A Software-Defined Networking Security Controller Architecture

Fengjun Shang, Qiang Fu

College of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing 400065, China

E-mail: shangfj@cqupt.edu.cn

Keywords: Software-Defined Networking; security architecture; controller security

Abstract. With the development of the software-defined networking (SDN), centralized and open network management has brought many security problems. In this paper, we analyzed the security problems in SDN architecture, and then designed a SDN security controller architecture. We verified the feasibility and effectiveness of the architecture by using DDoS attack defense as an example, and analyzed the influence of the architecture on the network performance.

Introduction

Since the software-defined networking (SDN) was put forward, it received great attention. Related theories and technologies are also evolving and developing continuously. In the traditional network, the expansion of the network scale and large quantities of applications lead to the complexity of network structure. In order to solve the problems of TCP/IP network architecture, a lot of researches are carried out for the future network architectures. Redesign and redeployment of network core devices should be the first consideration of future network research [1]. The characteristic of SDN is that control function is independent. This conform to the developing direction of future network.

Although SDN technology has many advantages such as centralized control, fine-grained network control, and the reduction of management complexity, but as a new kind of network architecture, SDN technology is still in the development and test phase. It faces many new security issues. Security is the key to the development and popularization of SDN. Through the improvement of controller, development of security applications and innovation of security architecture, it can improve the security performance of SDN network and give full play to the characteristics of this SDN.

SDN Architecture

The core of SDN is separating control and data plane. Based on this, researchers at Stanford University proposed OpenFlow [2] technology as a way to the realization of SDN. Then ONF was established, it developed the OpenFlow protocol standard and the SDN white paper [3]. SDN network architecture is composed of infrastructure layer, control layer and application layer. Infrastructure layer consists of network devices which support SDN technology standard. Control layer shield the difference of the underlying devices by control data plane interface (Southbound Interface) to communicate with network devices. Control layer provides programmable network management environment, through the centralized control of the network devices, network resources can be configured flexibly and dynamically. Application layer can realize more web services through the Northbound Interface which is provided by control layer.

Researchers designed many implementation scheme of control layer based on SDN architecture and OpenFlow protocol. NOX [4] first introduced the concept of network operating system (NOS). NOS is the control software in SDN. In the OpenFlow network based on NOX, NOX is the control core. Because of NOX is the first SDN control layer based on OpenFlow, it has become a design template of OpenFlow controller [1]. Cisco, IBM and many companies developed OpenDaylight [5] controller. It supports both the “classic” OpenFlow-based approach and emerging model-driven network management and programmability technologies. The Open Networking Lab (ON.LAB)

developed an Open Network Operating System (ONOS) [6]. ONOS implements an open distributed control plane, it can provide scalable, high performance and high reliability NOS for large networks.

Security Analyses of SDN

Security Problems of Controller.

In the SDN based on OpenFlow, the direct manager of network is controller. So the running state of controller is related to the running of whole network. But the number of controller is limited, the centralizing of management device will be a weakness of network. When network is encountering with DDoS attack, large amount of traffic in network will be processed by controller. This may lead to the load of controller increases sharply, even lost processing capacity.

Security Problems of Flow Table.

Flow table is the basic guarantee of OpenFlow network. So the stability and reliability of flow table are also important to the security of SDN. In the process of network configuration, many flow tables may already exist in network devices. Each configuration may conflict with the existing policy. If controller do not coordinate the conflicts, configuration of network will be very confusing. And attackers can disable forwarding function or create a channel for malicious attacks by adding malicious flow entries through Northbound Interface.

Security Problems of Application.

As a feature of SDN, open Northbound Interface allows developers develop applications which could run on controller. But open interface can be used to attack network and cause other issues.

If the Northbound Interface can be used freely, the attack to network will be so easily. Attacker can develop malicious application to break controller. Even non-malicious applications can also cause problems of flow table.

Security Problems of Southbound Interface.

The reliability of Southbound Interface is also an important indicator of security. At present, Southbound Interface mainly refers to OpenFlow protocol. SSL/TLS protocol is adopted for the secure channel between controller and switch. But SSL/TLS protocol is not enough to establish and assure trust between controllers and switches. Attacker can gains access to the control plane by exploit the weaknesses of SSL/TLS [7].

