IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2023.0322000

Securing MQTT Ecosystem: Exploring Vulnerabilities, Mitigations, and Future Trajectories

SHAMS UL ARFEEN LAGHARI[1], WENHAO LI[1], (Graduate Student Member, IEEE), SELVAKUMAR MANICKAM[1], PRIYADARSI NANDA[2], (Senior Member, IEEE), AYMAN KHALLEL AL-ANI[3], and SHANKAR KARUPPAYAH[1], (Member, IEEE)

[1]National Advanced IPv6 Centre (NAv6), Universiti Sains Malaysia, Gelugor, Pulau Pinang 11800, Malaysia
[2]Faculty of Engineering and IT, University of Technology Sydney, NSW 2007, Australia
[3]Department of Cybersecurity Engineering Technology, Al Hikma University College, Baghdad, Iraq

Corresponding author: Shankar Karuppayah (kshankar@usm.my) and Wenhao Li (wenhaoli@ieee.org)

ABSTRACT Amid the exponential rise of Internet of Things (IoT) devices, the Message Queue Telemetry Transport (MQTT) protocol has gained prominence due to its efficiency in facilitating device-cloud interactions. Yet, the surge in IoT device usage and MQTT's popularity has spotlighted potential security risks. Vulnerabilities in this realm can lead to substantial disturbances and financial setbacks. While there is a noticeable increase in IoT-related attacks, comprehensive reviews on MQTT security remain scarce. Existing studies often exhibit shortcomings, such as a broad but superficial discussion of MQTT attacks and countermeasures. Additionally, many essential components and roles in building or implementing MQTT-based applications have not been adequately addressed. This research fills this void by offering a contemporary analysis of MQTT ecosystem security challenges, encompassing prevalent attacks, their repercussions, mitigation strategies, and prospective areas for further research. This study presents a comprehensive taxonomy of security attacks within the MQTT ecosystem, offering a systematic framework to guide researchers, businesses, and end-users in mitigating these risks. As a result, this work serves as a crucial resource for enhancing the security of IoT devices utilizing MQTT, marking a significant stride in safeguarding IoT infrastructure.

INDEX TERMS IoT Security, MQTT Attacks, MQTT Ecosystem, MQTT Security.

I. INTRODUCTION

SINCE Kevin Ashton first coined the term "Internet of Things" in 1999 [1], IoT technology has profoundly influenced everyday life. It has enabled physical objects to interact and share information through digital networks, fundamentally altering our interaction with the digital environment. Accelerated by advancements in 5G/6G and associated technologies, IoT's reach now extends to sectors like media, logistics, transport, healthcare, energy, retail, residential infrastructure, and urban development [2]–[7]. The global count of IoT devices is on an upward trajectory. An estimated 26 billion IoT devices are operational worldwide, with projections indicating a dramatic increase to 75 billion by 2025 [8]. Both individuals and businesses reap significant advantages from diverse IoT implementations. For instance, smart infrastructures optimize energy use and spatial management, while intelligent urban systems refine transport and civic amenities [98]. In the same vein, smart farming introduces the concept of distant agriculture and uses this technology to manage the resources and monitor the plants [101], [102]. Smart homes centralize household device management and automate actions based on set conditions [9]. Moreover, the advent of Industry 4.0 marks a significant shift in manufacturing and business. This new industrial phase, propelled by IoT, integrates advanced technologies like Artificial Intelligence (AI), robotics, and cloud computing, revolutionizing manufacturing processes and enhancing client engagement while minimizing operational halts [10]. In the healthcare sector, smart solutions promise superior health surveillance for patients and tailored medical assessments [11], [12]. The augmented data flow and speed courtesy of IoT empower users to engage more dynamically with their environment, leading to heightened efficiency, comfort, and convenience, all while conserving time and resources.

