Information Security and Computer Fraud, 2019, Vol. 7, No. 1, 1-6
Available online at http://pubs.sciepub.com/iscf/7/1/1
Published by Science and Education Publishing
DOI:10.12691/iscf-7-1-1

Lightweight Man-In-The-Middle (MITM) Detection and Defense Algorithm for WiFi-Enabled Internet of Things (IoT) Gateways

Justice Owusu Agyemang, Jerry John Kponyo*, Isaac Acquah

Faculty of Electrical/Computer Engineering, Kwame Nkrumah University of Science and Technology, Kumasi, Ghana
*Corresponding author: jjkponyo@ieee.org

Received December 12, 2018; Revised January 13, 2019; Accepted January 14, 2019

Abstract Man-In-The-Middle (MITM) attack is one of the well-known threats in computer security. With the convergence of smart objects and the Internet (Internet of Things), there has been the need to ensure confidentiality, integrity and availability of information. MITM targets the data flow between endpoints, and the confidentiality and integrity of the data itself. In this paper, we present a lightweight and real-time MITM detection and defense algorithm that can be implemented on WiFi-enabled IoT gateways. The algorithm works for statically assigned host IP addresses and also IP addresses assigned via DHCP. We employed Asynchronous Method Dispatch (AMD) to reduce performance overhead. Subsequently, we evaluated the performance of the algorithm with respect to CPU utilization, detection rate and network latency.

Keywords: *IoT, MITM, IDS, DoS, AP, EAPOL, ARP, DHCP*

Cite This Article: Justice Owusu Agyemang, Jerry John Kponyo, and Isaac Acquah, "Lightweight Man-In-The-Middle (MITM) Detection and Defense Algorithm for WiFi-Enabled Internet of Things (IoT) Gateways." *Information Security and Computer Fraud*, vol. 7, no. 1 (2019): 1-6. doi: 10.12691/iscf-7-1-1.

1. Introduction

The internet revolution has redefined business-to-customer (B2C) industries such as media, retail and financial services. This revolution has led to the emergence of Internet of Things (IoT); a ubiquitous global computing network where everyone and everything is connected to the Internet. The number of networked devices keeps increasing daily. It is estimated that about 50 billion devices will be connected by the year 2020 (shown in Figure 1).

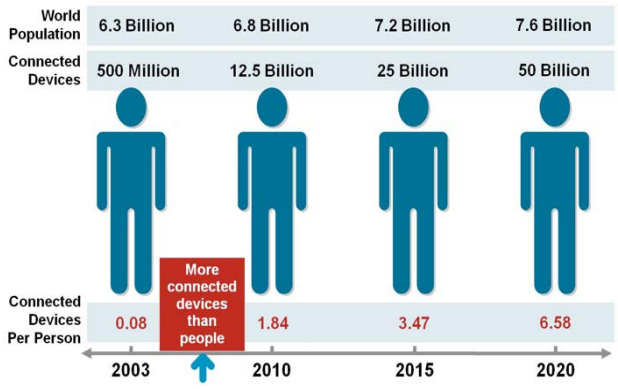

Figure 1. The future of connected devices [1]

The direct connection of these heterogeneous devices to the Internet increases their chances of being exposed to several security threats. These threats include cloning of things, malicious substitution of things, firmware replacement, extraction of security parameters, eavesdropping, Man-In-The-Middle (MITM) attack [2]. Most IoT devices use WiFi technology [3]; hence susceptible to conventional wireless attacks. The susceptibility of IoT devices to MITM attack has been demonstrated [4].

