Enhanced SDN Security using Firewall in a Distributed scenario

DHAVAL SATASIYA
CDAC, Pune
GTU PG School, Ahmedabad
dpsatasiya003@gmail.com

RUPAL RAVIYA
CDAC, Pune
GTU PG School, Ahmedabad
raviya004@gmail.com

HIRESH KUMAR
CDAC, Pune
AurionPro, Pune
hireshmalagar@gmail.com

***Abstract* – Software Defined Network (SDN) attract much attendance from research and industrial area; reason behind is the open interface, user controlled management and lower operating cost for data/flow handling rules that implements on software module rather than embedding in hardware with greatly improve performance. Network with the high security demanding higher in this generation. New service and products provide advanced technology and flexible architecture but changing in network landscape can introduce new security issues. SDN simplifies hardware, software and management of network device but faces many forms of attack. SDN or TNA (Traditional Network Architecture) has same uses firewall to manage legacy network traffic.**

This paper introduce firewall for SDN infrastructure to secure network attached device from suspicious and malicious traffic regarding to attack and access control to enhancing SDN security by distributing system.

***Keywords* - Software Defined Network, Distributed Firewall, Security, POX and OpenFlow.**

I. INTRODUCTION

Network security is a harder job than anyone can think, provide a security is like walking onto a liquid iron to a network connected machine compares to stand alone machine. There's no matter how hard you prepared against vulnerabilities but attacker can find new ways to get into a network. Cover a network infrastructure with efficient security manner; two step to do, First is to secure all entry point and consider all holes that can responsible for launching attack is second [11]. Against the war with cracker much many hardware and/or software based system used by administrators to secure their networks. In all of this firewall is a corner stone in cyber defense.

Firewall Technology is usually a most common and efficient manner at a different security level for the environment from the user access the system is safe. Firewall is a simple security mechanism that placed between a different zones of trust (public or private networks) and control the traffic [3]. Goal of this mechanism is to;

- All the network traffic must pass through the firewall [3]
- Only authorized traffic, based on security policy will be allowed to pass [3]
- The firewall itself is immune to penetration, maintain logs [3]

Firewall filers the traffic based on security policy that contain enforcement rules to filter network traffic. Source and destination address, protocols, port address and services; this entities are used to write rules. Firewalls are a powerful tool for network security. However, they are many things they cannot do; 1) The degree of protection provided by firewall is how carefully the permissive part is written. 2) Problem against a higher level protocols that uses encryption. 3) Can implemented defenses against only known flaws [12].

Now, firewall is been a crucial part of all type of network infrastructure and devices; SDN or TNA they have a different network technology but the core functionality of both is same to provide connectivity between network device and manage traffic across them.

Existing complex and hardware based network infrastructure that require high ability to install and configure network devices take placed by systematic software based approach. His idea is given by Nicira Networks based on their earlier development at UCB, Stanford, CMU, Princeton. This SDN concept not now too new, people will started to use cloud infrastructure to separate network OS from cloud instance services [4]. Same as the cloud computing SDN also improved network performance in terms of Quality of Service (QoS), network management and data handling. Because of that it can windy used in a datacenter and work optimized system.

Service focused requirements, centralized controller and independent (that are because of control is separate out of individual network node) logic has emerged SDN service. To collects the information network OS control the SDN switches [5].

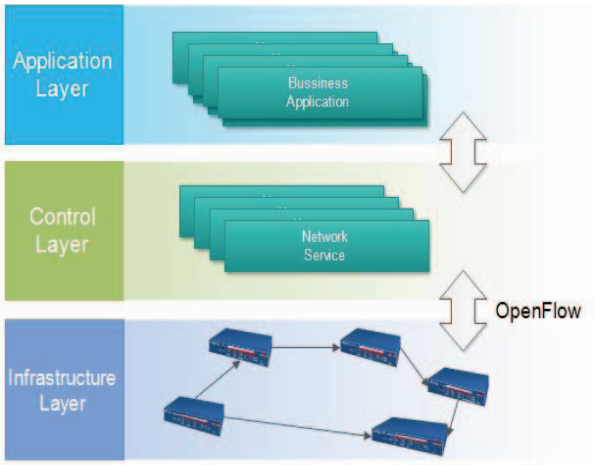

Fig. 1: SDN Architecture [3]

