Novel privacy vulnerabilities and challenges of OpenFlow-based SDN in network security

Yujie Xie and Pankoo Kim*
Chosun University, Kwangju, Republic of Korea
Yujiexie1106@gmail.com, pkkim@chosun.ac.kr

Abstract

In the last decade, software defined networking (SDN) with novel mobile network environment has drawn so many attentions from both IT industries and academic because of its huge convenient and economic cost. Advanced mobile networks provide abundant entertainments and rich lifestyle for citizens, which requires larger storage and higher quality of internet environments such as innovative mobile cloud computing and especially potential SDN [1]. Then emerging SDN technology satisfying more users' demands gains a lot of momentum, which is a new network structure with higher security than traditional networks. Meanwhile, novel vulnerabilities and challenges play the most important roles in the deployment of software defined networking. In this paper, at first, we briefly introduce software defined networking including three main planes. Then OpenFlow-based SDN are illustrated. The procedure of OpenFlow-based SDN working is also provided in this part. And related security vulnerabilities and challenges on each layer are presented in the last part.

Keywords: software defined networking; OpenFlow ; security vulnerabilities and challenges

1 Introduction

With the development of new technologies such as big data and cloud computing, higher demands have been put forward on the traditional network of bandwidth, security, speed, etc. There are a lot of shortages on the traditional networks such as security, flexibility, and scalability, which can't meet users' higher requirements [9]. Many scholars and IT companies are actively involved in the deployment of the next-generation Internet architecture accordingly, and SDN is one of the most popular areas. The emerging SDN is becoming a hotpot rapidly after it is proposed with high flexibility and high programmability.

Software defined networking (SDN) is a new paradigm to virtualize network infrastructure by decoupling control planes from data forwarding planes, which provides APIs to the application layer to build an open programmable network loop [17]. In SDN, several control planes are logically centralized into one controller, meanwhile, forwarding devices like routers and switches in the data plane will receive commands from it. Then SDN can make precise changes to networks by managing control planes so that it will definitely secure the network environments better than simple network structure [6].

This new technology makes it easier for network administrators to create dynamic networks through open interfaces. The controller have visibility of the entire network and make forwarding decisions for each node, which means that control planes protection should draw more attention from IT engineers and enterprises while they deployed SDN structure [11]. A slight vulnerability on control planes (the node was attacked by DDoS) will lead to heavy data leakage and paralysis of the entire network. The benefits of SDN:

1. Reducing the operating costs of the network.
2. Speeding up the deployment of new services

Research Briefs on Information & Communication Technology Evolution (ReBICTE), Vol. 3, Article No. 8 (October 15, 2017)

*Corresponding author: 309 Pilmun-daero, Dong-gu, Gwangju, Republic of Korea. Tel: +82-62-230-7799

