IETE Journal of Research

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/tijr20

MQTT Vulnerabilities, Attack Vectors and Solutions in the Internet of Things (IoT)

Ahmed J. Hintaw, Selvakumar Manickam, Mohammed Faiz Aboalmaaly & Shankar Karuppayah

To cite this article: Ahmed J. Hintaw, Selvakumar Manickam, Mohammed Faiz Aboalmaaly & Shankar Karuppayah (2021): MQTT Vulnerabilities, Attack Vectors and Solutions in the Internet of Things (IoT), IETE Journal of Research, DOI: 10.1080/03772063.2021.1912651

To link to this article: https://doi.org/10.1080/03772063.2021.1912651

Published online: 04 May 2021.

IETE JOURNAL OF RESEARCH
https://doi.org/10.1080/03772063.2021.1912651

REVIEW ARTICLE

MQTT Vulnerabilities, Attack Vectors and Solutions in the Internet of Things (IoT)

Ahmed J. Hintaw[1,2], Selvakumar Manickam[1], Mohammed Faiz Aboalmaaly[2] and Shankar Karuppayah[1]

[1]National Advanced IPv6 Centre (NAv6), Universiti Sains Malaysia, Penang, Malaysia; [2]Department of Computer Techniques Engineering, Alsafwa University College, Kerbala, Iraq

ABSTRACT
Internet of Things (IoT) paved the way for devices and machine communication using TCP/IP protocol. Lightweight and stateless communication is imperative especially in a situation requiring conservation of energy usage, *e.g.* wireless sensor network. Representational State Transfer (REST) API method is based on web communication protocol, Hyper-Text Transfer Protocol (HTTP), and is widely used in IoT messaging. Some of these protocols are DPWS, XMPP, MQTT, COaP, AMQP. Among these protocols, MQTT is the most preferred protocol and is expected to be the de facto messaging IoT standard. MQTT uses a publisher/subscriber model to facilitate messaging between devices making messaging lightweight. Nevertheless, there are a number of security issues due to the design of the protocol itself. Some of the issues are denial of service, identity spoofing, information disclosure, elevation of privileges and data tampering. These issues can be caused by both internal and external perpetrators. Researchers have proposed various security techniques and mechanisms to address these issues. Incorporation of security has added processing overhead to the devices and this will have a bearing on IoT devices that are powered by a battery. This issue has opened up new research challenges in making the protocols more lightweight and at the same time not compromising the level of security provided.

KEYWORDS
COaP; Data communication protocols; DPWS; Internet of things; MQTT; MQTT threats; Publish-subscribe; Security; XMPP

1. INTRODUCTION

This innovation of machine-to-machine (M2M) concepts or Internet of Things (IoT) is the predictable choice in the near future. The next revolutionary technology that will most likely happen is transforming the traditional Internet into the future IoT that is fully integrated. According to [1], IoT relates to enlarging the ability of non-computer devices such as sensors or actuators in terms of making mutual communications between these devices with no human intervention to generate, exchange and use data. Besides, [2], IoT is uniquely identifiable things (sensors or actuators) connected through the network to control the status of the “Things” or retrieve its data. The “Things” are programable and provide services with an insignificant intervention of the human. Through several definitions from IoT Corporation [1,2], these explicate the major components of IoT which are the *Internet* and *Things*. The Internet permits Things to communicate and by using standard communication protocols it gives humans the ability to communicate with Things. Accordingly, IoT, represented as Things, has a unique identifier that can be blended within the environment in a seamless manner and Internet connectivity makes it accessible by other objects (Things) to exchange data and achieve a common goal. Of importance is that IoT objects are expected to be on the rise, as stated by Ericsson Mobility, where 28 billion IoT devices will be interconnected and active in 2020 [3]. Besides that, the number of connected devices is speculated to exceed the number of human beings and its connectivity will increase continuously.

Examples of IoT objects (things or nodes) that have been used in a wide range of applications, to augment humans’ conveniences are; smart sensors and health trackers [4], smart home automation systems and smart TVs [5], smart refrigerators [6], IP cameras [7], and others. Because of a large number of reasons, perpetrators can target the IoT nodes. These IoT nodes are active on the Internet network [8], remotely controlled [9], use insufficient security mechanisms due to resource limitations [8], and generate sensitive data that can be easily manipulated [10]. Furthermore, due to reasons such as cost, time to ship, and easy user familiarity, security takes low priority by the vendors when planning to develop IoT devices [1,11].

Due to the growing menace of cybercriminals, these cyber-attacks threaten IoT devices. A group of malwares that target smart nodes was reviewed in 2016 by a report from L3 Communications. The report indicates that bots that were hosted in Brazil, Colombia, and Taiwan; there are more than one million infected nodes in these bots [12]. Distributed denial of service (DDoS) attacks that were recently reported [13,14] in many network infrastructures have been attributed to botnets that are made up of infected IoT devices. According to an article presented by [15] explicated that more than 70 different vendors that provide Digital Video Recorder (DVR) services were exposed to run a remote code. Moreover, Cybersecurity Proofpoint in 2014 indicated that there were 100,000 consumer devices in a botnet including a refrigerator linked to the Internet, that dispatched 750,000 spam email messages to organizations and people all over the world [16]. A smart vehicle that was tested in 2014 was attacked by a buffer overflow and it was remotely controlled [17]. Therefore, according to the above-mentioned events, it's important to identify the threats that target the IoT nodes through the traditional Internet network and the need for a suitable security solution that should be properly deployed.

Typically, there are different types of hardware and software in the area of IoT; functionalities can rely upon the hardware, and managing them by software applications. Hence, IoT devices need to be able to easily communicate in order to manage all the smart nodes that is owned by the same user. Scientific literature has exhibited various IoT communication protocols to achieve bidirectional communication between devices and server/cloud (D2S) and between IoT devices (D2D) [18,19]. Message queuing telemetry protocol (MQTT), constrained application protocol (CoAP), extensible messaging and presence protocol (XMPP), and devices profile for web services (DPWS) are the most notable and popular IoT data protocols. Among these data protocols, MQTT is the most widely adopted in IoT [20]. MQTT uses minimal resources and consumes less energy [21]. Exchanging messages among IoT nodes are facilitated by the MQTT Internet-facing broker/server. Hence, the security threats in the IoT domain that adopts MQTT needs to be identified to be protected.

In this paper, the focus will narrow on MQTT, an IoT data protocol that follows publish/subscribe programing model that grew quickly and became the de facto standard. Thus, the structure of the paper is as follows: Section 2, shows the related work. Section 3, depicts an overview of the IoT paradigm. Section 4, outlines the technical aspects of MQTT. Section 5, explicates the widespread security solutions of MQTT. MQTT potential threats and their challenges are depicted in Section 6. Section 7, presents an elaboration of IoT malware evolution. The possible solutions to secure MQTT are explained in Section 8. Overall discussion based on the aforementioned sections and the numerous promising security choices at each layer are demonstrated in Section 9. In Section 10, a number of challenges allied to IoT-based MQTT are exhibited. Finally, the conclusions of the study are elaborated in Section 11.

2. RELATED WORK

For the purpose of reviewing the research work done in literary works and assessing if the subject has been thoroughly investigated, the approach adopted to execute the conducted survey is outlined here. Considering that the IoT arena and its security are relatively recent, the research focus was on the publications concerning the MQTT protocol that was presented in 2000–2020. Those publications include reports, books, white-papers, conferences, and journals.

A variety of security surveys have been carried out in the field of IoT. However, most of these surveys have argued about the conventional security of IoT in general such as its security mechanisms and types for the conventional networks. There are several studies available in the literature regarding defending against IoT-based attacks which can be used significantly to build a base model for something like a defense to tackle the latest MQTT-based assault on an IoT network. The search keywords are "IoT security", "IoT security issues", "IoT attacks", "MQTT attacks", "MQTT security" and "IoT countermeasures" as samples. Although a significant number of studies have been carried out to tackle IoT-based MQTT threats in the state-of-the-art, in a comprehensive approach, a handful of studies have attempted to explore IoT-based MQTT threats. Among the much more common of which are:

The Authors in [21] introduced a new approach to addressing IoT vulnerabilities and attacks focused on a four-layer model consisting of transport, storage, interfaces, and objects. While this study outlined several attacks on these stages, it did not comprehend all potential MQTT attacks in IoT. In [22] communication, edge computing, and edge nodes have been reviewed by identifying the potential threats on each stage. In addition, a number of defense systems were presented to mitigate such threats. Despite identifying potential threats and their defense systems at these levels, other critical elements in IoT networks were untouched. For instance, threats against data on IoT nodes locally or on the cloud remotely have been uncovered. The study in [23], discussed the existing solutions such as symmetric encryption or Transport Layer Security and they proposed a method known as Value-to-HMAC, but the study does not cover all MQTT vulnerabilities.

Table 1: Comparison among several surveys on MQTT threats in IoT

Existing surveys	Year	Security issues	MQTT based attacks Taxonomy	Botnet Over MQTT	MQTT secure mechanisms Comparison	Challenges and issues	Possible solutions
[24]	2017	✓	✔	✔	✗	✗	✗
[25]	2017	✓	✗	✔	✔	✔	✔
[26]	2018	✓	✗	✗	✓	✗	✗
[27]	2018	✓	✗	✗	✔	✗	✔
[28]	2019	✓	✗	✗	✔	✗	✓
[29]	2019	✓	✗	✗	✗	✗	✓
[30]	2020	✗	✔	✔	✗	✗	✓
[31]	2020	✓	✗	✗	✗	✗	✔
Our paper	2020	✓	✓	✓	✓	✓	✓

✗ not covered; ✓ covered; ✔ partially covered

Our study not only describes conventional attacks or IoT-based attacks but it also explicitly presents the latest trends and scenarios in the MQTT arena within the IoT context. Modern threats in the IoT environment have contributed to alternative techniques of malware activity. Therefore, it is also mandatory to study those kinds of recent variants of attacks and their characteristics to develop a strong defensive mechanism against these attacks. Table 1 compares our study with other recent studies conducted in threats directed at the MQTT protocol. We have gathered studies on attacks of the MQTT in IoT particularly we gathered to achieve a better understanding of the content to be covered. Nevertheless, most of them are based truly on conventional IOT attacks and their defense mechanisms.

3. IOT REVIEW

In 1998, Kevin Ashton introduced the IoT concept [32,33]. He declares that IoT are computers that are linked to each other over the Internet and knows everything about "things." IoT has the ability to collect and use data from these "things" without any guidance from a human being. In IoT, the electronic devices that are used frequently are interconnected to each other with the capability of sensing and contextual knowledge [34,35]. These devices are permitted to gather and give-and-take data. Besides that, domains such as smart environment, healthcare, transportation, industry, logistics as well as city information, social gaming robot, and personal lives have a wide range of IoT these days [36]. As depicted in Figure 1, the market of IoT has and will experience significant rapid growth as well as in the coming generations. The market of IoT value is expected to hit a market cap of 771 billion USD by 2026, starting from 157 billion USD in 2016 [37,38]. Controlling all the IoT objects can be anywhere or anytime because service is presented on the cloud which allows actions quickly. For example, the smart home can get commands through smartphones. These features were presented in IoT [39,40].

3.1 IoT infrastructure

The IoT platform includes a variety of intelligent objects that gather, process, compute, and interact with other intelligent objects. There are three layers in the IoT domain which are application, network, and physical layer. There are also many things recently presented by industries that are embedded with smart things. As presented in Figure 2 some emerging technology which is integrated with IoT. The IoT technology could either be IoT-Fog-Cloud or IoT-Cloud based. For reliable IoT technologies, security concerns such as real-time monitoring [42], data privacy [43], IoT test bed [44], and machine to machine communication [45] should have to be discussed. The architecture of the IoT can be organized in a centralized, distributed, decentralized way. One of the most complicated issues in IoT applications is computing as well as processing in real-time. Cloud computing offers guarantees data protection and more storage. But nowadays, most IoT application monitoring in real-time involves processing and computation at the network edge. This allows immediate action could be taken, such as pursuing serious fire detection, patient's health condition. It will be much more prone to the adversary when processing and computation are done on the network edge utilizing fog devices, as these devices are limited resource devices, the conventional defense is not sufficient. A technique such as a machine learning has lately been adopted in IoT which makes it more intelligent and autonomous to make a decision. Various smart nodes are linked together to provide an application using certain common protocols such as the MQTT protocol. To obtain the device tamper-proof and construct trust among end-users, IoT infrastructure security concerns need to be tackled. Utilizing an

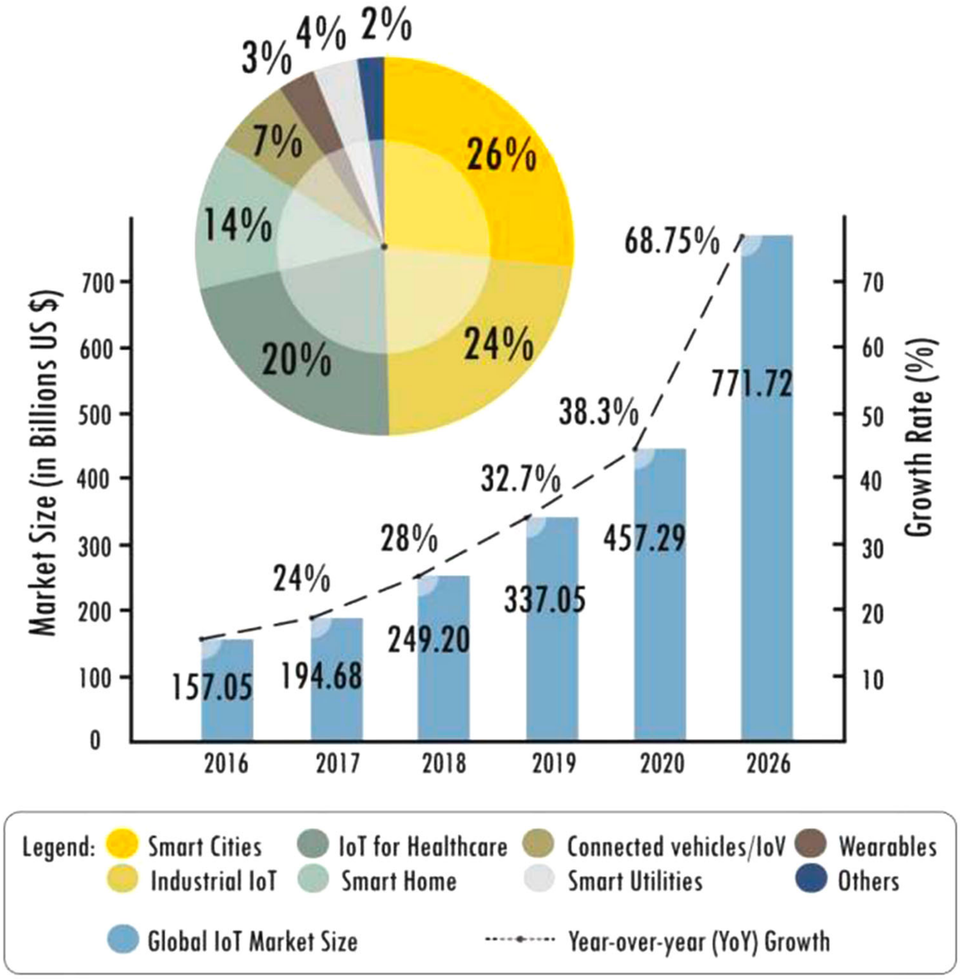


Figure 1: Global IoT market size (2016–2026)

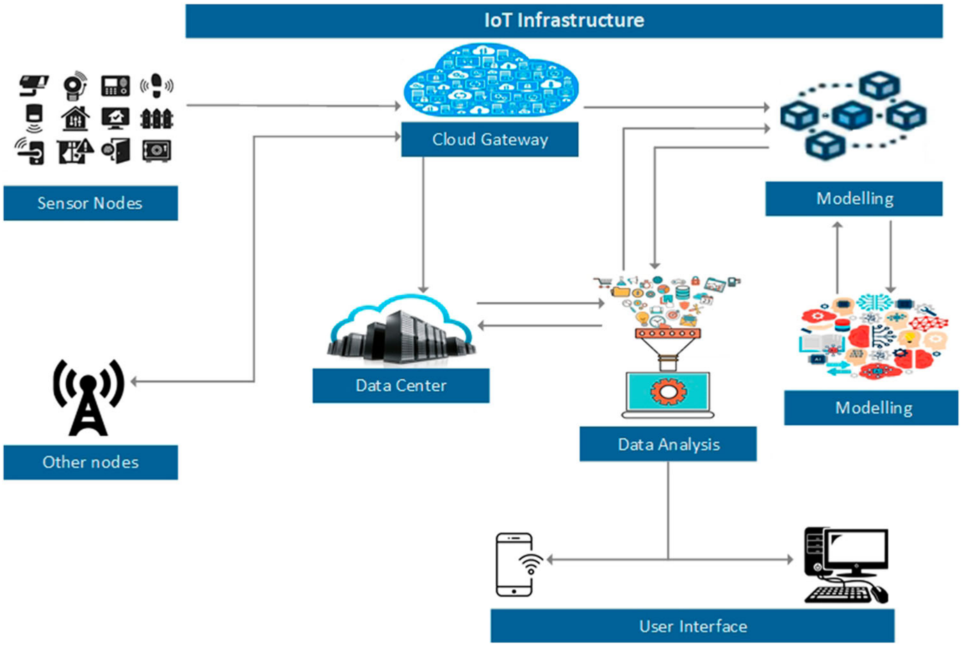


Figure 2: Internet of things infrastructure [41]

intelligent algorithm in the IoT platform could afford data interoperability [46].

