A Study on SDN security enhancement using open source IDS/IPS Suricata

1st Kiho Nam
Dept. of Computer Science & Engineering,
Konkuk University
Seoul, Korea
namgeo@konkuk.ac.kr

2nd Keecheon Kim
Dept. of Computer Science & Engineering,
Konkuk University
Seoul, Korea
kckim@konkuk.ac.kr

***Abstract*—Software Defined Network (SDN) is a next-generation networking technology that transforms existing closed network environments based on individual network vendors technology into a software-based, flexible, centralized management environment that is programmable based on simplification through network abstraction. For this reason, unlike traditional networks, SDN has some strengths over some security issues. However, most of the existing network security problems and vulnerabilities exist in the SDN environment. And various attacks targeting this are occurring. This paper examines how to implement network security functions using SDN technology for these security problems. In addition, this paper proposes a structure for enhancing the security function of SDN by using existing open source IDS / IPS software Suricata.**

***Keywords*—SDN, Security, Network, Suricata, IDS, IPS,**

I. Introduction

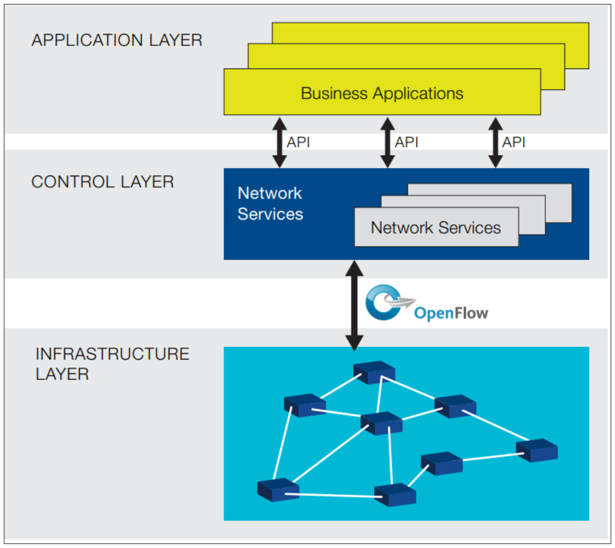
Recently, Software Defined Network(SDN) is an issue in the field of networks. SDN is a next-generation networking technology that transforms existing closed network environment based on fixed hardware and software centered on network hardware vendors into a flexible, software-based, centralized management environment that simplifies programming by network abstraction. In the SDN environment, the network administrator can control the communication functions performed on the data plane according to various management purposes of the network by programming the control planes by software. With this SDN, general network control, routing, complex operation management functions can be easily handled, and advanced functions such as load balancing and QoS can be easily implemented. Various technologies and functions in a flexible SDN environment can be used in many areas of network security. Because of the flexible nature of SDN, which can be controlled by software, structurally linking SDN with various existing open source software security solutions can enhance SDN security. However, there are not many cases in which the technology of SDN is analyzed and implemented from the viewpoint of network security. Also, there are not many studies to enhance security by linking SDN with existing open source security solutions. This paper suggests how to implement network security functions using SDN technology in SDN environment. In addition, we propose a method to enhance the security of SDN by using Suricata which is open source IDS / IPS software in SDN environment.

II. Background & Related Works

A. SDN(Software Defined Network)

Software-defined networks abstract existing closed networks centered around fixed hardware and software. Then, it is modeled by a three-plane structure composed of a data plane, a control plane, and an application plane.

Fig. 1. Architecture of Software Defined Network



The data plane is a layer that is controlled through a specific interface of the SDN. The data plane is responsible for the transmission of the data flow. The control plane is a layer that controls the flow of data. The control plane determines whether to route, forward, or reject data flows through applications and network services. In addition, the control plane transfers the operations of the data plane to the application layer in the form of an API. Finally, the application plane performs various functions of the network using the APIs provided by the control plane. They support device / resource abstraction, control abstraction, and service abstraction. SDN defines an open interface called the southbound interface and a northbound interface that connect these three planes. The southbound interface connects the data plane and the control plane, and the northbound interface connects the control plane and the application plane. Figure 1 is a simplified illustration of the conceptual structure of SDN [1].

