Journal of Artificial Intelligence and Data Science (JAIDA)
e-ISSN: 2791-8335
https://dergipark.org.tr/pub/jaida

Vol. 3, No. 2, pp. 99-106, 2023
Research Article
Received: 10/09/2023 Accepted: 01/12/2023

MitM Attacks and IoT Security: A Case Study on MQTT[1]

Serhat ÇELİK[1], Nesibe YALÇIN[2,*], Semih ÇAKIR[3]

Abstract

The number of devices connected to the Internet has increased with the development of Internet of Things (IoT) technologies. It is foreseen that this situation will increase daily, and the concept of the IoT will become more popular. However, security vulnerabilities in IoT devices have not been eliminated, and these devices are vulnerable to attacks because their resource-limited features increase security concerns. The security problem of the Message Queuing Telemetry Transport (MQTT) protocol, which is widely used in the IoT field, is of great importance. In this study, a smart-home system application that provides communication between devices using the MQTT protocol has been developed. A Man in the Middle (MitM) attack, which is one of the first attacks that come to mind when it comes to privacy violation, was carried out, targeting data packets between users with a temperature sensor used in the application.

Keywords: Attack Detection; IoT; IoT Security; MitM Attacks; MQTT.

1. Introduction

IoT is a technology that allows any "thing" that can connect to the internet so that things communicate (send/receive data) and work synchronously with other things [1]. With this definition, it would be incorrect to limit IoT devices to appliances, such as smart televisions or smartwatches. Printers, refrigerators, washing machines, air conditioners, home heating systems, and many similar or different devices can be defined as IoT devices. With the development of technologies such as IoT, the use of the Internet is increasing; however, the security of data sent and received via the Internet still poses a fundamental problem. According to analysts, the use of IoT technologies is expected to increase gradually, but owing to the security vulnerabilities of IoT devices, many companies still worry and take extra caution when investing in this field. The use of IoT poses security challenges [2]. Devices must communicate only with users that belong to them. Most devices do not have an authentication interface to enter a username and password. However, authorization must be achieved. Every device in an IoT environment should have guaranteed security. These devices must be capable of remote updates. The device and updates must be compatible. Furthermore, version control should be performed to prevent the device from being downgraded from its current version to older versions.

Information security, including IoT security, includes three main terms: confidentiality, integrity, and availability [3]. Confidentiality aims to ensure that information is accessible only to authorized persons. Because information is processed, stored, and shared, it must be protected. Integrity tries to prevent data corruption. These corruptions can occur by changing the information, adding to the information, and deleting some or all of its content. The goal of availability is to make information accessible to users anytime and for as long as they want. As an illustration of a situation in which some of these security terms are violated: have a baby camera for parent usage. This device can be a simple live tracking camera or a face analysis camera that uses artificial intelligence and image processing techniques. It is assumed that the login information of the home Internet used by this device for communication is captured by a malicious user. If the attackers are on the same network as the devices they intend to attack, they can execute a Man in the Middle (MitM) attack. In this way, an attacker can view or alter important data in network traffic. Therefore, confidentiality and integrity are violated.

*Corresponding author

SERHAT ÇELİK; Erciyes University, Faculty of Engineering, Department of Computer Engineering, Türkiye; e-mail: 1030510344@erciyes.edu.tr; 0000-0002-4717-1507

NESİBE YALÇIN; Erciyes University, Faculty of Engineering, Department of Computer Engineering, Türkiye; e-mail: nesibeyalcin@erciyes.edu.tr; 0000-0003-0324-9111

SEMİH ÇAKIR; Zonguldak Bülent Ecevit University, Karadeniz Ereğli Vocational School of Higher Education, Information Technology Security Program, Türkiye; e-mail: semih.cakir@beun.edu.tr; 0000-0003-3072-9532

[1]An earlier version of this paper was presented at the ICADA 2023 Conference and was published in its Abstract Book (Title of the conference paper: "MQTT Protokolü Tabanlı IoT Sistemine Yönelik Saldırı Uygulaması ve Güvenlik").