At their core, IoT devices rely on specific protocols to facilitate Machine-to-Machine (M2M) interactions, which underpin the myriad activities and functionalities intrinsic to IoT. Broadly, IoT protocols can be categorized into two main types: those that function at the application layer and those active at the physical and data link layer. The physical/data link layer protocols primarily oversee device communication and network connectivity. In this category, networks operate over extensive distances, such as 2G/3G/4G/5G, NB-IoT, WiFi, and ZigBee. These are juxtaposed with short-distance communication systems like RFID, NFC, and Bluetooth. Additionally, wired communication methods like RS232 and USB also fall under this umbrella. Alternatively, application layer protocols mainly operate on the foundation of TCP/IP. They play a crucial role in managing data transmissions and bridging the communication between nodes and platforms hosted on the internet. Noteworthy instances of these application layer protocols encompass the MQTT, Advanced Message Queuing Protocol (AMQP), Extensible Messaging and Presence Protocol (XMPP), Constrained Application Protocol (CoAP) and the Hypertext Transfer Protocol (HTTP), where MQTT stands out for its widespread adoption, attributed to its resource and energy efficiency, and user-friendliness [13]–[15], which is even being used in the communication of vehicular networks [16]. This makes it particularly suitable for environments with limited resources, finding applications in domains like IoT, mobile internet, smart devices, connected transportation, and energy sectors. MQTT serves as a dual exceptional access point for device-side communication and a conduit between devices and cloud platforms. Figure I, sourced from the Shodan [17] cyber search engine as of this paper's drafting, reveals that more than 528,992 MQTT-operated servers have been identified, with the highest numbers located in Korea, China, and the United States. This data from Shodan underscores recent growing adoption of MQTT servers. The widespread use of MQTT servers globally is also highlighted by data from FOFA [107], a cyber search engine based in China. In 2023, a query revealed 689,956 servers running the MQTT protocol, as illustrated in Figure I. The specific query used is *protocol="mqtt" && before="2024-01-01" && after="2022-12-31"*, indicating that most of these servers are located in China, Korea, Japan, and the United States of America.

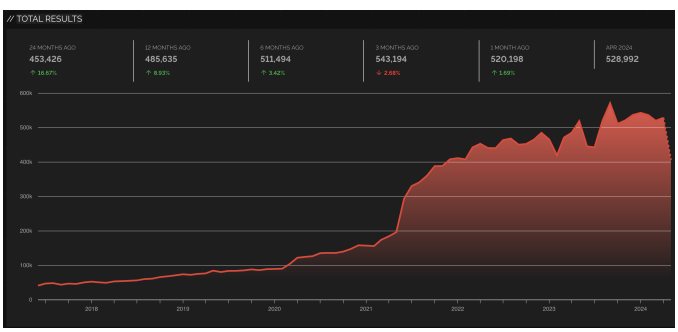

FIGURE 1. Distribution of MQTT Servers Detected in Shodan [17].

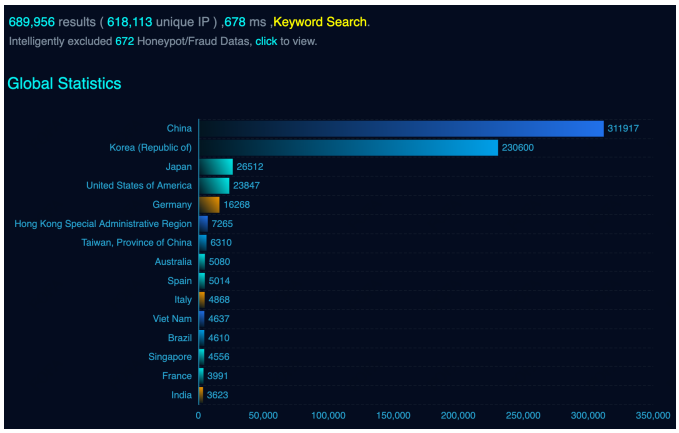

FIGURE 2. MQTT Servers Detected in FOFA in 2023 [107].