3.2 Technological Challenges of the IoT

On its own, there are various challenges in the IoT network on its own, this allows one a guide to identifying the overall challenges when designating IoT security algorithms. As exhibited in [47], IoT challenges are still not limited and therefore can be shortened as follows:

Heterogeneity: IoT comprises a number of devices that came from different families including such actuators, sensors, smart appliances, mobile systems, gateways, etc. Diverse communications protocols are utilized by these devices as well as operate on different systems and utilizing diverse algorithms for data processing.

Scalability: There are unprecedented challenges in the conventional network that have been appeared in the IoT including such millions of nodes needs to servicing, managing, naming, and addressing.

Communications: IoT devices utilized numerous technologies including such wireless communications, *e.g.* ZigBee, LPWAN, and Bluetooth as well as wired.

Energy consumption: Due to most IoT nodes are a limited resource, energy consumption is main issue in the IoT. Inventing any kind of algorithm to be run on the IoT nodes should have less complexity to obtain lightweight specifications.

Data Privacy: User data privacy for certain particular use scenarios may be a matter in the IoT. For illustration, in the public mode operation, location information of the IoT objects could be provided to other surrounding machines including the administrator of the system when needed. Nevertheless, if they are in the private mode operation, location details could be kept in secret to make the privacy of their own location details protected [48,49].

Self-awareness: some pre-determined particular tasks to be fulfilled and with no the interference of humans to respond in real-word environmental circumstances, IoT smart nodes should self-organize in order to perform such activities.

Interoperability: The format of the data sharing should be standardized as well as pre-determined to let heterogeneous nodes to interact, cooperate and share data with other IoT nodes.

In addition, wireless sensor networks architecture in IoT raises numerous challenges and concerns regarding network security, owing to a variety of factors including such unreliable links, lack of infrastructure, dynamic topology, limited physical security, and resource limitations. Such “things,” due to their vulnerability, have to be secured from internally or externally threats. The next section will explores security issues and IoT vulnerabilities.

3.3 Security Issues in IoT

It is a highly complicated job to secure IoT from a number of potential threats. However, when referenced under its layered architecture, it will to be manageable to some extent. Vulnerabilities and limitations exist in every layer that must be identified to guarantee the security of that layer by protecting it from various threats [50]. An appropriate and an effective security system is required to address observed vulnerabilities of the IoT object and prevents such threats.

In addition, malicious activities could be common on IoT networks because IoT devices may be reached from an untrusted network such as the internet remotely. Consequently, sensitive data can also be revealed at any moment because of security vulnerabilities that haven’t yet been fixed. Thus, security in the IoT sector must be taken into account and concerned, specifically for security goals as well as real deployments [51,52].

It is important to zoom in for that on weakness and how it assists in a threat. Vulnerability reflects the system’s incompetence, which helps the intruder to detect the scope to breach system protection. This can lead to an increased risk which if ignored, can contribute to an attack. As stated by [53], Table 2 lists the vulnerabilities list along with their contribution factors, which are potentially responsible for the occurrence of an attack on IoT machines.

The results are for some of the vulnerabilities are caused through careless and irresponsible behavior in handling the IoT device. Therefore, there is a very important measure above all, that can easily be taken care and that is called self-awareness. In case of some unexpected behavior in the IoT nodes, the user should be well aware of all the hazards involved and there should be an appropriate intervention.

To deal with smarter, newer attack-launch possibilities, the existing security system must be supplemented with features such as firewalls, content filtering, intrusion

Table 2: IoT arena vulnerabilities and their responsible weaknesses

Vulnerability	Responsible weak points
Lack of sufficient authentication and authorization	The intruder could have access to a certain interface with default weak passwords, weak password retrieval systems, lack of granular access control, and vulnerable protected credentials
Unreliable user interfaces	The controls or data can be reached by using no transport encryption, weak password retrieval systems, vulnerable of login credentials, and plain-text credentials
Insecure network services	Services of susceptible networks could be utilized to target a system or node by the attacker.
Privacy issues	The information which has weak protection could be reached by the intruder due to the inadequate authentication, the service of the network not secure, lack of transport encryption, as well as unreliable interfaces.
Insufficient transport encryption/integrity verification	Inadequate transport encryption enables an attacker to access data sent across the network.
The inadequacy of the security configuration	The adversary can be able to reach controls of the system and its data due to the lack of password options or encryption, and lack of the permissions. Any node in the IoT network could be a cybercriminal.
Poor physical security	The adversary could access the operating system and data via memory cards, USB ports, and other storage/peripheral devices.

detection system, application whitelisting, and inspection technologies [54].

3.4 Identify IoT Security Goals and Security Attack

The most basic concepts in the IoT domain will be clarified in this section: a security attack and a secure object [55]. It is important to understand the security goal to distinguish the required security. Terms confidentiality, integrity, and availability are three main groups of traditional security and known as the CIA triad in the state-of-the-art. Confidentiality is correlated with a series of rules in which the information can be accessed only by the authorized entities. With the emergence of the Internet of things, the confidentiality of IoT-based MQTT objects is important because such objects can interact with sensitive information. For the service to be reliable in the IoT-based MQTT, the published data should be not modified and reached as it is by the subscriber and the subscriber has obtained only legitimate information as well as commands. Availability guarantees that the MQTT broker is accessible only by authorizable nodes in the MQTT network. Although the CIA triad is common, authors in [56] have shown that the CIA triad has failed to resolve recent threats that exist in a secure environment.

Table 3: IoT security requirements

Security goals	Description	Abbreviations
Confidentiality	Prevent the intruder from sniffing the data shared between the publisher and the subscriber and maintaining privacy and data safety in general.	C
Integrity	Ensuring that the data transmission from a sender as it is and it should be not manipulated by the intruder node.	I
Nonrepudiation	An intermediate node could store data packet and replay it later. Critical data may be used in the replay packet. Consequently, detection duplicate messages are required.	NR
Availability	The services of all IoT applications should be continually accessible and the deny access by the unauthorized systems or objects should not occur.	A
Privacy	The procedure in which policies or privacy rules is followed by an IoT system and enables users to manage the sensitive information.	P
Auditability	Guaranteeing the capability of the IoT system to do on its activities with solid monitoring.	AU
Accountability	This procedure in which to ensure that an IoT system makes the users responsible of their activities.	AC
Trustworthiness	Make sure that communicating with authentic nodes which are part of the network by confirming the other objects identities, so as to stop unauthorized access.	TW

An IAS-octave, alluded to the Information Assurance and Security presented by the authors which are security goals with a comprehensive set, in terms of assurance and security investigated of a vast number of data. The security goal details are presented in Table 3 with their abbreviations and descriptions suggested by the IAS-octave. The security attacks and the secure object can be described as follows, once classified as major security goals.

- Secure object is an entity matched to or fulfilled all security goals.
- Security attack is an attack with at least one of the security targets being breached.

3.5 IoT Data Protocols

This section paints a background of the common IoT data communication protocols. It should be noted that there is no data communication protocol adopted as a standard for IoT. Figure 3 depicts the chart of how AMPP, DPWS, COAP, and MQTT have enlarged these past fifteen years, which motivates the focus on MQTT.

3.5.1 The Devices Profile for Web Services (DPWS)

DPWS is a service-oriented approach on limited resources devices; an OASIS standard [57]. DPWS offers a publish/subscribe model, employs both TCP and UDP,

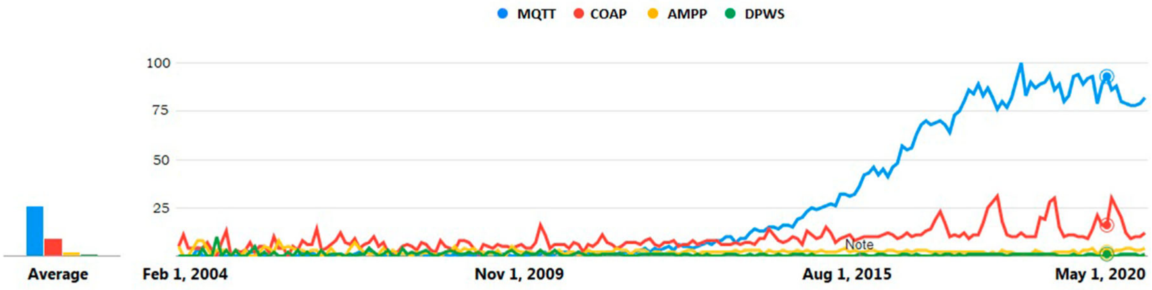


Figure 3: Google trends of MQTT, COAP, DPWS, and AMPP

seamless M2M interactions, does not support built-in QoS features, relying solely on TCP's delivery mechanisms and has no explicit support for integration with higher layers. DPWS is based on the Web Services Description Language (WSDL) [58] and Simple Object Access Protocol (SOAP) [59] which define an interconnect device service. The substantial merits of DPWS for event-driven in the IoT field are its secure Web services, eventing features, and dynamic discovery. Moreover, TLS merit is supported by DPWS.

Nevertheless, developers encounter various difficulties with DPWS when employing it on IPv4 Internet infrastructure. The main issue is difficulty in implementing the dynamic discovery mechanism in a public network because of the limited range of UDP multicast messages to the local subnet. Another issue is due to multiple bidirectional data communication and data representation in XML format which generates massive traffic on the Internet and grows latency in application/device communications. Thus far, home entertainment, automotive systems, and automation have widely employed DPWS in their systems [60]. Also, enterprise infrastructures and Internet can maintain integration with DPWS [61]. However, some features of DPWS such as publish/subscribe eventing and dynamic discovery will be limited in global device deployments due to a large number of IoT gadgets.

3.5.2 The Extensible Messaging and Presence Protocol (XMPP)

XMPP, a data protocol for message-oriented middleware which is an Internet Engineering Task Force (IETF) standard, was developed for real-time message stream and instant messaging (IM) applications [62]. XMPP is employed over TCP and it is not limited to only publish/subscribe (asynchronous) but also request/response (synchronous) messaging model that is more suitable for IoT. Low latency message exchange and small message footprint are supported by XMPP [58]. It is not practical for M2M communications because it does not offer QoS options. However, XMPP messages follow eXtensible Markup Language (XML) structures and incur significant overheads to message sizes. Lately, XMPP has regained a lot of attention as a data protocol suitable for IoT. One of the XMPP core specifications is the TLS/SSL support as a built-in security feature. XMPP affords ad hoc (P2P) via DNS service discovery (DNS-SD) [63] and multicast DNS(mDNS) [64].

3.5.3 The Constrained Application Protocol (CoAP)

CoAP is designed for devices that have limited resource and the design goal was to limit packet fragmentation and minimize the message overhead. CoAP is a synchronous request/response, and is a standard by the IETF [65]. CoAP runs over UDP and offers unicast as well as multicast. CoAP employs UDP and supports unicast and multicast. The commands of HTTP like GET, PUT, POST, and DELETE are used by CoAP to achieve resource-oriented connections in the architecture of client/server. CoAP includes a built-in pub/sub mechanism, a mechanism for discovering and advertising resource descriptions. It contains 4 bytes binary fixed header followed by compact binary options, header length between 10 and 20 bytes. CoAP transactions can be protected only by the use of DTLS because of no built-in security options offered by the protocol. Since CoAP packet structure is diverse, hence, HTTP servers might expose additional confusion when using DTLS. Other protocols for securing CoAP can be found in the literature [66,67].

3.5.4 Message Queuing Telemetry Protocol (MQTT)

MQTT is designed to target limited resources device and bandwidth limited communications by IBM/Eurotech [68] and an OASIS standard [69]. MQTT is suitable for IoT and M2M to link devices/applications because MQTT employs various techniques for routing such as many-to-many, one-to-many, or one-to-one. MQTT is based on publish/subscribe programing model, runs on top of TCP/IP, consisting of three components which are subscribe, publish, and broker. MQTT offers quality of service (QoS) feature [70] which is unique amongst IoT protocols. Moreover, MQTT messages contain a mandatory fixed-length header (2 bytes) and an optional message-specific variable length header and message payload as illustrated in Figure 4. It has limited security features in particular; MQTT employs SSL/TLS to protect data communication and simple authentication (username/password) [71]. MQTT does not have discovery capabilities and is not asynchronous. These are not significant in some IoT scenarios. Finally, according to [72], MQTT is considered as the second data communications protocol employed today by IoT.

Table 4 depicts the features comparison of all these protocols. There are numerous comparisons of IoT data communication protocols, theoretically and feature-by-feature presented by the researcher's community as well as some of them focusing on the performance evaluations and power efficiency in real implementations. [76] provides a high-level study of IoT data communication protocols, but it does not include DPWS. Within the field of smartphone applications, the authors present a comparative study between MQTT and CoAP [77]. Additionally, the authors in [78] have depicted a close study of MQTT, DPWS, and CoAP. The authors [79] have shown the behavior of the protocols (CoAP and MQTT) when the conditions of the network are changed. Finally, in [80], three data communication protocols (OPC-UA, MQTT, and CoAP) were compared in the laboratory environment in the scenario of using cellular networks for communication.

The focus is on the MQTT protocol because it is considered as the de facto standard protocol that is deployed in various IoT systems; it is found in applications in several domains. Various libraries for several development platforms like Arduino are available in IoT, for different programing languages, *e.g.* Python, Java, C, and Android and iOS for mobile platforms [81]. Notwithstanding, MQTT was recently standardized, it has been studied by many researchers in various domains such as smart homes [82], smart grid and WSNs [83], mobile IoT contexts [84], eHealth applications [85], and many more.

4. TECHNICAL ASPECTS OF THE MQTT PROTOCOL

This section depicts the technical aspects of the MQTT protocol and its features as presented in the following:

4.1 Publish/Subscribe

MQTT, as in general for all the publish/subscribe interaction models [74], is publishing messages and subscribing to topics. It consists of a small set of operations, including

Table 4: Feature comparison of the main IoT protocols

Protocol	Version	Protocol Type	Transport	Synchronous	Asynchronous	Discovery	QoS	Security	Computer Resources	Power
DPWS	1.1, [57]	Service Oriented	• TCP • UDP	Yes (Service Invocation)	• Publish-Subscribe to Service • WSEventing	WS-Discovery	No	• Payload encryption. • WS-Security. • TLS/SSL. • IPSec. • 802.15.4.	10Ks of RAM-Flash	Fair
XMPP	RFC 6120, IETF [62]	Message Oriented	TCP	Near-real time	Publish-Subscribe	XEP-0030	No	• SASL • TLS/SSL • Non-native end-to-end encryption	10Ks of RAM-Flash	Fair
COAP	RFC 7252,IETF [65]	Resource Oriented	• UDP • TCP planned	Yes (Request-response, via HTTP)	Observe Resource - RFC 7641	• RFC 5785 • RFC 6990	Limited	• Payload encryption. • DTLS. • IPSec. • 802.15.4.	10Ks of RAM-Flash	Excellent
MQTT	3.1.1, OASIS [69], ISO/IEC, 20922/2016 [73–75]	Message Oriented	TCP	No	Publish-Subscribe Topics	No	Three levels	• Payload encryption • TLS/SSL. • IPSec. • 802.15.4	10Ks of RAM-Flash	Good

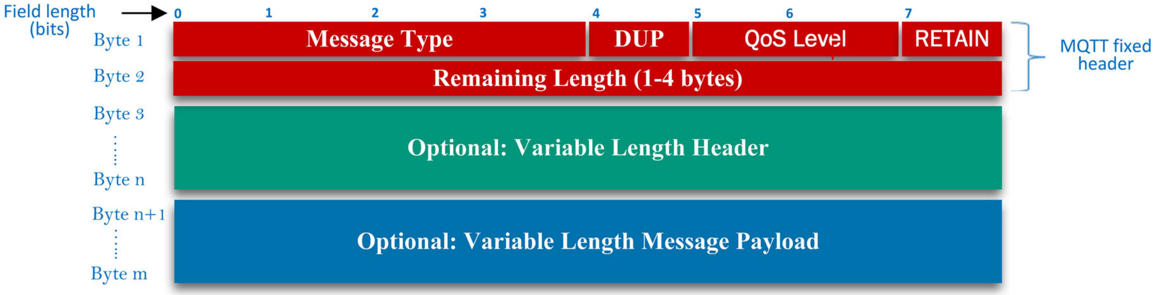


Figure 4: MQTT header

Table 5: MQTT operations set

Operation	Depiction
Publish	A publisher uses this operation to dispatch a message.
notify	Notifies the subscriber regarding the topic updates.
subscribe	A client employs this operation to subscribe to a particular topic.
unsubscribe	A client employs unsubscribe operation to cancel the subscription to a particular topic.