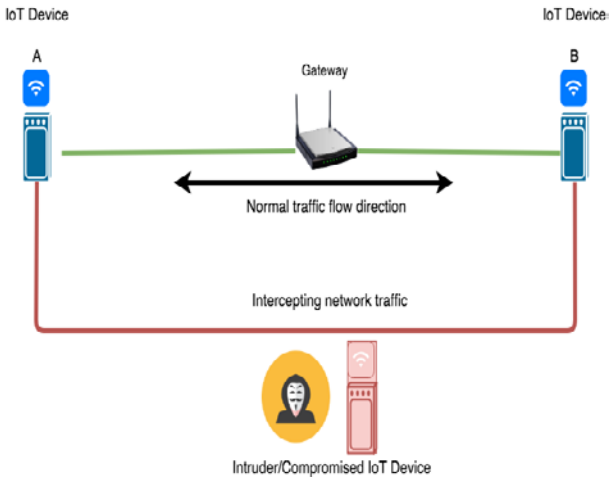

Figure 2. MTIM Attack

MITM attack is an eavesdropping attack, where in a communication session between two client devices A and B, the attacker deceives A by pretending to be B. This enables the attacker to read or modify messages sent from A to B (shown in Figure 2). This attack is possible due to

the weakness in the Address Resolution Protocol (ARP). ARP is a protocol used by the data link layer to map IP addresses to MAC addresses [5]. Before encapsulating the network layer packet in a data link layer frame, the host sending the packet needs to know the recipient's MAC address. Given the IP address of a host, to find its MAC address, the source node broadcasts an ARP request packet which asks about the MAC address of the owner of the IP address. This request is received by all nodes inside the local area network (LAN). The node that owns this IP address replies with its MAC address (unicast) [6].

ARP is a stateless protocol and has a lack of security in its caching system [7]. It accepts ARP replies without considering if an ARP request was sent. This weakness can be exploited by an attacker to initiate MITM attack. A denial of service (DoS) attack can occur if the attacker drops the received packet without forwarding it to the intended destination.

Current MITM solutions are inadequate for IoT systems, because of certain characteristics of IoT that affect IDS development [6].

In this paper, we present a lightweight MITM detection and defense algorithm for WiFi-enabled IoT gateways. The algorithm works for statically assigned host IP addresses and IP addresses assigned via DHCP. We adopt Asynchronous Method Dispatch (AMD) in the detection and prevention of MITM attack so as to reduce performance overhead. In addition, we evaluated the performance of the algorithm with respect to CPU utilization efficiency, detection rate and network latency. The rest of the paper is organized as follows: Section 2 reviews related literature. Section 3 describes the methodology. Section 4 discusses and presents results from testing the proposed algorithm. Section 5 is the conclusion.

2. Related Works

Researchers have developed a number of conventional ARP spoofing MITM detection mechanisms. [8] introduced low-cost embedded IDS capable of detecting and preventing ARP spoofing attacks automatically and efficiently but it is required to be plugged into a switch or hub. This targets wired LAN environments hence not applicable in a wireless IoT scenario.

A unicast ARP request was proposed instead of the broadcast ARP request by assigning IP addresses via DHCP [9]. They assume that the DHCP will resolve the IP/MAC mapping without the need for broadcast. This approach is not applicable for statically assigned IP addresses.

A backward compatible extension to ARP that relies on public-key cryptography to authenticate ARP replies was proposed [10]. All hosts create public and private key pairs during the initial contact with the network, and send them with signed certificates to the Authoritative Key Distributor (AKD). This technique leads to performance overhead and it is also not feasible in a wireless network.

A Ticket-based ARP (TARP) that implements security by distributing centrally issued secure IP/MAC ticket via DHCP was proposed [11]. These tickets are sent to the clients when they join the network and are subsequently distributed through existing ARP messages. It leads to performance overhead in generating the public/private key pairs and it is not suitable for dynamic networks where hosts can join and leave the network anytime.

An approach to prevent ARP cache poisoning in wireless LAN by implementing the defense mechanism in the access point (AP) was proposed [12]. The AP constructs the list of correct IP-to-MAC address mappings by monitoring DHCP ACK messages or referring to the DHCP leases file, and blocks all the ARP packet with a false mapping based on the constructed list.

