Authentication Based Security Solutions in SDN

[1] Dr. Thomas Feldman, [2] Dr. S.M. Prasad

[1] Associate Professor,Department of Computer Science and Engineering(Data Science), SSGBCOET Bhusawal,

[2] UG Student,Department of Electronics and Telecommunication Engineering, JSPM Rajarshi Shahu College Of Engineering , Tathawade,Pune,India

***Abstract*—In today's advancing world, there is an increase in the size and requirements of networks which can be a burden since moving around with switches is quite chaotic. SDN helps switches to be programmed and implemented independently. SDN is a way of providing programmability for network application development by separating the control plane from the data plane. Security of Software Defined Networking (SDN) is an open subject. Separating the control plane from the data plane opens up a number of security challenges such as a man-in-the-middle attack (MITM), a service denial (DoS), overload saturation attacks, etc. In this paper, we have focused on the overview of software defined network (SDN), it's challenges, it's issues, and their solutions. First, we have discussed about the architecture of SDN, followed by proposing three security solutions in SDN, firewall based security, entropy based security for DDoS and authentication based security. We have used POX controller of SDN.**

Keywords—SDN, firewall, DDoS, entropy, POX, authentication

I. INTRODUCTION

In SDN environments, SDN network security needs to be everywhere within a software-defined network (SDN). SDN security needs to be built into the architecture, as well as delivered as a service to protect the availability, integrity, and privacy of all connected resources and information.

Within the architecture, you need to:

1. Secure the Controller: as the centralized decision point, access to the SDN Controller needs to be tightly controlled.
2. Protect the Controller: if the SDN Controller goes down (for example, because of a DDoS attack), so goes the network, which means the availability of the SDN Controller needs to be maintained.
3. Establish Trust: protecting the communications throughout the network is critical. This means ensuring the SDN Controller, the applications loaded on it, and the devices it manages are all trusted entities that are operating as they should.
4. Create a Robust Policy Framework: what's needed is a system of checks and balances to make sure the SDN Controllers are doing what you actually want them to do.
5. Conduct Forensics and Remediation: when an incident happens, you must be able to determine what it was, recover, potentially report on it, and then protect against it in the future.

Beyond the architecture itself, how SDN security should be deployed, managed, and controlled in an SDN environment is still very much up for grabs. There are competing approaches – some believe security is best embedded within the network, others feel it is best embedded in servers, storage, and other computing devices. Regardless, the solutions need to be designed to create an environment that is more scalable, efficient, and secure. They must be:

- Simple: to deploy, manage, and maintain in the highly dynamic SDN environment.
- Cost-effective: to ensure security can be deployed everywhere.
- Secure: to protect against the latest advanced, targeted threats facing your organization.

A new category is emerging for security within next-generation environments called software-defined security, which delivers network security enforcement by separating the security control plane from the security processing and forwarding planes, similar to the way SDN abstracts the network control plane from the forwarding plane. The result is a dynamic distributed system that virtualizes the network security enforcement function, scales like virtual machines, and is managed as a single, logical system.

SDSec is an example of network functions virtualization (NFV), which offers a new way to design, deploy, and manage SDN network security by decoupling the network function, such as firewalling and intrusion detection, from proprietary hardware appliances, so they can run in software. It's designed to consolidate and deliver the networking components needed to support a fully virtualized infrastructure – including virtual servers, storage, and even other networks.

In this paper we have address the security of SDN based on firewall, DDoS and authentication. The remaining paper is organized as follows. Section II provides a review on SDN architecture. Section III contains problem statement, security issues in SDN. Section IV explains proposed methods and solutions, whereas in Section V further conclusions are drawn.

II. SDN ARCHITECTURE

SDN has emerged as a flexible, secure and well-managed network. The architecture of SDN provides a central

network control and its management via controller [1]. It segregates the data forwarding functions from the control plane of network. The control is transferred to a centralized controller to take decisions related to routing and then communicate those decisions to the data-forwarding plane [2]. Despite of all its features and functions, security of SDN is still considered to be a major concern. The configuration errors can lead to serious consequences as well as the aspects of programmability makes it vulnerable to attacks. The authentication, security and integrity of the network are severely affected. The architecture of SDN can be exploited to improve network security by providing security monitoring, analysis and response system [3]. The basic architecture has been shown in Fig. 1. SDN is cost-effective, dynamic, manageable and adaptable. Initially, it was being used for wired networks but with swift increase in the use of devices including smartphones and tablets has led to a great increase in data traffic in these devices. WLANs are used in homes, businesses and in public environments. There is a one-to-one mapping between a client and a light virtual access point with a unique and different BSSID. The client can switch control from one AP to another without any notification that connection was reestablished. There is no delay in communication or hardware as one device can move LVAP form one device or AP to another.

