International Journal of Computer Science and Engineering Communications,
Volume.5, Issue.3 (2017): Page.1566-1573
www.ijcsec.com

A Novel Approach to Dynamic Policy Based Security in SDN: A Survey

Veena S[1] and Manju R[2]
[1] *PG Scholar,* [2] *Assistant Professor,*
[1,2] *Department of Information Technology,*
[1,2] *Government Engineering College Idukki, India*
veenadasbs@gmail.com[1] *,manjurajanv@gmail.com*[2]

Abstract: *Software Defined Networking (SDN) is a proposal to networking world for a global administration to manage network services and also to provide an abstraction of networking elements. Most of the security innovations in SDN are bound to an OpenFlow security framework which creates and implement security policies when malicious traffic is detected, in a flexible way. Existing methodologies convert policies in to human readable form, which reduces the complexity to use it. In this article, security in SDN is analyzed, which considers the research works and industry growths in this area. The challenges to securing the network from the sticky attacker are mentioned and the integrated approach to the security architecture that is essential for SDN is also described.*

Keywords *- SDN, OpenFlow, OpenSec.*

I. Introduction

The basic idea of software-defined network discusses about decoupling the data plane and control plane. The network components in the data plane will communicate with the control plane through OpenFlow protocol, as in Fig.1, which act as an interface between data and control plane. SDN operators uses OpenFlow for remote access of data set from data plane.

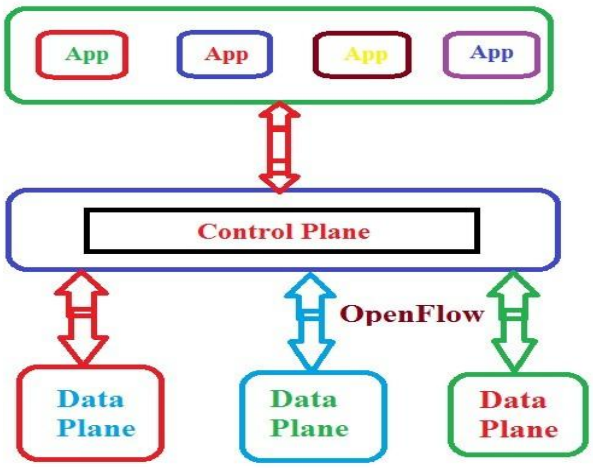

Fig. 1. SDN Overview

SDN concept of central control over the entire network may provide several features such as flexibility, scalability, portability, abstraction, orchestration etc. Vulnerability of central control is the most challenging task in SDN. Security of central controller is the only choice to eliminate such

vulnerabilities. One of the noticed security framework is OpenSec [4], an OpenFlow-based security framework. Suppose a campus operator needs to mirror incoming web traffic to an intrusion detection system (IDS) and e-mail traffic to a spyware detection device. Suppose the IDS detects malicious traffic and the sender needs to be blocked from accessing the network. Instead of having the operator configure the edge router to manually disable access to the source, operator can lock the block the sender automatically using OpenSec. It allow the operator to write a high-level policy to achieve security, instead of having to manually configure each device.Because OpenSec gives an abstraction of the network, the operators mainly focus on creating simple and human readable security policies by configuring all the devices to achieve the desired security. OpenSec software layer runs on top of the network controller as well as multiple external devices that act as security services (such as firewall, IDS ,encryption, DPI, etc) and report the results to the controller.

The policies includes description of the flow, list of security services that apply to the flow and how to react, in case of suspicious traffic is found. The reaction can be alert only, or quarantine traffic or even block all packets from a specific source. The design requirements of OpenSec shown in Fig.2 as follows. First, policies should be human-readable. Second, data plane traffic should be processed by the processing units (network devices, middle boxes or any other hardware that provides security services to the network). When the controller manages entire tasks, it becomes a bottleneck and the solution does not scale well. In OpenSec [4], the controller is subject to a low workload and is responsible for implementing policies and modifying forwarding rules which is based on the security alerts received from the processing units. Third, the framework should react to security alerts automatically to reduce human intervention, when suspicious traffic is detected.

