International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 07 Issue: 10 | Oct 2020 www.irjet.net p-ISSN: 2395-0072

SDN MODEL FOR DETECTION AND PREVENTION OF FLOODING ATTACKS

Dhanasekar V[1], Dr. Thangaraj N[2]

[1] *MTech- Information Technology, Department of Information Science and Technology, Anna University, Chennai, Tamilnadu, India*
[2]*Assistant Professor, Department of Information Science and Technology, Anna University, Chennai, Tamilnadu, India*

Abstract - *The primary goal of SDN is to improve network control by enabling service providers to respond quickly for changing business requirements and also enables the programmers to control the network behaviour as per their requirements. The implementation of SDN technology makes a significant difference in areas such as network programming, centralize control and abstract of the network. There are layers which construct the SDN architecture where there are various components integrated together to form a software defined network. Despite the maturity of the internet in years the computer networks of today are still prone to attacks, Security is an important criterion of the computer networks where there should not be any compromises. To protect the SDN network from the flooding attacks a model is created to detect and prevent the various flooding attacks by monitoring the packet flow to devices in the network through the device interface and stored in database as log files. If a flood is detected when monitoring the network, information of the attacker is retrieved from database and the attacker information is passed to the controller along with rule to drop the packets that is being sent by the attacker to all the devices in the network. So with this method the flooding is prevented from affecting the devices in the network. The objective of this work is to prevent the SDN network from the various flooding attacks, since the flooding attack is one that has a very high impact on the network performance as it can cause the victim to be unresponsive, preventing the victim from providing services, wasting the network resources. To maintain the network performance these attacks has to be prevented, since the network is not centralized it is difficult to prevent these attacks in the distributed environment as the devices in the SDN network may be virtualized.*

Key Words: SDN, Flooding attack, networks.

1.INTRODUCTION

Software Defined Network can be explained as the network architecture that enables the network to be centrally controlled (or) monitored using the software applications which helps the operator in managing the network consistently and holistically regardless of the base technology. The network is said to be secure only if it provides the basic properties such as confidentiality, integrity, availability and authentication. The security is an important criterion of the networks as there should not be any compromise in the network security if any could cause our network to be vulnerable to the attacks. The virtual network concept plays a major role in the form of virtualizing hardware and controlling the hardware devices through software application. The SDN architecture enables us to configure the switches at runtime and thereby it enables to manage the network resources efficiently. The SDN is very essential concept for various services such as deploying cloud services to incorporate new architecture for the further enhancement of the network efficiency or IOT in the networks. Switches usually run on the hardware but the virtual switches run on the computers as well as the servers hence it enables us to create a complex network architecture through software configuration. A typical network has the property of providing static and inflexibility as they possess little agility with the help of some network protocol, whereas SDN can make use of more flexibility over the network. The Software Defined Network divides the network into two separate components namely data plane and control plane. If the integration between these planes are not proper, then the network cannot be managed efficiently. The data plane can parse the packet from the network devices and then pass the content to the control plane where the control logic is present to manage the network efficiently. The planes can be separated and developed individually to form a system. The SDN architecture broadcast the information about the network to the application by the use of the network controller. There are various types of the network controllers that are available to control the activities in the network. The SDN controller has the ability to receive the instruction from the application so that it can pass this info to the network components and also it collects the information about the hardware devices and then communicate it back to the application to provide the abstract details of the network devices connected. The information includes events, logs, statistics that is required by the application to communicate with the network devices The SDN networking devices control the forwarding and data processing capabilities of the network.

The SDN API is of two types they are northbound interface and the southbound interface. The northbound interface is the bridge between the controller and the applications and the southbound interface is the bridge between the controller and the physical hardware devices in the network since SDN is a virtualized architecture where the network components need not be located in the same location. Control layer is the main area where the logic in SDN controllers would reside that would control the network infrastructure. This is the domain where each of the vendor is coming up with their own logic for building the products that can work with the controller and the framework.

An important issue regarding SDN network is the security, because the network is divided and virtualized in SDN environment making it vulnerable for the attackers to attack the network. The flooding attacks in SDN can affect various metrics in the network such as delay in providing network service, making the device in the network unresponsive, wasting the network resources, packet transmission delay in the network. Hence the flooding attack has a huge impact on the network performance and it is important to detect and prevent the flooding attacks in the SDN network to maintain the network performance. The main objective is to create rule-based attack detection model by refering the existing SDN security frameworks. The SDN model is then tested with common types of DoS attacks and ways to prevent these attacks. The tools that are used in the project are described as, ONOS (OPEN NETWORKING OPERATING SYSTEM), Mininet (NETWORK SIMULATOR), Snort (NETWORK MONITOR), sflow-rt (FLOW ANALYSIS TOOL.

1.1 ONOS - is an open source SDN controller maintained by the Linux foundation. The main aim of the ONOS is to create a SDN operating system for managing network components such as links, routers, switches and to provide communication to end hosts or to neighbouring networks. ONOS controller has its own set of applications that is built in to the controller to provide some of the basic functionality such as packet forwarding, openflow suite which makes the controller to monitor the switches.

1.2 Mininet - is an open source emulator that creates a virtual network environment on a single machine it is used to emulate SDN network which is comprised of hosts, layer 2 switches and SDN controllers, it runs on the linux based operating system. Open flow is one of the communication protocols that allows that instructs the switches where to send the packets in a network.

1.3 Snort - Snort is a network intrusion detection system which is capable of capturing the real time traffic analysis and packet tracking on the networks as well as carrying out some of the functions such as protocol analysis, content matching, port scans, cgi attacks and also detect variety of attacks. The packet filtering can be done in the snort by the use of the rules in the local folder or there is a set of default rules in the snort community where there are different types of rules created by the developers for the tracking of the packets by the various parameters.

