Scientific Inquiry and Review (SIR)
Volume 3, Issue 1, January 2019
ISSN (P): 2521-2427, ISSN (E): 2521-2435
Journal DOI: https://doi.org/10.32350/sir
Issue DOI: https://doi.org/10.32350/sir.31
Homepage: https://ssc.umt.edu.pk/sir/Home.aspx
Journal QR Code:

Article

DDoS Hazard and Precautionary Measures in Software Define Network

Author(s)

Tanzeel Sultan Rana

Online Published

January 2019

Article DOI

https://doi.org/10.32350/sir.31.04

Article QR Code

Tanzeel S. Rana

To cite this article

Rana TS. DDoS hazard and precautionary measures in software define network. *Sci Inquiry Rev*. 2019;3(1):33–42.
Crossref

Indexing Agency

Crossref
ULRICH'S PERIODICALS DIRECTORY
OCLC WorldCat
GENAMICS ...research from your desktop Genamics JournalSeek
ROAD DIRECTORY OF OPEN ACCESS SCHOLARLY RESOURCES
Academic Resource Index ResearchBib
J-Gate

A publication of the
School of Science, University of Management and Technology
Lahore, Pakistan.

DDoS Hazard and Precautionary Measures in Software Define Network

Tanzeel Sultan Rana

School of System and Technology
University of Management and Technology, Lahore, Pakistan

tanzeelsultanrana1@gmail.com

Abstract

Software Defined Network is an emerging technology which is flourishing due to its diversity and by virtue of the fact that there are decoupled planes in this architecture which have some benefits as well as drawbacks, such as the execution of cyber attacks are easy at northbound and southbound interfaces and DDoS attack can easily be manipulated in this architecture. It has been identified that DDoS attack can be countered at northbound API so that appropriate decision about illegitimate traffic can be taken. Java has provided us with a very reliable support for three decades. Hence, all controls are governed by programming interfaces in this architecture with the help of this feature and according to the entropy of information which allows us to track the traffic and compare it with the threshold to identify the malware in the network. Floodlight controller is used in this paper to accommodate the illegitimate traffic. This paper allows the programmers to program such applications in Python or Java based on the basic mechanism of entropy which uses a threshold value from which DDoS attack can be countered, as we are well aware that a large number of systems are involved in producing illegitimate traffic on a network which creates distraction for the legitimate traffic.

Keywords: Software Defined Network (SDN), Distributed Denial of Service (DDOS), Floodlight controller, open flow, DDoS attack, entropy

1. Introduction

Computer networks have been major means of communication for many decades. It is important to have advancements in computer networks and topologies with the passage of time to overcome the complexity of the network. Traditional network devices were made with the concept of individual self-base service, such as L3 devices take an entire bunch (control as well as the forwarding plane for the data) after the route is found by routing protocols such as OSPF and BGP. At present, with the advent of the concept of cloud and fog computing, there must be an architecture that reflects the issues which will arise within the next five to ten years. Big data is also a concept that is making a great impact in

the field of computer networks because it shows that data flow between the data center and client has been caught at a great density. Today, data is independent of a single controller as all devices in a network have their own controller. So, the attacks can be easily manipulated in a network and it may be highly affected by STRIDE attack [1] .

Therefore, to overcome the expected problems regarding a large network, the researcher proposed a model known as Software Defined Network (SDN). It is a model that decouples the network planes in such a way that it provides feasibility to the flexible mechanism of forwarding data packets in the network [2]. This is a software based architecture, as we know that traditional networks are vendor specific. It is difficult to work with multiple vendors in an environment in an enterprise and STRIDE can affect that architecture due to integrity between these devices. SDN is a centralized, open vendor, flow based means which is not IP based routing. In this regard, all flows are instructed individually by the controller for each of the switch in the data plane. It provides programmable interfaces to make it more attractive, since android platform is open source and provides a wide range of applications.

SDN is a concept in which V-Switch will be used. Due to a large number of programmable interfaces, there must be security issues unlike all the threats in the traditional network but major issues that may result in DDoS (Distributed Denial of Service) attack [3]. The motive behind this is to separate planes accordingly while interfaces collaborate with each other, hence it will be easy to penetrate some sort of instruction on them. A major issue possibly faced in SDN will be software vulnerabilities, since it will not be similar to the traditional OS, that is, on L3 and L2 devices. Therefore, penetration in such an environment will be easy. SDN uses languages define which belong to low level programming languages and are near to the machine and far from human understanding. In fact, the structure of SDN provides the interfaces that enhance the portability and modification of network elements like applications and services, which are a much needed factor in the network zone in a campus or an enterprise network [4] .

