International Journal of Current Advanced Research
ISSN: O: 2319-6475, ISSN: P: 2319-6505, Impact Factor: SJIF: 5.995
Available Online at www.journalijcar.org
Volume 7; Issue 2(E); February 2018; Page No. 9960-9963
DOI: http://dx.doi.org/10.24327/ijcar.2018.9963.1665

Research Article

FIREWALL BASED ON THE CONCEPT OF SDN

Yash Bajaj., Yogesh Rohra., Viren Wadhwani and Arthi CI

Computer Department, Vivekanand Institue of Technology

ARTICLE INFO

Article History:

Received 7th November, 2017
Received in revised form 13th December, 2017
Accepted 3rd January, 2018
Published online 28th February, 2018

Key words:

Open-Flow,SDN, Firewall,Mini-net,Open-Day-Light.

ABSTRACT

In todays digital world many people have utterly felt the need to restructure the current internet work into one which is much more like a dynamic networking environment. Software Defined Networking finds a solution to these problems. Software Defined Networking is an exciting yet beautiful technology that enables innovation and flexibility in designing and managing networks present currently. But on the other hand it also introduces new security threats and issues beforehand. Hence our aim is to build a firewall over this new technology to protect the integrity of it. So designing and developing Open-Flow based firewall application will form the basis of the project. The making and implementation shows that most of the firewall functionalities can be built using a software, without use of any hardware explicitly. We are using open source OpenDayLight Controller Carbon Version for our experiments. To perform experiment, we have installed Mininet emulator for creating network topologies. In this paper, we present the implementation details of the firewall application.

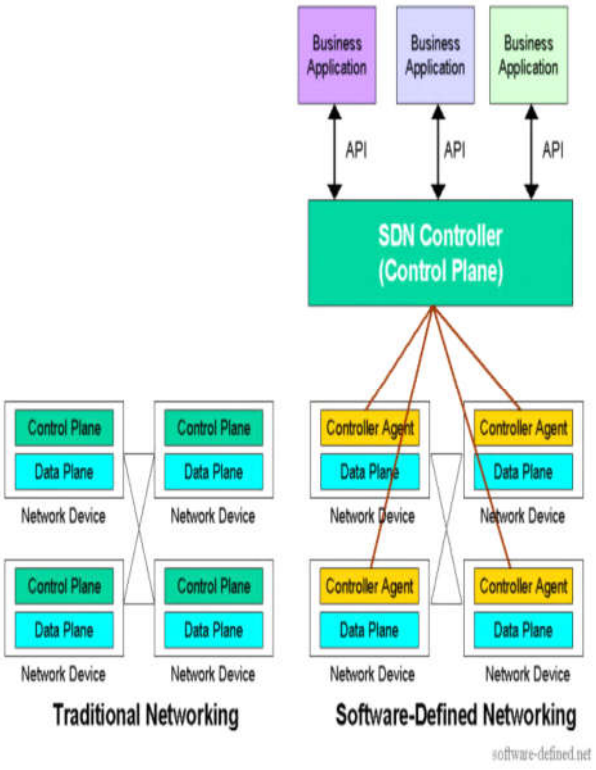
INTRODUCTION

The current internet network is well set and huge in terms of topologies and its connections all over the world. But it has its drawbacks too. The data and the control plane being together yet makes it difficult in finding optimality in terms of routing, maintaining security and much more. Also adding or removing even a single router makes us change the whole network and is tedious. In a view of finding solution the SDN technology was introduced in 2005 where in the control plane gets totally removed and what we get is a controller i.e. the brain of the network and a bunch of routers that are meant for handling data and routing them. The new brain so formed is called as the OpenDayLight controller in the world of SDN. A controller responsible for taking the routing, protocol decisions whenever a packet arrives at its door. Our project aims at building a firewall over this controller that will in turn help the controller to take the decisions regarding the packets that arrive at the controller. The controller accompanied with the firewall will have the ability to accept, drop, or reject the incoming packets thus ensuring the safety of the system from malicious packets. Also the firewall which is created on POX platform, will be able to save the system from any malicious attack by an intended attacker. The remainder of this paper is organized as follows, Section 2 presents the comparison of this project with the existing system, followed by Section 3 which states the proposed system and the implementation of each block of the system, the Limitations are shown in section 4 and in the last section conclusions and references are presented.

