Firewalls construction for Software-Defined Networks

Ashok Kumar Reddy Nadikattu

Data Scientist & Department of Information Technology

California USA

Firewalls construction for Software-Defined Networks

Abstract

A firewall plays a significant role as a security device for the computer hardware and software; it protects the computer network from unauthorized access through filtration and blocking access from outsiders. It filters both unwanted traffic and also blocks any malicious software from attacking the computer system. There are different levels of protection provided by firewalls. On the other hand, the software-defined network plays a role in introducing significant granularity and visibility to the networking field. However, it is supposed to undergo some changes in network security to have its operation protected from unauthorized access and malicious attacks. Security changes that can be implemented in the software-defined network include protection of the Open Flow-based networks. Protection of the Open Flow-based network involves frequent changing of the network states and traffic. Protection of the network sites and traffic can be addressed by introducing the FLOWGUARD, a comprehensive framework required to facilitate right identification and provide adequate correction of the violations in Open Flow-based networks. The FLOWGUARD system acts by checking all network flow systems in the path spaces to determine whether there is any violation of the Firewall policy during updating of the network states. The system also performs an automatic solution to the violation of the Firewall policy. Implementation of the FLOWGUARD, including its efficacy and effectiveness in the provision of resolutions to Firewall violation and the various challenges that arise in using the Open Flow system, are discussed in this paper.

Keywords: Network monitoring, Open Flow, Firewalls. Software-Defined Networking, security

Introduction

Software-define Networking is responsible for enabling network controllers to run different network services by configuring the data handling process in the various network devices. The network controller is also supposed to control the networking process by directly configuring the packet-handling processes in the networking devices (Arashloo et al., 2016). Most companies use the Open Flow system to manage their networks since t is cost-effective and saves time. Open FlowFlow is also used to detect any violation in the Firewalls and any intrusion and prevention of the running of

networking systems. Firewalls form part of the network security in that they represent the first line of defense in any computing company. The device monitors the operating system of the computer and blocks outside access to the device (Akhunzada et al., 2016). It also acts as a filter and a barrier between the network system of an individual's computer and another network. The operating system of a computer and the security software is usually pre-installed in the Firewall.

The Firewall works by analyzing all the traffic network-based rules and only allows incoming connections configured in the networking device. The firewall scrutinizes the incoming connections through blockage of specific data packets and only allows trusted sources to be allowed into the computer's operating system (Jararweh et al., 2016). Also, it works by identifying the IP addresses, which are crucial in a computer as they help identify the source or the computer device. Different types of firewalls can be used for security purposes, including the software and the hardware firewalls. The software firewall is an internal program installed in the computer system which works through port figures and various applications installed in the computer system.

On the other hand, the hardware firewall is physical and stored between the network and the gateway. It appears like a broadband router. Other types of firewalls include cloud-based firewalls, which are commonly called the Firewall as a Service. These are similar to the hardware firewalls (Wang, 2016). Packet filtering firewalls are software firewalls that act as programs installed in the computer, and they act by blocking the network IP protocol, the IP address, and the port figure (Abbes et al., 2016). It is mainly used for smaller networks (Scheid, 2016). Stateful multilayer inspection firewalls are responsible for keeping track of the established connections; filter traffic based on the port, protocol, and state.

The open Flow system is commonly used. However, there are different challenges encountered for the construction of software-based networks. The challenges are experienced in various areas, including examining dynamic network policy updates- network systems are updated frequently in an open flow network. They are also configured upon updating (Sood, 2015). Checking the violation of the flow packet s done by monitoring the behaviors in the Firewall. However, a significant challenge in this area is that checking the violation in the flow packet through monitoring of the packet in properties in a firewall policy is ineffective (Kaur et al., 2016). The violation of the flow policy is supposed to be detected, and an appropriate solution is made within real-time. There are challenges experienced in the checking of indirect security violations. The open FlowFlow accepts various set field actions which have the capability of changing the packet headers. However, there is a high probability of adversaries strategically leveraging the flow rules to cause the system to evade the networking mechanisms, which are the firewalls (Hu, 2016). There are also chances of overlapping the flow rules. There are also increased chances of the Software-Defined network firewall to immediately force new rules in the firewall policy to detect for any violations. Software-based Networking may also resolve the flow policy through propagation and the enforcement of a firewall policy (Fayaz et al., 2016). The Open Flow

stem also provides less opportunity to the access to packet-level data in the controller, thereby making it challenging to supply support to the stateful packet inspection process in the software-defined firewalls.