It is essential to have diverse knowledge about the languages that will be used in SDN architecture to control the interfaces and other core functionalities defined through the application that are deployed at the most upper layer to control the lower layers. Recommended languages, which are most likely to be applicable in SDN, are Python and Java. Most of the researchers and bloggers recommend that Python has the potential that may meet the feasibility of work in this architecture. The growth of Python is tremendously increasing day by day, thereupon, it will be used

in SDN. Most of the present attacks are driven by Python based tools. DDoS attacks are also driven by Python.

2. Literature Review

Distributed Denial of Service (DDoS) is not as other DoS attacks; rather, it is distinct because in case of Distributed DoS there are more than one systems resource and they also have multiple internet connections. On that account, they can manipulate a host and hence it will be very difficult to cater them when the attack has been executed. The first DDoS was executed in 1999 and afterwards a new era of such attacks began [6]. In this type of attacks, it is very normal for the network traffic to flow in its right direction; as far as we know DDoS attacks are just like the addition of abnormal traffic, which disperses or distracts the legitimate traffic due to a virtual IP [5]. Different tools were developed using C language as most of Microsoft operating systems were developed in this language, so it was easy to restrict a legitimate user from its source.

In DDoS, there are two phases in which this attack can be manipulated. In the first phase, the main function is to locate the legitimate hosts of a network so we can cater them easily but it is inefficient to locate these systems. A number of systems are required to generate a large number of data or traffic packets. When the attacker gets hold of the system, DDoS tool is installed on them with paraphernalia like rootkit. Rootkit provides a bench to the attacker and DDoS attack remains invisible to the legitimate host. In actuality, this scenario is secondary to DDoS because it is the source of traffic packets to targeted sites. The generation of traffic packets belongs to the second phase. It is very difficult for the network handler to accommodate this artificial traffic in the network for the reason that congestion will occur and will cause the denial of service [7]. It is very effective in case of SDN, since control plane is dependent on the application and the application will be easily captured using some sort of code.

Intellectual attackers have keen interest in SDN because this approach provides them a new platform to execute their attacks; they are often young and energetic people who want to show their capabilities to the audience. Attackers mostly use the spoofed IP, so it is difficult to trace them [8]. Indeed, SDN provides centralized control to the network but it is also a single point of failure in case of DDoS attack, so some precautionary measures must be followed.

The prime target is to achieve the high availability in case of SDN owing to the fact that the bombarding of legitimate and spoofed packets makes the controller unavailable in case of this architecture. The recommended architecture contains eastbound and westbound interfaces

for the backup controller but when the primary controller gets bogged down due to a large number of spoofed packets, it is obvious that the backup controller also gets bogged down in the absence of the primary controller. It is to rectify that a controller should be available maintaining all the three properties of a security architecture, that is, confidentiality, integrity and availability. Since this architecture provides portability against configuration through controller; therefore, it will be very helpful to use an alarming command to accompany the malicious activities in the architecture [9] .

We cannot deal DDoS attacks in SDN in the same vein as the traditional production networks because there is a difference in their basic layouts. In all related work, DDoS attack patterns are dealt much like the traditional network which is unbearable. Network switches in SDN on a similar note as ordinary hub means that they are non-intelligent devices, so once the attacker gets the controller it is very difficult to accommodate it again, shortly. DDoS contain four activities which are attack prevention, attack detection, attack source identification and attack reaction [10]. Open Flow protocol in SDN architecture is the backbone of the procedures and flow table are made on behalf of the instructions proceeded by this protocol [11]. If this protocol will be held by DDoS then it will be a very exhaustive situation because spoofed IP base table will be created; the administrator will not be aware of this fact and hence the data packets will go to the black hole [12]. Network operating system provides an upper layer on which network application can be deployed, such as NOX is a platform which is open source and based on C++ language which is easy to operate. SDN based applications are developed in it. NOX decides what to do with packets which are received from the executor [13] .

In the current era with enterprise networking, there is a need of few amendments in service provider networks, ranging from small networks to large specified cloud networks. Hence, there will be a need of security essentials as well to facilitate such a tremendous number of hosts. We can cater the problem of DDoS attacks somehow by analyzing the flow table but it will not be highly effective in case of huge numbers of packets [14] . It is observed that a method to control the packet flow can be used in the form of SDN-Guard, which will be further divided into three parts performing different operations to stop the malicious DDoS attacks and packets bombarding; these parts include flow management module, rule aggregation module and monitoring module [15]. In SDN architecture, there is also a possibility of DDoS attack using scanning of the network resources and this scanning may exploit all the planes of SDN.