Corresponding author:* **Yash Bajaj
Computer Department Vivekanand Institue of Technology

Comparison With Existing System

On viewing the Existing systems and comparing it with the project we find that the basic disadvantages are the more priveleged uses of SDN technology.



As mentioned earlier the existing networks are highly complex. The actual comparison between the two can be depicted by the figure above. The traditional and yet existing system consists of networking elements that posses both the control and the data plane in every element that is installed in the network. This puts a slight disadvantage as setting up a new element would require making changes in control plane of almost every element that exists in the network. This leads to decrease in optimality goal of the network. SDN on the other hand takes the control plane from every element and creates the SDN controller and the elements left without the control pane are now termed as SDN switches that are just used to regulate the flow data packets. This provides more configuration flexibility and accuracy accompanied with Data Flow optimization. Same thing happens in the case of Firewalls in the traditional and SDN networks. The firewalls in the traditional networks have Limited functionalities where as in SDN more privileges are granted to the same firewall thus providing better security.

The Proposed System

The System proposed is a firewall that is based on the concept of SDN technology i.e. it will perform the basic functionalities of a firewall based on the fact that the network that it is working in is SDN based and not the traditional one. Keeping this view in record, the Firewall will be implemented in following steps

1. Creating the Test Bed for OpenDayLight.
2. Configuring the OpenDayLight Controller
3. Setting up the Firewall (Firewall setup phase).
4. Checking the functionalities(Operation phase)
5. Learning about the vulnerabilities while implementing the firewall and performing attacks such as Ip spoofing, Source routing.

Creating Test Bed for OpenDayLight

The Linux Foundation hosted a jewel OpenDayLight Project (ODL), whichis an open source SDN project aimed at enhancing SDN by offering a community-led and industry-supported framework for the OpenDayLight Controller, which has been renamed the OpenDayLight Platform. Being an Open Source commodity even including the end users and customers, it provides a shared platform for those with SDN goals to work together to find new solutions.OpenDayLight includes support for the OpenFlow protocol, but can also support other open SDN standards. The first SDN standard, as made a consideration by OpenFlow, defines the open communications protocol that allows the SDN Controller to work with the forwarding plane and make changes to the network. This gives businesses the ability to better adapt to their changing needs, and have greater control over their networks. The OpenDayLight Controller can support a modular controller framework, but can provide support for other SDN standards and upcoming protocols and also is able to deploy in a variety of production network environments. Exposing the northbound APIs is one of the basic functionality of the OpenDay-LightController(ODLC), which are then used by applications to collect information about the network, run algorithms to conduct analytics, and then use the same to create new rules throughout the network. The Controller is the actual point of communication of Firewall with the system because it is where the actual decision of packets will take place. The packets that arrive twill either be dropped or accepted by the Firewall and then forwarded by the controller to the specific Switch in the network.

