Recent Trends in Intensive Computing
M. Rajesh et al. (Eds.)

doi:10.3233/APC210186

Study on SDN with Security Issues Using Mininet

Ms. Florance G [a,1] and Dr R.J Anandhi [b]
[a] *Research scholar, Department of CSE, TOCE, Bangalore, Affiliated to VTU, Belgaum*
[b] *Professor & HOD, Department of ISE, NHCE, Bangalore, Affiliated to VTU, Belgaum*

Abstract. The internet is faced with many problems daily, one of them is decrement in network bandwidth because of Distributed Denial of Service (DDoS) attack on host server, which deplete host resources. Researchers has been invented many protection mechanisms such as detection, trace back, prevention, reaction, and characterization are in case of DDoS attacks, which will control the number of malicious packets received by the victim. But it does not provide efficient detection technique with high rate in real time network infrastructure. Thus, modern technologies are prepared on Mininet network simulators, which give more impact to simulate the real network. The architecture of Software Defined Networks (SDN) and OpenFlow architecture is used to demonstrate a programmable network model and centralized management of real network. In this research work, we provide design of software defined network (SDN) using mininet simulator and security issues related to the Software Defined Network.

Keywords. DDoS attacks, Software Defined Network, OpenFlow, and Mininet simulator.

1. Introduction

Distributed denial-of-service (DDoS) attack makes any organization fail, thus resulting to stop providing services to legitimate users and exhaust the victim's resources. DDoS attacks can be classified as resource and bandwidth consumption. Attacker will perform DoS attack on more than one network to destroy the victim's resources, so that victim is unable to provide regular network services [2].

The primary cause of a Distributed denial of service attack is that attackers are typically accomplished by flooding the victim network from different sources. Nowadays high-rate DDoS (HDDoS) attacks certified with the persistent detection techniques quickly [2]. Presently, sending more packets to different network by using DDoS attacks which will flood traffic. Such attacks have been increased because the attacker wants to disturb the entire network to stop the legitimate packet to reach the destination. Due to presence of weakness in Internet Protocols, it becomes very easy for the attacker to find and exploit different loop holes in different applications.

The data plane and control plane are combined very strictly on the device in traditional networks. Consequently, creating the new set of programs to perform the task and changes in devices which is already present is a very monotonous task [10]. Software Defined Networks (SDN) surmounts these difficulties by separating the data plane and control plane, simplifying network management.

[1] Ms. Florance, Research scholar, Department of CSE, TOCE, Bangalore, Affiliated to VTU, Belgaum, Email: vijiflorance59@gmail.com.

The important role of SDN is to increase the capabilities of traditional networking system, where the network control is changing dynamically, data transfer, managing, adjustable, and extracted from SDN devices [1].

Software defined networks (SDNs) are one among the most frequently used software-based network model having loosely coupled control and data planes. By supporting the centralized control techniques and application programming interface, SDNs have enlarged the changeability of network management and its functions [1]. The network's control logic plane has controller is disconnected from the data plane by the Software-Defined Networking (SDN). SDN solves the problem of dynamic nature, scalable computing and storage needs of more computing environments that the consistent architecture of conventional network does not support [3].

2. Background and motivation

This portion expresses the biography as well as the necessary of defence mechanism for DDoS attacks. In the beginning, DDoS attacks were first started in August 1999 in case of different consortiums and keep attacking in different organization [1]. In the year of 2009, a DDoS attack was started that disturb the organization assistance of most preferred websites like Live Journal, Facebook, Amazon, and Twitter. In the year of 2010 to 2011, more than 80,000 computer systems in 2500 consortiums and 4 million computers in 100 countries were disturbed by DDoS attacks correspondingly. For each day, attacker started sending more than 8000 DDoS attacks. But recently DDoS attack has happened in short duration [2].