Many studies have been conducted to examine and secure IoT technologies. This research focuses on the security of the Message Queuing Telemetry Transport (MQTT) protocol used in various IoT systems. In particular, MQTT contains a limited number of security elements [4]. A new design for the MitM attack against IoT devices using MQTT is presented in [5]. Experiments have been conducted in a test environment containing Mosquitto, Raspberry Pi, Pineapple, Wi-Fi, etc. The designed attack has successfully evaded various machine learning-based intrusion detection models. The study shows how difficult it is for MiTM attacks to be detected by traditional security defense mechanisms. In [6], the K-Nearest Neighbor approach has been adopted for attack detection in MQTT-based IoT systems. A basic smart home system has been developed in that study. Attacks have been conducted on a machine running Kali Linux and Wireshark has been used for packet listening on a machine with Windows installed. A switch has been utilized to interact with the device. An access point has been used for the communication of IoT devices, and a Raspberry Pi for control. Three different attacks have been performed to test the security of MQTT and security measures have been recommended by Şimşek and Atılgan [7] for systems using the MQTT protocol. In [8], a lightweight authentication system has been proposed to provide secure MQTT communication between IoT devices. The energy consumption and communication overhead have been analyzed using the Cooja simulator for optimal performance. A machine learning approach has been presented for an MQTT-based IoT platform in [9]. An attack is first carried out on the MQTT server of a smart-home network. MitM attacks are added when an attack is observed on the Wireshark. The post-attack data and MQTT dataset have been combined in a comma-separated value format. Training and test implementations have been performed separately, and the results are recorded as accuracy, F1 score, and training and test times. The authors in [10] have studied On-Off Attack (OOA) detection for IoT devices in a Contiki–Cooja simulation environment. Multilayer perceptron neural networks have been applied to improve the detection of OOA-based attacks. A real network environment has been created to record the on/off status of each node. Incoming radio messages have been captured simultaneously, and unstable conditions have been controlled. Varma and UniKrishnan [11] have carried out some experiments to evaluate the payload of MQTT when exposed to a MitM attack. ARP spoofing (via MitM attack) has been successfully conducted in the study. However, sniffing has failed because the payload has been secured using the AES encryption algorithm. To make the MQTT protocol more secure, it has been suggested that object-level security rather than Transport Layer Security (TLS) should be considered.

This study addresses the importance of MQTT security for IoT systems. The security requirements of IoT devices using MQTT have been discussed and the attack surface in IoT has been examined over the local network. We have also developed an IoT-based testbed with a particular focus on MitM attacks for MQTT security. So IoT security awareness can be improved specifically for MQTT. The next section presents general information about the MQTT protocol, and Section 3 provides details of the design and implementation of the testbed. The ARP-spoofing-based MitM attack and an example scenario are presented in Section 4. The results of the attack are discussed in Section 5, and the security risks and measures of MQTT are evaluated in the final section.

2. MQTT Protocol

The MQTT protocol [12] is a messaging protocol used for machine-to-machine communication. IoT is widely utilized [4] as the application layer protocol in data exchange [13]. The main reasons why they are preferred by a multitude of IoT devices [14], [15] are as follows.

- Resource efficiency: IoT devices do not consume a large amount of resources owing to their nature. Because the MQTT protocol is also lightweight, it is suitable for use with IoT devices. For example, an MQTT message can be as small as two bytes.
- Dynamic scalability: The coding difficulty required for MQTT is minimal. In addition, built-in features support IoT devices.
- Security: This allows message authentication with options such as OAuth, TLS, and personal certificates.
- Support: Many languages such as Python offer extensive library support for the MQTT protocol.

2.1. MQTT architecture

A publish/subscribe communication method is employed in the MQTT architecture. This publishing device is called a publisher. Publishers transmit data to specific topics. Subscribed devices, known as subscribers, can follow the content of a topic by subscribing to the topic. Figure 1 shows the basic structure of the MQTT architecture.

