# Detection and Mitigation of ARP Poisoning Attack in Software Defined Network

Saritakumar N[1], Anusuya K V[2], Ajitha S[3]
{nsk.ece@psgtech.ac.in[1], kva.ece@psgtech.ac.in[2], ajithaanjali97@gmail.com[3]}

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

**Abstract.** The Software Defined Networking (SDN) is an emerging network paradigm that separates the control plane from thedataplane.SDN facilitates network management through simple, scalable, and programmable architecture. However, the centralized control in SDN architecture is vulnerable to attacks. In this work, a mechanism is proposed toimprove and eliminate the problem of ARP poisoning attacks. The two most prominent limitations of ARPattacks are - unauthenticated and stateless nature of ARP. ARP poisoning launches higher-level attacks likeMan in the middle attack, Denial of Service, and session hijacking. Hence, the proposed algorithm is to resolve the problemofARPspoofing. It is implemented as an extension module in POX and RYU Controllers and is evaluated under different attack scenarios. Mininetis used for SDN network emulation. ARP poisoning attackoverthenetwork isinitiatedusing the Dsnifftool.

**Keywords:** ARP POX , RYU , SDN.

## 1 Introduction

Software-Defined Networking is a novel network architecture currently considered as a best practice for network function virtualization. This new networking design advocates the separation of the data plane from the control plane, which would accelerate the development and deployment of new network applications. The data plane represents the data that is being forwarded through the network. The control plane is responsible for deciding how and where to send the data. The centralized controller in the control plane provides enough information for the operator to optimize network utilities and improves network performance. Traditional networks depend on physical components like switches and routers to make connections and transfer information among users. However, a software-defined network requires a user to control the placement of resources inside the network through the control plane. Instead of interacting with physical infrastructure, the user interacts with software to provide new devices. Open Flow (OF) protocol which is the open standard for SDN architecture. The communication interface is provided between the controller and switches. The controller manages a set of flow table in an OF switch. The packets arriving the OF switch port is compared with the entries of flow table. Further actions are recommended to the controller for the mismatch of rules against the incoming packets. Also, the installation of new rules is suggested to the controller as a corrective action. This mechanism leads to serious issues such as scalability and security[8].

ICCAP 2021, December 07-08, Chennai, India

DOI 10.4108/eai.7-12-2021.2314502

The SDN architecture has three distinct planes. The bottom plane is the data plane, which consists of hardware such as network switches. The Control plane lies above the data plane. This consists of a centralized controller that computes and communicates policies to individual devices that are related to network. The centralized controller might be a simple server machine that is attached tothe network running on controller software. Application plane resides above the control plane. This plane consists of individual applications that could be monitoring the utilities related to network utilities, applications related to Voice over IP applications that demand low delay, latency, etc. The data forwarding plane comprises of SDN switches that are connected by the wired or wireless network. Every switch is a device that does the forwarding of network packets and maintains a forwarding table termed as Flow table, which consists of rules that are used for making forwarding decisions. Forwarding plane takes care of forwarding, dropping, and changing packets. The control layer manages and controls the whole network. In general, the SDN Controller is deployed as a separate physical device that runs with specific software. Through a standard south-bound API, the Controller communicate swith the switch and has overall view of the entire network at the data forwarding layer. Through north-bound API, the application layer communicates with the control layer.

Address Resolution Protocol (ARP) is a layer2 (Data link layer) protocol used to find the MAC address of the known IP address. ARP cache is a table maintained by ARP that contains an IP address with its associated MAC address andtype. If MAC address is learned dynamically then the type will be dynamic and if MAC address is added manually then the type will be static. If the cache is hit, the corresponding destination MAC address is obtained. ARP request is a broad cast message generated by the source to find the target MAC address if the ARP is not resolved initially. ARP reply is a unicast message from target to the sender device containing the destination MAC address.

## 2 Literature Review

The paper [15] assesses the existing security status and other states that can be done to improve the security through SDN. In this paper, an SDN-based security related approach has been considered with sophisticated analysis on security aspects which shows how the security of the network can be improved with SDN. The paper [16] explains the performance of SDN using Mininet emulator, where the decisions on routing are done by a centralized controller and the forwarding is done by a switch. The paper [17] explained the ARP cache poisoning attack which is mostly seen in LAN networks, that has no effective solution to solve in traditional networks but SDN provides solution to solve this problem without any modifications in the network.

