Stateful Firewall Application on Software Defined Networking

Nan Haymarn Oo, Aung Htein Maw
University of Computer Studies, Yangon , University of Information Technology
nanhaymanoo@ucsy.edu.mm, ahmaw@uit.edu.mm

Abstract

Software defined networking(SDN), one of the advanced networking technology, is more flexible, faster and more secure than traditional network technology because of using OpenFlow in separating the network control operations from the forwarding devices to develop networks. However, many security issues still exist in the SDN architecture. To solve these issues, firewall is one of the most important SDN security applications. Firewall can be generally categorized into stateless and stateful depending on the capability of connection state tracking. Stateless firewall does not check the states of the connection session. As a result, it has some limitations in solving the security issues. But this limitation can be overcome by stateful firewall. An acl application has been implemented in Open Network Operating System (ONOS) as a stateless firewall. In order to increase the security level of acl application in ONOS, this paper proposes a stateful firewall. The firewall application is implemented by taking into account both security and performance perspectives in order to be a proper SDN security application. Finally, the experiment will be conducted how proposed stateful firewall tackle the security issues and affect the performance by comparing with acl application.

Keywords- Stateless Firewall, Stateful Firewall, SDN, ONOS

1. Introduction

SDN is a network with three-tier architecture: data plane, control plane, and application plane. SDN switches existing in data plane do not have intelligent - i.e., how to transmit packet among them. Thus they pass the packet to the controller in control plane. The controller passes the packet again to the respective application. The application produces flow rules for the packet and installs them into the switches via controller.

Open Vswitch is commonly used as forwarding device. For firewall application, it can act as a stateless firewall because it has OpenFlow table including the fields such as source MAC address, destination MAC address, source IP address, destination IP address, source Port, destination Port, action and count. But it does not have state field and inspect the state of the packet. Therefore, packets are needed to send to the controller in order to check the state of the packet in implementation of the stateful firewall application.

Firewall application sets the flow rules depending on the firewall rules set. It can be implemented to install necessary flow rules either by itself or by using the help of forwarding application. The forwarding application can be differentiated into two types: forwarding packets within the same network, and routing packets among the different networks. This paper only emphasizes on the packet forwarding within the same network. Hence, the stateful firewall application proposed in this paper uses the reactive forwarding application as its assistant in installation of flow rules.

According to the stateless and stateful firewall, flow rule installation depends on whether it inspects the connection state or not. For stateless firewall application, it makes decision of flow rules installation by considering only the firewall rules set. But, stateful firewall application takes into account both its firewall rules set and state of each packet when setting the flow rules into switches. Therefore, stateful firewall can be protected the attacks more than stateless one.

The acl application in ONOS controller installs flow rules with different manners according to the action of acl rule as shown in figure 1. The application solely installs permanent drop flow rules for deny action acl rules in the source switch. For allow action acl rules, the application installs permanent flow rules with To Controller output action. Such flow rules installation means that the acl application delegates the packet forwarding to the default fwd application running in controller. The forwarding application installs temporary forwarding flow rules. It removes the installed flow rules after flow timeout value expire. It does not consider any state of the connection session during the flow rules installation and removing.

Therefore, in order to overcome the main limitation of acl application - lack of connection state inspection, this paper proposes stateful firewall application by adding connection state inspection into the acl application. State-aware application has to send packets to controller to check state of the connection session, though, the proposed stateful firewall application only send necessary packet according to both connection tracking and SDN nature in order not to reduce performance while increasing security level.

ISBN 978-99971-0-381-9

As ONOS controller[1] has distributed and topology-aware nature, it's acl application has tackled the problems such as single point of failure occurs in centralized firewall, less sensitive to topology change, complicated firewall configuration, additional cost in rule maintaining, and longer rule matching time that may occur in simple distributed firewall.

The rest of this paper is organized as follows: section 2 lists the related work of this paper. Section 3 presents stateless vs stateful firewall. And section 4 introduces the process of proposed stateful firewall application. Section 5 describes the testbed for experiment. Section 6 evaluates the results of the proposed stateful firewall application and existing acl application. Section 7 describes conclusion and future works.

