International Journal of Engineering Research And Advanced Technology(IJERAT)
ISSN: 2454-6135 [Volume. 02 Issue.10, October– 2016]

SOFTWARE-DEFINED SECURITY

Matthew N. O. Sadiku, Adebowale E. Shadare, Siew Koay, and Sarhan M. Musa

Roy G. Perry College of Engineering

Prairie View A&M University

Prairie View, TX 77446

United States

ABSTRACT

The current security mechanisms are not suitable for software-defined networking (SDN) environment. They are unsuitable to deal with virtualized environments. Software-defined security (SDS) introduces simplicity to network security. SDN is quickly being followed by SDS as the foundation for software-defined data center. This article presents SDS as being new and important for SDN.

Key words: *Software-defined security, Software-defined protection.*

I. INTRODUCTION

Security has been an important task in computer communication networks. Current security mechanisms are facing challenges in dealing with network threats and attacks. Software-defined security (SDS or SDSec) has been proposed to meet these challenges. It is a security model in which the information security is controlled and managed by security software. The functions of network security devices, such as firewalling, intrusion detection, access controls, and network segmentation are extracted from hardware devices to a software layer. Thus software is used to control and manage resources. Protection is based on logical policies, not tied to any security device. SDS exploits the software-defined networking (SDN) to enhance network security. SDN makes the network control plane programmable through protocols such as OpenFlow.

Security has been recognized as one of the advantages of the SDN [1].

11. SDS ARCHITECTURE

The architecture of SDS is similar to that of SDN. It is designed to be modular, scalable, and secure. Typically, SDS architecture is as shown in Fig. 1. It separates security data and control planes, thereby automating detection and protection with standardized control messages [2]. It is organized into three layers [3,4]:

- *Physical Layer*: This layer is also known as the data layer or base layer since it is at the bottom of the three-tier hierarchy. It contains hardware forwarding devices such as switches, routers, virtual switches, and access points.
- *Control Layer*: This is the brain or the core of SDS since it handles all the control and management operations. All security mechanisms are abstracted from the security devices and set inside the controller in the control layer. Security solutions that are normally implemented in control layer include anti-virus, firewall, anti-spam, and intrusion prevention system (IPS).
- *Application Layer*: All applications reside in this layer. It contains SDN applications. Network security techniques can be deployed as applications in this layer. To be effective, security must be built into the architecture and must protect the availability, integrity, and privacy of information. SDS would be more secure because many breaches are due to misconfigurations of security products.

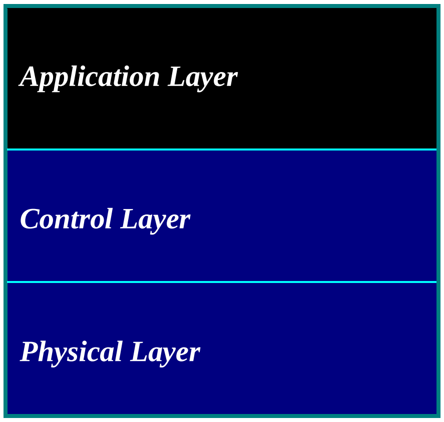

Figure.1: SDS Architecture.

III. FEATURES OF SDS

The following features and attributes of SDS distinguish it from traditional security schemes [3].

- *Abstraction*: SDS abstracts security policies from the hardware layer and runs it at software layer. Abstraction establishes common security models that can be deployed repeatedly.
- *Automation*: Deploying each asset or device in the system and putting it in a security trust zone are carried out automatically. This is done manually in traditional security approaches.
- *Flexibility*: Because SDS is entirely software-based, security policy is elastic and security is available "on demand." It is relatively easy to scale up and adapt to changes. It consistently enforces security policies across board irrespective of the location of the network systems.
- *Concurrency control*: Network security controls (such as intrusion and prevention, network segmentation, firewalling, violation monitoring, etc.) work together concurrently. Such an orchestration improves security and reduces the cost.
- *Visibility*: Since security can be virtualized, network professionals can discover abnormal activities that would not be possible with physical devices.
- *Portability*: SDS allows assets to carry their security settings with them as they scale or move from one location to another.

IV. ELEMENTS OF SDS

SDS is divided into three components, which are integrated to protect the network [3]. See Fig. 2.

