ADV MATH SCI JOURNAL
Advances in Mathematics: Scientific Journal **9** (2020), no.6, 4067–4075
ISSN: 1857-8365 (printed); 1857-8438 (electronic)
https://doi.org/10.37418/amsj.9.6.88 Spec Issiue on ICAML-2020

SECURITY EVALUATION FOR SDN BASED NETWORKS

HARSIMRATPAL KAUR[1], NAVDEEP SINGH, AND LAKHWINDER KAUR

ABSTRACT. SDN is ruling the network markets across the world and penetration testing frameworks used on traditional networks are not much effective against the SDN and its vulnerabilities. This paper encompasses around the DELTA security testing framework, which is one of the first for Software Defined Networks (SDN). DELTA is implemented on Linux Machines with division in mainly three different agents, which can be configured on either different machines or in the same machine by using Containerization. This can be used to find both known and unknown vulnerabilities in SDN. DELTA uncovered two unknown vulnerabilities in ONOS controller i.e. Flow Obstruction Attack and Host Tracking Neutralization, apart from that DELTA also made a successful MiTM attempt on ONOS. Along with DELTA, we have also used Kali Linux for performing DDoS attacks using Hping3 utility on ONOS controller.

1. INTRODUCTION

SDN is rising with new sub-areas like SD-WAN, SD-Security, SD-Access etc. and is taking it out of Data Center LANs to service provider/enterprise edges and now more and more vendors are using SDN in one or other way with many benefits that it deliver [5]. SDN is different from traditional network architectures as SDN decouples the control and data plane of the network [3,9]. Security Evaluation or penetration testing for SDN also differs in some concepts. Traditional Penetration Testing Framework do not play well with SDN and for better

[1]*corresponding author*

2010 *Mathematics Subject Classification.* 68M25.

Key words and phrases. SDN, DELTA, Man-in-the-Middle, Containers, LXC, Docker.

security evaluation, a framework dedicated for SDN should be used. Kali Linux is another security assessment OS which is a suite of various penetration testing tools. DELTA [1] is a penetration testing framework for SDN based networks having two main functions:

- It automatically expresses the attack cases against different SDN based elements over the different environments.
- It uncovers the security issues which are still unknown in SDN implementation with the help of a blackbox fuzzing method [4].

Delta is one of the most advanced frameworks used for SDN penetration testing. It also brings with it the fuzzing methods which play a big role in case of unknown SDN attacks. Various agents used with DELTA framework are explained below:

1.1. **Agent Manager.** This is the control-tower. This one is mainly installed directly over the operating system and it controls the agents implemented in the SDN based networks.

1.2. **Application Agent.** This one is an SDN based application running over the Northbound interface that performs attack functions and is controller dependent entity.

1.3. **Channel Agent.** It is implemented between the controller and OF based switch. The agent analyzes and updates the plain-text control messages. This agent works independently of the controller.

1.4. **Host Agent.** This agent behaves as if it is a valid host engaging in the SDN based network. Agent exhibits the attack where control plane by controller is attempted to be compromised. Architecture of DELTA framework is shown below in Figure 1.

2. SECURITY EVALUATION

One of the most popular features of using DELTA for security assessment of SDN is that it helps in uncovering new set of vulnerabilities. As we have mainly used ONOS as a SDN controller in our work, Table 1 below summarizes few vulnerabilities related with ONOS and mainly revolves around intra-controller control flow.