ISBN No.978-1-4673-9545-8

Common block infrastructure of SDN is shown in Fig. 1. Lower block is contain infrastructure layer where all network devices like switches, hubs and routers are placed this called as a 'data plan'. Middle block is collection of one or more controllers that have a complete knowledge of network for optimize flow management. Controller uses southbound API (eg. OpenFlow) and Northbound API to communicate with data plan and application respectively. And the most upper layer contains business and network related applications.

When the first packet is coming to the switch from user switch is use flow table (OpenFlow switches have flow table that maintain by controller) to check flow rules for packets. If rules is found than packet is forwarded to the respective outgoing switch port. If no one rules match than switch foreword packet to the controller via southbound API. Controller can add, modify or delete flow entities; that execute routing algorithm to install flow entries into the switch's flow table [5].

Separation of data plan and control plan enable the centralized view and control of network. Programmability of applications enables greatly improved use of resources and opens up the potential for new applications. Major advantages of SDN is 1) Intelligence and Speed, 2) Easy Network Management, 3) Multi-Tenancy, 4) Virtual Application Networks [4] [5].

SDN comes with the much newer and easy functionality to manage network infrastructure using software modules. Brightening of this new technology magnet the datacenter and cloud user to use as a core infrastructure. SDN is work to manage the communication and data traffic between many of connected devices. Security is required to protection against vulnerability. Network forensic, Policy modification and insertion of new security apparatus is now straightforward because SDN architecture enable networks to actively monitor traffic and diagnose threads [5][8].

So, in this paper proposed a firewall mechanism for SDN infrastructure to secure the traffic flow across the network. Here proposed system is follow distributed approach to provide greater level of efficiency and flexibility. Direction to design distributed scenario is address problem related to latency and firewall overload; and enhanced capacity and efficiency to block traffic.

II. MOTIVATION

Everything have a many disadvantages along with their important pros. A newer programmable network service that simplifies the network with the lower managing cost holds great promise. Whatever, but there was many challenges remains to addresses. Not much research done on SDN security; a number of issues are highlighted here that further study and underlined the need for the development of security solutions. On the plus side, SDN architecture is a highly reactive security monitoring, analysis and response system supports.

To take maximum benefits of SDN, the challenges should be listed so that appropriate security mechanism can adopt continuously. From a fundamental point of view, SDN security vulnerabilities are centered on their three plans [8].

Application Plane Security
- Authentication and Authorization
- Access Control and Accountability
- Fraudulent flow rules insertion

Control Plane Security
- Threats from Applications
- Threats due to Scalability
- DoS Attacks

Data Plane Security
- Flooding attacks
- Controller hijacking or compromise
- TCP-Level attacks
- Man-in-the middle attack

After studying existing systems of anonymous network, there are few points to consider:

- Encryption, firewalling and filtering and access control system can slow down the network.
- With real-time operation also storage functions, the standard example of a core slices are periodic heavy traffic can introduce latencies.
- Existing Firewall become a single point of failure (SPOF); a central OpenFlow switch is work as a firewall, the whole networks traffic was passed through firewall (OpenFlow switch), if too much traffic is passed than firewall is down and whole network is not accessible.
- Firewall overload is a common cause of failure.

This proposed system can overcome the present and future security challenges in SDN and future directions for secure SDN.

III. DETAILED DESIGN DISCRIPTION

Keeping in mind the limitations of existing systems and models SDN Firewall, here proposing which will contain following characteristics:

- Stage focus on control plane to manage the entire network is more extensive, and will bring a bunch of new features to the table. These changes streamline and networking operations must accelerate.
- An important consideration is to balance the load across the number of firewalls.
- And also tries to reduce overhead on SDN firewall and
- Provide good for flexible and easy configuration of firewall object on users need and demand.

The whole network flow by looking at the initial stage to blocking network attacks, improve the argument for SDN capabilities.