3. Research Methodology

Ultimately, the above study shows that it is the feasibility in this architecture that DDoS attack can be driven. Therefore, it is essential to have a mechanism which provides a firewall against them. An attacker must not easily encounter the whole infrastructure of SDN and enjoy access to the network. For this purpose a method must be deployed which may act as a precautionary measure against the attacks. Entropy is the method which was discovered many years ago and remains very useful while working with the threshold defined value[16]. Open Flow is the primary API which is associated with southbound interface of the controller; however, this protocol is responsible for the flow of traffic between the switches in the data plane. Moreover, all instructions are governed by the controller to the switch including which packet has to be sent and where. With Open Flow API, centralized control enables the programmatically controlled network infrastructure.

Consequently, with the addition of Open Flow routing, decisions are made by the controller and if a DDoS attack is driven then with the help of Open Flow it is easily countered. It is being identified if a mechanism is defined for time based entropy on the application layer. Furthermore, threshold will be responsible for the suspicious activity as it has the values which are convened by the entropy based methods. If an illegitimate inhabitant wants to exploit the architecture using the uncertain traffic then this could be measurable using this method. Integrity, confidentiality and availability are significant while data packets are travelling in any of the architecture to assure that there will be no failure. Moreover, SDN provides many edges over the traditional network but it also has a flaw. The mechanism behind the packet transmission resembles the master slave relation. So, when a switch receives new packet which is unknown to them, the whole packet or its essential address parts will be transferred to the controller who will make new rules for that packet. Due to this DDoS can easily be driven.

In entropy, it is very difficult to detect DDoS attacks with low traffic pattern flows. However, it can be upheld while using this technique with time based intimations methodology. Otherwise, architecture would bring a defunct from resources and no one will be responsible for this fact, since the precautionary measures were not taken. Foremost bombarding of packets is essentially the caveat for the administrator.

4. Proposed System

REST (Representational State Transfer) is a northbound API which is responsible for the control of controller through the application developed by the developer to handle the network. Open Flow is deployed at

southbound to control the pattern of forwarding the packet in the forwarding plane. Floodlight controllers are based on the northbound pattern which are defined by the community, since the natives propose work and a lot of work is being carried out. Floodlight is compatible with both physical as well as with virtual Open Flow compatible switch, which means that it is also compatible with the traditional non-Open Flow switches. Floodlight provides the feasibility to accommodate the architecture using application written in Java. Accordingly, it has been suggested that if an application is written in a way that follows the mechanism of flows of packet through the Open Flow in the network that the threshold will be recorded. Hence, if an intruder wants to enter illegitimate traffic in the network, it will be countered by comparing with the threshold. REST will be responsible for the essential part of supposed system because this API will be responsible for the manipulation of the application developed for the recognition of legitimate and illegitimate traffic. So, the distraction of network traffic can easily be countered. Optimization and verification of application traffic falls under the realm of Application Delivery Controllers (ADC). Thus, ADC through REST API will be responsible for the mechanism proposed [17].

Java is a high level language which has remained highly compatible with the hardware since 1991, so it is used as IDE language that is, used in Floodlight controller based application. Hence, we can easily introduce the entropy of information technology method to implement in this SDN technology. It is evident that Floodlight controller also has the edge in compatibility over OpenStack, which is a tool based on software helpful in building and managing the cloud infrastructure of both types, that is, public and private clouds. As compared to Floodlight, OpenStack will be helpful in triggering the DDoS attack as cloud infrastructure may increase the complexity in SDN infrastructure.

5. Results and Evaluations

Consequently, with this approach a scalable solution for DDoS is being proposed and the software based solution is a good way to handle the diversity found in the nature of STRIDE attack, since DDoS are very effective in case of SDN. Number of packets per flow are recorded. Hence, random values will be recorded and will intimate the administrator that the intruder wants to take charge of the network. Hence, this technique will help us to cater the issue and also strengthen the APIs of this infrastructure. Our required result will be achieved with high probability towards success regarding the precautionary measures against the DDoS attacks. Experiments are performed to check the numbers of packets that can be generated by the DDoS attackers and we can use negative log loss function to reduce the probability of attacks

while making the graph of our legitimate traffic. Reference tools will be used in the reference [18]. A number of systems will be required to produce such a great traffic which will be used in simulation, so that on the basis of the following table 1 we can calculate the number of packets that can be generated over the specific bandwidth.