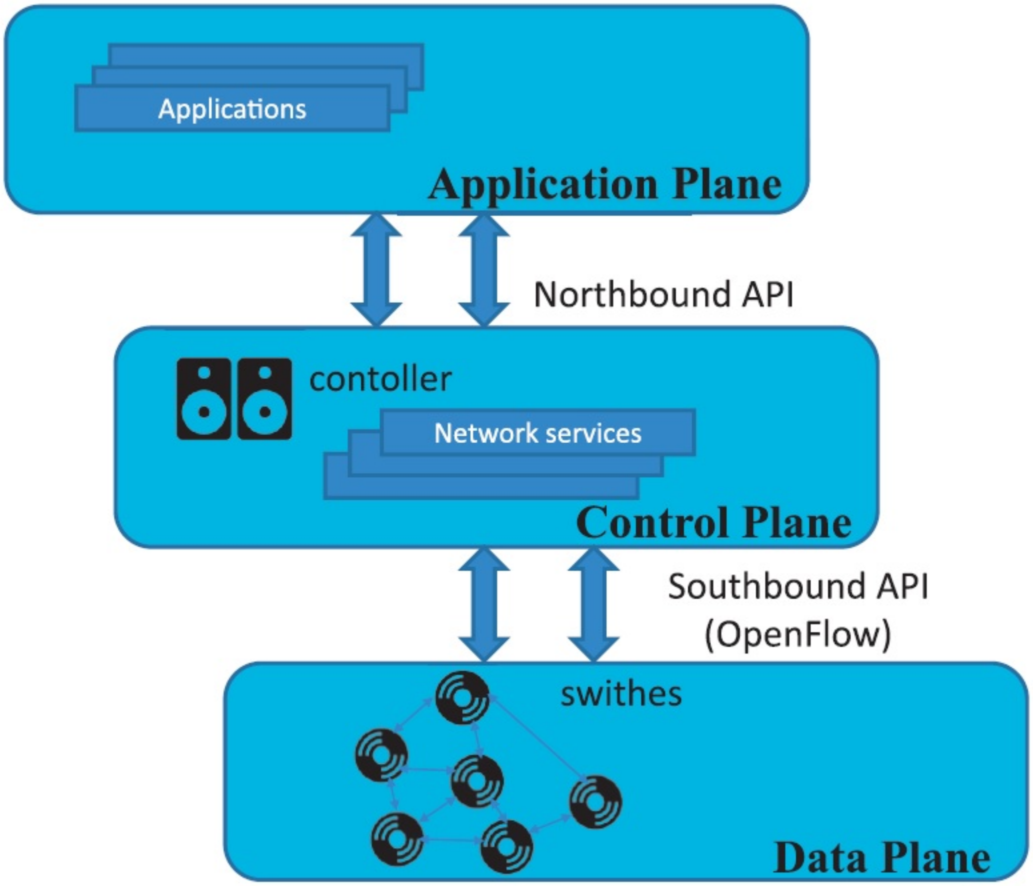

Figure 1: SDN logic structure.

3. Pushing the achievement of network virtualization.
4. Simplifying network deployment.

In the Figure 1, it shows that SDN architecture has three layers: the first layer is application layer including most of applications; the middle layer is made of a SDN controller with decision-making function, the bottom is the network infrastructure consisting of forwarding devices like routers and switches to support the data forwarding. The Northbound Application Programming Interface (API) is located at the top of the controller to provide network abstraction interfaces for applications. Southbound API (for example, OpemFlow) allows the controller to define the behavior of the switches at the point of the SDN stack [4].

2 OpenFlow-based SDN Structure

OpenFlow is the first standard interface of communications between control plane and data plane, which is an extensible network control protocol to decide where flow rules forwarded to switches are installed with the controller's command [16]. OpenFlow-based SDN architecture has realized the prototype design idea of SDN, and it is a typical example of SDN technology.

When the host A communicates with the host B for the first time in the OpenFlow-based SDN network, the basic communication procedure consists of seven steps in [8] [10] is described as following:

1. The host A connects with networks and sends packets to the switch 1.

2. The switch 1 queries flow table itself. The switch 1 will forward the packet to the controller via the PACKET-IN event if there is no flow table matching the packet in the flow table. When sending a message to the controller, the switch 1 can send the data packet directly to the controller through the TLS protocol, and can also encrypt the data packet by using transport layer security.