The Paper[18] reviews on the principle outcomes of a literature overview on the effects and challenges of SDN. With SDN, the admin have the flexibility to abstract the networking infrastructure for network services. It suggests that most of the SDN papers address the implementation of it as a challenge. Paper discusses the security challenges, issues arising with SDN and the high demand from the end-host combined with the trepidation of changing conventional networks. The significance of SDN is discussed by the unique features of SDN like decoupling software from the hardware and the overall view of the entire network architecture. SDN makes better decisions than IoT and Big data, on data and security. Furthermore SDN affects the network management, including the changes that have to be

made in the deployment of policies, network maintenance and programmability. This paper also reviews cost efficiency and cost reduction.

The Paper[19] reviews Software Defined Network that introduces opportunities and provides the potential to successfully overcome the challenges. This paper presented a novel SDN-based infrastructure for networking. The edge controllers needed to work in a newly distributed environment to ensure each domain's independence in case of failure. The paper also proposed architecture to include sensors based network in SDN. The paper also discussed about the interconnection of multiple domains and described how to improve the protection of each domain. It also suggested the ways to distribute the rules related to security in order that the security of one domain is not negotiated. Based on a grid of security model, they improved the exchange of protection policies and implementation between SDN control domains. Finally, the paper discussed about IoT (Internet of Things) and proposed a new architecture for it which is protected and distributed.

## 3 Problem Statement

In this section, the background of ARP protocol and ARP Poisoning attacks [2] are discussed.

*A. Background on ARP Poisoning attack*

In a normal ARP process, when a source wants to send a packet to the destination device, the source ARP cache is checked if the ARP is resolved or not. If the ARP table is not resolved, it puts the packet on hold and generates an ARP request. If the ARP is already resolved then the packet will be delivered to the destination host. The ARP request is broadcast all over the network to find out the device having a destination IP address. If the target is present in the same network then ARP finds out the target MAC address but if it is present in a different network then ARP finds out the default gateway MAC address. When the device having he destination IP address receives the ARP request, it updates its own ARP cache. The destination host machine generates an ARP reply containing its own MAC address. Now, the device having the source IP address receives the ARP reply and updates its ARP cache. Since both source and destination, IP address and MAC address are available; therefore, the packet is delivered to the destination host.

ARP protocol contains many issues like authentication and the acceptance of a reply even though the request has not been sent by the host. The ARP poisoning attack is introduced by these vulnerabilities [3].ARP Poisoning [4] may be asort of attack during which an invader sends falsified ARP messages over a local area network. This leads to the linking of an invader's MAC address with the IP address of a genuine computer or server on the network. Once the hacker's MAC is hooked up to a real IP, the attacker starts receiving any information that's sent for that IP. ARP spoofing can allow malicious nodes to interrupt, regulate or perhaps prevent information-in-transit. ARP spoofing attack can arise on LAN that make use of the ARP.

The effects of ARP Poisoning attacks can have serious implications for enterprises. In the most essential application, ARP Poisoning attacks are used to take sensitive data. Other than this, these attacks are often used to assist other attacks [11] like:

- DoS attacks [1]:This attack [5][6][14] often influence Address Resolution Protocol poisoning to link many IP addresses with only one target's MAC address. As a result, traffic that is supposed to be sent for many IP addresses will be redirected to

the destination's MAC address, creating congestion in the destination with the traffic.

- Session hijacking: Session hijacking attacks can use ARP poisoning to get the session IDs, granting attacker's access to private systems and data.
- MITM attacks: This attack [10] depend on ARP poisoning to intercept and modify traffic between victims.

In this paper, an ew algorithm that uses an SDN controller [7][13] is proposed. This proposed algorithm can prevent ARP poisoning in LANs without any change of the traditional ARP protocol. This proposed algorithm operates to link the controller in the decision-making of either forwarding or dropping the ARP packets.

*B. Tools used*

Mininet [20] is a network emulator which discovers a network with virtual switches, virtual hosts, controllers, and links. Mininet is used to create a virtual network, switch, host, and software code on a personal computer. Mininet allows users to quickly create, interact with a software-defined network prototype to emulate a network topology that uses Open Flows witches.

Dsniff is a tool to generate an attack on the network by running a python script or by using the Dsniff command line. Network sniffing is an important tool for monitoring network activity. The tool named ARP spoof, which is a part of the suite Dsniff is used for ARP poisoning attacks, redirecting the flow of packets and making it flow through the device. The suite Dsniff consists of many programs used to launch ARP Poisoning attacks.