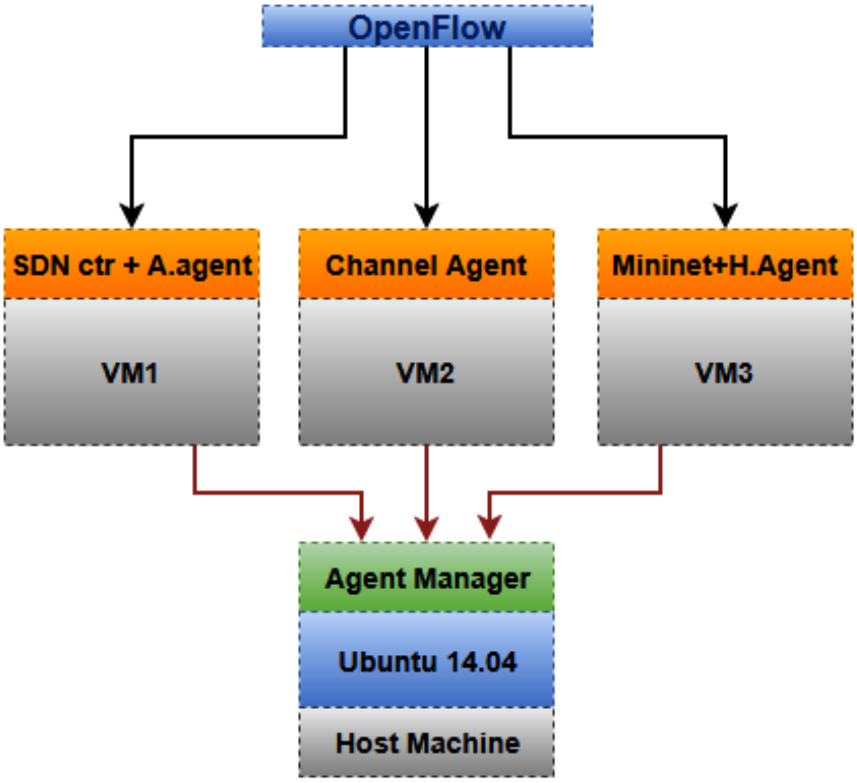

FIGURE 1. DELTA Framework for SDN Penetration Testing

TABLE 1. Unknown attacks uncovered by DELTA

Attack Name	Victim Controller/Flow
Flow-Rule Obstruction	ONOS/INTRA-CONTROLLER
Host Tracking Neutralization	ONOS/INTRA-CONTROLLER

Fuzz-modules of application agent plays a major role. There are various services offered by the controller and these offered services creates the targets. Two unknown attack cases are found with ONOS controller:

2.1. **Flow Obstruction Attack.** There are applications in ONOS [1] that are having configuration properties. Network Engineer can update the variable which was declared by an application. ONOS also offers ComponentConfigService, that monitors and updates the configuration properties for applications.

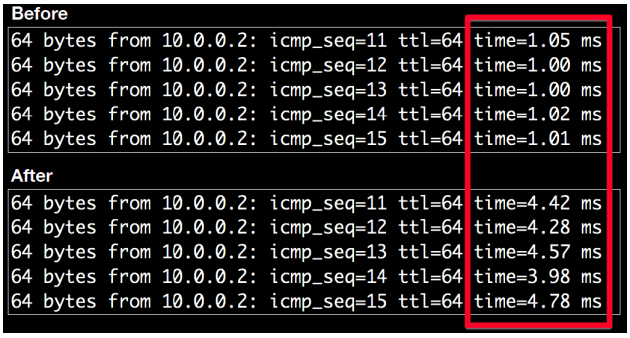

FIGURE 2. Increase in Latency due to Flow Obstruction Attack

This service also can update unnecessary configurations. DELTA uses this service, where a value fuzzer selects this service to randomize the input values which further randomizes some specific properties of Reactive Forwarding and that is when Agent Manager finds a degradation in performance of Switch. Fuzzing module randomizes the PacketOut property of reactive forwarding which turns true from false and no FLOW_MOD packets are sent to switch that automatically increases the latency. As no FLOW_MOD packets are sent to the switch, therefore on arrival of every new FLOW, a PACKET_IN message is generated to the ONOS. The rise in latency can be seen in Figure 2.

2.2. **Host Tracking Neutralization.** ONOS runs a HostLocationProvider service that works like Cisco Discovery Protocol (CDP) or Link Layer Discover Protocol (LLDP) that keeps monitor the end-host connected with the switches by maintaining host information like IP address, VLAN-ID, MAC Address and interface ID. So, if any end-host connects with the switch, it automatically updates the information regarding the end-host using its service. ComponentConfigService has the ability to change the properties of HostLocationProvider. Engineer can use DELTA for input fuzzing and input value randomization can be done by selecting the ComponentConfigService. As the value fuzzer is in use, ONOS tends to get the error messages from the switch. As the switch sends error messages to ONOS, it automatically matches one of the seven vulnerability detection standards. Agent Manager stores the information of fuzzing module randomizing the hostRemovalEnabled property from true to false which straightway stops the monitoring of end-hosts. For example, if any new end-host is connected, then switch does not detect its connection. Attack is analyzed using a packet capture

tool. Channel agent sniffs the error messages over the switch, which means the ONOS is not available because of invalid host. Even though the communication terminated, error messages are still sent to ONOS with the time interval of 10 seconds till ONOS shuts down.

