2018 9th International Conference on Information Technology in Medicine and Education

ARP Poisoning Prevention in Internet of Things

Weihua Gao, Yuhao Sun, Qingying Fu, Zhouzhe Wu, Xiao Ma, Kai Zheng, Xin Huang
Dept. of Computer Science and Software Engineering
Xi’an Jiaotong-Liverpool University
Suzhou, China
Weihua.Gao16@student.xjtlu.edu.cn, Yuhao.Sun16@student.xjtlu.edu.cn,
Qingying.Fu16@student.xjtlu.edu.cn, Zhouzhe.Wu16@student.xjtlu.edu.cn, Xiao.Ma16@student.xjtlu.edu.cn,
Kai.Zheng@xjtlu.edu.cn, Xin.Huang@xjtlu.edu.cn

***Abstract*—Address Resolution Protocol (ARP) is responsible for parsing an IP address into a corresponding MAC address. ARP attacks still threaten the Internet of Things (IoT). In order to find a method to prevent ARP attack from attacking IoT, this paper describes an ARP attack defense method for the IoT. This ARP defense mainly involves binding the IP address of the single-chip microcomputer to MAC address of Ubuntu in the router’s static ARP cache table. Use Ubuntu as a monitor through enabling IP forwarding function. Detect attacker via Wireshark and intercept malicious ARP packets with arptables tool in the end. The method intercepts the attacker’s information and blocks the ARP attack successfully to a certain extent.**

Keywords—Internet of things (IoT), Address resolution protocol (ARP) poisoning, Kali Linux

I. Introduction

The Address Resolution Protocol (ARP) is a TCP/IP protocol that acquires physical addresses based on the IP addresses of nodes in the network [1], and it is widely utilized in existing IoT systems. This protocol has been proven to be highly efficient in general situations. However, it is often at risk of being attacked because ARP protocol is built on the mutual trust of the nodes in the network. When there is a malicious node in the network, it causes the attacked node to match its physical address with the IP address of the original communication node by sending a fraudulent message to the attacked node, and the error pairing will be recorded in ARP cache table [2]. In this way, the attacked node will mistakenly believe that the malicious node is the node to communicate and sends data packets to the attacker. Since ARP follows the principle of post-priority principle [2], as long as the attacker continues to send wrong messages, the execution of the spoofing will be guaranteed.

In this paper, an ARP defense approach will be designed and implemented. This method adds an extra node serving as an agent of protected node to ensure that the node which need to be protected do not receive malicious ARP packets that is all packets sent to this node are checked and filtered, thereby preventing the node from encountering ARP spoofing. The contribution in this paper is shown as following:

- Design and implement a defensive node that serve as a “gateway” to protect other nodes from ARP spoofing and ARP attacks.
- Through experiments, present the difference of the results of ARP attack before and after the node taking the defense methods proposed in this paper.
- Analyze this defense method and proposes preliminary ideas for its improvement.

The structure of this paper is as follows. Section Ⅱ introduces the related work and make a comparison of our proposal with the related ARP defense methods. Section Ⅲ shows the platform of our experiments. Section Ⅳ describes the specific process of the experimental operation. Section Ⅴ presents the conclusion of our work.

II. Related Work

So far, Internet of Things (IoT) is still vulnerable and suffers the various kind of attacks, which are including but not limited to DoS attack (Denial of Service), MIM attack (Man-in-Middle) and ARP attack/poisoning (Address Resolution Protocol).

The professionals in this area have raised some proposals focus on IoT attack specifically. Generally, there are two potential main approaches which are hardware-based and software-based.

In the paper [3], [4], both based on the idea that adding the entity of the server in network, to try to decrease computational load (more specifically, the load of authentication and authorization) from IoT devices, in order to prevent the attacks to IoT. Also, the paper [5], [6] can prove rationality and legitimacy of approach above. All proposals which are mentioned in [3], [4], [5], [6] are high-cost. According to illustration by scholars in [3] and [4], this approach is not most-efficiency currently.

In the paper [7], [8], [9], Software-defined networking (SDN) is regarded as a novel kind of solution to IoT attacks. Under the protection of SDN, which is different from static firewall, the control plane and the data plane within SDN separates. It allows some dynamic software to manage network resources freely and safely, more significantly. Due to the big capacity of SDN demonstrated in [7], [8], none of above approaches related to SDN are low-consumption. According to the scientific analysis by Hossain and his colleagues [10], unstable system and unrestricted platform will be attacked easily when without the latest patches and updates.