Wireshark [21]is a tool is used to analyze the traffic generated by the network. It is also used as a packet sniffer tool. It captures the data traffic on the local network and stores that data for offline analysis. Wireshark captures traffic from Bluetooth, Ethernet, IEEE.802.11, Frame Relay, Token Ring connections, and many more. Wireshark intercepts traffic and converts that binary traffic into a human-readable format. This makes it easy to identify the type of traffic is crossing the network.

## 4 Proposed Algorithm

To address the problem discussed in Section II, an effective algorithm as an extension for securing the SDN control plane is proposed. The main objective of this algorithm is to detect and mitigate ARP Poisoning attacks in SDN. In this section, the overall architecture of the proposed counter measure is illustrated.

Attack detection is done by considering two details:

- Based on IP-MAC address binding.
- The number of times ARP reply is repeated.

The primary step is to identify the common nature of the ARP reply packets. With the details obtained from the controller, it can identify similar packets getting received from the attacker. The tolerance to such identical packets being receive dis-determined with the threshold value.

The detection is based on ARP packet and IP-MAC addresses bindings and count value. Once the topology gets initialized, the controller starts dynamic allocations of the MAC address for every IP address in the network. The controller stores the created MAC address in the MAC_to_PORT table. During an attack, the attacker duplicates its own MAC address and

spoofs arrived packets. But the controller is unaware of the arrived packets having legitimate MAC addresses or duplicated MAC addresses.

Dsniff is used to generate an ARP poisoning attack.The attacker launches the MAC address between the target and client and to make the connection, it will send the ARP reply packets continuously so that the controller assumes that the connection is legible. With the help of an arp spoof, the attacker will send 1000 packets/second and since the switches receive numerous duplicate packets continuously, a threshold is fixed to control the attack. Hence, to detect thepackets in the millisecond range, the threshold value is fixed as 100.Ifthepacketcount

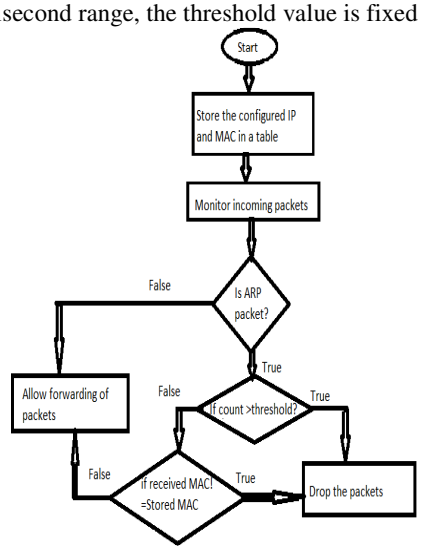


Fig. 1. Process Flow

Increases the threshold, the controller drops the packets and If the MAC address in the packet and the record on the list were dissimilar, the module would issue a warning of an ARP poisoning attack.

This module can also issue rules to switches so that incase of observing this MAC address all the arrived packets are sent to the controller. The controller takes the action and stops the attacker's performance.

## Results And Discussion

In this section, a complete simulation study is carried out and the results are examined to estimate the performance of the mitigation algorithm. In normal SDN, there is no defense mechanism against ARP Poisoning attacks, and how security [12] is improved in SDN by using this mitigation algorithm is shown.

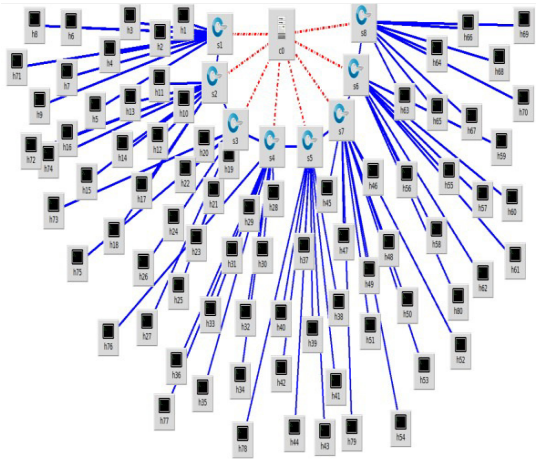

Fig. 2. Network Topology

Using Mininet, a linear topology with 80 hosts and 8 switches is created. The experiment is done on a Delllaptop with a Quad-Core processor with 2 GHz, and 4GB of RAM. The operating system is windows10 and Mininet version2.2.2 runs on the VirtualBox.