The escalating integration of IoT devices and the prevalent adoption of MQTT have not gone unnoticed by potential attackers. Indeed, IoT security, especially in the rapidly evolving landscape of Industry 4.0, is increasingly recognized as a rapidly emerging concern in cybersecurity [18]–[20], particularly in data and privacy [9], [21]. A report [22] highlights that when improperly configured, the open-source messaging protocol, MQTT, can inadvertently expose sensitive equipment such as prison security systems, cardiac monitors, insulin dispensers, nuclear reactors, particle accelerators, and oil conduits. Such exposures render these devices susceptible to unauthorized intrusions and cyberattacks. The stakes are particularly high with MQTT protocol vulnerabilities, as they differ from conventional system flaws. Given their widespread application in daily routines, essential sectors, and critical infrastructures directly interacting with humans or physical devices, any security breach can have severe consequences, including threats to human safety, physical disruptions, and significant financial losses. Compounding the issue, many existing security solutions are unsuitable for IoT devices due to their unique characteristics. This scenario increases the complexity of ensuring MQTT ecosystem's security and associated devices.

To comprehensively assess the current state of MQTT security research, we initially identified 629 articles from Google Scholar, 19 from Web of Science, and 22 from Scopus from 2018 to 2023 using the keyword "MQTT Security". We then included only review/survey papers written in English, excluding duplicates, three papers due to their superficial discussion, and one paper for its striking resemblance to a previously published journal article. Ultimately, only eight review/survey articles were included [23]–[30]. Papers that review, compare, or analyze various IoT protocols were excluded because they only partially emphasize the MQTT protocol, resulting in a surprisingly low number of relevant articles given the rising trend of IoT-related attacks. The final count includes four journal articles and four conference papers, all of which delve into the attack methodologies and defensive measures associated with the MQTT protocol.

Table 1 provides a detailed breakdown, showcasing the objectives, attack methodologies, defensive strategies, identified limitations, and unresolved issues highlighted in the studies above.

Bilal et al. [27] have conducted a brief survey concentrating on the MQTT and CoAP, explaining their architectures, messaging format, and some general security enhancements over MQTT protocol from different layers without identifying the exact threats. Then Harsha et al. [29] further illustrated the threats in the MQTT protocol, including authentication, plaintext transmission, and authorized data access, and pointed out that using encryption and access control can help deal with these issues. Hintaw et al. [23] reviewed security measures implemented within the MQTT protocol for IoT, shedding light on the limitations of current mechanisms. These limitations include their intricate nature, the necessity for supplementary services, and their provision of only fragmentary protection. Notably, this review omits specific attack vectors or scenarios, concentrating primarily on encryption techniques for MQTT within IoT. Oza & Kamdar [24] embarked on an analytical comparison of several security strategies, yet their study is somewhat superficial, not delving deeply into the specific methods of attacks and strategies of countermeasures related to MQTT. A study proposed by Chen et al. [25] examined pertinent literature, identifying attacks on MQTT such as MiTM attacks, DoS attacks, and replay attacks. Their conclusions highlight defense strategies like encryption, blockchain utilization, and ML applications. Atigan et al. [28] examined attacks such as DoS and MiTM and categorized the current solutions on MQTT security into protocol enhancement and attack detection. Hintaw et al. [26] provided a most comprehensive analysis of the security aspects of the MQTT protocol, highlighting various potential threats such as DoS, Identity Impersonation, Information Leakage, Privilege Escalation, and Data Alteration. Their research also outlines current defensive measures, encompassing firewalls, policies, access controls, encryption, ML, blockchain, and AI. Despite the work proposed by Harsha et al., [29] where they analyzed the security of MQTT broker software, newer research underscores the need to focus on the broader MQTT application ecosystem, where additional vulnerabilities might be leveraged by malicious actors [31]–[33]. It is also pivotal to understand the tools at the disposal of these attackers targeting MQTT to bolster the protective mechanisms of IoT devices.

Consequently, this research aims to offer a thorough and contemporary review of MQTT ecosystem security challenges, encompassing its attack vectors, cutting-edge attack techniques/vulnerabilities, tools employed to exploit MQTT, defensive measures, and potential avenues for future research. The primary contributions of this review include:

- A thorough exploration of recent attacks on the MQTT ecosystem, covering their impacts, countermeasures, and future research directions, supported by an extensive review of relevant literature.
- A detailed exploration of the MQTT security ecosystem, including its critical components, offering insights into its operational dynamics and vulnerabilities.
- The development of a comprehensive attack taxonomy tailored to the MQTT landscape provides a systematic framework to assist researchers, institutions, and individuals in understanding and mitigating these threats effectively.
- An in-depth identification and investigation of various threats within the MQTT ecosystem, featuring real-world examples and a review of existing attack tools to highlight practical security challenges and their implications.
- A critical analysis of current countermeasures against MQTT attacks, evaluating their strengths and weaknesses, coupled with a forward-looking discussion on potential areas for future research and development in enhancing MQTT security.

