iJRASET

International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 4 Issue: XI Month of publication: November 2016

DOI:

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

Volume 4 Issue XI, November 2016
ISSN: 2321-9653

International Journal for Research in Applied Science & Engineering Technology (IJRASET)

Model Survey on Policy Making for Software Defined Networks

Anshuman Kumar[1], Abhilash Kamtam[2], Prof. U. C. Patkar[3] (Guide)
[1]Department of Computer Engineering, BVCOEL, Pune-412115, India
[2]Department of Computer Engineering, BVCOEL, Pune-412115, India
[3] HOD Department of Computer Engineering, BVCOEL, Pune-412115, India

Abstract - As the nature of threat is evolving day by day so it very important that network defence method should also evolve. This lead in increased demand of Software Defined Network(SDN) and OpenFlow, the policy based network management and security. It is very difficult task to manually configure multiple devices in a network so it is being replaced by automated approach where a software based, network controller handles the configuration of devices. We propose OpenSec, an OpenFlow based security framework that allows a network security operator to create and implement security policies in human-readable language. SDN consist of decoupling the control and data planes in a network. It relies on the fact that the simplest function of a switch is to forward packets according to a set of rules. OpenSec converts security policies into a series of OpenFlow messages needed to implement such policy. Different techniques like Intrusion detection, spam detection, data packet inspection are having performance issues regarding space and time. So, this paper concentrates on analysing different techniques and to find a proper way to improve Policy based Security over a network.

Keywords: Software Defined Network (SDN), Intrusion Detection, OpenFlow, Network Security, OpenSec Framework.

I. INTRODUCTION

As the number of networks are increasing day by day so it is important to make a network more secure and reliable [6]. It is very important to provide security to all the components of the network. A recent approach to programmable networks is the Software Defined Network(SDN) architecture [4]. In SDN we move away from manual configuration of multiple devices, we get closer to automated approach of configuring each device in the network with security policies and rules. Due to the Centralized nature of control plane more complex network-control application can be implemented at the controller. OpenFlow is a protocol that standardizes how an SDN controller communicates with network devices [4]. OpenFlow provides a protocol to program the flow table of routers and switches.

In this survey paper, we propose OpenSec, an OpenFlow based security framework that allows a network administrator to create and implement security policies in human-readable languages. OpenSec consists of a software layer running on top of the network controller and multiple external devices that perform security services (such as firewall, encryption, spam detection, deep packet inspection (DPI) and others) and report the results to the controller [5]. The main purpose of OpenSec is to allow network operator to define policies for the flows. Policies include how to react on any intrusion or any malicious content is found in the network. The reaction can be in three types: to alert only, or to quarantine traffic, or to fully block packets from specific source node. We think that there is still room for further improvement and also most of the work should be done by processing unit with minimal to no human intervention. OpenFlow network has special capabilities. For example, it is possible to control multiple switches and routers from single controller.

OpenFlow offers great feature but also faces many challenges. The availability of network depends on single controller at a particular time this gives room to scalability as well as availability problems. Since all the network information and policies are contained in one single server so this might become a security issue. Protecting the end-point in any network is very important. Any non-sanitized end user connected to a network can become harmful threat to the network [6]. This non-sanitized end user then becomes the weakest link in the network and can easily by targeted by an attacker [6].

In this paper, we describe SDN and alternative standards. Also, we explain how implementing machine learning and neural networks are helpful in detecting malicious contents in networks and also helpful in taking quick reactions if any threat is found. We also explain how OpenFlow has received attention in SDN technologies.

We begin by giving brief introduction about Artificial Intelligence, and detailed information on OpenFlow, Software Defined

©IJRASET: All Rights are Reserved

Network, and OpenSec framework and all Related Work in Section II. We conclude in by providing Conclusion and Future research work in Section III.

II. RELATED WORK

A. *Artificial Intelligence And Intrusion Detection*

Artificial Intelligence can be defined in two ways: (i) the field of science that studies the synthesis and analysis of computational agents that act intelligently. (ii) Creating a system in such a way that there will be less or no human intervention in solving any type of problems or tasks put forward to that system irrespective of its complexity [1]. In implementing the proposed model the second definition stands true which supports the ultimate aim of the paper.

There exist various sub-sections of AI but the mostly used methods like machine learning and neural networks best suits for implementing the paper. The above-mentioned AI methods are actually the computational models that are inspired by the biological immune system which are capable of adapting the changing environment and dynamically learning on its own. Here, the immune systems are responsible to detect intruders and accordingly dealing with them.