MR-ARP, which is the first voting-based ARP spoofing resistant protocol was proposed [13]. When the MR-ARP machine receives an ARP request or reply declaring an (IP, MAC) mapping for a new IP address, it requests neighbor machines to vote for the new IP address. For this mechanism, the voting can be fair only when the voting traffic rates of the responding machines are almost the same. This condition can be satisfied in the Ethernet, but may not be valid in the 802.11 network due to the traffic rate adaptation based on the signal-to-noise ratio (SNR).

To overcome the limitation of MR-ARP, EMR-ARP was proposed [14]. The new protocol improves the voting procedure through the incorporation of computational puzzles. This mechanism requires too much computational time from devices.

GMR-ARP which is an improvement over EMR-ARP was proposed [15]. It decreased the voting traffic overhead (lower than MR-ARP and EMR-ARP). Since voting requests are used in broadcast, this approach can also cause additional overhead.

From the related works, it can be inferred that the conventional approach used in detecting MITM attacks are not applicable to IoT devices due to resource constraints; hence the need for a lightweight MITM detection algorithm.

3. Methodology

The MITM detection and defense algorithm comprises of three sub-level processes coordinated by an interprocess controller (shown in Figure 3). The three sub-level processes consist of the *packet analyzer*, *the detection subprocess* and the *defense subprocess*. Asynchronous Method Dispatch (AMD) is used in the interprocess communication.

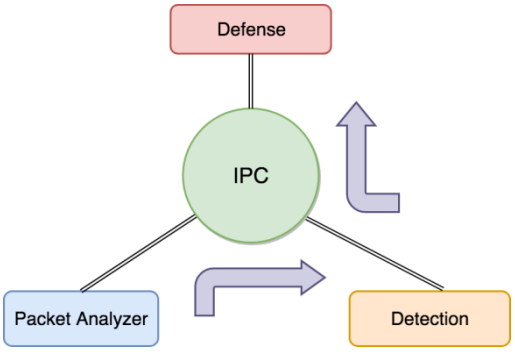

Figure 3. MTIM Attack

The algorithm works by first detecting (*IP*, *MAC*) mappings of the client nodes connected to the gateway.

Legitimate (*IP*, *MAC*) mappings are added to the ARP cache of the gateway which in this case is the AP.

3.1. Packet Analyzer

The packet analyzer subprocess is responsible for capturing and decoding wireless traffic. The algorithm for packet decoding is shown in Table 1. The following packets were captured and analyzed:

- EAPOL/EAP (Extensible Authentication Protocol); shown in Figure 4.
- DHCP (shown in Figure 5).
- IP (shown in Figure 6).
- ARP (shown in Figure 7).