The subsequent structure of this paper is delineated as follows: Section II introduces the foundational aspects of the MQTT protocol; section III presents the prevailing attack vectors targeting the MQTT ecosystem, derived from academic sources and practical simulations; Section IV elucidates the avant-garde defensive strategies against MQTT security threats, and Section V offers concluding remarks and outlines potential research directions.

II. BACKGROUND

This section explores the fundamental concepts necessary to understand the core elements of the MQTT communication ecosystem.

A. MQTT PROTOCOL

MQTT, a streamlined messaging protocol operating on a publish/subscribe model over TCP/IP, was conceived by Andy Stanford-Clark of IBM and Arlen Nipper of Arcom in 1999 [34]. It later gained standardization in 2013 under the auspices of the Organization for the Advancement of Structured Information Standards (OASIS/ISO). MQTT's design facilitates the delivery of dependable real-time messaging services to remote nodes while using minimal code and bandwidth. Within the MQTT framework, there are three primary roles: the publisher, the broker, and the subscriber. By adopting this publish-subscribe messaging paradigm, which diverges from the traditional client-server model, MQTT distinctly differentiates between the message initiator (publisher) and the message recipient (subscriber). This distinction negates the need for direct interaction between the two. This model allows for multiple subscribers to disseminate and retrieve messages. Additionally, it enables a singularly published message to be accessed by several subscribers, a process depicted in Figure 3. The dialogue between the nodes encompasses the exchange of a topic and its associated payload. Here, the topic serves as a designated path or channel, which both parties must recognize to transmit or obtain the payload, essentially the message's content. To cater to potential IoT device malfunctions, MQTT incorporates a "will message" mechanism. Through

TABLE 1. Summary of Existing Literature on Security Aspects of MQTT

Review Works	Publication Year	Primary Focus	Identified Threats	Proposed Safeguards	Open Issues
Bilal et al. [27]	2018	Brief exploration of MQTT and CoAP on their architecture, format, and security	Not specified	General Security Protection over the network, transport and application layers	Not specified
Harsha et al. [29]	2018	Analysis of vulnerabilities in MQTT security using Shodan Application Programming Interface (API) and implementation of its countermeasures via authentication and Access Control Lists (ACLs)	Data Security, Man-in-the-middle (MiTM)	Encryption, Access Control	Not specified
Hintaw et al. [23]	2019	Exploration of techniques to bolster MQTT security within IoT	Not specified	Methods include Encryption and Access Control mechanisms	Emphasis on integrated security solutions for IoT devices with limited resources
Kotak et al. [30]	2019	Discussion on the vulnerabilities of MQTT broker software	Denial of Service (DoS) attacks, Data security	Not specified	Evaluate other forms of attacks on MQTT software
Oza & Kamdar [24]	2020	In-depth study on the diverse security methodologies for MQTT	Not specified	Not specified	Focus on evaluating Quality of Service (QoS) attributes tailored to application necessities
Chen et al. [25]	2020	Overview on pressing threats and subsequent offensive and defensive tactics in MQTT	Threats like Replay, MiTM, and DoS attacks	Encryption, Blockchain, Machine Learning (ML), AI and Blockchain	Suggestion to execute real-world threat assessments across different attack environments
Hintaw et al. [26]	2021	Thorough insight into MQTT's vulnerabilities and countermeasures	Threats include DoS, Identity Spoofing, Information Exposure, Privilege Elevation, and Data Tampering	Defense strategies encompass Firewall/Policy/ACL, Encryption, ML, Blockchain, and AI	Recommendation for a lightweight yet potent security upgrade to MQTT.
Atilgan et al. [28]	2021	An overview on MQTT protocol security issues and solutions.	DoS, MiTM, Brute Force	Protocol improvement, Attack Detection	A method to efficiently select MQTT and network data features and develop real-time detection solutions.

this feature, a client, upon connecting to a broker, can set a will message. Should any unexpected disconnections arise, the broker disseminates this "will message" to all subscribers aligned with the client's topic.

