Journal of Advances in Computer Research
Quarterly pISSN: 2345-606x eISSN: 2345-6078
Sari Branch, Islamic Azad University, Sari, I.R.Iran
(Vol. 8, No. 2, May 2017), Pages: 95-
www.jacr.iausari.ac.ir

SDN-based Defending against ARP Poisoning Attack

Zeynab Sasan✉ and Majid Salehi
Department of Computer Engineering, Sharif University of Technology, Tehran, Iran
sasan@ce.sharif.edu; masalehi@ce.sharif.edu

Received: 2016/08/07; Accepted: 2016/08/27

Abstract

In the field of computer networks, the introduction of SDN has been associated with new concepts. In SDN networks, control plane is separated from the data palne. Traditional networks suffer from difficult configuration and management. In other words, a change in the network needs to be configured on the whole equipment. With the introduction of SDN, various modules can be designed and run in controller in order to perform expected policies and rules on all switches. One of the areas of network management is to deal with cyber-attacks. In SDN networks, security modules can be designed to run in the controller and generate rules on switches. Due to the importance of intranets, this paper aimed to detect and prevent ARP poisoning attack on LAN. The tests in a LAN showed that the module can detect the ARP poisoning attack and block the attacker operation.

Keywords: *Software Defined Networking, ARP Poisioning Attack, Network Security, Mininet, POX*

1. Introduction

Computer networks consist of two sections: core and edge. Devices and hosts are located in the edge and routers and switches in core. In other words, computer networks are highly distributed structures. If we aim to implement an end-to-end service in this distributed structure or impose changes in the existing configuration, we are faced with a very difficult task. Traditional networks are typically faced with such difficulties. In other words, management and implementation of new services is too difficult because the changes must be performed on all equipments.

The emergence of SDN networks has brought new principles into the field of computer networks. In SDN approach, control and data planes are separate. Control plane is run in a central entity, called controller. The data plane is implemented in hardware switches. Controller is considered the brain of the network, which can be planned and various performances can be designed and run through modules. Switches are not able to make decisions and while facing new traffic, send the traffic to the controller in order to generate new rules for responding.

Using SDN approach, network manager can implement and run different modules in controller. Detection and prevention of attacks in network is one of the management topics. Attacks can also be one of the most important threats for every network by the users. Attackers, who manage to access the intranet, can carry out every type of attack in LAN. ARP poisoning is one of attacks in LAN which can prepare the ground for other attacks. This paper aimed to design modules in the controller to detect and block such attacks. Tests in certain topologies help us observe that the module was able to detect and block the attack.

The paper consists of the following sections: The second section is briefly dedicated to ARP poisoning. Detection and prevention methods were also described in traditional networks in this part. In the third section, SDN was explained and the protocol used for the link between switches and controllers. The ARP poisoning was also raised. In the fourth section, we described the designed module in the controller for detecting and preventing the ARP poisoning followed by the results. Evaluations were made in the fifth section. Finally, conclusion and future studies were taken into account.

2. ARP Poisoning Attack

In a LAN, devices communicate with each other using MAC address. While sending a message, the sender must know the recipient's MAC address. The sender first checks the ARP cache to find the recipient's MAC address, and if he did not find it in ARP table, he would create a new ARP request and send it to all the devices on the local network as all broadcast. In this request, in MAC address and IP field, the sender writes his own addresses. Recipient's IP address part is also completed, but MAC address field of the recipient remains empty. The device whose IP address is equal to IP address of the destination in ARP request, replies this request with ARP reply. In other words, what ARP protocol does is converting IP address to MAC address.

However, there are some security weaknesses in this protocol, which make it vulnerable to various attacks. For example, this protocol is stateless. In other words, even if it has not sent ARP request, it receives ARP reply and updates ARP table and this makes ARP poisoning attack possible. As this attack happens at LAN level, it is classified as internal attacks and threats.

In the Figure 1, attack scenario is fully shown. Host A wants to send a message to Host B in the form of Ethernet frame, but it does not know its MAC address, so it searches its address in form of ARP request. Host B informs Host A of its MAC address via ARP reply, and A updates its ARP table. However, as we stated, ARP is a stateless protocol and Host C (attacker) could send another ARP reply with its MAC address to Host A. the attacker claims that its MAC address corresponds to IP address of Host.

