SDN simulation in mininet to provide security via Firewall

Nivedita De[1], Khushbu Kahar[2], Prof. Prathamesh Tugaonkar[3]

[1,2] Student, Dept. of Computer Engineering, Terna Engineering College, Nerul , Navi Mumbai
[3]Professor, Dept. of Computer Engineering, Terna Engineering College, Nerul , Navi Mumbai

---

Abstract - *Software Defined Networking is a progressive expertise that provides improvement and flexibility in scheming and dealing with networks, but it also offers new security threats. Our main goal is to build powerful firewall applications for shielding software based networks. In this paper we are developing an OpenFlow based firewall application. The implementation shows that most of the firewall functionalities can be built using software, without need of devoted and exclusive hardware. We are using open source POX Controller based on python for our experiments. To implement the project, we have used Oracle virtual box and installed Mininet emulator in it for creating SDN network topologies. In this paper, we have discussed the implementation particulars as well as experimentation outcomes of firewall application.*

Key Words: **SDN, Firewall, Network Security**

1. INTRODUCTION

Software defined networking enables the network creation without any use of hardware, hence it is economical and cost saving. A firewall is a system that secures incoming network packets, which come from various sources, as well as outgoing network packets. It can monitor and control the flow of data which comes into the network from different sources, and works on the basis of predefined rules.

Firewalls generally keep a barrier between a confidential, protected internal network and another outside network, such as the Internet, that is presumed not to be safe or reliable. They can be characterized as either hardware or software firewalls. Network firewalls are software programs running on different hardware appliances in the network.

Software based firewalls provide a layer of software on a host, which controls network traffic in and out of that particular machine. Firewall appliances may also provide other functionality to the internal network they protect, such as acting as DHCP or VPN servers for that network.

The system examines data packets for factors like layer2 or layer3 switch packet formats. It can also perform deep packet inspection for higher layer parameters (like application type and services, etc) to filter network traffic. Firewalls are an important element of any protected network communication for bi-directional packet flow.

1.1 Software defined networking

Software defined networking (SDN) is a network technology that stresses the separation of the network and the control plane. Responsibility is divided between both the planes. The forward plane is responsible for packet forwarding in the network. The control plane is responsible for policy creation and its implementation, based on predefined rules. It acts as the entry point of the system and can replace the conventional router. Any packet entering into the network is checked by the control plane and a choice is made whether to leave the packet or make it available to the next host. It can also update the IP table entry.

The controls are centralized on SDN controllers. SDN is an appealing stage for network virtualization, since control logic can run on a controller rather than on physical switches. It is a method to computer networking that allows network administrators to accomplish network services by the generalization of complex level functionality. So, SDN networks give flexibility, programmability and easiness to network operations. Traffic can be fixed, adjusted or personalized without requiring physical wiring changes. An SDN promises combined control and traffic management, which deals with computerized network security that is more adaptive.

The characteristics of SDN are:

- It can be programmed straight away.
- Responsive and alert
- Controlled centrally
- Testing and research is not costly
- Fast improvements

1.2 SDN architecture

SDN architecture is encouraged on authorizing network administrators to manage and control the whole network through a software program based controller. This goal is met through the separation of the data plane and control plane, which simplifies the networking services. The architecture of SDN consists of 3 layers such as Forwarding Layer (OpenFlow switch), control layer (Controller), Management layer (Applications) as shown

Forwarding Layer (OpenFlow switch)

This layer consists of Physical or virtual openflow switches and other network devices such as routers. It is also called as data plane.

Management Plane (Applications)

In this, the controller instructs the data plane. It consists of number of applications such as load balancer, firewall, router, switch. The organization of new applications become simplify due to separation of control and data plane in Software Defined Networking. Control plane provides us an integrated view of whole network. Switches become unassuming since now only forwarding actions are performed by data plane. Control decisions are shifted to control plane. The control decision depend upon a management layer that is application.