Table 6: MQTT QoS types

Level	Message Guarantee	Behavior
QoS 0	Fire and forget	MQTT message is dispatched once and no confirmation is needed.
QoS 1	Delivered at least once	MQTT message will be disseminated at least once, with an acknowledgment required
QoS 2	Delivered exactly once	Handshake technique is employed to guarantee the message arrives specifically one time.

the primitives pointed out in Table 5. The topics can be subscribed by clients and then receive all messages that are dispatched to those topics. Alternatively, the topics can be received all the published messages from clients and be readily available to subscribers on those topics. All these operations between publishers and interested subscribers are controlled by the mediated entity called a broker as illustrated in Figure 5.

4.2 Topics and Subscriptions

MQTT publishes and subscribes its messages at a particular topic, which can be represented as a subject field. Wildcard designators feature in the MQTT topics can restrict the received messages to particular topics [74]. MQTT topics depict the following characteristics:

- Topics are expressed as UTF-8 strings managed by the mediated entity (broker) to filter and clarify all messages of the connected clients.
- Topics consist of different levels in MQTT and can be more than one. A forward slash (/) is used to separate the level, representing as a structure of a logical tree.
- Topics are employed by the publishers and subscribers for dispatching messages and subscribing from other publishers.
- The wildcards feature in the topic allows subscribing at once to a multiple/exact topic, as described in the following:
  ○ +, precisely individual level.
  ○ #, several random levels.

4.3 Quality of Service Levels

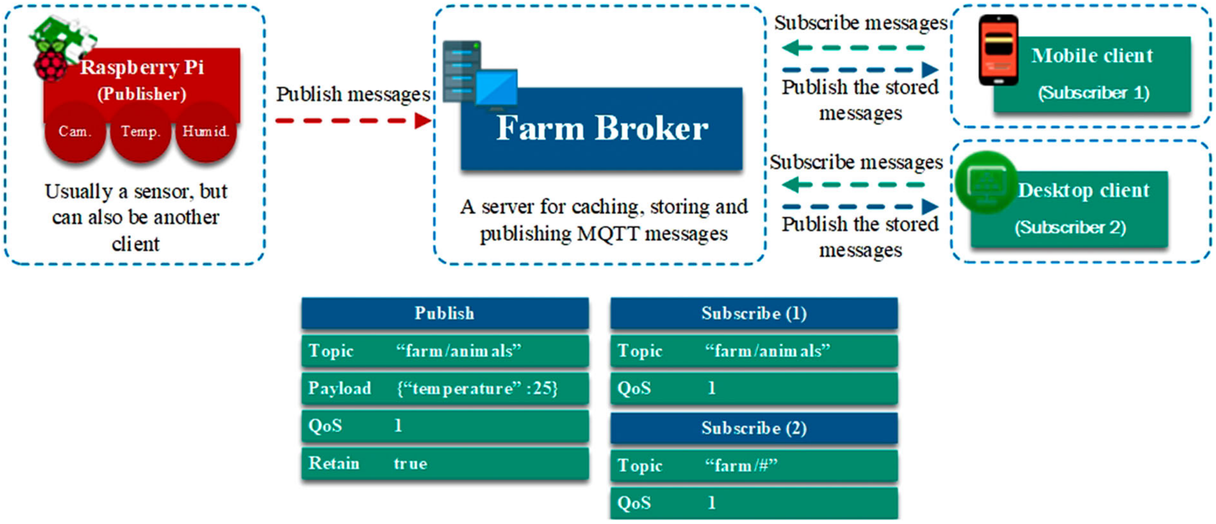
MQTT presents the quality of service (QoS) feature with three levels of message delivery as illustrated in Figure 6. Message is delivered through the server by assigning a higher level of effort with each level. High reliability of message delivery can be ensured through higher levels of QoS but due to latency issues, these will cause a delay or add extra network bandwidth. Table 6 presents these QoS levels.

4.4 Retained Messages

With MQTT, the message will be stored in the server even after dispatching it to all current subscribers. If the same topic in the server receives a new subscription, then the new subscribing node will receive retained messages.

4.5 Clean Sessions and Reliable Connections

Clean session flag will be set in MQTT client when it is connected to the server. If the flag is set to true, the server will remove all client's subscriptions when they are disconnected. If the flag is set to false, the server will deal with the connection as durable and any disconnection will not affect the client's subscriptions. After connection is reestablished, a high QoS designation that is carried by subsequent messages will be cached for delivery purposes. Clean session flag is an optional feature.


Raspberry Pi
(Publisher)
Cam.
Temp.
Humid.
Usually a sensor, but can also be another client
Publish messages
Farm Broker
A server for caching, storing and publishing MQTT messages
Subscribe messages
Publish the stored messages
Mobile client
(Subscriber 1)
Subscribe messages
Publish the stored messages
Desktop client
(Subscriber 2)

Publish	
Topic	"farm/animals"
Payload	{"temperature":25}
QoS	1
Retain	true

Subscribe (1)	
Topic	"farm/animals"
QoS	1

Subscribe (2)	
Topic	"farm/#"
QoS	1

Figure 5: MQTT publish/subscribe interaction model

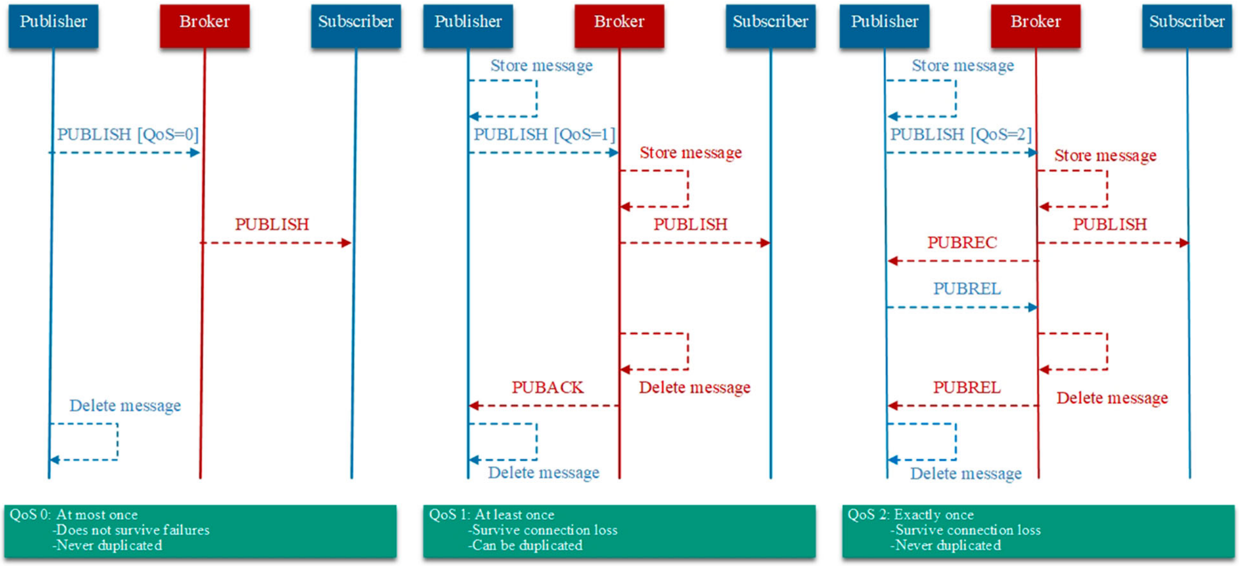


Figure 6: MQTT QoS levels

4.6 Wills

In MQTT a *will* is used to inform the server that there is a message or a "will" must be published to particular topic/topics in the case of unexpected disconnection. A *will* is usually helpful in security settings or alarm, to help the system administrator to get an idea immediately about when the lost connection occurred between the remote sensor and the network.

5. MQTT SECURITY

The current MQTT implementation verifies only simple security objects such as identity, authentication, and authorization policies [75]. Identity in MQTT states that the IoT node has the rights to access. Authentication gives the node identity and it confirms whether a node has rights to access. The client can set these policies by utilizing the username/password for specifying the identity or through SSL protocol which validates the client certificate in the MQTT server. IP address, as well as the digital certificate of the MQTT broker/server, is used to identify the MQTT broker/server. Encrypted communication is not given by the MQTT protocol itself. MQTT brokers provide the authorization security objects that restrict the connection only for the authorized client and rules control the node to publish or subscribe to the authorized topics. MQTT has various authentication and data encryption protection mechanisms [29] that are not supported and/or configured by default. However, the security of the MQTT protocol is based on a non-encryption authentication mechanism

Table 7: MQTT security mechanism at each layer

Layer	Application	Transport	Network	Perception
Protocol used	MQTT	TCP	IPv6, RPL	IEEE 802.15.4 PHY, MAC
Security protocol	Not fixed	TLS	IPSec	IEEE 802.15.4 security
Confidentiality	No	Yes	Yes	Yes
Integrity	No	Yes	Yes	Yes
Authentication	Not fixed	Broker	Not fixed	Node
To be secured	Publisher-to-subscriber	Publisher-to-subscriber	Node-to-node	Air interface

[24,86]. Sensitive data can be extracting from data in-transit by an attackers through traffic analysis. Information such as public IP address of the MQTT broker, Port number, and payload data of the nodes are mostly targeted by the sniffing attacks. list of threats that could be targeting MQTT due to its security gaps are detailed below [29].

- Data Privacy: MQTT has no embedded mechanism for data encryption, by default. Whatever the authentication method is used by the broker or not, when data is transmitted between the broker and MQTT node the data still can be sniffed by the intruder.
- Authentication: The traffic can be sniffed by the intruder via publisher “Connect” packet node if intruder with publisher are at the same network. The username and password which will be utilized to make connection with broker are existing in such packet. In addition, “Keep Alive” packet can be tracked by the intruder. MQTT header is attached in this packet during the process of the authentication which indicates how long the MQTT broker connection will remain with IoT node. Consequently, the connection will be restarted after time expires and the resend “Connect” packet is initiated.
- Data integrity: Data could be altered while transmitting from publisher and subscriber by the intruders. After the ARP poisoning is successfully executed is performed, the packages could be altered via utilizing compiled filter by the intruders to make the network connection pass via the node of intruders.
- Botnet Over MQTT: In this scenario, Shodan search engine utilized by the intruder to search for a device which is to act as broker. Next, free broker server used by the intruder to redirect the victim’s node to it. In this case, “unsecured” broker utilized by intruder as intermediate to build an IoT botnet.

The authors in [87] Considered that the MQTT security controls cannot be adopted in the IoT network. Also, the authors mentioned that any message dispatched by the broker in IoT requires having data obfuscation, dynamic context-based policies, or anonymization that must be dynamically evaluated. Moreover, the authors addressed the requirement of data protection and privacy by presenting a model-based security toolkit at MQTT layer called SecKit.

MQTT-based IoT needs to adopt a security solution to assure that the communication channel is secured. The authors [75] have replaced the adoption of SSL/TLS with a new solution called secure MQTT (SMQTT) because SSL/TLS is not viable in constrained devices that have limited resources. The new mechanism implements over lightweight attribute-based encryption (ABE) over elliptic curves. The protocol stack and the security mechanism available at each layer are presented in Table 7.

6. MQTT THREAT MODEL

The principal elements of MQTT in IoT are publishers, subscribers, brokers, and non-IoT endpoints which include a server, a personal computer (PC) or smartphone. The abilities of threat agents can be identified by performing threat modeling to deal with potential attacks against IoT systems and specify countermeasures. In [88], a threat model was depicted for IoT security as well as privacy hazards, but the authors have not included the data communication protocols threats.

The author in Security Compass [89] has explained that the threat model targets publish/subscribe communication, but the article has not specified the threats that target the MQTT protocol.

6.1 Threat Actors

The depicted threat agents in [90] that target IoT systems, are similar to MQTT threat agents. Figure 7 describes the different threat agents of MQTT.

A- Malicious internal user: The user who is already a member of the MQTT network and owning the node or having full access to the node for malicious purposes. This user attempts to learn critical messages such as certificates or even username/password as well as alter the software of the IoT node to launch various attacks. Other IoT nodes can be vulnerable to these attacks from internal user.

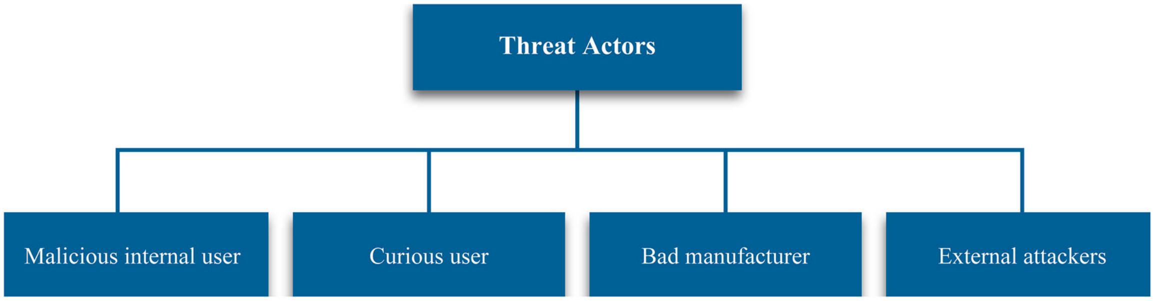


Figure 7: Threat agents in an MQTT system

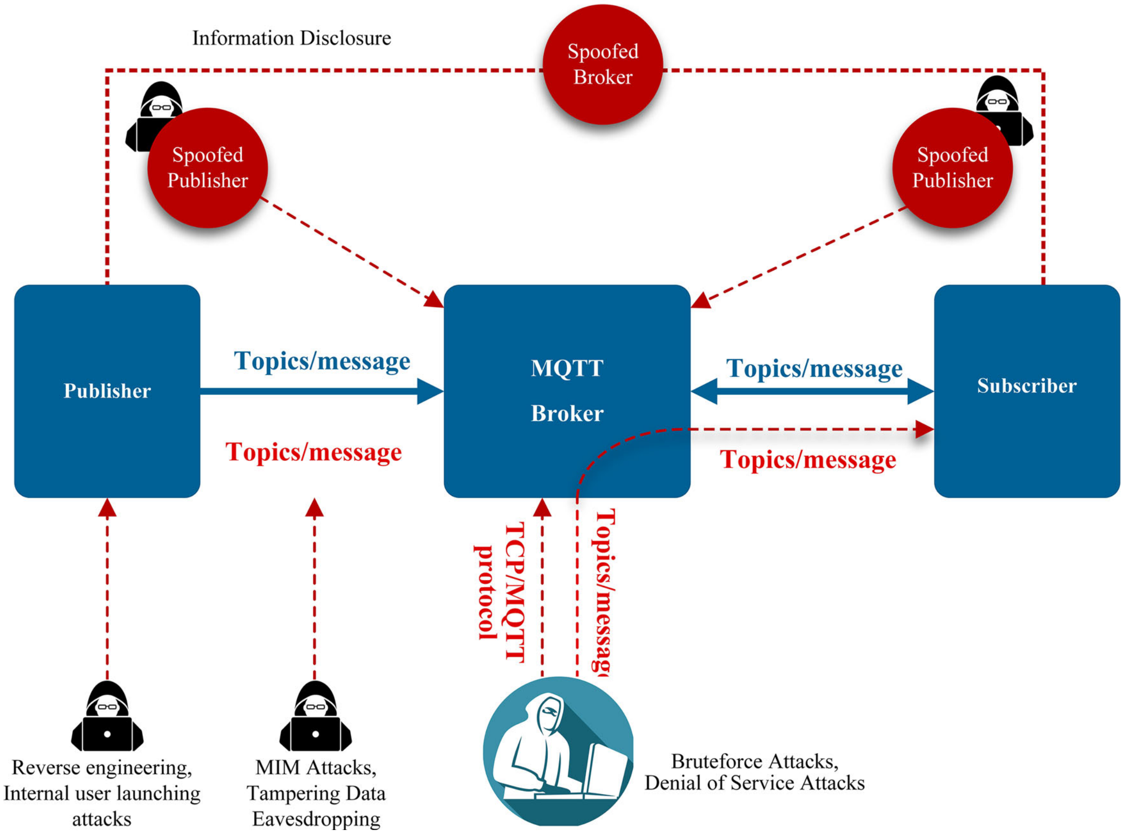


Figure 8: MQTT threat model

B- Curious user: This kind of user could be an inquisitive user or even a researcher who tries to find gaps in the MQTT network to investigate the vulnerabilities of the protocol.