2. Related Work

Software-Defined Network (SDN), new modern network, has many security issues. In order to countermeasure them, firewall application has been implemented on every type of controller. Two types of firewall with inefficient and efficient ways has been created on Ryu[2]. [3] proposed reactive stateful firewall with a global orchestrator on Ryu controller. Stateless firewall application also has been implemented on Floodlight[4]. And its stateful firewall application has been researched[5]. Likewise, acl application is written in ONOS controller. It can be said this application as stateless firewall because it filters the packet by using the rules that are the same as stateless firewall rules including source IP, destination IP addresses, source port, destination port and action. Stateful firewall on ONOS controller has proposed and researched to be able to early block the DDOS attack especially for the Internet Service Provider Network[6].

Early firewall applications are implemented as centralized applications according to the SDN architecture. But the centralized applications have weaknesses such as single point of failure, controller overhead, and overloaded communication between control plane and data plane. In order to overcome those weaknesses, firstly, distributed firewall applications have been proposed[7]. Although these applications can tackle the single point of failure problem, they can cause further problems: less sensitive to topology change, complicated firewall configuration, additional cost in rule maintaining, and longer rule matching time. Therefore, later firewall applications are implemented as a topology-aware selectively distributed firewall[8-9].

In this paper, stateful firewall application is mainly implemented based on the ONOS acl application in order to reduce communication between control plane and data plane, and controller overhead while adding connection state inspection.

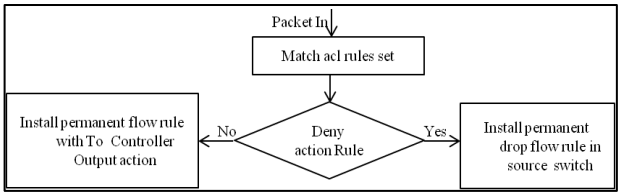

Figure 1. Flow chart of overall acl application

3. Stateless Firewall vs. Stateful Firewall

Stateless firewall or packet filtering firewall allows or denies the packets according to the action defining in firewall policy or firewall rules set. The filtering rule in firewall policy includes layer-3 and layer-4 information such as source IP, destination IP, destination port, protocol type and action. It does not inspect the state information including in the packet. Consequently, it may allow the SYN_ACK packet without sending its prior SYN packet.

Stateful firewall itself can filter such packet because it not only filters the incoming packets by matching them with firewall policy like stateless firewall but also checks the state of connection session. The connection state is described by TCP flag value which represents as the code bit in TCP header. Table 1 and table 2 show the binary code bit order and list of TCP flag respectively[10]. Those tables only present the flag types used in this paper.

Normally, the TCP packet is transmitted sequentially from SYN packet, SYN_ACK packet, ACK packet vice versa between source and destination. Sending the three packets establishes connection by three-way handshake. Finally, FIN_ACK packet is sent to the destination to close the connection formally. FIN flag cannot be seen in the flow because the packet with ACK flag is needed while sending FIN flag to the destination for informing both confirmation of the previous received packet and termination of the connection as in Figure 2.

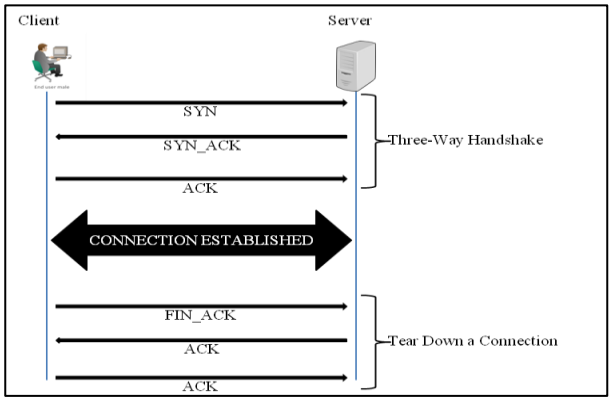

Figure 2. TCP Packet Sequence

Table 1. Binary code bit order for TCP flag

Code	1	2	3	4	5	6
Name	Urgent Pointer	ACK	PUSH	RST	SYN	FIN

Table 2. TCP flag

Flag	Code bit			Description
	Binary	Hexa	Dec	
SYN	000010	0x002	2	First segment of a new TCP connection
SYN_ACK	010010	0x012	18	Return acknowledgement for received SYN
ACK	010000	0x010	16	Acknowledge the successful receipt of packets
FIN_ACK	010001	0x011	17	Tear down the created virtual connection
RST	000100	0x004	4	Reject the request that is not intended for current connection

4. Proposed Stateful Firewall

The proposed stateful firewall application is implemented by adding connection state inspection into the packet forwarding operation of acl application. As state inspection cannot be done by forwarding devices such as Open Vswitch, all packets must be sent to the controller. Consequently, overloaded communication between control layer and data layer, and controller overhead problems are caused. To solve the consequence problems, the application sends only the essential packets to the controller by considering both the nature of connection tracking and SDN security.