The Internet hosts are used to emanate a DDoS attack, because of having the frail protocols, insufficient attestation plans, insecure computer systems and operating systems. Weak systems require stringent protocols for implementing a higher security standard which give the better remedy to the secure systems. Attack detection techniques do the experiment of analysing on the attack packet and disclose the attacks then remove attack packet from normal traffic. One of the most important technique is Traceback technique which is used to trace source of attack and even in the case of spoofed IP addresses before the origination of attack [2]. During the processing time attack reaction strategy try to reduce the loss, which is created by DDoS attack. The Reaction stage mitigates the impact of attack and increments the quality of services delivered to the authenticate users under attack.

3. Software Defined Network architecture

Software Defined Network (SDN) is a new networking innovation, which permits centralized, programmable control logic planes and data plane conceptual that can overcome the drawbacks of present network infrastructures [3]. SDN is designed to make a network flexible and is logically centralized through SDN controllers. SDN controller provides a centralized vision of the whole network [9]. The reason behind SDN is to keep the data forwarding plane will be separated from the control plane so that network operators and service providers can directly control and manage their own virtualized resources and networks beyond using hardware mechanism. It is an application used to control packet forwarding all the devices in the network and manages flow control for improved network management [1]. Manual configuration is reduced, for individual network devices because of the forwarding policies. In SDN, the control and data planes are distinguished which in turn allows control to be

programmable and manageable such that control remains centralized and data plane to be shorten and conceptual [10].

The motivation that SDN is necessary for network operator and service providers are as follows [3],

- Network operator and service provider need to use SDN technologies to easily and efficiently control and manage the network.
- High complexity of operations and management in network can be reduced with software defined network rather than configured networks.
- Network operator and service providers should provide an interaction method between the infrastructure layer and network layer so that service should be securely isolated from existing traffic.

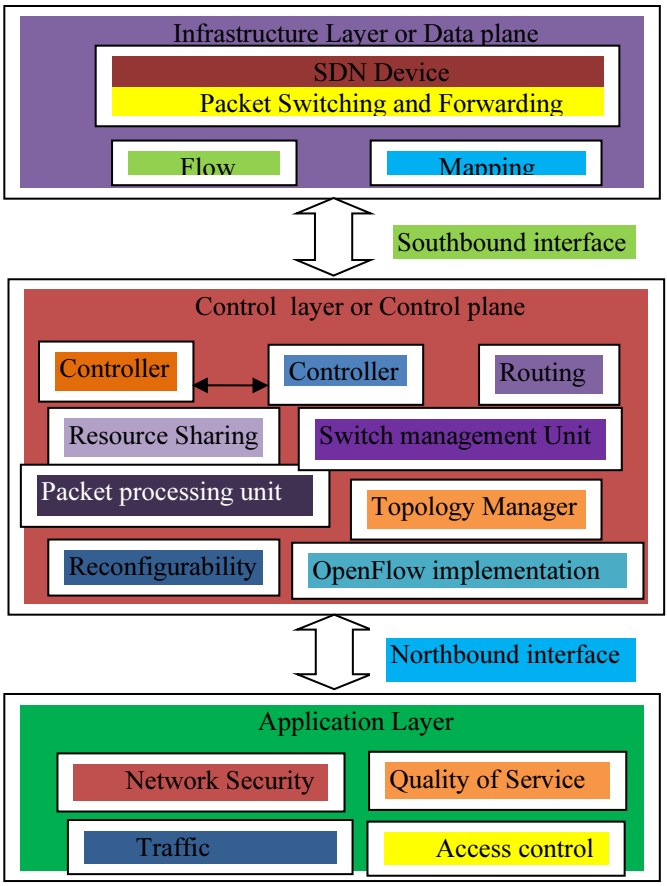

Figure 1. Architecture of SDN

- The controller aware of the necessities of the applications and resources available in the network infrastructure by using its north bound and south bound interface [3].

- Application program interface (API) open to grow a benefit from network infrastructure in terms of services in application layer, devices in infrastructure layer and programmability in control layer.

Figure 1 shows the typical architecture of SDN. It consists of three open interfaces [10]:

1. Southbound interface provide interface between control plane and data plane. It is used for creation of virtual network, dynamic reconfiguration of network, accessing resources and provide secure virtual network [3].
2. Northbound interface provide interface between control plane and application layer. It is used to provide routing related information, management related information and policy related information.
3. East west interface provide interface between controllers. It is used for intra domain; inter domain, scalability and interoperability.