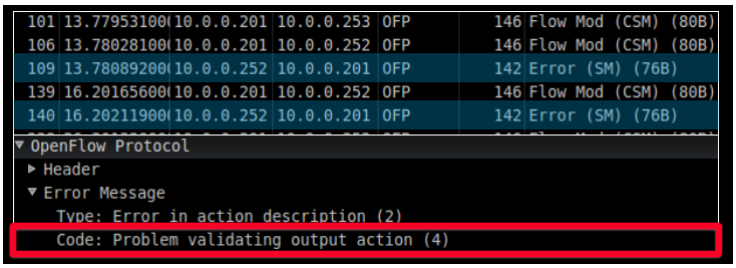

FIGURE 3. Host Tracking Error in ONOS

2.3. **Man-in-the-Middle Attack.** Using DELTA, variety of penetration testing or security evaluation can be done on ONOS. It can be used find Unknown and Known Attacks. DELTA uses fuzz testing to find newer SDN related vulnerabilities [2]. We have mainly worked on finding the known attacks and used the case of Man-in-the-middle attack [6]. As there are various number of switches which uses plain-text control messages because of performance limitation. So attacker can easily use a sniffer to get the messages and exploit them, which can become a very serious issue. In the channel-agent, we have created the rule-modification function which can update the true OpenFlow action to a mastermind action in FLOW_MOD, that is used to impose the flow rule to switch, but now in to a attacker application.

Next four steps explain the functions of the attack performed against the ONOS [8] controller.

(1) After DELTA initialization is done, all agents will be connected with the agent-manager. Then the agent-manager displays the available commands. In this, we have used Replaying known attack(s).
(2) The agent-manager then queries the attack code. Attack Code B-2-A is selected which denotes the MiTM.
(3) Agent-Manager then organizes the actions of every agent on the basis of the attack steps. Loading displays the attack process. Channel Agent starts to check FLOW_MOD messages, while the attack replica in the

FIGURE 4. Step for Man-in-the-Middle attack in DELTA

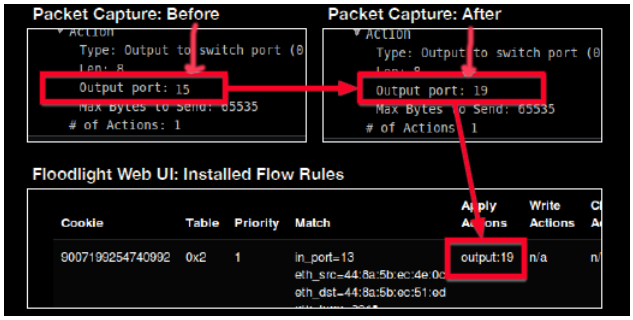

FIGURE 5. Success in MiTM attack using DELTA

channel agent updates the action field of the messages which are defined in the configuration file.

(4) The Agent-Manager fetches the output of the attack via agents and specify if the attack is a success or failure.

After applying the four steps above, we need to verify if the attack made was successful or not. Figure 5 shows that the captured packet before applying the attack has output port 15, but after applying MiTM, output port becomes 19.