A. Design Methodology

According to Existing Firewall system OpenFlow Switch is work as a Firewall for SDN network; here we use same things but in some different way, the

OpenFlow switch is implement an intermediate firewall. Also every host in network have individual firewall object. Firewall object on host can also done using OpenFlow mechanism for light weight and flexible implement.

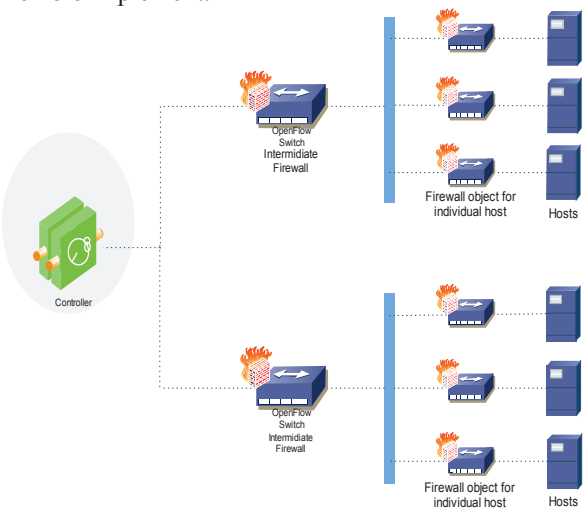

Fig. 2 Proposed Model

The intermediate firewall is define general rules for overall network; such as block broadcast packet or disable ping test into network. And individual firewall object is responsible for counter enforcement mechanism for individual host or network; like block incoming traffic from individual IP address or allow only some type of traffic http/https, ftp etc.

B. Algorithm Description

The valuable implement of a proposed idea we decide to make an algorithm for workflow in proper manner. Algorithm is responsible for the…

- Proper traffic flow across the number of firewall in SDN
- Handel the trigger when new host is adding to the network
- Assign firewall object to all individual host; and
- Link-up all firewall objects into intermediate firewall and work as a soul system
- Also manage the network traffic through intermediate firewall and individual firewall object.

Flow Diagram shows that assign a firewall objects to new add host in network. When the new host is connected to network the first event is called. In the first event a trigger is generated for new host; one process is listening for the trigger, if the trigger is generated then it's create a new firewall object, if not than continue to listen for a trigger.

After the create a new firewall object;

- First it's make an entry in Flow Table of OpenFlow switch, and
- Assign a individual object to host

After that the new host is connected to network with firewall object and OpenFlow switch is transfer controls to host and allow passing traffic.

Fig 4 is describe traffic flow diagram. Handle the incoming traffic by the controller, then passed to the OpenFlow switch (work as intermediate firewall) which distribute traffic in the network.

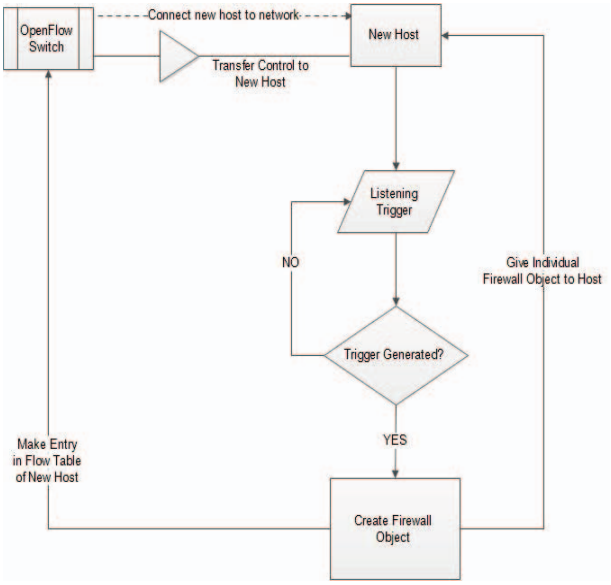

Fig. 3 Work Flow Diagram

It is match the incoming packet with rule table, if packet is matched then pass it to host where individual firewall also match with its rule. Either packet is matched for accepted or drop the unwanted packet.