3.1. SDN consists of three main layers:

3.1.1. Infrastructure Layer or data plane

The infrastructure layer responsible for transferring data between SDN devices (both physical devices and virtual devices) used to perform packet switching and forwarding. Flow table is used to store the rule populated by the controller for controlling and directing the packets [3]. Mapping table is used to store the data transfer between the different network and SDN devices communicate with control plane through south bound interface.

3.1.2. Control Layer or control plane

Control layer is placed between the infrastructure layer and application layer of the SDN network where network services are specified [10]. This layer manages with overall view of the network. Control plane is directly programmable in a centralized manner to provide hardware abstractions to SDN applications. It contains, set of controllers that interact with network services and SDN devices through north bound interface and south bound interface. Software that manages all the resources in network infrastructure is called Controller. Network infrastructure consist of packet switching and forwarding, mapping table and flow table provide a abstract view of the overall network and the controllers interact with each other through their east west bound interface to provide a stable view of the whole network infrastructure. Important functions of controller as follow [10],

- **Packet processing unit:** With respect to the protocol, processing of packet header and its payloads in the network.
- **Switch management unit:** Modification in switches and message arrival can be informed by controller.
- **Topology manager:** Maintain up to date information about network topology and changes in topology can be identified.
- **Routing:** By using forwarding table routing manager find out the routes to reach the destination address based on protocol.
- **Openflow implementation:** Controller performs the function related to openflow protocol such as action, table entry, flow rules, and message queues.

- **Management interface:** It provides access to functions that the controller provides.

SDN controller usually remains aware of all available network routes and can send packets based on different network characteristics are the important benefit.

3.1.3. Application Layer

Basically, it contains end-user applications that perform the SDN communications and network services [3]. SDN services such as access control management, network security, traffic engineering, load balancing, quality of service and other network function virtualization services. Flow of SDN devices in data plane is affected by communicating their necessities over northbound interface.

3.1.4. Advantages of SDN [4]:

- It allows a quicker response to modifying traffic (group of spoofed packets) conditions.
- It also supports more opportunity for dynamic provisioning, load balancing, monitoring specific traffic engineering, increase the utilization of network resource, and improve better occasion to implement many different types of solution.

4. Open Flow architecture

SDN is implemented using OpenFlow communications protocol which will access data from the data plane of a switch or router to control plane through the network. Communication between control plane and data plane is provided by this protocol. Packet can be moving with centralized decisions by using OpenFlow protocol [6]. Switch operations remains in control of OpenFlow controller. The action may be either Reactive or Proactive.

Reactive approach signifies that a switch will remain unaware of actions when a packet arrives. So, the packet is sent to SDN Controller [5]. By using OpenFlow protocol SDN controller responsible for inserting a flow entry into the flow table of switch. Switch totally dependent on the SDN controller is considered a major drawback.

Proactive approach overcome the drawback of reactive approach. Each entry in the flow table is pre populated by flow entries of each switch in case of Proactive approach. It does not disrupt traffic, even though if the switch loses the connectivity with control plane.

In OpenFlow architecture, we have set of OpenFlow instructions or commands that are transmitted from openflow control plane to open flow switch [7]. OpenFlow switch perform the following operations such as 1) Depends on the packet header fields identify and categorize packet from an incoming port 2) Packets can be processed in various ways by changing the header field. 3) Drop or push the packets to a respective egress port (outgoing port) or to the Openflow control plane.

Figure 2 describes the typical architecture of OpenFlow switch. OpenFlow switch consists of number of flow tables (organizing flows in table), group table (collection of action to be performed) and secure channel [7]. Flow table consist of flow entries which are forwarded, each entry match with its correspondent flow and packets, then provide the functions that are to be performed on the packets then sent. These flow entries have some set of parameters; 1) match fields used to perform for matching the

incoming packets and it uses the information there in the packet header, ingress port (incoming port), and metadata; 2) counters, used to make up the statistical data for each flow will be the count of received packets, amount and time limit for a particular flow; and certain group of commands, which apply when there is a match in table; they signify how to manage matching packets [6].

