

SECURITY THREATS ON DATA-CONTROL PLANE COMMUNICATION CHANNEL IN SDN

SUKHVEER KAUR¹, KRISHAN KUMAR, AND NAVEEN AGGARWAL

ABSTRACT. Software Defined Networking (SDN) is an emerging technology that gives flexibility, vendor neutrality, and centralized management by separating the network operating system (control plane) from the forwarding hardware (data plane). The OpenFlow is the standard protocol that is used for communication between the data and control plane. However, it is vulnerable to various security threats due to the lack of TLS adoption. In this study, we have performed the various attacks on data-control plane communication channel and analyzed the impacts on these attacks on the whole SDN network. To perform the experiment, we have created a virtual network lab using the GNS3 and VMware workstation. In the experiment, we observed the attacker can perform the MITM attack to modify the flow table entries for redirecting the traffic. Moreover, it can breakdown the control channel communication to disconnect the network from the controller machine. Finally, we have implemented TLS security to secure the data-control plane communication channel.

1. INTRODUCTION

SDN is a promising technology to make the network dynamic, flexible, and programmable to meet the requirements of modern data centers. It addresses the challenges of traditional network by separating the network control logic from the forwarding hardware. Moreover, the cost of SDN devices is significantly

¹*corresponding author*

2010 *Mathematics Subject Classification.* 68M25.

Key words and phrases. Software Defined Networking; OpenFlow; Flow Table modification; Data Plane; Control Plan; Security.

lower than the traditional network devices due to the use of open-source network operating system instead of using vendor-specific software [1]. The rapid adoption of SDN underscore the importance of security as its rising demands presents a lucrative target for attackers around the world. The architecture of SDN is divided into three layers: Data plane, Control plane, and Application Plane. Each of these plane along with its interfaces is susceptible to various security threats [2]. Existing studies work on the security of control plane [3, 4] and data plane [5, 6], but the security of data-control plane communication channel is still an open issue. The OpenFlow is the most popular protocol that is used for data-control plane communication. However, OpenFlow does not enforce the secure communication as the usage of TLS is recommended but not mandatory in the OpenFlow specifications [7]. The lack of TLS adoption will clear the path for attackers to perform various attacks such as eavesdropping, Man-in-the-Middle attack (MITM), etc.

Benton et al. [8] identifies the vulnerabilities that emerge from the separation of control and data plane and vulnerabilities within the OpenFlow Protocol. Antikainen et al. [9] discussed the possible attacks in SDN environment by compromising one or more OpenFlow enabled switches. Aseeri et al. [2] provided a remedy to deal with eavesdropping attack in the data plane by using the multiple routing paths to reduce the severity of information leakage. Some of the above works [8, 9] provided a theoretical study of SDN security threats, but none of these studies describe the practical details of how the attacker exploits the data-control plane communication channel to launch various types of attacks. Moreover, the existing studies [2–4] evaluated their experiments using Mininet tool [10] in which there is difficult to configure the daemons separately. In this experiment, we need the Virtual Machines (VM) to implement the daemons. The GNS3 tool is used to integrate the VMware VMs with the Open Flow switch that is implemented in the docker container. In this work, we answer the following research questions:

- Can we exploit the data-control plane communication channel to perform the flow table modification attack?
- What are the impacts of flow table modification attacks on the entire SDN network?

2. EXPERIMENT SETUP AND EVALUATION

To analyse the security threats in SDN, we have built a topology using GNS3 emulation tool. The topology consists of 5 virtual machines as shown in Figure 1. The Ubuntu16.04 LTS 64-bit operating system is installed on all the virtual machines. Out of 5, one machine acts as a Ryu controller [11]. We have installed and configured the DNS server on another virtual machine. We assumed that 2 virtual machines are compromised by the attacker that used the Ettercap tool to perform the MITM attack. One machine acts as a DNS client that is used to access the web page from the DNS server. All these virtual machines are running on VMware workstation and connected to Open vSwitch using the GNS3 emulation tool.

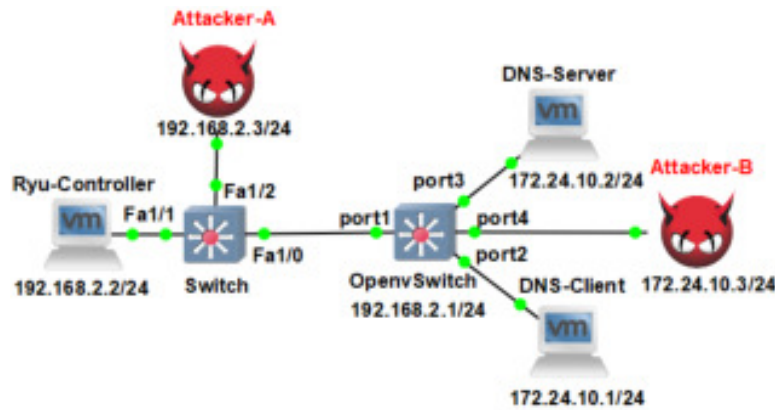


FIGURE 1. Network setup for attack scenario

3. ANALYSIS AND TESTING

In this section, we analyze attacks against SDN in a scenario where the TLS security is not implemented on data-control plane communication channel.