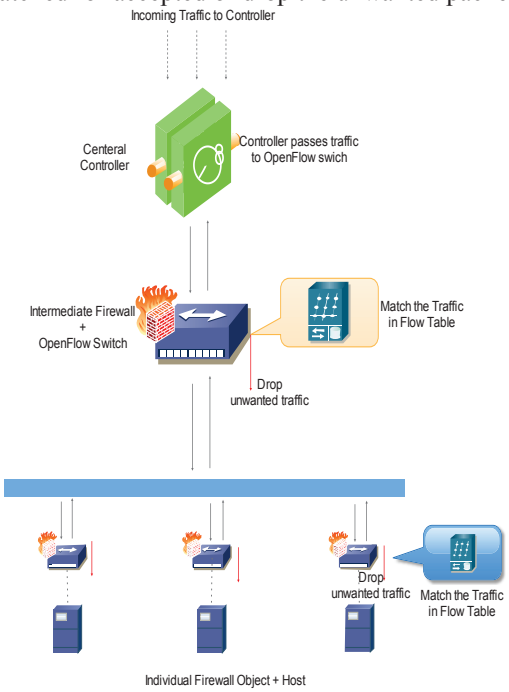

Fig. 4 Traffic Flow Diagram

Both Firewall is dropped traffic according to enforcement rules in Policy and generate a log entry for administrative purpose.

IV. PROTOTYPE DEVLOPMENT

First task to do in implementing proposed idea is to create a software defined network infrastructure as a platform for next task. Here we use LINUX DESTRO for the create software defined network infrastructure.

Steps to create a SDN are;

1) Install the Mininet network simulator
2) Install POX SDN controller

Implement firewall into the SDN architecture, must know how traffic flow into the software based network. And also OpenFlow switch is maintain a flow table to handle the network traffic within the SDN. So, identify packet flow and how flow table should maintain we gone through this experiment.

mn –topo=single,3 –mac –switch ovsk –controller remote

That creates SDN topology (Fig. 5) with,

3 Hosts (Named h1, h2, h3),
1 OpenFlow Switch (s0),
And, Controller (port 6633)(c0)

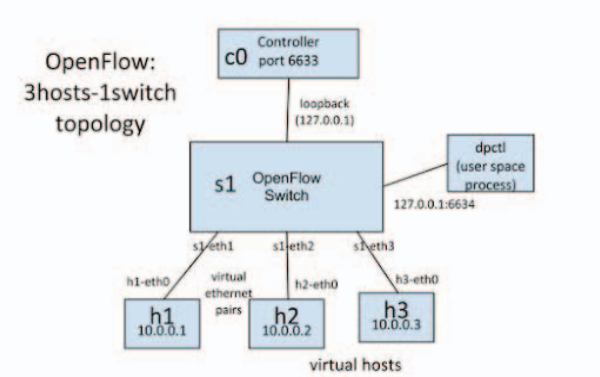

Fig. 5 SDN topology

python ~/pox-carp/pox.py forwarding.l2_learning info.packet_dump samples.pretty_log log.level –DEBUG

used to Invoking POX controller.

Test ping for connection between host h1 and h2 use following command into the mininet terminal, can also check connection with other link.

mininet> h1 ping h2

Now into the pox terminal debug log should be displayed. In which you found

DEBUG: forwarding.l2_learning:Connection [00:00:00:00:00:01]

That means entry for host1 is installed into the flow table when first packet was send by the host1 (in ping h2), and

DEBUG: forwarding.l2_learning:installing flow for 00:00:00:00:00:02.2 → 00:00:00:00:00:01.1
DEBUG: forwarding.l2_learning:installing flow for 00:00:00:00:00:01.1 → 00:00:00:00:00:02.2

That defines the flow rule into the flow table from host2 to host1 and 2nd is for host1 to host2. Where .1 and .2 indicates switch port for respective hosts.

A. Firewall Prototype

This firewall module implements firewall-like functionality for POX. The code uses python as a platform to develop firewall prototype. When a switch connects to the controller, the component initializes the connection to the switch as well as adding low priority flow entries to allow certain types of packets to pass through (i.e. ICMP, IP, ARP), but block all packets that are not specified by a rule in the firewall configuration file. It pushes medium priority flow entries for rules from the configuration file.

