An Approach to Secure Software Defined Network against Botnet Attack

Sreevidhya Ravindran\textsuperscript{1}, Vaishnavi Moorthy\textsuperscript{2}, Revathi Venkataraman\textsuperscript{3}

\textsuperscript{1,2,3}Department of Computer Science and Engineering, SRM Institute of Science and Technology Chennai, India

sreevidhyaravindran@srmuniv.edu.in\textsuperscript{1}, vaishnavi.m@ktr.srmuniv.ac.in\textsuperscript{2}, revathi.n@ktr.srmuniv.ac.in\textsuperscript{3}

Abstract – The traditional hierarchical networking models, although still in use by many networks of today, are inflexible and unable to manage the huge networking infrastructure. This has led to shifting to a more flexible, dynamic and virtualized network infrastructure called Software Defined Network. SDN being a dynamic network manages the traffic flow efficiently due to its OpenFlow controllers. Although SDN is the most conventional model used today, some of its drawbacks have caused disruption on a large scale. Security is one such drawback, which is of major concern in recent times. There have been multiple researches and attempts to address this issue with an aim to permanently eliminate the threat, but with the rapid technological development, an equivalent powerful trust computing protocol is required to be implemented in order to overcome it. This paper focuses on a Botnet Distributed Denial Of Service (DDOS) attack that takes place in the control plane of the Software Defined Network. This malicious attack has proved to be the most dangerous threat of the web. From flooding a website to bringing down the entire server, this ‘zombie’ network has created haphazard among reputed businesses. An attempt is made to implement a powerful secured technology that can tackle the botnet attack, protect the networks and minimize their vulnerability to such threats that can eventually bring down the ratio of cybercrime in today’s world.

Index Terms– Software Defined Networking, OpenFlow, Leaf-spine, Botnet, command and control, Distributed Denial of Service (DDOS).

1. INTRODUCTION

Software Defined Networking (SDN) is a simplified, revolutionary and more flexible approach that uses an open protocol like OpenFlow. It is of two models namely- centralized and decentralized. The common basic elements in the two models are physical devices, controllers and connecting technology. The decentralized model focuses on the control software with increased reliability, scalability and privacy of the system. It overcomes the problem of single point of failure, which is considered to be a major drawback in the centralized model. SDN ensures packet routing to only valid routes and drops the other packets. It enables security policies and services that can be customized in a dynamic way [9]. Without a single central server, this model ensures proper workload distribution. The switches and routers are deployed throughout the network. This increases the quality of performance of the network.

The Software Defined Network is a three layered distributed architecture. The three layers of SDN are Application layer, Control layer and the Infrastructure or Data layer.

\textit{Application Plane}: The Application plane, the topmost layer, manages the control logic of the Software Defined Network. In this layer, SDN facilitates routing, load balancing, coordination and automation of the network through existing Application Program Interfaces (APIs) [2].

\textit{Control Plane}: The second and the most important layer is the control plane layer. The
Southbound interface (OpenFlow), an interface between the control plane and the infrastructure plane enables easy communication between them [1]. The SDN is mainly vulnerable to DOS attacks due to the segregation between the two layers. The communication path between them could cause the attacker to spam and flood the servers with requests hence denying the service [3].

**Data Plane:** The last layer is the data plane or infrastructure plane. Also known as the forwarding plane, it comprises of the network devices that controls, processes and validates the transfer of data packets through the routers. Some of the other devices enabling the traffic flow are switches, base stations, load balancers etc. [2]

There are three interfaces in the SDN distributed architecture namely the northbound interface, southbound interface and the East-West Interface. Northbound interface permits the control plane to communicate with the application layer, southbound interface enables communication between the control plane and the data plane and the East-West acts as a communication interface between multiple controllers. BGP and OSPF are some of the distributed protocols used by the East-West interface [2].

![Distributed SDN Architecture](image)

**Fig. 1:** Distributed SDN Architecture [7]
The OpenFlow protocol provides an open communication that enables the controllers to communicate with the forwarding planes. This also enables them to make changes in the network. With greater control over the network, the companies can adapt well to the changes as per their needs. The OpenDaylight controller can be implemented both on hardware and software [16].