C- Bad manufacture: A manufacturer can access the IoT nodes remotely or gain clients' information by injecting software on the IoT nodes deliberately. Adversaries can collect critical user data by altering the client script of MQTT with a malicious script as well as the script of MQTT broker.

D- External attackers: This kind of user is an external entity trying to make the services of the MQTT network unavailable or to gain monetary or political benefits through ransoming the MQTT nodes. They can also steal critical information of the nodes or they are cyber criminals, expert hackers, script kiddies attacking to obtain illegal access to the MQTT nodes.

6.2 Threats and Impact

MQTT-based IoT is still prone to various attacks due to vulnerabilities of the depicted security options Figure 8 describes MQTT threat model. In this section, the attacks that target MQTT are outlined.

(A) Denial of Service (DoS): This attack can disrupt broker services to coordinate and disseminate messages among MQTT nodes in the IoT arena.

1- DoS attacks-based TCP: In the arena of DoS attacks that exploit MQTT network vulnerabilities, an attacker can exhaust the bandwidth of the MQTT broker due to TCP vulnerabilities.

2- DoS attacks-based payload: An attacker can exhaust the resources of the MQTT broker by dispatching messages which more than MQTT payload size (256 MB) are resulting in denied service.

3- DoS attacks-based QoS: Adversaries can exhaust the broker resources by sending a considerable number of messages that have QoS level 2 resulting in denied service. QoS level 2 requires more resources compared to QoS level 0 and 1. Hence, QoS level 2 is denied service.

(B) Identity Spoofing: The client identifier is used by MQTT clients for connecting to the broker. However, harvesting the client's data, such as identifiers of MQTT client, certificates of MQTT client and username/password of MQTT client or fake credentials, all these can impersonate a legitimate MQTT client. Using spoofed identities by adversaries can harm the members of the MQTT network or even damage the infrastructure because the client's node will be prone to publishing/subscribing unauthorized messages from malicious clients. In addition, legitimate MQTT brokers are vulnerable to identity spoofing through illegitimate brokers that can intercept and alter the published messages or even the subscribed messages [91].

(C) Information Disclosure: MQTT broker's information must be protected from unauthorized disclosure. The traditional Internet arena and even Google Play makes the client software of MQTT available to anyone. Hence if the client software of MQTT has embedded malicious code, this will allow hackers to disclose and to capture all activities of message exchanges between all clients and even between clients and broker.

(D) Elevation of Privileges: Through unauthorized publishing/subscription occurs elevation of privileges in the MQTT arena. The wildcard feature in MQTT enables the nodes to subscribe/publish message from all topics that match the wildcard. All dispatched messages by MQTT nodes will be vulnerable to access by the attackers because of wildcard (*e.g.* #) topics.

(E) Tampering Data: Tampering in MQTT messages takes place by altering the published messages and these are received by the subscriber's node. The integrity of MQTT messages in the IoT arena is crucial because the decisions taken by many nodes are based on the last updates of the encircling environment. MQTT nodes take wrong decisions because the message is not integrated hence negatively affecting the end nodes. Moreover, tampering MQTT data is not limited to altering the published messages, but can also be dropped from the node data, log data that are stored in MQTT broker and crucial information of the session. Below Table 8 depicts the different attack vectors that can accrue to the MQTT and Mitigation Methods.

6.3 MQTT Attacks Taxonomy and Countermeasures

Due to the vulnerabilities of the MQTT protocol in the IoT arena, the protocol itself is still prone to various potential threats. Figure 9 illustrates the potential threats that target MQTT as a data communication protocol and the details of these threats are clarified in this section below.

6.3.1 TCP-based Attacks

There are various data communication protocols in the IoT stack such as MQTT, COAP, DPWS, and AMPP, unlike there are only two standard protocols in the transport layer which are UDP and TCP. Due to the MQTT work on top of TCP, here we will describe the main threats that target the TCP protocol.

TCP Port scan: A port scanning attack is among the most effective methods for exploring services for attackers to hack them. The weaknesses may be discovered by the attacker through testing each port by sending a message to them by utilizing a particular tool for or scan [92].

TCP Hijacking: Monitoring the TCP session is one of the first moves to accomplish such an attack. Throughout this context, check-sums and the sequence numbers of the communicated entities could be guessed and detected by the intruder. Furthermore, due to a lack source mechanism to recognize the packet from a legitimate node, a malicious TCP packet could be injected by the intruder which contains sequence numbers as well as check-sum expected by the recipient [93].

TCP SYN flooding: [94] states that over 90% of DoS attacks are aimed against the TCP protocol, SYN flood attack has been the most common of these. The attack is made up of a sequence of TCP SYN eavesdropped packets targeted at the port of the victim.

TCP fragmentation: In general DoS attack could be caused by fragmentation attacks in both UDP as well as TCP. These types of attacks are target reassembly

Table 8: Different attack vectors that can accrue to the MQTT and mitigation methods

Attack	Vulnerabilities	User	Vectors	Mitigation
Denial of Service	TCP	• External • Internal	Sending multiple SYN messages will create multiple half-opened TCP sessions	• Preventing DoS-based TCP attacks by utilizing firewall policies. • Handling the MQTT messages load by appropriating clients rate limit and distributed MQTT-based IoT brokers. • Blocking anomalies by employing a firewall at the application layer.
	CONNECT packets		Dispatching many MQTT CONNECT packets accompanied by diverse nodes identifiers.	
	Payload		Dispatching multiple MQTT messages accompanied by a large size of the payload.	
	QoS		Using level two of QoS beside large size of MQTT payload.	
Identity Spoofing	Client identifier	• Internal • Curious	Publishing MQTT messages with a malicious script by the legitimate IoT node or even subscribing to the unauthorized topics in the broker.	• Identify the MQTT servers and nodes can be achieved by employing Certificates. • TLS and VPN among MQTT nodes and server.
	• Brute-force • Side channel • Physical attacks	External	Pretending as a legitimate node and carry out unauthorized actions through attack IoT node to gain its information by employing brute-force attacks, physical security breach, or side channel attacks.	
	DNS service	• External • Internal	All MQTT nodes will be directed to an illegitimate broker through Spoofing the service of DNS hence all the transferred messages among all MQTT nodes will be captured by adversary's broker.	
Information Disclosure	Unsecured broker	Internal	Accessing the broker and reveals confidential information.	• Encrypting the cached data in MQTT broker and nodes. • Employing trusted frontend software for the client on smartphones as well as PC. • End-to-End ciphering. • Applying VPN between MQTT brokers and clients.
		External	Obtaining the cached information in the MQTT broker through compromising an unprotected broker.	
Elevation of Privileges	wildcard topics	• External • Internal • Curious	The adversaries can get and analyze the confidential data later through employing vulnerabilities of wildcard topics feature "#" in MQTT.	• ACL to topics. • Deactivating # subscriptions. • Anomaly detection.
	MQTT Configuration	• External • Internal	Publishing privileged messages (control message).	
Tampering Data	Unencrypted data	Internal	Modifying the usage data being sent via Hijacking the communication channel.	• Employing hash techniques to assure the MQTT messages is not altered.
		External	Alter the data cached in the broker due to broker vulnerabilities.	• Using channel encryption techniques like TLS/SSL in the broker of MQTT • X509

mechanisms of the TCP. In such a scenario, the intruder prevents fragmented packets from achieving reassembly. The result is, the targeted server could eventually fail and be overwhelmed entirely. Table 9 exhibits TCP-based attacks with the corresponding countermeasures and the achieved security goals.

6.3.2 MQTT protocol-based attacks

In the IoT paradigm, MQTT acts a significant role. CoAP and MQTT [98] are the most prevalent protocols in IoT. Below is exhibited a brief description of the most common attacks that makes the MQTT protocol a target for the adversary in the IoT.

Table 9: TCP-based attacks with associated counteractions and security objectives

TCP-based attacks	Security objectives	Counteractions
Port scan	A,AC,AU,NR,P	External firewall, IDS [92]
Hijacking	P,I,AU,TW,NR, C	Utilizing encrypted transport protocols like SSL,SSH,IPSec [95].
SYN flooding	A,AC,AU,NR,P	Firewalls, SYN cookies,SYN Cache mechanism, ACL capability, switches and routers with rate-limiting [96].
fragmentation	A,AC,AU,NR,P	[97] a secure proxy, whitelisting/Blacklisting technique.

Pre-shared key attack: Security protocols rely on pre-shared keys in some IoT implementations, including the MQTT protocol. The keys are hard-coded inside the code

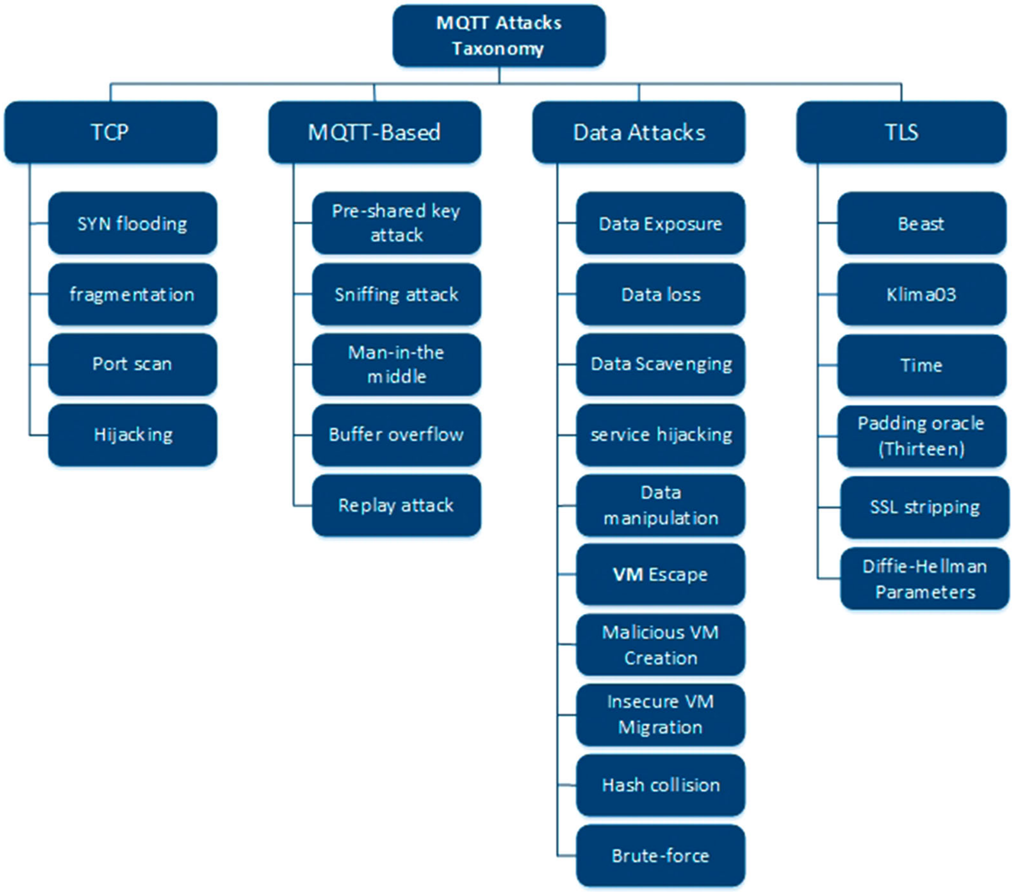


Figure 9: Taxonomy of MQTT protocol attacks.

in some situations. Thus, if the library files are reached by the attacker then he will gain access to this key [99].

*Sniffing attack***:** Most MQTT protocol implementation has no security mechanism by default which makes the sensitive data sniffed by the attacker via sniffer applications or even monitoring the traffic of the MQTT network [100].

SSL stripping: Moxie Marlinspike [101] initially stated Secure Socket Layer (SSL) stripping. The key objective of such an attack is to exploit unencrypted protocols to demand the utilization of TLS to strip out (SSL/TLS).

Beast: Even with the security implementation of the MQTT base defense SSL/TLS, still the protocol could be suffering from these attacks [102]. This attack is highly based on TLS 1.0 vulnerabilities being exploited, as Cipher Block Chaining (CBC) is deployed. After utilized MQTT over TLS, either part of message could be decrypted by the attacker through utilizing the CBC [103].

Diffie-Hellman Parameters: When the exchange uses the pre-shared key via Diffie-Hellman as well as Elliptic Curve Diffie-Hellman parameters are used, cross-protocol attacks will affect any versions of TLS and become vulnerable to this kind of attack [104].

Klima03: This is a form of RSA-related attack and certificate on TLS. All session keys elicitation process totally dependent on premaster-secret value. Once an intruder holds a premaster-secret value [105], all captured SSL/TLS may be decrypted.

Time: It is a compression attack type, where the cipher-text could be compromised by the adversary via employing TLS-level compression with the TLS [106].

Padding oracle (Thirteen): When any version of TLS utilizes the MAC-then-encrypt this will result in this kind of attack. The ciphertext could be compromised by a timing side channel which is a new type of attack known as a thirteen [107].

Man-in-the middle (MITM): The authentication process of MQTT dispatching the credentials as plaintext without utilizing any encryption method, hence this will result in vulnerabilities in the protocol itself which results in this type of attack [108]. The design of this attack was done by the authors in [109], as well as they demonstrate how hard to detected by the conventional defence mechanisms to stop the serious damage that is affected by this kind of attack.

Buffer overflow: This type of attack could be occurring as the result of the opening port of MQTT protocol [110]. The study in [111] analyzed the MQTT vulnerabilities, where inadequate mechanisms of the authorization/authentication, as well as insufficient validation/parsing of messages, leads to a lot of vulnerabilities in the MQTT protocol. An MQTT node could be crashed or even arbitrary code could be performed remotely by the adversary due to the typical weakness of the MQTT which is buffer overflow. SERVER-OTHER Eclipse Mosquitto MQTT SUBSCRIBE request topic parsing buffer overflow attempt (CVE-2019-11779).

Replay attack: The execution of the data unit instructions of the protocol is one of the heavily relies by this attack to performing an attack (ISO/IEC1443). In this case, the request of victim's reader will be dispatched by this attack to the individual malicious and as quickly as possible its response will be replayed back [112]. Below Table 10 shows MQTT threats with their impacted security objectives and counteractions.

6.3.3 Data at Rest-based Attacks

This section will implicitly discuss all potential risks and threats that targeting MQTT data locally or remotely. Below is a quick overview of all threats targeting the rest of the MQTT data.

Data exposure: An IoT data is vulnerable to numerous threats when it is stored centrally in data centers without their holders being supervised. The number of threats will grow, when malicious entities have access to such data as the lack of encryption and key management indicates that they are not appropriately protected [114]. In particular, data can be stored in multiple data centers located in separate geographical regions that have a large capacity to reach these data without their holders' permission [115].

Data loss: Prevention of data loss should be supplied by the cloud providers as well as MQTT nodes to deal with data loss probability that is resulting in such threat knows as a ransomware attack which causes serious outcomes [116]. Moreover, as stated by [117], if *n* objects are accessible on the network, workload as well as data may be transmitted through *n* of connections via the MQTT protocol.

Table 10: Exhibits MQTT threats with their impacted security objectives and counteractions

Attacks	Security objectives	Counteractions
Pre-shared key attack	P,I,AU,TW,NR, C	The use of the ephemeral keys as in ECDH key exchange guarantees PFS [98]
Sniffing attack	C, NR, P	TLS [105]
SSL stripping	P,I,AU,TW,NR, C	HTTP Strict Transport Security (HSTS) [112]
Beast	P,I,AU,TW,NR, C	Authenticated encryption algorithm like AES-GCM [108]
Diffie-Hellman Parameters	P,I,AU,TW,NR, C	The of predefined DH groups [111]
Klima03	P,I,AU,TW,NR, C	TLS 1.1, [110]
Time	P,I,AU,TW,NR, C	Disabling TLS compression [109]
Padding oracle (Thirteen)	P,I,AU,TW,NR, C	The encryption-then-MAC instead of the TLS default of MAC-then-encryption [108].
Man-in-the middle	C, I, P, NR	Secure MQTT [106]
Buffer overflow	P,I,AU,TW,NR, C	Close the opening ports, awareness of security [39]
Replay attack	A,AC,AU,NR,P	Timestamps and wake up patterns, IDS [113]

Account/service hijacking: Account hijacking could be potentially used due to feeble passwords as well as social engineering. The crucial data could be compromised, exploited, and redirected by the adversary [118]. Numerous services are provided by the cloud environment through employing application program interfaces like HTTP, SoAP, and REST. Nevertheless, such interfaces have various issues, validation of the input data, inadequate authorization inspections, and weak passwords these are the quite notable issues facing such interfaces [119]. The notable vulnerabilities of the MQTT protocol are Hijacking and MITM attacks. In addition, employing the certificates in MQTT is a complex way to achieve authentication of the nodes, and IoT topics ownership is not possible to validate as well as permissions assignment of the publisher and subscriber [120].