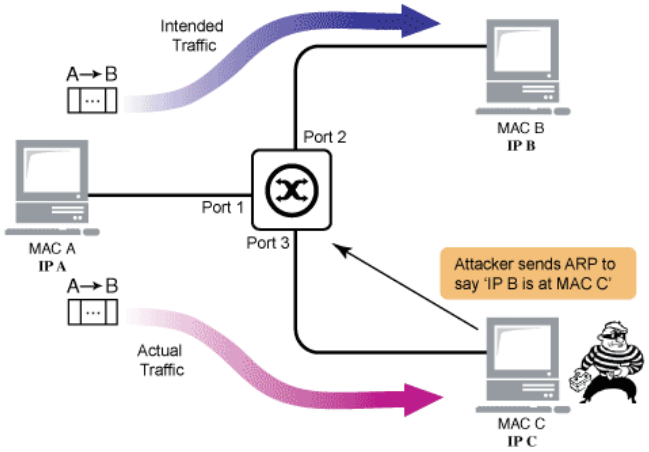

Figure 1 ARP Poisioning Attack

ARP poisoning attack is classified as man-in-the-middle attacks, and by doing so, the attacker could intercept network traffic. Having traffic, the attacker can extract sensitive and important information from it and make other attacks such as session hijacking. It is possible that the messages forwarded to the attacker are not sent again to the real recipient, in which case, denial of service (DoS) attack will happen. If the attacker is able to manipulate traffic, the receiver will receive wrong messages and its operations may be disrupted.

Study [1] has conducted a comprehensive review on methods of detecting and preventing ARP poisoning attacks in traditional networks. For a closer look at these methods, you can refer to this study. In this section, some of the most important methods have been described briefly. Different versions of ARP protocol have provided so far, one of which is safe ARP [2]. This version uses encryption and digital certificates to prevent this attack. Using these certificates, the users could authenticate ARP reply at the network level.

In some methods, some patches have been introduced at kernel level. An example of this method is Antidot [3] that is implemented on hosts. In this method, the host receiving ARP reply checks its table and if another MAC address is already registered for that IP address, it checks previous address being alive. If the previous host is alive, it will ignore the new MAC address and the address will be blocked.

3. Software Defined Networking

Network management is a very complex and challenging process, as high-level policies should be applied on different low-level equipment. SDN is a new approach in the field of computer networks, which has introduced the concept of separating data plane from control plane. In this approach, switches are only responsible for packet forwarding and management of networks runs in a central part called controller. With the advent of SDN, network configuration and management are greatly facilitated [4].

In SDN, in the case that switches see a new traffic, they send it towards the controller, so that the controller issues proper policies and rules for correct management of the traffic, and they are set on the switch. Using SDN will have many advantages for network administrators. One of them is that new ideas can easily be implemented at the network level. The other advantage is that we are not faced with a series of fixed and limited commands and can develop various software programs and run them on the controller. Second, out of a complex and distributed approach in network management, we have reached a centralized solution [5].

Figure 2 shows a schematic of the layered structure in SDN. At the lowest level, there are the switches and hardware equipment that forward packets merely based on rules they receive from the controller. In the higher layer, there is the controller, which is considered as the decision-maker brain of the network. In the higher layer, there are software and different applications that run on the controller.

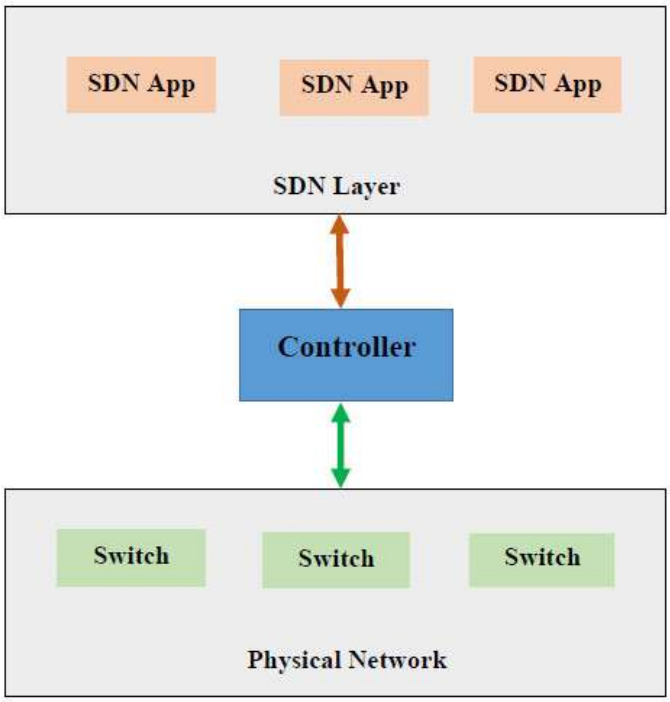

Figure 2 Schematic of Software Defined Networking Concept

In rest of this section, we introduce openflow protocol, which is used for communication between the switches and controller. Moreover, considering the limitation in test environment in this study, we have used a simulated environment. Mininet is used to design a network of switches, and POX sample is used for the controller. Each of these tools will be described briefly.

