An Approach for Detection of Attacks in Software Defined Networks

Omkar Chippalkatti
Dept. Computer Science and Engineering
G. H. Raisoni College of Engineering, Nagpur
Nagpur, India
chippalkatti_omkar.ghrcemtechcse@raisoni.net

Prof. S. U. Nimbhorkar
Dept. Computer Science and Engineering
G. H. Raisoni College of Engineering, Nagpur
Nagpur, India
sonali.nimbhorkar@raisoni.net

***Abstract*— Software Defined Network is new turning up prototype architecture which efficiently separates the controlling and forwarding planes to provide flexibleness for the network manager to acquire control over network centrally. This centralized concept of SDN provides lower costing with maximum visibility of network and high throughput. Using smart devices in the network, SDN contributes to make use of network virtualization, which in turn helps in energy conservation and centralized security for increasing overall performance of the network. Software defined networking represents a latest approach at how networks are operated through centralized system. The SDN controller is the heart of SDN architecture where we can apply customized security rules and policies for every data flow in the network. Controller provides decision making facilities when malicious traffic is generated. Our approach in this project is to design security policies for the controller for detecting DDOS attacks and take action against it.**

***Index Terms*— SDN, DDOS, Network Architecture, Network Attack, IDS.**

I. Introduction

Traditionally the network appliances such as switches, routers were associated with the network functionality. It requires the network administrator to manually configure network when new device is added into the network. This is time consuming and error prone job. Following figure shows traditional network where all devices are manually configured and managed.

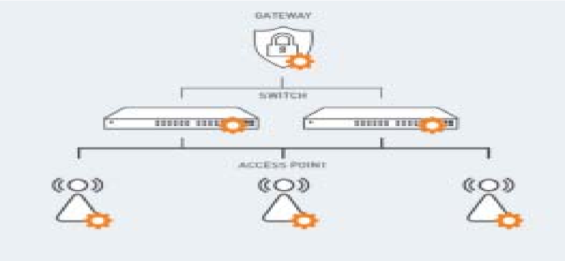

Fig. 1. Traditional Networking

This manual configuration approach makes it difficult to the network administrator to configure and manage each device. It becomes complex to set security policies which leads to security breaches. In SDN, this complexity of manual configuration of each and every device is abstracted and the whole underlying network is controlled by the control plane. In SDN architecture the control plane and data plane are decoupled and the whole network is controlled by the control plane. Control plane determines where to forward the message whereas the data plane executes the decision and forwards the data. This concept is implemented using network virtualization where the data plane is linked with the hardware and control plane with software. Following figure shows Software Defined Network.   It has very crucial to keep an eye on network activities. In SDN, controller is the center point and the main target for attackers because of centralized network activities. If the controller is compromised, then the whole network can be compromised easily. As the switches are responsible only for forwarding purpose, attack can also be easily forwarded which will be biggest threat to the entire network. In SDN we do not configure each device manually; in fact we depend on automatic implementation of network policies and rules applied in the controller. SDN decouples the data functionality and controlling functionality and moves the former to a software based logical centrally located network controller.

Network devices communicate with the controller through a secure channel of OpenFlow protocol. OpenFlow protocol is the backbone of Software Defined Networks. The OpenFlow switches are controller by the controller through set of rules and policies. Using OpenFlow, the incoming and outgoing traffic in an application can be easily controlled by the controller.

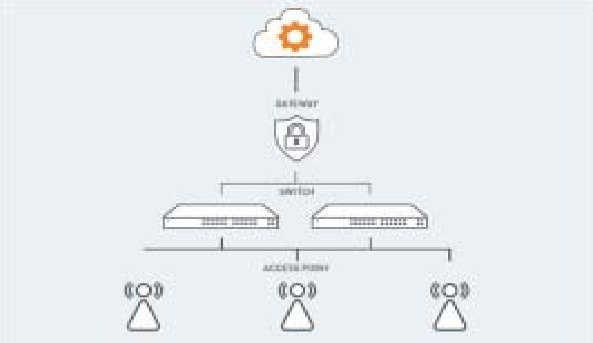

Fig. 2.SDN Configuration

Network administrator can implement its own designed security rules and policies through network framework. Suppose an administrator in an organization wants to monitor incoming and outgoing web traffic to an intrusion detection

978-1-5090-3294-5/17/$31.00 ©2017 IEEE

system (IDS) and e-mail traffic to a spyware detection device. Our aim is to take the advantage of SDN to allow the operator to code a high-level policy to produce this, rather than manually configuring every device. Moreover, suppose the IDS find unwanted traffic and want to avoid the traffic and the source has to be blocked from accessing the network. Rather than manually disabling the access to the network by the administrator, we are keen in automatic prevention of the source. Network administrator does not need to worry about the underlying network and just rely on the security policies and rules. Framework has a layer of software that runs on topmost part of the network controller, and all other devices that are connected to the controller including the security systems such as IDS are controlled by the controller. The ultimate goal of framework is to provide network administrators to customize security rules and policies for desired flows. The rules contain information of the flow, a list of security services that apply to that flow and its required action in case of detection of unwanted data. This reaction can be a just a warning only, or to prevent the traffic, otherwise fully stop flow of all packets from a specific source.is utilized so that defiled records can't be transferred.