3.1. Breakdown the data-control plane communication. The data-control plane communication channel is one of the most crucial points in the SDN architecture. Our goal in this scenario is to observe what happened when this channel is breakdown by the adversary system. To achieve this, we performed the ARP spoofing on Attacker-A machine to sniff the traffic between the controller and the switch. Then, Attacker-A machine used the Ettercap filter to drop the

communication between the controller and the switch. According to the OpenFlow specification, when the communication channel is interrupted then, Open vSwitch either enter in "fail standalone mode" or "fail secure mode" mode. In the standalone mode, switch behaves like an ordinary switch. In the fail secure mode, switch does not setup the flow rules by its own when the connection got lost. Therefore, it starts dropping all traffic received from the connected hosts. In the experiments, we have changed the default fail mode of Open vSwitch to secure mode. As can be seen in the Figure 2, when the flow table of switch got empty, it starts dropping the packets after the breakdown of connection.

```

root@mininet-vm:~# ping -c 2 172.24.10.2
PING 172.24.10.2 (172.24.10.2) 56(84) bytes of data.
64 bytes from 172.24.10.2: icmp_seq=1 ttl=64 time=28.6 ms
64 bytes from 172.24.10.2: icmp_seq=2 ttl=64 time=2.85 ms

--- 172.24.10.2 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 2.856/15.774/28.693/12.919 ms
root@mininet-vm:~#
root@mininet-vm:~# ping -c 2 172.24.10.2
PING 172.24.10.2 (172.24.10.2) 56(84) bytes of data.

--- 172.24.10.2 ping statistics ---
2 packets transmitted, 0 received, 100% packet loss, time 1007ms

```

FIGURE 2. ICMP packets dropped after the breakdown of data-control plane communication channel

3.2. Flow Table Modification Attack. By modifying the flow table rules, attacker can perform other possible attacks: eavesdropping by traffic duplication and MITM attack as shown in Figure 3.

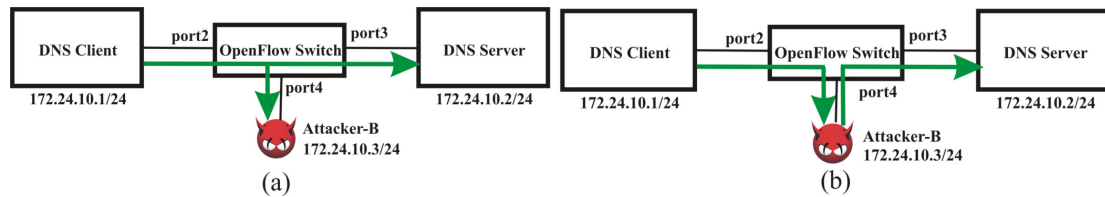
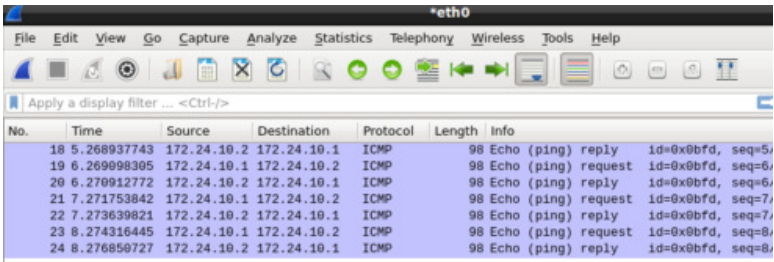


FIGURE 3. (a) Eavesdropping Attack (b) MITM Attack

- (i) Eavesdropping by traffic duplication: the basic principle of the eavesdropping attacks is to modify the flow table entries for duplicating the

web traffic to Attacker-B machine. To do this, Attacker-A machine captures the Flow_mod packet from the controller and modified it by inserting an extra OpenFlow action. This action duplicate the traffic to port4 (Attacker-B machine) that is transmitted between the DNS Server and the client. Now, the Attacker-B machine eavesdrop the communication when we try to ping to DNS server from the client machine as shown in Figure 4