Data scavenging: As stated in [121], if the IoT data is not removed or destroyed adequately, the network could be prone to several threats.

Data leakage: The key effect of this threat is the absence of secure techniques to process, store, and transfer data, *i.e.* unencrypted data stored either on the IoT nodes or on the cloud [122].

Data manipulation: There are two ways for achieving data manipulation illegally at rest; the first way is by exploiting numerous API vulnerabilities such as cross-site scripting, and SQL injection, and the second way by exploiting the advantage of the security methods weakness like a small password [122].

Virtual Machine (VM) Escape: In this scenario, the vulnerabilities of the hyper-visor may be exploited by the VM escape. The aim of the threat is to control the underlying infrastructure that meets the requirements of businesses for its consistency in code complexity and configuration [121]. MQTT protocol is the most excellent, effective choice in environments such IoT cloud due to its lightweight nature of its message exchange and the use of publish/subscribe paradigm particularly within limited resource objects [21]. but VM escape is one of the notable threats which can exposed cloud IoT [123].

Malicious VM creation: The deployed VM images are too many in the environments which are unattended, therefore the adversary could attached such Trojan horse which is an example of the malicious code which could be attached in his legitimate VM account which was build previously [122].

Insecure VM migration: Illegally the data could be compromised by the adversary during the process of the immigration of a VM to a trusted or malicious host resulting to the exhibit its information to the network [124].

Brute-force attack: This is a trial and error method performed by the attacker to extract information such as personal identification number (PIN) or the passcode of the user. Automatic tools are utilized by the intruder to produce multiple sequential guesses to compromise the ciphertext [125]. Moreover, as elaborated in [126], the credentials of the user which is used by the MQTT could be retrieved through running possible tries by the attacker during the authentication phase.

Hash collision: Detecting the same hash value which is obtained by the hash function of two input strings is the critical key aim of the collision attack. As noted in [127], two different inputs are the potential to result in the same output if the hush function consists of the lengths of the variable input and the lengths of the short fixed output. This scenario is called a collision. Regarding MQTT, the vulnerable hush function which may be applied to secure MQTT could result in such an attack. Below Table 11 presents the details of the security objects achievements and counteractions related to the data-based attack.

Table 11: Security objects achievements and counteractions related to the data-based attack

Attacks	Security objectives	Counteractions
Data Exposure	C, I, P	Effective encryption systems and schemes of key management [114].
Data loss	ALL	Efficient storage and management, key generation, backup and retention strategies, and destruction practices [128].
Data Scavenging	C, I, P	Private key Cryptography [129].
Data leakage	C,I	Encryption [129], homomorphic encryption [130], and digital signature
service hijacking	ALL	Dynamic credentials, identity and access management guidance [131].
Data manipulation	ALL	Firewall [132].
VM Escape	ALL	None [129]
Malicious VM Creation	ALL	Mirage [129]
Insecure VM Migration	ALL	The protection offered by VNSS by virtual machine live migration [133], protection aegis for live migration of VMs (PALM) [134].
Hash collision	C, I	Hush function such as SHA-2, SHA-3, or bcrypt [135].
Brute-force	C, I	Brute force scanners, detection tools, IP address lock-out, and lockout technique [136].

7. EVOLUTION OF IOT MALWARE

A large number of IoT nodes interacts via MQTT protocol and are prone to infection due to default settings. For that, IoT malwares evolution will be expounded in this section briefly in recent years. Due to the modification of the known malwares by cyber criminals as well as script kiddies to take the advantages of the latest vulnerabilities or even to make the new IoT devices their victim, hence the IoT botnet malwares expounded below will be not comprehensive due to these reasons.

- *Linux/Hydra:* This is the first malware reported to attack IoT devices. It has been published in 2008 as an open source botnet platform. It is planned to be extensible and has a spreading function as well as DDoS [137,138].
- *Psyb0t:* This type of attack exploited DSL modems, as well as routers, were identified in 2009. Estimated 100,000 infected machines have been malware compromised. Psyb0t is run by command-and-control servers based on Internet Relay Chat (IRC). SSH as well as Telnet access are key methods of infecting IoT nodes utilized by Psyb0t through a basic brute force attack with 6000 usernames and 13000 passwords [139].
- *Chuck Noris:* It's an IRC bot reported in 2010 to infect routers and DSL modems. Chuck Noris Psyb0t-like, could be take the advantage of D-Link routers bypass vulnerability and it extends its surface area by brute forcing passcode [140].

- *Tsunami:* This is another IRC bot that manipulates the setting of the DNS server for the configuration of compromised machines to guide IoT traffic to malicious attacker-controlled servers [138].
- *LightAidra/Aidra:* This bot is an IRC-based mass scanning and supports exploitation tool for different architectures, such as ARM. Malware is built to look for open telnet ports accessible using established default credentials [141]. Its source code can be openly viewed on the Internet as an open source project [142].
- *Carna:* The key aim of this bot is to determine internet range and to achieve an approximation of the use of IP address. Data were gathered via Internet-activated IoT devices, in particular routers with unused or standard credentials [143]. Carna normally searches for LightAidra from compromised IoT nodes and tries to remove block ports and delete files used for contact with LightAidra. Carna Botnet was released from March 2012 to December 2012.
- *Linux-Darlloz:* This is known as an ioT Worm of Symantec [144] which spreads across default and popular credential lists by utilizing an old PHP vulnerability to reach privilege escalation and system. Similar to LightAidra, different architectures like ARM architectures are supported. After the node is infected, the Telnet traffic will be dropped by IP tables and the telnet procedure will prevent users from accessing the infected node via Telnet. Symantec noticed that by February 2014 Darlloz compromised more than 31000 devices[145]. A modern release of Darlloz utilizes compromised devices to mine cryptocurrencies [145]. Similar to Carna botnet, Darlloz threatens LightAidra in particular. It tries to remove blocks and files any LightAidra connectivity ports.
- *Linux.Wifatch: It* is an open-source malware that infects IoT devices with weak or default credentials. Once infected, it removes other malware and disables telnet access while logging the message "Telnet has been closed to avoid further infection of this device. Please disable telnet, change telnet passwords, and/or update the firmware." in the device logs. Wifatch uses peer-to-peer network to update the malware definition and deletes remnants of malware that remain in the IoT devices [146].
- *The Moon:* It is an IoT worm identified in February 2014 by Johannes Ullrich of SANS. This bot threatens routers of the Linksys, and does use a command execution vulnerability when scanning the value of the ′ttcp ip′ parameter dispatched in a POST request. The malware's Command-and-Control servers C2s will employ SSL for end-to-end interaction with their bots [147].
- *Spike Dofloo:* It was reported to exploit Linux based PCs, Windows family, and IoT nodes operating on ARM and MIPS architectures in the middle of 2014. It was used in a variety of attacks against organizations in the US and Asia. Akamai reported that the height of one of the attacks was 215Gbps [148]. This bot seems to be a gang originating from China that can release activities with different payloads like GET floods, DNS query floods, UDP floods, and SYN floods against target groups.
- *BASHLITE / Lizkebab / Torlus / gafgyt:* It is Linux based IoT devices attack which is one of the most prevalent malware to launch DDoS attacks. It was stated that out 1 million IoT devices, primarily Internet-enabled cameras, and DVRs, were enslaved by BASHLITE [149]. It can launch attacks of up to 400 Gbps. TCP, UDP, and HTTP are notable of the most threats in BASHLITE. The telnet access of the IoT nodes brute-forcing by BASHLITE utilizes known default passwords/usernames. A fascinating feature of BASHLITE is that malware payload in IoT nodes has hard-coded and simpler to monitor BASHLITE's C2s IP addresses. The majority of compromised devices are reported in Brazil, Columbia, and Taiwan. The BASHLITE source code was partially released in early 2015 and resulting in numerous releases. BASHLITE is identified to be Mirai's predecessor and competes explicitly for insecure IoT real estate.
- *KTN-RM / Remaiten:* It is a hybrid IoT malware that mixes the BASHLITE and Tsunami functionality [150]. It compromises Linux-based IoT nodes via brute forcing utilizing commonly default credentials in the list. C2s communicates with bots from a true IRC channel with IRC interactions. It is much more complex than the malware derivative BASHLITE and Tsunami. This malware may adapt to the form of attacks it needs to conduct depending on the design of the IoT nodes [150].
- *Mirai:* Mirai is among the most common malware in the world for its incontrovertible effect on the 2016 DDoS Attack [151]. It is malware based on Linux which has transformed millions of Linux systems and other IoT nodes into a bot. The largest DDoS attack reported to date is carried out by Mirai on a French hosting supplier targeting more than 15 million IoT nodes with 1 Tbps flood speed. The most incredible and an obstacle is that the source code is freely accessible online and therefore provides openings for more illegal activities by improvising the code [152]. That has 62–68 default login credentials that are specified in its code to attempt and brute-force the log-in system to reach unsecured IoT nodes.

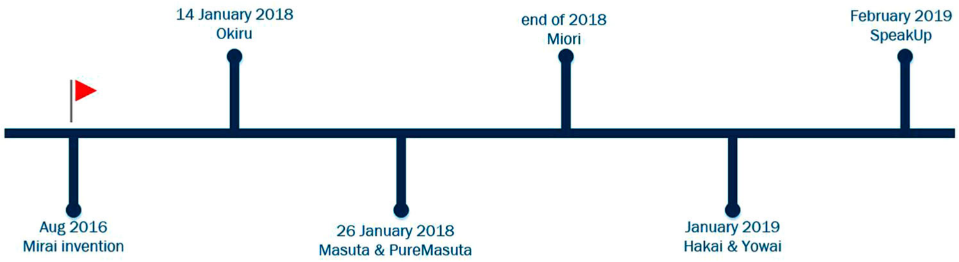


Figure 10: Latest progress timeline of Mirai versions

A large number of IoT nodes interact via MQTT protocol and are prone to infection due to default settings. When compromised, a command and control server is monitored, which implies an attack objective. Mirai as a threat to the internet of things (IoT) was not stopped until the actors were arrested. Although another intruder has the freely shared source code of Mirai malware online to utilize it or improve Mirai in newer forms and extend the botnet node to unaffected IoT devices before. Details on these versions' recent progress are offered below.

A new release of Mirai known as "Okiru" reported on 14 January 2018 targeting for the first time ARC processors-based Linux devices as well as embedded processors such as ARM. The second most prevalent embedded 32-bit processor is the Argonaut RISC Core Processor (shorted: ARC Processors), delivered in more than 1.5 billion items annually, this covers cameras, TVs, hard drives, laptop computers, networking equipment, radio, servers, utility meters, telephone, automotive, and Internet of Things. Linux runs only a relatively few ARC-based machines and is hence vulnerable to Mirai.

A Mirai successor is reported to have been modified to hijack Crypto-monetary mining operations on 18 January 2018.

Two related Mirai version bots were recorded on 26 January 2018, the most updated version that weaponized routers such D-Link EDB 38722 for obtaining extra vulnerable IoT nodes. The Home Network Administration Protocol (HNAP) vulnerability is used to create a malicious query for infected routers that may escape authentication and inappropriate deserialization execution. The less updated model of Mirai is named "Masuta" while the more further updated is named "PureMasuta"

Also at end of 2018, a Mirai model recognized as "Miori" began to distribute in a ThinkPHP system, impacting releases 5.0.23–5.1.31, via a remote executing weakness. The further enhanced Mirai forms classified as "Yowai" and "Hakai" in January 2019 and the form "SpeakUp" in February 2019 are continually exploiting this weakness. Figure 10 illustrates the latest progress timeline of Mirai malware versions in the recent year.

- *Linux/*IRCTelnet*:* This is the recent IRC botnet ELF malware targeting IoT devices with IPv6 capability. It was detected in 2016 by Malware Must Die. IRCTelnet is a combination of the tsunami idea for IRC protocol, BASHLITE for targeting strategies, and IoT credential list on Mirai botnet. LightAidra/simple Aidra's source code is employed to develop the updated botnet malware. The botnet utilizes TCP, UDP flooding, and other sequences of IPv6 and IPv4 threat techniques. The newest malware has the additional IP spoof function both in IPv6 and IPv4.
- *Wirex:* New and various service suppliers, as well as content delivery networks (CDNs), have been reporting a DDoS attack that was carried out adopting a botnet known as Dubbed Wirex. It involved thousands of Android devices running applications that appeared legal, but actually malware. Google, Oracle Dyn, Flashpoint, Cloudflare, Akamai, and several other organizations contributed to combat with such botnet. Malware in hundreds of applications has been deleted by Google from a variety of Play Store devices [153].
- *Reaper:* Since Mirai could only use devices with default credentials to improve usability, Reaper Botnet is capable to capable of takeing advantage of other ample weaknesses on IoT nodes [154]. This malware has been impacted by many well-known routers of Netgear, D-link, Cisco Linksys, as well as the Internet-connected cameras.
- *Torii:* Recently, Dr. Bontchev reported Torii malware on his honeypot "Tor" exit nodes [155]. It was ready to compromise many of today's state-of-the-art tablets, smartphones, computers with ARM, x86, MIPS, x86 (64-bit), etc. Like Mirai, it often aims for a telnet port to hack through bad passwords but is quite powerful because of its ability to retrieve the specific payload for infecting those with common architectures.

Table 12: IoT malware capabilities with its potential targets

Botnet	Year	Source code	functionality	Target	Architecture Model
Linux/Hydra	2008	Open Source	DDoS	IoT devices	IRC-Based
Psyb0t	2009	Reverse Eng.	DDoS	Routers & modems	IRC-Based
Chuck Noris	2010	Reverse Eng.	DDoS, Data Stealing	D-Link routers	IRC-Based
Tsunami/ Fbot	2018	Reverse Eng.	DDoS	DNS provider	IRC-Based
LightAidra/Aidra	2012	Open Source	Telnet-based attacks	ARM-based devices running Linux	IRC-Based
Carna	2012	Reverse Eng.	Map all ipv4 IP addresses	routers	IRC-Based
Linux.Darlloz	2013	Open Source	Delete files, mine cryptocurrencies	Routers, security cameras, set-top boxes	IRC-Based
Linux.Wifatch	2014	Open Source	Removes other malware and disables telnet access	IoT devices	IRC-Based
TheMoon	2014	Open Source	Credential brute-forcing, general traffic obfuscation	Routers and modems: Linksys,ASUS, MikroTik and D-Link	IRC-Based
Spike / Dofloo	2014	Reverse Eng.	DDoS, MITM, cookie hijacking	Routers	Agent-Handler
BASHLITE/Lizkebab / Torlus / gafgyt	2015	Open Source	DDoS	Linux based IoT devices	Agent-Handler
KTN-RM / Remaiten	2016	Reverse Eng.	DDoS	Linux based IoT devices	IRC-Based
Mirai	2016	Open Source	Application-layer attacks, volumetric attacks,and multi-vector attacks	CCTV cameras, DVRs, and home routers	Agent-Handler
Linux/IRCTelnet	2016	Reverse Eng.	DDoS	IoT devices, routers, DVRs and IP cameras	IRC-Based
Wirex	2017		DDoS	Android devices	Agent-Handler
Reaper	2017	Open Source	DDoS, other cyber attack	IoT devices, routers of Cisco Linksys, Netgear, D-link, and surveillance cameras	Agent-Handler
Torii	2018	Reverse Eng.	DDoS, cryptojacking	Modern computers, smartphones, tablets	Agent-Handler
3ve-2018	2018	Reverse Eng.	Impersonate legitimate node	IoT devices	Agent-Handler

- *3ve-2018:* This malware is one of the most innovative digital ad fraud strategies that the collaborative operation involving WhiteOps, the FBI, Google, and other ad fraud fight organizations closed recently [156]. It has compromised more than 1.7 million PCs to make false clicks to defraud web advertisements for years. This is distinct from other botnets, as they were able to build their own botnet, invent fake release for both users as well as websites, hide its IP address via proxies, and hijack IP addresses of the Border Gateway Protocol, and trading ad fraudulent to advertizers for money. The IoT malware abilities with its potential targets are elaborated in Table 12.