When it receives a packet, it checks the configuration rules to ensure that there is a match, then pushes symmetric flow entries from the packet specifics. If there is not a match, it pushes a flow entry with null action, so the switch will drop packets from that flow.

Our firewall module has contains two files;

fw.py: firewall source code

rule.conf: configuration file that contain enforcement rule for firewall.

To run this firewall module, must require running firewall code file (fw.py) with the controller.

Here rule.conf is a policy file for firewall modules, in that write firewall enforcement rule into table format. Entitles used to write firewall policy is; 1) Firewall Type, 2) Source and Destination IP Address, 3) Source and Destination MAC Address, and 4) Protocol type. The column Firewall type is for the rule is applied on which firewall of distributed architecture. Use '0' for the intermediate firewall means that rule is applied on whole network and if want to apply rule for particular host write 'ip address' for individual firewall. IP Address or MAC Address (source and destination) are used both or anyone to write rules. And protocol is mandatory entity.

Table 1: Firewall Policy

Firewall Type	Source IP	Desti IP	Source MAC	Desti MAC	Prot-ocol
0	10.0.0.1	10.0.0.3	00:00:00:00:00:01	00:00:00:00:00:02	icmp
10.0.0.3	10.0.0.3	10.0.0.1			https
10.0.0.3			00:00:00:00:00:03	00:00:00:00:00:02	ftp

Table 1 is shows the sample firewall rule policy. Here first rule is for block icmp ping between host 10.0.0.1 and 10.0.0.2 that apply on intermediate firewall (because firewall type is 0). And second and third rules are apply on individual firewall of host 10.0.0.3, for allow https traffic with host 10.0.0.1 and ftp traffic with host 10.0.0.2.

V. PERFORMANCE EVALUATION

Introduced firewall mechanism is uses distributed scenario so we conduct many evaluation of proposed system to identify loopholes and to prove the system is satisfy all necessary and promised functionality and perform well. Here two major testing scenario will discussed to analyses functionality and performance of proposed mechanism.

A. Functional Testing

Here we scribe the functional testing of our firewall module. We block the all network traffic and allow only icmp flow into the network between particular hosts.

Open *rule.conf* file and put the firewall rule for allow icmp packet in host1 (10.0.0.1) and host2 (10.0.0.2). Here we write in configuration file "1, 0x806, 10.0.0.1, 10.0.0.2". Here 1 for ID, 0x806 for icmp packet type

and remaining are hosts ip address. Create firewall with POX controller.

Use following command…

$ ./pox.py py log.level pox.forwarding.l2_learning fw

Here include firewall module "fw" that run over the controller. Can show the firewall module is running in Fig. 6 and inserted firewall rule also.

Fig. 6 Running Firewall module over POX

Now in the mininet console we check for the firewall work, so that we test ping configuration in virtual topology. For that type following command.

mininet> pingall 0

Show the output in bellow image.

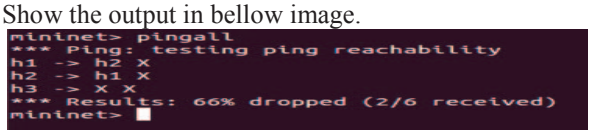

Fig. 7 Testing result

Here as per firewall rule here only host1 and host2 can enable to ping each other successfully. The *pingall* command used to send a single ping between every possible host pairs to test the connectivity of the entire network. Since there are 66% packets dropped in Fig. 7, this means that network is fully function with and according to rules written in policy file.

B. Performance Testing

Testing of proposed architecture into distributed scenario used network topology with central switch connected with the multiple hosts and controller also connected with central switch. Two different condition takes to test ping into network with 60+ times. First time use network without firewall. After that use firewall module into network and test performance of both individual firewall and intermediate firewall. Figure 8 shows the results of this testing output of both condition in Minimum, Maximum and Average time to pass traffic into network in graph format.