A Security Controller Architecture

In this section, we design a security controller architecture. The architecture is shown in Fig. 1. This architecture is composed of basic control module and customizable multi-granularity security module. The basic control module implement basic functions, following the SDN architecture. And customizable multi-granularity security module provides customizable security features.

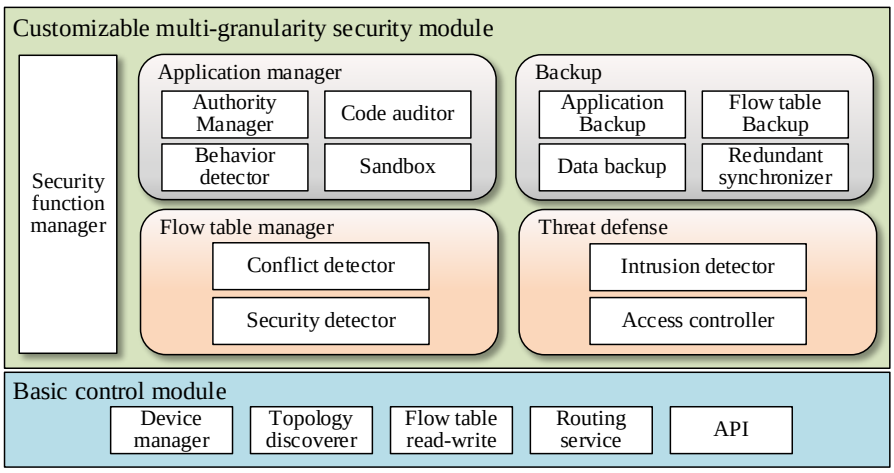


Fig. 1. Security Controller Architecture

The threat defense unit integrated intrusion detection and access control. Intrusion detection function provide defense based on controller and traffic detection based on SDN network. Network manager can use access control function to develop and implement the flow rules. Flow table manager is designed for the detection and resolution of conflict. It inspect the flow entry before it is sent to switch, if the flow entry is against existing flow entries or access control rules, flow table manager will rebuild flow table or reject this flow entry. Backup unit backup applications, flow tables, security rules and other important data in real time. These backups will make the recovery rapid after controller fails. The standby controller can synchronize with main controller through redundant synchronizer so that standby controller will immediately take over the network when main controller is down. The Northbound Interface is protected by application manager in this architecture. All of these functions defense the threats from Northbound Interface. The security function manager manage all units in customizable multi-granularity security module. User could make particular security strategy for their own network.

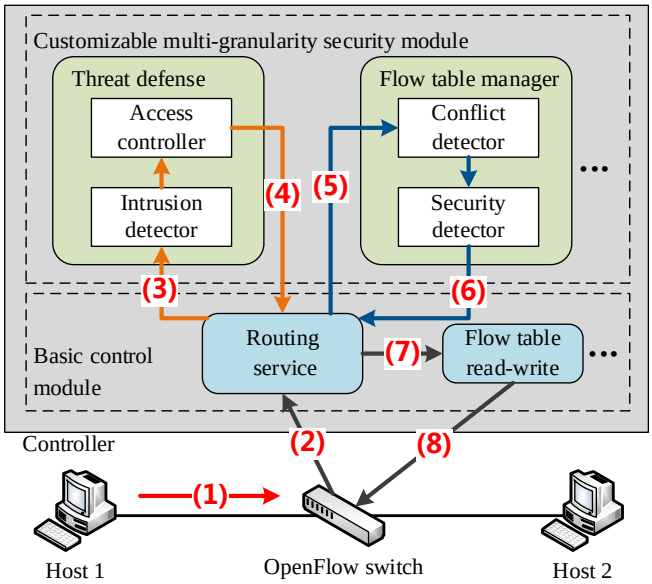


Fig. 2. Module operation mechanism

The operation mechanism of security controller is shown in Fig. 2. And the security control message is shown in Table 1. The complete operation process is as follows:

1) Switch receive a packet and it cannot match any flow entry.
2) Switch send PACKET_IN message to controller
3) Routing service send PACKET_INFO message to threat defense unit.
4) Threat defense unit return PACKET_SEC or PACKET_RFS message to routing service. If message type is PACKET_RFS, processing jump to step 7, otherwise continue.
5) Routing service generate forwarding flow entry for packet, then send FLOW_INFO message to flow table manager.
6) Flow table manager return FLOW_RSLT or FLOW_RFS message to routing service.
7) If routing service received PACKET_RFS or FLOW_RFS message, it generate dropping flow entry and send it to flow table read-write unit. If message type is FLOW_RSLT, send the flow entry in message to flow table read-write unit.
8) Flow table read-write unit send FLOW_MOD message to switch to add or modify flow table.

