Effect of payload security in MQTT protocol over transport and application layer

Arunima Varma* and Srija UniKrishnan

Department of Electronics Engineering, Fr. Conceicao Rodrigues College of Engineering, Fr. Agnel Ashram, Bandstand, Bandra (W), Mumbai – 400050, Affiliated to University of Mumbai, India

E-mail: arunimavarma20@gmail.com

Abstract. Digital revolution has made our life dependent on devices that are can be connected to the internet. Hence, making Internet of things (IoT) an inevitable part for now and years to come. The transition from normal devices to those devices that can be connected to internet, has been exponential, in contrast, the technique used by the protocols in those devices to prevent cyber-attacks and keep user data intact has not been in the same pace as that of its usage. This concerns the end user as it risks the data, allowing it to be misused. In this paper we discuss, MQTT (message queue telemetry transport) a lightweight messaging protocol, wherein the payload is acting as the driver for the data to be carried. This payload can include all sort of information, private as well as public. The paper experiments with the vulnerability of the payload when exposed to a cyber-attack, here, Man in the middle (MITM) attack when the same payload is provided with 2 different types of security. Initially at the transport layer and later at the application layer. At first a channel-based security using Transport security layer (TLS) is provided to the payload and later, an object-based security using Advanced Encryption Scheme (AES) is provided to the same payload. Both of these payload encryption technique for MQTT protocol is discussed elaborately.

1. INTRODUCTION

India has gone from providing mobile in every hand, to wifi/ internet in every place. Clearly, this is also what the India post pandemic aims to offer. This concludes that not only mobile will facilitate more than what is expected but also internet will come in handy to devices that can be connected to the internet. Giving an open opportunity to Internet of Things and its potential. More devices can be connected to the internet and can be controlled with or without mobile phone. A plethora of available IoT protocols, can be utilized for the same. The application’s purpose conjunct with the correct protocol within IoT, will not only provide data security but will also ensure smooth functioning of the application end to end.

[1]
Sensitive information, always exists on network and as we can see, that we are now and henceforth, going to greatly rely on computer network to share information. It is no different for Internet of things as shown in figure 1. Assuming, each and every one will have connectivity to internet 24x7, it risks the Iot protocols, that are sending sensitive data to and fro. There is chance for the data being transferred from one Iot device to another Iot device to be at risk for any cyber-attack. For safeguarding sensitive data between two systems, that could potentially include personal information, we make use of secure socket layer (SSL). SSL is essentially a transport layer security of the OSI model, that makes use of encryption algorithms that will make hackers difficult to decrypt whether it is between two users or two websites or a user and a website. Transport layer security, updated version of SSL, also present at the transport layer, makes use of encryption techniques such as ECC (elliptic curve cryptography), RSA (Rivest–Shamir–Adleman) etc. These act as digital certificate to provide security between two systems at transport layer. [1] [2]

Protocol stack for Iot, visualization can be seen as an extended version of the TCP/IP protocol model and encompasses 6 layers namely physical layer, which is the primary one, followed by the link layer, which is the second layer, followed by the network layer, then later it has, transport layer. Last layer is an amalgamation of application layer and application services layer. One of the Iot application protocol is MQTT (message queue telemetry transport) a lightweight messaging protocol, that is responsible for M2M (machine to machine) communication. MQTT uses low power and bandwidth, based on subscriber-broker-publisher model and is extraordinarily light. This gives an extra edge for MQTT over other Iot protocols. [3] [4]
MQTT is an application layer protocol, message handover is promised by the elementary layer below it which is the transmission protocol layer (TCP/IP). The Quality of service (QoS) setting which is present for MQTT, has the functionality to deliver a message At Most Once, At Least Once or Exactly Once. However, conveying message handover does not ensure accuracy and consistency of data, seclusion or originality of the message. In order to ensure these considerations, a security layer must be implemented on top of the TCP/IP layer. To provide extra security features to the payload of MQTT is traditionally the responsibility of the implementer. [5]

Figure 2 represents the MQTT architecture. The following are the fundamental targets of this work:

- To analyse TLS security for the payload in MQTT against a cyber-attack such as MITM [7]
- To propose an AES encryption technique for MQTT protocol for wireless sensor networks [8]

We divide the paper into sections, which represent: Section 2 presents related work in IoT with Mqtt protocol w.r.t security. Section 3 talks through the overall methodology/ implementation of two scenarios MQTT payload when exposed to Man in the middle attack. Lastly, Section 4 talks about the results and conclusion.