The OpenFlow switch protocol is one of the most widely consented and deployed open standard (Southbound) in Software Defined Network. An OpenFlow channel connects a switch and a controller. Sometimes multiple controllers handle the same switch at the same time making up a Control Channel [4]. The OpenFlow is required to address two main requirements. Firstly, in order to control the network devices like switches and routers by the SDN controller, they require a common and uniform logical switch function. Secondly, a secure protocol is to be deployed between the controllers and the physical network devices [8].

While SDN is considered to be the most convenient option, the threats posed to SDN are of major concern. The Botnet Distributed Denial of Service (DDOS) attack is one such major threat. This paper focuses on a leaf-spine topology by analysing its performance in terms of bandwidth, latency and throughput. A DDOS attack is generated and the change in parameters are recorded and compared. In this decentralized model, an attempt is made to investigate the possibility of attenuating the attack in the most efficient way possible to enhance Quality of Service (QoS).

The further organization of the paper is as follows: Section II details about custom Leaf-Spine topology implementation and its working, Section III introduces about botnets, its detailed architecture and working, Section IV describes the attack generation and implementation and Section V concludes the paper.
2. TOPOLOGY

A number of defaults topologies exist that can be implemented to design the network. In this research project a custom topology is created which is of the leaf-spine architecture model. The architecture consists of an access layer that is formed by the leaf switches. The sequence of leaf switches is completely meshed to the sequence of spine switches. This ensures that the leaf switches are at a distance of only one hop from each other. This helps to minimise the latency and avoid bottleneck attack on the network [10].

This topology is chosen since it proves to be a solution for the limitations of Spanning Tree. It is able to achieve a dynamic network and utilize other networking protocols. The routes are configured using Equal Cost Multipath (ECMP). ECMP enables traffic of same session or flow to be transmitted to multiple paths at equal costs. It also helps in load balancing and increasing bandwidth [11] [12]. An added advantage of this topology is that it is easy to add additional hardware and capacity. It can increase the capacity in case of oversubscription of links.

The customized leaf-spine topology used in this project as shown in Figure-3 consists of 10 nodes, 9 switches and a remote controller. The packets are sent from one host to another and the resulting transmission is analyzed using appropriate network analyzer tool. The packet transmission is similar to the controls in Software Defined Network. The OpenDaylight application is used in order to control the transmissions. After setting up the topology, commands are given to run the controller. Respective IP addresses are assigned to the 10 hosts created in this topology. The packet forwarding depends on the applications running on top of OpenDaylight. After successful pings, under all conditions, the network is analyzed using the Wireshark tool.

The OpenFlow packet transfer- PACKET-IN and PACKET-OUT is analyzed using Wireshark. Wireshark captures the OpenFlow traffic that uses TCP as its transport protocol. The default TCP ports are 6633 and 6653. After capturing the OpenFlow packets, the values generated are tabulated. From the tabulated values, the throughput, latency and bandwidth are calculated.

![Fig. 3: Custom Leaf-Spine Topology](image-url)
The bandwidth is the amount of data that is carried from one node to another in a given period of time i.e. cumulative bytes per relative time. The latency specifies the time it takes for data (in bits) to travel across the network from one node to the next node i.e. packet length (in bits) by the link bandwidth (bps). The throughput measures the number of units of information a system processes in a given amount of time. It is basically the average of all bandwidths between two nodes.

**Fig. 4:** Graph showing throughput before botnet attack generation.

Figures 4 and 5 shows the graph output for the given topology in terms of throughput and round trip time respectively. It is observed that at a certain time the average throughput increases with the increase in sequence length. At around sequence length value of 45 the average throughput remains constant. The round Trip Time decreases with increase in time as observed in Figure 5.

**Fig. 5:** Round Trip Time Graph

Figure 6 depicts the sequence numbers versus time graph before the attack is generated.
3. BOTNET
In this section, firstly we introduce about botnets. We further explain its detailed architectural models and their working. As aforementioned, this paper focuses on botnet attacks. A botnet attack is defined as a large swarm of computers that are connected to perform a number of repetitive tasks requested by the cybercriminals. Its root words are “robot” and “network”. In a botnet attack, an attacker attacks in the following ways:

1) The attacker acquires a virus and disguises in an attractive name and combines into a Trojan horse.
2) Next the attacker uploads the Trojan to various P2P networks.
3) An Internet user notices this freeware that seems to be attractive in a peer to peer.
4) The user downloads the freeware in his PC.
5) The installation of the freeware fails leaving behind the virus in the PC.
6) Now the virus communicates with the attacker through the botnet command center (IRC servers).
7) The attacker instructs the virus to install necessary updates and wait for further instructions.
8) The attacker never communicates with the client directly. It does so through the IRC server and eventually attacks the user's E-commerce server.