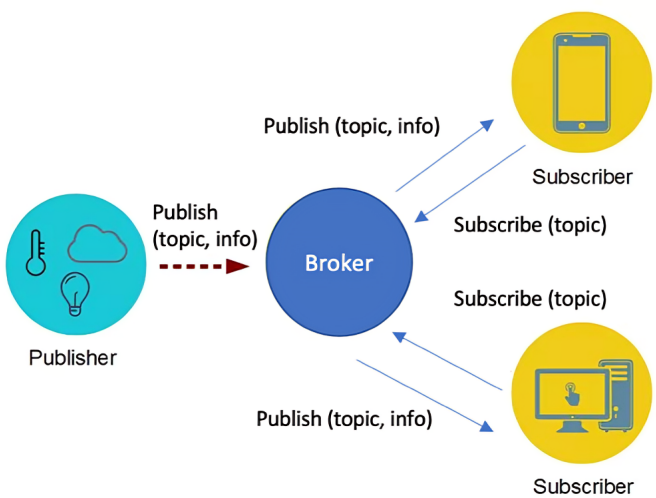

FIGURE 3. MQTT's Pub-Sub Communication Workflow [35].

B. CONTROL PACKET AND QOS PROFILES

Figure 4 illustrates the architecture of MQTT's control packet. A Fixed header is a consistent feature across all MQTT control packets. However, variable headers and the payload, which convey the message's content, might not always be present in every control packet. Typically, for communication to commence, a client must establish a TCP session with the broker. During this process, the client transmits packets labeled with PINGREQ and PINGRESP as their message types, which are referred to as Ping Request and Ping Response messages, respectively.

A salient feature of MQTT is its emphasis on QoS levels. Despite the foundational connection being TCP-based, vulnerabilities can arise, especially due to radio interference in wireless settings. To address this, MQTT delineates three QoS levels [36].

- At QoS level 0, a message is dispatched at most once, with no provisions for retries or acknowledgments from the recipient's end.
- At QoS level 1, a message is dispatched at least once and is accompanied by a confirmation of receipt from the recipient.
- QoS level 2, on the other hand, ensures that the message not only reaches the recipient but also receives an additional acknowledgment. This level incurs the most

overhead compared to the other two QoS tiers, given its enhanced reliability measures.

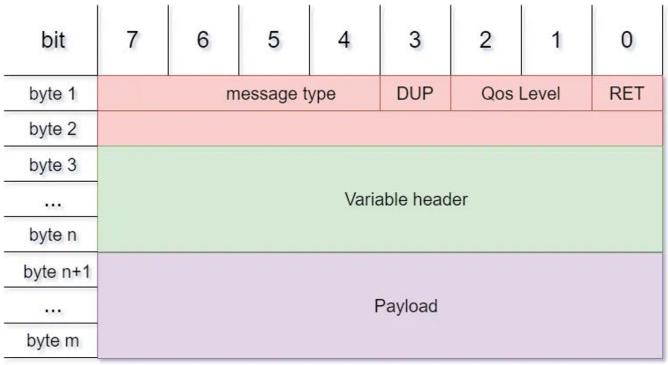

FIGURE 4. Layout of MQTT's Control Packet.

C. MQTT ECOSYSTEM AND COMPONENTS

In practical situations, publishers and subscribers exchange their information through a broker, which is usually hosted on a server. These nodes (i.e., publishers and subscribers) can be part of devices, online applications, or mobile ecosystems. Considering the diverse range of devices, platforms, and functionalities, identifying the key elements of the MQTT security framework becomes essential. This is because security challenges are intrinsically linked and cannot be tackled in a piecemeal fashion. Figure 5 succinctly showcases the primary roles and elements within the MQTT security framework, encompassing developers, users, and potential malicious actors.

