Detection and Mitigation of MITM Attack in Software Defined Networks

Saritakumar N[1], Anusuya K V[2], and Balasaraswathi B[3]
{nsk.ece@psgtech.ac.in[1], kva.ece@psgtech.ac.in[2], balasaraswathi1997@gmail.com[3]}

Assistant Professor, ECE, PSG College of Technology, Coimbatore,India[1,2], PG Scholar, ECE, PSG College of Technology, Coimbatore, India[3]

Abstract. Software Defined Network (SDN) is the networking architecture that segregates the activities of the control plane from the data plane. Man In The Middle (MITM) is a type of digital attack in a network where the attacker utilizes duplicated ARP messages by spoofing the attacker's MAC address with the authorized user's IP address. This paper sorts out the ARP spoofing, which is the suite of MITM attacks using IP-MAC address bindings. SDN is emulated using Mininet and the MITM attack over this network is done using arpspoof, which is the segment of the tool named Dsniff. For the evaluation of the proposed algorithm, various network parameters are compared and analyzed in both RYU and POX controllers. As a result, the proposed algorithm mitigates the MITM attack successfully by dropping the attacked packets.

Keywords: ARP Spoof, Dsniff, Mininet, MITM attack, SDN.

1 Introduction

In this digital world, wired and wireless technologies play the predominant role as they reducehumanefforts.Sincethenumberofusersincreasesduetoitsflexibility, the network becomes more complex. To deal with such complex networks, SDN is used.

SDN is the networking architecture that decouples the data plane from the control plane and thereby greatly minimizing the network complexity. The control plane is responsible for providing instructions about directing the traffic through the network and forwarding data traffic to the data plane in the path specified by the control plane. The Open Flow (OF) protocol [2] provides flow tables to direct the network traffic flow.

The infrastructure layer comprises network routers and switches to forward the data traffic. The controllers in the control plane control the network infrastructure by monitoring the topology, statistics, and state, etc. The application layer is developed by users interested in developing the applications by leveraging the network information such as topology, state, and statistics of the network.

The control layer and application layer are communicated through a North bound interface in which the applications notify the network requirements such as data, bandwidth, etc..so that the network can deliver those resources. The bottom layers such as the control layer and infrastructure layers having switches and network nodes are communicated through the Southbound interface used to identify the network and implement the data sent by the northbound interface.

ICCAP 2021, December 07-08, Chennai, India

DOI 10.4108/eai.7-12-2021.2314735

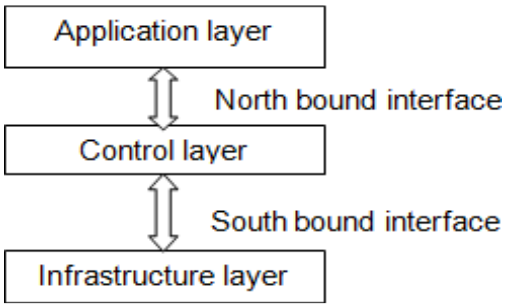

Fig.1. SDN Architecture

Since SDN [4] uses centralized architecture, security is a major concern. If the attacker fakes the controller as legitimate, then there has been the chance of data stolen from the end-user. It is necessary to mitigate those attackers so that users can use the network without any inconvenience.

2 Literature Review

The paper [3] explains different kinds of MITM attacks, such as ARP or TCP based MITM attacks. CMD obtains the forwarding rules of the infrastructure layer from the northbound API of the SDN controller, generatesa real-time Global Flow Table, and starts the real time MITM attack detection using generated Global Flow Table. The design of the MITM attack detection algorithm is based on the topology and connection of network flow. The MITM attack detection mentioned in this paper analyzes the contents of network packets, which is simple and efficient.

The paper [8] proposes two scenarios to describe the methodology of MITM attack in different controllers like OpenDayLight (ODL), Open Network Operating System (ONOS) and RYU. The simulation results indicate that the attackers can control the SDN controller easily and the communication between the control layer and infrastructure layer is not secure. This paper recommends the tool named snort which is based onan IPS intrusion prevention system and an ARP spoof pre-processor.