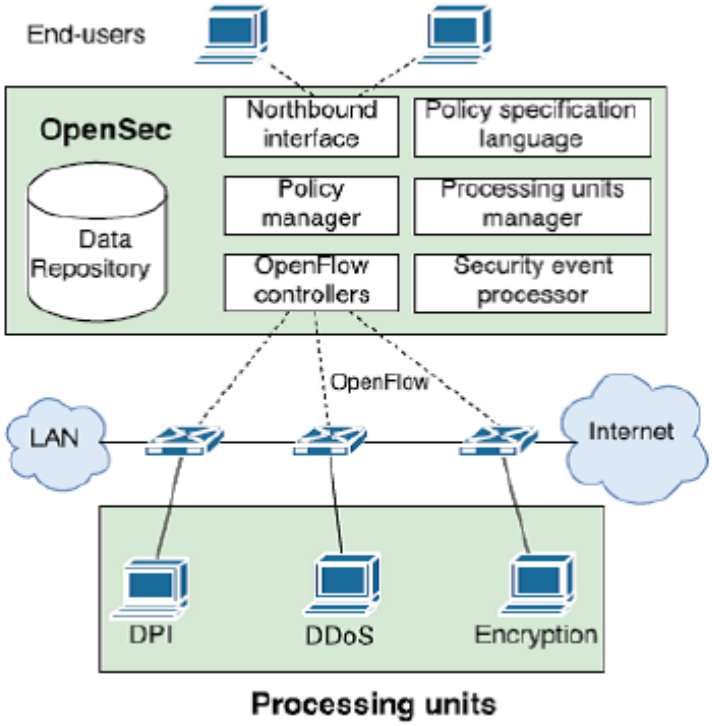

Fig. 2. The OpenSec Framework [4]

II. Related Works

CLOUDWATCHER[5], provides monitoring services which focus on large and dynamic cloud networks. This framework automatically detours network packets to be inspected by preinstalled network security devices. The benefits are: (i) it controls network flows to ensure that all needed network packets are tested by some security devices and (ii) it provides a simple policy script language to people for easily accessing services. CLOUDWATCHER fluctuate the routing paths for network flows, and it generate the flows to transmit through network nodes where security devices are located. To construct a security monitoring service for any individual security concern, a cloud administrator may generate a security policy, which contain of 2 fields: (i) flow condition, which represents the flow to be probed, (ii) device set, which demonstrate necessary security devices for investigation. The security policy is described in a SLIpolicy script as in Fig.3.

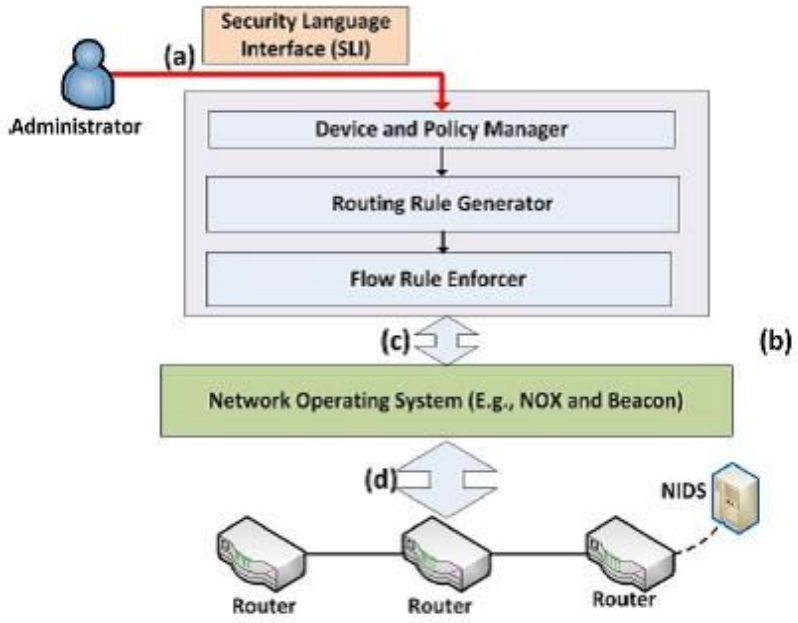

Fig. 3. CloudWatcher Architecture [5]

FRESCO [6], an OpenFlow enabled security application development framework intended to enable the quick design, and modular structure of OF-enabled detection and mitigation modules. FRESCO present a scripting API that motivate security practitioners to code security monitoring and threat detection in modular libraries. These modular libraries discusses about the elementary processing units in FRESCO, and it is also shared and combined to provide complex network security applications. Which reduce the complexity of defense services in the development and deployment of OpenFlow networks. FRESCO propose minimal overhead and that it enables rapid creation of simple security functions with minimum lines of code. Illustration of FRESCO Script as shown in Fig.4.