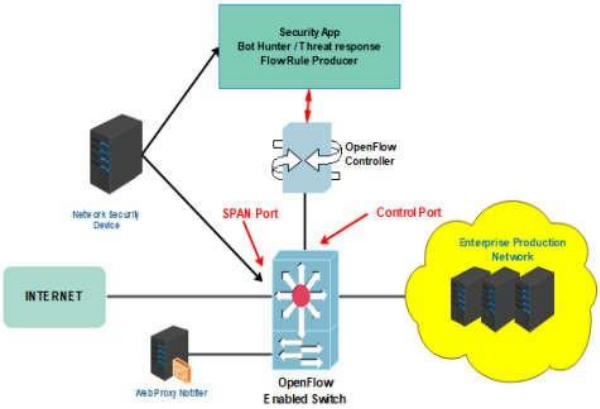


Figure 1: SDN Architecture

This paper focuses on the security issues of SDNs. It presents some specific design issues of securing SDN.

.

III. SECURITY ANALYSIS OF SDN

Software defined networks are replacing the traditional networking systems due to its centralized control approach. The privacy, integrity and confidentiality of the system may get affected due to the attacks on the system's vulnerabilities which ultimately reduce the performance and efficiency of the network [4]. This paper provides a security analysis to enforce security within SDN through attack graph and alert correlation model to lessen the false positive alerts. The security challenges to SDN include open programmable API in which the open nature of the API's makes the vulnerabilities more visible to the attackers. An unauthorized access to the central controller may cause a huge damage to the information and inject malicious codes into the system. More attacks faced by SDN consist of application layer attacks, control layer attacks and infrastructure layer attacks.

Application layer attacks include:

- Rules insertion: creating and implementing security rules for SDN in different domains lead to various conflicts.
- Malicious Code: injecting different programs lead to various attacks where attackers inject malicious code which leads to the corruption or loss of data.

Control layer includes the following attacks:

- Denial of Service Attacks: these attacks can occur at channel, controllers or between the controller and the switches.
- Attacks from Applications: the attacker who gets illegal access from the application layer gets the sensitive data about the network which leads to attacks against control layer.

Infrastructure layer attacks:

- Dos Attack: An attacker can dowse the buffer flow and the flow table by transmitting frequent large mysterious packets, which will generate new rules to be inserted into flow tables.
- Man-in-the-middle Attack: The switches and controllers are not directly connected for the transfer of information so the "man-in-the-middle" monitors can intercept important information without being detected ad can result in eavesdropping and black hole attack. Security is analyzed through attack graph and alert correlation model. Attack graph measures the ability to overcome the attacks whereas the alert correlation model classifies the alerts.

In this paper, many exterior and interior threats are being explored in the architecture of SDN. As the integrity and security of SDN is still not proven in terms of the functionality management settlement in a single central server. Cyberattacks which are launched throughout SDN have bigger destructive effects as compared to simple networks [5]. Every layer of SDN architecture has its separate requirements of security such as the configuration errors. If these requirements are not provided, they may result in various categories of security threats and attacks. The communication flooding attack linking the switch and the controller will have an effect on all the corresponding three layers. The upper three layers can be affected from the policy enforcement security attacks. Authorization attacks may result in prohibited access to the controller. For a safe SDN environment, it is necessary to make sure that every component of SDN is secured. The primary task is to ensure that the SDN controller is secured as it controls the complete management of the network. The operating system should also be secured. If the SDN controller gets compromised, it will cause the failure of the whole network. The flow model of SDN should be secured by encrypting the flows so that the injection of malicious flows is avoided. An SDN agent constitutes the environment therefore its security is very essential. The installation of identity management modification techniques and threat isolation is the main requirement. IPS, IDS, and firewalls should be dynamically updated. The communication channel must be protected between each layer. Secure coding, digital signing of the code and deployment of integrity checks are the security measures taken for this purpose.