3.1 Openflow

Openflow protocol is one of the first standards in the field of SDN. This protocol is used for communication between the controller and hardware such as switches [6]. Both switches and the controller defined on the network must use this protocol to establish communication. This protocol uses TCP protocol for sending messages in the transport layer and for security in the communication channel, we can use TLS protocols as well. Controller sends the new rules to switches with this protocol. Switches inform the controller of status messages via this protocol. Moreover, if switches observe a new traffic, they have no rules to comply with, to issue new rules; they send the traffic to the controller.

3.2 Mininet

In situations that require a large network of switches and routers, new ideas cannot be tested on the real networks for efficiency reasons. In some cases, it is not possible to access SDN equipment. Mininet [7] simulation environment has been introduced to solve such problems in the process of developing SDN solutions. Large networks with different topologies can easily be designed in this environment that support technologies, processes, and namespace defined in SDN.

3.3 POX

POX is an open source controller that using application programming interfaces (API) considered in it, one can write applications for SDN environment [8]. Programming in this controller is done using the Python language. Programs and modules defined to run on the controller must be placed in POX directory. Moreover, while defining topology in Mininet, POX controller address should also be included.

4. ARP Poisoning Defense Module

As mentioned in the previous sections, controller is programmable and various security modules can be designed and run on it. In this section, a module is designed and implemented on a controller to detect and prevent ARP poisoning attack. The main function of this module is shown in Figure 3. First, we assume that using a pre-defined list, IP address records and correct MAC are at the disposal of security module. In case of receiving ARP reply, switches send it towards the controller.

Security module extracts source IP and MAC addresses from this ARP reply. Then from the predefined list, it finds the record in accordance with this IP address. If the MAC address in the packet and the record on the list were different, the module would issue a warning of an ARP poisoning attack. This module can also issue rules to switches, so that in case of observing this MAC address, block its traffic. However, if the MAC address in the packet and record are the same, there is no threat, and the packet is delivered to another management module to produce rules.

The intended module called *arppois* has been written using Python language. Mininet simulation environment and POX as a controller have been used for networking. arppois.py module is in POX directory and runs on it. We will explain step-by-step scenario of attack and detection using aprpois module.

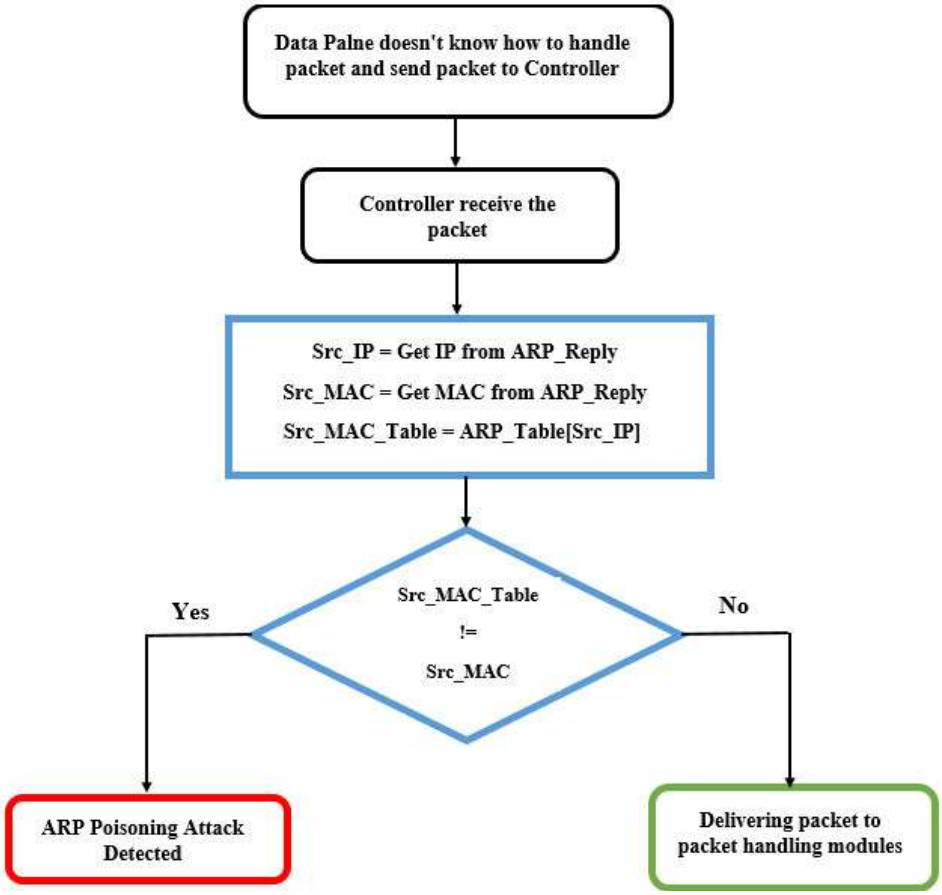

Figure 3 Functionality of controller and ARP Poisoning Defense module.