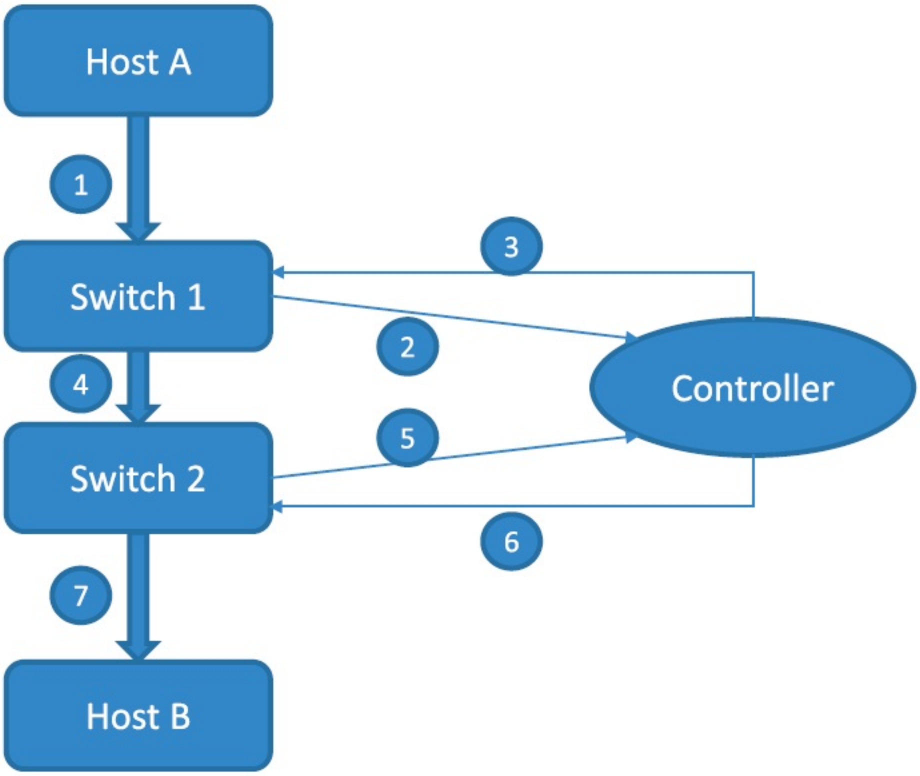

Figure 2: OpenFlow-based SDN processing structure.

3. After receiving the request information from the switch 1, the controller generates the corresponding response strategy and sends it to the designated port of the switch 1 through the Packet-Out event.

4. Switch 1 executes the response policy from the controller and forwards the packet to the switch 2.

5. If there is no data packet matching in the flow table of the switch 2, the switch 2 will forward the received packet to the controller through Packet-In event. If there is a match in the flow table of the switch 2, then it will jump to Step 7, which means the switch 2 forwards the data packet to the host B according to the corresponding flow table.

6. Similar to the Step 3, the controller sends the corresponding response policy to the designated port of switch 2 according to the request information of the switch 2.

7. Switch 2 executes the response policy from the controller, and the data packet is forwarded to the host B.

In the procedure above showed in Figure 2, we can see that a flow of OpenFlow-based SDN is a sequence of packets between a source and a destination so that forwarding decisions are based on flow not destination [12].

3 Security Vulnerabilities and Challenges in SDN

SDNs make network programming much easier. As the widely deployment of SDN, security vulnerabilities and challenges are coming out to most of researchers' consideration. In the table 1, there are obviously some common security issues on three layers but acting differently. Then those security threats will be separately introduced in this part.

3.1 Application Plane

There are so many various applications involved in application plane. Havoc of networks will occur if these applications are suffered from malicious attacks because that most of network functions are completed as applications [18]. Security threats on this layer mainly includes [7]: malicious code hidden in applications, application malicious code tampering, identity fraud, unauthorized access and application configuration defects.

Table 1: SDN Threats on each layer [12].

Categories	Threats
Application Plane	Fraudulent Rule Insertion, Unauthorized Access,Identity Fraud, configuration defects.
Northbound APIs	Standard Issues, Unauthorized Access, Data Leakage.
Control Plane	DDoS/DoS Attack,Single Point of Failure, Fraudulent Rule Insertion,Unauthorized Access,Configuration Issues, Network Sniffering.
Southbound APIs	SSL/TLS protocol security, OpenFlow Issues.
Data Plane	DDoS/DoS Attack, Fraudulent Rule Insertion, Unauthorized Access, Configuration Issues, Data Leakage, Identity Fraud, Scanning, Spoofing.

In SDN, expecting most flow rules designed by administrator, OpenFlow applications or other third party applications can also formulate flow rules which will be forwarded to related forwarding devices such as routers and switches. Authentication and Authorization should be a major threat on this layer and potential security issues to other two layers. Sometimes applications completing main functionalities on the controller can get access to network resources without a good security authentication system to defend external attacks [9]. On the other hand, there are no compelling SDN security protection mechanisms designed for applications to build a stronger trust relationship between the controller and applications, which is the same between control plane and data plane. In the present mechanism, switches and basic infrastructures almost fully trust and carry out flow rules forwarded by the controller [5]. Therefore it will bring SDN unpredictable security threats if applications involved in the development of flow rules are suffered from tampering and attacking. A complete Authentication and Authorization system is really in needed to provide a more secure environment for applications which is also a protection for other planes.

The act of attacking the north interface protocol can also be a security consideration. These northbound interfaces attacked are managed by the controller, which can be encapsulated by Python, Java, XML, and so on [1]. Attackers can control the communication of the SDN network through the controller and will achieve their own commercial purpose if they take fully advantage of these openness and no authentication northbound interfaces [11].

