, DDoS assaults devastatingly affect SDN networks. On the off chance that the network isn't safely secured, a DDoS assault could surpass the control data transmission or OpenFlow (OF) switch. There are many records to safeguard SDN networks from DDoS assaults. One such innovation that draws in specialists is centered around utilizing a machine learning to recognize DDoS assaults. Be that as it may, shielding SDN networks from dangers stays a reasonable area of examination. This article centers around such a technique pointed toward distinguishing DDoS assaults on live networks and deciding the most fitting machine learning algorithm for relief.

2. DDoS CLASSIFICATIONS AND FEATURES

2.1 DDoS Classification

As referenced before, DDoS assaults are pointed toward flooding an organization with an enormous number of parcels in various areas. DDoS assaults can flood the casualty's organization in numerous ways, including TCP, UDP, ICMP flood assaults, arbitrary IP flood assaults, and botnet utilization.

2.1.1 TCP Flood

The most well-known DDoS assaults are TCP flood assaults. TCP flood assaults send countless TCP association solicitations to the casualty without actually looking at the SYN-ACK on the casualty's server. Many somewhat open connections dwell on the casualty's server. This halfway association consumes all or a large portion of the assets and makes them inaccessible to genuine clients[6].

2.1.2 ICMP Flood

One more kind of DDoS is an ICMP flood assault, otherwise called a smurf assault. Fill the casualty with an enormous number of ICMP parcels utilizing the satirize IP address. The casualty's server answers with an ICMP reaction to a misleading IP address holder. This influences the presentation and accessibility of both the casualty's server and the genuine proprietor of the phony IP address[6].

2.1.3 UDP Flood

The third sort of DDoS assault is UDP flooding. They filled the casualty with heaps of UDP parcels. One such model is the intrusion of DNS intensification. In this assault, the assailant deceived the casualty's IP address and sent a little question to the DNS server. The DNS server answers with a huge reaction that debases the casualty's presentation. UDP flooding can likewise be brought about by flooding the casualty with countless UDP bundles to keep it from happening to typical clients[6].

2.1.4 Random IP Flood

DDoS assaults can likewise be begun by producing irregular IP bundles, making the controller occupied with answering messy parcels and incapable to answer other genuine traffic[4]. Viable DDoS assaults can consume most of the day to arrive at a high level of malevolent bundles and can happen at specific times[2].

2.1.5 Botnets

A mind boggling and dangerous strategy for DDoS assault is a botnet. Botnets are a large group of imperiled PC’s[5]. Some simple to-utilize assault creating instruments are accessible for nothing or for minimal price. Anybody can undoubtedly track down assets or recruit others to do a wide range of online assaults. Botnets are worked by introducing malignant programming on your PC utilizing dubious strategies. This can be accomplished by phishing tricks, spam messages, site connects, or downloads shipped off unforeseen clients. A malware program utilizes a contaminated PC to interface with the botnet proprietor's order and control (O&C). The O&C server then, at that point, utilizes distributed correspondence and cooperation to send guidelines to every contaminated PC (around thousands) to harm the casualty's organization/server.

3. Problem with SDN

The absolute most normal yet frightful assaults on SDNs are DDoS attacks.Such assaults influence the presentation and conduct of the network. By closing down applications, they are handicapping or debasing network services, and authentic clients can't speak with the SDN controller or send parcels over the network[9]. DDoS assaults are accomplished on SDN's by making a few new streams that flood the controller bandwidth, OpenFlow switches, and SDN controls, prompting network disappointments for legitimate hosts. Obviously, the assailants are creating a few new streams that have harmed IP addresses however were sent over different