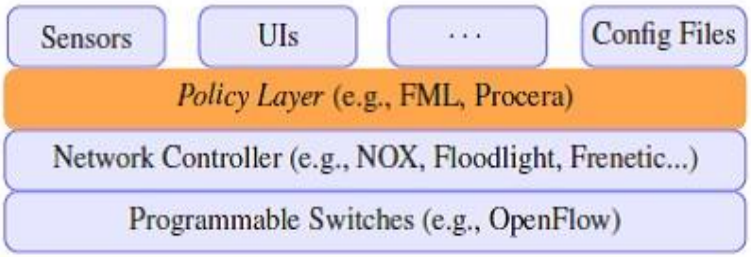

Fig. 4. Operational illustration of running FRESCO script [6]

Procera [1], a model architecture for software-defined networking (SDN), it contains a declarative policy language based on the action of functional reactive programming method. They implement the functional unit with both signals relevant for expressing high-level network policies in a different network settings, containing home and enterprise networks, and a group of constructs significant temporal queries over event streams that occur simultaneously in network policies. Procera also incorporates events that generate from sources other than OpenFlow switches, allowing it to disclose policy that reacts to several facts such as user authentications, time of day, using bandwidth, or server load. Procera is also expressive and extensible, so users can simply extend the language by adding recent constructs. The key factors of the Procera policy language are listed as (1) a core language followed by functional reactive programming, (2) event comprehensions to evaluate event streams, (3) windowing and aggregation signal functions, (4) the use of function values to represent high-level policy.System architecture of Procera as shown in Fig.5.

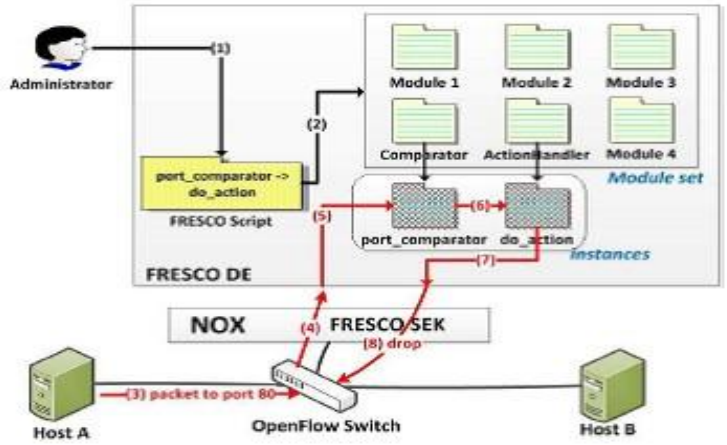

Fig. 5. Procera System architecture [1]

A policy server [11], which provides centralized management of packet voice gateways and soft switches in next generation circuit and packet telephony networks. The policies running in the policy server are specified using a domain independent policy description language (PDL). They developed a policy based management system that is being used to do Operations, Administration, Maintenance and Provisioning (OAMP) in carrier-grade communication networks.

Policy analysis techniques [2] have no dependencies with any of the applications or they have some correlation to policies with specific purposes. It contain analysis techniques to detect redundancy and incompleteness of policies. The present techniques used to identify modality conflicts such as obligations, which not be fulfilled because of the lack of authorizations, and also to identify circular dependencies of obligations. They proposed obligation notion, the execution of an obligation can invoke the execution of another obligation. A subject that run an obligation also needs an authorization to execute the action needed by the obligation. The access control policy also provides such authorization which may require the execution of some other obligations. A central policy enforcement point (PEP) and multiple policy decision points (PDP) are the modules in system architecture shown in Fig.6. , which supports collaborative access control. While considering a single PEP and a single PDP are entirely different from the typical access control policy framework.

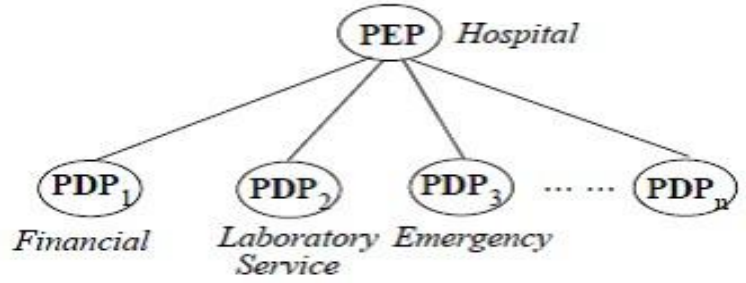

Fig. 6. System architecture [2]

In SDN Enhanced Campus Network Authentication and Access Control System [8], they discuss about controlling traffic flow in standard hierarchical networks, which is one of the possible ways to gradually implement by applying OpenFlow driven SDN architecture and commodity access switches. The controller module is responsible for controlling and managing all nodes in the network, Their design consists of : 1) to register and authenticate the switches to the controller, 2) to authenticate the hosts and to bind them to the switches (and ports), 3) to provide the authentication of users, 4) to manage data flows and users/hosts mobility, were developed.