1.4 Sflow-rt - it is a tool that receives continuous data from the open flow protocol present in the network device and converts the flow of data packet using the REST-API which can be viewed in the browser interface. It can define the flows, threshold's, flow records and periodic tasks. With these capabilities, monitoring of the network can be done easily that provides us the ability to maintain the performance of the network.

The rest of the paper is organized as follows. Chapter 2 describes the literature review; Chapter 3 describes the security model with the architecture and the various components in detail. Chapter 4 describes the implementation of functional modules and working of the modules. Chapter 5 describes the conclusion and future work on this model.

2. RELATED WORKS

The software defined networks are an emerging domain that has the ability to change the working of the traditional networks. It is done by dividing the network's control logic and the network devices which in turn gives us the way to program our network [9]. SDN makes the network management easy by dividing the control logic with the network devices. There are two types of interfaces one is northbound interface and the other is the southbound interface. The northbound interface is the bridge between the application and the controller whereas the southbound interface is the bridge between the controller and the network devices [10]. Some of the terminologies that are used to identify the elements in the SDN environment are forwarding devices, data plane, southbound interface, northbound interface, control plane and management plane. The SDN network is composed of various layers such as infrastructure layer, southbound interface layer, network hypervisor and controller layer. The infrastructure layer is composed of the network devices where the network intelligence is removed from the data plane devices (switch, routers) to centralized control system. The southbound layer interface that connects the control and forwarding elements which is considered as the crucial instrument for separating control and data plane functionality. The network hypervisor layer is used to share the same hardware resources to the distinct virtual machines which is common in the cloud service environment. The important feature of the virtualization is that the virtual machines can be easily migrated from one physical server to another, can be created or destroyed which provides the flexible and easy management services [1].

The network operating systems/controllers enables us to facilitate the network management and solve network problems by logically centralized control offered by the operating system. NOS is a critical element in an SDN

architecture as it is the key of the control logic for the creation of the network configuration based on the policies by the network operator. The ONOS architecture has basically four layers and two APIs they are application layer, core layer, provider's layers, protocol layers and southbound interface API, northbound interface API [5]. Some of the issues can be in the components in the SDN network such as network protocol, switches, controller and network size. The issues in the network protocols are flow rule prioritization, protection against traffic overload, security services composition. Switches in SDN has some of the issues to be handled such as flow table capacity, performance of the switches, switch designs and switch enhancements, SDN support for new switch designs. The controllers can also have some of issues such as flexibility, modularity, interoperability, portability and availability. Based on the network size there can be various issues in the SDN network such as decoupling of the control plane and data plane, network traffic which increases the network load, latency or overhead on the control plane because the controller has to process millions of flow per second without compromising on the quality of the service [1]. The counter measures that can be taken to improve the security of the SDN network such as classifying applications, rule prioritization, rules generation by security application and creation of a framework for developing security applications in the software defined networks.

Monitoring is an important term in the network management which helps the network operator to monitor the behaviour of the network and as well as the operation of the individual components in the network. Traffic engineering is one of the important aspects for decision making by monitoring the network. There are other functions such as Quality of service and anomaly detection that plays an important role in providing secure networks. There are various challenges associated with the monitoring of the SDN networks. Monitoring the SDN networks is complex due to the way the network is laid out which is in a distributed manner so there are various challenges that has to be faced while monitoring the SDN networks Quality of service is an important aspect where the service has to be evenly distributed or provided irrespective of the data packet flow [8]. There are some tools which are used to monitor the flow of packets in the network such as Ntop, Wireshark and Argus. These tools enable us to control the network by means of providing the information about the flow of the data packet. Monitoring is complex in SDN because of the separation of the control plane and the data plane since the central controller has the information about the network and has the ability to modify it based upon the user requirement. The open flow protocol is used by the opendaylight controller that operate in the Mininet simulator environment and also there are various network devices such as routers, links and switches. The controller communicates with the switches by using the open flow protocol so that the controller can control the flow of the data packets in the switches. Wireshark is a packet capture tool that is able to recognize the open flow protocol and it has the capability to filter the packets by using various filtering parameters such as IPaddress, protocol type and flow rate. It is used in many applications or framework for providing network security [10]. The monitoring of SDN can be a difficult task due to the virtualization of the network. There is sequence of steps are to be followed while monitoring the network such as identification, measurement and decision making. The monitoring is done by capturing the data packets in the network and analysing the various aspects which is then compared with the user metrics and if the match is found then the appropriate action is to be triggered [3]. The issues in monitoring the SDN network is one of the important aspects as it can have a negative effect on the performance of the network if the issues are not resolved. Some of the issues that can be faced while monitoring the networks are adaptive measurement technique based on the size of the network which to avoid the wastage of the resources, the real-time analytics focuses on the various parameters such as flow management and fault tolerance, topology update is done to effectively to monitor the network and cyber security support is incorporated to cater the control plane attacks. Cloud application integration is required due to the increase in cloud services as there is need to enhance the network functionality for handling traffic in the cloud. Application aware networking is needed to incorporate the application-driven adaptations and on-demand services into the software defined networks [8].