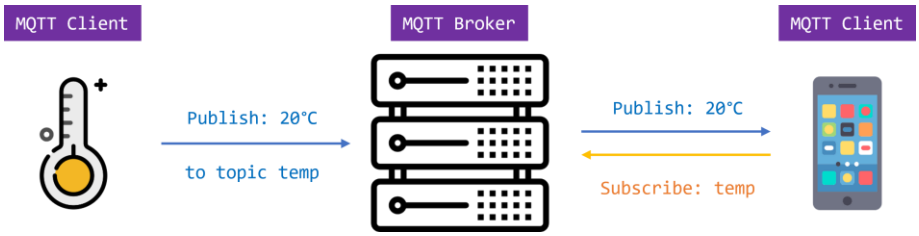


Figure 1. *MQTT architecture.*

The three key components in the architecture are the MQTT client, broker, and connection. MQTT clients can be defined as any device that sends/receives data in the network using the MQTT protocol. If this device sends data, it is referred to as the publisher. If it receives data by subscribing to a topic, it is called a subscriber. The MQTT broker is responsible for managing the messages among different clients. It performs the tasks of authorization, authentication, message filtering, and subscriber identification. Using an MQTT connection, clients communicate with each other. First, any client sends a CONNECT message to the MQTT broker. When a connection is established, the broker sends the CONNACK message to the client. Thus, clients communicate with each other only through the broker. There is no direct connection between the two.

2.2. MQTT broker

Many commercial and open-source MQTT brokers are available on the market. In this study, HiveMQ [16] has been used as the MQTT broker. The public HiveMQ MQTT broker offers some features (100 connectable devices for free, 10 GB data limit, etc.) required to build MQTT applications. This supports the latest MQTT version and fully managed cloud service. In addition, users can be created in this service, and their identities can be determined. User manuals and other documentation are available for multiple development environments including Arduino, NodeMCU, Raspberry Pi, and Python.

3. MQTT-Based IoT Platform

A smart home system has been developed as a testbed for the practical evaluation of IoT security. The software and hardware requirements for the design and implementation of a smart-home system are detailed in this section.

3.1. Software

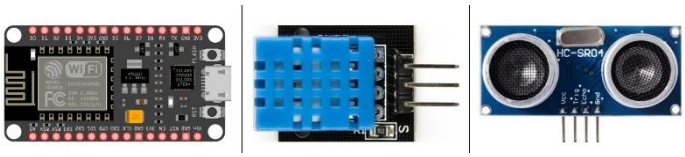
Arduino IDE, Fritzing, Wireshark, hARPy, and ARPspoof are used in the application testbed.

- The code written in C++ using the Arduino IDE is permanently loaded into the NodeMCU device memory.
- Fritzing is a tool for drawing hardware connections.
- The Wireshark software has been used as a network monitoring tool in this study. Wireshark is a free and open-source software for monitoring and listening/sniffing network traffic.
- hARPy is a tool developed in Python for active and passive Address Resolution Protocol (ARP) scanning. In this manner, the Internet Protocol (IP) addresses of the devices in the current network and their corresponding Media Access Control (MAC) addresses can be learned.
- ARPspoof is a tool for performing MitM attacks using the ARP poisoning technique in Python.

3.2. Hardware

The NodeMCU development kit, DHT11 humidity and temperature sensor, and HC-SR04 ultrasonic distance sensor shown in Figure 2 are used to develop the IoT platform.

- NodeMCU V3 ESP8266 ESP-12F Development Kit: Unlike Arduino devices, the NodeMCU is a development kit that allows developers to connect to the Internet with the built-in ESP8266 WiFi module. In addition, it is preferred, particularly in IoT applications, owing to its lower power consumption.
- DHT11: A Digital Humidity and Temperature (DHT) sensor is used to obtain the temperature information of the environment and is ideal for long-term use. The temperature measurement range is between 0-50 °C degrees, and the measurement accuracy is acceptable [17].
- HC-SR04 Ultrasonic Distance Sensor: It is utilized to measure the distance of the target with the ultrasonic sound waves it emits. With a minimum range of 2 cm and a maximum range of 400 cm, this sensor can measure with an accuracy of 3 mm [18].