sources (DDoS). These casualty addresses don't match any of the principles that as of now exist in the OpenFlow switch flow table, bringing about a table disappointment. Such a circumstance prompts the creation of huge parcel messages shipped off the SDN controller from the OpenFlow switch, which consumes network bandwidth, memory, and CPU in both the control and flight of the SDN information[10]. Moreover, as OpenFlow switch cradles parcel in messages prior to sending it to the controller, in the event that few new streams are recognized in an exceptionally brief time frame, the capacity is full. This outcomes in sending all the new stream parcels to the controller as opposed to sending header-only package header messages only just, bringing about higher utility bandwidth control and postpones in the establishment of new flow rules found in the SDN controller. Another element that can bring about a huge new stream is filling the OpenFlow switch forwarding table. As referenced before, such a table incorporates an assortment of flow rules that oversee the change in regards to parcel move, and is checked on and overseen by the controller[11]. Having a few new flows brings about the presentation of new flow rules in the flow table. Some of the time, the flow table tops off, thus, when it gets another flow rule from the controller, it can't introduce it - so it disposes of the bundle and sends a error message to the controller[12]. Moreover, the switch can not move parcels until there is free memory in its sending table, prompting deferrals and scaling down approaching parcels[13].

On the controller side, the elevated degree of appearance of interior package messages that surpasses the handling force of the controller brings about disappointment of the controller and makes it blocked off to true traffic. This might bring about the disappointment of the whole network, as the controller utilizes SDN intelligence and oversees OpenFlow applications and switches[13]. Fig 3 shows an unmistakable perspective on DDoS assaults on SDN's.

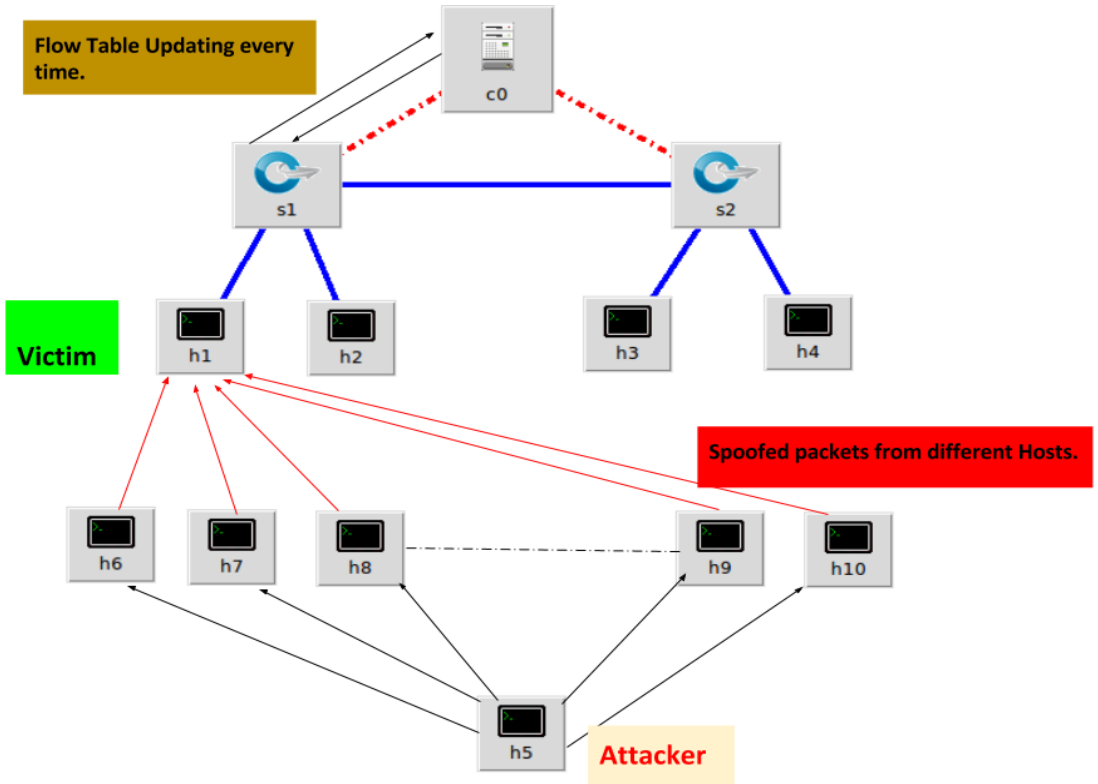

Fig 3: DDoS attack on SDN

4. SDN DDOS DETECTION AND MITIGATION MODEL

The experiment is carried out on Ubuntu (20.4) setting up a virtual machine in VMware with 4GB of RAM and 40GB of hard space. Mininet (2.3) is utilized to make SDN networks utilizing the RYU controller (4.3). Four hosts (h1, h2, h3, and h4) are associated with switch 2 (S2), and servers 1 and 2 (h5 and h6) are associated with switch 3 (S3). S2 and S3 are associated with switch 1 (S1). S1 is associated with the RYU controller. The topology is displayed in Fig 4.