2474-3828/18/$31.00 ©2018 IEEE
DOI 10.1109/ITME.2018.00166

IEEE computer society

There are also some other kinds of classical approaches to prevent attacks in IoT. A new trend idea, using Blockchain (basic technology for bitcoin) to prevent attacks in IoT are raised lately in paper [11], [12]. Bahga and Madisetti, which are authors in paper [12], didn't refute the impossibilities using Blockchain to prevent attacks in IoT but emphasized "potential" approach.

The approach in the paper will prevent ARP poisoning specifically in IoT via a software "arptables". The system will not only catch the information of attackers, but filter or block the attackers. Compared to aforementioned approaches, by using a lightweight and easy-to-learn software, for one thing, our approach can significantly reduce the cost and power-consumption. For another, based on the lightweight feature of the approach, it is easily to use and update essential and latest patches. According to several tests, this approach is stable and mature currently. More relevant details will be introduced following.

III. The Platform for ARP Attack Prevention

In Bergmann and Lin's article, they introduce "Gateway Architectures" as one of three most important and popular architectures for IoT security [13]. Gateways can centralize user authentication or act as a firewall, protecting smart devices and privacy from cyber threats [13]. Inspired by it, we consider transferring the concept of "Gateway Architectures" to a Linux machine and using a Linux machine as a "gateway". Then we utilize Linux machine to monitor, figure out the attacker, and finally intercept malicious ARP packets with a software "arptables".

A. The Framework of Experiment Platform

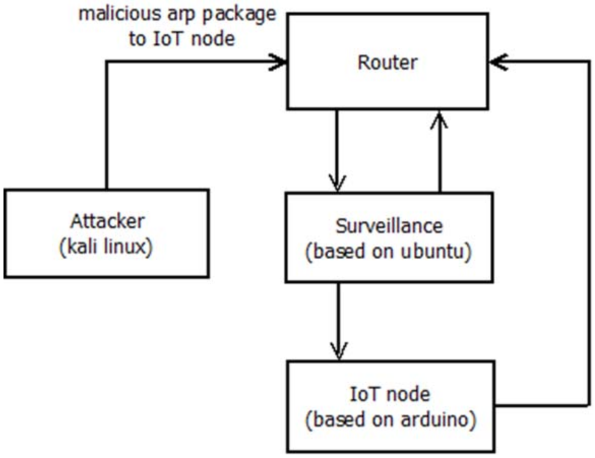


Fig. 1. The framework of our experiment

1) Router: The router connects the PC(Surveilliance) with Ubuntu operating system, Arduino and the Kali Linux.

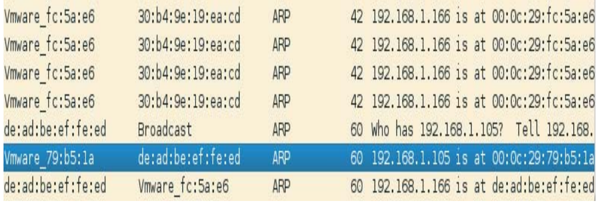
2) Surveilliance*:* Through enabling IP forwarding function in Ubuntu and binding Arduino's IP address to Ubuntu's Mac address in router's static ARP table, packets send to Arduino must get through Ubuntu first. As a result, Ubuntu works as a surveillance.

3) IoT node (based on Arduino): It sends out packets continuously.

4) Attacker (Kali Linux): Kali Linux is installed in a VM of VMware workstation. It is used to execute ARP attack to IoT node.

B. The Preparation for The Experienment

The ARP protocol contains inherent vulnerabilities that can be easily exploited by malicious attackers. One of the loopholes is in the form of ARP, which can be easily exploited to initiate an ARP storm [3]. In order to do ARP attack prevention experiment, first use Kali Linux to attack microcontroller Arduino and then execute prevention to test whether prevention works.



```
Vmware_fc:5a:e6     30:b4:9e:19:ea:cd   ARP     42 192.168.1.166 is at 00:0c:29:fc:5a:e6
Vmware_fc:5a:e6     30:b4:9e:19:ea:cd   ARP     42 192.168.1.166 is at 00:0c:29:fc:5a:e6
Vmware_fc:5a:e6     30:b4:9e:19:ea:cd   ARP     42 192.168.1.166 is at 00:0c:29:fc:5a:e6
Vmware_fc:5a:e6     30:b4:9e:19:ea:cd   ARP     42 192.168.1.166 is at 00:0c:29:fc:5a:e6
de:ad:be:ef:fe:ed   Broadcast           ARP     60 Who has 192.168.1.105?  Tell 192.168.
Vmware_79:b5:1a     de:ad:be:ef:fe:ed   ARP     60 192.168.1.105 is at 00:0c:29:79:b5:1a
de:ad:be:ef:fe:ed   Vmware_fc:5a:e6     ARP     60 192.168.1.166 is at de:ad:be:ef:fe:ed
```