The paper [1] investigates the potential threats of MITM attacks on the OpenFlow control channel and proposes a lightweight countermeasure using Bloom filters. Bloom filter monitor system is implemented in OpenvSwitch and Floodlight controller and the result shows that the method is lightweight and efficient. As the result, the attack detection is achieved within a short time and there are negligible delays in traffic.

The paper [5] eliminates the problem of ARP poisoning attack which is the key factor for many other network attacks such as MITM, DOS and session hijacking. Two scenarios are introduced to resolve the ARP spoofing problem based on whether a network host will be assigned a dynamic or a static IP address such as SDN_DYN and SDN_STA. Since, the controller has recorded the MAC-IP paired map in all attempts, the ARP spoofing attack has failed.

3 Problem Statement

This paper focuses mainly on the Man in the Middle attack considered as one of the security threats in the network. In a MITM attack, the attacker spoofs between the conversation of sender and receiver and eavesdrops on the information sent by those users. By this, an attacker can easily steal the sensitive information of the end-user. This type of attack can be detected by monitoring the incoming packets based on their IP and MAC addresses. Since this attack is detected at an earlier stage, the attacker cannot gain control of stealing the data.

4 System Design

The system design comprises SDN network architecture with an SDN controller detached from the data plane. The Open V Switch is used in the infrastructure layer and its operation is based on the instructions provided by the controller in the control layer.

Mininet [7] is the network emulator used to create SDN virtual environment. It utilizes virtual hosts and switches in a single OS kernel. The virtual hosts and switches are used to create network topology. The hosts run on Linux based operating system and the switches support OpenFlow protocol.

The proposed methodology is implemented on both RYU and POX controllers. RYU is the open-source controller that uses OpenFlow or other protocols to provide instructions to the forwarding plane on the handling of traffic flows in the network. POX is the popular controller for developing network applications in SDN using Python programming language for OpenFlow devices.

arpspoof - the poisoning tool, a suite of Dsniff tools written in python with CLI is used to spoof the MAC address of legitimate hosts.

IMPLEMENTATION

Whenever the PACKET_IN message arrives at the destination from the sender, initially it passes the switches and is sent to the firewall of the controller, if the details of the destination host are not found on the switch. Thus the proposed algorithm is fused in the firewall of the controller.

A. AttackGeneration

As a part of the MITM attack, an ARP spoof [9] is generated using the Dsniff tool. By using this tool, an attacker would redirect packets from the target host intended for the client host by sending ARP replies. Thus the attacker sniffs the traffic on a switch.

Fig.2. Redirected packets from the target host

As a result, the attacker replicates the own MAC address to the target and hence the ARP table [10] gets updated. Subsequently, all the packets from the client receive the attacker and the attacker can redirect the packets or modify the packets or even steal the data easily.

B. AttackDetection

There are several ways to detect ARP spoofing in SDN such as

- Basedontrafficpatterns
- BasedonIP-MACaddressbindings
- Based onflowgraphs

Since the attacker places its own MAC address between target and client, the ARP table [6] gets updated with replicated MAC address. Every switch maintains an ARP (Address Resolution Protocol) table that stores the MAC and IP addresses of the network devices and is used to find the corresponding MAC addresses of the hosts. Thus by analyzing packets after ARP spoofing, the MAC address gets replicated between the hosts with the different IP addresses found in the ARP table.

Here, the detection is based on IP-MAC addresses bindings. Once the topology gets initialized, the controller dynamically allocates MAC addresses for every IP address in the network. The controller stores the created MAC address in the mac_to_port table. During the attack, the attacker duplicates its own MAC address and spoofs the arrived packets. But the controller is unaware that whether the arrived packets have legitimate MAC addresses or duplicated MAC addresses.