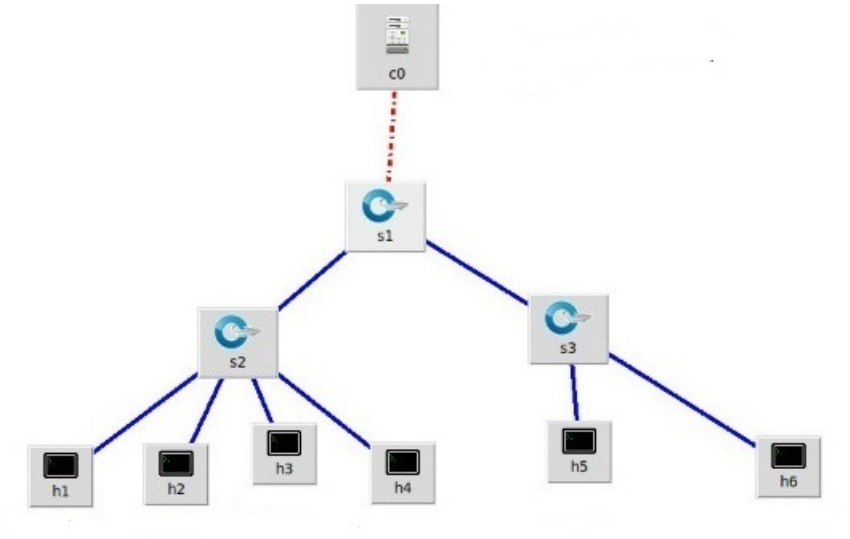

Fig 4: Topology of the network

The proposed system utilizes four contents:

- Customary Traffic Scripts - Randomize normal HTTP and ICMP parcels to all hosts and servers.
- DDoS Traffic Script - Floods ICMP and TCP parcels to the web server at a pace of 100 bundles each subsequent utilizing parodied source IP tends to haphazardly produced utilizing hping3.
- Detection Script-Uses the chosen MLP classifier to group OF switch approaching bundles into DDoS parcels and customary parcels.
- Mitigation Script - Use REST messages to add a stream section through the controller and block the DDoS aggressor's port on the OF switch.

4.1 Training and Testing Dataset

When the controller is in acquisition mode, the host generates Customary traffic scripts and DDoS traffic scripts to create the training dataset.The hping3 program has been utilized in Python contents to deliver standard DDoS bundles (ICMP and TCP floods). To keep away from disarray while making informational indexes, DDoS and

typical traffic were taken independently DDoS traffic loaded up with 80 parcels each second and went on for 15 minutes. After the DDoS examine was finished, ordinary traffic was utilized and taken for 15 minutes to gauge DDoS worth and typical traffic.

Caught information is put away in CSV arrangement to remove important highlights while barring superfluous information like Address Resolution Protocol (ARP). This paper centers around finding and lessening DDoS on SDN networks utilizing machine learning. Hence, you want to utilize a component that can be effortlessly taken out without over-burdening the network.The accuracy score of Naive Bayes, Support Vector Machines (SVMs), and MLP classifier is obtained from the training dataset and shown in TABLE I. To meet our objective, we zeroed in on information highlights depicted in [8], since they are quicker and easier to produce.

Machine Learning Algorithm	Accuracy(%)
Support Vector Machines (SVMs)	96.65
Guassian Naive Bayes	99.81
MLP classifier	100

TABLE I. Comparison between Naïve Bayes, SVM and MLP classifier

From our analytical comparison of Guassian Naive Bayes, SVM and MLP classifier, as shown in TABLE I, it was concluded that MLP classifier is best suited for our scenario due to its high accuracy. More details about the performance measures can be found in [7]. We saved the MLP classifier to be used for classification of online DDoS and normal network traffic in our SDN network.

