DOI: http://dx.doi.org/10.26483/ijarcs.v9i1.5498

ISSN No. 0976-5697

Volume 9, No. 1, January-February 2018

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

OPENFLOW CONTROLLER-BASED SDN:SECURITY ISSUES AND COUNTERMEASURES

Hamza Mutaher
Dept. of computer science & information technology
Maulana Azad National Urdu University
Hyderabad, India

Pradeep Kumar
Dept. of computer science & information technology
Maulana Azad National Urdu University
Hyderabad, India

Abdul Wahid
Dept. of computer science & information technology
Maulana Azad National Urdu University
Hyderabad, India

Abstract: Due to the complexity of managing and monitoring large-scale traditional Networks. Software-Defined Network (SDN) is the recent network paradigm that has come up to overcome the drawbacks of the traditional network. SDN decouples network control plane from data plane enabling network centralization control and network programmability. Thus simplifying network scalability. However, the vigor of SDN caused of several security challenges and issues associated with various attacks. The current paper aims at introducing a descriptive review of OpenFlow controller-based SDN and the recent existing countermeasures. Moreover, various methods of protecting the controller from such attacks have been discussed which deemed as the valuable contribution in the research field of SDN security.

Keywords: Software defined network, Control plane, Data plane, OpenFlow, Security issues, Countermeasures.

I. INTRODUCTION

As the digital society is growing and everything is almost connected to each other and accessible from everywhere. The network became hard and difficult to be managed and controlled. Software-Defined Network (SDN) promises to simplify management, control of the network, and make it scalable by promoting a centralization control and defining the ability to program the network. Software-defined network is archetypically built up from a huge number of network devices like router, switch and various types of middleboxes like a firewall operated under centralized controller with a lot of complex protocols used to implement them [1]. The logical idea behind software-defined network is to segregate control plane of the network which makes decisions about how the forwarded packet should flow in the network from the data plane which is used to forward the packet and allow to program network using external tools [3].

OpenFlow is a protocol provides a communication among control plane and data plane in SDN. OpenFlow implemented on the controller, switch and the channel connected between them. Security of OpenFlow based SDN is a major aspect that poses a threat to SDN. The protection of controller, switch, and the channel is the main assets in the security of OpenFlow based SDN as they considered the main threatened objects need to be protected from several attacks like DoS, spoofing, hijacking, Man in the middle and so on. Controller security is the key point of our research paper that has several security challenges. Due to the complexity control, OpenFlow controller considered as one of the most important devices in SDN architecture where it has to collect network status to update packet forwarding rules to OpenFlow switches and control the performance of the whole network. OpenFlow controller considered as a bottleneck issue in SDN security [4]. In section II, we overview the various security issues of OpenFlow controller-based SDN and the challenges that threaten SDN network. In section III, we discuss the existing countermeasures of those issues and challenges providing the various methods to overcome SDN security attacks. In section IV, we provide a discussion about the countermeasures methods and the future work to improve the security of OpenFlow controller-based SDN.

II. SECURITY ISSUES FOR OPENFLOW CONTROLLER BASED SDN:

Security issues in OpenFlow controller-based SDN mostly refer to the vulnerabilities at the control plane in which attacker can compromise SDN. Since the controller is responsible to contributing the incoming networking flows, the controller becomes the main spot for numerous attacks. We overviewed open controller security issues based some attacks as follows:

A. Flooding and Denial-of-Services Attacks:

The controller becomes the main spot for flooding and denial of service (DoS) attacks. For example, control plane in SDN has to get the request from forwarding plane to provide flow rules in the time of receiving unknown network packets (as it cannot manage) thus; the attack can make control plane unavailable to respond the request from data plane [4]. An attacker may apply a DoS attack or some other means to make controller down. For example, attackers may

perform some methods of resource consumption on the controller to slow it down. So controller response will be slow to incoming packets hence makes it down. Shin and GU [7], elucidate an effective and credible DoS attack to SDN network that contains two steps:

- Investigate if a given network is using SDN OpenFlow switches.
- Conduct resource consumption attack, since the attacker has already known the condition of the flow rules of intended network.