2. RELATED WORK

Security protocols in IoT network such as MQTT have notable amount of research being done on them. MQTT falls in the category of resource-constrained protocol, TLS (transport layer security), is the suggested protocol over MQTT. Providing end-to-end security by the means of payload encryption are the other regions of research, that is being concentrated on. The publish/subscribe is a typical example which is useful in IoT scenarios because of its incident-built nature, disassociated reporting, high yield and flexibility. [9] [10]

In order to keep data private these factors raise doubts. Both messages and subscriptions must be kept confidential, if the broker is disturbed then, it might act as a point of failure. Due to the disassociated nature of reporting, end-to end security cannot be guaranteed by using TLS. When using a secure channel for providing security, the channel from publisher to broker and broker to subscriber, if the data is stored in the broker, it is still at a potential risk. In contrast, payload encryption will ensure object-based security, wherein irrespective of the secure or unsecure channel, the MQTT payload is still confidential to the probable cyber-attacks. [11] [12]

Whether it is using symmetric, asymmetric or hybrid technique, several arenas of research that facilitate security to IoT’s protocol – MQTT are in trend. Let us look at the first proposed technique. For using over MQTT an adaptation of the Augmented Password-Only Authentication and Key Exchange (AugPAKE) which is based on symmetric framework is proposed. The algorithm of AugPAKE is such that between the client and broker, it hands on, authenticated session keys. Hence, using the established session key, a secure symmetric-key encryption data is then sent and because the initial authentication of clients is made simpler compared to TLS and the final transmission overhead is decreased. However, because of the fact that the communication is only between client and broker, and not end-to-end this method is not full proof. Now, let us look at asymmetric key encryption method [2]. This makes use of lightweight attribute-based encryption (ABE) instead of elliptic curves. Only approved clients can decrypt the data, since ABE encrypts the data and works on access control policy. Hence, even though the broker has access privileges, it cannot decrypt it. Thus, assuring end-to-end encryption as only valid subscribers may decrypt the data. Here, one encryption process is required to convey the message to all subscribers, hence proving advantageous for its support in broadcast encryption. The suggested protocol is secure against most of the familiar cyber-attacks. Although, a significant processing overhead is incurred for implementing ABE. [13]

MQTT, as discussed is not efficient enough to provide any data encryption to the payload hence, puts a potential threat on payload. Whether the payload system uses any authentication mechanism explicitly, the attacker, if a man in the middle, per say, can intrude the data. To suit the IoT environment many areas of research look into tailoring TLS. TLS, does not exist on the application layer but in the transport layer hence it is possible for TLS and payload encryption can work hand in hand. [1] [14]
3. IMPLEMENTATION

The official document of MQTT v3.1.1 says “As a transport protocol, MQTT is concerned only with message transmission and it is the implementer’s responsibility to provide appropriate security features. This is commonly achieved by using TLS.” So, on one hand while the protocol document suggests us to make use of TLS, it also suggests that using TLS alone might not suffice. Hence, it asks us to make use of additional payload encryption techniques such as cipher encryption or use of public/private key encryption techniques. [2]

To conclude which payload encryption technique is appropriate, we deal with 2 different cases, each using same data and same security threat (man in the middle attack) but different encryption technique for the payload. Our first scenario includes usage of TLS certificate for payload and its ability to prevent a MITM attack and second scenario is where the encryption of payload will take place using a private key pair and its ability against MITM attack will be checked.

Pre-requisites for scenario 1 and 2 are:
Installing Paho client, CA certificate, Kali Linux and ARP spoofing.
Paho client will be the subscriber to a topic on the broker (broker can be the local machine or the ip address of another laptop/ computer) and kali linux will execute urlsnarf to sniff the data that is subscribed by the Paho client.
Create your own certificate using openssl, this will help creating CA certificate and server certificate.
The CA certificate installation steps are shown in figure 3, figure 4, figure 5, figure 6, figure 7, figure 8 and figure 9:

![Open File - Security Warning](image1)

**Figure 3.** Step 1: Open CA certificate after installing OpenSSL

![Certificate Import Wizard](image2)

**Figure 4.** Step 2: Choose the location to store the certificate
Figure 5. Step 3: Specify the location

Figure 6. Step 4: Click finish for the specified settings

Figure 7. Step 5: Click ok for a successful import
Figure 8. Check the expiry of the certificate

Summarising the CA certificate installation steps in a flow chart:

1. Open CA certificate after installing OpenSSL
2. Choose the location to store the certificate
3. Specify the location
4. Click finish for the specified settings
5. Click ok for a successful import
6. Check the expiry of the certificate

Scenario 1: Digital certificate is installed in chrome, IE as well as edge. A topic is subscribed by Paho client and simultaneously ARP spoof tried to urlsnarf. Since this is a TLS level encryption, a http url is easily attacked by Man in the Middle.
Figure 9. Check on the web browser if the certificate got imported

The GUI at the Python IDE (as shown in the figure 10) where we (PAHO client) would select the TOPIC to be PUBLISHED. The TOPICS will direct to the respective http url that will give information about that TOPIC. So here the dropdown has 2 topics to select from, each giving information of that particular topic to the subscriber by re-directing to the url that holds information.