(a) (b) (c)

Figure 2. *(a) NodeMCU, (b) DHT11, (c) HC-SR04.*

3.3. Application Testbed

To prepare the IoT environment, first, the access point setting is made so that the devices can communicate with each other via an Internet connection. The IP address of the default gateway is set at 192.168.43.1. The ESP8266 module is set as the publisher to share the information it receives from the temperature sensor and ultrasonic distance sensor with the MQTT protocol, and its IP address is determined as 192.168.43.229. The IP address of the computer with the Lubuntu Linux operating system used for the MitM attack is 192.168.43.10. This computer was also used to monitor the network traffic during an attack using the Wireshark tool. Figure 3 shows a representative view of the IoT platform obtained as a result of these adjustments.

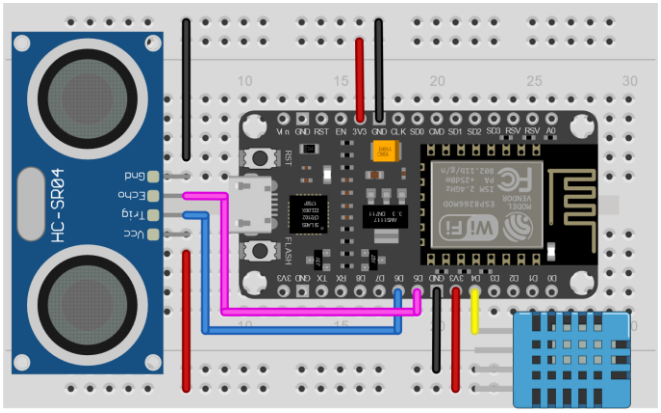


Figure 3. *Fritzing sketch of hardware connections.*

4. ARP Spoofing Based MitM Attack

A publisher/subscriber model is used by MQTT to enable messaging between devices. However, its protocol design has many security risks such as denial of service, poisoning, sniffing, tampering, MitM, elevation of privileges, and spam [4], [8]. A MitM attack is a cyberattack type that is possible if malicious users are on the same network as the devices they will attack. Multiple techniques can be used to carry out these attacks. ARP poisoning is one of these techniques. This section contains details on how the ARP works, the ARP poisoning technique, and how this technique is applied.

4.1. ARP

Before devices can communicate with other devices on the same network, they require their MAC addresses as well as their IP addresses. To accomplish this, the current device sends a who-is (request) packet to the other device that it wants to communicate with [19]. For example, if the device asks, “I am 192.168.0.1 and my MAC address is AA-BB-AA-11-11-11, what is the MAC address of the device 192.168.0.5?”. After this request, the device that receives the message sends an is-at (response) packet. As shown in Figure 4, the device says “I am 192.168.0.5, and my MAC address is AA-BB-AA-55-55-55.

