International Journal of Research (IJR) e-
ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 06, June 2015
Available at http://internationaljournalofresearch.org

Establishing User-Defined Firewall in Software Defined Network

Senthil Kumar S& Perumalraja R
Department of IT, Velammal College of Engineering and Technology, Madurai, Tamil Nadu, India
{Senthilkumar7390}@gmail.com and rpr@vcet.ac.in

Abstract—

Software Defined Network (SDN) – Latest trend in current networking technology with many advanced features and security by decoupling the data plane and control plane using OpenFlow Protocol (OFP). The centralized controller does all the activities in the network and it needs more security than other parts of the network. The firewall can be an effective means of protecting a local system or network of systems from network based security threads while at the same time affording access to the outside world via Wide Area Network (WAN) and the Internet. This study focusing on to establish the user defined firewall rules with advanced technologies.

Keywords-MAC;SDN;Firewall;OpenFlow;Big Switch;

I. INTRODUCTION

SDN firewall is one of the security mechanisms to secure the SDN. This SDN firewall is fully based on the Internet Protocol address and only suitable for fixed network. It fails in dynamic environment, when a mobile node tries to access the network. Firewall is used to protect almost every network and individual systems. Most of the firewalls are very often misconfigured[11]. Testing of firewall rules is very important to ensure the networks' security.

OpenFlow was proposed to standardize the communication between the switches and the software based SDN controller. OpenFlow initially deployed in academic campus network. The main goal of OpenFlow was to provide a platform that would provide a flexible environment that run and analyze the experiment [4].

Configuring the firewall in SDN controller and configuring firewall on each and every switch of the SDN. The ordinary SDN switches have very limited space to place the firewall rules. The SDN firewall needs more number of rules to provide high security. Hence, the new mechanism is used with the name of Big Switch Abstraction. "Big Switch" abstraction is introduced as a network specification mechanism. The packet forwarding which is one of a core function of networking can be defined by the two kinds of rules. One is a routing policy to define how the packets can traverse in the network from source to destination, and the other one is, an endpoint policy to describe other aspects such as, packet monitoring, access control and so on.

The rule placement is constrained by the rule capacity limitation in each switch. Ternary Content Addressable Memory (TCAM) is the core component placed in the SDN switches, which provides parallel packet matching mechanism. TCAM is very expensive in terms of consuming the power. It limits the number of rules per switch. The size of TCAM is usually 1k ~ 2k [2].

The main reason of we go for the SDN is security. Without the controller plane, any intermediate cannot access the network. One of the major problems is Denial of Service (DoS) attack. The SDN centralized controller manages all the network activities. Obviously the problem can be occurs only in the controller part by assigning more workload. To prevent this we can use Access Control list (ACL) in firewall to allow or deny the packets. But the traditional firewall fully based on IP address. If dynamic IP environment, this will not be perfect. If we use user defined firewall mechanism we can overcome the problem. User defined firewall mechanism ensures the security in SDN firewall also by implementing the firewall on SDN switches, we can reduce the workload of the SDN controller as well as can protect the network from DoS attack.

II. LITERATURE STUDY

A. *Software Defined Network*

SDN separates the control plane from the data plane by means of centralized controller which controls the every switch in the data plane. We can

International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 06, May 2015
Available at http://internationaljournalofresearch.org

implement the network rules through SDN controller to installing and deleting rules in the SDN switches. Each switches has its own matching table contains strict polices to performs the parallel packet matching. The actions performed by the switches are, forwarding the packet, send the packet directly to the port, or drop the packet. The incoming packets' header will be matched with the predefined matching rules in the matching table. The controller can add or delete the rules by sending special messages. Similarly the switches send the messages to the controller regarding, packet mismatch, link status, and rules information.

B. Big Switch Abstraction

The recent research topic in the network is Big Switch abstraction which is introduced as a network specification mechanism. The core function of the network is packet forwarding which is done by the two major rules. The first rule is a routing policy to define the packet traverse path along the network and the second one is an endpoint policy to define the core functions such as access control packet monitoring, accounting, etc. These two polices may change dynamically. The key advantage of the switch policy is to reduce total networks' workload. In SDN aspects, the major possible is Dos attack. This also can be prevented by reducing the workload of the controller. The rule placement is constrained by the rule capacity limitation in each switch. The main component inside an SDN switch (or OpenFlow supported switch) is a Ternary Content Addressable Memory (TCAM) that provides parallel packet matching. However, a TCAM is very expensive in terms of the power consumption and thus is a scarce resource for each switch. This limits the number of rules per switch. The size of the TCAM is usually 1k ~ 2k (1.5k TCAM slots in the 5406zl switch [2].

III. EXISTING METHOD

The existing firewall in SDN controller handles the packets directly to achieve flexible management [3]. The firewall fragments all the incoming packets' header and match with the existing firewall rules. Any unmatched rules found then that packets are dropped. The firewall rules are installed as the external entity via text based user interface, similar to Access Control List (ACL) mechanism. The ACL performs based on the Internet Protocol addresses. It is only capable in static IP environment. In dynamic IP environment, ACL fails to achieve its' goal [11]. This is one of the major drawbacks of existing firewall in SDN. Also the firewall implemented on SDN centralized controller. It makes the controller to process heavier workload. Because of this the performance of the whole network can be degraded.