First, the controller must be set up. In this scenario, we want to run designed security modules on network switches. It is also essential that modules related to operations of layer 2 switches such as l2_learning module to be run. Thus, we run the following command in the terminal of Linux machine to run the controller:

```
./pox.py arppois forwarding.l2_learning
```

Then we should form the intended topology for the network. In this scenario, we want the network to be composed of a switch and three hosts, which is called single topology. Using Mininet, we build this topology and enter remote controller's address:

```
sudo mn --controller=remote, ip=127.0.0.1, --topo single,3
```

We run pingall command to operate and establish a normal traffic at the network level.

```
pingall
```

Finally, the process of attack begins. For this purpose, the ready tool in Linux called arpspoof has been used. To this end, we open an Xterm on Host 1 and enter the following command. By running this command, Host 1 will poison traffic between Host 2 and 3.

```
arpspoof -t 10.0.0.2 10.0.0.3.
```

In this scenario, Host 1 is considered as the attacker that poisons traffic between Host 2 and 3. In this case, arppois security module has been used; the controller issues a

message based on an IP address written for two different MAC addresses and blocks Host 1 as an attacker.

5. Evaluation

In study [9], some features have been presented to provide an ideal solution to detect and prevent ARP poisoning attack. In this section, we mention some of the most important features and review the proposed solution in terms of meeting the requirements. Some important requirements are:

- The proposed solution should not require major changes on a lot of equipment because management cost increases.
- As far as possible, encryption algorithms should not be used or their use should be minimized because encryption algorithms slow ARP protocol.
- The proposed solution must be compatible with ARP protocol and should not require changes in ARP request and reply.
- Single point of failure (SPOF) should not occur in the desired solution.

Arppois module, which was introduced in the previous section, does not need changes in the hosts and network equipment and also no encryption algorithm is used in it, so items 1 and 2 are satisfied in the proposed solution. However, in the proposed solution because the ARP reply is sent towards the controller, delay may occur in the process of ARP protocol, but this module will not perform any manipulation on ARP request and reply. Concerning SPOF feature, as the module runs on a controller and the controller can be targeted by attackers, this feature it not satisfied by the proposed solution, which is a centralized solution. However, using the special protection of the controller or using distributed controllers, one can somewhat improve this problem.

6. Conclusion

As mentioned, the discussion of management and configuration of new services in traditional networks is a complex and difficult. SDN approach has been presented to overcome this issue, where the controller as the brain of the network is responsible for all management operations. In this study, we showed that SDN could be used to detect and prevent attacks on LAN. A module called arppois was designed and run on the controller that had the ability to detect and prevent ARP poisoning attack. Mininet simulation environment and POX controller were used for testing and evaluation. The results showed that using this module, one could detect such attacks at LAN and block the attacker.

Although in this paper, we solely dealt with designing modules for ARP poisoning attack, using SDN approach can be generalized to detect and prevent other attacks of data-link layer. In other words, the centralized approach at SDN controller enables network administrator to design and implement various security modules on the controller. Thus, designing a security infrastructure using SDN approach can be considered as future work.

References

[1] Tripathi, Nikhil, and B. M. Mehtre. "Analysis of various ARP poisoning mitigation techniques: A comparison." Control, Instrumentation, Communication and Computational Technologies (ICCICCT), 2014 International Conference on. IEEE, 2014.

[2] N. Saputro and K. Akkaya, "An efficient and secure arp for largescale ieee 802.11 s-based smart grid networks," in Ad Hoc Networks. Springer, 2014, pp. 214–228.

[3] "Antidote : Open source detection of arp poisoning," http://antidote.sourceforge.net/, accessed: 2015-08-26.

[4] H. Kim and N. Feamster, "Improving network management with software defined networking," Communications Magazine, IEEE, vol. 51, no. 2, pp. 114–119, 2013.

[5] W. Xia, Y. Wen, C. H. Foh, D. Niyato, and H. Xie, "A survey on software-defined networking," Communications Surveys & Tutorials, IEEE, vol. 17, no. 1, pp. 27–51, 2014.

[6] McKeown, Nick, et al. "OpenFlow: enabling innovation in campus networks." ACM SIGCOMM Computer Communication Review 38.2 (2008): 69-74.

[7] Mininet. (2013, Mar). An Instant Virtual Network on your Laptop (or other PC). [Online]. Available: http://mininet.org/.

[8] Fernandez, Marcial. "Evaluating OpenFlow controller paradigms." In ICN 2013, The Twelfth International Conference on Networks, pp. 151-157. 2013.

[9] Abad, Cristina L., and Rafael I. Bonilla. "An analysis on the schemes for detecting and preventing ARP cache poisoning attacks." *Distributed Computing Systems Workshops, 2007. ICDCSW'07. 27th International Conference on*. IEEE, 2007.