In order to deal with the modern and sophisticated malware releases, a more effective defense technique is required to overcome a broad variety of threat vectors.

7.1 Threat Detection

The most critical technique for identifying threats is threat modeling with a more strategic point of view. Typically, this requires various steps to systematically understand the system which will be modeled that then recognizes possible threats. As exhibited in Figure 11, Microsoft Azure IoT noted the phases of threat modeling [157].

Optimally, depending on the functionality, IoT applications will be subdivided as well as classified as a service, a cloud gateway, a field gateway, or a device will be subdivided. As stated in [158], each one is split with its own confidence boundary and has its requirements of authentication, authorization as well as information used that will impact the process of the threat model. Each element of the system can be measured for identifying threats via utilizing the STRIDE model after modeling the system. STRIDE stands for Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, and Elevation of Privileges. Further, the possible threat could be established via using attack trees. Moreover, whereafter identifying the threats, a suitable security mechanism would be clear and can be determined its adoption and where. Next section will describe the possible solutions as its applied to IoT.

8. POSSIBLE SOLUTIONS

In this study, the focus was on IoT security issues particularly MQTT protocol. The aim of this study is to identify security issues and potential threats in the MQTT protocol and figure out the possible solution. Among the most utmost demanding tasks in the IoT arena is security in which must be successfully addressed. So technologies such as Machine learning, Blockchain, and Artificial intelligence are promising solutions and fall

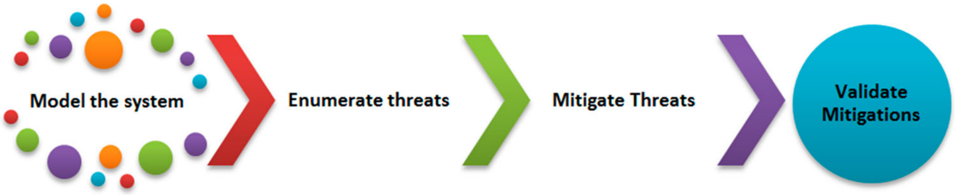


Figure 11: The threat modeling steps according to Microsoft Azure IoT.

Table 13: Depicts promising technologies that address security issues to secure MQTT in IoT

Technique	Addressed security issues	Related work to MQTT
Machine learning	Identification of unauthorized IoT nodes, Anomaly detection, Malware detection, DDOS, Intrusion detection system, Spoofing attack, false data injection, Impersonation, Eavesdropping, and Authentication	[126,159–162]
Blockchain technology	Secure communication and Data integrity, Self-healing and firmware detection, Privacy and Address space, and Access control and Information Sharing	[163–166]
Artificial intelligence	Authorization, malware detection, and privacy preservation	[167–169]

with some other new technology as a rescue to deal with the IoT security issue. Table 13 depicts these technologies with the addresses security issues to secure MQTT in IoT. Below is a brief description of such technologies as a possible solution to secure MQTT in IoT.

8.1 Machine Learning

The machine learning technology is either unsupervised or supervised. A huge immense of data is generated by the IoT module. To avoid duplication of data or malicious, data are needed before the computation process through the process of the verification. The authors in [41] elaborated the related studies on IoT security along with machine learning as a possible solution to deal with the IoT security issues. Security issues such as (Identification of unauthorized IoT nodes, Anomaly detection, Malware detection, DDOS, Intrusion detection system, Spoofing attack, false data injection, Impersonation, Eavesdropping, and Authentication) can be solved through machine learning. An innovative MEML communication protocol based on machine learning was proposed which provided the ML model offloading via MQTT protocol [159]. The authors in [160] suggested a machine learning based intrusion detection in an MQTT-based IoT system to detect normal/anomalies behavior and generates warnings for such anomalies.

8.2 Blockchain Technology

Blockchain technology is a distributing/decentralized network, each of which is linked to another in a particular form. The authors in [89] discussed in detail the application area as well as the architecture of this technology. The authors in [90,91] illustrated the method and associated studies on IoT security associated with Blockchain in the paper (91,92). Moreover, the study in [93] shows a brief explanation of the Blockchain technology uses in the IoT platform. Blockchain solutions to deal with several security challenges of IoT are elaborated in [94]. The security issue of IoT could be settled via utilizing blockchain technology. Security issues such as (Secure communication and Data integrity, Self-healing and firmware detection, Privacy and Address space, and Access control, and Information Sharing) can be solved through blockchain technology. The authors in [163] introduced a novel OTP authentication method for MQTT that utilizes Ethereum blockchain to achieve a second-factor out-of-band channel. The design allows all local and remote nodes to authenticate, ensuring trust and accountability and ensuring user privacy through Ethereum's smart contracts. Similarly, in paper [164] authors present an innovative OTP-authentication scheme for MQTT that utilizes Ethereum blockchain to implement an independent logic channel for the second-factor authentication. Additionally, the study [165] contributed to ensure the availability and privacy of MQTT data based on blockchain including the usage of cryptographic methods. Detection framework based machine learning for preventing Dos Attacks on the MQTT Broker proposed by the authors in [161].

8.3 Artificial Intelligence

Artificial intelligence is promising solutions and fall with some other new technology as a rescue to deal with the IoT security issue. Authors in paper [41] described how Artificial intelligence could address the IoT security issues. Security issues such as (authorization, malware detection, and privacy preservation) can be solved through Artificial intelligence. The study in [167] introduced the DDoS detection technique by employing the basic principles of Artificial Neural Networks for IoT. Below Table 13 depicts the promising technologies as possible solutions that address security issues to secure MQTT in IoT.

9. DISCUSSION

This section aims to provide a guideline for the researcher to create or develop new security solutions to secure the MQTT protocol within the IoT in different contexts. The MQTT is prone to numerous attacks such as DoS, identity spoofing, information disclosure, the elevation of privileges, and tampering data. Security objects of MQTT such as node authentication, message integrity, and message confidentiality will be affected by these attacks. Furthermore, nodes privacy will be impacted. Each layer of IoT depicts a solution of inbuilt rudimentary security but each layer still has vulnerabilities.

In the IoT arena, there are various devices employed, particularly in the context of resources (constrained/unconstrained). Consequently, the authentication mechanisms and classical cryptography will be not suitable for IoT nodes that have limited resources and energy such as real-time execution.

Therefore, in the context of unconstrained nodes resources, a security option such as TLS [170] must be presented to prevent adversaries from attacking the nodes of MQTT and tampering the exchange data. The best option to secure the MQTT nodes in the context of constrained resources is ciphering the MQTT payload which requires fewer resources. Table 14 illustrates the most promising security option with its security objects at each layer. Depicted studies that have been done by the research community have focused on certain points. Nevertheless, these are not yet integrated. Therefore, the presented security options of MQTT need to be developed and particularly in the context of the IoT arena that employs resource-constrained nodes. Moreover, technologies such as Machine learning, Blockchain, and Artificial intelligence are promising solutions and fall with some other new technology as a rescue to deal with the IoT security issue.

Table 14: Illustrates the existing security option with its security objects in each layer

Layer	Mechanism	Resource-constrained
Application	[20,75,113,171,172]	Yes
Network	IPsec /VPN	No
Transport	TLS [170]	No
Perception	IEEE 802.15.4, MAC	No

10. RESEARCH CHALLENGES

Publisher and subscriber need to distinguish which nodes inside/outside the MQTT network are trustworthy. Need to monitor every interaction between publisher and subscriber on the MQTT network to pursue activities that are of a malicious nature. In the MQTT network security such as end-to-end should be present between publisher and subscriber to transfer sensitive information. Specific defense choices to be employed for publisher and subscriber to fend off possible threats that target the MQTT nodes and thereby prevent any potential harm needed.

Moreover, MQTT node identity need not be revealed to another node without first making sure that node is a legitimate node. Significant attention is needed to employ fewer processes in security options to achieve less consumption of energy in MQTT-based IoT objects. A safe key should be present to be maintained and only distributed among legitimate nodes. The decision to subscribe for a particular data should be made by the MQTT node owner to make sure the data is genuine.

In addition, In the context of communication groups particularly in Ns2N (nodes to node) and N2Ns (node to nodes) complexity issues need attention such as controlling group memberships as well as employing the same ideas that are used to create nodes. High effort and attention are required to present security options for big data that are produced by IoT objects, inclusive of maintenance and transfers, and data syncing, without affecting the system. Incidents must be investigated by running procedures of digital forensics on IoT communication channels, services, or nodes which can be a tool, object, or a subject associated with the crime.

Besides, due to the grown number of IoT nodes linked through the MQTT protocol with each other, consensus algorithm issues as well as system throughput still exist. Meanwhile, working on MQTT security, the scalability of IoT must be considered due to its existing issues. Furthermore, the protection mechanism should really be built to satisfy the poor resource devices in lightweight terms.

11. CONCLUSION

This paper presented the design of IoT communications and messaging protocol with emphasis on MQTT. The concepts of the MQTT protocol and its security aspects were defined. Since MQTT lacks some security elements, it is prone to various attacks and infiltrations. This study presented in detail, a threat model of MQTT protocol with attack vectors applicable to a conventional MQTT-based IoT domain. This MQTT publish/subscribe messages are vulnerable to security threats, *i.e.* spoofing that can cause DoS attacks on MQTT broker. This study also presented existing work and current research challenges for researchers to address in the near future. Besides, the authors outlined the emerging technologies such as Machine learning, Artificial intelligence and Blockchain combine with IoT as possible solutions to secure MQTT.

REFERENCES

1. K. Rose, S. Eldridge, and C. Lyman, "The internet of things: an overview," Internet Soc., no. October, p. 53, 2015.

2. I. I. Initiative, *et al.*, "Towards a definition of the internet of things," IEEE IoT Initiat. white Pap., 2015.

3. Ericsson, "Ericsson mobility report," 2016. [Online]. Available: http//www.ericsson.com/res/docs/2016/ericssonmobility-report-2016.pdf. Accessed Sept. 2, 2018.

4. S. M. R. Islam, D. Kwak, H. Kabir, M. Hossain, and K.-S. Kwak, "The internet of things for health care: a comprehensive survey," *IEEE. Access.*, Vol. 3, pp. 678–708, 2015.

5. M. Miller. *The Internet of Things: How Smart TVs, Smart Cars, Smart Homes, and Smart Cities are Changing the World*. Indiana, USA: Que Publishing Incorporation, 2015.

6. M. Dunn, "The next generation of smart fridges," 2017, [Online]. Available: http//www.news.com.au/technology/gadgets/the-next-generationof-smart-fridges/news-story/7b75572b8cfbe90432754c8b76abc017. Accessed Sept. 2, 2018.

7. A. Rajput, "Smart cctv and the internet of things: 2016 trends and predictions," 2016. [Online]. Available: https//www.ifsecglobal.com/smartcctv-and-the-internet-of-things-2016-trends-and-predications/. Accessed Sept. 2, 2018.

8. T. Heer, O. Garcia-Morchon, R. Hummen, S. L. Keoh, S. S. Kumar, and K. Wehrle, "Security challenges in the IP-based Internet of things," *Wirel. Pers. Commun*, Vol. 61, no. 3, pp. 527–542, 2011.

9. S. Li, L. Da Xu, and S. Zhao, "The internet of things: a survey," *Inf. Syst. Front*, Vol. 17, no. 2, pp. 243–259, 2015.

10. A.-R. Sadeghi, C. Wachsmann, and M. Waidner, "Security and privacy challenges in industrial internet of things," in Proceedings of 52nd Annual Design Automation Conference – DAC '15, 2015, pp. 1–6.

11. J. T. J. Penttinen. *Wireless Communications Security: Solutions for the Internet of Things*. Chichester, UK.: Wiley, 2016.

12. securityaffairs, "Bashlite botnets peaked 1 million internet of thing devices," 2016. [Online]. Available: http//securityaffairs.co/wordpress/50824/iot/bashlite-botnets.html. Accessed Sept. 2, 2018.

13. B. Krebs, "Krebsonsecurity hit with record ddos," 2016. [Online]. Available: https//krebsonsecurity.com/2016/09/krebsonsecurityhit-with-record-ddos/. Accessed Sept. 2, 2018.

14. FPAnalyst, "Attack of things!," 2016. [Online]. Available: https//www.flashpoint-intel.com/attack-of-things/. Accessed Sept. 2, 2018.

15. kerneronsec, "Remote code execution in cctv-dvr affecting over 70 different vendors," 2016. [Online]. Available: http//www.kerneronsec.com/2016/02/remote-code-execution-incctv-dvrs-of.html. Accessed Sept. 2, 2018.

16. ProofPoint, "More than 750,000 phishing and spam emails launched from 'thingbots' including televisions, fridge," 2014. [Online]. Available http//investors.proofpoint.com/releasedetail.cfm?releaseid=819799. Accessed Sept. 2, 2018.

17. M. Charlie, and C. Valasek, "A survey of remote automotive attack surfaces," 2014. [Online]. Available http//www.ioactive.com/pdfs/IOActive Remote Attack Surfaces.pdf. Accessed Sept. 2, 2018.

18. Cisco, "Securing the internet of things: a proposed framework," 2012 [Online]. Available: http//www.cisco.com/c/en/us/about/securitycenter/Secur. Accessed Sept. 2, 2018.

19. A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, & M. Ayyash (2015), "Internet of things: A survey on enabling technologies, protocols, and applications," *IEEE communications surveys & tutorials*, Vol. 17, no. 4, pp. 2347–2376.

20. A. Niruntasukrat, C. Issariyapat, P. Pongpaibool, K. Meesublak, P. Aiumsupucgul, and A. Panya, "Authorization mechanism for mqtt-based internet of things," in 2016 IEEE International Conference on Communications Workshops (ICC), 2016, pp. 290–295.

21. B. Dorsemaine, J. P. Gaulier, J. P. Wary, N. Kheir, and P. Urien, "A new approach to investigate IoT threats based on a four layer model," in 13th International Conference on New Technologies for Distributed Systems NOTERE 2016 – Proceedings, no. Notere, 2016.

22. G. Perrone, M. Vecchio, R. Pecori, and R. Giaffreda, "The day after mirai: a survey on MQTT security solutions after the largest cyber-attack carried out through an army of IoT devices," in IoTBDS 2017 – Proceedigs of 2nd International Conference on Internet Things, Big Data Security, no. January, 2017, pp. 246–253.

23. S. T. Zargar, J. Joshi, and D. Tipper, "A survey of defense mechanisms against distributed denial of service (DDOS) flooding attacks," *IEEE Commun. Surv. Tutorials*, Vol. 15, no. 4, pp. 2046–2069, 2013.

24. S. Andy, B. Rahardjo, and B. Hanindhito, "Attack scenarios and security analysis of MQTT communication protocol in IoT system," in International Conference on Electrical Engineering, Computer Science and Informatics, Vol. 2017-Decem, no. September, 2017, pp. 19–21.

25. G. Perrone, M. Vecchio, R. Pecori, and R. Giaffreda, "The day after mirai: a survey on MQTT security solutions after the largest cyber-attack carried out through an army of IoT devices," in IoTBDS 2017 – Proceedings of the 2nd International Conference Internet Things, Big Data Security, no. IoTBDS, 2017, pp. 246–253.

26. M. B. Yassein, M. Q. Shatnawi, S. Aljwarneh, and R. Al-Hatmi, "Internet of things: survey and open issues of MQTT protocol," in Proceedings – 2017 International Conference on Engineering MIS, ICEMIS 2017, vol. 2018-January, 2018, pp. 1–6.

27. M. S. Harsha, B. M. Bhavani, and K. R. Kundhavai, "Analysis of vulnerabilities in MQTT security using Shodan API and implementation of its countermeasures via authentication and ACLs," in 2018 International Conference on Advances in Computing, Communications And Informatics, ICACCI 2018, 2018, pp. 2244–2250.

28. G. Potrino, F. De Rango, and A. F. Santamaria, "Modeling and evaluation of a new IoT security system for mitigating DoS attacks to the MQTT broker," IEEE Wirel. Commun. Netw. Conf. WCNC, Vol. 2019-April, pp. 1–6, 2019.

29. D. Dinculeană, and X. Cheng, "Vulnerabilities and limitations of MQTT protocol used between IoT devices," *Appl. Sci*, Vol. 9, no. 5, p. 848, 2019.

30. A. Lakshmanan, "Literature Review on the latest security & the vulnerability of the Internet of Things (IoT) & a Proposal to Overcome," no. April, 2020.