3.2 Control Plane

SDN takes centralized control of networks by the controller which is definitely the biggest weakness as the kernel of SDN. The controller in control plane is known as a central point in systems because of its overall network management that all functionalities take place on it. Furthermore, it's also an entity to make decisions for switches on next plane [14]. Then the logically centralized controller is obviously a main attack target and the attacker will hold the control of whole networks only by getting access to the controller which brings destructive security problems for SDN. The typical security problem on control layer is the controller single point failure problem because of the centralized management.

DoS/DDoS attacks are considered as an important security threats that the attacker makes a series of illegal access to produce excess load for controllers by creating a false expression that legitimate users can't be able to access system resources of control layer [1]. Because that if there are no matching response flow rules on switches after the requirement of new flows reaching at switches, data packets will be forwarded to the controller to make corresponding response flow rules [2]. Therefore, some of attackers will send a large amount of false information requests to the controller by utilizing switches in SDN, resulting in the heavy load of controller to interrupt switch request service. Otherwise, applications of the first layer which usually have several functional demand from the control plane will be a potential security issue to control plane. Isolation, auditing and tracking of authorizing resources those applications utilize are hard works to build a strong security environment. However, it's necessary to authorize and authenticate applications before they get access into resources. Effective monitoring should be put into the SDN security mechanism to solve this problem. Physical destruction is also a vulnerability in this layer, which mainly refers that the key controller in SDN is destroyed logically so that user's normal request can't get related respond [13]. Then end users will still wait for the replying that they can't be informed the controller damage, which is time consuming.

In addition, the main security threats on the control layer include illegal access, identity fraud, malicious flow rules and configuration defects etc. Attackers may target SDN controllers for different purposes [14]. For instance, an attacker may send Southbound / Northbound interface false dialogue message to controller. If the controller make a replying to them, then the attacker has the ability to pass security detecting deployed by the controller [10].

3.3 Data Plane

The infrastructure layer consists of switches and routers, which is mainly responsible for data processing and forwarding. Even though OpenFlow forwarding devices enable SDN better to collect a variety of information from the network that makes DDoS flooding attacks detection much easier [15]. There are still a lot of novel DDoS and other vital security problems on the data plane of OpenFlow-based SDN. As show in Table 1, the major security threats on this layer include: Fraudulent rule insertion, DDoS/DoS attacks, data leakage, illegal access, identity fraud and configuration defects.

An attacker might target a node from the network such as the OF switches. For example, an attacker can firstly get access into unauthenticated network, and then attack an unstable network node, which could be a Denial of Service attack or a Fuzzing attack on a network infrastructure like a switch [14].

The malicious flow rules: there are many southbound interface protocols (especially OpenFlow) for the communications between controllers and switches of data layer [3]. Each protocol actually has its own security communication mechanism but cannot be able to be protected effectively. Furthermore, attackers can add new flow tables to the OF switches by using performance of these protocols, which is really dangerous for data plane. Authentication and authorization system also takes an important position to secure the packet values during transmission.

4 Conclusion

Novel security vulnerabilities and challenges play the most important role in the development of SDN, which requires more researchers involved in this area. SDN provides a new approach of centralizing network control plane and enabling network programmability to enhance its security and flexibility, at the same time, which exposes new security problems in networks. In this paper, we briefly introduced SDN security structure and basic procedure of OpenFlow-based SDN, and also, vulnerabilities and challenges of SDN on each layer are proposed in the session 3. The future networking will depend much more on software-related [10].Solutions targeting these threats and challenges must be completed in the future to solve potential SDN security issues.

Acknowledgments

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIP) (No. NRF-2016R1A2B4012638) and the MIST(MInistry of Science & ICT), Korea, under the National Program for Excellence in SW(2017-0241) supervised by the IITP(Institute for Information & communicatins Technology Promotion) (2017-0241)

References

[1] I. Ahmad, S. Namal, M. Ylianttila, and A. Gurtov. Security in software defined networks: A survey. *IEEE Communications Surveys & Tutorials*, 17(4):2317 – 2346, April 2015.

[2] C. Choi, J. Choi, and P. Kim. Abnormal behavior pattern mining for apt attack detection. *Computer Systems Science & Engineering*, 32(2):8351 – 8361, March 2017.