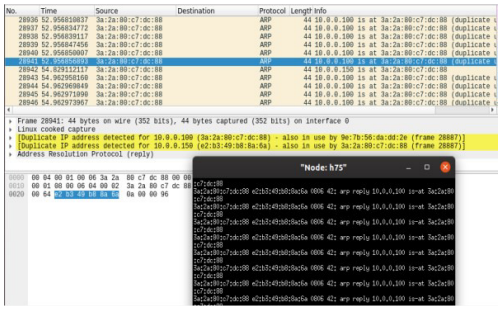

Fig.3. Attack detection

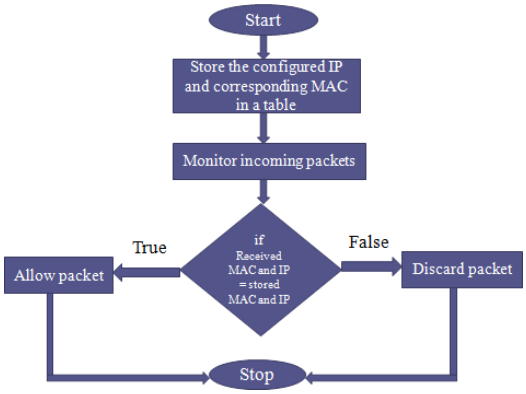

Fig.4. Proposed methodology

C. AttackMitigation

As a part of the MITM attack, an ARP spoof is generated using the Dsniff tool. Attacker redirects packets from a target host intended for client host by casting ARP replies. Thus the attacker replicates the attacker's MAC address to the target so that the ARP table gets updated. Subsequently, all the packets from the client are received by the attacker.

Due to arpspoof, the ARP table gets updated with reproduced MAC address. ARP table is maintained by all switches which are used to store the IP and MAC addresses of the network devices. By this process, the attacker can redirect, modify or steal the data from the client.

The OpenFlow switch maintains the flow table with four entries related to ARP protocol, Ethernet source address, Ethernet destination address, IP source address, an IP destination address. But the controller forwards packets only based on the MAC address.

IP addresses of hosts are obtained and configured IP addresses and corresponding MAC addresses are stored in the mac_to_ip table. If the arrived packet is ARP, then the source IP and MAC addresses are obtained. The incoming MAC address is compared with the MAC address that is stored in the table. If the MAC address entry is found in thetable, then the corresponding IP address is analyzed. If the corresponding IP address is unmatched with the incoming IP address, then the attack is detected.

Algorithm1 MITMmitigation

Input:*Arrivedpackets*
Output:*Droppingattackedpacket*

1:**procedure**FUNCTION
2: Monitorarrivedpackets
3: **for**(packet=1;packet<=i;packet++)**do**
4: StoreIPand MACin mac_to_iptable
5: **if**(incomingMAC= storedMAC)**then**
6: Allowpacket
7: **else:**

8: Packetdrop
9: **endif**
10: **endfor**
11:**endprocedure**

---

5 Results And Discussions

The proposed methodology is carried out on the POX and RYU controllers with a network emulator, Mininet. POX and RYU controllers are based on python programming language and run on OpenFlow protocol version v1.0.

A. Network topology

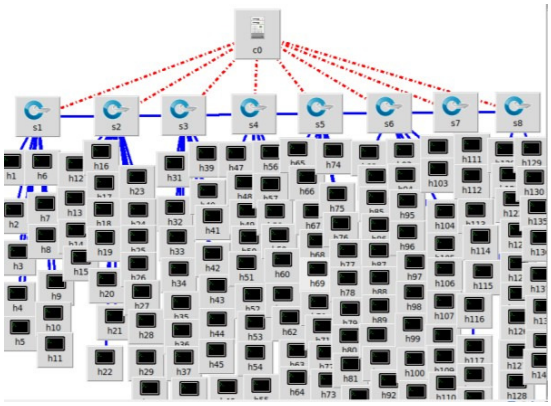

Fig.5. Network topology

B. Network Setup

TABLE I. SYSTEM SPECIFICATION

Parameters	**Values**
Processor	Intel(R)Core(TM)i3-5005U CPU@2.00GHz
RAM	4GB
Diskspace	255GB
OperatingSystem	Windows10
ApplicationSoftware	VirtualBox MininetVersion2.2.1

Fig.5 shows a tree network topology created using Mininet with 200 hosts, 8 switches, and a remote controller. The remote controller can be either an RYU or POX controller.