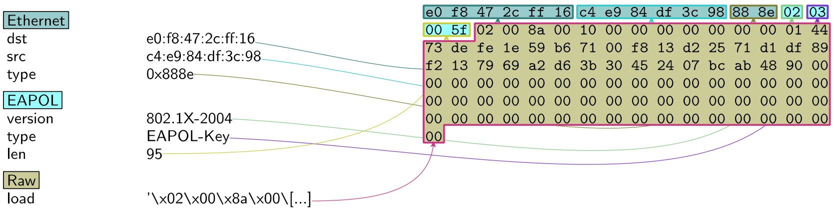

Figure 4. EAPOL Decoded Frame

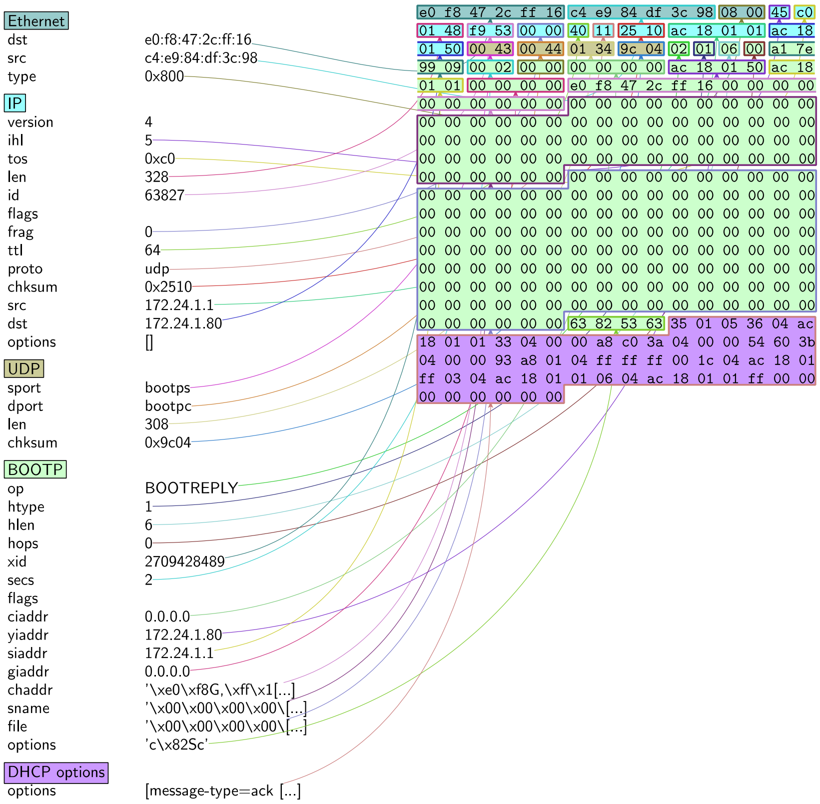

Figure 5. DHCP Decoded Frame

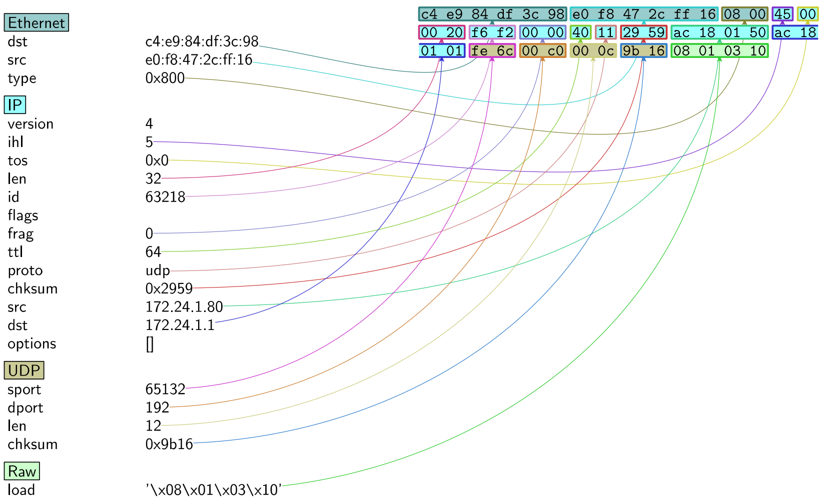

Figure 6. IP Decoded Frame

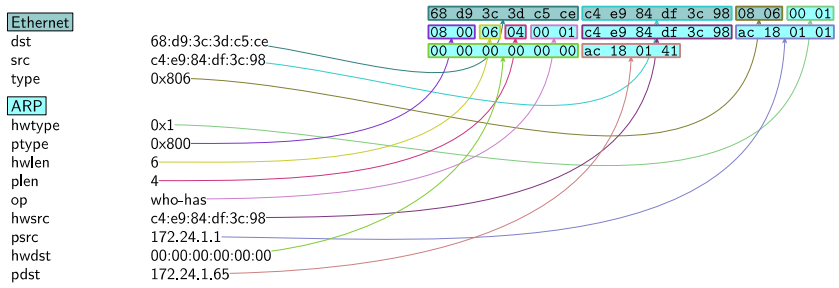

Figure 7. ARP Decoded Frame

Table 1. Packet Decoding Algorithm

Algorithm 1 Packet Decoding Algorithm

```
1:  while True do
2:      if sniff(interface) then
3:          if packet.hasLayer(EAPOL) then
4:              ← IP, MAC, Time-Seen
5:          else if packet.hasLayer(DHCP) then
6:              ← IP, MAC, Time-Seen
7:          else if packet.hasLayer(IP) then
8:              ← IP, MAC, Time-Seen
9:          else if packet.hasLayer(ARP) then
10:             ← IP, MAC, Time-Seen
11:         end if
12:     end if
13: end while
```