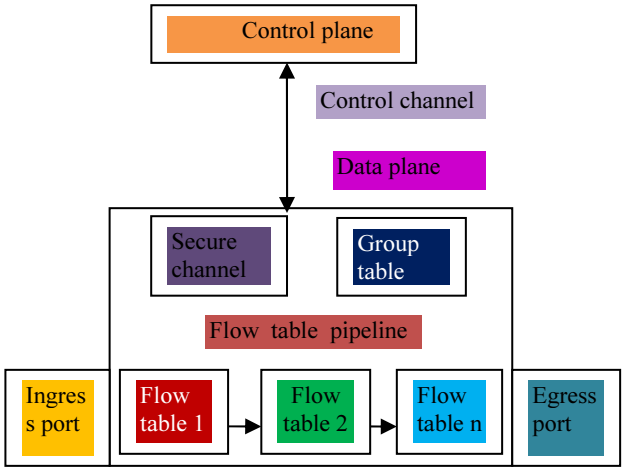

Figure 2. Architecture of OpenFlow switch

Group table consists of set of group entries. Each entry in group has certain specific semantics dependent in group type which consists of collection of action buckets. Action bucket state the action, which are performed in a group [7]. The group will select one or more bucket for each packet. So SDN controller will manage all communications by using openflow.

5. Mininet Simulator

Large number of network devices implemented through a network is very costly and difficult to implement. To reduce these problems the strategies made to for purpose of structuring and mitigating these kind of network technologies is MININET [9]. It's a free open-source software that simulates software defined network which consists of devices and controllers. Mininet was created in Python language and provide its API for user interaction and capable of emulating different network elements. SDN network can be easily virtualized and tested by using mininet [11]. It utilizes virtualization for the purpose of simulating real network by decoupling of data forwarding plane form control plane in Mininet VM.

6. Characteristics of Mininet [4]:

- **Flexibility:** Software is capable of managing modern features and newly introduced topologies, using various programming languages and variety of system software.
- **Applicability:** Without changing source codes, implementations of Network conducted in real networks.

- **Interactivity:** Simulated network should be manageable and runnable in real time network.
- **Scalability:** The prototyping environment can be scaled to larger network on only a computer.
- **Realistic:** The simulation behaviors represent real network behavior with a high degree of assurance on application of the network, so that application remains usable without any modification.

Mininet simulator is possible to create software defined network by using network programmer in a simple manner.

7. Experimental Result.

In MININET, a single command is used to create the network using linear topology with three host such as h1, h2, and h3.

$ sudo mn --topo linear, 3

8. Basic commands with Mininet [9]

Only a single console is required to control and manage entire virtual network. The basic commands such as ping, pingall, pingallfull, dump, net and iperf.

i) Ping: It uses to check the connectivity between different hosts by using ICMP protocol.
ii) Pingall: Connectivity between all hosts and tells which hosts are connected to each other
iii) Pingallfull: It gives more detail about how the hosts are connected. And tells minimum, maximum and average time between two hosts in millisecond.
iv) Dump: It is used to display the IP address and process identification of the host.
v) Net: It is used to list out links available in network between interface, host and switch.
vi) Iperf: It is used for TCP connection and to test bandwidth between hosts.

```
mininet> h2 ping -c2 h3
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=3.66 ms
64 bytes from 10.0.0.3: icmp_seq=2 ttl=64 time=0.607 ms

--- 10.0.0.3 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 0.607/2.135/3.664/1.529 ms
mininet>
```

Figure 3. Connectivity between h2 and h3

```
mininet> pingallfull
*** Ping: testing ping reachability
h1 -> h2 h3
h2 -> h1 h3
h3 -> h1 h2
*** Results:
 h1->h2: 1/1, rtt min/avg/max/mdev 2.556/2.556/2.556/0.000 ms
 h1->h3: 1/1, rtt min/avg/max/mdev 1.491/1.491/1.491/0.000 ms
 h2->h1: 1/1, rtt min/avg/max/mdev 1.190/1.190/1.190/0.000 ms
 h2->h3: 1/1, rtt min/avg/max/mdev 1.398/1.398/1.398/0.000 ms
 h3->h1: 1/1, rtt min/avg/max/mdev 1.928/1.928/1.928/0.000 ms
 h3->h2: 1/1, rtt min/avg/max/mdev 1.460/1.460/1.460/0.000 ms
mininet>
```

Figure 4. Min, Max and Average time between two hosts in ms.