B. Host Hijacking Attack:

Host Hijacking Attack is a spoofing attack by exploiting the Host Tracking Service in the OpenFlow network. The Host Tracking Service (HTS) is a network-wide view and an essential service in SDN controller. The issue with HTS is that it gets poisoned through host impersonation attack, man-in-the-middle attack or DoS attack [14]. The controller is aware of whole network information management hence the attacker can use scanning attack to get the whole information about the network and modify the sensitive information and configurations that may slow down SDN performance. If an attacker successfully hijacked the controller, then sensitive information such as password and communication data can be manipulated or modified and can redirect the traffic to any destination as well [4]. In this case, the compromise of OpenFlow will be easy to the attacker. For example, Hong et al. [5] introduced a Host Location Hijacking Attack which can spoof the identification information of the targeted host for the sake of hijacking its location in OpenFlow controller.

C. Tampering attack:

In tampering attack, northbound and southbound API messages might be spoofed to insert malicious flow rules towards network devices. If attacker succeeded to tamper flows from the certain controller, then the traffic will be allowed to flow across SDN network, and the possibility to add new security policies will increase which may cause network misbehavior [4].

Dynamic Flow Tampering is an obvious example of tampering attack. The attacker may try to set up various rules which no flow infringes any constant rules of the firewall, but in fact, they can infringe those rules in co-operative aspect and that called Dynamic flow Tunneling. Porras et al. [6] introduced an attack and pointed out that OpenFlow controller can generate a new rule to optimize the routing flow from remote client to the network resources. As the condition of an OpenFlow switch should reconfigure to match new flows, it is not easy for OpenFlow switch to predict a new rule since a new rule may be inserted by different OpenFlow controller applications dynamically and this is the main challenge. Thus, it is difficult to ensure that this new rule is not in clash with the previous rules. An example of Dynamic Flow Tunneling is given by [6]. Suppose there are three hosts, one is OpenFlow controller and the other two are nodes of OpenFlow switch:

- OpenFlow Controller: 10.1.1.22
- Node A: 10.1.1.23
- Node B: 10.1.1.24

And taking into consideration the existing firewall rule that blocks incoming network packets to web service operated on node B from OpenFlow controller. Also, take on the position that some other OpenFlow applications add new rules to the OpenFlow controller as below:

- The source IP address of the packet should be modified to 10.1.1.21 if the packet delivered from 10.1.1.22 to 10.1.1.23 (port 80).
- The destination IP address of the packet should be modified to 10.1.1.24 if the packet delivered from 10.1.1.21 to 10.1.1.23 (port 80).
- The packet should be allowed to be forward from 10.1.1.21 to 10.1.1.24 at port 80.

In this case, the packet can pass through the firewall if it gets sent from 10.1.1.22 to 10.1.1.23. The reason is that the packet has indirectly sent to 10.1.1.23, not 10.1.1.24 .however, this packet eventually can be forwarded to 10.1.1.24 even if there is a firewall depriving such traffic.

D. Spoofing attack:

The attacker can impersonate the controller using the spoofing attack. If this attack successfully conducted, then the attacker can create and update the entries of flow table in SDN network components. Network specialists may not get a visible view of those flows from the production controller. Thus the attacker would have to control the network entirely [4].

There are some other attacks which strive to compromise the performance of the controller like replay attacks, Host impersonate attacks and some other. Those attacks may use different vulnerabilities in control plane to manipulate network efficiency.

III. EXISTING COUNTERMEASURES:

To secure the OpenFlow controller-based SDN from various threats the reliable techniques must be conducted to secure these impedimenta. Several approaches and researchers have been conducted; we summarized them as follows:

Tootoochina et al. [8] Designed a Hyperflow. A distributer event-based control plane for OpenFlow. It enables the network to be scalable and keeps the benefits of network control centralization. Hyperflow application operated on OpenFlow controller and put network-wide views in synchronization through generating actions that affect controller status. Hyperflow restricts decision making to be localized in distinct controllers. Thus, the response time of data plane request minimized by the control plane. It also enables the communication between self- managed OpenFlow networks, and this is a fundamental character which is not available in existing OpenFlow network deployments. This ensures availability of the controller to antagonize DoS attacks.

DoS attack is also known as the main attack that overhead easily the controller processing and enables the flooding of MAC tables. So that criticized the performance of SDN. Dridi et al. [13] Designed SDN-Guard. An innovative method redirects the flow of malicious traffic and sets long timeouts aggregation of flow rules related to malicious traffic to secure SDN controller. SDN-Guard succeeds to minimize the effects of DoS markedly reduced up to 32% of controller throughput and control plane bandwidth.