4.2 Experimental Results

The controller runs in two modes. Acquisition mode and detection mode. When the controller is in acquisition mode, the host generates Customary traffic scripts and DDoS traffic scripts to create the training dataset. When the controller is in detection mode, the host provides a pattern of incoming traffic by calling a machine learning algorithm that uses the training dataset to detect the type of traffic.For normal traffic the controller's output prediction logic is 0,returning to flow monitoring..For attack traffic, the controller's output prediction logic is 1. This is a DDoS attack and means that the MAC address of the host that caused the DDoS attack is blocked and the controller returns to Monitor the flows.

5. Conclusion

In this paper, we have fostered a SDN structure that distinguishes and safeguards controls and switches of DDoS assaults. This structure incorporates preparing machine learning models with information gathered to anticipate DDoS assaults. The forecasts are then utilized in relief archives to go with choices on the SDN organization. We utilized information gathered to test Naïve Bayes, SVM, and MLP classifier. Our experimental outcomes show that MLP classifier is the most suitable for our network.In future work, we can continue to reduce the time to detect DDoS attacks by using highly efficient machine learning tools to reduce the number of packet separation steps.

References

[1] "Toward an Optimal Solution Against Denial of Service Attacks in Software Defined Networks - ScienceDirect." [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167739X18302930#bb40. [Accessed: 30-Mar-2019].

[2] N. I. G. Dharma, M. F. Muthohar, J. D. A. Prayuda, K. Priagung, and D. Choi, "Time-Based DDoS Detection and Mitigation for SDN Controller," in 2015 17th Asia-Pacific Network Operations and Management Symposium, 2015, pp. 550–553.

[3] C. Lin, C. Li, and K. Wang, "Setting Malicious Flow Entries Against SDN Operations: Attacks and Countermeasures," in 2018 IEEE Conference on Dependable and Secure Computing, 2018, pp. 1–8.

[4] C. Tselios, I. Politis, and S. Kotsopoulos, "Enhancing SDN Security for IoT-Related Deployments through Blockchain," in 2017 IEEE Conference on Network Function Virtualization and Software Defined Networks, 2017, pp. 303–308.

[5] J. Smith-perrone and J. Sims, "Securing Cloud, SDN and Large Data Network Environments from Emerging DDoS Attacks," in 2017 7th International Conference on Cloud Computing, Data Science Engineering- Confluence, 2017, pp. 466–469.

[6] B. Zhang, T. Zhang, and Z. Yu, "DDoS Detection and Prevention Based on Artificial Intelligence Techniques," in 2017 3rd IEEE International Conference on Computer and Communications, 2017, pp. 1276–1280.

[7] S. Choudhury and A. Bhowal, "Comparative Analysis of Machine Learning Algorithms along with Classifiers for Network Intrusion Detection," in 2015 International Conference on Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials, 2015, pp. 89–95.

[8] M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita, "Towards Generating Real-life Datasets for Network Intrusion Detection," J Netw. Secur., vol. 17, pp. 683–701, 2015.

[9] Ombase P.M., Kulkarni N.P., Bagade S.T., Mhaisgawali A.V.Survey on DoS attack challenges in software defined networking Int. J. Comput. Appl., 173 (2) (2017), pp. 19-25

[10] G. Shang, P. Zhe, X. Bin, H. Aiqun, R. Kui, FloodDefender: Protecting data and control plane resources under SDN-aimed DoS attacks, in: IEEE INFOCOM 2017 - IEEE Conference on Computer Communications, 2017, pp. 1–9.

[11] Ahmad I., Namal S., Ylianttila M., Gurtov A.Security in software defined networks: A survey IEEE Commun. Surv. Tutor., 17 (4) (2015), pp. 2317-2346

[12] R. Kandoi, M. Antikainen, Denial-of-service attacks in OpenFlow SDN networks, in: Proceedings of the 2015 IFIP/IEEE International Symposium on Integrated Network Management, IM 2015, 2015, pp. 1322–1326.

[13] H. Wang, L. Xu, G. Gu, FloodGuard: A DoS attack prevention extension in software-defined networks, in: 45th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, 2015, pp. 239–250.