IV. PROPOSED SOLUTION

We have proposed security solution to SDN based on three aspects

- Firewall based security
- DDoS detection based on entropy
- Authentication based protection

We have used POX controller of SDN and Mininet for simulation of the environment.

A. SDN Firewall Using POX Controller for Wireless Networks

The firewall application implemented in this paper is responsible for filtering of packets according to their parsed headers.[6] The packets are permitted or dropped based on the specified rules in the firewall. Hence the firewall application needs to run with the learning switch module of POX controller. The learning switch module enforce the OpenFlow switch to act as a type of l2_learning switch. The two modules are required to run at the same time. But running the two modules at the same time may cause following two issues:

- Running of these two modules exactly at the same time may lead to OpenFlow error messages since both of them will be accessing the same buffer.
- Also policy conflict may occur as both modules might install different rules, flow entries to the openflow switch.

To deal with these issues, there are two methods:

- Create one “Master” class which will include both firewall module and switch module. The “Master” class will be responsible for organizing the order of calling module’s functions.
- In the another method both modules are separated. The switch module is to be modified sothat it will listen to events fired by our firewall module in instead of listening to events triggered by OpenFlow protocol.

In our work, we have considered the second approach. L2_learning switch of POX controller is chosen. The learning switch is run along with the firewall application. The l2_learning switch listen to triggered events of firewall application. Thus, both module will work together in harmony.

We have evaluated the performance of proposed firewall application using different wireless network scenarios including

- varying number of access points in networks viz 2 and 4
- varying n umber of moving stations in networks from 4,8,12,16 and 20.

Table 3 shows the firewall rules containing the source and destination IP address where the traffic is blocked between them.

Table 3: Firewall Rules

Source IP Address	*Destination IP Address*
10.0.0.2	10.0.0.5
10.0.0.3	10.0.0.4
10.0.0.4	10.0.0.2
10.0.0.5	10.0.0.3

We have compared the performance in terms of delay, TCP bandwidth and jitter. Also the comparison of delay, TCP bandwidth and jitter is done for running the same topology with and without firewall.

ICMP test

Figure 4 shows the testing of ping reachability after executing the firewall application. Figure 4 shows that all traffic from sta1 having IP address 10.0.0.2 to sta4 with IP address 10.0.0.5 is blocked as if it is considered a suspicious or malicious host. Thus, sta1 cannot ping sta4 and similarly sta4 cannot ping it back. Figure 4 displays the POX output of blocking the traffic coming from the source sta1 to destination sta4 (10.0.0.5)

```
mininet-wifi> sta1 ping 10.0.0.5
'PING 10.0.0.5 (10.0.0.5) 56(84) bytes of data.
From 10.0.0.2 icmp_seq=1 Destination Host Unreachable
From 10.0.0.2 icmp_seq=2 Destination Host Unreachable
From 10.0.0.2 icmp_seq=3 Destination Host Unreachable
From 10.0.0.2 icmp_seq=4 Destination Host Unreachable
From 10.0.0.2 icmp_seq=5 Destination Host Unreachable
From 10.0.0.2 icmp_seq=6 Destination Host Unreachable
From 10.0.0.2 icmp_seq=7 Destination Host Unreachable
From 10.0.0.2 icmp_seq=8 Destination Host Unreachable
From 10.0.0.2 icmp_seq=9 Destination Host Unreachable
From 10.0.0.2 icmp_seq=10 Destination Host Unreachable
^C
--- 10.0.0.5 ping statistics ---
13 packets transmitted, 0 received, +10 errors, 100% packet loss, time 12277ms
pipe 4
```