Artificial Neural Networks (ANNs) [1] which basically works same as the human brain in which all the logic related work and instantaneous simulation is controlled by so called neurons. Here, the same terminology has to be implemented which consists of artificial neurons which will learn and solve problems accordingly when combined together. Neural Networks have ability to learn, process distributed information, self-organize and adapt, are applicable to solve problems that require considerable precision, conditionality and ambiguity at the same time [1]. This also provides an additional functionality of parallel learning and decision-making with high speed, which will later enable them in learning pattern recognition and selection of responses to attacks.

B. *Open flow And Software Defined Networking*

1) *Open flow* **-** *A Mechanism:* OpenFlow was proposed to standardize the communication between the switches and the software-based controller in an SDN architecture [4]. OpenFlow networks distinguish themselves from legacy network infrastructures by dramatically rethinking the relationship between the data and control planes of the network device. The authors of the OpenFlow identified that it was difficult for the networking research community to test new ideas in current hardware. This happened because the source code of the software running on the switches were unable to be modified and the established network infrastructure has been “ossified” [2], [4], as new network ideas cannot be tested in realistic traffic settings. On analysing the flaws and the limitations, the authors provided a standardized protocol to control the flow table of a switch through software. OpenFlow provides a means to control a switch without requiring the switch-vendors to expose the code of their devices.

For an OpenFlow switch, the data plane is made programmable, where flows are dynamically specified within a flow table. The flow table contains a set of flow rules, which specify how the data plane should process all active network flows. In short, OpenFlow’s flow rules provide the basic instructions that govern how to forward, modify, or drop each packet that traverses the OF-enabled switch [3], [5]. The switch’s control plane is simplified to support the OpenFlow protocol, which allows the switch to communicate statistics and new flow requests to an external OpenFlow network controller. In return, it receives flow rules that extend its flow table ruleset. An OF controller is situated above a set of OF-enabled switches, often on lower-cost commodity hardware. It is the coordination point for the network’s flow rule production logic, providing necessary flow rule updates to the switch, either in response to new flow requests or to reprogram the switch when conditions change. As a controller, may communicate with multiple OF switches simultaneously, it can distribute a set of coordinated flow rules across the switches to direct routing or optimize tunnelling in a way that may dramatically improve the efficiency of traffic flows.

OpenFlow networks have specific capabilities. For example, it is possible to control multiple switches from a single controller [4]. It is also feasible to analyse traffic statistics using a software. Forwarding the related information can be updated dynamically as well and different types of traffic situations can be abstracted and managed as flows. These capabilities have been exploited by the research community to experiment with innovative ideas and propose new applications. Ease of configuration, network management, security, availability, network and data centre virtualization and wireless applications are those that have been investigated the most using OpenFlow. They have been implemented in different environments, including virtual or real hardware networks and simulations. Researchers have also focused on evaluating the performance of OpenFlow networks and on proposing methods to improve their performances.

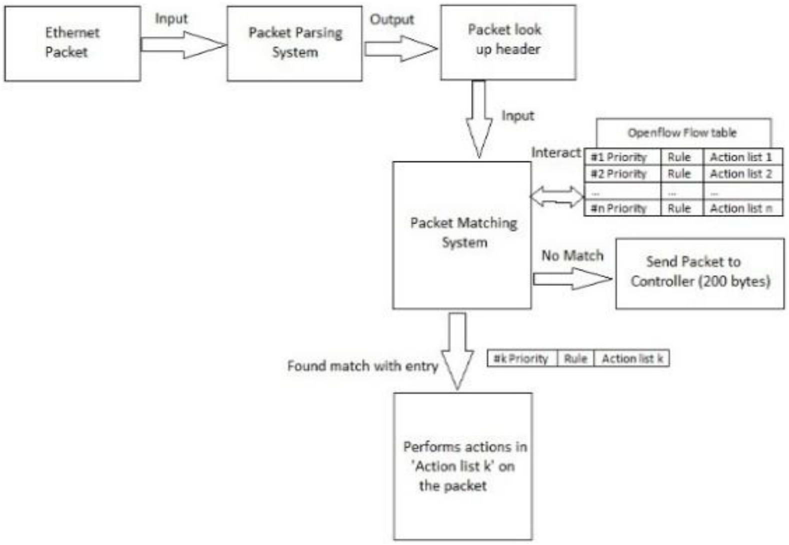

Ethernet Packet
Input
Packet Parsing System
Output
Packet look up header
Input
Interact
Openflow Flow table
#1 Priority | Rule | Action list 1
#2 Priority | Rule | Action list 2
... | ... | ...
#n Priority | Rule | Action list n
Packet Matching System
No Match
Send Packet to Controller (200 bytes)
Found match with entry
#k Priority | Rule | Action list k
Performs actions in 'Action list k' on the packet

Fig. 1 Steps involved in processing and forwarding a packet in OpenFlow.

