ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 1, January 2017

A Survey on Detection of Attacks in Software Defined Networks

Omkar Chippalkatti[1], Assistant Prof.S.U.Nimbhorkar[2]

M.Tech Student, Department of CSE, G.H. Raisoni COE, Nagpur, India[1]

Assistant Professor, Department of CSE, G.H. Raisoni COE, Nagpur, India [2]

ABSTRACT: Software-defined networking (SDN) is a network architecture separates the control panel and the data panel, moving the control panel to an application called Controller. Software defined networking (SDN) defines latest approach at how networks are configured, controlled, and operated. An SDN controller is the centralized repository of policy and control instructions for the network or application infrastructure. Without proper security, the data over the network is prone to attacks and should be secured. Using SDN policies we can configure SDN controllers so as to detect attacks such as DDOS which can cause congestion in the network. Using the policies, we can decide whether to forward or drop the packet when the attack is detected.

KEYWORDS:SDN, DDOS, Network Architecture, Network Attack,IDS.

I. Introduction

Due to the arrival of software-defined networks (SDN), it has become very important to keep an eye on network operations. In SDN, controller is responsible to handle the complexity of network which in return provides abstraction to network administrator. We do not need to configure each device separately; in fact, we rely on automated implementation of network policies and rules in the controller. SDN divides the control functionality from the data functionality and shifts the former to a logical centralized software-based network controller.

Open Flow is a standard protocol that decides how an SDN controller makes contact with the network devices. An Open Flow based switch makes naked to the controller, and provides an abstraction of its flow table and gives permissions to the controller to change values by adding, customizing or removing rules from the table. Using Open Flow, applications running on the network controller can easily control traffic of incoming and outgoing packets in switches.

Network operator can implement security rules and policies as per their own need through frameworks based on Open Flow protocol. Suppose a campus operator needs to check incoming web traffic to an intrusion detection system (IDS) and e-mail traffic to a spyware detection device. Our goal is to leverage SDN to allow the operator to write a high-level policy to achieve this, instead of having to manually configure each device. Furthermore, suppose the IDS detects malicious traffic and the sender needs to be blocked from accessing the network. Instead of having the operator configure the edge router to manually disable access to the source, we are interested in blocking the sender automatically. Abstraction in the network helps the operators to concentrate on specifying easy rules and policies, rather than separately configuring each and every device in the network. Framework contains a software layer that runs on top of the network controller, as well as various external devices which perform security based services such as firewall, intrusion detection system (IDS) etc. reports to the controller. The actual aim of framework is to grant network operators to write security rules and policies for desired flows. The rules contain a description of the flow, a list of security services that apply to the flow and its reaction in case of detection of malicious data. This reaction may be a warning only, or to restrict the traffic otherwise stop flow of all packets from a specific source.

DOI: 10.15680/IJIRCCE.2017. 0501022

II. LITERATURE REVIEW

A.Lara and B. Ramamurthy [1],It describes the importance of network management to be handled with the invention of Software defined Networks and OpenFlow which is based on policies. Software based applications which runs on top layer of the network controller abstracts the topology and supports the task of operating the network with the help of Controller. Author claimed OpenSec, a framework based on OpenFlow protocol which lets a network security operator to design and execute security rules implemented in simple and readable language. OpenSec, allows user to define a flow in based on OpenFlow matching fields, define various security services to be applied to that flow and level of security that define how OpenSec will react with detection of malicious traffic. Evaluation of accuracy, elasticity and correctness of the OpenSec framework was done in GENI testbed.

A. Lara, A. Kolasani, and B. Ramamurthy [2], Software Defined Networks is presently based on OpenFlow protocol. As compared with traditional networks the control plane and the data plane are separated in the Software Defined networking. A Controller which is based on software is having the power for handling the communication between the switches whereas the hardware is having the power only to forward the packets according to the rules and policies applied by the controller. OpenFlow standard SDN technology which decides how the communication takes place between the controller and other devise SDN architecture. Researchers were allowed to test new inventions in a production environment. Implementation of Controlling logic of switch into the controller is provided by the OpenFlow.