4. ARCHITECTURE
In order to understand the architecture of botnet, we classify them on the basis of their C&C traffic protocol [5].

a) IRC Botnets: A chatting environment is designed using Internet Relay Chat [5]. It has

![Fig. 6: Sequence Numbers VS Time Graph](image)
a distributed structure and its properties like versatility, scalability and redundancy attracts an attacker. The bot receives the command from the bot master (controller) to initiate an attack via the IRC [5]. IRC has a centralized server [6] and once detected the whole botnet is disabled by shutting down the server [5].

b) **HTTP Botnets**: In this architecture the botnet members do polling to the HTTP server to get new commands and the servers distribute bot commands. It is a centralized botnet [6], following the C&C protocol.

c) **POP3 Botnets**: It is an email based botnet C&C. The bot in POP3 Botnet architecture takes control of a mail server and receives the emails through the channel that can include some confidential attachments and respond back through the same channel [5]. This botnet topology is centralized.

d) **Peer-to-Peer (P2P) Botnets**: This Botnet topology has a decentralized architecture based on Peer-to-Peer protocol [5]. In this paper we take the P2P Botnet attack into consideration since we are working in a decentralized environment (SDN). P2P botnets cannot redirect the commands directly to the bots. It has a set of commands and all bots subscribe to this set [5].

e) **Hybrid Botnet**: It is a union of multiple botnet models such as IRC, P2P and HTTP botnets. It has both centralized and decentralized structures [6].

5. **WORKING**

Before a botnet comes into picture, the devices are first infected with malware. Once infected, these devices become bots and communicate with the command and control servers (C&C). The Command and Control server directs and instructs the bot. It can either be single or multiple. Multiple servers can help attain higher availability in case anyone bot goes down. Once the C&C is down, the entire botnet is down [15].

The communication between the bots and the Command and Control server takes place through two main protocols - the Hyper Text Transfer Protocol (HTTP) and the Internet Relay Chat
8

IRC [15]. Distributed Denial of Service (DDOS) attack is one of the most predominant botnet attacks. In this attack, a target server is flooded with requests hence denying the legitimate request to get through for processing.

6. IMPLEMENTATION

After understanding the various botnet models and how they work, the implementation of the attack on the desired topology in order to observe the changes in the parameter values are elaborated. In this paper, the Peer-to-Peer (P2P) botnet is taken into consideration. A short description is given on TCP/IP connection establishment in this section.

In order to generate an attack on the working topology, the TCP/IP connection establishment is analyzed. This can facilitate to find the most appropriate technique to be implemented. The three-way-handshake helps to establish the successful connection. A client-server scenario can be considered as an example as shown in Figure 8 below.

**Fig. 8: TCP/IP Three Way Handshake [14]**

A three-way handshake between a client and server can be described as follows:

1) The client first waits for the server. When the server is active open for connections, the client directs the SYN data packet to the server. This is sent either on the same network or different network.

2) The server can establish new connection in case of open ports. After the server listens to the client, it responds to the client with an acknowledgement soon after it receives the SYN data packet. It sends the ACK or SYN/ACK data packet to the client.

3) Once the client receives the ACK or SYN/ACK data packet from the server as an acknowledgement to its request, the client responds back to the server with an ACK packet.
The DDOS attack is now generated using a Penetration Testing Tool named Hping3. Once the Hping commands are executed, the results directly reflect in Wireshark. The packets are captured now using Wireshark. In this case, the captured packets are dissimilar to those captured previously before the attack execution. With no acknowledgment in the handshake, successful attack has been generated. The values are tabulated and the three parameters – bandwidth, latency and throughput are yet again recorded. These values are compared to the initial values and the changes in the parameters are observed. The graph generated shows that the average throughput is zero when the attack is generated.

![Graph showing throughput after botnet attack generation.](image)

**Fig. 9:** Graph showing throughput after botnet attack generation.

7. CONCLUSION

This paper aims to explain the detailed architecture of Software Defined Network. The importance of OpenFlow controllers for the efficient traffic flow that being the most important advantage of using SDN has been elaborated. Some of the botnet models and their working including the C&C server have been observed for better clarification of concepts in order to choose an appropriate attack technique. The TCP/IP three-way handshake concept is explained for a better understanding. The various parameters such as bandwidth, throughput and latency are recorded and compared before and after the attack. The values are tabulated and the statistical data have been recorded.