Literature Review

Various research studies have been carried out to address the challenges faced when using the Open Flow system. Some of the challenges that have been researched and solutions addressed include scanning attack prevention, assessing the vulnerability, saturation attack mitigation, and detecting DDoS attack in the Software-defined Network. Various strategies have been put in place to design an effective Software-defined network firewall that supports access to comprehensive network control. Various design requirements such as accuracy have been addressed (Gurtov et al., 2016). The design requirements for the SDN firewall include accuracy- the SDN firewall is required to detect precisely any violation in the system caused by traffic modification and rule dependencies. It is also required that once the mistakes are identified, effective corrections should be done in real-time.

FLOWGUARD design

Flexibility is another element required in the SDN firewall. The SDN firewall needs to have the ability to inspect available networks and configure different changes. The Firewall is also supposed to be efficient in its operation; it should continuously work promptly. The FlowGuard, therefore, has to be effectively designed to accumulate the requirements for the SDN Firewall (Arashloo et al., 2016). The FlowGuard should address significant change challenges in the SDN firewall to allow for the detection of violations and correction of violated policies.

Detection of violation

Violations in the flow packets can be detected using ordinary techniques for the firewall packet filtering process. Detection of violation in the FLOWGUARD system involves examination of the space of the flow path against the authorization space of the Firewall. The FLOWGUARD can detect violations by careful monitoring of flow paths in the whole network system concerning the changes in the packet filter. It also checks rules overlaps in the flow tables and the firewall policies (Kumar, 2016). The SDN firewall needs to check violations of each flow system's ingress switch and track the flow path states and determine the origin and the end source. The detection process can occur through various measures as discussed below:

Flow path space analysis

The above include tracking of the flow network whereby the firewall application figures out the origin of the address and its final destination for each of the flow networks by tracking the low path. Also, an effective flow track mechanism needs to be put in place to identify the flow paths in the system. Different network modification tools can help in the flow analysis process, including verification tools which help check of the reliability in real-time and help find the flow paths in the Open Flow networks (Balan et al., 2016). The leverage strategy used in this paper is the use of Header Space Analysis as a tool for verification of flow paths through the flow tracking process. The research landed on the HSA tool because it contains different features required for effective tracing of flow systems. The features include the header space,

which is the geometric model of analysis of packet processing. In this model, a protocol-independent network model is provided. The HSA can model network boxes to support the different packet modifications by switching transfer functions. HSA is also in n position of constructing a graphical representation of the intra table dependencies rules where the indirect and direct flow paths are automatically captured.

Flow Path Space Calculation

The difference between the fields needed for the assessment of firewall policy violation and the flow policy procedure is calculated to get the flow path space. The difference between the two gives the flow path space (Ibarra et a., 2016). A source recognizes the fields of policy. For example, the source-destination and the source address are designated as Ps and Pd. The Ps and Pd specify the flow path space and using the IP 5 tuple sense, and the source value address has values from three fields, including the source IP, source port, and the protocol (Kalita & Sharma, 2016).

Firewall Authorization Space Partition

There are cases where the system administration intentionally introduces overlaps in the firewall rules. Space partition is mainly used to eliminate specific sections from a particular action in the firewall rules. For accurate detection of the firewall policy violation in the Open Flow networks, there should be connection between "allow" and "deny" policies, and these rules should be decoupled (Neu et al., 2016). The firewall authorization space is introduced to bring out the idea of collecting all packets from either allow or deny firewall rules. Various rules for the header space are formulated to perform the various set operations on the rules.