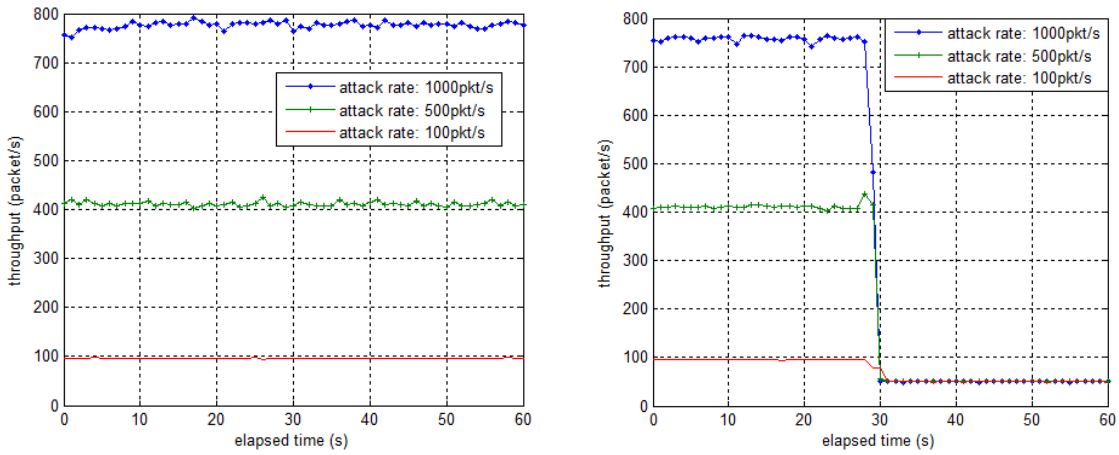
Table 1. Security control messages

Message type	Description
PACKET_INFO	Carry packet information
PACKET_SEC	Packet is allowed to forward
PACKET_RFS	Packets is refused
FLOW_INFO	Carry flow entries infomation
FLOW_RSLT	Send flow entry in message to switch
FLOW_RFS	Flow entry is refused

Evaluation

In order to verify the feasibility of our security controller architecture, we built an experimental environment. We tested performance and defense effect of controller. The experimental environment is based on Floodlight controller, Open vSwitch and Iptables. We monitor the network traffic by sFlow agent and sFlow-RT. Experimental environment was hosted on an Intel Xeon 1.80 GHz CPU with 16 GB RAM. Virtual test host is running on an Ubuntu-Server v14.04 with 2 vCPU 1.80 GHz and 2GB RAM.

In the experiment, we simulated SYN Flood attack from Host A to Host B three times by hping tool. The packets rates of attacks are 100 packets/s, 500 packets/s and 1000 packets/s. On the condition that the defense function of controller is disabled, downlink traffic statistics of Host B during 60 seconds is shown in Fig. 3 (a). When we enable defense function during attack, the traffic statistics is shown in Fig. 3 (b).