31. R. Da Paz, A. Sehovic, D. M. Cook, and L. Armstrong, "A novel approach to resource Starvation attacks on message queuing telemetry transport brokers," pp. 150–154, 2020.

32. T. Borgohain, U. Kumar, and S. Sanyal, "Survey of security and privacy issues of internet of things," arXiv Prepr. arXiv1501.02211, p. 7, 2015.

33. H. Zhou. *The Internet of Things in the Cloud: A Middleware Perspective*, 1st ed. Boca Raton, FL: CRC Press, 2012.

34. M. Hossain, M. Fotouhi, and R. Hasan, "Towards an analysis of security issues, challenges, and open problems in the Internet of things," in 2015 IEEE World Congress on Services, 2015, pp. 1–8.

35. S. Singh, and N. Singh, "Internet of things (IoT): security challenges, business opportunities & reference architecture for E-commerce," in 2015 International Conference on Green Computing and Internet Things (ICGCIoT), 2015, pp. 1577–1581.

36. P. P. Ray, "A survey on Internet of Things architectures," *J. King Saud Univ. – Comput. Inf. Sci.*, Vol. 30, no. 3, pp. 291–319, 2018.

37. G. Enabler, "Market pulse report, internet of things (IOT). Discover key trends and insights on disruptive technologies in IOT innovations." 2017.

38. M. S. Report, "Industrial IoT (IIoT) Market Size &amp; Forecast to 2026." 2019.

39. Q. Gou, L. Yan, Y. Liu, and Y. Li, "Construction and strategies in IoT security system," in Proceedings of the 2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing. GreenCom-IThings-CPSCom 2013, 2013, pp. 1129–1132.

40. Y. Wang, and X. Zhang. Internet of things: international workshop, IOT 2012, Changsha, China, August 17–19, 2012. Proceedings, Vol. 312. Springer, 2012.

41. B. K. Mohanta, D. Jena, U. Satapathy, and S. Patnaik, "Survey on IoT security: challenges and solution using machine learning, artificial intelligence and blockchain technology," *IoT*, Vol. 11, p. 100227, 2020.

42. V. Casola, A. De Benedictis, A. Riccio, D. Rivera, W. Mallouli, and E. M. de Oca, "A security monitoring system for internet of things," *IoT*, Vol. 7, p. 100080, 2019.

43. A. Al-Hasnawi, S. M. Carr, and A. Gupta, "Fog-based local and remote policy enforcement for preserving data privacy in the internet of things," *IoT*, Vol. 7, p. 100069, 2019.

44. S. Siboni, V. Sachidananda, Y. Meidan, M. Bohadana, Y. Mathov, S. Bhairav, A. Shabtai, Y. Elovici, "Security testbed for internet-of-things devices," *IEEE Trans. Reliab.*, Vol. 68, no. 1, pp. 23–44, 2019.

45. K. C. Chen, and S. Y. Lien, "Machine-to-machine communications: technologies and challenges," *Ad. Hoc. Netw.*, Vol. 18, pp. 3–23, 2014.

46. R. Nawaratne, D. Alahakoon, D. De Silva, P. Chhetri, and N. Chilamkurti, "Self-evolving intelligent algorithms for facilitating data interoperability in IoT environments," *Futur. Gener. Comput. Syst*, Vol. 86, no. 2018, pp. 421–432, 2018.

47. M. R. Hosenkhan, and B. K. Pattanayak, "Security issues in internet of things (IoT): a comprehensive review," *Adv. Intell. Syst. Comput.*, Vol. 1030, no. 4, pp. 359–369, 2020.

48. I. Butun, P. Osterberg, and M. Gidlund, "Preserving location privacy in Cyber-Physical systems," in 2019 IEEE Conference on Communications and Network Security. CNS 2019, 2019, pp. 1–6.

49. I. Butun, and M. Gidlund, "Location privacy assured internet of things," in ICISSP 2019 – Proceedings of the 5th International Conference Information Systems Security and Privacy, no. Icissp 2019, 2019, pp. 623–630.

50. K. Zhao, and L. Ge, "A survey on the internet of things security," in Proceedings– of the 9th International Conference on Computational Intelligence and Security (CIS) 2013, 2013, pp. 663–667.

51. H. Upadhyay, H. B. Patel, and T. Sherasiya, "A survey: intrusion detection system for Internet of things," *Int. J. Comput. Sci. Eng.*, Vol. 5, no. 2, pp. 91–98, 2016.

52. T. Zhang, and X. Li, "Evaluating and analyzing the performance of RPL in contiki," in Proceedings of the First International Workshop on Mobile Sensing, Computing and Communication – MSCC '14, 2014, pp. 19–24.

53. "Top IoT Vulnerabilities." [Online]. Available: https://www.owasp.org/index.php/Top_%0AIoT_Vulnerabilities. Accessed Mar. 4, 2020.

54. C. Kolias, G. Kambourakis, A. Stavrou, and J. Voas, "DDoS in the IoT: Mirai and other botnets," *Computer (Long. Beach. Calif.)*, Vol. 50, no. 7, pp. 80–84, 2017.

55. A. Mosenia, and N. K. Jha, "A comprehensive study of security of internet-of-things," *IEEE Trans. Emerg. Top. Comput.*, Vol. 5, no. 4, pp. 586–602, 2017.

56. Y. Cherdantseva, and J. Hilton, "A reference model of information assurance & security. availability, reliability and security (ARES)," in Proceeding of the 18th International Conference, 2013, pp. 1–11.

57. OASIS, "Devices profile for web services version 1. 1," OASIS Mai, no. July, pp. 1–43, 2009.

58. R. Chinnici, J.-J. Moreau, A. Ryman, and S. Weerawarana, "Web services description language (wsdl) version 2.0 part 1: core language," *W3C Recomm.*, Vol. 26, no. 1, p. 19, 2007.

59. D. Box, *et al.*, "Simple object access protocol (SOAP) 1.1," 2000. [Online]. Available: https://www.w3.org/TR/2000/NOTE-SOAP-20000508/. Accessed Feb. 3, 2021.

60. T. Cucinotta, A. Mancina, G.F. Anastasi, G. Lipari, L. Mangeruca, R. Checcozzo, and F. Rusina, "A real-time service-oriented architecture for industrial automation," *IEEE Trans. Ind. Inform.*, Vol. 5, no. 3, pp. 267–277, 2009.

61. P. Spiess, *et al.*, "Soa-based integration of the internet of things in enterprise services," in 2009 IEEE International Conference on Web Services ICWS 2009, 2009, pp. 968–975.

62. P. Saint-Andre, "Extensible messaging and presence protocol (XMPP): Core," 2011.

63. IETF, "DNS-based service discovery," IETF, Internet-Draft, Des. 2011. [Online]. Available https//tools.ietf.org/html/draft-cheshire-dnsext-dns-sd-11. Accessed Sept. 2, 2018.

64. S. Cheshire, and M. Krochmal, "Multicast DNS," IETF, Internet-Draft, Des. 2011. [Online]. Available https//tools.ietf.org/html/draft-cheshire-dnsext-multicastdns-15. Accessed Sept. 2, 2018.

65. Z. Shelby, K. Hartke, and C. Bormann, "The Constrained Application Protocol (CoAP)," pp. 1–112, 2014.

66. S. Raza, H. Shafagh, K. Hewage, R. Hummen, T. Voigt, "Lithe: lightweight secure CoAP for the internet of things," *IEEE Sens. J.*, Vol. 13, no. 10, pp. 3711–3720, 2013.

67. T. A. Alghamdi, A. Lasebae, and M. Aiash, "Security analysis of the Constrained Application Protocol in the Internet of things security analysis of the Constrained Application Protocol in the Internet of things," in 2013 Second International Conference on Future Generation Communication Technology (FGCT), no. November, 2013, pp. 163–168.

68. IBM and Eurotech, "MQTT v3.1ProtocolSpecification," 1999. [Online]. Available: http//public.dhe.ibm.com/software/dw/webservices/ws-mqtt/mqtt-v3r1.html. Accessed Sept. 2, 2018.

69. O. Standard, "MQTT version 3.1. 1," URL http//docs.oasis-open. org/mqtt/mqtt/v3, vol. 1, 2014.

70. D. Chen, and P. K. Varshney, "QoS support in Wireless sensor networks: A survey," in International Conference on Wireless Networks, (ICWN '04), Las Vegas, Vol. 13244, 2004, pp. 227–233.

71. M. HiveMQ Enterprise, "Broker, 'MQTT security fundamentals: TLS/SSL,'" 2015. [Online]. Available: http//www.hivemq.com/blog/mqtt-security-fundamentalstls-ssl Accessed Sept. 2, 2018.

72. I. Skerrett, "IoT Developer Survey 2016," Eclipse IoT Work. Group, IEEE IoT Agil. IoT, 2016.

73. ISO/IEC and 20922 2016, "MQTT v3.1.1," 2016. [Online] Available: https//www.iso.org/standard/69466.html. Accessed Sept. 2, 2020.

74. V. Lampkin, *et al.* Building smarter planet solutions with MQTT and IBM websphere MQ telemetry. IBM Redbooks, 2012.

75. M. Singh, M. A. Rajan, V. L. Shivraj, and P. Balamuralidhar, "Secure MQTT for internet of things (iot)," in 2015 Fifth International Conference on Communication Systems and Network Technologies (CSNT), 2015, pp. 746–751.

76. V. Karagiannis, P. Chatzimisios, F. Vazquez-Gallego, and J. Alonso-Zarate, "A survey on application layer protocols for the internet of things," *Trans. IoT Cloud Comput.*, Vol. 3, no. 1, pp. 11–17, 2015.

77. N. De Caro, W. Colitti, K. Steenhaut, G. Mangino, and G. Reali, "Comparison of two lightweight protocols for smartphone-based sensing," in IEEE SCVT 2013 – Proceedings of the 20th IEEE Symposium on Communications and Vehicular Technology in the BeNeLux, 2013, pp. 0–5.

78. K. Fysarakis, I. Askoxylakis, O. Soultatos, I. Papaefstathiou, C. Manifavas, and V. Katos, "Which IoT protocol?," in 2016 IEEE Global Communications Conference, 2016.

79. D. Thangavel, X. Ma, A. Valera, H. X. Tan, and C. K. Y. Tan, "Performance evaluation of MQTT and CoAP via a common middleware," in IEEE ISSNIP 2014 – 2014 IEEE 9th International Conference on Intelligent Sensors, Sensor Networks and Information Process, no. April, 2014, pp. 21–24.

80. L. Dürkop, B. Czybik, and J. Jasperneite, "Performance evaluation of M2M protocols over cellular networks in a lab environment.," {Icin}, pp. 70–75, 2015.

81. M. Collina, G. E. Corazza, and A. Vanelli-Coralli, "Introducing the QEST broker: scaling the IoT by bridging MQTT and REST," in IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), 2012, pp. 36–41.

82. S. M. Kim, H. S. Choi, and W. S. Rhee, "Iot home gateway for auto-configuration and management of MQTT devices," in 2015 IEEE Conference on Wireless Sensors, ICWiSE 2015, 2016, pp. 12–17.

83. P. Papageorgas, D. Piromalis, T. Iliopoulou, K. Agavanakis, M. Barbarosou, K. Prekas, and K. Antonakoglou, "Wireless sensor networking architecture of polytropon: An open source scalable platform for the smart grid," *Energy Procedia*, Vol. 50, pp. 270–276, 2014.

84. J. E. Luzuriaga, J. C. Cano, C. Calafate, P. Manzoni, M. Perez, and P. Boronat, "Handling mobility in IoT applications using the MQTT protocol," in 2015 Internet Technologies and Applications ITA 2015 – Proceedings of the 6th International Conference, 2015, pp. 245–250.

85. Y. F. Gomes, D. F. S. Santos, H. O. Almeida, and A. Perkusich, "Integrating MQTT and ISO / IEEE 11073 for health information sharing in the internet of things," in 2015 IEEE International Conference on Consumer Electronics, 2015, pp. 200–201.

86. J. J. Anthraper, and J. Kotak, "Security, Privacy and Forensic Concern of MQTT Protocol," SSRN Electron. J., no. December, 2019.

87. R. Neisse, G. Steri, and G. Baldini, "Enforcement of security policy rules for the internet of things," in 2014 IEEE 10th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), 2014, pp. 165–172.

88. A. W. Atamli, and A. Martin, "Threat-based security analysis for the internet of things," in 2014 International Workshop on Secure Internet of Things (SIoT), 2014, pp. 35–43.

89. SecurityCompass, "Publish-subscribe threat modeling," [Online]. Available: https://blog.securitycompass.com/publish-subscribe-threatmodeling-11add54f1d07%.w5(9zfbr7, 2016.

90. Y. Abed, and G. Boivin, "Treatment of respiratory virus infections," *Antiviral Res.*, Vol. 70, no. 2, pp. 1–16, 2006.

91. J. Kotak, A. Shah, A. Shah, and P. Rajdev, "A comparative analysis on security of MQTT brokers," In 2nd Smart Cities Symposium (SCS 2019). pp. 1–5. IET, 2019.

92. M. University, "The Five-Layer TCP/IP Model: Description/Attacks/Defense – Computing and Software Wiki," 2008.

93. O. Zheng, J. Poon, and K. Beznosov, "Application-based TCP hijacking," in Proceedings of the 2nd European Workshop on System Security EUROSEC'09, 2009, pp. 9–15.

94. D. Moore, G. M. Voelker, and S. Savage, "Inferring internet denial-of-service activity," Proc. 10th USENIX Secur. Symp., vol. 24, no. 2, pp. 115–139, 2001.

95. K. Lam, D. LeBlanc, and B. Smith, "Theft on the web: prevent session hijacking." 2005.

96. S. Kumarasamy, and G. A. Shankar, "An Active Defense Mechanism for TCP SYN flooding attacks," arXiv.org, pp. 1–6, 2012.

97. Incapsula, "What is an IP Fragmentation Attack (Teardrop ICMPUDP) — DDoS Attack Glossary — Incapsula".

98. T. Jaffey, "MQTT and CoAP, IoT Protocols,Eclipse Newsletter," 2014.

99. S. Jucker, "Master's thesis securing the constrained application protocol by Stefan Jucker," no. October, pp. 1–103, 2012.

100. A. J. Hintaw, S. Manickam, S. Karuppayah, and M. F. Aboalmaaly, "A brief review on MQTT's security issues within the internet of things (IoT)," *J. Commun.*, Vol. 14, no. 6, pp. 463–469, 2019.

101. M. Marlinspike, "Sslstrip,Thoughtcrime Labs," [Online]. Available: http://www.thoughtcrime.org/software/sslstrip/ (2009). Accessed Oct. 2011.

102. M. S. Bernard, T. Pei, and K. Nasser, "QoS strategies for wireless multimedia sensor networks in the context of IoT at the MAC layer, application layer, and cross-layer algorithms," *J. Comput. Networks Commun.*, Vol. 2019, pp. 1–33, 2019.

103. T. D. Juliano Rizzo, "Browser exploit against SSL/TLS packet storm," 2011. [Online]. Available: https://packetstormsecurity.com/files/105499/Browser-Exploit-Against-SSL-TLS.html. Accessed Nov. 24, 2020.

104. N. Mavrogiannopoulos, F. Vercauteren, V. Velichkov, and B. Preneel, "A cross-protocol attack on the TLS protocol," in Proceedings of the ACM Conference on Computer and Communications Security, 2012, pp. 62–72.

105. V. Klíma, O. Pokorný, and T. Rosa, "LNCS 2779 – attacking RSA-based sessions in SSL/TLS," *Int. Work. Cryptogr. Hardw. Embed. Syst.*, Vol. 2779, pp. 426–440, 2003.

106. Y. Sheffer, R. Holz, and P. Saint-Andre, "Summarizing known attacks on transport layer security (tls) and datagram tls (dtls)," 2015.

107. N. J. AlFardan, and K. G. Paterson, "Lucky thirteen: breaking the TLS and DTLS record protocols," in Proceedings of the IEEE Symposium on Security and Privacy, 2013, pp. 526–540.

108. M. Wang, "Understanding security flaws of IoT protocols through honeypot technologies." Master of Science), Delft University of Technology, Netherlands. Retrieved ~ … , 2017.

109. H. Wong, "Man-in-the-Middle attacks on MQTT-based IoT using BERT based adversarial message generation," pp. 1–6.

110. S. N. Swamy, D. Jadhav, and N. Kulkarni, "Security threats in the application layer in IOT applications," in Proc. Int. Conf. IoT Soc. Mobile, Anal. Cloud, I-SMAC 2017, 2017, pp. 477–480.

111. G. Nebbione, and M. C. Calzarossa, "Security of IoT application layer protocols: challenges and findings," *Futur. Internet*, Vol. 12, no. 3, pp. 1–20, 2020.