The security is an important criterion of the networks there should not be any compromise in the network security if any could cause our network to be vulnerable to the attacks. Identify various challenges associated with each layer of the framework such as application, data and control. Some of the areas where the security issues can occur in a SDN framework are stride threat methodology on open flow switch, what provides authentication between the controller and the switch using the TLS. The security categorization in SDN is defined by the type of attack as well as by the layer /interface which the attack is intended on. If the TLS is not in use then there is a strong chance that there could be rule modification or irrelevant rule insertion that may cause some of the flows to be modified in the network [9]. The SDN provide us with various features to be incorporated in our network such as to reprogram the network and creation of dynamic flow policies. In fact, these advantages act as the main root cause of the security attacks along with dynamic network flow policies. There are very few models created for a secure SDN environment that does not ensure the full security protection as there is always place for vulnerabilities. New methods and techniques must be explored to expand on how to program the network by making some changes in the security monitoring and as well as detection of the attack [7]. There are various issues in providing security in SDN

network and the issues are categorized as per the level of damage it can cause to the network such as unauthorized access, data leakage, data modification, malicious applications, denial of service and configuration issues. A number of security analyses have recently been done which shows the relationship between elements in the SDN framework that introduces new vulnerabilities [9]. Some of the attacks in the openflow networks are spoofing which defines authentication of data, tampering defines integrity of data, repudiation which defines modification of data, information disclosure which defines the confidentiality of the data, denial of service refers to the availability of the data and elevation of privilege defines the authorization of the data packet in the network [1]. The countermeasures to overcome the attacks in the openflow protocol are access control, attack detection, event filtering, firewall, flow aggregation, forensics support, intrusion tolerance, packet dropping, rate limiting, shorter timeouts and with these measures it can restrict the security attacks in the network protocol [9]. Despite the maturity of the internet in years the computer networks of today are still prone to attacks, the distributed nature of the network for wide area led to the high difficulty in the design and implementation phase of the networks.

SDN provide us an option of monitoring and defence across the network but the problem is that the current software defined security systems are just centralized framework that introduces significant control plane overhead so to overcome this a new framework was introduced which is capable to control and monitor with the functionality of resilience and as well as scalability of a distributed network systems [1]. This framework offers effective monitoring and remediation, compatibility with wide available legacy networks and modern distributed design and network threat detection and protection in traditional network can be achieved with dedicated hard security appliances which are costly and require careful placement in network to ensure traffic capture. This framework defines the working of the multiple controllers on network for the attack detection and prevention, the framework consists of various services that can be used by the co-ordinator who has ability to monitor the flow of the network as well as to control the network.Some of the components that is used in this framework are snort - is a packet inspection tool to detect the flow rate of the packets and it will inspect the packet based upon the rules that has been written in the rules file. Snort is capable of monitoring the network flow and also the log file from the snort tool can be written into a database for the tracking of the packets in the network. Sflow-RT is a tool that collects and provides sample flow record statistics when the flow volume is high in the network the sflow collector collects the info from the sflow-rt and passes the packets to the central co-ordinator where the decisions are made based upon the packets that are sent inside the network [6]. The tennison co-ordinator collects information form the various services that is running in the SDN network with the help of these information the co-ordinator can monitor and control the network and remediate against various attacks in the network. There is a component called the policy engine where the decision is taken based on the some of the flow parameters from the network such as threshold, flow rate if the current network conditions matches with the conditions already stored it takes the actions as per the predefined conditions. Security pipeline is present in this framework consists of the following components such as tunnel table which is used to classify tunnelled traffic, stripping table is where the forwarding the logic from the mirrored flow, Remediation table which consist of the tunnelling management this is done to control the network performance by blocking or dropping traffic identified as malicious before it uses further resources, ipfix table contains the monitoring intents which forwards the inspected data packets to the forwarding pipeline [9]. Network controller is the one of the core components in the framework so it has some of the main functions such as getting the input data from the data plane and transfer this information to the control plane where the co-ordinator is present where the decision taking is done according to the rules in the policy engine. This implementation supports optimization of network monitoring and protection based on appropriate controller, switch assignment and monitoring rule placement. some of the issues includes automation of the scaling process, including provisioning additional controllers to meet increased network load. In addition, the potential for scaling at different layers within the tennison architecture. In the history of networking the DoS/DDoS attack is a common type of attack which is simple to implement but it is very trusted method which attacks the internet system. DoS attack is a method of trying to make a network, machine resource unavailable to the legitimate users which in turn causes service interruption to the hosts that are present in the network. In DoS attack, the attacker who is known as a master zombie tries to control many systems and these systems are known as slave zombies. By using these slave zombies, the master zombie tries to attack the system by flooding the packet [2].

The DoS attack can be done using the automated tools that can cause flooding if the attacker finds the system with very weak security then he immediately attacks the system with these tools. DoS attack not only attacks a system or a server it can also attack cloud environment making the resources in the cloud environment unsafe or the attacker can degrade the quality of the service or breakdown the victim's connectivity to the internet. There are various types of DoS attacks are denial of sleep attacks, udp flood attack, ICMP flood attack, syn flood attack, ping of death attack [9]. Denial of sleep attack is defined as the attack that is aimed towards the nodes power consumption and also the attacker knows the information about the MAC layer protocol and it possess ability to bypass authentication and encryption protocol. UDP flooding is a deceptive protocol because