Violation discovery

The violation of the firewall rules is detected after the flow path space is calculated and the firewall policy of firewall authorization space is determined. The violations are detected by evaluating the tracked space of the flow path. The tracked space allows the flow to pass through the network against the denied authorization space. In case there are overlaps between the headers space and the tracked space, then the overlapping space is referred to as the violated space (Li et al., 2016). Violations can be of two types; the entire violation occurs if the denied authorization space is inclusive of the entire tracked space. The partial violation occurs when the denied authorization space partially includes the tracked space.

Violation Resolution

The Software-defined network has the capability of rejecting new flows that violate the firewall policy. The system goes ahead to correct the flow packet violations. It does this by either disabling the violated inflows by rejecting the installing a particular policy- for the new policy. The already existing policies in the system that violate the floe rules are resolved by directly removing the network. However, direct removal of the network has disadvantages (Abbes, et al., 2016). For example, in case of partial violation, direct removal of the network affects the network utility services. There are also chances of impacting other flow policies in the system. Even the creation of a new violation sine a rule in the flow system may sometimes depend on other policies in the system. Therefore, a flexible resolution of the policy violation should be applied to reduce the negative

effects associated with the direct deletion of the network.

A suitable violation resolution mechanism is the use of a comprehensive violation resolution process. In this mechanism, four different strategies are used in the resolution of the violation.

First, dependency breaking- this entails the overlapping of the new policies introduced in the system. The dependency can be broken using the flow rerouting process where the Firewall can detect any violation in the new flow policy and asks for the control system to use another routing path for the same flow to avoid overlaps.

Flow tagging- Dependency can be broken through the flow tagging process where the new flow policy undergoes reprocessing by adding a tag as a differentiator between the match pattern from the other rules.

Update rejecting- this is also a comprehensive mechanism of violation resolution. It is applied in three possible situations, including a new policy, changing the rule which induces an entire violation, and deleting a rule that creates a new entire violation in the system. The update rejecting mechanism is suitably applied in the three scenarios. In this case, the update operation is directly rejected once a violation is sensed.

Flow Removing- the strategy is applied in the following scenarios when updating a new policy in the firewall policy, which is detected to be a violation. It is also applied when a change or deletion operation on a rule is allowed, even when there is an entire violation. The mechanism is mostly used in the flow paths that entirely violate the firewall policy. These flow paths are entirely removed from the network switches.

Packet blocking is applied in case of partial violation, which is detected by the firewall systems. It is applied in old and new flow policies where the firewall application wants to block the fire packets in both the ingress and egress switch.

Conclusion

Firewalls are essential in the protection of the computers from various insecurities and unauthorized access to the machines. Therefore, any detection made by the firewalls in the violation of the flow policies should be resolved within time. Either entire or partial violation, they need to be corrected. Therefore, a suitable choice of correction mechanism should be applied to avoid creating a new entire or partial violation. In this paper, the comprehensive violation mechanism centrally enforces the firewall policies to eliminate all the flow packets. For partial flow policy violation, the blocking mechanism in the FlowGuard requires that the firewall rules are propagated and enforced in the egress and ingress switches of the network. Therefore, the FlowGuard system should use a hybrid architecture to build the Software-Defined Network firewalls to facilitate effective violation resolution. The information in the paper will be useful in the United States as the states will be able to develop and integrate stateful packet inspection and be able to incorporate FLOWGUARD framework as a support model for the stateful firewall for the Software-defined Networks.

References

1. Abbes, T., Bouhoula, A., & Rusinowitch, M. (2016). Detection of firewall configuration errors with updatable tree. *International Journal of Information Security*, *15*(3), 301-317.