Configuring the OpenDayLight Controller

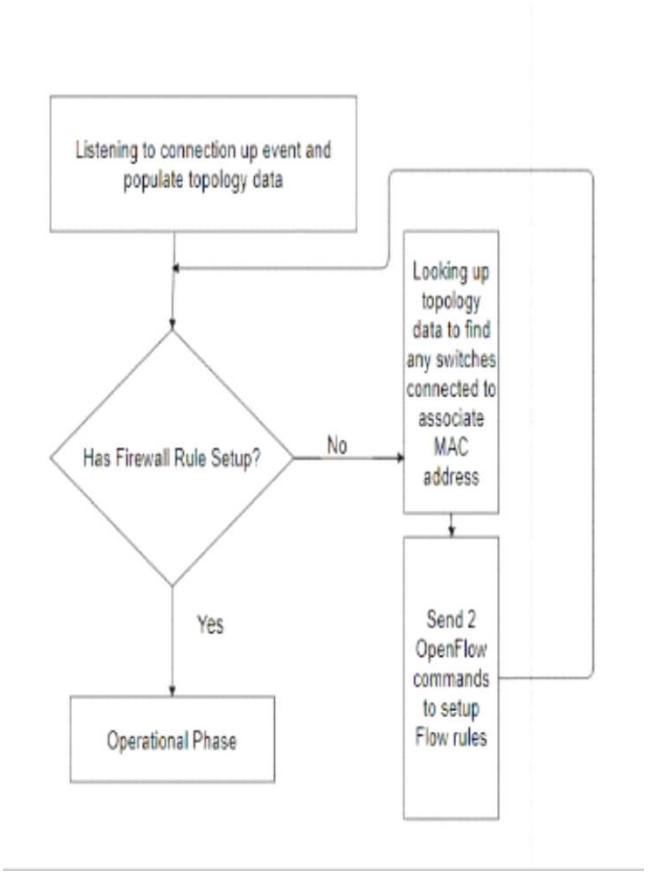
As mentioned above, the Controller is the actual point of communication of Firewall with the system because it is where the actual decision of packets will take place. Hence configuring the controller and setting up its functionality is one important step in this project. OpenDayLight is a project under Linux Foundation that is meant for the enhancement of SDN technology. Apart from this it also helps in Network Visualisation Formation.Starting from Hydrogen this controller has had 6 releases and the latest one i.e. Carbon will be used for setting up the SDN controller. OpenDayLight has a 3 layered architecture where in the 2nd layer is the OpenDayLight Controller consisting of two interfaces viz the northbound and the southbound interfaces. The northbound interface connects the ODLC to the first layer consisting of northbound APIs i.e. the applications where actual data is generated and the appropriate processing on it is done. On the other hand the southbound interface connects the controller to the 3rd layer consisting of OpenVSwitches that allow the redirection or sending of data packets to addresses provided by the controller. So in a nutshell it can be said that the ODLC works as the brain(control plane) of the SDN technology whereas the OpenVSwtiches as the data plane. For a packet filtering Firewall, this happens to be easily of lot of help as the main aim of it is to filter data packets and allow the specified ones and ODL does the same. So the ODLC will be setup on the topology created in the mininet emulator to show the actuality of SDN working and also the project. The topology setup there will have a flow of data packets in turn controlled by ODLC and forwarded by the southbound interfaces of the same. It will comprise a firewall setup by ip-tables which will perform the work of packet filtering and verify the packets that arrive to the host.

For verifying the few of the possible things which could be checked are

1. Source IP address of the packet. This is necessary because IP spoofers might have changed the source IP address to reflect the origin of packet from somewhere else, rather than reflecting the original source.
2. Destination IP Address. The firewall rules should check for IP address rather than DNS names. This prevents abuse of DNS servers.
3. IP Protocol ID.
4. TCP/UDP port number.
5. ICMP message type.
6. Fragmentation flags.
7. IP Options settings.

Setting up the Firewall (Firewall setup Phase)

This is obviously the essential step of the project as after the setting of the firewall the rules that guard the firewalls judgment are decided. It is a continuous loop in the system that ends when all the rules are decided and we dont any more rules to be added. The functioning of this step can be shown with the help of picture below



It can be explained as follows

Step1: Listening to Connection up Event: After de-ployment or every time after start-up the firewall waits for connection up event i.e. it waits network to go inline. The main reason for this is for the firewall to be able to understand the network topology accompanied by association of mac to ip addresses. This is done to understand the network that the firewall is working in which in turn helps the admin to properly decide the rules needed to be setup for the firewall.