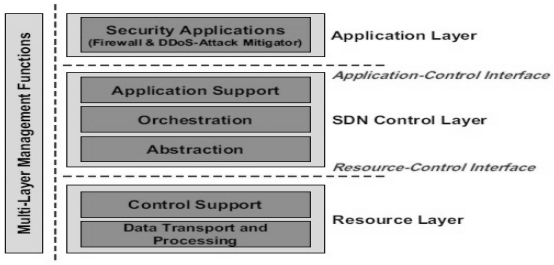

Figure 1 SDN Architechture

The centralized controller is the heart of SDN. It only needs to manage all the process of a particular network. To maximize the efficiency of the network, the workload of the controller need to be reduced. One of the way to achieve this by, to split the controllers' workload by separating the firewall from the controller and assign it to the SDN switches, which is the most suitable part in SDN environment [12].

IV. PROPOSED METHOD

From the existing firewall over SDN controller, we have separated the firewall from the SDN controller, and we have implemented MAC filtering by replacing the IP based packet filter. Instead of checking each and every frame of the packet header, the MAC address field alone is analyzed with the existing MAC list. If the MAC address does not exist then the system cannot process or utilize the network; and that particular system will be included to the authentication mechanism which is handled by the proxy controller. Once the authentication process completed, then the MAC address and other credentials or stored and it allowed to accessing the network. The authentication mechanism checks the size and length of the incoming packets from the new MAC. If it exceeds the default values then that particular mobile will be halted for certain time limit to ensure the medium getting affected by overload. The threshold values can be varied based on the protocols. We have implemented and tested the ICMP type protocol. The l2 learning switch contains the default values of ICMP protocol [5]. We have design a simple

International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 06, May 2015
Available at http://internationaljournalofresearch.org

topology as shown in the fig 1. There is a centralized controller and three individual hosts are connected through the Openflow Switch. Openflow Switches contains the default packet forwarding rules; these are defined in a java file named packetStraming. To implement and test the topology, we have used a open source tool called Mininet which provides a virtual SDN network topology and default network structures. We have used floodlight controller to design the topology and firewall. Floodlight is an open source platform which enables the users to design their SDN topology in java environment. It also provides some features like packet forwarding and learning switch as editable; with this we can overwrite the packet forwarding file for our convenience. It enables us to design our own topology with unlimited switches and hosts.

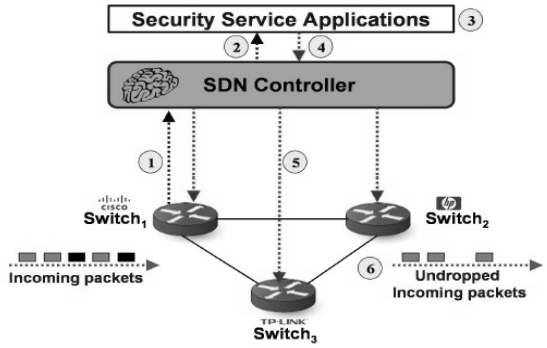

Figure 2 SDN topology used for testing

V. IMPLEMENTATION AND RESULTS ANALYZIS

In order to emulate and test our proposed method we used the tools;

1. *VirtualBox* - Provides a virtual environment to run the mininet tool.
2. *Mininet* - Provides a virtual SDN network topology and default network structures [9].
3. *Floodlight* – Java based SDN controller

The ordinary firewall was tested by attaching to a learning switch Floodlight controller. It uses a simple forwarding algorithm that updates the OpenFlow Switch by registering the packets' port, MAC, and IP addresses as they entering into the network. The topology has a remote controller attached to an OpenFlow Switch, which connects three or four hosts (Figure 1).

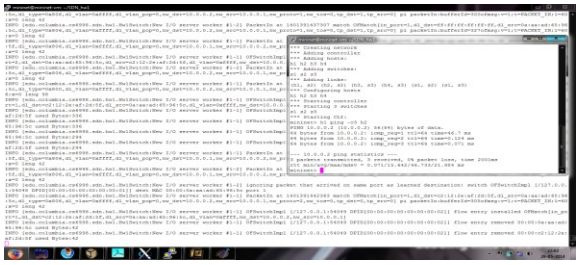

Figure 3: Ping Test result by authentication mechanism Switch1 forwards the incoming flow's packet to SDN centralized Controller.

1. SDN Controller forwards the incoming flow's packet to firewall application.
2. Firewall analyzes the headers and contents of the packet by Deep Packet Inspection.
3. If Firewall regards the packet as a malware's packet with a suspicious pattern, it reports the malware's packet to SDN Controller.
4. SDN Controller installs new rules (e.g., drop packets with the suspicious pattern) into switches.
5. The malware's packets are dropped by switches.