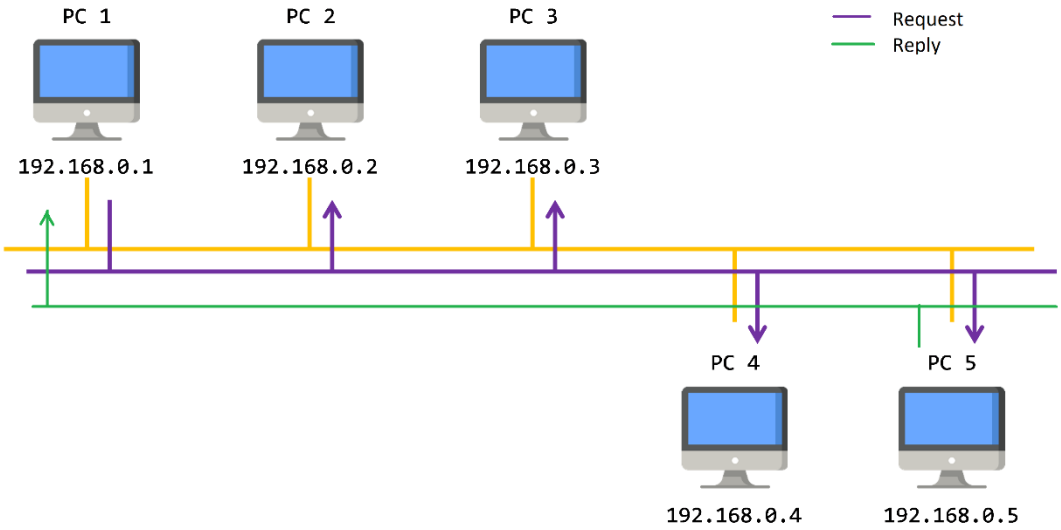


Figure 4. *ARP communication process.*

The process normally occurs in this example according to the definitions in the protocol. However, ARP is a vulnerable protocol, based on its nature. Because during the sending of these packets, it is not verified whether the incoming packets come from the original owner. Thus, ARP poisoning can be performed using this vulnerability.

4.2. ARP Poisoning

To perform MitM attacks with ARP poisoning [20]

a) The attacker sends a request packet to the victim device, meaning "I am the modem" as if the modem's MAC address is its own.
b) The attacker sends a response packet to the modem that shows the MAC address of the victim device as if it is its own MAC address, meaning "I am a trusted device."
c) Thus, the attacker becomes a man in the middle between the modem and the victim, and the traffic between these two devices flows over itself.

4.3. Attack Scenario

Before performing the attack, some setups are required on the Lubuntu Linux device used as the attack machine. The ARP discovery tool, hARPy, should be installed in the system to learn more about the devices connected to the network. To collect more detailed information about the network after an attack, Wireshark should be installed on the system. In addition, for the attack to be successful, the packets must be able to flow through the attack machine. IP forwarding must be implemented for this purpose. Following the completion of these preparations, a network scan is performed with hARPy for the MAC address information of the NodeMCU publisher that will be targeted by the attack. The data obtained as a result of the scanning is listed in Table 1.

Table 1. *IP and MAC address information of the devices before the attack.*

Address	NodeMCU	Access Point	Lubuntu Linux
IP	192.168.43.229	192.168.43.1	192.168.43.10
MAC	F4-CF-A2-XX-XX-XX	D8-5B-2A-XX-XX-XX	C0-4A-00-XX-XX-XX

ARPspoof has been used to perform the attack [21]. After this attack, it is aimed to get the MQTT topic, which contains important data. As with the network used in this application, the MQTT topic can be learned if the traffic is unencrypted. Figure 5 depicts the attacker's position on the IoT platform in the case of a successful attack.