Xing et al. [3] describes of implementing Openflow to SNORT. It is a common intrusion detection tool for large networks, which will help to reconfigure the network on detection of attack through SNORT. The above architecture was not implemented to detect DDOS attacks in the cloud but capability of SDN for intrusion detection to reduce severity in a complicated network such as a cloud network. The important goal of this research is to find a way to protect the architecture of SDN, in particularly, the controller.

Fonseca et al. [4] discussed the possibility of retaining the controller and may need another controller for a backup. The paper proposed an extra controller which runs simultaneously with the present controller. The second controller is added in the switches so that in case of communication loss with the current controller they can look for the backup controller. This will help in case of DDOS attack on the controller which may exhaust its resources and the controller may shutdown. The answer for situations like these is making use of machine learning like SOM which starts recovery during attack.

Qin et al. [5] proposed a Entropy based detection method with a window sized of 0.1 seconds and threshold with three levels. This method is related with aim of avoiding false positive and false negatives in the network. Moreover, as the authors themselves declared that the method is time consuming and makes use of more resources.

Ra et al. [6] tells a faster means of computing entropy by referring the calculation on both packet volume and its type within the network. Window with time period is used in this method. Several datasets were run to find out the threshold value and it is a multiple of standard deviation of entropy values. In this method, the false positives are less than other methods and false negatives are more. Accurate percentages are not mentioned. Not even availability of resources is mentioned for faster calculations.

Oshima et al. [7] introduced a detection technique which is a short-term stat based on calculation of entropy. Small window sized entropy computation refers to the word Short-Term. Statistics are gathered with a window sized of 50 frames. For the measurement of lowest entropy window of various sizes were tested.

III. METHODOLOGY

There are various types of network simulating tools such as NS2, Mininet, W3, and FatTire.

NS2 is used for this experiment. A normal network is created by deploying number of nodes which will act as hosts in which normal traffic is generated between the nodes and one of the intermediate nodes will be implemented with policies and rules of controller. All the messages should pass through this node so that malicious data can be detected. This node will be called the controller.

An attack will be generated through an attack node which will try to exhaust the resources of the controller by flooding the network with malicious data.

DOI: 10.15680/IJIRCCE.2017. 0501022

In this project, we find the weak point of the SDN controller by which it can be overwhelmed when a DDoS attack happens and, propose a solution that is, specifically, tailored for SDN. Entropy is the method used in this research to detect DDoS attacks in SDN. Few parameters to DDoS detection using entropy includes; window size and a threshold value. Window size is either based on a time period or number of packets. Entropy is calculated within this window to measure uncertainty in the coming packets. A threshold value is needed to detect the attack. If the calculated entropy passes a threshold value or is below it, depending on the scheme, an attack is detected and as per specified rules the node acting as the controller will react to the attack either to forward the packet or to discard it.
Here in this project we will test following scenarios and compare the outcome based on results and will prove that the security policies and rules applied on the controller works against DDOS attacks.

- ➢ Normal Traffic generation without controller.
- ➢ Attack Traffic generation without controller.
- ➢ Normal Traffic with controller having security policies and rules.
- ➢ Attack Traffic with the same controller so as to detect the attacks.