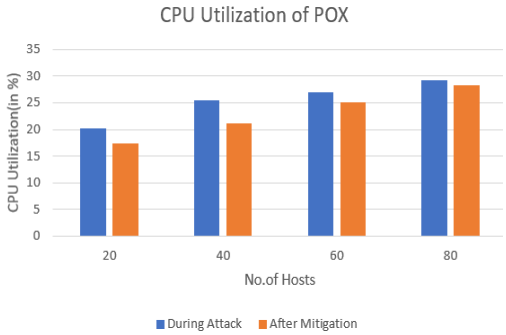


Fig. 3. CPU Utilization of POX

From Fig3,the CPU utilization of the POX controller during an attack and after mitigation is compared and from the graph. It is inferred that after mitigation the CPU utilization is reduced after the prevention mechanism.

From Fig 4, the CPU utilization of the RYU Controller during an attack and after mitigation is compared. From the graph, it is inferred that after mitigation the CPU utilization is reducedafter the preventionmechanism.

Table II. CPU Utilization OF RYU

| No. of hosts | CPU Utilization (in %) | |
|---|---|---|
| | During Attack | After Mitigation |
| 20 | 17.3 | 6.3 |
| 40 | 22 | 16.9 |

| | | |
|---|---|---|
| 60 | 25 | 20.1 |
| 80 | 28 | 24.5 |

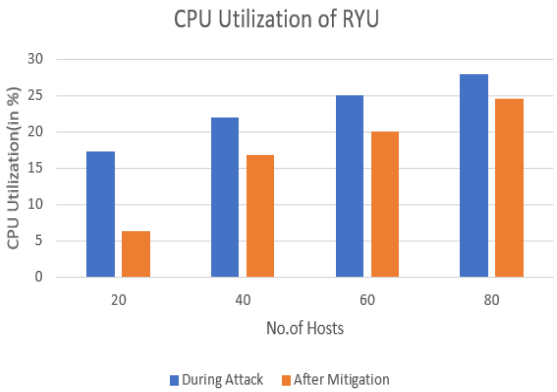


Fig. 4. CPU utilization of Ryu

Fig 5 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 9% less compared with the POX controller.

TABLE I. CPU UTILIZATION OF POX

| No. of hosts | CPU Utilization (in %) | |
|---|---|---|
| | During Attack | After Mitigation |
| 20 | 20.2 | 17.3 |
| 40 | 25.5 | 21.2 |
| 60 | 27 | 25 |
| 80 | 29.3 | 28.3 |

TABLE III.EXECUTIONTIMEFORPOXANDRYU

| **No. of hosts** | **Execution time (inμs)** | |
|---|---|---|
| | ***POX*** | ***RYU*** |
| 20 | 0.0023 | 0.0011 |
| 40 | 0.0032 | 0.0028 |
| 60 | 0.0053 | 0.0042 |
| 80 | 0.0082 | 0.0065 |

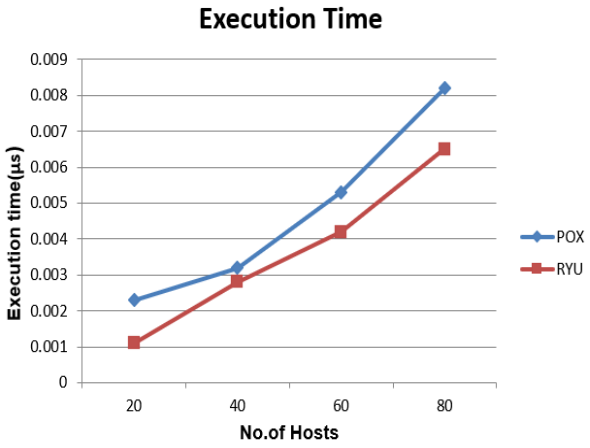


Fig. 5. Execution Time for POX and RYU

## Conclusion

The ARP poisoning attack is detected based on the ARP packet count value and IP-MAC address matching with the configured IP-MAC address. By the mitigation algorithm, the net work is secured from attack. There are different methods for detecting attacks and each method is used differently. With the detection algorithm implanted on to the 2 controllers, two different behaviors are observed. A comparative study of CPU Utilization of POX and RYU controller is studied and obtained that RYU occupies 15% lesser space for CPU utilization. The Execution time forPOX is 9% more than the RYU controller. The future work is to filter out the attacked packets from normal packets by designing a proper firewall in the switch.