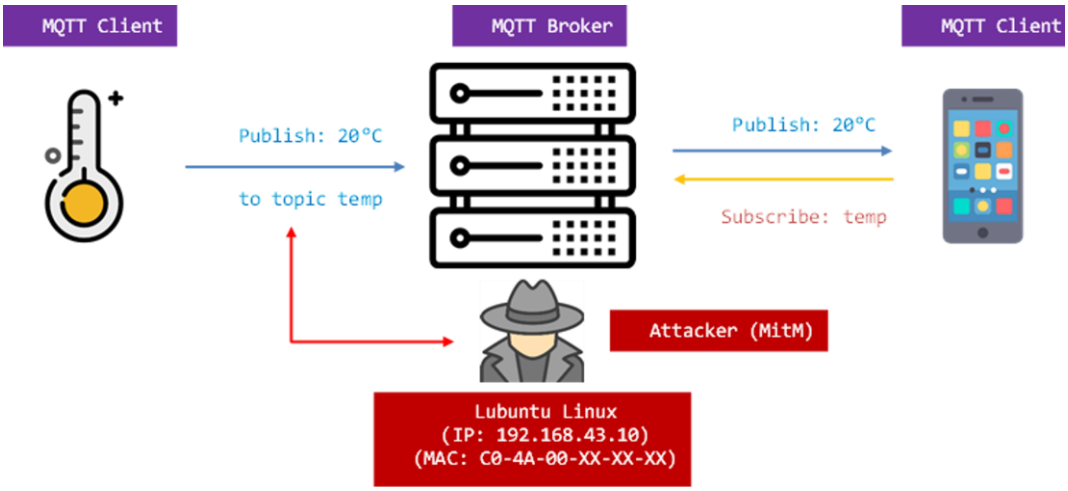


Figure 5. *The position of the attacker in the MitM attack.*

5. Results and Discussion

Table 2 presents the changes in device information after the attack. The MAC address of the Lubuntu Linux attack device remained the same, the MAC addresses of the NodeMCU publisher device and the access point are the same as the MAC address of the Lubuntu Linux device.

Table 2. *IP and MAC address information of the devices after the attack.*

Address	NodeMCU	Access Point	Lubuntu Linux
IP	192.168.43.229	192.168.43.1	192.168.43.10
MAC	C0-4A-00-XX-XX-XX	C0-4A-00-XX-XX-XX	C0-4A-00-XX-XX-XX

While the attack is in progress, it can be seen that the network traffic can be monitored using Wireshark. Figure 6 shows that the MQTT topic can be discovered by the attacker.

```
▾ MQ Telemetry Transport Protocol, Publish Message
  ▸ [Expert Info (Note/Protocol): Unknown version (missing the CONNECT packet?)]
    ▾ Header Flags: 0x30, Message Type: Publish Message, QoS Level: At most once delivery (Fire and Forget)
        0011 .... = Message Type: Publish Message (3)
        .... 0... = DUP Flag: Not set
        .... .00. = QoS Level: At most once delivery (Fire and Forget) (0)
        .... ...0 = Retain: Not set
    Msg Len: 33
    Topic Length: 24
    Topic: home/kitchen/temperature
    Message: 32362e37303030
```

Figure 6. *MQTT topic obtained by MitM attack.*

The MQTT message can be found following the Transmission Control Protocol flow. For instance, as can be seen in Figure 7, the temperature information is 26.7000 °C.

```
0!..home/kitchen/temperature26.7000!..home/kitchen/temperature26.7000!..home/kitchen/temperature26.7000!..home
/kitchen/temperature26.7000!..home/kitchen/temperature26.7000!..home/kitchen/temperature26.7000!..home/kitchen/
temperature26.7000!..home/kitchen/temperature26.7000!..home/kitchen/temperature26.7000....0!..home/kitchen/
temperature26.7000!..home/kitchen/temperature26.7000!..home/kitchen/temperature26.7000....home/kitchen/
temperature26.7000....0!..home/kitchen/temperature26.7000!..home/kitchen/temperature26.7000!..home/kitchen/
```

Figure 7. *Temperature information obtained by MitM attack.*

Various precautions must be taken in advance to prevent attackers from succeeding with such an ARP attack. Some of these measures are given below:

- All MAC addresses (on a home or enterprise network) can be statically mapped to their original IP addresses. This is quite effective in preventing ARP poisoning attacks but can be very workload-intensive. Because it requires manual updating of ARP tables for all computers after any changes are made to the network.
- ARP messages cannot be processed beyond a local subnetwork. Therefore, focusing important resources on a dedicated network segment that is well-segmented and has enhanced security may make the network less susceptible to ARP cache poisoning.
- Various software has been developed to observe suspicious IP and MAC matches on a network, for example, Arpwatch and X-ARP. These tools constantly monitor the network and can alert administrators if they detect any signs of an ARP attack. However, they can occasionally produce false positive results. Considering this possibility, the settings should be customized if necessary.
- Traffic can be encrypted. Although encryption does not technically prevent an ARP attack, it can reduce the potential damage. Connections encrypted with SSL/TLS will serve to defeat attackers' purposes.