Figure 10: MQTT GUI with topic dropdown

Once the SUBSCRIBER has selected the TOPIC, the messages between the subscriber and the broker will be as follows as shown in figure 11:

- CONNECT (from the subscriber to the Broker)
- CONACK (from the broker to the subscriber), here the connection between subscriber and broker is achieved.
- SUBSCRIBE (from the broker to the publisher)
- PUBLISH (from the broker to the subscriber)
- SUBACK (from the broker to the publisher), here the broker has published the topic for the subscriber
- DISCONNECT (from the broker to the subscriber)
Figure 11: Paho client connecting to online broker and acknowledgement messages

After the topic is selected from the dropdown, the arp spoofing already set in KALI LINUX, which will try to spoof into the re-directed url i.e. the payload or the topic to be published to the subscriber.

Scenario 2: A topic subscribed by Paho client is encrypted using AES encryption. The Man in the middle attack fails

The pseudo code looks like:

1. Define the Tinter GUI
   ```python
define the Tinter GUI
app = tk.Tk()
app.geometry('400x100')
labelTop = tk.Label(app, text = "Choose your subscribed topic")
labelTop.grid(column=0, row=0)
comboExample.grid(column=0, row=1)
bt = tk.Button(app, text = "Enter", command= onclick)
```

2. Define all components for the GUI
   ```python
define all components for the GUI
app = tk.Tk()
app.geometry('400x100')
labelTop = tk.Label(app, text = "Choose your subscribed topic")
labelTop.grid(column=0, row=0)
comboExample.grid(column=0, row=1)
bt = tk.Button(app, text = "Enter", command= onclick)
```

3. Define Paho client
   ```python
define Paho client
Eg:
cname = "192.168.56.1"
```

4. Define broker
   ```python
define broker
broker = "sig.ddns.net"
```

5. Subscribe to topic
   ```python
subscribe to topic
client1.subscribe(topic)
```

6. Publish the topic
   ```python
publish the topic
client1.publish(topic, webbrowser.open('http:// //'))
```

4. RESULT
A simulation with 100 PUBLISH ‘http’ url TOPICS was done, one of which result is shown in here in figure 12 and 13. Each TOPIC was 4 bytes which was redirected as a url once the dropdown of the python GUI was clicked. A MITM cyber attack was launched for both cases, that is, case 1 or scenario 1, when a CA certificate was provided to the server as security to payload and the TOPIC was PUBLISHED and later in case 2 or scenario 2 when the AES encryption is done for the payload. Since the payload of MQTT has the actual data from publisher to subscriber, it is more vulnerable of a cyber-attack. Here, since we are using MITM, the system is analysed under two lenses, the suitability
for the MQTT-based IoT environment when the payload is provided with TLS certificate on the transport layer and when the payload is provided with encryption using AES technique on the application layer.

The Paho client PUBLISHES the TOPIC to the SUBSCRIBER using client1.publish() as shown in the pseudo code. The topic could be any http url.

KALI Linux is already running in the background, it has been set up with the ARP spoofing for MITM attack. So, now we analyse, the success or failure of the spoofing using MITM for a PUBLISHED topic such as this one.

Below is the Table 1 that shows the topics (out of the 100 TOPICS attempted, 5 are mentioned below) that were given as dropdown menu on the python GUI where, the subscriber chooses (subscribes to the TOPIC):

| Topics to Be Published | Encryption Applied | MITM attack |
|------------------------|--------------------|-------------|
| TOPIC 1 http://www.foxnews.com/ | TLS certification | Failure |
| TOPIC 1 http://www.foxnews.com/ | AES Encryption | Success |
| TOPIC 2 http://cg.nic.in/rajnandgaon | TLS certification | Failure |
| TOPIC 2 http://cg.nic.in/rajnandgaon | AES Encryption | Success |
| TOPIC 3 http://cesembedded.com | TLS certification | Failure |
| TOPIC 3 http://cesembedded.com | AES Encryption | Success |
| TOPIC 4 http://scratchpads.eu/explore/sites-list | TLS certification | Failure |
| TOPIC 4 http://scratchpads.eu/explore/sites-list | AES Encryption | Success |
| TOPIC 5 http://icmsmt.com/ | TLS certification | Failure |
| TOPIC 5 http://icmsmt.com/ | AES Encryption | Success |

The following Table 1. Can be shown in a column bar graph as well, wherein, 1 stands for success and -1 is failure.

Table 1: Column graph depicting MITM attack success and failure.
One of the topics has been illustrated below:

- The scenario 1 experiment conducted on the payload of MQTT while publishing and using TLS protocol on the transport layer, which was the installation of the certificate on the browser, for both client and publisher proved to fail in case of a URL sniffer in terms of a Man in the Middle attack.