Figure 4: Ping unreachability for firewall installed rules

Since there is no firewall rule installed between sta1(10.0.0.2) and sta2(10.0.0.3), both can communicate with each other using ping as shown in figure 5.

```
*** Unknown command: clear
mininet-wifi> sta1 ping 10.0.0.3
PING 10.0.0.3 (10.0.0.3) 56(84) bytes of data.
64 bytes from 10.0.0.3: icmp_seq=1 ttl=64 time=1.55 ms
64 bytes from 10.0.0.3: icmp_seq=2 ttl=64 time=1.15 ms
64 bytes from 10.0.0.3: icmp_seq=3 ttl=64 time=1.62 ms
64 bytes from 10.0.0.3: icmp_seq=4 ttl=64 time=1.47 ms
64 bytes from 10.0.0.3: icmp_seq=5 ttl=64 time=1.95 ms
64 bytes from 10.0.0.3: icmp_seq=6 ttl=64 time=1.54 ms
64 bytes from 10.0.0.3: icmp_seq=7 ttl=64 time=1.96 ms
64 bytes from 10.0.0.3: icmp_seq=8 ttl=64 time=1.53 ms
64 bytes from 10.0.0.3: icmp_seq=9 ttl=64 time=1.41 ms
64 bytes from 10.0.0.3: icmp_seq=10 ttl=64 time=1.47 ms
^C
--- 10.0.0.3 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9012ms
```

Figure 5. Output for hosts connectivity after running the Firewall

TCP Test

TCP/UDP layer 4 rules re added and TCP/UDP traffic is checked and matched with firewall policies for deciding whether to forward or drop the packet. We have evaluated the performance of the firewall application for TCP bandwidth. Iperf is used for establishing the connection between sta2 (10.0.0.3) and sta4 (10.0.0.5). As depicted in figure 6 iperf server is initiated at sta1.

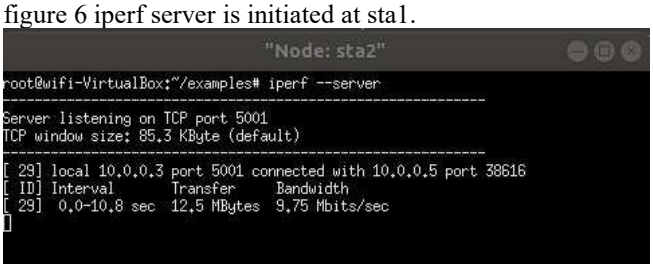


Figure 6: TCP Server initiated at sta2 using Iperf

Similarly at sta4 we initiate the iperf client to communicate with iperf server at 10.0.0.3(sta2)
As depicted in figure 7 sta4 client sends TCP packets to sta2 for 10 seconds.

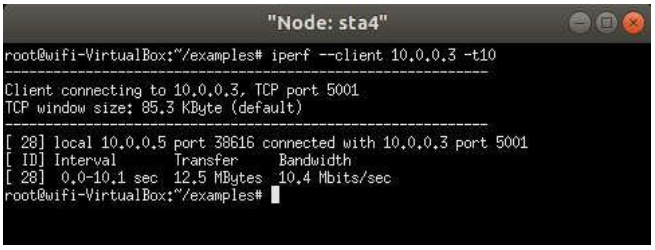


Figure 7: TCP client sending packets to server
As shown in figure 6 , TCP bandwidth utilization at server is 9.75 Mbits/sec.

RTT Evaluation

Sta2(10.0.0.3) is able to ping with sta4 (10.0.0.5) since there is no firewall rule installed. RTT Evaluation Based on layer3 policy installed, sta2 can ping sta4t4. The following figure 8 show that sta2 successfully sent ICMP messages to sta4 and got responses. The round trip time required to send 10 ICMP messages was 9015ms.

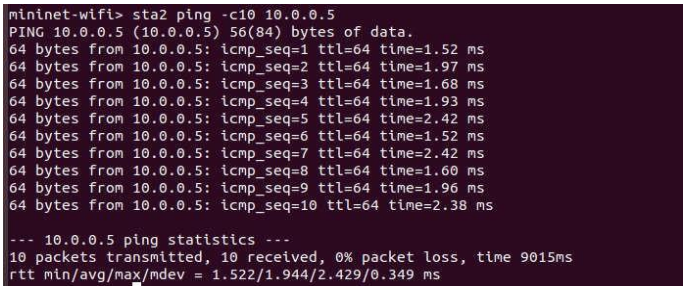


Figure 8. ICMP traffic allowed from sta2 and sta4 with firewall running

UDP Test

Iperf is used for establishing the connection between UDP server and client. As depicted in figure 9, UDP server is established at sta2 with IP address 10.0.0.3. The Iperf client sta5 sends UDP datagrams to server. Latency variation (jitter) is delay amount of time to transmit data form a source to a destination that varies over time.
Figure 9 shows measured bandwidth and Jitter of UDP traffic with firewall installed on POX controller whereas Figure 10 shows UDP client sending the UDP datagrams.