3.2. Detection

The detection subprocess keeps virtual ARP cache entries of legitimate IP and MAC mappings and the last time seen. When a new ARP reply is received, the detection subprocess checks the virtual ARP cache to verify if such (*IP*, *MAC*) entry exists. If the entry does not exist, the new (*IP*, *MAC*) mapping is added to the virtual ARP cache and also to the ARP cache of the gateway.

If the received (*IP*, *MAC*) of an ARP reply contains the same IP and MAC address as one of the entries in the virtual ARP cache, the time seen value of that entry is updated. If the MAC address of an ARP reply is the same as one of the entries in the virtual ARP cache but the IP address varies, the detection subprocess performs an inverse ARP to determine whether the host which previously had the associated MAC address is alive. If the host is alive, then it is flagged as MITM attack. If the host is not alive, two tests are performed. The first test is to determine the number of hop counts by performing a traceroute to the host IP address. If the hop count is greater than 1, that means the traffic is being intercepted by an unauthorized client. This is flagged as an MITM attack. If the hop count is 0, then it is possible the host has been denied of service. A second test is performed to validate whether this is MITM attack. The time difference between the last seen MAC address entry in the virtual ARP cache and the incoming ARP reply's time is computed. The incoming ARP reply is flagged as MITM attack if the resulting time difference is less than the ARP cache entry's time-to-live (TTL). The detection algorithm is shown in Table 2.

3.3. Defense

When a particular ARP reply is flagged as MITM attack, the defense subprocess deletes the IP address of the host performing the spoofing attack from the ARP entry and blocks all traffic originating from that host. The defense algorithm is shown in Table 3.

The algorithm was implemented on a Raspberry Pi [17] acting as a gateway for seven IoT devices (NodeMcu [18]), with one of the nodes acting as a malicious client (shown in Figure 8).

Table 2. Detection Algorithm

Algorithm 2 Detection Algorithm

```
1:  if arpData then
2:      exists, res = arp.findMac(arpData.mac)
3:      if exists then
4:          if arpData.ip != res.ip then
5:              if InvARP(res.ip) then
6:                  ← Host alive, MITM detected
7:              else
8:                  if hopCount(res.ip) != 1 then
9:                      tDiff = (arpData.t - res.t)
10:                     if tDiff < ARPTTL then
11:                         ← MITM leading to DoS
12:                     end if
13:                 end if
14:             end if
15:         else
16:             ← Update Last Seen
17:         end if
18:     else
19:         ← New ARP Entry
20:     end if
21: end if
```

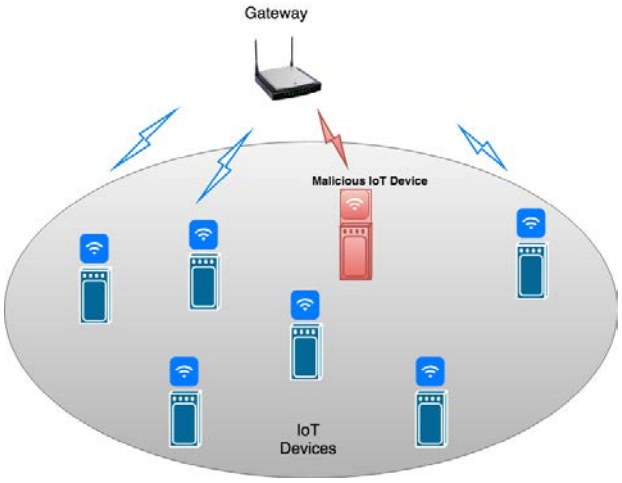

Figure 8. Architecture

Table 3. Defense Algorithm

Algorithm 3 Defense Algorithm

```
1: if mitmData then
2:     deleteARPEntry(mitmData.IP)
3:     dropPackets(mitmData.IP)
4: end if
```

4. Results and Discussion

Three key performance indicators were used in determining the efficiency of the proposed algorithm: CPU utilization efficiency, detection rate and network latency (using the round-trip time (RTT)).