information packets or request may arrive out of order, may appear to be duplicate or may be delayed. UDP protocols generate a larger bandwidth DDoS attack because they have the property of connectionless and easy to generate as it does not require any permission to transfer packets. ICMP is similar to UDP, ICMP Ping request continuously sends packets as fast as possible without expecting any replies, so that both incoming and the outgoing bandwidth will be increased leading to attack on the size of the ports and the ICMP protocol is basically for the request and reply communication between the source and the destination they continuously send packets. SYN flood attack is where the packets have been sent continuously to the server by the attacker in order to prevent the connection being closed. During the connection period other systems will not be able to access the server this is one type of DDos attack, other type of the SYN flood attack called as a spoofed SYN flood attack where the attacker tries to send a huge amount of TCP SYN packets with a false IP address. DDoS can be detected by several methods they are fast entropy approach, naïve Bayesian classification method, Tennessee Eastman challenge, TCP congestion window analysis. Fast entropy method is purely based on the packet header information. The header consists of the IPs of sender and the receiver and also the details of the flow of traffic so it doesn't care about the data in the flow. Among the available data it is important to give priority to the data that is needed for detection and the data that is not necessary. Tennessee Eastman identification method is able to identify which is attacking the networking systems which grant full access to network and they disrupt the equipment. In the TCP method the detection is done using the TCP based flooding attacks by using TCP congestion window which is analysed using the cumulative sum [2]. Issues in SDN Networks are, Performance of the network, Scalability of the network, Security of the network, Interoperability, Cost of implementation, Reliability of the network, Denial of service, Virtualization of the network, Network resource usage. Issues Solved in Proposed Method, Performance of the network, Security of the network, Denial of service, Network resource usage.

3. PROPOSED DESIGN

The proposed system is designed to work with the SDN network environment for identifying and preventing the common types of DoS attacks such as TCP flooding, UDP flooding and ICMP flooding. There are various tools that has been used to work together as a system by referencing already existing SDN security frameworks and the working of this SDN model is based on gathering various information from the devices in the network and passing the information to the network tools such as the snort and sflow-rt. The sflow-rt stores these parameters and provide the necessary action to prevent the attack based upon the information from the devices and the rules are pushed from the main controller to the necessary devices in the network. The functional modules consist of three modules, namely

- Topology Creation
- Tcp Flooding
- Icmp Flooding

There are some of the pre-requisites that are to be taken care in order to run the ONOS controller such as topology creation by setting up the controller and building controller with the bazel framework. The Mininet network simulator is used to run the topology.

- **Topology Creation-** The topology creation and ensuring communication among entities is the initial process where the user creates the virtual network in a Mininet simulator with various components such as controller, switch, host, legacy switch. The switch is one of the main components by which the hosts get connected and as well as every packet must pass through switch from one host to the other host in the network. Monitoring the switch will be able to keep track of the packets that is being transferred from one host to another host. forwarding applications has to be enabled in order to transfer the packets from one host to the other host in the SDN controller as well as enabling open flow suite applications for the packet transfer**.**

- **TCP Flooding-** The detection and prevention of the TCP flooding is done by analyzing the packet in the network. If the flow of the packets matches the rule that is present in the snort detector then the snort will alert us with the message that the TCP flood is happening in the network. It also provides us information about the host that is trying to cause the flooding. The communication with the controller is established to push the rule to drop the packet sent by that host and hence the prevention of the host trying to flood the other host in the network is carried out. Further it can filter the data packets by various parameters such as protocol, IP address of the source and the destination and flow of the packets.

- **ICMP Flooding-** The ICMP flooding floods the victim with high rate of echo request packets expecting replies from the victim that may ultimately reduce the performance or stall the victim from providing services. If the flow of packets matches the rule that is present in the snort detector, the snort will alert with the message denoting the ICMP flooding along with the information about the host that is trying to cause the flooding. The communication with the controller is established to push the rule to drop the packet sent by that host and hence the prevention of the host trying to flood the other host in the network is achieved.

These flooding attacks can stall or reduce the performance of the network; hence it is important to detect these attacks and to prevent them. For each type of flooding attack, the detection rule is different since each attack uses different types of protocol. It is important to identify the type of attack and take the preventive measures to stop the attack.

3.1 System Design

The system design shows the various components that are integrated as a system. The components used in this model are snort, barnyard, MySQL database and sflow-rt. The Mininet simulator is the tool where the creation of the network is done as per the user requirement and then the network is connected with the controller ensuring the hosts in the network can communicate with each other as the Mininet uses the open flow protocol to communicate with the controller.

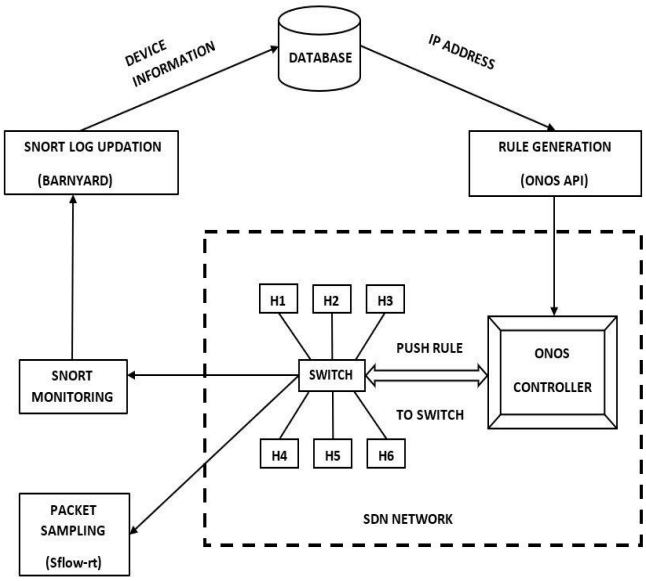

Fig.-1: SDN Dos Attack Model

Fig. 1. explains the security model that is developed for detection and prevention of the common type of DoS attack by referring various security frameworks. The interface of the SDN is divided into two types namely, the northbound interface and the southbound interface. The northbound interface is the communication between the SDN application and the controller whereas the southbound interface is the communication between the devices in the network and the controller. Southbound interface is responsible for collecting information from the various components in the network and providing them to the controller so that the controller can control or monitor the network. Sflow-rt is a tool that collects sample flow record statistics which is used for the tracking of the data packets based on the parameters such as source IPaddress, destination IPaddress, protocol and rate of flow. There are various components that has been used in this model which has their own functionality but are integrated together as a system. The snort tool monitors the switch interface for any communication between the hosts and tracks the packets flowing through the switch in real time and then store the packet flow information as a snort log file which contains the information about the packets such as source address, destination IP address, checksum, data payload and HTTP header.