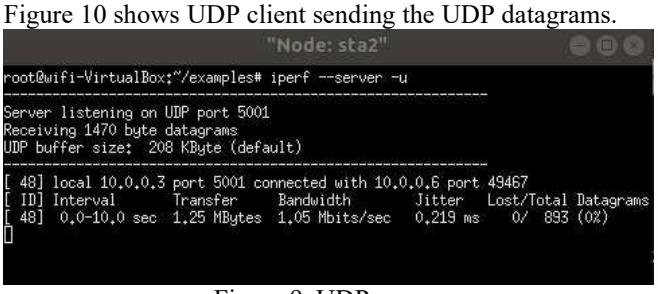


Figure 9: UDP server

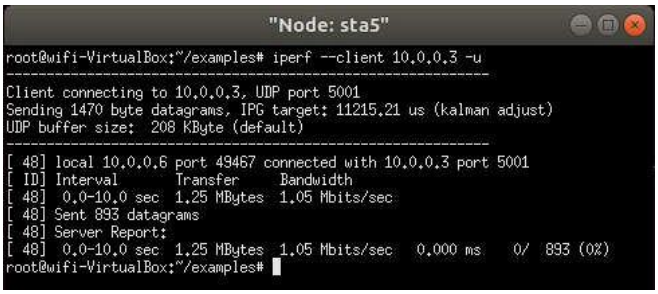


Figure 10: UDP client

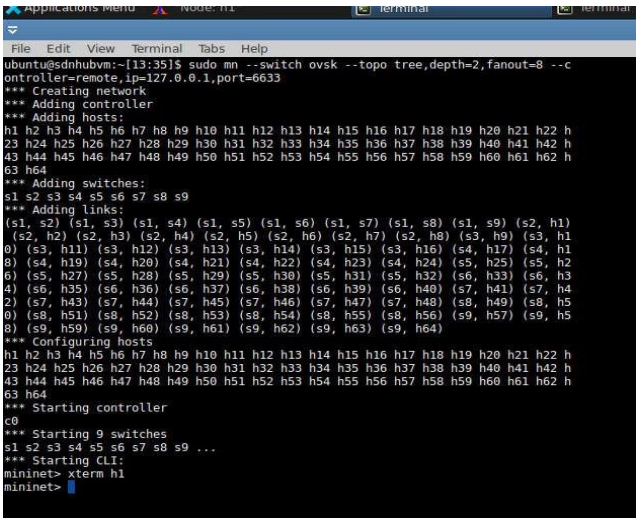
B. DDoS Detection Using Entropy

Basically, Distributed Denial of Service (DDoS) attack is a DoS attack which utilizes multiple distributed attack sources. We know that every network in the system has an entropy and increase in the randomness causes entropy to decrease[7-9].
For preventing this DDoS threat, we want to use POX for attack detection and want to provide a solution that is effective in terms of the resources used.
More precisely, this paper shows how DDoS attacks can consume controller resources and provide a solution to detect such attacks based on the entropy variation of the destination IP address. Now based on this entropy value, we shall block that specific port in the switch if it drops below certain threshold value, and then bring the port down.[10]

Steps for performing the project task:

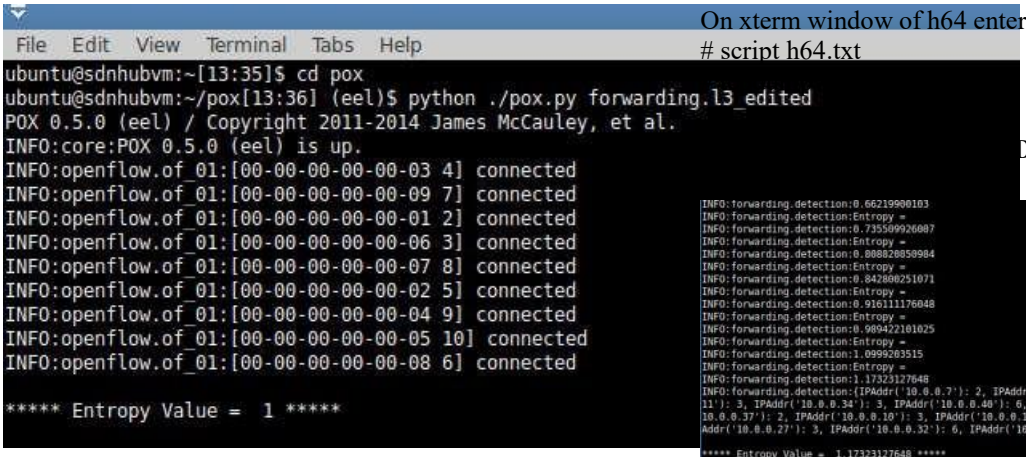
a. Creating a Mininet topology by entering the following command:

$ sudo mn --switch ovsk --topo tree,depth=2,fanout=8 --controller=remote,ip=127.0.0.1,port=6633



b. In the Mininet terminal of virtual box enter the following command for running the pox controller:

$ cd pox
$ python ./pox.py forwarding.l3_edited



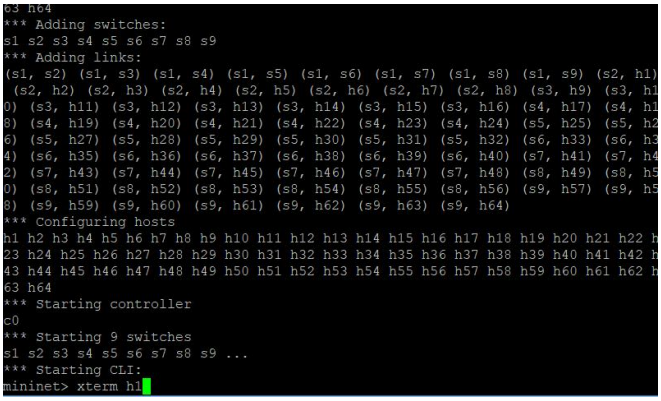
c. To determine the IP address of the POX controller, entering the following command in Mininet:

$ ifconfig

The loopback address is the IP address in the above command.

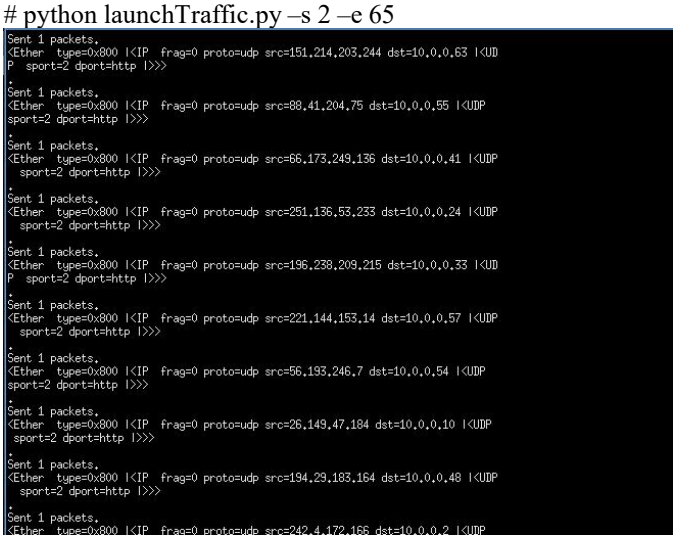
d. Now opening xterm for a host by typing the following command:

mininet>xterm h1



e. In the xterm window of host h1 running the following command:

python launchTraffic.py –s 2 –e 65

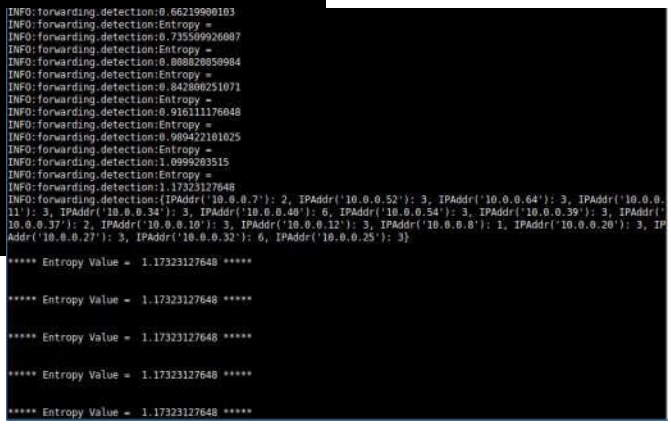


Detection of DDoS threat using the value of Entropy:

On xterm window of h64 entering the following commands:

script h64.txt

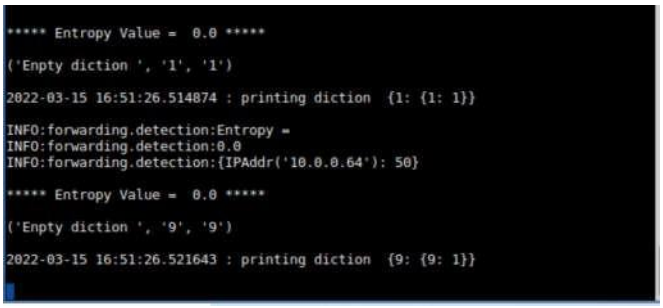
DoS attack:



Now repeating step e) on h1 and parallelly entering the following commands to run the attack traffic from h4 and h6 xterm windows to attack on 56

python launchAttack.py 10.0.0.56

Entropy value after DDoS attack:



Observing the entropy values in the POX controller. The value decreases below the threshold value (which is equal to 0.5 here) for normal traffic. Thus, we can detect the attack within the first 250 packets of malicious type of traffic attacking a host in the SDN network.

After the hosts stop sending attack packets, the switches are started again by the POX controller after shutting them down during the attack which is shown as follows:

On successful completion of the above steps, terminating tcpdump on h64 by entering ‘control/command + c’

Stop running Mininet topology by entering the following command:

mininet>exit

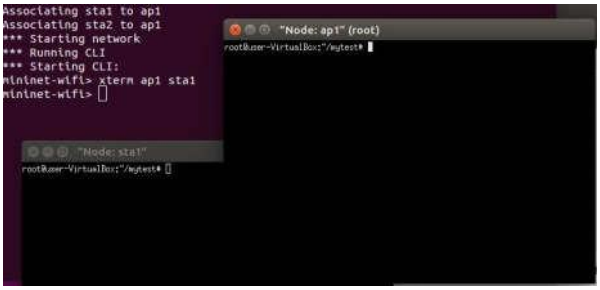
C. Authentication in SDN

We use script to create a simple wireless network environment , including an AP and two stations. In the code below , use net.addStation to add a station and set its IP address ; use net.addBaseStation to Add an AP, and set its ssid name , mode and transmission channel used[11-12].

Steps for performing the project task:

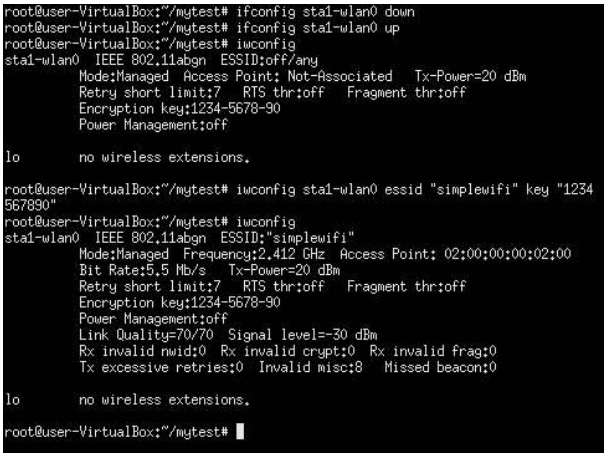
1. Execute the script

2. Use xterm to open the windows of ap1 and sta1



3. Open the packet capture program tcpdump in ap1 , and store the captured data in ap1.log .

4. After executing this script, the process of authentication and association has ended , so turn off and then turn on the wireless network card of sta1 again , and perform the actions of authentication and connection again. (Note: After the authentication and connection are executed , it will take a while to use iwconfig again to observe whether the ap is connected . If it is executed immediately , the connection state will be displayed) .