C. Experimental Results

The controller compares the received IP address and MAC address with the configured IP and MAC address.

Fig.6. Dropping the arpspoofed packets

In Fig.6, when compared to the configured and received IP addresses, there is a mismatch in those IP addresses and thus the ARP spoofing gets detected and thus the packet gets dropped.

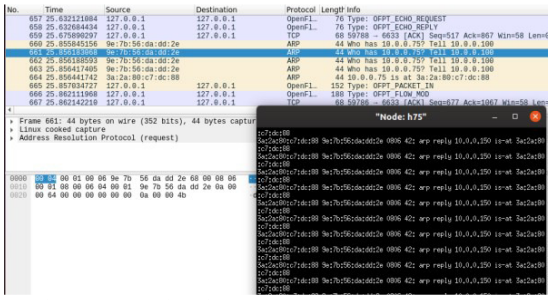

Fig.7. Analysis of dropped packets in Wireshark

Fig.7 shows that the filtered-out attacker's spoofed the MAC address from the controller and directs only the legitimate MAC and IP addresses.

D. CPUUtilization

Performance evaluation is done to ensure the mitigation algorithm under MITM attack under various conditions. The CPU utilization is observed for switches with two controllers named POX and RYU before and after implementing the mitigation algorithm. The CPU utilization metric is based on average CPU usage over a certain period.

TABLE II.CPU UTILIZATION FOR POX

Number of hosts	**Before mitigation (%)**	**After mitigation (%)**
50	70.8	61.2
100	80.5	76.6
150	87.2	78
200	88.4	82.6

Table II infers the CPU utilization for the POX controller before and after attack mitigation. The result analysis shows that CPU utilization has reduced after the prevention mechanism.

Fig 8 represents the graphical CPU Utilization for before and after mitigation of MITM attack. During the attack, as the number of hosts increases the CPU utilization increases. After implementing the mitigation algorithm, CPU utilization gets reduced up to 10% in POX.

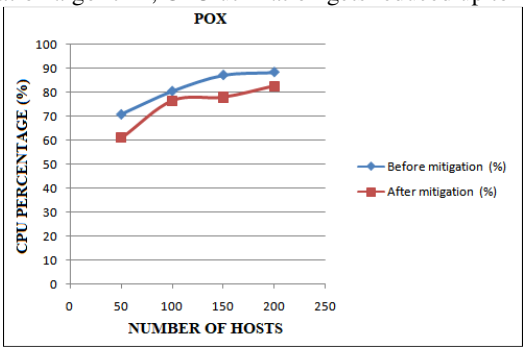

Fig.8. CPU utilization for POX

Table III infers that the CPU utilization for the RYU controller before and after attack mitigation. The result analysis shows that CPU utilization has reduced after the prevention mechanism.

Fig9 represents the graphical CPU utilization before and after mitigation of the MITM attack. The plot represents the number of hosts versus the CPU resource used respectively during the MITM attack. As the number of hosts increases the CPU utilization increases. The CPU utilization gets decreased up to 20% during the mitigation in the RYU controller.