112. M. Roland, J. Langer, and J. Scharinger, "Practical attack scenarios on secure element-enabled mobile devices," in 2012 4th International Workshop on Near Field Communication, 2012, pp. 19–24.

113. F. De Rango, G. Potrino, M. Tropea, and P. Fazio, "Energy-aware dynamic internet of things security system based on elliptic curve cryptography and message queue telemetry transport protocol for mitigating replay attacks," *Pervasive Mob. Comput.*, Vol. 61, p. 101105, 2020.

114. C. Rong, S. T. Nguyen, and M. G. Jaatun, "Beyond lightning: a survey on security challenges in cloud computing," *Comput. Electr. Eng.*, Vol. 39, no. 1, pp. 47–54, 2013.

115. N. Kaaniche, and M. Laurent, "Data security and privacy preservation in cloud storage environments based on cryptographic mechanisms," *Comput. Commun.*, Vol. 111, pp. 120–141, 2017.

116. C. Chandra, "Data loss vs. data leakage prevention what's the difference," 2017. [Online]. Available: https://blogs.informatica.com/2017/05/03/data-loss-vs-data-leakage-prevention-whats-difference/%:~:text=Data%20Loss%20Prevention,-I%20had%20thought&text=In%20a%20data%20loss%2C%20the,are%20usually%20systems%20of%20records. Accessed Feb. 03, 2021.

117. T. Authors, C. C. By-nc-nd, and C. P. Chairs, Available: www.sciencedirect.com, vol. 9, pp. 1–8, 2020.

118. C. S. Alliance, "Top threats to cloud computing V1.0." 2010. Available: https://ioactive.com/wp-content/uploads/2018/05/csathreats.v1.0-1.pdf. Accessed Feb. 5, 2021.

119. W. Dawoud, I. Takouna, and C. Meinel, "Infrastructure as a service security: challenges and solutions," in INFOS2010 – 2010 7th International Conference on Informatics and Systems, 2010.

120. P. Gallo, U. Q. Nguyen, G. Barone, and P. Van Hien, "Decymo: decentralized Cyber-Physical system for monitoring and Controlling industries and homes," in IEEE 4th International Forum on Research and Technology for Society and Industry (RTSI 2018), 2018.

121. W. A. Jansen, "Cloud hooks: security and privacy issues in cloud computing," in 2011 44th Hawaii International Conference on System Sciences, 2011, pp. 1–10.

122. B. Grobauer, T. Walloschek, and E. Stocker, "Understanding cloud computing vulnerabilities," *IEEE Secur. Priv.*, Vol. 9, no. 2, pp. 50–57, 2011.

123. H. F. Atlam, A. Alenezi, A. Alharthi, R. J. Walters, and G. B. Wills, "Integration of cloud computing with internet of things: Challenges and open issues," Proc. – 2017 IEEE Int. Conf. Internet Things, IEEE Green Comput. Commun. IEEE Cyber, Phys. Soc. Comput. IEEE Smart Data, iThings-GreenCom-CPSCom-SmartData 2017, vol. 2018-Janua, pp. 670–675, 2018.

124. J. W. Rittinghouse, and J. F. Ransome. *Cloud Computing: Implementation, Management, and security*. Boca Raton, NW, USA: CRC Press, 2016.

125. K. Jackson, "Hacker's Choice Top Six Database Attacks," 2008. [Online]. Available: https://www.darkreading.com/risk/hackers-choice-top-six-database-attacks/d/d-id/1129481. Accessed Feb. 3, 2021.

126. I. Vaccari, G. Chiola, M. Aiello, M. Mongelli, and E. Cambiaso, "MQTTset, a new dataset for machine learning techniques on MQTT," *Sensors (Switzerland)*, Vol. 20, no. 22, pp. 1–17, 2020.

127. M. Stevens, A. Lenstra, and B. De Weger, “Chosen-prefix collisions for MD5 and colliding X. 509 certificates for different identities,” in Annual International Conference on the Theory and Applications of Cryptographic Techniques, 2007, pp. 1–22.

128. C. S. Alliance, “Data Loss Prevention,” 2012. [Online]. Available: https://downloads.cloudsecurityalliance.org/initiatives/secaas/SecaaS_Cat_2_DLP_Implementation_Guidance.pdf?_ga=2.263928460.1471820541.1606222125-441287358.1606222125. Accessed Feb 3, 2021.

129. S. Chandna, R. Singh, and F. Akhtar, “Data scavenging threat in cloud computing,” *Int. J. Adv. Comput. Sci. Cloud Comput.*, Vol. 2, no. 2, pp. 106–111, 2014.

130. N. P. Smart, and F. Vercauteren, “Public key cryptography – PKC 2010,” *Lect. Notes Comput. Sci. (Including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, Vol. 6056, pp. 420–443, 2010.

131. SYBASE, “Dynamic credentia,” 2011. [Online]. Available: http://infocenter.sybase.com/help/index.jsp?topic=/com.sybase.infocenter.dc01218.0200/doc/html/vhu1249594001338.html. Accessed Feb. 3, 2021.

132. Tenable, “Tenable.io web application scanning — tenable,” 2020. [Online]. Available: https://www.tenable.com/products/tenable-io/web-application-scanning. Accessed Feb. 3, 2021.

133. “VNSS: A NETWORK SECURITY SANDBOX FOR VIRTUAL COMPUTING ENVIRONMENT Gao Xiaopeng, Wang Sumei, Chen Xianqin State Key Laboratory of Software Development Environment BeiHang University,” 2010.

134. Webopedia, “PALM,” 2021. [Online]. Available: https://webopedia.dev.wordpress.relay.cool/2020/10/13/palm-inc/. Accessed Feb. 3, 2021.

135. E. Z. Goodnight, “What Is SHAttered SHA-1 Collision Attacks, Explained,” 2017. [Online]. Available: https://www.howtogeek.com/238705/what-is-sha-1-and-why-will-retiring-it-kick-thousands-off-theinternet/#:∼:text=The%20SHA%20in%20SHA%2D1,important%20transmissions%20on%20the%20internet. Accessed Feb. 3, 2021.

136. A. VAULT, “Brute Force Attack Mitigation Methods & Best Practices.” 2016.

137. Infodox, “Hydra IRC bot, the 25 min overview of the kit,” 2011. [Online]. Available: source: http://insecurety.net/?p=90. Accessed Nov. 20, 2020.

138. M. Janus, “Heads of the Hydra. Malware for Network Devices,” 2011. [Online]. Available: https://securelist.com/heads-of-the-hydra-malware-for-network-devices/36396/. Accessed Nov. 10, 2020.

139. Psyb0t, “In Wikipedia,” 2013. [Online]. Available: https://en.wikipedia.org/wiki/Psyb0t. Accessed Nov. 10, 2020.

140. R. McMillan, “Chuck Norris botnet karate-chops routers hard,” 2010. [Online]. Available: https://www.computerworld.com/article/2521061/chuck-norris-botnet-karate-chops-routers-hard.html. Accessed Nov. 10, 2020.

141. Fitsec, “New piece of malicious code infecting routers and IPTV’s,” 2012. [Online]. Available: http://www.fitsec.com/blog/index.php/2012/02/19/new-piece-of-malicious-codeinfecting-routers-and-iptvs/. Accessed Nov. 10, 2020.

142. F. Fazzi, “LightAidra Source Code on GitHub,” 2012. [Online]. Available: https://github.com/eurialo/lightaidra. Accessed Feb. 3, 2021.

143. Anonymous, “Internet census 2012 Port scanning /0 using insecure embedded devices,” 2012. [Online]. Available: https://internetcensus2012.github.io/InternetCensus2012/paper.html. Accessed Nov. 10, 2020.

144. J. Cowan, “Linux.Darlloz,” 2014. [Online]. Available: https://www.iot-now.com/2014/03/26/19228-symantec-finds-new-variant-linux-darlloz-worm-targets-internet-things/. Accessed Nov. 10, 2020.

145. K. Hayashi, “IoT worm used to mine cryptocurrency,” 2014. [Online]. Available: https://community.broadcom.com/symantecenterprise/communities/community-home/librarydocuments/viewdocument?DocumentKey=00fcdbad-954d-42ff-af50-4d74001bdcbb&CommunityKey=1ecf5f55-9545-44d6-b0f4-4e4a7f5f5e68&tab=librarydocuments. Accessed Nov. 10, 2020.

146. M. Ballano, “Is there an internet-of-things vigilante out there,” 2015. [Online]. Available: https://community.broadcom.com/symantecenterprise/communities/community-home/librarydocuments/viewdocument?DocumentKey=ef23b297-5cc6-4c4a-b2e7-ff41635965fe&CommunityKey=1ecf5f55-9545-44d6-b0f4-4e4a7f5f5e68&tab=librarydocuments. Accessed Nov. 10, 2020.

147. J. Ullrich, “Linksys Worm (TheMoon) captured,” 2014. [Online]. Available: https://isc.sans.edu/forums/diary/Linksys+Worm+TheMoon+Captured/17630. Accessed Nov. 10, 2020.

148. Akamai, “Spike DDoS toolkit,” 2014. [Online]. Available: https://www.akamai.com/fr/fr/multimedia/documents/state-of-the-internet/spike-ddos-toolkit-threat-advisory.pdf. Accessed Nov. 10, 2020.

149. T. Spring, K. Carpenter, and M. Mimoso, “Bashlite family of malware infects 1 million iot devices,” Threat Post, 2016.

150. P. Paganini, “The Linux Remaiten malware is building a Botnet of IoT devices,” 2016. [Online]. Available:

http://securityaffairs.co/wordpress/45820/iot/linux-remaiten-iot-botnet.html. Accessed Nov. 10, 2020.

151. E. IRGC, "Governing cybersecurity risks and benefits of the internet of things: connected medical & health devices and connected vehicles," 2017. Hentet fra https//irgc.org/wp-content/uploads/2018/09/IRGC.-2017.-Cybersecurity-in-the-IoT.-Workshop-report.pdf.

152. M. Abomhara, and G.M. Køien, "Cyber security and the internet of things: vulnerabilities, threats, intruders and attacks," *J. Cyber Secur. Mobil.*, Vol. 4, no. 1, pp. 65–88, 2015.

153. J. Cochran, "The Wirex Botnet," 2018. [Online]. Available: https://blog.cloudflare.com/the-wirex-botnet/. Accessed Nov. 20, 2020.

154. "The reaper IoT botnet has already infected A Mil- lion networks," 2017. [Online]. Available: https://www.wired.com/story/reaper-iot-botnet-infected-million-networks/. Accessed Nov. 10, 2020.

155. Smith, "New Vicious Torii IoT Botnet Discovered," 2018. [Online]. Available: https://www.csoonline.com/article/3310222/new-vicious-torii-iot-botnet-discovered.html. Accessed Nov. 10, 2020.

156. "Alert (TA18-331A) 3ve—Major online Ad Fraud operation," 2018. [Online]. Available: https://us-cert.cisa.gov/ncas/alerts/TA18-331A. Accessed Nov. 10, 2020.

157. Microsoft Azure, "Internet of Things security architecture," 2017. Available: Microsoft Azur. https//docs.microsoft.com/enus/Azur. Accessed Sept. 2, 2018.

158. A. Kliarsky, and K. Leune, "Detecting attacks against the internet of things," SANS Inst. Inf. Secur. Read. Room, 2017.

159. A. Shalaginov, O. Semeniuta, and M. Alazab, "MEML: resource-aware MQTT-based machine learning for network attacks detection on IoT edge devices," in UCC 2019 Companion: Proceedings of the 12th IEEE/ACM International Conference on Utility and Cloud Computing Companion, 2019, pp. 123–128.

160. E. Ciklabakkal, A. Donmez, M. Erdemir, E. Suren, M. K. Yilmaz, and P. Angin, "ARTEMIS: An intrusion detection system for MQTT attacks in internet of things," in Proceedings of the IEEE Symposium on Reliable Distributed Systems, 2019, pp. 369–371.

161. N. F. Syed, Z. Baig, A. Ibrahim, and C. Valli, "Denial of service attack detection through machine learning for the IoT," *J. Inf. Telecommun.*, Vol. 4, no. 4, pp. 482–503, 2020. DOI: 10.1080/24751839.2020.1767484

162. H. Alaiz-Moreton, J. Aveleira-Mata, J. Ondicol-Garcia, A. L. Muñoz-Castañeda, I. García, and C. Benavides, "Multiclass classification procedure for detecting attacks on MQTT-IoT protocol," *Complexity*, Vol. 2019, pp. 1–11, 2019.

163. F. Buccafurri, V. De Angelis, and R. Nardone, "Securing MQTT by blockchain-based otp authentication," *Sensors (Switzerland)*, Vol. 20, no. 7, pp. 2002, 2020.

164. F. Buccafurri, and C. Romolo, "A blockchain-based OTP-authentication scheme for constrainded IoT devices using MQTT," in ACM International Conference Proceeding Series, 2019.

165. A. E. Guerrero-Sanchez, E. A. Rivas-Araiza, J. L. Gonzalez-Cordoba, M. Toledano-Ayala, and A. Takacs, "Blockchain mechanism and symmetric encryption in a wireless sensor network," *Sensors (Switzerland)*, Vol. 20, no. 10, p. 2798, 2020.

166. M. Katende, "Combining MQTT and Blockchain to improve data security," in 3rd International Workshop on Emerging Trends in Software Engineering for Blockchain (WETSEB), 2020.

167. T. A. Ahanger, "Defense scheme to protect IoT from cyber attacks using AI principles," *Int. J. Comput. Commun. Control*, Vol. 13, no. 6, pp. 915–926, 2018.

168. S. Hernández Ramos, M. T. Villalba, and R. Lacuesta, "MQTT security: a novel fuzzing approach," *Wirel. Commun. Mob. Comput.*, Vol. 2018, pp. 1–11, 2018.

169. H. HaddadPajouh, R. Khayami, A. Dehghantanha, K. K. R. Choo, and R. M. Parizi, "AI4SAFE-IoT: an AI-powered secure architecture for edge layer of internet of things," *Neural Comput. Appl.*, Vol. 32, no. 20, pp. 16119–16133, 2020.

170. P. C. Kocher, and T. Dierks, "The TLS Protocol Version 1.0," 1996.

171. S. Katsikeas, "A lightweight and secure MQTT implementation for Wireless Sensor Nodes," Tech. Univ. Crete, 2016.

172. A. Mektoubi, H. L. Hassani, H. Belhadaoui, M. Rifi, and A. Zakari, "New approach for securing communication over MQTT protocol A comparaison between RSA and elliptic curve," in Proceedings of the 2016 3rd International Conference on Systems of Collaboration (SysCo), 2016, vol. 0, 2017.

AUTHORS

Ahmed J. Hintaw was born in Karbala Province, Iraq, in 1986. He received the B.S. degree, Hefei, in 2009 and the M.S. degree from Jamia Hamdard University (JHU), New Delhi, India in 2012, both in computer science. He is currently pursuing the PhD degree with the National Advanced IPv6 Center (NAv6), Universiti Sains Malaysia (USM). His research interests include Internet of Things, Cryptography, and Network Security.

Email: aj.hintaw@nav6.usm.my

Selvakumar Manickam is the senior lecturer at National Advanced IPv6 Centre (NAv6), Universiti Sains Malaysia. He received his Bachelor of Computer Science and Master of Computer Science in 1999 and 2002, respectively. He obtained his PhD from Universiti Sains Malaysia (USM) in 2013. His research interests are Internet security, cloud computing, IoT, Android and open source technology. He is an Executive Council member of Internet Society (ISOC), Malaysian Chapter and also the Head of Internet Security Working Group under Malaysian Research and Education Network (MyREN).

Corresponding author. Email: selva@nav6.usm.my

Mohammed Abomaali head of Computer Techniques Engineering Department at Alsafwa University College, Iraq. He received a bachelor's degree in software engineering from Mansour University College and a master's as well as a PhD degree in computer sciences from Universiti Sains Malaysia in Penang, Malaysia. His research interests include parallel computing, cloud computing and IoT.

Email: mohammadfaiz2003@gmail.com

Shankar Karuppayah is currently a Senior Lecturer and researcher at National Advanced IPv6 Centre (NAv6), Universiti Sains Malaysia. He obtained his B.Sc Computer Science (USM), Malaysia and the M.Sc. Software Systems Engineering (KMUTNB), Thailand. He obtained his PhD in 2016 from Technische Universität Darmstadt in the field of Cyber Security. His main research interests are P2P Botnets, Distributed Systems and Cyber Security in general. To date, he has authored and co-authored many articles in journals, workshops, and conference proceedings. He is also a reviewer in many esteemed network and security journals.

Email: shankar@nav6.usm.my