- *Host*: The host is to send or receive data through the network. In traditional networks, security techniques reside in the host; the host checks new packets and see if they have threats. For the SDS, all security techniques are transferred to the controller.

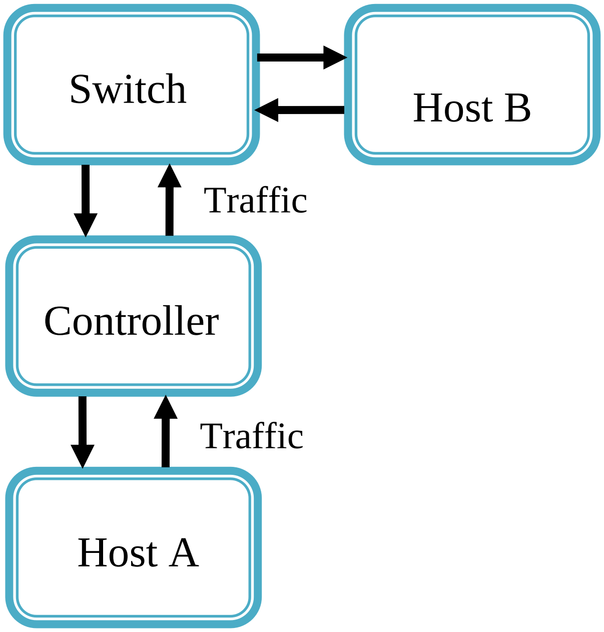

Figure .2: Elements of SDS.

- *Controller*: The controller is fully software-based. All security checks are done inside the controller. The controller can have better access control by stating which types of packets should be carried within the network. It has visibility of the traffic flows. It collects and processes information about the network.
- *Switch*: The switch consults the controller to decide whether to accept or reject a request. The current switches have limited storage capacity and cannot store all the rules. A reactive caching mechanism is adopted in SDN. However, this makes switches vulnerable to a DoS attack [5].

V. CONCLUSIONS

Software-defined security is now a hot topic in network security. It is a new approach to improve security within software-defined networking environment. Current security architectures are rigid and complex. In SDS, all security "devices" are controlled and obey some underlying rules which are translated by software. This allows for quick response and reduces human error.

Network security vendors include Check Point, Crossbeam, and Juniper. Some commercial products claim to provide software-defined solutions [6]. Examples of SDS are Software Defined Perimeter, Catbird, vShield, OneControl, and vArmour. It is hoped that these solutions will detect and mitigate individual attacks. These solutions must be scalable, simple, secure, and cost-effective.

REFERENCES

[1] H. Tu et al., "A scalable flow rule translation implementation for software defined security," *Proceedings of Asia-Pacific Network Operation and Management Symposium*, 2014.

[2] L. Wenmao et al., "SDN oriented software-defined security architecture," *Journal of Frontiers of Computer Science and Technology*, vol. 9, no. 1, 2015, pp. 63-70.

[3] A. Darabseh et al., "SDSecurity: a software defined security experimental framework," *Proceedings of IEEE Workshop on cloud computing systems, networks, and applications* (CCSNA), 2015, pp. 1871-1876.

[4] L. Yanbing et al., "SDSA: a framework of a software-defined security architecture," *China Communications*, Feb. 2016, pp. 178-188.

[5] M. Dabbagh et al., "Software-defined networking security: pros and cons," *IEEE Communications Magazine*, June 2015, pp. 73-79.

[6] S. Luo and M. B. Salem, "Orchestration of software-defined security services," *Proceedings of IEEE Workshops on Orchestration for Software-Defined Infrastructure*, 2016.

ABOUT THE AUTHORS

Matthew N.O. Sadiku, is a professor at Prairie View A&M University, Texas. He is the author of several books and papers. He is a fellow of IEEE.

Adebowale Shadare, is a doctoral student at Prairie View A&M University, Texas. He is the author of several papers.

Siew Koay, is currently a professor and the computer engineering program coordinator at the Department of Electrical and Computer Engineering, Prairie View A&M University.

Sarhan M. Musa , is a professor in the Department of Engineering Technology at Prairie View A&M University, Texas. He has been the director of Prairie View Networking Academy, Texas, since 2004. He is an LTD Spring and Boeing Welliver Fellow.