2.4. **DDOS Attacks.** DDoS is another headache which is faced by network engineers from last two decades or so and it keeps on increasing with the time with recent ones on websites like Github saw 1.33 Tbps DDoS attack. DDoS can be HTTP, TCP, UDP, ICMP based. There are many tools available in the

```
root@kali-linux:~# hping3 -c 10000 -d 120 -S -w 64 -p 8181 --flood --rand-source 192.168.230.101
HPING 192.168.230.101 (eth0 192.168.230.101): S set, 40 headers + 120 data bytes
hping in flood mode, no replies will be shown
^C
--- 192.168.230.101 hping statistic ---
991511 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

FIGURE 6. Hping3 flood attack on ONOS controller

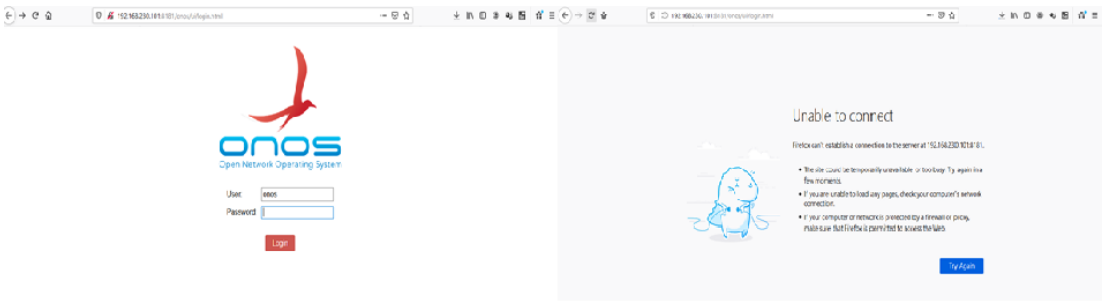

FIGURE 7. Before and after performing DDoS testing

market with which can perform DDoS attack on our network or server to test if our controller is secure in case some DDoS hits it. Tools like Hping3, LOIC, HOIC, Xerxes, HULK, Tor's Hammer, XOIC, Solarwinds Security Event Manager, Slowloris etc. are some of the most used and popular DDoS tools available. We have used Hping3, a package that comes preinstalled in Kali Linux for DDoS evaluation.

Above Hping3 command includes following parameters:

- c 10000 - Number of packets that have to be sent
- d 120 - packet size
- S - TCP SYN packet
- w 64 - TCP Window Size
- p 8181 - Destination Port
- flood - To send packets in burst mode
- rand-source - source address will be randomly generated
- 192.168.230.101 - ONOS ip address

The output in Figure 7 shows the impact of DDoS attack as access to ONOS machine is lost after performing DDoS on ONOS controller.

So, we have found four vulnerabilities in ONOS which are categorized in known and unknown as shown in Table 2.

TABLE 2. Vulnerabilities found in ONOS

Vulnerability in ONOS	Type
Distributed Denial of Service (DDoS)	Known
Man-in-the-Middle (MiTM)	Known
Flow Obstruction	Unknown
Host Tracking Neutralization	Unknown

3. CONCLUSION AND FUTURE SCOPE

Security Evaluation or Penetration Testing is important for any organization to find the vulnerabilities in their network and applications. Traditional Penetration Testing Frameworks do not integrate well with SDN based networks. DELTA Framework is one of the first and most popular penetration testing framework for SDN and can be used to find both known and unknown vulnerabilities. DELTA can be installed on linux either on a single machine using containers or different agents should be configured on different machines. Man-in-the-Middle Attack which is a known attack is successfully tested in the work between the controller and the host [7]. Using DELTA Framework is fruitful in order to secure the SDN network. Kali Linux is a penetration testing OS which is used extensively for security evaluations. DDoS attack is performed using Hping3 package and has successfully performed the DDoS on ONOS. Major work in this paper is on testing of known and unknown attacks on ONOS controller. DELTA can also be used to perform known and unknown attacks on other controllers like OpenDayLight, FloodLight etc. Along with this, automated scripts integrated with DELTA can also be used, on which we want to work in future to enhance the security options more than before.

REFERENCES

[1] S. LEE, C. YOON, S. SHIN S. S. HAYWARD: *DELTA: SDN SECURITY EVALUATION FRAMEWORK*, http://opensourcesdn.org/projects/project-delta-sdn-security-evaluation-framework.

[2] *SDN Security Vulnerabilities Genome Project*, http://sdnsecurity.org/project SDN-Security-Vulnerbility-attack-list.html

[3] A. SHALIMOV, D. ZUIKOV, D. ZIMARINA, V. PASHKOV, R. SMELIANSKY: *Advanced study of sdn/openflow controllers*, In Proceedings of the 9th Central & Eastern European

Software Engineering Conference in Russia, CEE-SECR '13, New York, NY, USA, 2013. ACM., 1–6.

[4] A. D. HOUSEHOLDER, J. M. FOOTE: *Probability-based parameter selection for black-box fuzz testing.*, Technical report, 2012, CERT Technical Report.

[5] https://www.opennetworking.org/

[6] J. D'ORSANEO, M. TUMMALA, J. MCEACHEN, B. MARTIN: *Analysis of Traffic Signals on an SDN for Detection and Classification of a Man-in-the-Middle Attack*, 2018 12th International Conference on Signal Processing and Communication Systems (ICSPCS), Cairns, Australia, (2018), 1–9.

A. K. ARAHUNASHI, S. NEETHU, H. V. RAVISH ARADHYA: *Performance Analysis of Various SDN Controllers in Mininet Emulator*, 2019 4th International Conference on Recent Trends on Electronics, Information, Communication & Technology (RTEICT), Bangalore, India, (2019), 752–756,

[7] MININET: *[online]* Available: http://mininet.org/

[8] ONOS INSTALLATION AND GUIDE: *[online]* Available: https://wiki.onosproject.org/display/ONOS/Installation+Guide

[9] J. PUSHPA, P. RAJ: *Topology-Based Analysis of Performance Evaluation of Centralized Vs. Distributed SDN Controller*, 19th Asia-Pacific Network Operations and Management Symposium (APNOMS), (2017), 1–8.

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING, PUNJABI UNIVERSITY PATIALA, PUNJAB, INDIA

Email address: chahalsimar1@gmail.com

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING, PUNJABI UNIVERSITY PATIALA, PUNJAB, INDIA

Email address: navdeepsony@gmail.com

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING, PUNJABI UNIVERSITY PATIALA, PUNJAB, INDIA

Email address: mahal2k8@gmail.com