3.1.1 Snort Monitoring- The snort tool captures the data flow from the switch and stores it as a log file which is further used for the retrieval of the information about the host trying to flood the network. The IPaddress of the attacker host is retrieved and the rule is pushed from the controller to the switch and drop the packets received from this host which in turn protects other hosts in the network from the flooding attack. Barnyard is a tool that works in conjunction with the snort tool which has the functionality to read the snort log file which is of unified 2 format. Then it stores the information into the database in a table format which is retrieved for further processing of the data packets and to identify the host that is trying to flood the network. Basic Snort rule to monitor network:

alert any any -> any any(flags: s; msg: Test rule; sid:100001;rev:1;). Snort can monitor based upon the pre-defined basic rules or the user can write custom rules to monitor the interface, the filtering of the packets can be done based upon the protocol that is used or by source address, destination address, flow rate. The snort monitors the interface based upon the rules that the user has defined in the local rules file into the snort folder.

3.1.2 Snort Log Updation - The barnyard tool is used for storing the data into the database and is also capable of decoding the unified 2 data format that the snort log generates. To store the log data into the database the barnyard configuration file has to be added with database details such as database name, password, database connector and local host information. After establishing the communication with the database, IP address of the attacker host is retrieved in a binary format from the database and then converted to IP address format. with this information the controller will be able push the rule to the particular device in the network. To run the barnyard tool, the path of the snort configuration file along with configured barnyard file has to be specified. The barnyard can run in the daemon

service where the process will run in the background as the service need not run every time manually.

3.1.3 Rule Generation- As soon as the information about the attacker host is retrieved from the database. By using the API, the controller exposes the device information in the network. With the device information from the controller the rule is generated based on some parameters such as priority, timeout, criteria, deviceid, ethtype, IPtype, treatment and it is pushed to the particular switch to which the attacker host is connected. The request method is used to communicate with the SDN controller which returns status code 200 for success otherwise with status code 400. The device information response by the controller to the request method is in json format which is then converted as text. The controller represents the rule in a specific format and it is pushed to the switch using POST method.

3.2 Flow Diagram of SDN Model

The topology creation and ensuring communication among entities is the initial process where the user creates the virtual network in a Mininet simulator with various components such as controller, switch, host, legacy switch. The connectivity between the hosts should be ensured so that the hosts in the network can communicate with each other which is necessary for the flooding attack to be initiated. The flooding attack detection and prevention workflow is explained in the Fig.2. The snort configuration file has to be configured with the details of the network interface to be monitored as well as the snort rule to identify the flooding attack. The snort monitors the packet flow in the switch interface, If the rule matches the packet flow rate then the flood is identified and stored into the database by using barnyard tool which is capable of reading the log files generated by snort. The information about the devices in the network is stored in the database and then from that information about the device causing the flood is retrieved and checked with the controller using the API that the controller exposes. If the communication is successful then the device details is retrieved. If the device is found and then the rule is pushed from the controller to the particular switch in a specific format stating to drop the packet sent from the particular device. This prevents other hosts in the network from flooding effect.

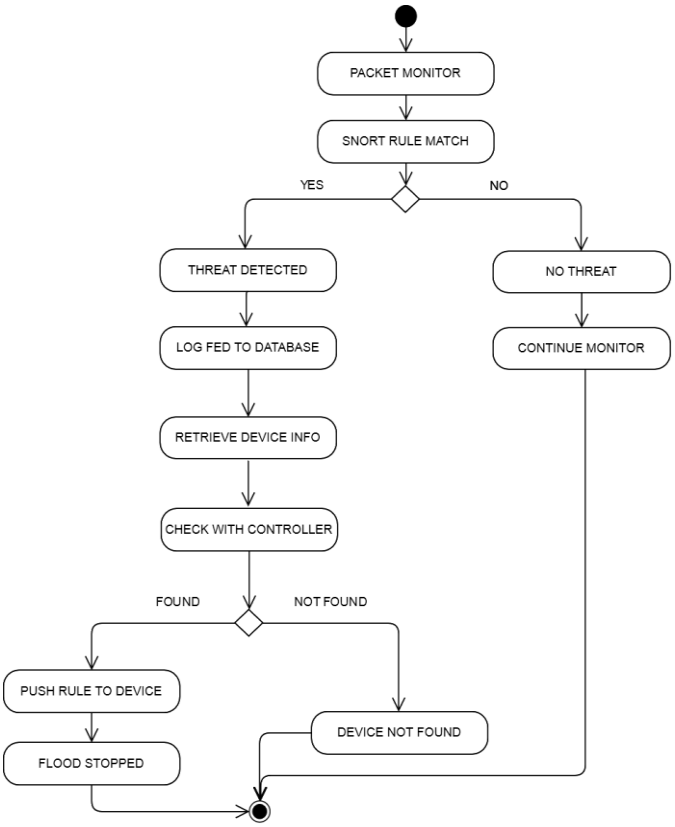

Fig- 2: Flow diagram of SDN model

4. IMPLEMENTATION AND RESULTS

4.1 Topology Creation- The topology creation is the initial step in the SDN network and the network topology can be created as per the user requirement. The topology used in this work consist of six hosts connected to a switch. The below mentioned steps are to be followed in Mininet simulator for creating the network topology. Steps for the network topology creation:

Step 1: Import appropriate headers.
Step 2: Initialize network function.
Step 3: Initialize the controller with necessary parameters.
Step 4: Add the components to the network.
Step 5: Assign the hosts to the switches.
Step 6: Initialize the network.
Step 7: Run the controller.
Step 8: configure the switches and execute the host.