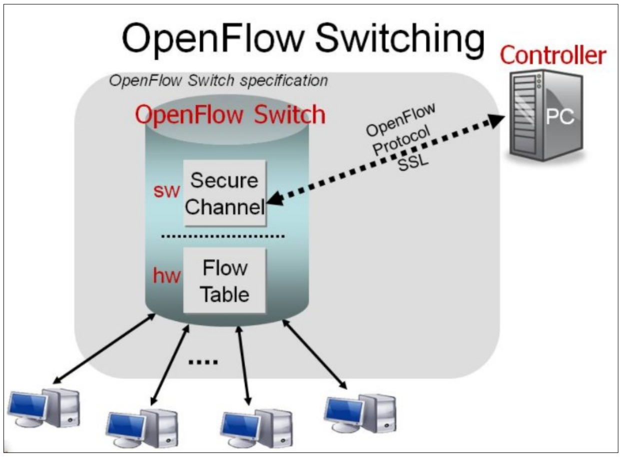
B. OpenFlow

OpenFlow started in the project [2] to implement the concept of SDN centered on Stanford University. Currently,

978-1-5386-5041-7/18/$31.00 ©2018 IEEE

many vendors are participating in ONF(Open Networking Foundation) and doing research.

Fig. 2. OpenFlow Switching



OpenFlow [3] is the Southbound interface specification between controller and network device. It is implemented as open source software, providing FloodLight [4] as a controller and Open vSwitch [5] as a switch.

III. Proposed SDN Security Method

In existing network security environments, firewall, network scan, abnormal traffic detection, intrusion detection and intrusion prevention technologies are important and indispensable. And it requires a lot of hardware and high cost to implement. This is not so different in the SDN environment.

	SDN (OpenFlow)	Suricata IDS/IPS	Automated intrusion prevention	Mirroring
firewall	possible	Unnecessary	unnecessary	unnecessary
network scan	possible	Additional	unnecessary	Additional
abnormal traffic detection	possible	Additional	unnecessary	Additional
lintrusion detection	impossible	necessary	unnecessary	necessary
lintrusion detection	impossible	necessary	necessary	necessary

TABLE I. Requirements for implementing SDN security

In this paper, we propose a method to implement firewalls, network scans, abnormal traffic detection, intrusion detection, and intrusion prevention techniques that are important in general security environment without additional hardware or cost based on the use of OpenFlow technology [6] in SDN environment. In addition, we propose a structure that provides an automated intrusion prevention function by implementing a program that connects Open Source IDS/IPS Suricata and OpenFlow according to the need to enhance security of SDN. This program works by integrating Suricata 's intrusion detection function and OpenFlow' s intrusion prevention function. When Suricata detects an intrusion, the program automatically implements the intrusion blocking function through OpenFlow. For more information on this, refer to "Table 1.Requirements for implementing SDN security" above.

C. Implement SDN security using OpenFlow

1) Firewall Implementation: In existing networks, firewalls are implemented as separate devices from network devices. SDN applications support the security policies required to implement firewalls. One of the features of OpenFlow is packet forwarding and dropping, which is very easy to implement in the SDN environment by passing it to network devices that support OpenFlow. This means that all network devices that support OpenFlow can function as firewalls, so that distributed firewall functions can be implemented without difficulty.

2) Network scan detection: Among the network scans, the Threshold Random Walk [7] detection method monitors the success or failure of the TCP connection and regards it as a scan attack when there is a statistically large number of failures than success. In order to find out the success or failure of each TCP session, it is necessary to be able to distinguish packets which start a TCP session. In SDN, it is possible to control the flow rules so as to monitor each network flow by each session. Most of the network scan attacks are based on the TCP protocol. However, there are cases where a network scan is attempted using a protocol such as ICMP or UDP. These protocols detect the scan by analyzing the amount of packets transmitted to the server because there is no session information.

