Firewalls Policies Based on Software Defined Networking: A survey

Sailen Dutta Kalita[1], Rupam Kumar Sharma[2]
[1,2] Department of Computer Science and Engineering and Information Technology,
School of Technology, Assam Don Bosco University,
Azara, Guwahati, Assam, India
[1]sailen.dk@gmail.com.[2]rupam.sharma@dbuniversity.ac.in

Abstract: ***Software-Defined Networking (SDN) introduces granularity, visibility and flexibility to networking, which separates the control-logic from networking devices. SDN programmatically modifies the functionality and behaviour of network devices. It separates control plane and data plane, and thus provides centralized control. Though SDN provides better performance but there are some security issues that need to be taken care of. This includes firewalls, monitoring applications, IDS(Intrusion detection systems) etc. Therefore, this research work reviews the related approaches which have been proposed by identifying their firewall scope, their practicability, their advantages and drawbacks related with SDN. This paper describes the firewall policies as the forth new security challenges.***

Keywords: *Software defined networking, Architecture, OpenFlow, Firewalls, Anomaly detection.*

1. Introduction

Network Security is an important issue of today's Networking**.** Recently, Software Defined Networking is an evolving technology that decouples the Control from forwarding network devices like switches, routers, hubs etc. SDN splits data plane and control plane . It controls the flow of data through high level program. Since in SDN network control is done programmatically network security is more vital task .This paper reviews how SDN can play a major role in implementing the network security functions mainly Firewall policies.

2. Openflow

OpenFlow has been introduced in [19] is tends to be the de facto standard for SDN. This is developed by the Open Networking Foundation.. With OpenFlow, every switch communicates to a controller in order to install, delete or modify flow-based rules to process incoming packets. Examples of controllers are NOX, POX or Floodlight etc. Installing a rule creates an entry in the switch flow table. Each entry is composed of two main components, match fields and Instructions fields. Match fields is a filter to match packet headers like Ethernet addresses, TCP ports, IP addresses, Time-to- Live value .Instructions field is a definition on how to handle an incoming packet by matching the rule and it is composed of a set of actions to send the packet towards a single or multiple ports, to drop or to apply some header modifications . In addition, the flows entries maintain counters indicating number of matched packets, number of bytes, number of errors.

3. Firewalls

Firewalls are essential components in a network that act as a first level of access control. It protects the local network from other hosts in Internet, which are not trustworthy. Packet filtering can be done at the different levels of the network layer. Most of the packet filtering are analyze packet headers up to the transport layer but there also exist application level firewalls. There are very few switches now supporting layer 7.

As a conclusion, achieving traffic filtering above layer 4 can be enabled by OpenFlow by inspecting the packets at the controller side assuming that all incoming packets are forwarded to it with Packet In messages. So traffic filtering clearly increases the latency and thus only viable by limiting the analysis to a few packets.On the other hand, the inspection on highest layers in switches is also contradictory with the SDN paradigm which aims at keeping switches as forwarding devices only because other processes are resource consuming.

Firewall may be Stateless firewall or statefull firewall.

4. Stateless firewall

A stateless firewall filters the packets based on the values in the headers like the IP address or the port numbers and decide to accept or drop. It does not check the status of a connection to check the legitimacy of a packet.

5. Stateful firewall

In case of a TCP-based stateful firewall, only incoming packets part of a flow which has been initiated by a machine of the local network are usually allowed.

6. Literature reviews

One of the fundamental challenges of SDN is to build robust firewalls for protecting OpenFlow-based networks where network states and traffic are frequently changed. An example of SDN firewall application has been introduced in Floodlight [1] where each packet-in behavior triggered by the first packet of a traffic flow which is matched against a set of existing firewall rules that allow or deny a flow at its ingress switch. This preliminary implementation of OpenFlow-based firewall application can only examine flow packet violations when new flows come in the network but it cannot check flow policy violations with respect to dynamic network policy updates.