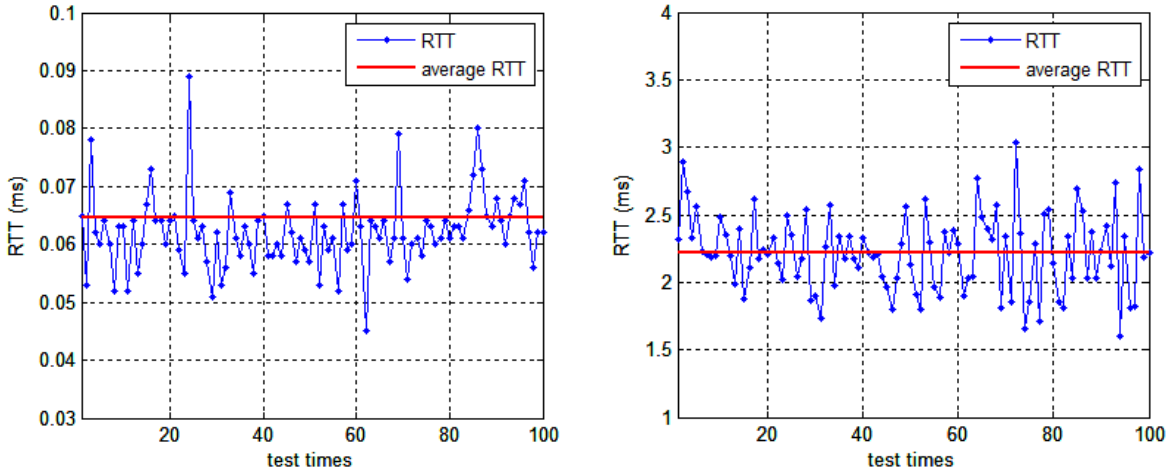


(a) Disabled defense (b) Enable defense during attack

Fig. 3. Downlink traffic statistics of Host B

According to the test results, when we disable defense, the average packets rates of Host B received are 95.48 packets/s, 411.23 packets/s, and 776.79 packets/s. However, if we enable defense during attack, the rate decrease to average 50.05 packets/s in 5 seconds and start to be steady. Attack test shows that security controller is effective for SDN security protection.

We tested the effect of security controller on network communication performance by measuring the round-trip time (RTT). We tested RTT between Host A and Host B 100 times in the case of disabling defense and enabling defense. Test results are shown in Fig. 4.



(a) Disabled defense (b) Enabled defense

Fig. 4. RTT statistics

According to RTT statistics, when defense is disabled, the average RTT between Host A and Host B is 0.065ms. After the defense function is enabled, it come to 2.219ms. RTT increase 2.154ms. This shows that adding security module to controller will degrade the performance of network to some extent. But considering the performance optimizing is not enough in experimental environment, the switches is also virtual, and the increase of delay is just millisecond. This kind of security controller architecture will not cause significant degradation to network performance.

Conclusions

We analyze the security problems in SDN architecture. And design a security controller architecture based on the security problems. The customizable multi-granularity security module of this architecture provide application threats defense, flow table management, traffic detection and other functions. Modules exchange information by security control messages. They protect network cooperatively. Our experiment shows this architecture is effective on network security protection based on traffic detection. And the network performance degradation caused by security functions is not much. But there are still many details need to be designed and perfected in this architecture. The effect and performance are also need be tested more (e.g., test it in real network environment). This will be the emphasis of my next work.

Acknowledgements

The author would like to thank the Chongqing Natural Science Foundation under Grant No.cstc2012jjA40038, the Chongqing Basic and Frontier Research Project under Grant No.cstc2013jcyjA40023, the Ministry of Industry and Information Technology for the special funds of Development of the Internet of things (2012-583) and the Special Foundation for Young Scientists of Chongqing (No.cstc2014kjrc-qnrc40002).

References

[1] ZUO Qing Yun, et al. "Research on OpenFlow-Based SDN Technologies." *Journal of Software* 24.5(2013):1078-1097.

[2] Mckeown, Nick, et al. "OpenFlow: Enabling Innovation in Campus Networks." *Acm Sigcomm Computer Communication Review*38.2(2008):69-74.

[3] “Software-Defined Networking: The New Norm for Networks” 2012. [Online]. Available: https://www.opennetworking.org/images/stories/downloads/sdn-resources/white-papers/wp-sdn-ne

wnorm.pdf

[4] Gude, Natasha, et al. "NOX: Towards an operating system for networks."*Acm Sigcomm Computer Communication Review* 38.3(2008):105-110.

[5] Medved, Jan, et al. "OpenDaylight: Towards a Model-Driven SDN Controller architecture." *A World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2014 IEEE 15th International Symposium on*IEEE, 2014:1-6.

[6] “Introducing ONOS - a SDN network operating system for Service Providers” 2014. [Online]. Available: http://onosproject.org/wp-content/uploads/2014/11/Whitepaper-ONOS-final.pdf

[7] Kreutz, Diego, F. M. V. Ramos, and P. Verissimo. "Towards secure and dependable software-defined networks." *Second Acm Sigcomm Workshop on Hot Topics in Software Defined Networking* 2013:55-60.