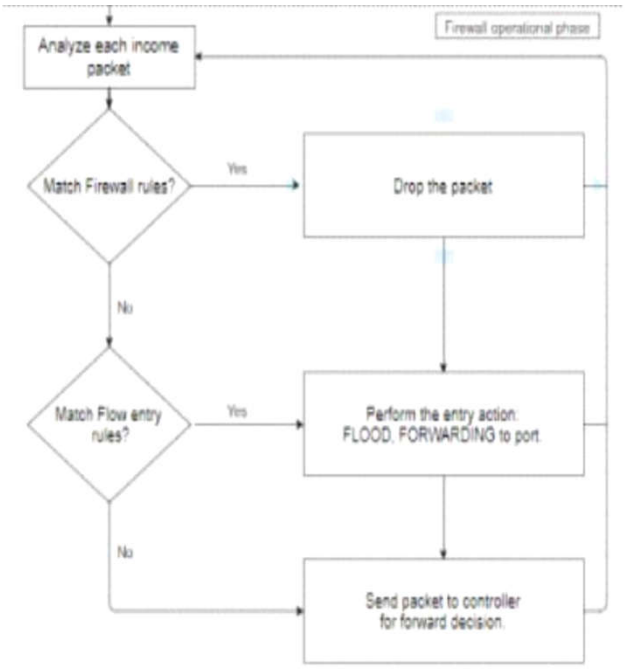
Step2: Checking for Rules: After understanding the net-work checking of rules is done. Here in checking is of two types one to check if there are any pre existing rules and if any understand them and accept or reject the rule as per requirement. Other is to decide the rule from the scratch i.e. no rule pre exists and each and every rule is decided. Irrespective the way be, proper checking is done and necessary action is taken whether or not the rules are all set. If all the required ones are set then the firewall moves to the operational phase else it moves on to use the understanding of the step1 and set up rules.

Step3: Finding switches connected to associate MAC addresses:: In here the firewall now already knows that new rules needed to be set so it uses the understanding gained in step1 to use The firewall associates MAC addresses to the different OpenVSwitches that are specially designed for SDN technology in the view of distribution of packets. Now the job of these OpenV Switches is to remember the MAC addresses allocated to them and whenever a packet arrives at their depart forward it to that specific Mac Address itself thus ensuring redirection and network handling. Now in the present scenario, the system starts allocating the MAC address To the particular switches and then switches to step 4

Step4: Sending OpenFlow Rules: This is indeed the main step. In here the firewall sends up the rules that are required and need to be followed by the Switches that are assigned and follow the controller. The rules set by the firewall over here are basic commands that contain the conditions for the data packets when they arrive at the respective switch and the switches follow it to perform the job of acceptance or rejection of the packet. This is the actual place where the firewall comes in the picture. The basic function of the firewall of setting up a wall of fire for maliciousness is performed here. With these steps being done, a continuous loop is repeated until all the rules are setup. Setting up the firewall brick by brick is done here and after the job being done it moves on the Operational Phase of the project.

Checking the Functionalities (Operational Phase)

In this step the firewall that is developed is put to function. Here in each and every functionality of the Firewall is exploited and checked upon to verify its efficiency. This step can be depicted diagrammatically as below.



Step1: Analysis: The packet that arrived at the firewall needs to be analyzed for various factors like the source address, destination address, the network subnet and the headers etc. Each and every aspect that needs to be checked for maintaining integrity of the data packet received and to be assured that the contents are error-less are performed in this very step. The end of this step marks a verified data packet and that packet itself has to face the firewall created by us. The analysis verifies the packet and if non suitable case found, the packet is thrown away into the internet pool and is not sent further for any verification. On the other hand the packet that packet analysis moves ahead to face the wall. Thus analysis proves as the door step verification or as the first line of verification that can be performed by the firewall.

Step 2: Match Firewall Rules:: The second step marks the second line of verification in the project. Here the packet that

passed the analysis is put to face the Firewall itself. The firewall uses the rules that it sets up in the Setup Phase of the project and verifies the packet accordingly. It checks for contents of the packet and check for malicious data, bugs, worms that may or may not be present in the data packet. The rules that are set help the firewall in deciding whether the packet received is suitable for the host or not. The questions like whether the packet is true or not? and whether the packet will cause harm or no harm at all? are to be answered buy the firewall and then as per the answers it decides the quality of the packet. This quality factor proves helpful in deciding the further fate of the data packet. If the firewall feels that the packet is suitable for the host then it is passed and forwarded to the respective OV Switch by the OpenDayLight controller. On the other hand if it fails to pass, then it is straight away thrown back into the internet pool and left there to travel until its TTL comes to an end. There also may exist a possibility where in the packet keeps on re-arriving at the same firewall and getting rejected a multiple times by the same procedure.