Developers play a pivotal role, chiefly tasked with crafting applications that leverage MQTT message broker software. Examples of such software include Eclipse Mosquitto, EMQ, and VerneMQ, all of which operationalize the MQTT protocol. Some of these software offerings even furnish a web management system tailored for developers. Additionally, developers might also design auxiliary tools to facilitate user-device interactions. These broker systems can either be anchored to local servers or be cloud-centric, with the latter witnessing a surge in adoption, especially given the rising inclination towards IoT cloud solutions proffered by tech giants.

In a smart home environment, an array of interconnected nodes communicate with the central broker through a designated gateway. This setup streamlines information interchange between the devices and end-users, who typically control the whole system via mobile or portable devices. In this context, the devices, their manufacturers, and the gateway emerge as pivotal components warranting attention. Concurrently, potential malicious actors remain on the lookout, scouting for exploitable weak points within the ecosystem to orchestrate attacks. Given this landscape, a holistic approach becomes indispensable when bolstering MQTT's security.

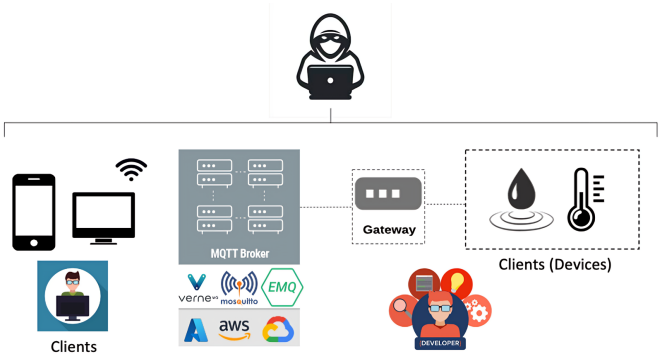

FIGURE 5. MQTT Security Ecosystem.

III. ATTACKS

This section delves into the burgeoning security threats targeting the MQTT ecosystem. Earlier research introduced a taxonomy of MQTT attacks, categorizing them based on potential threats stemming from TCP, MQTT-specific attacks, data-centric attacks, and Transport Layer Security (TLS) vulnerabilities within MQTT [26]. However, our study endeavors to craft a more encompassing taxonomy for MQTT attacks. This is achieved by taking into account all the components and stakeholders that could potentially influence the security dynamics of the ecosystem. As depicted in Figure 6, this approach offers a more holistic perspective on MQTT ecosystem security, ensuring a comprehensive understanding of the threats and vulnerabilities.

A. MQTT-BASED ATTACKS

1) Unauthorized Access

One of the predominant challenges in MQTT security is unauthorized access. This vulnerability has been highlighted in prior research [37], [38]. While MQTT does offer a variety of authentication methods, such as passwords and JSON Web Token (JWT), these are not activated by default. This oversight means that if adversaries can reach MQTT servers, they can potentially gain unauthorized access. To gain a clearer understanding of how this vulnerability manifests in real-world IoT devices, a dataset comprising 5000 server records running MQTT was sourced from Shodan [17]. This dataset was randomly chosen, and the top five countries of these records are Korea (3526 records), China (539 records), the United States (199 records), Germany (97 records), and the Netherlands (56 records), Other countries (583 records). A Python script was crafted using the paho-mqtt, which can be adopted in a wide range of programming languages. This open-source library facilitates the implementation of MQTT on the client side [39]. It also defines connection return codes that denote the status of an MQTT connection. Table 2 lists six primary return codes. The objective of the script was to initiate connections to the servers listed in the dataset and record the returned connection codes. A return value of 0 indicates a successful establishment of a connection between