If data is protected by encryption, when an attack occurs the resulting data will appear as encrypted text in the "Encrypted Application Data" section. Figure 8 shows that the data obtained as a result of the attack is encrypted.

Figure 8. *Encrypted data obtained as a result of a MitM attack on a secure network.*

6. Conclusion

This study focuses on the MQTT protocol and a case study has been conducted on the MitM attack. MitM attack is a powerful attack type and can cause serious damage that is hard to detect by typical defense mechanisms. In networks that lack security measures, the confidentiality of data can never be fully assured. Some precautions and counteractions to be taken to protect against this attack, including transport layer and application layer security are given as follows:

- The firewall must be active and updatable.
- Whenever possible, only devices that are on the whitelist should be allowed to join the network.
- By default, the MQTT protocol does not require a connection using a username and password. For this reason, it must be mandatory to establish a connection with a username and password.
- The current, most up-to-date version of the MQTT protocol should be preferred.
- MQTT brokers let clients create their own rules. Some of these rules are: publish only, subscribe only, publish only on a specific topic, and subscribe only on a specific topic. In this way, even if unauthorized users know about the MQTT topic, they will not be able to take any action because they are not authorized.
- Traffic should be encrypted using TLS technology. The HiveMQ service has explained how to implement certificates on NodeMCU devices. To achieve this, Distinguished Encoding Rules (DER) files are created via the Privacy Enhanced Mail (PEM) texts provided by the "Mozilla Trusted Certificate Authorities" page. PEM texts are simply Base64-encoded DER files. A DER file to be obtained in this way can be uploaded to the NodeMCU device, and the connections can be encrypted.

Encrypting payload data is the best option to secure MQTT. TLS must be offered to ensure the integrity of MQTT data and to prevent attacks on MQTT nodes. In addition, machine learning and blockchain technologies present promising solutions to deal with the MQTT security issues.

Declaration of Interest

The authors declare that there is no conflict of interest.

Author Contributions

Serhat Çelik: Conceptualization, Methodology, Software & Hardware of IoT Platform, Validation, Investigation, Results and Discussion, Visualization, and Writing - original draft.

Nesibe Yalçin: Conceptualization, Methodology, Writing - Review - Editing, and Supervision.

Semih Çakir: Methodology, Validation, Investigation, and Review - Editing.

All authors reviewed the manuscript.

References

[1] C. C. Sobin, "A Survey on Architecture, Protocols and Challenges in IoT," Wireless Personal Communications, vol. 112, pp. 1383-1429, 2020. doi: 0.1007/s11277-020-07108-5

[2] O. Yavuz, "Nesnelerin İnterneti (IoT) ve Güvenliği," btkakademi.gov.tr, 2023. [Online]. Available: https://www.btkakademi.gov.tr/portal/course/nesnelerin-interneti-iot-ve-guvenligi-10625. [Accessed July 2, 2023].

[3] M. B. Younes and N. N. El-Emam, "Information Security and Data Management for IoT Smart Healthcare," In Intelligent Internet of Things for Smart Healthcare Systems, CRC Press, pp. 69-80, 2023.

[4] A. J. Hintaw, S. Manickam, M. F. Aboalmaaly, and S. Karuppayah, "MQTT Vulnerabilities, Attack Vectors and Solutions in the Internet of Things (IoT)", IETE Journal of Research, vol. 69, no. 6, pp. 3368-3397, 2023. doi: 10.1080/03772063.2021.1912651

[5] H. Wong, L. Tuo, "Man-in-the-Middle Attacks on MQTT-based IoT Using BERTBased Adversarial Message Generation", KDD'20 Workshops: the 3rd International Workshop on Artificial Intelligence of Things (AIoT), 2020, San Diego, CA.