Fig. 2. Arduino sent packets to PC

Finding. Figure 2 shows that packets send to Arduino are captured by Wireshark in Kali Linux after execute ARP attack using Kali Linux. We can also listen to the package issued by Arduino, which means that ARP attack is successfully implemented.

In the end, APR attacks were successfully implemented and the information of IoT nodes users was eavesdropped as well.

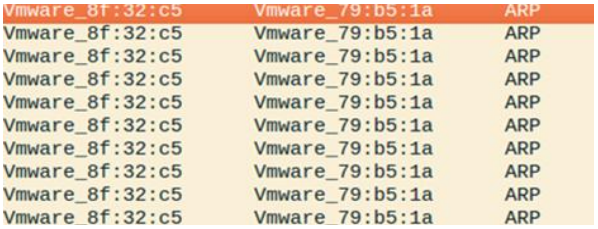
IV. The process of Arp Attack Prevention Experience

A. Monitoring and Detecting for ARP Attack

In the experience, a virtual machine with Ubuntu operating system works as a surveillance. It needs two steps to make virtual machine work as a monitor. The first step is enabling IP forwarding in the Linux kernel to make virtual machine work as a router. The second step is binding the IP address of IoT nodes to the Mac address of virtual machine. The first step could be achieved by typing in "echo 1 > /proc/sys/net/ipv4/ip_forward" command line in the terminal. This command sets the value of ip_forward to 1 therefore Ubuntu enables IP forwarding function. Afterwards, in the experience, bind the Arduino's MAC address to the Ubuntu's Mac address in the static ARP cache table on the router so that all packets sent to the Arduino via the router will be sent to Ubuntu first and then to Arduino via Ubuntu. In this way, the virtual machine achieves the function of monitoring all ARP packets send to IoT nodes.

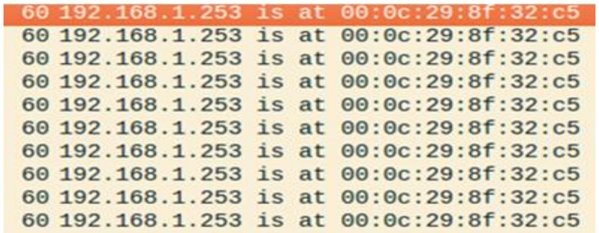
As for detecting function, after installing Wireshark software on Ubuntu, we use the promiscuous mode to capture packets through the network adapter, and use the "eth.dst==UbuntuMacAddr" filter command to perform packet analysis. Above could be done through input "wireshark -f 'arp and ether host ubuntuMacAddr ' -c 500 –k" command in the terminal. " ubuntuMacAddr" would be replaced by Ubuntu's Mac address. " -f 'arp and ether host ubuntuMacAddr ' " is the filter when capture packets. " -c 500" means Wireshark terminates when captures 500 packets.

"-k" means that Wireshark starts to capture right now. When users figure out ARP packets which replace router's real Mac address with fake Mac address, in general, the sender of these packets is the attacker. As shown in Figure 3 and 4, "00:0c:29:8f:32:c5" is attacker's Mac address.



```
Vmware_8f:32:c5        Vmware_79:b5:1a        ARP
Vmware_8f:32:c5        Vmware_79:b5:1a        ARP
Vmware_8f:32:c5        Vmware_79:b5:1a        ARP
Vmware_8f:32:c5        Vmware_79:b5:1a        ARP
Vmware_8f:32:c5        Vmware_79:b5:1a        ARP
Vmware_8f:32:c5        Vmware_79:b5:1a        ARP
Vmware_8f:32:c5        Vmware_79:b5:1a        ARP
Vmware_8f:32:c5        Vmware_79:b5:1a        ARP
Vmware_8f:32:c5        Vmware_79:b5:1a        ARP
Vmware_8f:32:c5        Vmware_79:b5:1a        ARP
```