To build robust firewall, Alaauddin Shieha introduce FLOWGUARD [2], is a comprehensive framework, to facilitate accurate detection as well as effective resolution of firewall policy violations in dynamic OpenFlow-based networks.when network states are updated, FLOWGUARD checks network flow path spaces to detect firewall policy violations. In addition, with the help of several innovative resolution strategies designed ,it conducts automatic and real-time violation resolutions for diverse network update situations. ALAAUDDIN SHIEHA also implement his framework and demonstrate the efficiency of the proposed detection and resolution approaches in FLOWGUARD through experiments with the help of a real-world network topology.

In [3], an earlier solution for building a security enhanced firewall application was introduced where it only focuses on addressing bypass threats in OpenFlow-based networks. In contrast, FLOW- GUARD is a comprehensive framework for building robust SDN firewalls to enable both accurate detection and resolution of various firewall policy violations in dynamic OpenFlow-based networks.

In the Frenetic Project [4], a higher-level language know as Pyretic [4] was recently introduced to allows SDN programmers to write modular network applications. By compiling conflicting policies into a prioritized rule set, Pyretic's sequential composition operator could potentially resolve direct policy conflicts. However, it cannot discover and resolve indirect security violations caused by dynamic packet modifications without a flow tracking mechanism [4].

FortNOX [5] was proposed as a software extension to provide security constraint enforcement for OpenFlow controllers. It can identify indirect security violations. However, They can not directly adopt FortNOX approach to design SDN firewalls due to several reasons. One of the reason is that the rule conflict analysis algorithm provided by FortNOX records rule relations in alias sets, which are unable to accurately track network traffic flows. In particular, the conflict detection algorithm in FortNOX only conducts pairwise conflict analysis between new flow rules and each single security constraint without considering rule dependencies within flow tables [6] and among security constraints [6]. The second reason is, when FortNOX detects a security violation caused by new rules installed by a non-security application, it simply rejects the rules without offering a fine grained violation resolution.

A couple of verification tools [7, 8, 9] for checking network invariants and policy correctness in OpenFlow networks have been already proposed. Mainly, VeriFlow [7] and NetPlumber [7] are capable of checking the compliance of network updates with specified invariants in real time. Even though these tools can be potentially used to detect firewall policy violations, they could not support automatic and effective violation resolution. They also ignore rule dependencies within security constraints, such as firewall policies, for compliance checking. In addition, they are also unable to check stateful network properties [10].

There are number of firewall algorithms and tools have been designed to assist system administrators to managing and analyzing firewall policy anomalies [6]. Yuan et al. [6] presented, a toolkit known as FIRE- MAN to check for misconfigurations in firewall policies through static analysis.

In [11, 12] introduced FAME, a visualization-based firewall anomaly management environment, to detect and resolve of firewall anomalies. However, existing firewall policy analysis tools can only detect policy anomalies within a firewall policy, but cannot be directly applied to deal with firewall policy violations against flow entries in dynamic Open- Flow networks enviroment with respect to network-wide access control.

In [14],an overview of machine learning techniques for anomaly detection proposed. Their experiments demonstrated that the supervised learning methods significantly outperform the unsupervised ones if the test data contains no unknown attacks. The best performance is achieved by the non-linear methods, such as SVM, multi-layer perceptron among the supervised methods and the rule-based methods. Techniques for unsupervised such as K-Means, SOM, and one class SVM achieved better performance over the other techniques although they differ in their capabilities of detecting all attacks classes efficient.

In [20] they shown the various categories of security threats associated with SDN layered framework defined in SDN architecture. They defines different security attacks like Data leakage i.e. spoofing, unauthorized access, denial-of-service, data modification, malicious applications etc. that are possible at different parts of SDN framework.

SDN Scanner [21] acieve the network header field change scanning, it scans networks as changing header fields and then records the response time of each packet. After that it compares the response time and use statistical tests.It is possible to conduct resource consumption attack on SDN with almost 85.7% accuracy with the fingerprinting results of SDN. So new defence solutions we need to be designed to overcome such threats.

AvantGuard [22] proposed new architecture as data plane extensions to protect network from control plane saturation attack that disrupts network operations. It introduces connection migration by actuating triggers over the data plane's existing statistics collection services. AvantGuard provides both detection of, and responses to, the changing flow dynamics within the data plane.The Connection migration enables the data plane to protect the control plane from saturation attacks.