[6] B. Erdem and O. Yaman, "KNN Based Intrusion Detection Method for IoT Applications Using MQTT Protocol," Fırat University Journal of Science and Technology, vol. 1, no. 1, pp. 225-229, 2022.

[7] M. M. Şimşek and E. Atılgan, "Attacks on Availability of IoT Middleware Protocols: A Case Study on MQTT", Eskişehir Türk Dünyası Uygulama ve Araştırma Merkezi Bilişim Dergisi, vol. 4, no. 2, pp. 16-27, 2023. doi:10.53608/estudambilisim.1297052

[8] S. Tian, V. G. Vassilakis, "On the Efficiency of a Lightweight Authentication and Privacy Preservation Scheme for MQTT", Electronics, vol. 12, no. 14, 3085, 2023. doi: 10.3390/electronics12143085

[9] A. N. Kaya and E. N. Yolaçan, "Attacks nn The MQTT-Based IoT System Detection Using Machine Learning," Journal of Engineering and Architecture Faculty of Eskişehir Osmangazi University, vol. 30, no. 2, pp. 159-170, 2022.

[10] A. H. Farea and K. Küçük, "Enhancement Trust Management in IoT to Detect ON-OFF Attacks with Cooja," International Journal of Multidisciplinary Studies and Innovative Technologies, vol. 5, no. 2, pp. 123-128, 2021.

[11] A. Varma and S. UniKrishnan, "Effect of Payload Security in MQTT Protocol Over Transport and Application Layer", IOP Conference Series: Materials Science and Engineering, vol. 1166, 012019, 2021. doi:10.1088/1757-899X/1166/1/012019

[12] MQTT, "MQTT: The Standard for IoT Messaging," mqtt.org, 2022. [Online]. Available: https://mqtt.org. [Accessed July 2, 2023].

[13] F. Chen, Y. Huo, J. Zhu, and D. Fan, "A Review on the Study on MQTT Security Challenge," 2020 IEEE International Conference on Smart Cloud (SmartCloud), Washington, DC, USA, 2020, pp. 128-133, doi: 10.1109/SmartCloud49737.2020.00032.

[14] M. Bender, E. Kirdan, M. -O. Pahl, G. Carle, "Open-Source MQTT Evaluation," 2021 IEEE 18th Annual Consumer Communications & Networking Conference (CCNC), Las Vegas, NV, USA, pp. 1-4, 2021, doi: 10.1109/CCNC49032.2021.9369499.

[15] Amazon, "MQTT Protokolü Neden Önemli?," amazon.com, 2023. [Online]. Available: https://aws.amazon.com/tr/what-is/mqtt/. [Accessed July 22, 2023].

[16] HiveMQ, "HiveMQ | Public Broker | MQTT Dashboard," hivemq.com, 2023. [Online]. Available: https://broker.hivemq.com/. [Accessed July 22, 2023].

[17] ArduinoModules, "KY-015 Temperature and Humidity Sensor Module," arduinomodules.info, 2021. [Online]. Available: https://arduinomodules.info/ky-015-temperature-humidity-sensor-module/. [Accessed July 25, 2023].

[18] F. T. Akgul, "Ultrasonik (Ultrasonic) Sensör Nedir? Nasıl Çalışır?," robotistan.com, 2021. [Online]. Available: https://maker.robotistan.com/ultrasonic-sensor/. [Accessed July 25, 2023].

[19] IPCisco, "Address Resolution Protocol (ARP)," 2020. [Online]. Available: https://ipcisco.com/lesson/address-resolution-protocol-arp/. [Accessed Aug. 1, 2023].

[20] rauf, "[TR] ARP Nedir ve ARP Spoofing Nasıl Yapılır?," 2021. [Online]. Available: https://pwnlab.me/tr-arp-nedir-ve-arp-spoofing-nasil-yapilir/. [Accessed Aug. 2, 2023].

[21] M. Brown, "System Setup and Scripts For Various MitM Activities," 2022. [Online]. Available: https://github.com/nmatt0/mitmtools. [Accessed Aug. 2, 2023].