2. Akhunzada, A., Gani, A., Anuar, N. B., Abdelaziz, A., Khan, M. K., Hayat, A., & Khan, S. U. (2016). Secure and dependable software defined networks. *Journal of Network and Computer Applications*, *61*, 199-221.
3. Arashloo, M. T., Koral, Y., Greenberg, M., Rexford, J., & Walker, D. (2016, August). SNAP: Stateful network-wide abstractions for packet processing. In *Proceedings of the 2016 ACM SIGCOMM Conference* (pp. 29-43).
4. Balan, T., Zamfir, S., Robu, D., & Sandu, F. (2016, June). Contributions to content-based software defined networks. In *2016 International Conference on Communications (COMM)* (pp. 159-162). IEEE.
5. Fayaz, S. K., Sharma, T., Fogel, A., Mahajan, R., Millstein, T., Sekar, V., & Varghese, G. (2016). Efficient network reachability analysis using a succinct control plane representation. In *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)* (pp. 217-232).
6. Gurtov, A., Liyanage, M., & Korzun, D. (2016). Secure communication and data processing challenges in the Industrial Internet. *Baltic Journal of Modern Computing*, *4*(4), 1058-1073.
7. Hu, S. (2016). Survey on Cross Layer Design and Big Data in Software Defined Networking: Problems And Solutions. *Published on ReserachGate.[Online. Available: https://www. researchgate. net/publication/310124217_Survey_on_Cross_Layer_Design_and_Big_Data_in_Software_Defined_Networking_Problems_and_Solutions*.
8. Ibarra, J., Bezerra, J., Lopez, L. F., Morgan, L., & Cox, D. (2016). Responding to the demands of big data scientific instruments through the development of an international software defined exchange point (SDX). *UbuntuNet Connect-Connect 2016*.
9. Jararweh, Y., Al-Ayyoub, M., Benkhelifa, E., Vouk, M., & Rindos, A. (2016). Software defined cloud: Survey, system and evaluation. *Future Generation Computer Systems*, *58*, 56-74.
10. Kalita, S. D., & Sharma, R. K. (2016). Firewalls Policies Based on Software Defined Networking: A survey. *ADBU Journal of Engineering Technology*, *4*.
11. Kaur, S., Kaur, K., & Gupta, V. (2016, October). Implementing openflow based distributed firewall. In *2016 International Conference on Information Technology (InCITe)-The Next Generation IT Summit on the Theme-Internet of Things: Connect your Worlds* (pp. 172-175). IEEE.
12. Kumar, R., & Nicol, D. M. (2016, November). Validating resiliency in software defined networks for smart grids. In *2016 IEEE International Conference on Smart Grid Communications (SmartGridComm)* (pp. 441-446). IEEE.
13. Li, W., Meng, W., & Kwok, L. F. (2016). A survey on OpenFlow-based Software Defined Networks: Security challenges and

countermeasures. *Journal of Network and Computer Applications*, *68*, 126-139.

14. Neu, C. V., Zorzo, A. F., Orozco, A. M., & Michelin, R. A. (2016, December). An approach for detecting encrypted insider attacks on OpenFlow SDN Networks. In *2016 11th International Conference for Internet Technology and Secured Transactions (ICITST)* (pp. 210-215). IEEE.
15. Scheid, E. J., Machado, C. C., dos Santos, R. L., Schaeffer-Filho, A. E., & Granville, L. Z. (2016, June). Policy-based dynamic service chaining in Network Functions Virtualization. In *2016 IEEE Symposium on Computers and Communication (ISCC)* (pp. 340-345). IEEE.
16. Sood, K., Yu, S., & Xiang, Y. (2015). Software-defined wireless networking opportunities and challenges for Internet-of-Things: A review. *IEEE Internet of Things Journal*, *3*(4), 453-463.
17. Wang, Z., Tao, D., & Lin, Z. (2016, December). Dynamic virtualization security service construction strategy for software defined networks. In *2016 12th International Conference on Mobile Ad-Hoc and Sensor Networks (MSN)* (pp. 139-144). IEEE.