No.	Time	Source	Destination	Protocol	Length	Info
18	5.268937743	172.24.10.2	172.24.10.1	ICMP	98	Echo (ping) reply id=0xbfd, seq=5
19	6.269098305	172.24.10.1	172.24.10.2	ICMP	98	Echo (ping) request id=0xbfd, seq=6
20	6.270912772	172.24.10.2	172.24.10.1	ICMP	98	Echo (ping) reply id=0xbfd, seq=6
21	7.271753842	172.24.10.1	172.24.10.2	ICMP	98	Echo (ping) request id=0xbfd, seq=7
22	7.273639821	172.24.10.2	172.24.10.1	ICMP	98	Echo (ping) reply id=0xbfd, seq=7
23	8.274316445	172.24.10.1	172.24.10.2	ICMP	98	Echo (ping) request id=0xbfd, seq=8
24	8.276850727	172.24.10.2	172.24.10.1	ICMP	98	Echo (ping) reply id=0xbfd, seq=8

FIGURE 4. Eavesdropping the traffic from Attacker-B machine

- (ii) Flow Table Modification for MITM Attack: in this scenario, Attack-A modifies the flow table entries not to duplicate the traffic but instead redirect the traffic to Attacker-B machine. After receiving the packets, Attacker-B machine performs the DNS spoofing to redirect the traffic that is going to the DNS server to his/her machine. To perform this attack, Attacker-A modifies the flow table entries with the ettercap to pass all the traffic going to the DNS server through Attacker-B machine. Next, Attacker-B machine edit the etter.dns file to redirect the traffic of www.example.com to his/her machine having IP address (172.24.10.3). In addition, Attacker-B activate the dns_spoof plug-in in ettercap for redirecting the web page as shown in Figure 5. After activating the dns_spoof plug-in, client gets the redirect web page instead of original web page (Figure 5).

4. TRANSPORT LAYER SECURITY

In section 3, we have shown how the adversary can exploits the data-control plane communication channel to downgrade the entire SDN networks. This is happening due to the lack of TLS adoption in the OpenFlow protocol. The Open

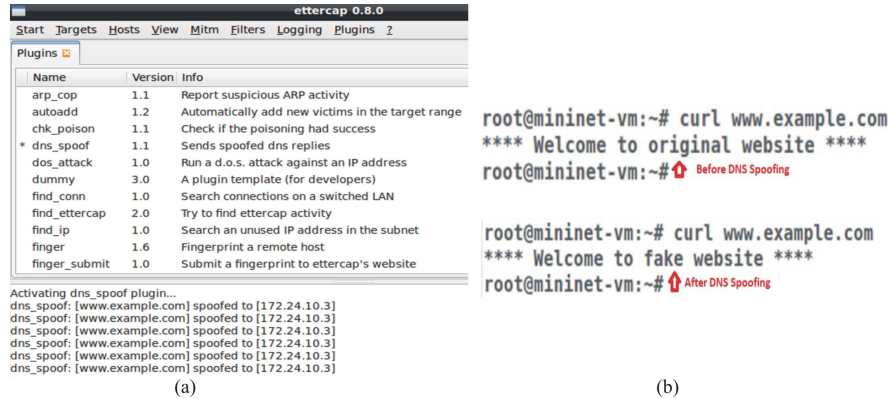


FIGURE 5. (a) Activating dns_spoof plug-in on Attacker-B machine (b) Accessing Web page before and after DNS Spoofing

Network Foundation (ONF) make the implementation of TLS optional on OpenFlow protocol as it gives the choice to network administrators to use any security protocol according to the type of programmable network [12]. However, if the security is not implemented properly, it can lead to many security threats. To prevent these types of attacks, we have implemented TLS security with the Public Key infrastructure (PKI). The initial PKI structure has created the root certificates for controller and switch certificate authority and private keys for signing these certificates. Moreover, it will create the public and private keys for controller and switch which are signed by the certificate authority as shown in Figure 6. The root certificate of controller certificate authority enables the Open vSwitch to authenticate the valid controllers. Similarly, the root certificate of switch certificate authority is used to authenticate the Open vSwitches. Finally, we have captured the traffic on the Wireshark from Attacker-A machine. As shown in the Figure 6, OpenFlow traffic that is transferred between the controller and the Open vSwitch is encrypted with TLS.