![Figure 12: MITM succeeds in sniffing the url](image)

- While, the digital certificate existed on both ends the client and publisher itself, the ARP spoof was successfully conducted by the means of MITM attack.
- Hence, the security at the transport layer level protocol on this attack i.e. MITM failed to secure the MQTT payload
- On the other hand, scenario 2 experiment, wherein the payload (PUBLISHED TOPIC) was provided AES encryption was able to prevent a MITM attack. The PUBLISHED topic here, was using cipher key generation for AES encryption.

![Figure 13: MITM fails in sniffing the url](image)

5. CONCLUSION

The experiment conducted on the payload of MQTT while publishing the topic and using TLS protocol for both client and publisher proved to fail to provide security to the payload at transport layer. When the payload is secured using channel-based security that is, while, the digital certificate existed on both ends the client and publisher itself, the ARP spoof, was successfully conducted by the means of MITM attack. Although, when the same payload topic was secured using AES encryption, MITM sniffing failed. Hence, the security at the object level i.e. at the application layer should be considered for the payload rather than at the transport layer (TLS certificate) so as to make MQTT protocol more secured.

6. REFERENCES

[1] A. Banks, R. G. (2014). Mqtt version 3.1.1. OASIS Standard.

[2] Monali S. Gaigole, Prof. M. A. Kalyankar, IJCSMC, Vol. 4, Issue. 5, May 2015, The Study of Network Security with Its Penetrating Attacks and Possible Security Mechanisms, A Monthly Journal of Computer Science and Information Technology

[3] OLASEHINDE Olayemi O., ALESE B. K., ADETUNMBI A.O., International Journal of Computer Science and Information Security (IJCSIS), Vol. 16, No. 12, December 2018, A Machine Learning Approach for Information System Security

[4] Rodrigo Emiliano, Mário Antunes, The 10th International Conference on Computer Science & Education (ICCSE 2015) July 22-24, 2015. Fitzwilliam College, Cambridge University, UK, Automatic network configuration in virtualized environment using GNS3
[5] Aadil Al-Mahrouqi, Sameh Abdalla, Tahar Kechadi, (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 6, No. 6, 2015, Cyberspace Forensics Readiness and Security Awareness Model

[6] R. Mohtasin, P.W.C. Prasad, Abeer Alsadoon, G. Zajko, A. Elchouemi, Ashutosh Kumar Singh, IEEE WiSPNET 2016 conference, Development of a Virtualized Networking Lab using GNS3 and VMware Workstation

[7] N. Angelescu, D.C. Puchianu, G. Predusca, L.D. Circiumarescu, G. Movila, ECAI 2017 - International Conference – 9th Edition Electronics, Computers and Artificial Intelligence 29 June -01 July, 2017, Targoviste, ROMÂNIA, DMVPN simulation in GNS3 network simulation software.

[8] Premkumar S and Saminadan V, International Journal of Applied Engineering Research ISSN 0973-4562 Volume 12, Number 14 (2017) pp. 4443-4447 © Research India Publications. http://www.ripublication.com, Impact of Denial of Service (DoS) attack in Smart Distribution Grid Communication Network

[9] Jihyun Kim, Jachyun Kim, Huong Le Thi Thu, and Howon Kim, 2016 IEEE, Long Short Term Memory Recurrent Neural Network Classifier for Intrusion Detection.

[10] Thomas Rausch, Stefan Nastic, Schahram Dustdar, 2018 IEEE International Conference on Cloud Engineering, EMMA: Distributed QoS-Aware MQTT Middleware for Edge Computing Applications

[11] Mahrouqi, P. Tobin, S. Abdalla, and T. Kechadi, International Journal of Computer Theory and Engineering, Vol. 8, No. 3, June 2016, Simulating SQL-Injection Cyber-Attacks Using GNS3

[12] Abdessamad MEKTOUBI, Hicham LALAOUI, Abdelouahed ZAKARI, 2016 IEEE, New approach for securing communication over MQTT protocol A comparaison between RSA and Elliptic Curve

[13] Joseph Jose Anthraper, Jaidip Kotak, International Conference on Sustainable Computing in Science, Technology & Management (SUSCOM-2019), Security, Privacy and Forensic Concern of MQTT Protocol

[14] Tetsuya Yokotani, Yuya Sasaki, The 2016 International Conference on Control, Electronics, Renewable Energy and Communications (ICCEREC), Comparison with HTTP and MQTT on Required Network Resources for IoT

[15] Stefan Profanter, Ayhun Tekat, Kirill Dorofeev, Markus Rickert, Alois Knoll, An-Institut Technische Universitat Münch, Munich, Germany, OPC UA versus ROS, DDS, and MQTT: Performance Evaluation of Industry 4.0 Protocols