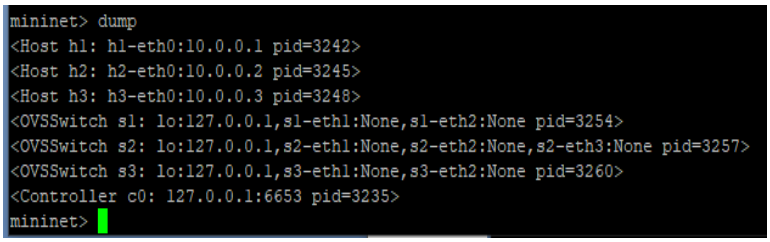

Figure 5. Display IP address and their process.

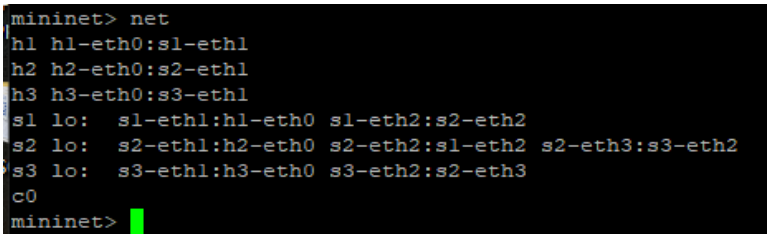

Figure 6. Links between interface, host and switch.

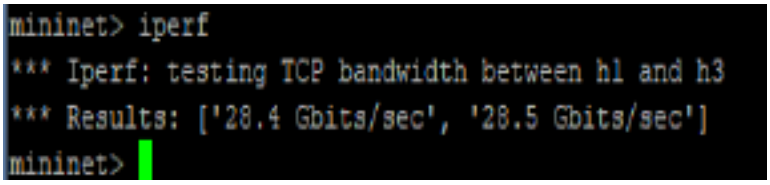

Figure 7. Testing bandwidth between hosts.

Experimental result of basic commands is shown. In Figure.3 how the connectivity between host h2 and host h3 by transferring two packets is shown. Figure 4 shows the connectivity between all host such as h1, h2, and h3 and also display the minimum, maximum and average time between all hosts in millisecond. Figure 5 tells the IP address and process id of each host and OVSswitch, controller dumps the details of all the nodes such as host list, switch list and controllers in the topology. Figure 6 display the link exists between the interface (eth0, eth1, eth2, and eth3), switch (s1, s2, s3) and host (h1, h2, h3). Figure 7 checking bandwidth between h1 and h3 using TCP and shows the result as measure of speed of the network bandwidth and helps us to check the bandwidth speed from one host to another host.

9. Security issues of SDN

Virtual network in SDN is provided to improve confidentiality, integrity and availability with security [6]. Due to attacks and vulnerabilities network performance and efficiency can be reduced which in turn affects the security properties such as confidentiality, integrity and availability. Modifying the information and inserting unnecessary codes can take place just because of an unauthorized user access.

There are various threads in SDN with respect to the different layers [8].

- **Rule's insertion:** New rules are created and implemented in different domain which causes various conflicts.

- **Malicious code:** Insertion of various malicious code leads to damage of information and corruption in data.
- **Distributed Denial of Service Attack:** Network traffics will be increased by attacks at channel, controller and switches.
- **Attacks from application:** Illegal access to the protected data about the network.
- **Man in the middle attack:** Data will be transferred to host without using any intermediate devices such as switches and router, etc. So, anyone in the middle with a connection enabled device can intercept the protected information without privacy.

Some security features of Distributed Denial of Service (DDoS) and Intrusion prevention systems are loop elimination and storm attack identification can be developed in SDN to take care of the security [8].