## References

[1] Andry Putra Fajar and Tito WaluyoPurboyo, “A Survey Paper of Distributed Denial-of-Service Attackin Software Defined Networking (SDN)”,International Journal of Applied Engineering Research, ISSN 0973-4562, Volume 13, Number 1 (2018), pp. 476-482

[2] MohammadZ.Masoud,YousefJaradatandIsmaelJannoud,“OnPreventingArpPoisoningAttackUtilizingSoftwareDefinedNetwork(Sdn)Paradigm”,IEEEJordan Conference on Applied Electrical Engineering and Computing Technologies(AEECT),2015

[3] Jaideep Singh, Sandeep Dhariwal and Rajeev Kumar, “A DetailedSurveyofARPPoisoningDetectionandMitigationTechniques”,International Journal of Control Theory and Applications, pp. 131-137,February2017

[4] Youngin Kim, SungwonAhn, Nguyen Canh Thang, Dongho Choi,Minho Park, “ARP Poisoning attack Detection based on ARP Updatestate in Software-DefinedNetworks”,International Conference on Information Networking, IEEE, 2019

[5] R. Braga, E. Mota, and A. Passito, “Lightweight DDoS flooding attackdetection using nox/OpenFlow,” in IEEE Local Computer NetworkConference, Oct2010,pp.408–415.

[6] D. Hu, P. Hong, Y. Chen, “FADM: DDoS flooding attack detectionandmitigationsysteminsoftware-definednetworking”,InGLOBECOM 2017- 2017 IEEE Global Communications Conference,pp.1-7,2017December4,IEEE
[7] ZeynabSasan and Majid Salehi “SDN-based Defending against ARPPoisoningAttack”,JournalofAdvancesinComputerResearch,vol.8,no.2,p.95-102,2017.
[8] Feghali, Antoine, Rima Kilany and Maroun Chamoun.” SDN securityproblems and solutions analysis.” in Proc. Protocol Engineering andInternationalConferenceonNewTechnologiesofDistributedSystem, p.1-5,2015
[9] WenfengXia,YonggangWen,ChuanHengFoh,DusitNiyato,HaiyongXie, “A survey on Software-Defined Networking”, IEEECommunication Surveys & Tutorials, Vol. 17, No. 1, First Quarter2015.
[10] AnassSebbar, Mohammed Boulmalf, and Mohamed Dafir, “DetectionMITMAttackinMulti-SDNController”,IEEE,2018,pp.583-587
[11] NeelamDayalandShashankSrivastava,“AnRBF-PSO Based Approach forEarly Detectionof DDoSAttacksinSDN”,10thInternational Conference on Communication Systems & Networks(COMSNETS), 2018
[12] BabatundeHafisLAWALandNurayAT,“ImprovingSoftware-defined Network Security via Sflow and IPSEC protocol”, AnadoluUniversity Journalof Science andTechnology, May2018.
[13] Wajda M. Othman et al, “Implementation and Performance AnalysisofSDNFirewallonPOXController”,9thIEEEInternationalConferenceonCommunicationSoftwareandNetworks,2017,pp.1461-1466
[14] Peng Xiao et al, “An Efficient DDoS Detection with Bloom Filter inSDN”,IEEE TrustCom/BigDataSE/ISPA,2016
[15] Pradeep kumar Sharma and S S Tyagi, “Improving Security through Software Defined Networking (SDN): AN SDN based Model”, International Journal of Recent Technology and Engineering, 2019, pp.295-300
[16] Chaitra N. Shivayogimath and N.V. Uma Reddy, “Performance Analysis of a Software Defined Network Using Mininet,” Springer Artificial Intelligenceand Evolutionary Computations in Engineering Systems, pp 391-398, Feb.2016.
[17] Ahmed M.AbdelSalam, Ashrad B. El-Sisi and Vamshi Reddy.K, “Mitigating ARP Spoofing Attacks in Software-Defined Networks”, ICCTA, 2015.
[18] RaphaelHorvath, Dietmar Nedbal and Mark Stieninger, “A Literature Review on Challenges and Effects of Software Defined Networking”,Procedia Computer Systems, Elsevier, 2015, pp. 552-561
[19] Olivier Flauzac, Carlos González, AbdelhakHachani and Florent Nolot, “SDN based architecture for IoT and improvement of the security”, 29th International Conference on Advanced Information Networking and Applications Workshops, IEEE, 2015
[20] http://www.mininet.org/
[21] https://www.wireshark.org/