Figure [4] mentioned illustrates the mechanism of the OpenFlow, that is, the flow of transmission of packet and the way it is been processed before it is been delivered at its destination point.

2) *Software Defined Networking:* Software Defined Networking (SDN) consists of decoupling the control from the data plane. Forwarding devices become simpler and transmit packets based on the forwarding table, which is manipulated by a software-based, logically centralized network controller [3]. SDN-based networks have several capabilities that can be exploited in the context of network security. First, the controller is network-aware [5]. This allows it to gather information from multiple locations of the network and to react accordingly. Second, SDN greatly simplifies dynamic updating of traffic rules. Software running on the controller can automatically modify the forwarding table of any network device, based on the observation of current traffic [4], [5]. OpenFlow [2] is the most commonly deployed SDN protocol. It standardizes the communication between a software based controller and layer 2 switches through the OpenFlow channel. The SDN architecture provides different features which enables the user to modify according to their needs. This comprises of the following features:

a) Directly Programmable.
b) Agile.
c) Centrally Manageable.
d) Programmatically Configurable.
e) Open standards-based and vendor-neutral.

3) *Opensec Framework:* OpenSec [2], [4] aims at allowing the network operators to create and implement the required network security policies. The policies include a description of the flow, a set of security services that should be applied to observed traffic and a security level for automatic reaction in case of detecting malicious traffic. The processing units provide specific security services such as encryption, denial of service detection, deep packet inspection and many more. The OpenSec framework consists of various components that are essential in order to have an uninterruptable communication between the software and the network. The essential components are [2], [3], [4]:

a) *Policy Parser:* The policy parser converts the policy definition file into data and structures that can be processed by OpenSec. Policies used can be defined using the keywords such as Flow, Service and React. One of the main goals of OpenSec is to allow operators to create very simple policies to control the network.

b) *Policy Checker:* The policy checker maintains the information about all policies that are currently enforced and checks that new policies do not conflict with existing ones. To have a clear picture let's take an example, suppose there are two policies that use exactly the same matching fields and same values, on implementing such policy will not be able to execute, since one of the two matches will occur at least once. Hence, by having a reference to all policy objects in the system, the checker can easily compare incoming policies with existing ones to verify them.

c) Processing unit manager: The processing unit manager collects all the registrations and ends up with a list of services and the location in the network where they can be found. In this model, the controller maps a service id (DPI, IPS) to a switch ID, an input port and an output port. This is all the data needed by OpenSec to manipulate the flow table of the devices in order to re-route traffic to the processing units.

d) Policy Implementer: Implementing a policy implies re-routing matching traffic to one or more security processing units. Here, OpenSec first queries the processing unit manager for the switch id where the processing unit is connected. In that switch, a rule is provided to duplicate traffic so that a copy is created and is sent to the input interface of the processing unit for further processing.

e) Security Event Processor: One of the most important features of OpenSec is the automatic reaction to security alerts. Usually, a network operator will react to an alert by either ignoring it or blocking the source of the suspicious traffic. In OpenSec, the network operator can define such a reaction in advance using three possible solutions: alert, quarantine or block. In OpenSec, the controller remains listening to alerts and reacts to those alerts by deciding how to modify the traffic rules. If the specified reaction is alert, forwarding rules are not modified and the network administrator is notified by e-mail or a message. In order to quarantine the traffic, a processing unit which is logging all the traffic is attached to one of the switches. OpenSec updates the forwarding table of the switch so that matching traffic is forwarded to the quarantine unit instead of to the host. The quarantine unit logs all incoming traffic so that a network operator can then analyse the data. Finally, if the policy requires blocking all traffic, then the forwarding rules of involved switches are modified so that matching traffic is dropped.

III. CONCLUSION AND FUTURE SCOPE

Above presented article is systematic survey done on Software defined networks, defining problem statement. Future work is to design and implement software defined Network framework and achieved policy based software defined networks.

REFERENCES

[1] Abhilash Kamtam, Anshuman Kamar, Prof. U. C. Patkar, “Artificial Intelligence approaches in Cyber Security”, April 16 Volume 4 Issue 4, International Journal on Recent and Innovation Trends in Computing and Communication (IJRITCC), ISSN: 2321-8169, PP: 05 – 09.