TABLE III. CPU UTILIZATION FOR RYU

Numberofhosts	**Beforemitigation (%)**	**Aftermitigation (%)**
50	71.4	47.4
100	74.1	59.2
150	76.3	64
200	77	70.9

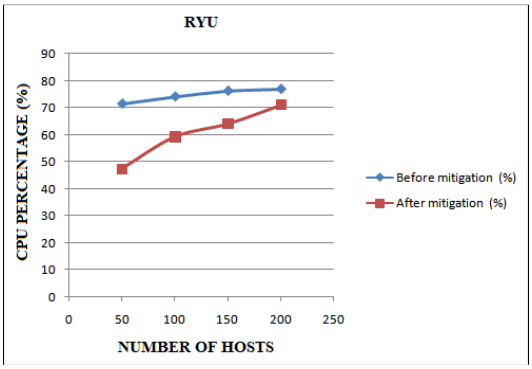

Fig.9. CPU utilization for RYU

E. ExecutionTime

It is necessary to calculate the execution time to evaluate the efficiency of the algorithm. The calculation of execution time is done in both POX and RYU controllers.

TABLE IV. Execution Time For Controllers

NUMBEROFHOSTS	POX(us)	RYU(us)
50	1.6	3.6
100	1.8	1.4
150	3	1.6
200	9	3.6

Fig.10 depicts the execution time required for the POX and RYU Controllers. The plot represents the number of hosts versus the time required for the execution of the mitigation algorithm.

As the number of hosts increases, execution time also gets increased. For the implemented algorithm, the execution time utilized by the RYU controller is only 5% of that found using the POX controller.

6 Conclusion And Future Work

With the detection algorithm implanted into the 2 controllers, two different behaviors are observed. This solution is not only efficient in detection, but it also has minimal code addition to the controller program and does not increase CPU load in either normal or attack conditions. POX controller takes more time to process the packets once an attack occurs but consumes less CPU usage than that of the RYU controller.

References

[1] Cheng Li, Zhengrui Qin, Ed Novak, Qun Li,Member, IEEE, "Securing SDN InfrastructureofIoT-Fog Networks from MitM Attacks,"IEEEInternetofThingsJournal,2017.

[2] XIAJing,CAIZhiping,HUGang,andXUMing, "An Active Defense Solution for ARP Spoofing in OpenFlow Network,"Chinese Journal of Electronics,Vol.28, No.1, Jan.2019.

[3] KaiZhangand XiaofengQiu,"CMD:AConvincing Mechanism for MITMDetectioninSDN,"IEEE International Conference on Consumer Electronics (ICCE), IEEE,2018.

[4] Tri-Hai Nguyen and Myungsik Yoo, "A hybrid prevention method for eave dropping attack bylink spoofing insoftware-defined Internet of Things controllers", International Journal of Distributed Sensor Networks, Vol13(II),2017.

[5] MohammadZ.Masoud,YousefJaradat,andIsmaelJannoud,"OnPreventingArpPoisoning AttackUtilizing SoftwareDefinedNetwork(SDN)Paradigm",IEEEJordanConference on Applied Electrical EngineeringandComputingTechnologies(AEECT),2015.

[6] TalalAlharbi,DarioDurando,FarzanehPakzad, "Securing ARP in Software DefinedNetworks",IEEE41stConferenceonLocalComputerNetworks,2016.

[7] ChaitraN.ShivayogimathandN.V.UmaReddy, "Performance Analysis of a SoftwareDefinedNetworkUsingMininet,"Springer Artificial Intelligence and Evolutionary Computations in Engineering Systems pp391-398,Feb.2016.

[8] AnassSebbar,MohammedBoulmalfandMohamed Dafir, “Detection MITM Attack inMulti-SDNController”,IEEE,2018,pp.583-587

[9] ZeynabSasan,MajidSalehi,“SDN-basedDefendingAgainstARPPoisoningAttack”,JournalofAdvancesinComputerResearch,Vol.8, No.2, May2017.

[10] Ahmed M.AbdelSalam, Ashrad B. El-Sisi andVamshi Reddy.K, “Mitigating ARP SpoofingAttacksinSoftware-DefinedNetworks”,ICCTA,2015.

[11] C. Amuthadevi, D. S. Vijayan, Varatharajan Ramachandran, “Development of air quality monitoring (AQM) models using different machine learning approaches”, Journal of Ambient Intelligence and Humanized Computing, https://doi.org/10.1007/s12652-020-02724-2