5. Use ctrl+c to interrupt the packet capture program.

V. CONCLUSION AND FUTURE SCOPE

We can conclude this paper by stating that the central controller is the most vulnerable part of the SDN architecture. Although a centralized network control system and the programmability of the network could lead to strong and viable implementation of security. Therefore, we have presented most of the security weaknesses in SDN and their proposed solutions in this paper. We have used Mininet and POX controller of SDN for simulation of network. Proposed solutions includes firewall based SDN security, early detection of DDoS attacks based on entropy and authentication based protection.

With the gradual development of SDN technology, it is highly possible that new security threats will arise. We assume that SDN can be one of the most important technologies with time to drive a number of network security advancements. SDN innovations also make it easier to computerize the cloud. The prominent characteristics supported by SDN are scalablability, highly reliablability, low latency distributed control plane etc, which helps SDN to miragte towards wide area networks and 5G core networks. In order to meet the high reliability and low latency requirements under varying amount traffic, the distributed control plane, needs to be managed dynamically

REFERENCES

[1] P. Krongbaramee and Y. Somchit, "Implementation of SDN Stateful Firewall on Data Plane using Open vSwitch," *2018 15th International Joint Conference on Computer Science and Software Engineering (JCSSE)*, Nakhonpathom, Thailand, 2018, pp. 1-5.

[2] M. Sinha, P. Bera and M. Satpathy, "An Anomaly Free Distributed Firewall System for SDN," *2021 International Conference on Cyber Situational Awareness, Data Analytics and Assessment (CyberSA)*, Dublin, Ireland, 2021, pp. 1-8.

[3] N. Zope, S. Pawar and Z. Saquib, "Firewall and load balancing as an application of SDN," *2016 Conference on Advances in Signal Processing (CASP)*, Pune, India, 2016, pp. 354-359.

[4] P. Rengaraju, S. S. Kumar and C. -H. Lung, "Investigation of security and QoS on SDN firewall using MAC filtering," *2017 International Conference on Computer Communication and Informatics (ICCCI)*, Coimbatore, India, 2017, pp. 1-5.

[5] A. Kumar and N. K. Srinath, "Implementing a firewall functionality for mesh networks using SDN controller," *2016 International Conference on Computation System and Information Technology for Sustainable Solutions (CSITSS)*, Bengaluru, India, 2016, pp. 168-173.

[6] M. F. Monir and S. Akhter, "Comparative Analysis of UDP Traffic With and Without SDN-Based Firewall," *2019 International Conference on Robotics,Electrical and Signal Processing Techniques (ICREST)*, Dhaka, Bangladesh, 2019, pp. 85-90.

[7] H. S. Abdulkarem and A. Dawod, "DDoS Attack Detection and Mitigation at SDN Data Plane Layer," *2020 2nd Global Power, Energy and Communication Conference (GPECOM)*, Izmir, Turkey, 2020, pp. 322-326.

[8] W. Sun, Y. Li and S. Guan, "An Improved Method of DDoS Attack Detection for Controller of SDN," *2019 IEEE 2nd International Conference on Computer and Communication Engineering Technology (CCET)*, Beijing, China, 2019, pp. 249-253.

[9] 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," *2015 17th Asia-Pacific Network Operations and Management Symposium (APNOMS)*, Busan, Korea (South), 2015, pp. 550-553.

[10] K. Hong, Y. Kim, H. Choi and J. Park, "SDN-Assisted Slow HTTP DDoS Attack Defense Method," in *IEEE Communications Letters*, vol. 22, no. 4, pp. 688-691, April 2018.

[11] B. Mladenov, "Studying the DDoS Attack Effect over SDN Controller Southbound Channel," *2019 X National Conference with International Participation (ELECTRONICA)*, Sofia, Bulgaria, 2019, pp. 1-4.

[12] Q. Yan, F. R. Yu, Q. Gong and J. Li, "Software-Defined Networking (SDN) and Distributed Denial of Service (DDoS) Attacks in Cloud Computing Environments: A Survey, Some Research Issues, and Challenges," in *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 602-622, Firstquarter 2016.