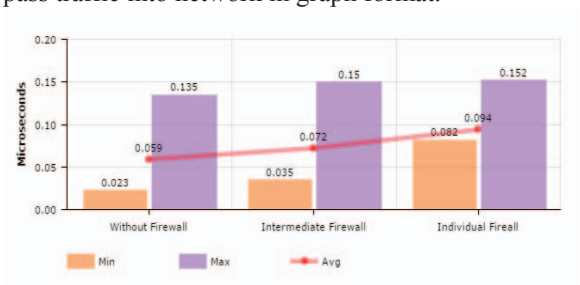

Fig 8 Times to handle packet in network

Deployment of the distributed firewall does not affect the overall efficiency of the network traffic filtering process. The results are shown on figure 8 there are not much different between average times; 0that means whatever the firewall used as a distributed scenario or only as single source of firewall has no negative effects in terms of latency.

VI. CONCLUTION

Easy and flexible configuration and robust control on application placed SDN to a popular network technology in today's era. Existing firewall technology is designed based on OpenFlow prototype, where firewall controller controls the traffic flow based on rule set define in policy and switch that makes enforce on controller regulating traffic flow according to their flow table rules. But the existing firewall cannot able to cover all explained security region.

SDN model proposed in the firewall is to develop a system of improved facilities. In the proposed system, we try to overcome the limitations of existing firewall system. That appropriate protection and reduces manual work. Many disadvantages of the current system to works well and have many more difficulties. The proposed system to eliminate or minimize these difficulties to some extent tries and will help users to reduce the workload and mental conflicts.

REFERENCES

[1] Karamjeet Kaur, Krishan Kumar, Japinder Singh, Navtej Singh Ghumman "Programmable Firewall Using Software Defined Networking" 2015 2nd International Conference on Computing for Sustainable Global Development (INDIACom) (978- 9-3805-4416-8/15) IEEE pp. 21252129.

[2] Michelle Suh, Sae Hyong Park, Byungjoon Lee, Sunhee Yang "Building Firewall over the Software-Defined Network Controller" 2014 Conference on Information Assurance and Cyber Security (CIACS) (ISBN 978-89968650-3-2) (978-1-4799-5852-8/14) IEEE pp 39-42.

[3] Dhaval Satasiya, Rupal Raviya "Analysis of Software defined network firewall (SDF)", 2016 international conference on wireless communication image processing and networking, IEEE.

[4] Fei Hu, Qi Hao, and Ke Bao "A Survey on Software-Defined Network and OpenFlow: From Concept to Implementation" 2014 Communication Surveys & Tutorials, VOL. 16, NO. 4, IEEE pp. 2181-2206.

[5] Sakir Sezer, Pushpinder Kaur Chouhan, Barbara Fraser, David Lake, Jim Finnegan, Niel Viljoen, Netronome Marc Miller and Navneet Rao, "Are We Ready for SDN? Implementation Challenges for Software-Defined Networks", 2013 (0163-6804/13/$25.00) IEEE.

[6] Smeliansky R.L., "SDN for network security", 2014 (978-1-4799-75952/14) IEEE.

[7] Jaehoon (Paul) Jeong, Jihyeok Seo, Geumhwan Cho, Hyoungshick Kim, and Jung-Soo Park, "A Framework for Security Services based on Software-Defined Networking", 2015 29th International Conference on Advanced Information Networking and Applications Workshops, (978-14799-1775-4/15) IEEE pp. 150-153.

[8] Ijaz Ahmad, Suneth Namal, Mika Ylianttila and Andrei Gurtov, "Security in Software Defined Networks: A Survey", 2015 IEEE.

[9] Scott Hogg, "SDN Security Attack Vectors and SDN hardening", Oct 28, 2014, http://www.networkworld.com/article/2840273/sdn/sdn-securityattack-vectors-and-sdn-hardening.html.

[10] Xin Vue, Wei Chen, Yantao Wang, "The Research of Firewall Technology in Computer Network Security", 2009 Second Asia-Pacific Conference on Computational Intelligence and Industrial Applications, (978-1-42444607-0/09), IEEE, pp. 421-424.

[11] Steven M. Bellovin and William R. Cheswick, "Network Firewalls", IEEE Communications Magazine September 1994 .

[12] Saeed Al-Haj and Ehab Al-Shaer, "Measuring Firewall Security".