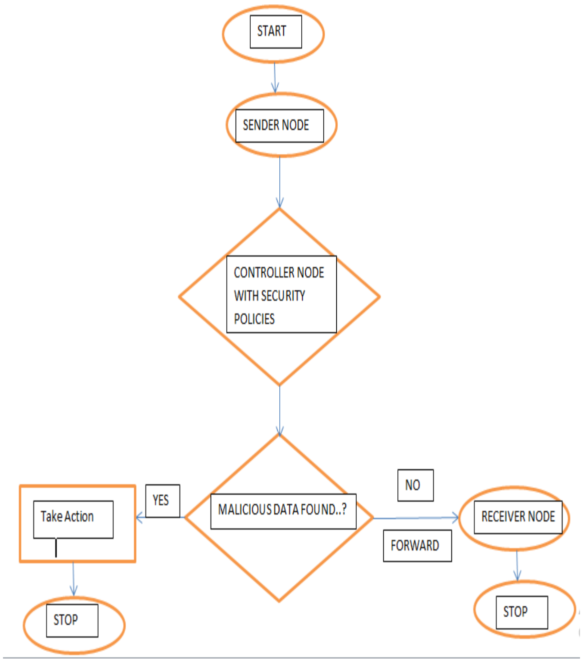

Fig 1. Architecture of system

In the above flowchart when the network is created a sender node will initiate communication and the message will be forwarded to the switch, the switch will look into its flow table and if present the message will be forwarded to the destination and if not present the message will be forwarded to the controller. The controller is equipped with security policies where the sender node will be scanned for malicious content. If the sender founds normal then a new entry will be made in the flow table of the controller and will be forwarded. If malicious data is found, the controller will react to the node according to the policies implemented.

IV. CONCLUSION

Controller is the main part of SDN and it is similar to Operating System and should be secured all the time. We tried to find out the gaps through which controller can be overwhelmed and the whole network will shut down. In this paper a solution is proposed to detect DDOS attacks on the controller of the Software defined network. There are various methods to detect attacks. We are attempting to detect attack based on Entropy. By implementing entropy as a detection method, we will able to detect attacks on one host or a subnet of hosts in a network

REFERENCES

[1] A. Lara and B. Ramamurthy, “OpenSec: A framework for implementing security policies using OpenFlow,” in Proc. IEEE Globecom Conf., Austin, TX, USA, Dec. 2014, pp. 781–786.

[2] A. Lara, A. Kolasani, and B. Ramamurthy, “Network innovation using OpenFlow: A survey,” IEEE Commun. Surveys Tuts., vol. 16, no. 1, pp. 493–512, Feb. 2014.

[3] D. Huang, L. Xu, C. Chung T. Xing, "SnortFlow: A openflow-based Intrusion Prevention System in Cloud Environment," Second GENI Research nad Educational Experiment Workshop , pp. 89-92, 2013.

[4] R. Bennesby, E. Mota, A. Passito P. Fonseca, "A Replication Component for Resilient Openflow-based Networking," in Network Operations and Management Symposium, 2012, pp. 933-939.

[5] Z. Qin, L. Ou, J. Liu, A. X. Liu J. Zhang, "An Advanced Entropy-Based DDoS Detection Scheme," in International Conference on Information, Networking and Automation, 2010, pp. 67-71.59

[6] I. Ra G. No, "An efficient and reliable DDoS attack detection using fast entropy computation method," in International Symposium on Communication and Information technology, 2009, pp. 1223-1228.

[7] T. Nakashima, T. Sueyoshi S. Oshima, "Early DoS/DDoS Detection Method using Short-term Statistics," in International Conference on Complex, Intelligent and Software Intensive Systems, 2010, pp. 168-173.

BIOGRAPHY

Omkar Chippalkatti is a M. Tech student from G.H Raisoni College of Engineering, Nagpur (An Autonomous Institute Affiliated to RTM Nagpur) he has completed his Bachelor of Engineering from Pune University in the year 2014. His area of interest includes Android Development, Networking.

Sonali U Nimbhorkar received Post Graduate degree in computer science from RTMNU, Nagpur. She has published more than 47 research papers in various international journals and international conferences as a main author and co-author in the field of issues related wireless network, wireless mesh network, network security and cryptography. At present she is assistant Professor in Computer Science & engineering Department in G.H.Raisoni College of Engineering Nagpur, India.