Step3: Matching Flow Entry Rules:: After passing the firewall the packet reaches the OV Switch where its fate is further decided. The OVS on receiving the packets forwards it to the ODLC where in the ODLC checks within its tables whether for the destination IP mentioned does there even exists a mac in our system. If it does then it is re directed to the associated Ovs. If it doesnt exist then an entry is made for the following purposes. They are a) There exists an IP for which we had no MAC address b)FLOOD FORWARD to the respective port where it belonged to. Irrespective the scenario, the packet is forwarded to its proper destination by the OVS or left to roam in the Internet Pool

The Attack

This is merely a testing step in the project. Inhere on purpose an illegal attack will be performed on the firewall to check for its functionalities. The attack can be of any type and the firewall must be able to defend against it.

Attacks possible and will be tried are

- IP spoofing: attacker imitating as legitimate user and performing undesired actions
- Source Routing: using special routes for packets in order to appear coming from known entities
- Port Scan: to open up covert channels or establish using open services such as telnet
- Fragmenting: fragmenting packets to bypass the firewall

Limitations

The various limitations of this whole system are

1. Economically not feasible.
2. Properly network needed to be set on a larger scale
3. Constant maintenance required.
4. Proper tools and switches need to be designed specially for this.
5. Not possible to display it manually.

CONCLUSION

Though Firewall has its own limitations, its a healthy and safe procedure for the betterment of networking in various countries round the globe. Not only our security is increased, but its various problems are solved due to it thus resulting in less preaching and disloyalty. It has a bright future and economic scope thus not only it ll generate employment but also it ll be a generator of money as well. Irrespective of the fact that with new technology new threats are released but dealing with that threats and breaking it down to the very core thus ensuring integrity is the scope of this project

References

1. Michelle Suh, Sae Hyong Park, Byungjoon Lee, Sunhee Yang Building Firewall over the Software-Defined Net-work Controller SDN Research Section, ETRI (Elec-tronics and Telecommunications Research Institute), Korea
2. Jake Collings, Jun Liu An Open Flow-based Prototype of SDN-Oriented Stateful Hardware Firewalls 2014, IEEE 22nd International Conference on Network Protocols
3. Justin Gregory V. Pena and William Emmanuel Yu Development of a Distributed Firewall Using Soft-ware Defined Networking Technology Department of Information Systems and Computer Science Ateneo De Manila University Loyola Heights, Quezon City, Philippines
4. Thuy Vinh Tran, Heejune Ahn A Network Topology-aware Selectively Distributed Firewall Control in SDN Department of Electrical and Information Engineering Seoul National University of Science and Technology Seoul, Republic of Korea
5. https://turbonomic.com/blog/on-technology/sdn-software-defined-networking-primer-and-why-we-need-sdn/
6. https://www.pressreader.com/india/opensource-for-you/20160610/282157880544330
7. https://en.wikipedia.org/wiki/SDN
8. https://www.slideshare.net/lz1dsb/why-sdn
9. http://opensourceforu.com/2016/07/implementing-a-software-defined-network-sdn-based-firewall/
10. http://sdnhub.org/resources/useful-mininet-setups/
11. https://www.techopedia.com/definition/4038/packet-filtering
12. http://securityworld.worldiswelcome.com/packet-filtering-firewall-an-introduction
13. https://www.sdxcentral.com/sdn/definitions/sdn-controllers/opendaylight-controller/
14. http://www.necoma-project.eu/m/filer-public/18/30/1830b40c-a2e2-4a0b-b83e-af8b2d846e61/imt-zhang-comnet2015.pdf
15. https://www.ietf.org/proceedings/91/slides/slides-91-i2nsf-0.pdf

How to cite this article:

Yash Bajaj *et al* (2018) 'Firewall Based on the Concept of Sdn', *International Journal of Current Advanced Research,* 07(2), pp. 9960-9963. DOI: http://dx.doi.org/10.24327/ijcar.2018.9963.1665