Figure 9 shows the performance overhead when the algorithm was implemented; averages **0.9545%**. It outperforms that of [12] which was **1.65%**.

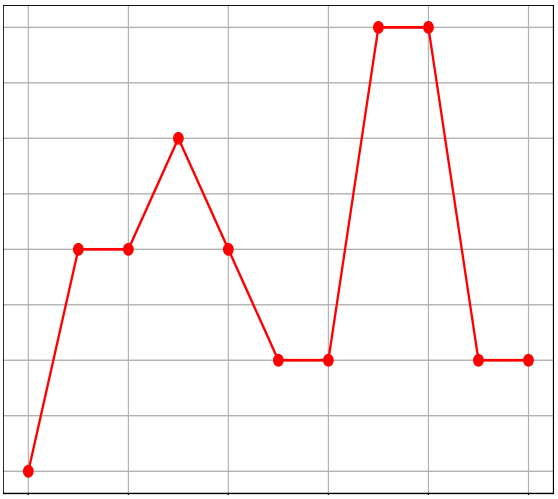

Figure 9. Performance Overhead

The average time (shown in Figure 10) for detecting MITM attack is **0.1686 seconds**.

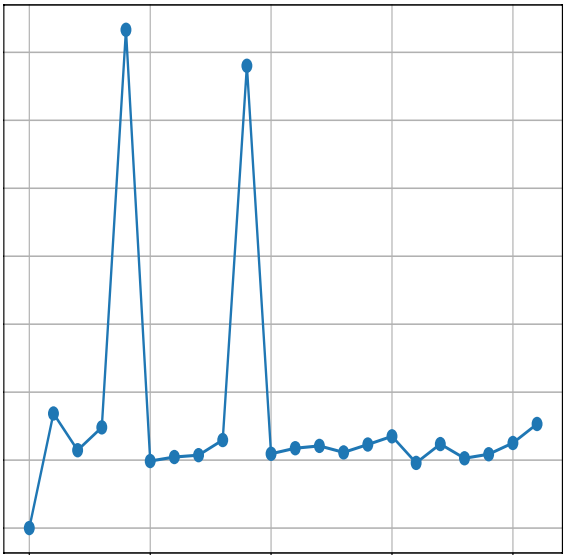

Figure 10. Detection Rate

The round-trip time, which is a measure of latency in a network, for the instance where the algorithm was not implemented is **1.298 seconds** whereas when the algorithm was implemented it is **1.335 seconds**. This shows that the MITM detection and defense algorithm does not significantly affect the latency of the network (shown in Figure 11).

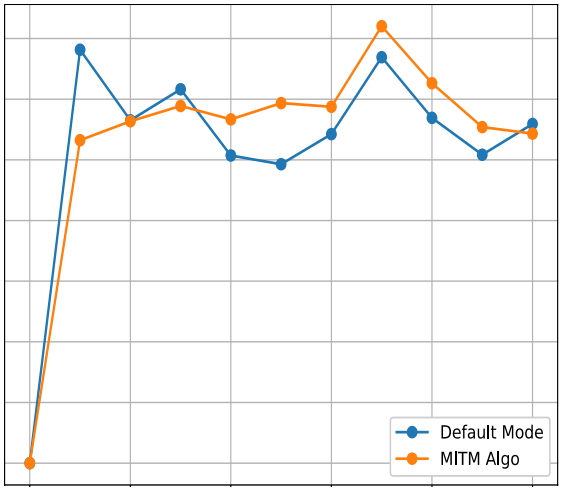

Figure 11. Round-Trip Time

5. Conclusion

In this paper, we have presented the weakness in the ARP protocol and proposed a lightweight MITM detection and defense mechanism that can be implemented on WiFi-enabled IoT gateway devices. We have also demonstrated the efficiency of the proposed algorithm with respect to performance overhead, detection rate and network latency. This will ensure data confidentiality and integrity in an IoT environment. The entire source code for the research can be found at [19].