In SDN environment, only the first packet of a flow is sent to the controller in order to get the flow rule. But it is also important to know the last packet or the packet contained FIN flag of TCP protocol for checking the teardown of connection session. Moreover, according to the connection tracking nature of TCP protocol, both initialization (SYN) and termination (FIN) of a connection session trigger from the source host. Thus, only packets come from source host have to be inspected carefully. By considering the combination of these factors, the stateful firewall application installs flow rule with group action in source switch and single output action in other switches.

Group action in this paper is the combination of output port action and To Controller action. The flow rule with this action sends packet to both destination and controller concurrently so that the controller can receive all packets including the last packet with FIN flag and terminate the connection by removing installed flow rules from the switches and deleting added state records in the state table immediately. As the flow rule with group action overloads the communication between control plane and data plane, it is only applied in essential source switch where the FIN packet comes from.

In addition, this application installs flow rules temporarily with timeout value like acl application but removes them as soon as the connection is terminated. When the connection is tearing down improperly, the controller will not receive the FIN packet and it will remove the flow rules after the timeout value expire. This fact is the main difference between acl application and this stateful firewall application. While acl application installs and removes flow rules temporarily without checking any state of the connection session, this application installs and removes them depending on the connection state.

From the SDN security point of view, attack packets can enter via the host connected switches and they must be protected tightly at those type of switches so that the attack cannot entered into the remaining network area. The host connected switches are source switch and destination switch. However, this application inspects the state of connection session at the source switch because it is safe enough to check one packet once at the entrance. Thus, firstly, the application in controller checks where the packet comes from. If it comes from the destination switch or intermediate switch, the application installs flow rule without checking any state of the connection session. Otherwise, connection state of the packet is checked by the application in order to make decision whether it installs flow rules or not for this packet in source switch.

If the packet does not pass through the controller as a TCP packet sequence, the application installs drop rule for this packet in the source switch. By this way, the firewall protects the possible attacks that breach the TCP protocol.

The connection state inspection process of proposed firewall application is shown in figure 3. For the packet with SYN flag or SYN_ACK flag comes from the destination switch or intermediate switches, the controller installs forwarding rule for it without checking the connection state. Thus, this figure does not consider the packet comes from those switches and it is especially for the inspection of incoming packet from source switch.

In this paper, default forwarding application in ONOS is used for forwarding packet. Hence, connection state inspection function is added into it. This application can be used to forward packet within the same network. Therefore, MAC address is used in state table construction instead of IP address. In addition, since only