In modern day, security has become a subject of greater concerns due to the rapid advancement in technology. Some live instances prove the amount of loss faced by well-recognized organizations due to occurrence of such rudimentary attacks. While an attack can cause substantial loss, the recovery period has proved to be even more extremely challenging in terms of time and money. Some well-established companies have faced reputational challenges and
Took years to recover from it. Hence, with this rapid development, the ratio of cybercrimes has also increased exponentially. This paper tends to approach towards the implementation of a strong secured technology in order to protect the control plane of the Software Defined Network against DDOS attack. The aim is to enhance the security and minimize the vulnerability of the network to such attacks. The results so far obtained will help to choose an appropriate trust computing protocol that can be implemented to almost avoid the threats posed to the network. The expected result shall minimize the probability of attack by the hackers (botnets) and ensure higher level of secured transfer of data from the control plane of the software defined network architecture.

8. REFERENCES

[1]. Benamrane, F., & Benaini, R. (2017). An East-West interface for distributed SDN control plane: Implementation and evaluation. Computers & Electrical Engineering, 57, 162-175.

[2]. Blial, O., Ben Mamoun, M., & Benaini, R. (2016). An overview on SDN architectures with multiple controllers. Journal of Computer Networks and Communications, 2016.

[3]. Hadianto, R., & Purboyo, T. W. (2018). A Survey Paper on Botnet Attacks and Defenses in Software Defined Networking. International Journal of Applied Engineering Research, 13(1), 483-489.

[4]. Li, Y., Zhang, D., Taheri, J., & Li, K. SDN components and OpenFlow.

[5]. Mahmoud, M., Nir, M., & Matrawy, A. (2015). A Survey on Botnet Architectures, Detection and Defences. IJ Network Security, 17(3), 264-281.

[6]. Seenivasan, D., & Shanthi, K. (2014). Categories of botnet: a survey. Int. J. Comput, Control Quantum Inf. Wng, 8(9), 1589-1592.

[7]. Benamrane, F., Mamoun, M. B., & Benaini, R. (2015). Performances of OpenFlow-based software-defined networks: an overview. Journal of Networks, 10(6), 329-338.

[8]. Stallings, W. (2013, March 1). Retrieved from https://www.cisco.com/c/en/us/about/internet-protocol-journal/back-issues/table-contents-59/161-sdn.html

[9]. DI, (2016, November 28). Retrieved from https://defintel.com/blog/index.php/2016/11/how-does-a-botnet-attack- work.html

[10]. SDX Central, Retrieved from SDX Central, Retrieved from https://www.sdxcentral.com/networking/sdn/definitions/opendaylight-controller/

[11]. Rodrigues, B., Bocek, T., Lareida, A., Hausheer, D., Rafati, S., & Stiller, B. (2017, July). A Blockchain-Based Architecture for Collaborative DDoS Mitigation with Smart Contracts. In IFIP International Conference on Autonomous Infrastructure, Management and Security (pp. 16-29). Springer, Cham.

[12]. Banks, E. (2013, November) Retrieved from https://searchdatatcenter.techtarget.com/feature/Data-center-network-design-moves-from-tree-to-leaf

[13]. Husseman, T. (2015, March 23) Retrieved from https://blog.westmonroepartners.com/a-beginners-guide-to-understanding-the-leaf-spine-network-topology/

[14]. Juniper Networks (2018, September 3) Retrieved from https://www.juniper.net/documentation/en US/junos/topics/concept/routing-policy-security-
ecmp-flow-based-forwarding-understanding.html#

[15]. Cisco Security Portal Retrieved from https://www.cisco.com/c/en/us/about/security-center/guide-ddos-defense.html

[16]. Kozierok, C.M. (2005, September 20). TCP/IP Guide, Retrieved from http://www.tcpipguide.com/free/tTCPConnectionEstablishmentProcessTheThreeWayHandsh-3.htm

[17]. NHK K. ISMAIL*, "Estimation Of Reliability Of D Flip-Flops Using Mc Analysis”, Journal of VLSI Circuits And Systems 1 (01), 10-12,2019.

[18]. MN BORHAN,”Design Of The High Speed And Reliable Source Coupled Logic Multiplexer”, Journal of VLSI Circuits And Systems 1 (01), 18-22,2019