Control Layer (Controller)

It is positioned at the middle of forwarding layer and management layer. It instructs the forwarding layer. There are two ways how control plane gives instruction to the forwarding plane. In Proactive approach, the control plane pre installs all rules into flow table of switch immediately after the connection is made between Openflow switch and controller. In Reactive approach, the controller reactively installs the rule into flow table of switch immediately after the openflow switch receive the packet but it has no knowledge about that packet.

1.3 Traditional vs SDN based firewalls

Here are a few differences between traditional and SDN based firewalls:

- Interior traffic cannot be seen and cannot be filtered by a traditional firewall.
- An SDN based firewall works both as a packet filter and a policy checker.
- The first packet goes through the controller and is filtered by the SDN firewall.
- The successive packets of the flow directly match the flow policy defined in the controller.
- The firewall policy is defined centrally and applied at the controller.

2. APPROACH

There were two tactics considered in implementing the firewall: a) pre-installing the rules into the switch's flow table and b) handling the packets directly as they come in. We opted to manage the incoming packets directly because of the flexibility in management. One disadvantage of this method is that too many packets can be delivered to the controller which take up a huge portion of its resources. It is a lot more convenient to block unnecessary packets at the switch level. To cope with this issue, the user can also decide to install a 'deny' flow adjustment on the switch to carry on dropping alike packets for a certain time period. The logic of this firewall is as follows: each packet headers are checked against the firewall rule from highest to lowest priority, and performs specified action once matching fields are found in the rule. Any unmatched packets are dropped.
Installing firewall rules are possible from an external entity through a text-based user interface.

3. METHODOLOGY

In order to test the workability of this firewall, the following programs were used:

- Virtual Box - offers a background for virtual network to be formed.
- Mininet - provides virtual SDN network topology.
- POX - SDN controller.

Finally a simulated network is built on mininet network simulator and random network traffic is generated from hosts to the servers. The firewall is able to identify any suspicious activity & alert the concerned parties.

4. IMPLEMENTATION

We created a SDN network for our project using an emulation tool called Mininet. We can also create SDN network using hardware testbeds such as GINI [10], VENI, Emulab, FIRE etc. So first we downloaded the oracle virtual box in windows 7. Then we downloaded the mininet tool and imported into the virtual box to run it. We configured the network settings of the mininet according to our project and then we run the code for pox controller. Miniedit was used which was run in Xming(open source visualization tool) to show the network topology consisting of hosts and switches also the controller. A dialog box in miniedit can be used to specify the ip address of hosts that we want to connect to or block the host from connecting to. The pox console and the mininet can be connected and hence we can efficiently monitor the traffic flow and also see it coming from different hosts. A virtual switch has been used in the mininet for connection between the host and the controller.

5. IMPLEMENTATION

We created a SDN network for our project using an emulation tool called Mininet. We can also create SDN network using hardware testbeds such as GINI [10], VENI, Emulab, FIRE etc. So first we downloaded the oracle virtual box in windows 7. Then we downloaded the mininet tool and imported into the virtual box to run it. We configured the network settings of the mininet according to our project and then we run the code for pox controller. Miniedit was used which was run in Xming(opensource visualisation tool) to show the network topology consisting of hosts and switches also the controller. A dialog box in miniedit can be used to specify the ip address of hosts that we want to connect to or block the host from connecting to. The pox console and the mininet can be connected and hence we can efficiently monitor the traffic flow and also see it coming from different

hosts. A virtual switch has been used in the mininet for connection between the host and the controller.

6. CONCLUSION

SDN firewalls are playing an important role in modern day security. Integration of stateful features to an SDN firewall makes the firewall more intelligent and aware. The stateful SDN firewall prevents many attacks like DoS attacks carried out by hackers or impersonators. The controller has been made intelligent to analyse the network behaviour and act more like distributed firewall. SDN is still developing in so many areas and further research must be done to enhance the network security by covering all the aspects.

REFERENCES

[1] opennetworking.org website

[2] www.brianlinkletter.com website

[3] Opensourceforu.com website

[4] github.com

[5] openflow.stanford.edu site