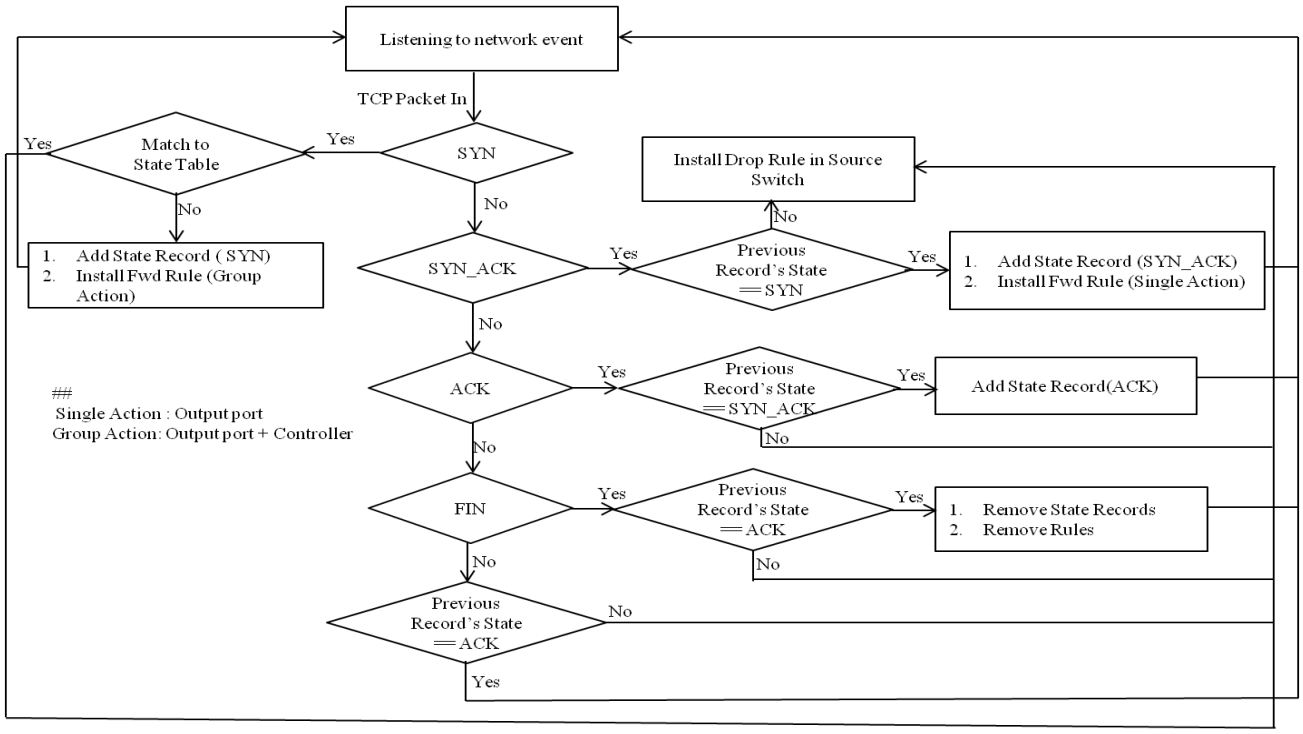

Figure 3. Flow chart of connection state inspection

the TCP protocol is stateful protocol, this paper presents the tracking of connection state for solely TCP when implementing stateful firewall.

5. Experimental Testbed

The testing is performed by using mininet emulator[11] with OpenFlow version 1.3[12] and ONOS controller. Both of them are running on Dell Desktop PC with Intel(R) Core(TM) i7-4790 CPU @ 3.60 GHz, 64 bits and 4 GB memory. The performance is measured on the linear topology of open virtual switch (OVS) with one host per switch. In order to compare the performance level, the acl application and stateful firewall application use the same linear topology.

6. Evaluation

As every security application has trade-off between security level and performance, this paper evaluates the two applications with two parts. Security level is measured by showing the filtering result of stateful firewall. Since stateful firewall tracks the network connection when filtering packets, the filtering results are shown by the log of filtered accessing information together with its state records. In addition, performance level is compared by taking the concurrent downloading time from web servers.

6.1 Filtering Result of Stateful Firewall

In this section, we experiment using a linear topology with three switches and three hosts in mininet and define

the acl rules set as shown in figure 4 and table 3 respectively. As this paper emphasizes solely on TCP protocol, the acl rules set is defined for only TCP. According to the rules set, all hosts are not allowed to access TCP protocol with port number 80 except those between host 1 and host 3.

Table 4 shows the flow rules of switch s1 installed by acl application together with its assistant, reactive forwarding application. According to the acl rules set in Table 3, acl application installs one drop rule and two allow rules with action To Controller.

Thus, the forwarding application takes responsible for installation of flow rules for allowed packets. It installs forwarding rule with output action to the destination port and group action to both controller and destination port concurrently in order to catch the connection termination packet with FIN flag from controller while sending packets of a flow to their destination especially in source switch.

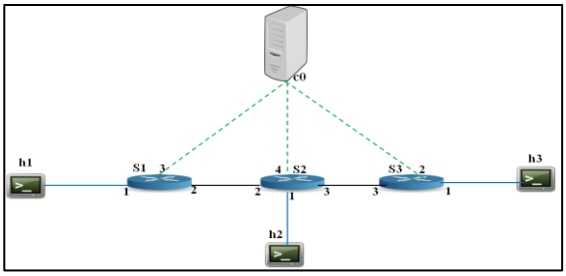

Figure 4. Linear Topology

Table 3. Acl rules set

Src Mac	Dst Mac	Proto	Dst Port	Action
10.0.0.1	10.0.0.3	TCP	80	Allow
10.0.0.3	10.0.0.1	TCP	80	Allow
10.0.0.0/24	10.0.0.0/24	TCP	80	Deny

Table 4. Flow rules in switch s1

Src Mac	Dst Mac	Proto	Dst Port	Action
10.0.0.1	10.0.0.3	TCP	80	Controller:65535
10.0.0.3	10.0.0.1	TCP	80	Controller:65535
10.0.0.1	10.0.0.3	TCP	80	group :1
10.0.0.3	10.0.0.1	TCP	80	output:1
10.0.0.0/24	10.0.0.0/24	TCP	80	drop

In figure 5, since only host 1 and host 3 are allowed to use TCP traffic with port 80 in defined acl rules set, host 1 can access web server in host 3 and cannot connect another web server in host 2.