Nguyen et al. [14] proffered an extension to SDN controller. It is dynamically secure controller from threats on host

tracking service which we discussed in section II. The extension contains three factors as follows:

- Port Manager: It is responsible for identifying the host which is generating the traffic. It also contains the list of host mapped with MAC address.
- Host Probing: It is responsible for verifying whether the host is reachable or not by providing ICMP echo request.
- Host Checker: To check if the host can be migrated and avoid ARP poisoning. That may cause some other attacks which have been discussed in section II.

Kuerban et al. [15] presented FlowSec strategy to prevent DoS attack on SDN controller. FlowSec calculates the gathered controller bandwidth statistics dynamically. If the attack found, the switch will be forced to slow down by using Floodlight module [16] which is also responsible for gathering switch statistics. Then, Suh et al. [9] designed a Content-oriented Networking Architecture (CONA) and explained how it works on NetFPGA-OpenFlow platform. In CONA an access router is capable of identifying which content gets request form attached, hosts. CONA agent receives the hosts' content request and sends back the response content. In such a way, CONA obtains the accountability and countermeasure can be taken to prevent threats which exhaust network resources like DDoS attack. Since controller is a single point failure, TCP SYN FLOOD attacks (Type of DDoS attack) can attempt to attack the controller. Fichera et al. [11] presented OPERETTA. It is an OpenFlow based solution to TCP SYN FLOOD attacks. OPERETTA is applied to allow TCP SYN packets to enter into the controller and deny bogus connection request. Later, a study has been done by Wang et al. [10] to data-to- control plane saturation attack in the active router and introduced FloodGuard. FloodGuard works under two approaches. The first approach is a proactive flow rule analyzer. It issued to conserve network policy enforcement by dynamically generating proactive flow rules and loading those rules into data plane switch. The second approach is a Packet migration. It is used to hand over the flooding packets to OpenFlow controller by using Round-Robin scheduling algorithm after caching them to maintain the controller performance.

Buragohain et al. [17] designed SDN framework called FlowTrApp. It detects and mitigates low and high rate DDoS attack on the data center. It classifieds incoming attack traffic by matching it with consistent flow traffic rules. The mitigation occurs if the user sends attack traffic very often but not from the first attempt.

Since the controller is the one who is responsible for performing the validation of source address, Yao et al. [12] proffered a solution to source address validation called Virtual Source Address Validation Edge (VAVE) that capable to identify the source address of the incoming packet. VAVE graced with many important characteristics. One of these significant characteristics is agility. VAVE agility decreases packet process overload and improves resource usage performance. An Address Resolution Mapping has been applied to the controller by Matias et al. [18] which can avoid unidentified ARP request by tracing MAC address.

Hong et al. [5] presented TopoGuard a security tool works on SDN controller that automatically affords a real-time detection discovery to poisoning attack on network topology. TopoGuard is capable of protecting network topology by introducing a minimum effect on a normal operator of SDN controller.

Canini et al. [19] designed a NICE which automates an OpenFlow application test by applying a checking model in network devices in a systematic way to speed up the discovery of state space of unaltered controller programs. Porras et al. [6] proposed FortNOX, a tool which grants an authorized rule-based and enforces security policy to NOX controller [22].

Kreutz el at. [20] Designed a secure and dependable SDN controller platform called FortNOX that operates on NOX controller. It can verify every rule and applies the strong technique to analyze these rules even if they try to insert into the flow rule which may cause a conflict in NOX controller. Then, Wen et al. [21] proposed PermOF, a fine-grained permission system to apply minimum privileges in applications and enforce permissions at controller API entries. They also proposed a technique provides an access list and isolation between controller operating system and applications which put the priority on operating system rules. Then, Hu et al. [23] proposed FlowGuard framework to perform a proper verification and efficient decision of firewall rule contravention in dynamic OpenFlow based networks.

Overall, to conquer these sorts of security issues. It is advised to implement such security policy enforcement, monitoring tools and trusted techniques to SDN controller. On the other hand, it is useful to perform recovery mechanisms to ensure the stability of the network.

IV. DISCUSSION

In this section, we summarize OpenFlow controller-based SDN that discussed in the previous section. Table I reflects brightly for the industrial companies interested in SDN. The various OpenFlow controller-based SDN countermeasures suggested by the authors mentioned respectively. This can facilitate the selection of a proper method for specific attack. These countermeasures are more useful for the researchers to have such knowledge about the recent methods proposed including frameworks and solutions to attempt to develop new countermeasures for the challenges of OpenFlow controller-based SDN. OpenFlow and SDN promise to dramatically facilitate network observation and enable invention by introducing programmable network. So that helps researchers to mitigate some other security issues and ease them to establish their evolution.