AMQ [23] proposed a technique to detecting and isolating insecure network devices in Data centres, before they effect negatively to the network. After discovering a potential threat, it automatically identifies the problem and download the patches necessary to resolve it. AMQ automatically allows the device to re-join the network on resolution. In AMQ,there are two primary security network service modules (NSM) hosted on the controller. First one is a Botunter monitor the network and detect a malware infected host in real time. Secondly one is a threat responder NSM that directs the controller to initiate the quarantine procedure to isolate the threat.The Web Proxy notifier is activated to inform the user on the infected host that security has been compromised,when a host is quarantined. It can be used for moderate speed links only.

In [24], the authors propose a language to define firewall rules relying on the POX controller. The rules are installed in a reactive way. The evaluation shows that the implemented functionalities work, allowing or blocking traffic, but does not assess any performance metric, for instance about underlying introduced delays.

There are Several other recent efforts have been introduced to address various security challenges, such as vulnerability assessment [13], DDoS attack detection [15], and saturation attack mitigation [10], scanning attack prevention [7], in SDNs.

7. Conclusions

SDN become a very efficient technology which is going to be the future of networking.SDN provides flexibility by programming the control, including the centralized control, which helps in handling the whole network. It becomes more beneficial in case of synchronization, controlling, providing scalability and management of data in large data centres. Also the Abstraction and Virtualization of resources helps in securing the network and hiding complexity. In SDN there are still different areas which are required to be taken care of, like securing the SDN control plane since as whole control

of SDN is centralized in control plane; hence security of Control plane is very important and prevention from several attacks is main area of concern at present.The openness of SDN system allows to write control programs so it is essential to design some protocols or use existing protocols efficiently that will check the correctness of programming logic before implementation of SDN .Also the Security of southbound interface needs special attention as control transfers through this interface.To secure the SDN,the firewalls policies needs to be taken care of.

8. References

[1]. Risdianto, Aris Cahyadi, and Eueung Mulyana. "Implementation and analysis of control and forwarding plane for SDN."*Telecommunication Systems, Services, and Applications (TSSA), 2012 7th International Conference on*.IEEE, 2012.

[2]. Hu, Hongxin, et al. "Towards a reliable sdn firewall." *Presented as part of the Open Networking Summit 2014 (ONS 2014)* (2014).

[3].Wang, Juan, et al. "Towards a security-enhanced firewall application for openflow networks." *Cyberspace Safety and Security*.Springer International Publishing, 2013.92-103.

[4].Khurshid, Ahmed, et al. "Veriflow: verifying network-wide invariants in real time." *ACM SIGCOMM Computer Communication Review* 42.4 (2012): 467-472.

[5]. Braga, Rodrigo, Edjard Mota, and Alexandre Passito. "Lightweight DDoS flooding attack detection using NOX/OpenFlow." *Local Computer Networks (LCN), 2010 IEEE 35th Conference on*.IEEE, 2010.

[6]. Yuan, Lihua, et al. "Fireman: A toolkit for firewall modeling and analysis." *Security and Privacy, 2006 IEEE Symposium on*.IEEE, 2006.

[7]. Kazemian, Peyman, et al. "Real Time Network Policy Checking Using Header Space Analysis." *NSDI*. 2013.

[8]. Kazemian, Peyman, George Varghese, and Nick McKeown. "Header Space Analysis: Static Checking for Networks." *NSDI*. 2012.

[9]. Mai, Haohui, et al. "Debugging the data plane with anteater." *ACM SIGCOMM Computer Communication Review* 41.4 (2011): 290-301.

[10]. Stoenescu, Radu, et al. "Symnet: Static checking for stateful networks." *Proceedings of the 2013 workshop on Hot topics in middleboxes and network function virtualization*. ACM, 2013.

[11]. Hu, Hongxin, Gail-Joon Ahn, and Ketan Kulkarni. "Fame: a firewall anomaly management environment." *Proceedings of the 3rd ACM workshop on Assurable and usable security configuration*.ACM, 2010.

[12]. Hu, Hongxin, Gail-Joon Ahn, and Ketan Kulkarni. "Detecting and resolving firewall policy anomalies." *Dependable and Secure Computing, IEEE Transactions on* 9.3 (2012): 318-331.