Figure 6 shows the log of both connection state records for connection establishment and its termination from client host 1 to web server host 3. For each TCP flow, the application inspects the incoming packets according to the inspection process as shown in the flow chart of figure 3 and recognizes three records for SYN(2), SYN_ACK(18), ACK(16) after it has established a connection session. The fields included in each state record are source Mac, destination Mac, source port, destination port, and TCP flag. The SYN_ACK record is stored after swapping the source and destination of Mac and port in order to map the records easily. When the connection termination packet with FIN flag comes to the controller, the application removes the installed flow rules and deletes the state records of the respective connection.

The stateful firewall application drops an abnormal TCP flow as shown in figure 7. An abnormal TCP packet is sent from client host 1 to web server host 3 by using hping3[15]. TCP traffic with port 80 between host 3 and host 1 is allowed in acl rule, but the application drops the packet with SYN_ACK flag from host 1 to host 3 because of its abnormal state. The packet with SYN_ACK flag comes without its prior packet with SYN flag. Thus, the application assumes its abnormal packet and installs drop rule to block it as shown in figure 8.

6.2 Latency Result

Measurement of latency for TCP is performed on the increasing number of simultaneous connection (10 to 100) by setting up web servers depending on the number of TCP connection and one host accesses the servers at the same time. The web servers are created by using SimpleHTTPServer in mininet hosts and parallel download HTTP requests are sent from the client host with combination of xargs[13] and wget[14] command. This command uses web server url list while sending parallel downloading requests to web servers.

In order to get a thorough latency result, we conduct our experiment with two types of flow: long lived flow and short lived flow. We download a 277MB file from servers for long lived flow. For short lived flow, we only access the web page from web servers and the size of the web page is 2.4KB.

We can examine the latency for long lived flow from figure 9 that the download time differences between acl application and stateful firewall application vary from 0.2s to 7.87s for the number of simultaneous connection from ten to one hundred. The average times recorded from stateful firewall application are in the range of 0.26s to 37.93s while these of acl application are 0.06s to 34.1s.