V. CONCLUSION

With the segregation of control and data planes. SDN facilitate the network management and promote the network programmability for external application to apply some new rules and policies. Network centralization and programmability bring security crisis. OpenFlow is the ultimate deployed SDN approach. Due to the flexibility provided by OpenFlow, the controller becomes a target for various security attacks. In this paper, we demonstrate the OpenFlow controller-based SDN security issues that extracted from our reading and literature review. Moreover, the existing research method presented and discussed to make it easy in the selection of the proper method associated with the specific attack for the specific application. Future work of this paper can be introduced by developing and

evaluating methods for the security of OpenFlow Controller Based SDN.

Table I. Summary for Openflow controller based SDN countermeasures

Author	*Method*	*year*	*Attack*	*Description*
Tootoochina et al. [8]	*Hyperflow*	*2010*	*DoS attack*	*Intended to minimize the control plane response time to oppose DoS attack.*
Suh et al. [9]	*CONA*	*2010*	*DoS attack*	*Introduced to achieve the accountability and take countermeasures against resource-exhaustive attacks.*
Yao et al. [12]	*VAVE*	*2011*	*Spoofing attack*	*Intended to prevent spoofing attack by verifying the source addresses of external packets.*
Matias et a. [18]	*ARM*	*2012*	*ARP spoofing attack*	*Implemented to track MAC addresses to prevent ARP spoofing attack.*
Wen et al. [21]	*PermOF*	*2013*	*Potential Attacks*	*Proposed to apply minimum privileges of OpenFlow applications to eliminate such attacks.*
Fichera et al. [11]	*OPERETTA*	*2015*	*DDoS attack*	*Implemented in SDN controller to reject bogus connection requests.*
Wang et al. [10]	*FloodGuard*	*2015*	*DoS attack*	*Intended to deny requests for data to control planes saturation attacks.*
Hong et al. [5]	*TopoGuard*	*2015*	*Poisoning attack*	*Intended to grant automatic and real-time disclosure of poisoning attack.*
Kuerban et al. [15]	*FlowSec*	*2016*	*DoS attack*	*Proposed to mitigate the DoS attack to control plane bandwidth by restricting the number of incoming packets to the controller.*
Buragohain et al. [17]	*FlowTrApp*	*2016*	*DDoS attack*	*A framework proposed to perform DDoS attack detection and mitigation for the data center.*
Dridi et al. [13]	*SDN-Guard*	*2016*	*DoS*	*A new scheme proposed to secure SDN controller to prevent DoS attack.*
Nguyen et al. [14]	*SDN Extension*	*2016*	*Host impersonate attack Man-in-the middle attack DoS attack*	*SDN extension designed to protect the controller dynamically from attacks on host tracking service.*

VI. REFERENCES

[1] Bruno Astuto A. Nunes, Marc Mendonca, Xuan-Nam Nguyen, Katia Obraczka, and Thierry Turletti, A Survey of Software-Defined Networking: Past, Present, and Future of Programmable Networks, IEEE Communications Surveys & Tutorials, Volume: 16, Issus: 3, Third Quarter 2014.

[2] Diego Kreutz, Fernando M. V. Ramos, Paulo Verissimo, Christian Esteve Rothenberg, Siamak Azodolmolky, and Steve Uhlig, Software-Defined Networking: A Comprehensive Survey, Proceedings of the IEEE Volume: 103, Issue: 1, Jan. 2015.

[3] Sakir Sezer, Barbara Fraser, David Lake, Jim Finnegan, Niel Viljoen, Marc Miller and Navneet Rao, Are We Ready for SDN? Implementation Challenges for Software-Defined Networks, IEEE Communications Magazine, Volume: 51, Issue: 7, July 2013.

[4] Wenjuan Li, Weizhi Meng, Lam For Kwok, A Survey on OpenFlow-based Software Defined Networks: Security Challenges and Countermeasures,Journal of Network and Computer Applications, Volume 68, Pages 126–139,June 2016.

[5] S. Hong, L. Xu, H.Wang, G. Gu, Poisoning Network Visibility in Sof tware-Defined Networks: New Attacks and Countermeasures. In: Proceedings of NDSS, San Diego, USA, 2015.

[6] P. Porras, S. Shin, V. Yegneswaran, M. Fong, M. Tyson, G. Gu, A Security Enforcement Kernel for OpenFlow Networks. In: Proceedings of the 1st Workshop on Hot topics in Software Defined Networks (HotSDN), 2012, pp. 121-126.