[13]. Wang, Zhiliang, et al. "Analysis of Comparisons between OpenFlow and ForCES." *ForCES, IETF* (2012).

[14].Omar, Salima, Asri Ngadi, and Hamid H. Jebur. "Machine learning techniques for anomaly detection: an overview." *International Journal of Computer Applications* 79.2 (2013).

[15] François, Jérôme, et al. "Network security through software defined networking: a survey." *Proceedings of the Conference on Principles, Systems and Applications of IP Telecommunications*.ACM, 2014.

[16]. N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner. Openflow: enabling innovation in campus networks. *ACM SIGCOMM Computer Commun. Review*, 38(2):69–74, 2008

[17] Shalimov, Alexander, et al. "Advanced study of SDN/OpenFlow controllers." *Proceedings of the 9th Central &Eastern European Software Engineering Conference in Russia*.ACM, 2013.

[18]. Al-Shaer, Ehab S., and Hazem H. Hamed. "Discovery of policy anomalies in distributed firewalls." *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*.Vol.4.IEEE, 2004.

[18]. Al-Shaer, Ehab S., and Hazem H. Hamed. "Discovery of policy anomalies in distributed firewalls." *INFOCOM 2004.Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*.Vol.4.IEEE, 2004.

[19].H. Hu, G.-J. Ahn, and K. Kulkarni. Detecting and resolving firewall policy anomalies. IEEE Transactions on Dependable and Secure Computing, 9(3):318–331, 2012.

[20]Shin, Seungwon, and Guofei Gu. "Attacking software-defined networks: A first feasibility study."*Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking*. ACM, 2013.

.21. S. Shin, V. Yegneswaran, P. Porras, G. Gu, "AVANT-GUARD: Scalable and Vigilant Switch Flow Management in Software-Defined Networks", ACM Proc. of CCS, Berlin, Germany, pp. 413-424, 2013.

22. M. McBride, M. Cohn, S. Deshpande, M. Kaushik, M. Mathews, S. Nathan, "SDN Security Considerations in the Data Center",Open Networking Foundation- ONF SOLUTION BRIEF, 2013.

23. Y. Zhang, "An adaptive flow counting method for anomaly detection in SDN", ACM Proc. of CoNEXT, Santa Barbara, California, USA December, 2013.

[24] M. Suh, S. H. Park, B. Lee, and S. Yang.Building firewall over the software-defined network controller.In Advanced Communication Technology (ICACT), 2014 16th International Conference on, 2014.

25. S. A. C. Risdianto, E. Mulyana, "Implementation and Analysis of Control and Forwarding Plane for SDN", IEEE Telecommunication Systems, Services, and Applications (TSSA) 7th International Conference,pp. 227 – 231, 2012.

26. N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar , L. Petrson , J. Rexford, S. Shenker , and J. Turner , "OpenFlow :Enabling innovation in campus networks", ACM SIGCOMM Computer Communication, Vol. 38, Issue 2, pp. 69-74, 2008.

[27].Garg, Gagandeep, and Roopali Garg. "Detecting anomalies efficiently in SDN using adaptive mechanism."*Advanced Computing & Communication Technologies (ACCT), 2015 Fifth International Conference on*. IEEE, 2015.

[28].Ng, Bryan, Matthew Hayes, and Winston KG Seah. "Developing a Traffic Classification Platform for Enterprise Networks with SDN: Experiences & Lessons Learned."

[29]. Cho, Hyunhun, et al. "An Optimal Path Computation Architecture for the Cloud-Network on Software-Defined Networking." *Sustainability* 7.5 (2015): 5413-5430.

[30]. Bozakov, Zdravko, and Panagiotis Papadimitriou. "Towards a scalable software-defined network virtualization platform."*Network Operations and Management Symposium (NOMS), 2014 IEEE*.IEEE, 2014.

AuthorsProfile

Sailen Dutta Kalita,Pursuing M.Tech in Department of Computer Science and Engineering and Information Technology, School of Technology, Assam Don Bosco University, Azara, Guwahati, Assam, India.

Rupam Kumar Sharma, working as Assistant Professor, in Department of Computer Science & Engineering and Information Technology, SOT, ADBU. He is also pursuing his research work at NEHU