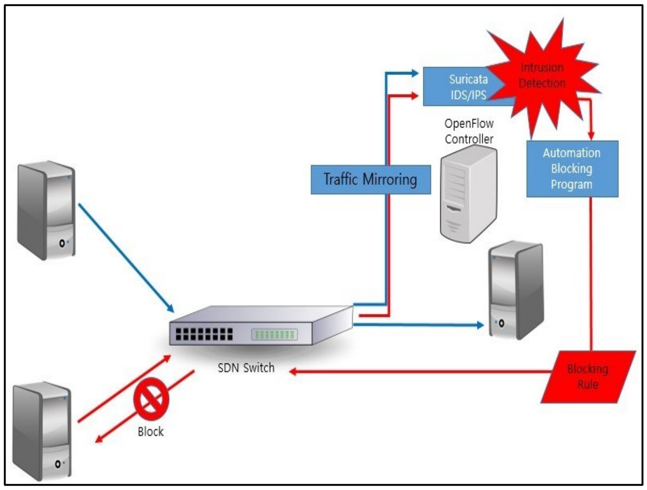
3) Abnormal traffic detection: The general network information used in the abnormal traffic detection system is as follows. First, the session information of the TCP protocol (success / failure information of each TCP session, information on the number of TCP session attempts and success / failure times for a specific network of a specific user). Second, the Nnumber of packets (PPS) or bytes (BPS) delivered per second for a particular user. Third, information such as the amount of change in BPS and PPS when traffic is transmitted. Here is how to get the necessary information mentioned above using OpenFlow. When First case, the TCP session is monitored as in scan detection. Second case, BPS and PPS are obtained by requesting the number of network packets and the number of bytes periodically transmitted to the network devices. Third case, it can be obtained by storing the above information in a separate data structure in the application program. In the future, based on this information, various algorithms can be applied to detect abnormal traffic and report it to the administrator, or to directly block related abnormal traffic or to interwork with the automatic blocking system.

D. Enhanced security by linking Suricata with OpenFlow

The intrusion detection system must analyze the contents of the packet to see if there is an attack pattern, so the contents of the packet in the data plane must be transferred to the SDN application. However, in the current OpenFlow, it is not easy to transfer the contents of the packet to the controller, so IDS implementation is also not easy. If a packet matches the flow rule of the device, the network device does not forward the packet to the controller but processes it according to the rule.

Therefore, the contents of the packet can not be transmitted to the controller. To solve this problem, a mirroring method should be used. The mirroring method refers to a method of copying all the packets going through the network equipment and transferring them to a specific server. This method is also widely used in existing network intrusion detection systems that do not use SDN technology. Locate the Open Source IDS / IPS Suricata that performs network intrusion detection on the specific server where the mirrored packet arrives and analyze the mirrored contents to see if there is a pattern related to the attack. If there is an attack pattern, judge it as an attack. In addition, a special program is created on a specific server to monitor the intrusion detection result of Suricata, and when the attack is detected, the function of sending the OpenFlow command to the network device is performed so that the SDN firewall function is operated. By doing this, the network intrusion prevention system is completed. This figure is shown in Figure 3 below.

Fig. 3. SDN security structure linked with Suricata



IV. Conclusion

In this paper, we investigated how to implement firewall, network scan and abnormal traffic detection in existing network environment using OpenFlow of SDN. In addition, in OpenFlow, it is difficult to implement the network intrusion prevention function because it is not easy to transfer the contents of the packet to the controller, but it is solved by Suricata interworking through mirroring. And, in this paper, In this paper, we propose an intrusion prevention system architecture that implements intrusion detection and automatic shutdown function by monitoring intrusion detection result of suricata and automatically blocking intrusion by using OpenFlow command.

V. Acknowledgment

This work was supported by Institute for Information & communications Technology Promotion(IITP) grant funded by the Korea government(MSIT) (2014-0-00547, Development of Core Technology for Autonomous Network Control and Management) and Next-Generation Information Computing Development Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science and ICT (No. NRF-2017M3C4A7083678).

References

[1] ONF, "Software-Defined Networking: The New Norm for Networks," ONF White Paper(https://www.opennetworking.org/images/stories/downloads/sdn-resources/white-papers/wp-sdn-newnor m.pdf

[2] J. Pettit, J. Gross, B. Pfaff, and M. Casado, "Virtual Switching in an Era of Advanced Edges," Proc. of the 2nd Workshop on Data Center-Converged and Virtual Ethernet

[3] ONF, "OpenFlow Switch Specification",(https://www.opennetworking.org/software-defined-standards/specifications/)

[4] Floodlight Project, http://www.projectfloodlight.org

[5] Open vSwitch, http://openswitch.org

[6] Shin Seungwon, Song Yongju, ″Designing Network Security Applications with Software-Defined Networking″, Telecommunications Review, vol.23, no.5, pp. 627-640 (14 pages), 2013.

[7] SCHECHTER, Stuart E.; JUNG, Jaeyeon; BERGER, Arthur W. Fast detection of scanning worm infections. In: Recent Advances in Intrusion Detection. Springer Berlin Heidelberg, 2004. p. 59-81.