The topology creation is done as per the steps mentioned above using python and the controller identifies the network. The output of the code compilation is shown in Fig.3.

```
dhanz@dhanz-VirtualBox:~/mininet/examples$ sudo python sflood.py
[sudo] password for dhanz:
*** Adding controller
*** Add switches
*** Add hosts
*** Add links
*** Starting network
*** Configuring hosts
h5 h2 h3 h1 h4
*** Starting controllers
*** Starting switches
*** Post configure switches and hosts
*** Starting CLI:
mininet> xterm h3
mininet> xterm h4
mininet> pingall
*** Ping: testing ping reachability
h5 -> h2 h3 h1 h4
h2 -> h5 h3 h1 h4
h3 -> h5 h2 h1 h4
h1 -> h5 h2 h3 h4
h4 -> h5 h2 h3 h1
*** Results: 0% dropped (20/20 received)
```

Fig-3: Network Creation

The network is created using Mininet and the connectivity of the hosts are verified by ping requests. The topology can be viewed by running the controller and the structure of the topology is shown in Fig. 4.

Commnad to build SDN controller : sudo bazel build ONOS

Command to run SDN controller : sudo bazel run ONOS-local

The above mentioned commands does the fucntion of building and running the ONOS controller which is used to monitor the network created by the mininet.

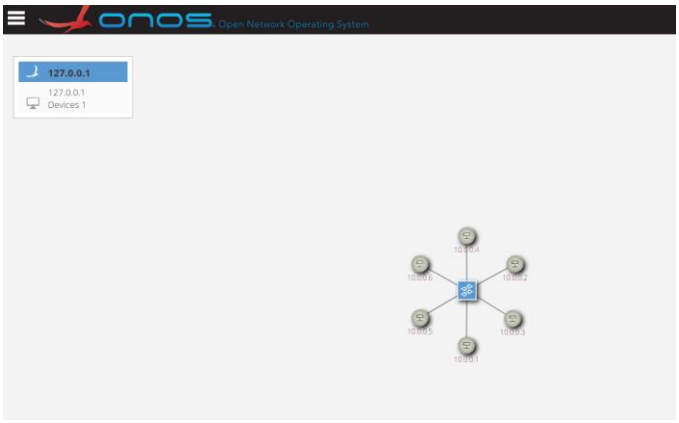

Fig- 4: Network in SDN controller

The above Fig.4. represents that the controller has identified the network which is created. Inorder for the host to communicate with each other activate the forward application as well as the openflow application from the applications tab in the SDN controller user interface. The connectivity of the hosts can be verified by pinging the respective hosts.

4.2 TCP Flooding- For the implementation of TCP flooding, a host is chosen as a attacker in the network which tries to flood other hosts in the network. The hping3 tool creates a TCP flood for the targeted host in the network. The command to execute TCP flooding is specified below

Command: sudo hping3 –S -c – i –flood "IPaddress to flood". After this command is executed, the attacker host starts to flood the targeted host with TCP packets. The Fig. 5. shows the screenshot for the host1 flooding host 5.

The snort interface monitors the switch and identifies that the flooding has occurred and passes the information to the barnyard tool which inturn stores the information about the attacker host and the targeted host into the database. Snort command to start monitoring network interface:

sudo snort -q -A console -u snort -g snort -c /etc/snort/snort.conf -i "interface to be monitored"

snort rule to detect TCP flood: **alert tcp any any -> any any (flags: s; msg: "TCP flood detected"; flow: stateless; threshold: track by_src, count 500, seconds 1; sid:10001; rev: 1;)**