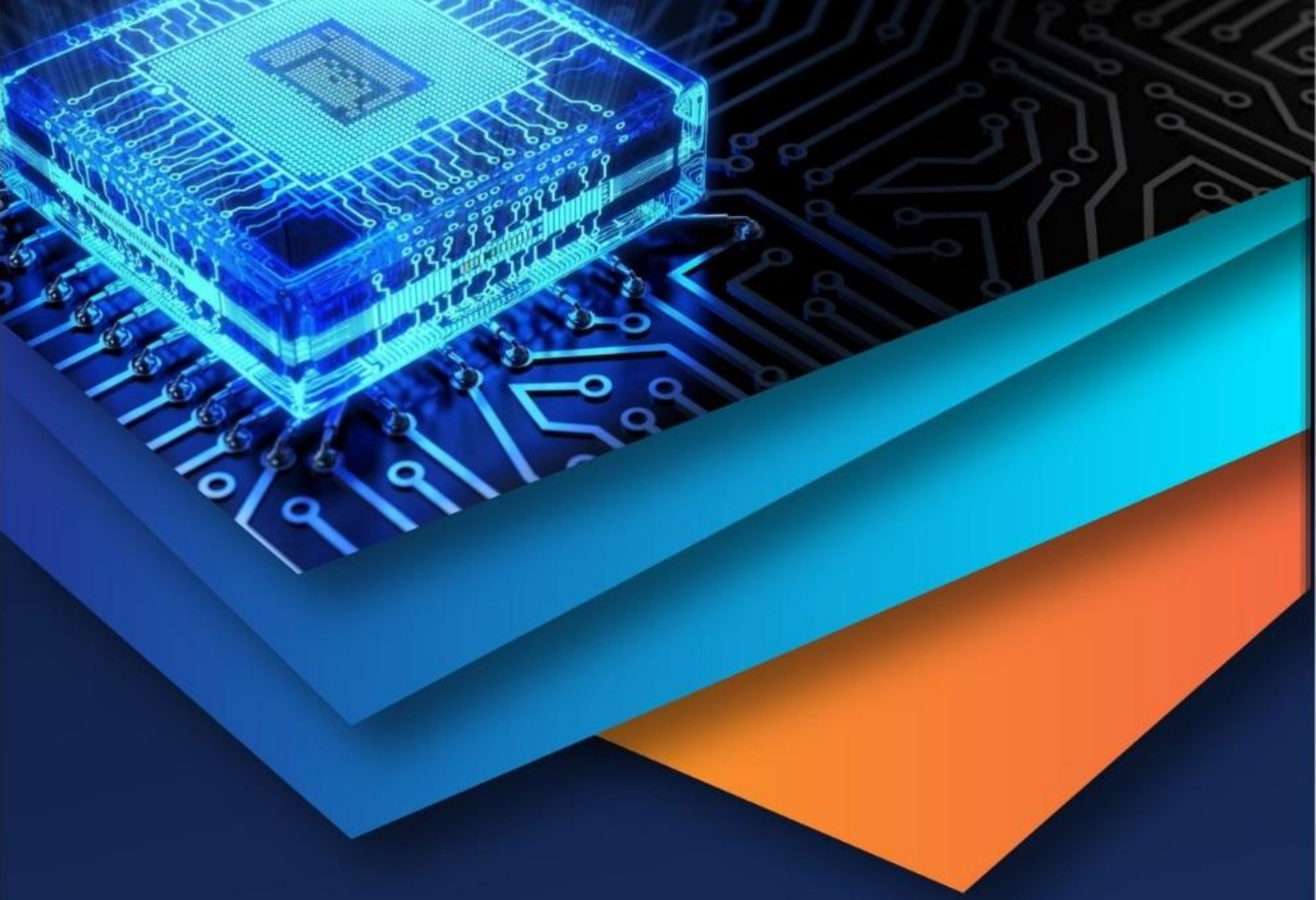
[2] Adrian Lara, Byrav Ramamurthy, “OpenSec: Policy-based Security Using Software-defined Networking”, DOI 10.1109/TNSM.2016.2517407, IEEE Transactions on Network and Service Management.

[3] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, “Openflow: enabling innovation in campus networks,” SIGCOMM Comput. Commun. Rev., vol. 38, no. 2, pp. 69–74, 2008.

[4] Lara, Adrian; Kolasani, Anisha; and Ramamurthy, Byrav, "Network Innovation using OpenFlow: A Survey" (2013). CSE Technical reports. Paper 159.

[5] Lara, Adrian and Ramamurthy, Byrav, "OpenSec: A Framework for Implementing Security Policies using OpenFlow" (2014). CSE Conference and Workshop Papers. Paper 272.

[6] Anshuman Kumar, Abhilash Kamtam, Prof. U.C. Patkar, “Self-Defending Approach of a Network”, Vol: 3, Issue: 4, April 2016, IRJET e-ISSN: 2395-0056 p-ISSN:2395-0072.



doi crossref

10.22214/IJRASET

INDEX COPERNICUS

45.98

ISRA JIF

IMPACT FACTOR: 7.129

TOGETHER WE REACH THE GOAL

IMPACT FACTOR: 7.429

ISI

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)