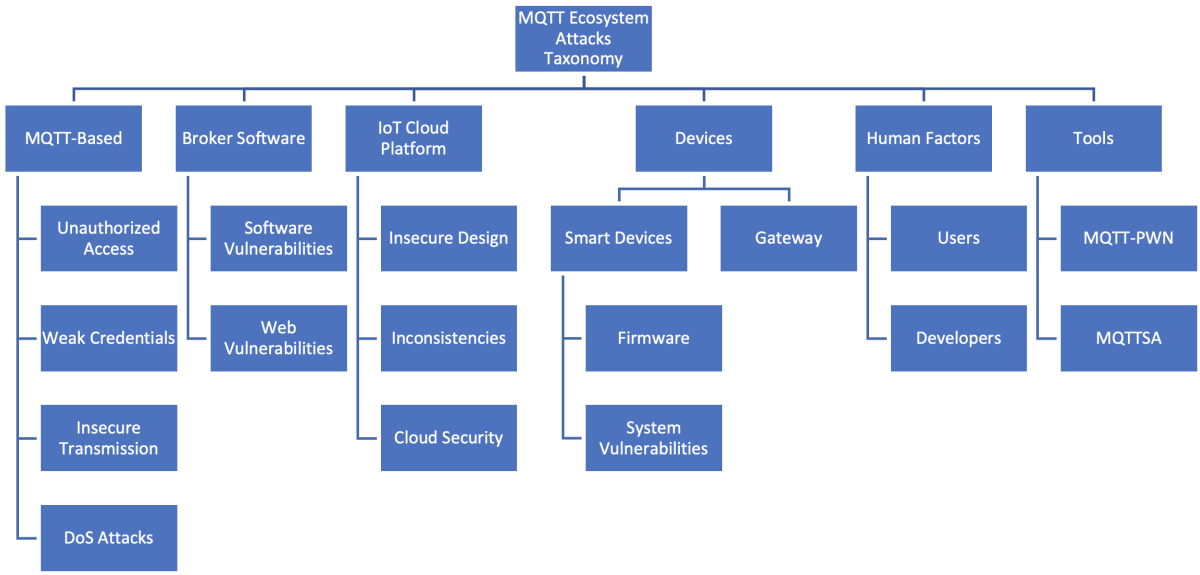

FIGURE 6. MQTT Ecosystem Attacks Taxonomy.

TABLE 2. Response Codes for Connection in Paho-MQTT

Code	Representation
0	Connection Successful
1	Incorrect Protocol Version
2	Invalid Client Identifier
3	Server Unavailable
4	Bad username or Password
5	Not authorized

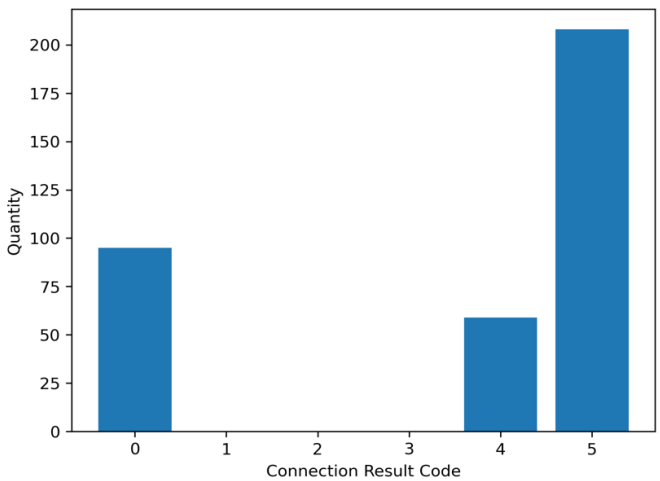

FIGURE 7. MQTT Servers of Unauthorized Access.

the client and the broker. As illustrated in Figure 7, out of the 5000 tested servers, 95 were found to be susceptible to unauthorized access. In contrast, 59 servers required credentials with a return code of 4, and 208 servers necessitated authorization with a return code of 5. These findings underscore that unauthorized access remains a significant threat to MQTT.