Table 1. DDoS Attacks on Protocol/TCP Architecture

Variable Inputs				Results	
Intensity	Scale	Regions	Simultaneous Attacks Per Test	Calculated Bandwidth	Calculated Packet Per Second
1	20	Asia	4	120Mb/s	16000
5	6	Asia	5	30,000Mb/s	4500000
3	10	Asia	2	2,000Mb/s	300000
4	18	Asia	4	57,600Mb/s	864000

6. Conclusion and Future Work

SDN is the technology of current era and as a new technology it has some drawbacks which will be rectified with the passage of time. It is identified that DDoS attack can be manipulated easily due to the infrastructure and requirements that a methodology should adopt to cater the malware. So, Floodlight may define the precautionary measures to the controller that will accommodate the whole mechanism. With the passage of time, Floodlight may provide feasibility to other attacks like spoofing, tempering, repudiation, information disclosure, elevation of privileges and eavesdropping. We will use stochastic approach (selected batches from the entire traffic) while making graph against the legitimate traffic. When there is illegitimate traffic, our graph will show the distraction from the normal operation and this will be calculated using negative log loss function in the application.

$$L(\check{Y},Y) = - [Y (\log \check{Y}) + (1-Y) \log (1- \check{Y})$$

Y is the actual traffic while $\check{Y}$ is the traffic generated by the DDoS attacker. By using this function distinction between them is easy.

References

[1] Levin D, Canini M, Schmid S, Feldmann A. Incremental SDN deployment in enterprise networks. *ACM SIGCOMM Comput Commun Rev.* 2013;43(4): 473–474.

[2] Alsmadi I, Xu D. Security of software defined networks: A survey. *Comput Secur.* 2015;(53): 79–108.

[3] Jantila S, Chaipah K. A security analysis of a hybrid mechanism to defend DDoS attacks in SDN. *Procedia Comput Sci.* 2016;86: 437–440.

[4] Trois C, Del Fabro MD, de Bona LCE, Martinello M. A survey on SDN programming languages: Toward a taxonomy. *IEEE Commun Surv Tutorials.* 2016;18(4): 2687–2712.

[5] Azaria J, Nakar O, Kogan E. *The world's most popular coding language happens to be most hackers' weapon of choice*. Available from: https://www.imperva.com/blog/the-worlds-most-popular-coding-language-happens-to-be-most-hackers-weapon-of-choice/

[6] Criscuolo PJ. *Distributed denial of service: Trin00, tribe flood network, tribe flood network 2000, and stacheldraht CIAC-2319*. Available from: https://apps.dtic.mil/docs/citations/ADA396999

[7] Shin S, Gu G. Attacking software-defined networks: A first feasibility study. Paper presented at: *Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking*; August 16, 2013; Hong Kong, China.

[8] Zargar ST, Joshi J, Tipper D. A survey of defense mechanisms against distributed denial of service (DDoS) flooding attacks. *IEEE Commun Surv Tutorials*. 2013;15(4): 2046–2069.

[9] Mousavi SM, St-Hilaire M. Early detection of DDoS attacks against SDN controllers. Paper presented at: *2015 International Conference on Computing, Networking and Communications (ICNC-2015);* February 16-19, 2015; Garden Grove, CA, USA.

[10] Peng T, Leckie C, Ramamohanarao K. Survey of network-based defense mechanisms countering the DoS and DDoS problems. *ACM Comput Surv.* 2007;39(1): 3.

[11] Kloti R, Kotronis V, Smith P. Openflow: A security analysis. Paper presented at: *21st IEEE International Conference on Network Protocols (ICNP);* October 1-6, 2013; Gottingen, Germany.

[12] Ma X, Chen Y. DDoS detection method based on chaos analysis of network traffic entropy. *IEEE Commun Lett.* 2014;18(1): 114–117.

[13] Braga R, Mota E, Passito A. Lightweight DDoS flooding attack detection using NOX/OpenFlow. Paper presented at: *IEEE 35th Conference on Local Computer Networks (LCN)*; October 10, 2010; Denver, Colorado.

[14] Dao NN, Park J, Park M, Cho S. A feasible method to combat against DDoS attack in SDN network. Paper presented at: *2015 International Conference on Information Networking (ICOIN);* January 12-14, 2015; Siem Reap, Combodia.

[15] Dridi L, Zhani MF. SDN-guard: Dos attacks mitigation in SDN networks. Paper presented at: *5th IEEE International Conference on Cloud Networking (Cloudnet);* October 3-5, 2016; Pisa, Italy.

[16] Zubaydi HD, Anbar M, Wey CY. Review on detection techniques against ddos attacks on a software-defined networking controller. Paper presented at: *Palestinian International Conference on Information and Communication Technology (PICICT);* May 8-9, 2017; Gaza Strip, Palestine.

[17] Zeus K. *WANSPEAK – What's the role of the application delivery controller in a software-defined network*. 2019. Available from: https://blog.silver-peak.com/whats-the-role-of-the-adc-in-a-sdn [Accessed 11th April 2019].

[18] *DOSarrest CAPP bandwidth calculator*. 2019. Available from: https://www.dosarrest.com/site/capp-bandwidth-calculator/ [Accessed 11th April 2019].