A spanning tree is created which automatically reconfigures the security function in loop elimination.

Storm attacks can be identified in the network by using unnecessary transmission of broadcasting spoofed packet. One of the famous detection techniques used in intelligent networks for DDoS attacks are anomaly detection, which are a subset of intrusion detection systems.

10. Conclusion

Manageability of network devices by SDN has grown beyond the expectation. This paper discusses about the overall concept of SDN such as SDN architecture and OpenFlow architecture. In addition to that analysis of SDN and detailed requirement on each standard interface has been described. During this study, design of SDN can be done using Mininet simulator. Mininet simulator acts as a useful alternative to run SDN problem cases on emulated network. Virtual machines provide an easier way to provide configuration and topology change but with real machine it is difficult. Because of decoupling data plane from control plane SDN is vulnerable to more attacks. Because of the vulnerability, performance of SDN could be rigorously affected.

References

[1] K. Benzekki, A. El Fergougui, and A. ElbelrhitiElalaoui, “Software-defined networking (SDN): A survey,” *Security and Communication Networks*, (2017), DOI: 10.1002/sec.1737

[2] Parneet Kaur, Manish Kumar &Abhinav Bhandari, “A review of detection approaches for distributed denial of service attacks”, *Systems Science & Control Engineering: An Open Access Journal*, (2017), VOL. 5, 301–320

[3] D. B. Hoang and M. Pham, "On software-defined networking and the design of SDN controllers",*6th International Conference on the Network of the Future (NOF), Montreal, QC, Canada*, (2015), pp. 1-3, doi: 10.1109/NOF.2015.7333307.

[4] F. Keti and S. Askar, "Emulation of Software Defined Networks Using Mininet in Different Simulation Environments",*6th International Conference on Intelligent Systems, Modelling and Simulation, Kuala Lumpur, Malaysia*, (2015), pp. 205-210, doi: 10.1109/ISMS.2015.46.

[5] Karamjeet Kaur, Japinder Singh and Navtej Singh Ghumman “Mininet as Software Defined Networking Testing Platform”, *International Conference on Communication, Computing & Systems* (ICCCS-2014)

[6] D. Dobrev and D. Avresky, "Comparison of SDN Controllers for Constructing Security Functions," *2019 IEEE 18th International Symposium on Network Computing and Applications (NCA)*, Cambridge, MA, USA, (2019), pp. 1-5, doi: 10.1109/NCA.2019.8935053.

[7] Sm Shamim, Shahadat Shisir, Ahosanul Hasan, Mehedi Hasan & Arafat Hossain "Performance Analysis pf Different OpenFlow based Controller Over Software Defined Networking", *Global Journal of Computer Science and Technology: C, Software & Data Engineering*, (2018),Volume 18 Issue 1 Version 1.0.

[8] Maham Iqbal, Farwa Iqbal, Fatima Mohsin, Dr.MuhammadRizwan, Dr.Fahad Ahmad "Security Issues in Software Defined Networking (SDN): Risks, Challenges and Potential Solutions", *(IJACSA) International Journal of Advanced Computer Science and Applications*, (2019),Vol. 10, No. 10.

[9] CasimerDeCusatis, Aparicio Carranza, and Jean Delgado-Caceres "Modeling Software Defined Networks using Mininet", *Proceedings of the 2nd International Conference on Computer and Information Science and Technology (CIST'16) Ottawa, Canada,* (2016) Paper No. 133.

[10] M. Shin, K. Nam and H. Kim, "Software-defined networking (SDN): A reference architecture and open APIs," *2012 International Conference on ICT Convergence (ICTC)*, Jeju, Korea (South), (2012), pp. 360-361, doi: 10.1109/ICTC.2012.6386859.

[11] Ligia Rodrigues Prete, A. A. Shinoda, C. M. Schweitzer and R. L. S. de Oliveira, "Simulation in an SDN network scenario using the POX Controller," *2014 IEEE Colombian Conference on Communications and Computing (COLCOM), Bogota, Colombia*, (2014), pp. 1-6, doi: 10.1109/ColComCon.2014.6860403.