Figure 5. Accessing information

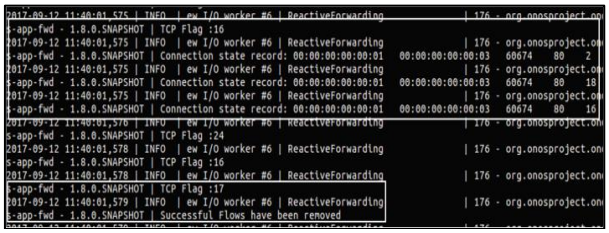

Figure 6. Log for connection state

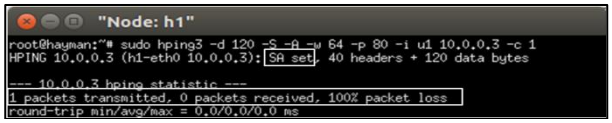

Figure 7. Abnormal TCP packet

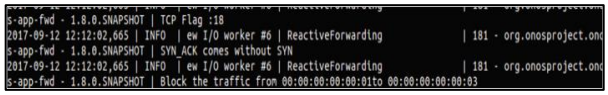

Figure 8. Log for blocking abnormal TCP packet

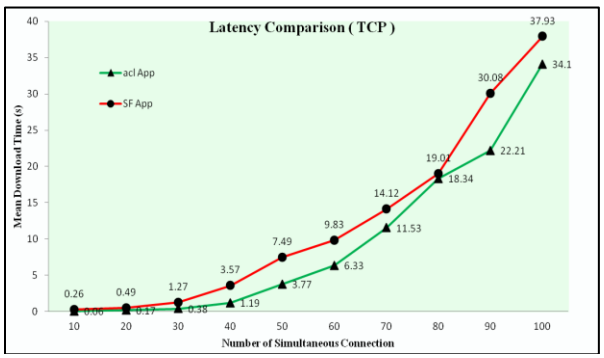

Figure 9. Latency result for long lived flows

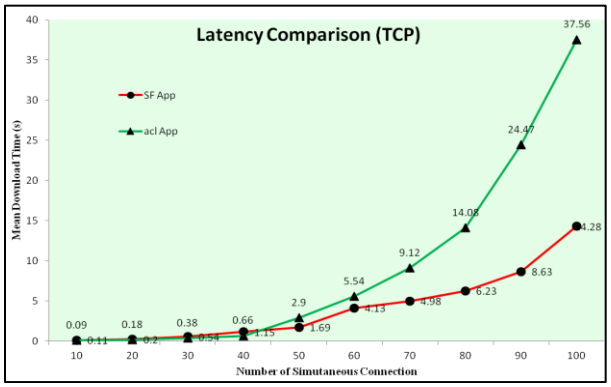

Figure 10. Latency result for short lived flows

In order to get the average delay time between these two applications, we calculate the time difference for each measurement step of their downloading time. Finally, we get the mean delay time, 2.6% by stateful firewall application.

Figure 10 shows the latency results for concurrent short lived flows. The acl application's average downloading time is more increase than stateful firewall's one when the number of simultaneous connections is above 40 because stateful firewall application uninstalls flow rules as soon as their respective connection is terminated while acl application is maintaining many flow rules without expiring their timeout value.

7. Conclusion

In order to keep higher security level in SDN, this paper proposes the stateful firewall application based on acl application in ONOS. Hence, the acl application becomes more secure due to the fact that it is enhanced from stateless firewall application into stateful firewall. However, the performance of stateful firewall is affected by added security operations. On the other hand, it is effected by removing flow rules immediately after terminating connection. By observing the latency comparisons in this paper, the average latency increased by stateful firewall application is 2.6% in long lived flow, though, the application increases the performance up to 8.5% when short lived flows pass through it. Therefore, we believe the proposed stateful firewall application is a proper SDN security application. We will implement and test the stateful firewall for not only TCP but also UDP and ICMP that can filter among different networks and improve the performance while preserving the security.

8. References

[1] ONOS controller, http://www.onosproject.org.

[2] Cornelius Diekmann and Florian Wohlfart, "Implementation and Performance Analysis of Firewall on Open vSwitch". Master's thesis. Technische Universitat Munchen, 2015.

[3] S. Morzhov, I. Alekseev, and M. Nikitinskiy, "Firewall application for Floodlight SDN controller", International Siberian Conference on Control and Communications (SIBCON), 2016- May-12, pp. 1-5.

[4] S. Vasudevan, "Firewall A New Approach to Slove Issues in Software Define Networking", 6th International Comference on Emerging trends in Engineering and Technology (ICETET'16), ISSN:2248-9622, pp.14-19.

[5] Salaheddine Zerkane, David Espes, Philippe Le Parc, Frederic Cuppen,. "Software Defined Networking Reactive Stateful Firewall", 31st International Conference on ICT Systems Security and Privacy Protection. Springer, Ghent, Belgium. 471, May 2016, pp.119-132.

[6] Andis Arins, Riga, and Lativa, "Firewall as a service in SDN OpenFlow network, 2015.

[7] Justin Gregory V. Pwna nd William Emmanuel Yu, "Development of a Distributed Firewall Using Software Defined Networking Technology", 2014.

[8] Thuy Vinh Tran, Heejune Ahn, "A Network Topology-aware Selectively Distributed Firewall Control in SDN", 2015.

[9] Thuy Vinh Tran, Heejune Ahn, "FlowTracker: A SDN Stateful Firewall Solution with Adaptive Connection Tracking and Minimized Controller Processing", 2016.

[10] Administrator, "TCP Flag Options - Section 4."Internet: https://www.firewall.cx/networking-topics/protocols/tcp/136-tcp-flag-options.htm.

[11] Mininet Network Emulator, http://mininet.org.

[12] OpenFlow Switch Specification, version 1.3.5, https://www.opennetworking.org/, 2015-March-26.

[13] Xargs command, Internet: http://man7.org/linux/man-pages/man1/xargs.1.html.

[14] GNU Wget 1.18 Manual, Internet: https://www.gnu.org/software/wget/manual/wget.html.

[15] Hping3 Security Tool, Internet: http://www.hping.org/hping3.html.