Fig. 3. Malicious ARP Packet



```
60 192.168.1.253 is at 00:0c:29:8f:32:c5
60 192.168.1.253 is at 00:0c:29:8f:32:c5
60 192.168.1.253 is at 00:0c:29:8f:32:c5
60 192.168.1.253 is at 00:0c:29:8f:32:c5
60 192.168.1.253 is at 00:0c:29:8f:32:c5
60 192.168.1.253 is at 00:0c:29:8f:32:c5
60 192.168.1.253 is at 00:0c:29:8f:32:c5
60 192.168.1.253 is at 00:0c:29:8f:32:c5
60 192.168.1.253 is at 00:0c:29:8f:32:c5
60 192.168.1.253 is at 00:0c:29:8f:32:c5
```

Fig. 4. Malicious ARP Packet

TABLE I. IP ADDRESS AND MAC ADDRESS

Device	IP Address	Mac Address	Role
Router	192.168.1.253	30:b4:9e:19:ea:cd	Local Network
Ubuntu	192.168.1.106	00:0c:29:79:b5:1a	Surveillance
Kali Linux	192.168.1.107	00:0c:29:8f:32:c5	Attacker
Arduino	192.168.1.166	de:ad:be:ef:fe:ed	IoT Node

Replacing the gateway address with fake address is simple to detect. However for most IP address it is difficult to store their real MAC address. Additionally, an attacker may be hidden behind high volume of packets so that it remains undiscovered for a significant amount of time [14]. In this way, user should compare the number of ARP request packets and ARP answer packets. If user find over 10 same content ARP answer packets almost at same time and less than one third ARP request packets for this IP address, generally MAC source address is attacker's MAC address.

B. *Intercepting Malicious ARP Packets*

In the experiment, use specialized lightweight tools arptables on Ubuntu to manage ARP packets. The arptables command in Linux can be used to set, maintain, and check the ARP packet filter rule tables in the Linux kernel. First, use the "vim arptables.sh" command to create the arptables.sh file and add input and output rules to it. For instance, the codes of intercepting malicious ARP packets from attacker's Mac address "00:0c:29:79:b5:1a" shown as follows:

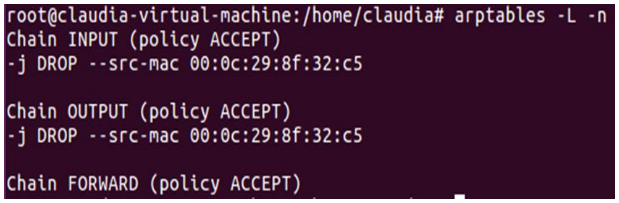
```
#!/bin/sh

PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

arptables -A INPUT --src-mac 00:0c:29:8f:32:c5 -j DROP

arptables -A OUTPUT --src-mac 00:0c:29:8f:32:c5 -j DROP
```

Afterwards, we can run the script using "./arptables.sh" command, and check arptables using "arptables –L -n" command, which is shown in Figure 5. Then use Kali Linux to attack Arduino with command "arpspoof –i eth0 –t 192.168.1.166 192.168.1.253". It can be seen that all malicious ARP packets have been intercepted, and use Wireshark in Kali Linux to capture packets again. It could be seen that Kali Linux cannot eavesdrop Arduino now. Compared with figure II, it is shown that prevention of an ARP attack on IoT nodes executes successfully.



```
root@claudia-virtual-machine:/home/claudia# arptables -L -n
Chain INPUT (policy ACCEPT)
-j DROP --src-mac 00:0c:29:8f:32:c5

Chain OUTPUT (policy ACCEPT)
-j DROP --src-mac 00:0c:29:8f:32:c5

Chain FORWARD (policy ACCEPT)
```

Fig. 5. ARP Tables

C. *Analysis*

This method could be used to monitor multiple IoT devices at the same time by binding IoT nodes' IP address to Ubuntu's MAC address in router's static ARP table. The system will not only catch the information of attackers, but filter or block the attackers. This is a lightweight and easy-to-learn method. Our approach can significantly reduce the cost and power-consumption and can be easily updated due to lightweight feature. However, this method does not apply to numerous IoT devices, because users need to find out attacker via Wireshark and manually add rules using arptables tool. If it could be designed to use the script to complete the detection of the attacker's machine and the setting of the arptables' rules in the future, this approach will be more effective.