5. CONCLUSION

In this paper, we have done the practical security analysis of SDN by modifying the flow table entries to perform different types of attacks such as communication channel breakdown, Eavesdropping, and DNS spoofing. The adversary can exploit the vulnerabilities in SDN architecture such as lack of TLS adoption, separation of data and control plane, to perform these attacks. To prevent these

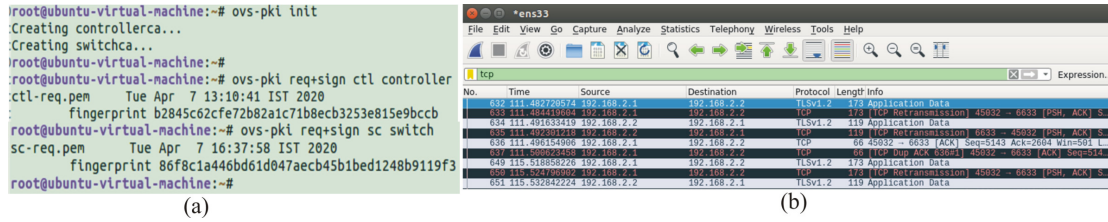


FIGURE 6. (a) Creating public and private keys for Controller and Switch using PKI (b) OpenFlow traffic encrypted with TLS

attacks, we have implemented the TLS security on data-control plane communication channel. In future, our aim is to implement the detection and mitigation solution to detect these attacks and drop all the communication from the malicious host.

REFERENCES

- [1] Y. JARRAYA, T. MADI, M. DEBBABI : *A survey and a layered taxonomy of software-defined networking*, IEEE Commun. Surv. Tutorials, **16**(4) (2014), 1955–1980.
- [2] A. ASEERI, N. NETJINDA, R. HEWETT : *Alleviating eavesdropping attacks in software-defined networking data plane*, ACM International Conference Proceeding Series, (2017), 1–8.
- [3] L. FICHERA, S. GALLUCCIO, C. GRANCAGNOLO, G. MORABITO, S. PALAZZO : *OP-ERETTA: An OPENflow-based REmedy to mitigate TCP SYNflood Attacks against web servers*, Comput. Networks, **92**(2015), 89–100.
- [4] P. KUMAR, M. TRIPATHI, A. NEHRA, M. CONTI, C. LAL : *SAFETY: Early Detection and Mitigation of TCP SYN Flood Utilizing Entropy in SDN*, IEEE Trans. Netw. Serv. Manag, **15**(4) (2018), 1545–1559.
- [5] B. YUAN, D. ZOU, S. YU, H. JIN, W. QIANG, J. SHEN : *Defending against flow table overloading attack in software-defined networks*, IEEE Trans. Serv. Comput, **12**(2) (2019), 231–246.
- [6] T. XU, D. GAO, P. DONG, C. H. FOH, H. ZHANG: *Mitigating the table-overflow attack in software-defined networking*, IEEE Trans. Netw. Serv. Manag, **14**(4) (2017), 1086–1097.
- [7] A. DANPING, M. POURZANDI, S. S. HAYWARD, H. SONG, M. WINANDY, Z. DACHENG: *Threat Analysis for the SDN Architecture*, IEEE Trans. Netw. Serv. Manag, (2016), 1–21.
- [8] K. BENTON, L. J. CAMP, C. SMALL: *OpenFlow Vulnerability Assessment Categories and Subject Descriptors*, Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking - HotSDN, (2013), 151–152.

- [9] M. S. M. ANTIKAINEN, T. AURA: *Spook in Your Network: Attacking an SDN with a Compromised OpenFlow Switch*, 19th Nordic Conference, NordSec, (2014), 229–244.
- [10] K. KAUR, J. SINGH, N. S. GHUMMAN: *Mininet as Software Defined Networking Testing Platform*, International Conference on Communication, Computing and Systems (ICCCS-2014), (2014), 3–6.
- [11] O. SALMAN, I. H. ELHAJJ, A. KAYSSI, A. CHEHAB: *SDN controllers: A comparative study*, Proc. 18th Mediterr. Electrotech. Conf. Intell. Effic. Technol. Serv. Citizen, MELECON 2016, (2016), 18–20.
- [12] B. AGBORUBERE, E. S. VELAZQUEZ: *OpenFlow communications and TLS security in software-defined networks*, Proc. - 2017 IEEE Int. Conf. Internet Things, IEEE Green Comput. Commun. IEEE Cyber, Phys. Soc. Comput. IEEE Smart Data, iThings-GreenCom-CPSCo-SmartData, (2017), 560–566.

DEPARTMENT OF INFORMATION TECHNOLOGY
UIET, PANJAB UNIVERSITY
CHANDIGARH
Email address: bhullarsukh96@gmail.com

DEPARTMENT OF INFORMATION TECHNOLOGY
UIET, PANJAB UNIVERSITY
CHANDIGARH
Email address: k.salujaiuet@gmail.com

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
UIET, PANJAB UNIVERSITY
CHANDIGARH
Email address: navagg@pu.ac.in