References

[1] Dave Evans, The Internet of Things: How the Next Evolution of the Internet is Changing Everything, Cisco Internet Business Solutions Group (IBSG), pp. 3, April 2011.

[2] Garcia-Morchon O. Kumar S., Struik R., Keoh S., Hummen R., Security considerations in the IP-based Internet of Things, IETF Internet-Draft, 2013.

[3] Notra S., Siddiqi M., Gharakheili H., Sivaraman V., Boreli R., An Experimental Study of Security and Privacy Risks with Emerging Household Appliances, In: Communications and Network Security (CNS), 2014 IEEE Conference on, pp. 79-84, 2014.

[4] Kolias C., Stavrou A., Voas J., Bojanova I., Kuhn R., Learning Internet -of-things Security "Hands-on". IEEE Secur. Priv. 20 (February), 2-11.

[5] Plummer, D. C. (1982), An Ethernet Address Resolution Protocol. RFC 826.

[6] AI Sukkar G., Saifan R., Khwaldeh S., Maqableh M., Jafar L., Address Resolution Protocol (ARP); Spoofing Attack and Proposed Defense, Communications and Network, 8, 118-130, 2016.

[7] Mauro Conti, Nicola Dragoni, Viktor Lesyk, A Survey of Man In the Middle Attacks, IEEE Communications Surveys & Tutorials, Vol. 18, No. 3, 2016.

[8] J. Belenguer, C. T. Calafate, A low-cost Embedded IDS to Monitor and Prevent Man-In-The-Middle Attacks on Wired LAN Environments, Proc. Int. Conf. SecureWave Emerging Secur. Inf. Sys. Technol., 2007, pp. 122-127.

[9] Isaac B., Secure ARP and Secure DHCP Protocols to Mitigate Security Attacks, International Journal of Network Security, 8, 107-118, 2009.

[10] D. Bruschi, A. Ornaghi, E. Rosti, S-ARP: A Secure Address Resolution Protocol, Proc. 19th Annu. Comput. Secur. Appl. Conf., pp. 66-74, 2003.

[11] Lootah W., Enck W., McDaniel P., TARP: Ticket-Based Address Resolution Protocol, Computer Networks, 51, 4322-4337, 2007.

[12] R. Philip, Securing Wireless Networks from ARP Cache Poisoning, Master's Thesis, San Jose State University, California, 2007.

[13] S. Y. Nam, D. Kim, J. Kim, Enhanced ARP: Preventing ARP Poisoning-Based Man-In-The-Middle Attacks, IEEE Commun. Lett., vol. 14, No. 2, pp. 187-189, 2010.

[14] S. Y. Nam, S. Jurayev, S.-S. Kim, K. Choi, G. S. Choi, Mitigating ARP Poisoning-Based Man-In-The-Middle Attacks in Wired or Wirelesss LAN, EURASIP Journal on Wireless Communications and Networking. 2012.

[15] S. Y. Nam, S. Djuraev, M. Park, Collaborative Approach to Mitigate ARP Poisoning-Based Man-In-The-Middle Attack, Comput. Netw. Vol 58, No. 18, pp 3866-3884, 2013.

[16] Bruno Bogaz Zarpelao, Rodrigo Sanches Miani, Caludio Toshio Kawakani, Sean Carlisto de Alvarenga, A Survey of Intrusion Detection in Internet of Things, Journal of Network and Computer Applications, pp 2-4, 2017.

[17] Raspberry Pi, https://www.raspberrypi.org/products/raspberry-pi-3-model-b/, [Accessed Dec 11, 2018].

[18] NodeMcu, http://www.nodemcu.com/index_en.html, [Accessed Dec 22, 2018].

[19] IoT IDS, https://github.com/jayluxferro/IoT-IDS/.