II. Related work

It describes the seriousness of network administration to be managed with the discovery of Software defined Networks and policy based OpenFlow protocol. Applications that are software oriented which execute on the uppermost layer of the network controller provide abstraction of the underlying topology and efficiently look after network operations with the support of Controller. Author insisted OpenSec, OPenFlow protocol based framework which allows a network security administrator to design and implement security rules executed in simple and human readable language. OpenSec, allows end user to design a desired flow depending on OpenFlow matching fields, design variety of security services to be applied to that path and kind of security that define how OpenSec framework will behave when malicious data is detected in the traffic. GENI test bed was used for Evaluation of accuracy, elasticity and correctness of the OpenSec framework. [1] Software Defined Networks is currently based on

OpenFlow protocol. Comparing with the traditional networks the control panel and the data panel are detached in the Software Defined networks. Software based Controller has the power to manage the communication between the devices whereas the hardware has the power just to listen to the rules of controller to forward the packets from source to destination. OpenFlow protocol based standard SDN technology which determines how the communication takes place in between the controller and all other devices in SDN architecture. Researchers were permitted to test new discoveries in a real production environment. OpenFlow allows implementing the logic of controlling the switches through the Controller. [2]

Represents implementing Openflow with SNORT. It is a simple intrusion detection appliance for huge networks, which will help to customize the network on exposure of attack through SNORT. The above architecture was not designed to detect DDOS attacks in the cloud but ability of SDN for intrusion detection to lessen seriousness in a complex network such as a cloud. The important aim of this research is to find a way to ensure protection of the SDN architecture, mainly, the controller. [3] figured the practicability of retaining the controller and may require one more controller for a backup purpose. The paper shows an extra controller which runs synchronously with the current controller. The new backup controller is added in the network so that in case of communication loss the switches in the network can look after another controller. This seems helpful in case of DDOS attack on the controller which may drain all its resources and the controller may go off. The solution for situations like these is making use of machine learning like SOM which starts recovery during attack. [4]

Proposed a Entropy detection method that contains window sized of 0.1 seconds and threshold value with three levels. This mechanism is related with goal of preventing false positive and false negatives in the network. However, the authors themselves declared that the method consumes lot of time and makes use of lot of resources. [5] proposed a faster way of calculating entropy by referring the calculation on both volume of packets and its type within the network. Method used in this is window with time period. Various 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 specified. Not even availability of resources is mentioned for faster calculations. [6]

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

III. Proposed Methodology

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

Network Simulator 2 (NS2) is used for implementation of this project. A normal network is generated by deploying number of nodes which will act as devices which and in which regular traffic is created between the nodes and one of the intermediate nodes and few nodes will act as switches and one node will act as Controller. All the messages should pass through the switches. In NS2, the nodes contain routing tables which stores data of other network devices. When a message arrives at switch, the switch looks in to the routing table for finding the source and destination. If required information is available in the routing table, the switch just forwards the message to the destination. Suppose a new message arrives at switch, there will no information stored in the routing table, hence the message is forwarded to the controller for processing. When a new message arrives at controller the controller will process it by adding its source address and destination address to its routing table and instructs the switch to forward that message to the destination. When the new message comes to the controller we will apply security policies to find whether the message is not malicious.

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.

A malicious data will be generated through a attack node which will try to exhaust the resources of the controller by generating congestion in the network with malicious data.

In this project, we search the fragile point of the SDN, i.e controller by which it can be overwhelmed when a DDoS attack hits 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; packet volume size and a threshold value. Window size is either based on a time period or number of packets. Entropy is assumed within this window to check uncertainty in the incoming packets. A threshold value is needed to detect the attack. If the calculated entropy passes a threshold value or is above 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 discard or forward that packet.

Here in this project we will perform following scenarios and compare the outputs.

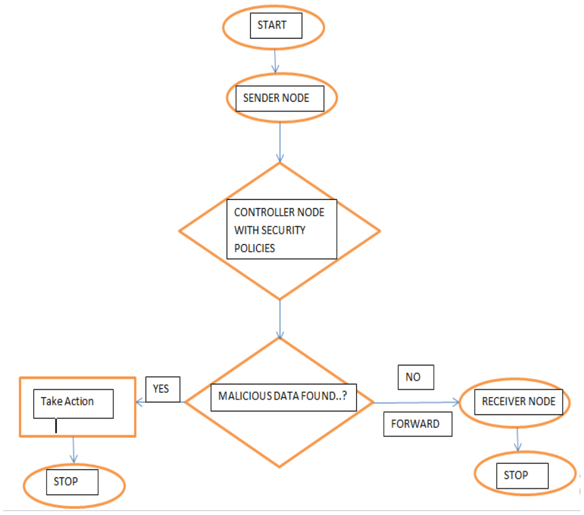

Fig. 3. System Architecture

IV. Conclusion

Controller is the heart of SDN and it is similar to Operating System and hence attraction for attackers to compromise and that's the reason it should be secured all the time. We tried to find out the black holes through which controller can be compromised and the whole network will shut down.

In this paper an initial solution is proposed to detect DDOS attacks on the controller of the Software defined network. There is variety of techniques to detect attacks. We are attempting to detect attack based on Entropy. By using this method, a start towards securing the centralized Controller is initiated.

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.