V. CONCLUSION

In conclusion, this article describes an approach of ARP attack defense on the Internet of Things. The technology of the IoT is not yet mature. IoT system based on the LAN is relatively vulnerable to ARP attacks and its anti-preventions need further improvement. For prevention method, first capture the ARP packets sent by the Kali Linux to the single-chip microcomputer and analyzes packets via Wireshark to obtain the attacker. Afterwards intercept malicious ARP packets with arptables tool to protect the Internet of Things from ARP attacks. Finally, the experiment result show that prevention method works.

In the future, using shell script to complete above whole process will be studied.

ACKNOWLEDGMENT

This work was supported by the XJTLU research development fund projects under Grant RDF140243 and Grant RDF150246, in part by the National Natural Science Foundation of China under Grant No. 61701418, in part by Innovation Projects of The Next Generation Internet Technology under Grant NGII20170301, in part by the Suzhou Science and Technology Development Plan under Grant SYG201516, and in part by the Jiangsu Province National Science Foundation under Grant BK20150376.

REFERENCES

[1] A. Cristina and I. B. Rafael., "An Analysis on the Schemes for Detecting and Preventing ARP Cache Poisoning Attacks." *27th International Conference on Distributed Computing Systems Workshops* (ICDCSW'07), Toronto, Ont., Canada, 2007

[2] G. Vipul, and R.Tripathy. "An Efficient Solution to the ARP Cache Poisoning Problem." *Australasian Conference on Information Security and Privacy*, 2005, pp. 40–51.

[3] Maheshwari, N., & Dagale, H. (2018, January). Secure communication and firewall architecture for IoT applications. *In Communication Systems & Networ*ks (COMSNETS), *2018 10th International Conference* on (pp. 328-335). IEEE.

[4] Kim, H., Wasicek, A., Mehne, B., & Lee, E. A. (2016, August). A secure network architecture for the internet of things based on local authorization entities. In *Future Internet of Things and Cloud* (FiCloud), *2016 IEEE 4th International Conference* on (pp. 114-122). IEEE.

[5] Dixit, A., Hao, F., Mukherjee, S., Lakshman, T. V., & Kompella, R. (2013, August). Towards an elastic distributed SDN controller. In *ACM SIGCOMM Computer Communication Review* (Vol. 43, No. 4, pp. 7-12). ACM.

[6] Aloul, F., Zahidi, S., & El-Hajj, W. (2009, May). Two factor authentication using mobile phones. In *Computer Systems and Applications*, 2009. AICCSA 2009. *IEEE/ACS International Conference* on (pp. 641-644). IEEE.

[7] Gonzalez, C., Charfadine, S. M., Flauzac, O., & Nolot, F. (2016, July). SDN-based security framework for the IoT in distributed grid. In *Computer and Energy Science* (SpliTech), *International Multidisciplinary Conference* on (pp. 1-5). IEEE.

[8] Flauzac, O., Gonzalez, C., Hachani, A., & Nolot, F. (2015, March). SDN based architecture for IoT and improvement of the security. In *Advanced Information Networking and Applications Workshops* (WAINA)*, 2015 IEEE 29th International Conference* on (pp. 688-693). IEEE.

[9] El-Mougy, A., Ibnkahla, M., & Hegazy, L. (2015, October). Software-defined wireless network architectures for the Internet-of-Things. In *Local Computer Networks Conference Workshops* (LCN Workshops)*, 2015 IEEE* 40th (pp. 804-811). IEEE.

[10] Hossain, M. M., Fotouhi, M., & Hasan, R. (2015, June). Towards an analysis of security issues, challenges, and open problems in the internet of things. In Services (SERVICES), 2015 IEEE World Congress on (pp. 21-28). IEEE.

[11] Khan, M. A., & Salah, K. (2017). IoT security: Review, blockchain solutions, and open challenges. Future Generation Computer Systems.

[12] Bahga, A., & Madisetti, V. K. (2016). Blockchain platform for industrial Internet of Things. *Journal of Software Engineering and Applications*, 9(10), 533.

[13] H. Lin and N W.Bergmann, "IoT Privacy and Security Challenges for Smart Home Environments," *Information*.,vol.7,no.3,pp.44, Jul.2016

[14] Abad, C. L. and Bonilla, R. I. (2007). An Analysis on the Schemes for Detecting and Preventing ARP Cache Poisoning Attacks. *International Conference on Distributed Computing Systems Workshops* (pp. 60-60). IEEE.