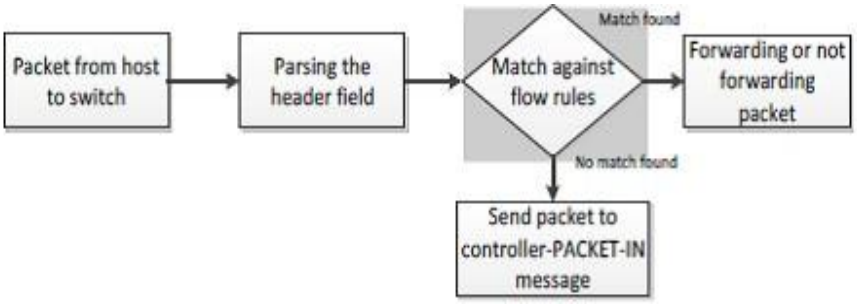

Fig. 7. Packet in message [7]

In FlowSec: DOS attack Mitigation Strategy on SDN Controller [7], examines the consequence of DOS attack on controller plane. FlowSec reduces an attack on the controller bandwidth by enforcing a rate limit on the number of packets issued to the controller. Also, it avoids an attack on the control plane bandwidth by restricting the number of packets sent to the controller. When an attacks occurs, the OpenFlow controller is informed to issue a control that drops traffic from the foreign attacker. After that flow is

monitored to verify that the control has come into effect. The processing of arrived packet as shown in Fig.7. Another policy management framework otential conflicts among the heterogeneous policy rules. In addition, it keeps consistency between the flow table rules and the on-demand changes in policy rules in the application layer. This framework consists of three control functions such as Trust Verify, Policy Conflict Resolve and Policy Consistency Check. The Trust Verify function verifies the trustworthiness of the network applications to understand the potential compromised applications and applies appropriate control.

In OpenFlow: Enabling Innovation in Campus Networks [3], focus is on an Ethernet switch, with an internal flow-table, and a standardized interface to add and remove flow entries. The data path of an OpenFlow Switch contains a Flow Table, and an action associated with each flow entry. The group of actions supported by an OpenFlow Switch is scalable, but below they illustrate a minimum requirement for all switches. An OpenFlow Switch consists of at least three parts listed as : (1) A Flow Table, which contain an action associated with each flow en- try, to informal the switch how to process the flow, (2) A Secure Channel that associated with the switch to a remote control process (called the controller), permitting commands and packets to be exchange between a controller and the switch using (3) The OpenFlow Protocol, which is used to provides an open and conventional way for a controller to communicate with a switch.

OpenSec[10] , an OpenFlow-based security framework which assists a network security operator to create and implement security policies written in human-readable language format. Using OpenSec, the user can specify a flow in terms of OpenFlow matching fields, describe which security services must be enforced to that flow (DPI, intrusion detection, spam detection, etc) and define security levels that specify how OpenSec reacts to it when malicious traffic is encountered shown in Fig.8. The policies consists of description of the flow, a set of security services that must be added to such traffic and a security level applied for automatic reaction in case of detecting malicious traffic. The processing units delivers specific security services such as encryption, DOS attack detection, DPI or any other.

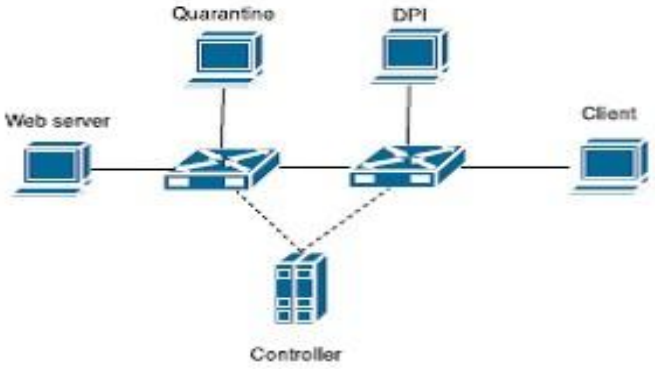

Fig. 8. Simple topology with DPI and quarantining [10]

III. Comparative Study

Comparative study of related works on OpenSec : Policy Based Security Using Software Defined Networking are shows in Table1.