[3] C. Esposito and C. Choi. Signaling game based strategy for secure positioning in wireless sensor networks. *Pervasive and Mobile Computing*, 40(2):611–627, September 2017.

[4] B. Ferguson. Understanding software defined network security! In Linda, editor, *Linked in*, pages 1–16. MCT Alumni, 2017.

[5] S. Hogg. Sdn security attack vectors and sdn hardening, 2014. `https://www.computerworld.com.sg/blogs/guest-blogs/sdn-security-attack-vectors-and-sdn-hardening/?page=1` [Online; Accessed on October 3, 2017].

[6] X. Huang, X. Du, and B. Song. An effective ddos defense scheme for sdn. In *Proc. of the 2017 IEEE International Conference on Communications (ICC'17), Paris, France*, pages 1–6. IEEE, May 2017.

[7] R. Kaur, A. Singh, S. Singh, and S. Sharma. Security of software defined networks: Taxonomic modeling, key components and open research area. In *Proc. of the 2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT'16), Chennai, India*, pages 2832 – 2839. IEEE, March 2016.

[8] D. Kreutz. Software-defined networking: A comprehensive survey. *Proceedings of the IEEE*, 103(14820214):14–76, December 2014.

[9] M. Liyanage, A. B. Abro, M. Ylianttila, and A. Gurtov. Opportunities and challenges of software-defined mobile networks in network security. *IEEE Security & Privacy*, 14(4):34–44, June 2016.

[10] ONF. Open Networking Foundation, January 2009. https://www.opennetworking.org/ [Online; Accessed on October 3, 2017].

[11] ONF. ONF White Paper, April 2012. http://www.bigswitch.com/sites/default/files/sdn_resources/onf-whitepaper.pdf [Online; Accessed on October 3, 2017].

[12] O. Salman, I. Elhajj, A. Chehab, and A. Kayssi. Software defined iot security framework. In *Proc. of the 4th International Conference on Software Defined Systems (SDS'17), Valencia, Spain*, pages 75 – 80. IEEE, May 2017.

[13] X. Su, A. Castiglione, C. Esposito, and C. Choi. Power domain noma to support group communication in public safety networks. *Future Generation Computer Systems*, 70(1016):1–15, July 2017.

[14] X. Su, C. Liang, D. Choi, and C. Choi. Power allocation scheme for femto-to-macro downlink interference reduction for smart devices in ambient intelligence. *Mobile Information Systems*, 4(7172515):8351 – 8361, November 2016.

[15] X. Su, Y. Wang, D. Choi, P. Kim, and C. Choi. Channel allocation and power control schemes for cross-tier 3gpp lte networks to support multimedia applications. *Multimedia Tools and Applications*, 36:1–17, January 2017.

[16] Y. Wang, X. Dai, J. J. Jung, and C. Choi. Performance analysis of smart cultural heritage protection oriented wireless networks. *Future Generation Computer Systems*, 69(33):1–13, April 2017.

[17] Y. Wang, X. Su, D. Choi, and C. Choi*. Coordinated scheduling algorithm for system utility maximization with heterogeneous qos requirements in wireless relay networks. *IEEE ACCESS*, 4:8351 – 8361, November 2016.

[18] I. You, J. D. Lim, J. N. Kim, H. Ahn, and C. Choi. Adaptive authentication scheme for mobile devices in proxy mipv6 networks. *IET Communications*, 10(17):2319 – 2327, November 2016.

Author Biography

Yujie Xie received her B.S degree in Software Engineering from Qingdao University of Technology in 2016. Currently, she is now studying as a student for the M.S degree in Computer Engineering from Chosun University, Gwangju, South Korea. Her major research interests focus on Mobile Networks, Security on Cloud Computing and Big Data, Cryptography.

Pankoo Kim received the B.S. degree from Chosun University, Gwangju, South Korea, in 1988 and the M.S. and Ph.D. degrees in computer engineering from Seoul National University, Seoul, South Korea, in 1990 and 1994, respectively. He is currently a Full Professor with Chosun University. His specific interests include semantic web techniques, semantic information processing and retrieval, multimedia processing, semantic web, and system security. He is the Editor-in-Chief of the IT CoNvergence PRActice Journal.