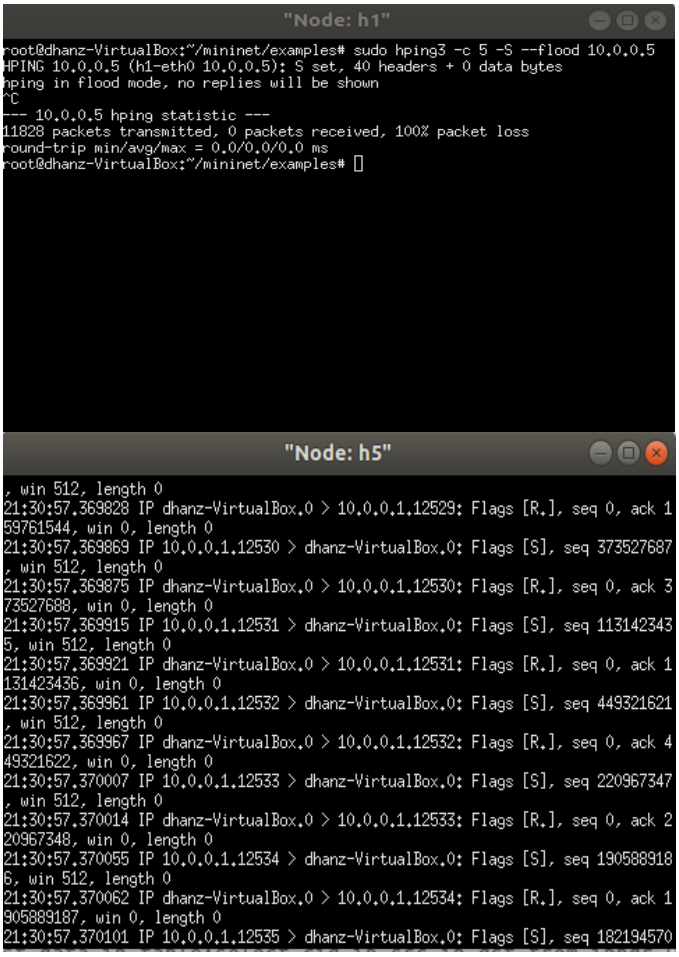

Fig- 5: Host flooding another host

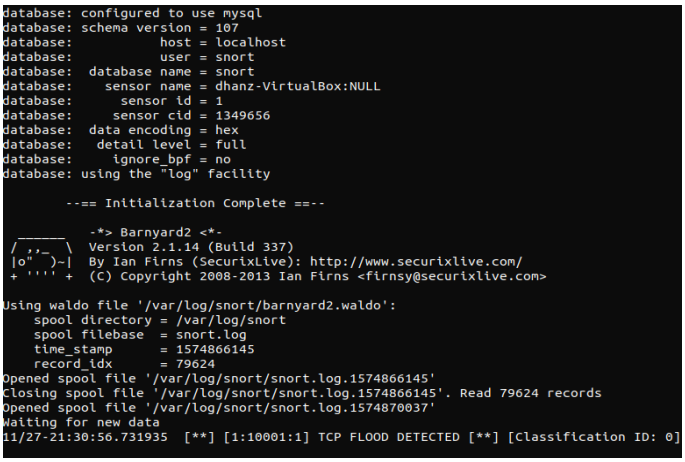

Fig- 6: TCP flood is detected using snort

The information about the attacker and victim hosts is fetched from the database using the python script which contains the code for the database communication.

connection = mysql.connector.connect(host='localhost', database='snort', user='snort', password='Dhanzs@1')

After the connection to the database is established successfully the controller is communicated to retrieve the device information in the network using the API that the controller exposes and the device information is sent by the controller and the rule is pushed to the switch by the controller.

4.3 Flow rule from controller to switch:

```
"flows": [
  {
   "priority": 5500,
   "timeout": 5000,
   "isPermanent": true,
   "deviceId": "of: 0000000000000001",
   "treatment": {drop
    "instructions": [ drop the packets received from the attacker host]
  },
   "selector":
   {
    "criteria": [
     {
      "type": "ETH_TYPE",
      "ethType": "0x800"
     },
     {
      "type": "IPV4_SRC",
      "ip": srcip
     },
     {
      "type":'IP_PROTO',"protocol":'6'
     }
    ]
   }
  }
 ]
}
```

To check if the rule is pushed to the device in the network by using the following command.

Command: sudo ovs-ofctl dump-flows s1

This command shows the new flow rule inserted into the switch the screenshot of the Fig. 7.

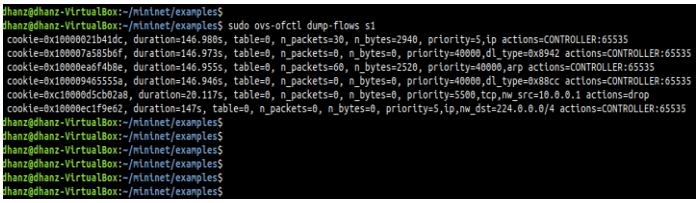

Fig-7: Rule pushed to switch

After the rule is pushed to the switch, the same flooding attack is executed where the host 1 tries to flood the host 5 but the host 5 is not affected by the flood because the rule to switch is as follows “drop packets sent from the host 1”.

4.4 ICMP Flooding

For the implementation of the ICMP flooding a random host is chosen as a attacker in a network which tries to flood other hosts in the network. The below mentioned command is used to start the ICMP flood attack

Command: sudo ping –s 100 –f "IPaddress to flood"

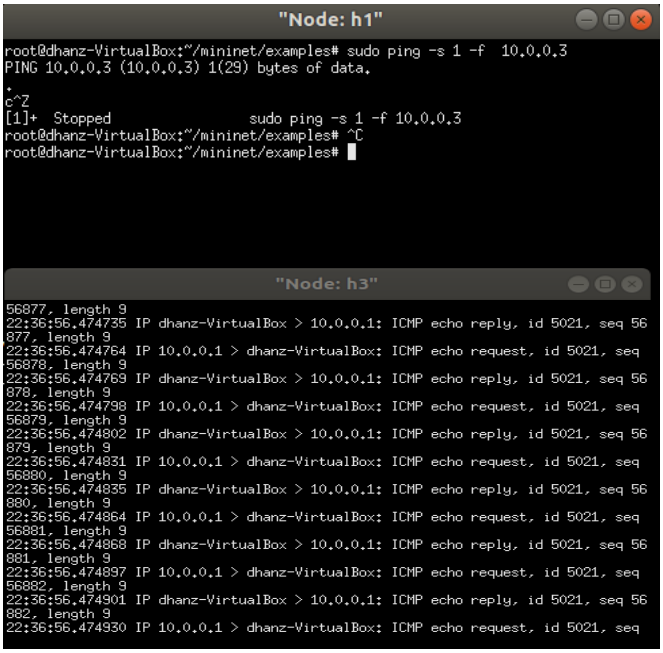

Fig- 8: Host flooding another host

The above Fig. 8. shows that the host h1 floods the host h3, where the host h3 receives high rate of packet flow from the host h3 which reduces the performance of the host h3 or causing the host to deny the services that it can provide.