Table.1 Comparative Study

TITLE	MECHANISM	ADVANTAGES	DISADVANTAGES
CloudWatcher: Network security monitoring using OpenFlow in dynamic cloud networks (or: How to provide security monitoring as a service in clouds?)	CloudWatcher	Monitoring services for large and dynamic clouds	Some cases, it may not generate routing paths
FRESCO: Modular composable security services for software defined networks	OF-enabled detection framework	Choose an Open source network anomaly detection sys- tem and then replicate identical functionality	Policy parsing is not time-efficient
Procera: A language for high level reactive network control	Functional reactive Programming framework	Enforcing network policies	Policies are little complex
Policy evaluation for network management	PE algorithm	Enforcing policies in complex network	Lack of automation process
Analysis of privacy and security policies	Analysis techniques	Detect modality conflicts	Vulnerability of central policy enforcement point
SDN Enhanced Campus Network Authentication and Access Control System	AAA (authentication, authorization and accounting) scheme	Capable of controlling and managing all nodes in the network	Vulnerability on the controller
FlowSec: DOS attack Mitigation Strategy on SDN Controller	FlowSec framework	mitigate a DOS attacks effect on the network	messaging overhead that must occur in order to collect the statistics
OpenFlow: Enabling Innovation in Campus Networks	OpenFlow	Simplify realworld traffic set- tings	Commercial switches and routers do not typically provide an open software platform
A Novel Secure and Efficient Policy Management Framework for Software Defined Network	policy management framework	on-demand changes in policy rules	Policies are little complex
OpenSec: A Framework for Implementing Security Policies using OpenFlow	OpenFlow-based security framework	human-readable policy, automatically react in case of detecting malicious traffic	Vulnerability of Openflow

IV. Conclusion

OpenFlow security is one of the challenging task in SDN. Abstraction of network operations may occur in SDN with the help of OPenFlow. So, an OpenFlow based security framework is implemented in control layer, which named as OpenSec, a policy based security framework. OpenSec acts as a virtual layer between the user and the OpenFlow controller and automatically converts security policies into a set of rules that are pushed into network devices. OpenSec also allows network operators to specify how to automatically react when malicious traffic is detected. The main advantage of OpenSec with respect to Procera is simplicity. The time needed by OpenSec to translate policies into OpenFlow messages achieved by CloudWatcher is comparatively high.

REFERENCES

[1] A. Voellmy, H. Kim, and N. Feamster(2012), *Procera: A language for highlevel reactive network control*, in Proc. Workshop Hot Topics Softw. Defined Netw. (HotSDN), Helsinki, Finland, Aug. 2012, pp. *4348*.

[2] E. Bertino et al.(2009), *Analysis of privacy and security policies*, IBM Journal of Research and Development, *pp 1-18*.

[3] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner (2008), *OpenFlow: enabling innovation in campus networks*. SIGCOMM Comput. Commun. Rev. *38, 2 (March 2008), 69-74*.

[4] Adrian Lara and Byrav Ramamurthy(2016), *OpenSec: Policy-Based Security Using Software-Defined Networking*, IEEE Transactions On Network And Service Management, *VOL. 13, NO. 1*,MARCH 2016 .

[5] S. Shin and G. Gu(2012), *CloudWatcher: Network security monitoring using OpenFlow in dynamic cloud networks (or: How to provide security monitoring as a service in clouds?)*, in Proc. 20th IEEE Int. Conf. Netw. Protocols (ICNP), Austin, TX, USA, Oct. 2012, *pp.16.*

[6] S. Shin, P. Porras, V. Yegneswaran, M. Fong, G. Gu, and M. Tyson(2013), *FRESCO: Modular composable security services for software defined Networks*, in Proc. Netw. Distrib. Syst. Sec. Symp. (NDSS), San Diego,CA, USA, Feb. 2013, *pp. 116.*

[7] Kuerban et.al. (2016), *FlowSec: DOS attack Mitigation Strategy on SDN Controller*, 2016 IEEE International Conference on Networking, Architecture and Storage (NAS).

[8] Feliksas Kuliesius , and Vainius Dangovas(2016), *SDN Enhanced Campus Network Authentication and Access Control System*, 2016 Eighth International Conference on Ubiquitous and Future Networks (ICUFN).

[9] Bata Krishna Tripathy, Ananta Gopal Sethy and Padmalochan Bera, *A Novel Secure and Efficient Policy Management Framework for Software Defined Network*, 2016 IEEE 40th Annual Computer Software and Applications Conference.

[10] A. Lara and B. Ramamurthy(2014), *OpenSec: A framework for implementing security policies using OpenFlow*, in Proc. IEEE Globecom Conf., Austin,TX, USA, Dec. 2014, *pp. 781786.*

[11] R. Bhatia, J. Lobo, and M. Kohli, *Policy evaluation for network management*, in Proc. IEEE INFOCOM, Mar. 2000, *vol. 3, pp. 11071116*.