[7] S. Shin, G. GU, Attacking Software-Defined Networks: A First Feasibility Study. In: Proceedings of the 2ndWorkshop on Hot topics in Software Defined Networks (HotSDN), Hong Kong, China, 2013, pp. 165-166.

[8] A. Tootoonchian, Y. Ganjali, HyperFlow: A Distributed Control Plane for OpenFlow. In: Proceedings of the Internet Network Management Workshop/Workshop on Research on Enterprise Networking (INM/WREN). San Jose, CA: Usenix, 2010, pp. 1-6.

[9] J. Suh, H. Choi, W. Yoon, T. You, T. Kwon, Y. Choi, Implementation of Content-oriented Networking Architecture (CONA): A Focus on DDoS

Countermeasure. In: Proceedings of the 1st European NetFPGA Developers Workshop, Cambridge, UK, ACM Press, 2010, pp. 1-5.

[10] H. Wang, L. Xu, G. Gu, FloodGuard: A DoS Attack Prevention Extension in Software-Defined Networks. In: Proceedings of the International Conference on Dependable Systems and Networks (DSN) pp. 239-250, 2015.

[11] S. Fichera, L. Galluccio, S.C. Grancagnolo, G. Morabito, S. Palazzo, OPERETTA: An OPEnflow-based REmedy to mitigate TCP SYNFLOOD Attacks against web servers. Computer Networks 92, pp. 89-100, 2015.

[12] T. Feng, J. Bi, H. Hu, G. Yao, P. Xiao, InSAVO: Intra-AS IP source address validation solution with OpenRouter. In: Proceedings of INFOCOM, 2012.

[13] Lobna Dridi, Mohamed Faten Zhani SDN-Guard: DoS Attacks Mitigation in SDN Networks, 5th IEEE International Conference on Cloud Networking, 2016.

[14] Tri-Hai Nguyen, Myungsik Yoo, Attacks on Host Tracker in SDN Controller: Investigation and Prevention, Information and Communication Technology Convergence (ICTC), International Conference, 2016.

[15] Mutalifu Kuerban, Yun Tian, Qing Yang, Yafei Jia, Brandon Huebert and David Poss, FlowSec: DOS attack Mitigation Strategy on SDN Controller, Networking, Architecture and Storage (NAS), IEEE International Conference, 2016.

[16] P. Floodlight, Floodlight OpenFlow controller, Project Floodlight, 2016. [Online] Available http://www.projectfloodlight.org/floodlight/

[17] Chaitanya Buragohain, Nabajyoti Medhi, FlowTrApp: An SDN Based Architecture for DDoS Attack Detection and Mitigation in Data Centers, 3rd International Conference on Signal Processing and Integrated Networks (SPIN), 2016.

[18] J. Matias, T. Borja, M. Alaitz, E. Jacob, T. Nerea, Implementing layer 2 network virtualization using OpenFlow: challenges and solutions. In: Proceedings of the 2012 EuropeanWorkshop on Software Defined Networking, 2012, PP. 30-35.

[19] M. Canini, D. Venzano, P. Pereˇs´ıni, D. Kosti´c, J. Rexford, A NICE Way to Test OpenFlow Applications. In: Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation (NSDI), San Jose, CA, USA, 2012, pp. 1-14.

[20] D. Kreutz, F.M.V. Ramos, P. Verissimo, Towards secure and dependable software-defined networks. In: Proceedings of the 2nd ACM Workshop on Hot Topics in Software Defined Networking (HotSDN), Hong Kong, China, 2013, pp. 55-60.

[21] X. Wen, Y. Chen, C. Hu, C. Shi, Y. Wang, Towards a secure controller platform for openflow applications. In: Proceedings of the 2nd ACM Workshop on Hot Topics in Software Defined Networking (HotSDN), Hong Kong, China, 2013, pp. 171-172.

[22] Natasha Gude, Teemu Koponen and Justin Pettit, NOX: Towards an Operating System for Networks, ACM SIGCOMM Computer Communication Review archive Volume 38 Issue 3, July 2008, Pages 105-110.

[23] H. Hu, W. Han, G.-J. Ahn, Z. Zhao, FLOWGUARD: Building robust firewalls for software-defined networks. In: Proceedings of the 3rd Workshop on Hot Topics in Software Defined Networking (HotSDN), Chicago, Illinois, USA, 2014, pp. 97-102.