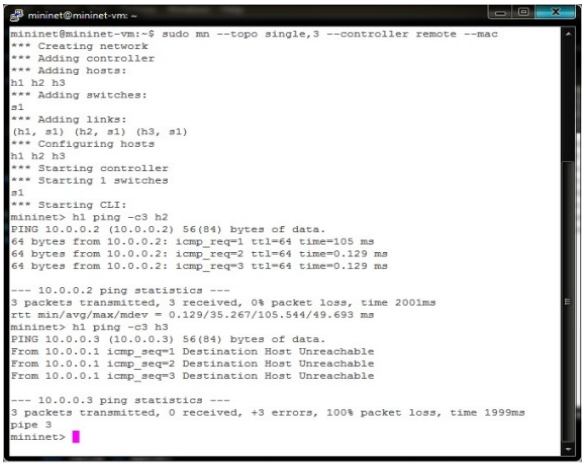

Figure 4: Ping Test result by user defined firewall

VI. RESULT ANALYSIS

In this experiment we have created and tested the SDN user defined firewall in Mininet emulation environment using floodlight controller. Floodlight is the open source controller which provides the user friendly environment to develop the network topology and testing that topology. A sample topology has been created in Mininet and firewall has been implemented in the SDN switches using Big Switch Abstraction concept. Some ICMP

International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 06, May 2015
Available at http://internationaljournalofresearch.org

packets have been tested by using ping from the existing IP address system which is successfully forwarded to the destination. Then ping command established from an unknown host. That host IP has not been registered with the Controller. Hence the packet has been forwarded to the proxy controller and authentication process has been started. The table 1 shows the comparison of the ordinary firewall with our proposed user defined firewall.

	Ordinary Firewall			User Defined Firewall		
Packets	Min	Max	Avg	Min	Max	Avg
P5	0.12	3.15	1.635	0.18	2.56	1.37
P10	0.89	3.18	2.035	0.20	2.89	1.545
P15	0.33	2.90	1.615	0.11	2.36	1.235
P20	0.45	3.11	1.78	0.25	2.76	1.505
P25	0.26	2.97	1.615	0.19	2.67	1.43

Table 1: Comparison of Firewalls.

The controller checks the packet size and type with the default values. That packet does not ensure the default values and has been dropped. Another ping command from a new host has been established with the network. In this time the new packet ensures the default values and gets added in the temporary access control list. At the end of the session the temporary ACL has been erased. With this a secured SDN topology can be implemented in the real time with our proposed SDN user defined firewall mechanism which will reduces the workload of the controller and increases the network performance. The rule predefined in the SDN switches can be Add, or delete, or modified from the SDN controller. This is one of the advantages of using the SDN.

VII. FUTURE WORK DEVELOPMENT

In this study experiment, we have developed user defined firewall mechanism over SDN Controller. Also we have tested using Internet Control Message protocol. In future work, we have to test other protocols such as Transmission Control Protocol, and so on which are used in Transport Layer. The Big Switch Abstraction techniques are used partially to store the MAC address and we are going to implement the MAC Filtering. In our next experiment, we have to fully utilize the Big Switch abstraction specifications. To avoid the rules conflict, the proxy controller can be used as virtual device for the mobile node authentication. To implement the MAC filter rules in each switch, we can cluster the SDN switches and implement the Big Switch abstraction to overcome the power utilization problem.

VIII. REFERENCES

[1] N. Kang, Z. Liu, J. Rexford, and D. Walker, "Optimizing the one big switch abstraction in software-defined networks," Proceedings of ACM CoNEXT, 2013.

[2] A. R. Curtis, J. C. Mogul, J. Tourrilhes, P. Yalagandula, P. Sharma, and S. Banerjee, "Devoflow: Scaling flow management for high-performance networks," SIGCOMM Comput.Commun. Rev., Aug. 2011.

[3] Michelle Suh, Sae Hyong Park, Byungjoon Lee, Sunhee Yang. "Building Firewall over the Software-Defined Network Controller", 2014 ICACT, pages 744-748, Feb.

[4] Adrian Lara, Anisha Kolasani, and Byrav Ramamurthy. "Network Innovation using OpenFlow: A Survey", IEEE COMMUNICATIONS SURVEYS & TUTORIALS, VOL. 16, NO. 1, FIRST QUARTER 2014.

[5] Ferro, G. (2013, Mar 18). "SDN Use Case: Firewall Migration in the Enterprise. Retrieve from Ethereal Mind", Mar 18, 2013.

[6] OpenFlow Switch Specification: Version 1.1.0 Implemented. (2011, Feb 28). Open Networking Foundation. Retriewed from https://www.opennetworking.org/

[7] POX. (2013). Retrieved from Open Source control platforms for SDN: NOXRepo.org

[8] POX Code Repository(2014). Retriewed from https://openflow.stanford.edu/display/ONL/POX+Wiki

[9] Mininet Team. (2013). Retrieved from Mininet: http://www.mininet.org

[10] Oracle Virtual Box Team (2014). Retrieved from https://www.virtualbox.org/

[11] AvishaiWool, "A Quantitative Study of Firewall Configuration Errors", 2005

[12] Justin Gregory V. Pena and William Emmanuel Yu, " Development of Distributed Firewall Using Software Defined Networking Technology", 2014.