Snort command to monitor network interface:

sudo snort -q -A console -u snort -g snort -c /etc/snort/snort.conf -i "interfacae to monitor"

snort rule to detect ICMP flooding:

alert ICMP any any -> any any (flags: s; msg: "ICMP flood detected"; flow: stateless; threshold: track by_src, count 1000, seconds 2; sid:10002; rev: 2;)

The snort interface monitors the switch interface identifies that the flood is happening as shown in Fig. 9. and then passes the information to the barnyard tool which inturn stores the information about the attacker host and the targeted host information into the database. The information about the attacker and victim hosts is fetched from the database using the python script which contains the code for the database communication.

connection=mysql.connector.connect(host='localhost', database='snort',

user='snort', password='Dhanzs@1')

After the connection to the database is established successfully the controller is communicated to retrieve the device information in the network using the API that the controller exposes and the device information is sent by the controller and the rule is pushed to the switch by the controller.

Fig. 9. ICMP flood detected using snort

Flow rule from controller to switch:

```
"flows": [
 {
"priority": 6000,
"timeout": 7500,
"isPermanent": true,
"deviceId": "of: 0000000000000001",
"treatment": {drop
        "instructions": [ drop the ICMP packets from
    attacker   host]
  },
  "selector": {
```

```
    "criteria": [
     {
      "type": "ETH_TYPE",
      "ethType": "0x800"
     },
{
      "type": "IPV4_SRC",
      "ip": srcip
},
{
            "type":'IP_PROTO',"protocol":'1'
    }

   ]
  }
 }
]
}
```

To check if the rule is pushed to the device in the network by using the following command.

Command: sudo ovs-ofctl dump-flows s1

This command shows the new flow rule inserted into the switch the screenshot of the Fig. 10.

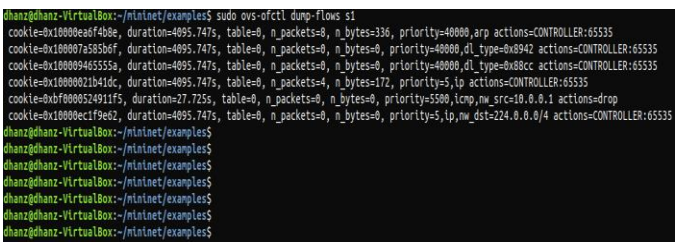

Fig. 10. Rule pushed to switch

After the rule has been successfully pushed to the switch, packets arriving from the particular host will be dropped, this prevents the attacker host from flooding the other hosts in the network.

5. CONCLUSION

The SDN security model developed is able to detect and prevent DoS attacks such as TCP flooding and ICMP flooding. The attack is detected by monitoring the data flow in the network and also by gathering the information about the devices in the network by identifying the host that generate flooding. Then the controller pushes the rule to the switch to drop the packets received from this particular host which ensures the prevention of other hosts affecting from flooding attacks in the network. The future work in this model includes integration of snort monitoring tool as an app in the ONOS instead of running it as a separate service. An intelligent module can be created to induce machine learning techniques to analyse the behaviour of the network and to predict an attack rather than sending all the traffic to the snort tool to detect an attack, which in turn reduces the chances of false identification of the attack.

REFERENCES

[1] D. Kreutz et al., "Software-defined networking: A comprehensive survey", in the journal of Proceedings IEEE, Vol. 103, pp. 14-76, 2015.

[2] Dong, S., Jain, R., & Abbas, K. “A Survey on Distributed Denial of Service (DDoS) Attacks in SDN and Cloud Computing Environments” in the journal of Proceedings IEEE Access, pp.1–1, 2019.

[3] G. Bianchi, M. Bonola, G. Picierro, S. Pontarelli, and M. Monaci, “StreaMon: A software-defined monitoring platform,’’ in the journal of Proceedings IEEE, Vol.65, pp. 1–6, 2014.

[4] K. Narasimha Mallikarjunan, K. Muthupriya, S. Mercy Shalinie, “A Survey of Distributed Denial of Service Attack”, in the Conference of Intelligent System and Control, Vol.32, pp. 1-6, 2016.

[5] Kim, W., Li, J., Hong, J. W.K., & Suh, Y.J, “OpenFlow monitoring system in ONOS controllers”, in the Conference of IEEE NetSoft and Workshops, Vol.14, 2016

[6] L. Fawcett, S. Scott-Hayward, M. Broadbent, A. Wright, and N. Race, “TENNISON: A Distributed SDN Framework for Scalable Network Security,” pp. 14, 2018.

[7] M. Tyson, “FRESCO: Modular composable security services for software-defined networks,” in

Proceedings of Network and Distributed Security Symposium, 2013.

[8] P.W. Tsai, C.W. Tsai, C.W. Hsu, C.S. Yang, "Network monitoring in software-defined networking: A review", in the journal of IEEE Systems, Vol. 12, pp. 3958-3969, 2018.

[9] S. Scott-Hayward, G. O'Callaghan, S. Sezer, "SDN security: A survey", in the journal of Proceedings IEEE Software-Defined Networks for Future Networks and Services, Vol. 42, pp. 1-7, 2013.

[10] S. R. Chowdhury, M. F. Bari, R. Ahmed, and R. Boutaba, ‘‘PayLess: A low cost network monitoring framework for software defined networks’’, in the journal of Proceedings IEEE Network Operating Management Symposium, pp.232-236, 2014.