Unauthorized access grants malicious actors the capability to connect to servers, allowing them to either publish tailored messages or subscribe to specific topics. This can lead to potential data breaches or induce hazardous operations on devices. Moreover, MQTT's wildcard characters, namely # and + , provide adversaries with a unique advantage. These characters allow them to subscribe to the unknown topics [31], [40]. In particular, the # symbol, positioned at a string's end, corresponds to various levels of a topic, whereas the + symbol aligns with just one level [41]. In MQTT, a topic is a string of alphanumeric tokens separated by a "topic level separator." Attackers can subscribe to all topics using the # wildcard or to system-centric topics with $SYS/#, which may compromise the application or system information. For instance, $SYS topics contain various information about system and network metrics, including the version of broker software, where an attacker can identify the certain version of MQTT broker software that clients are using by subscribing to the topic: $SYS/broker/version and exploit the broker with related vulnerabilities. Different implementations in MQTT broker software may compromise different information. In EMQX, the names of nodes will be included in the system topics (e.g., $SYS/brokers/emqx@172.20.0.2/version) where the internal IP address of a certain broker is compromised.

To gauge the extent of information that could be inadvertently exposed due to unauthorized access, we conducted an experiment. This involved connecting to MQTT servers deemed vulnerable and subsequently subscribing to their topics. A thorough examination of the received payloads revealed several categories of sensitive information that could be compromised. One prominent category was the disclosure of internal network or device details, as shown in Figure 8.

Here, IoT devices would inadvertently share their configuration data with other devices. This shared data often included IP and MAC addresses, SSIDs, and specifics about device names and types. Such information, in the hands of malicious actors, can be weaponized to orchestrate more sophisticated attacks.

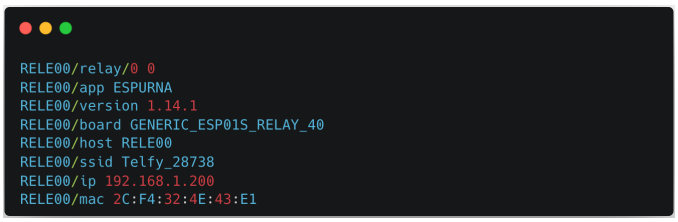

FIGURE 8. Internal IP address Disclosure.

The second category of compromised information pertains to personal details, as illustrated in Figure 9. Here, IoT devices transmit data, including personal identifiers, in plaintext. This can encompass details like names, email addresses, and gender information. When such sensitive data falls into the hands of malevolent entities, the consequences can be severe. Armed with this personal information, attackers can craft highly targeted and deceptive attacks, such as spear-phishing campaigns, which are tailored to specific individuals using the acquired data, thereby increasing the likelihood of the attack's success.

FIGURE 9. Private Information Disclosure.

The third category pertains to the details of configuration information, as depicted in Figure 10. Here, potential data from systems like Home Assistant, which is an open-source hub for home automation, is showcased. Possession of such configuration data grants attackers insights into the operational status of physical devices. Armed with this knowledge, malicious actors can craft specific commands to manipulate these devices based on the disclosed information, potentially compromising the security and functionality of the entire home automation system and threatening privacy and security.

FIGURE 10. Configuration Information Disclosure.

It's important to emphasize that the scope of the experiment was restricted to a select number of devices, implying that the dataset might not be exhaustive. A more extensive dataset might reveal sensitive details, such as daily living patterns or GPS coordinates. If such information were to fall into the hands of malicious actors, the ramifications could be gravely significant, posing heightened risks to individuals and their privacy.

2) Weak Credentials

In the burgeoning age of IoT, the use of weak passwords stands out as a glaring security vulnerability [42]. While the incorporation of usernames and passwords can bolster the security of IoT systems, this measure alone often falls short. A significant number of IoT devices neither mandate robust password protocols nor impose restrictions on login attempts. This laxity allows users to opt for simplistic username-password combinations, which are susceptible to brute-force attacks. Moreover, adversaries can exploit dictionaries containing frequently used username and password pairs, systematically attempting to connect to MQTT brokers. By iterating through these combinations, attackers aim to identify a valid set of credentials. Once they achieve a successful connection with the brokers, these malicious actors gain the ability to both subscribe to and publish messages. This access allows them to extract sensitive information or manipulate devices, leading to consequences akin to those observed in cases of unauthorized access.

3) Insecure Transmission

A recurring security concern highlighted in numerous studies regarding MQTT is insecure transmission. By default, data is relayed in plaintext, as depicted in Figure 11 [43]–[48], given that MQTT communication runs over TCP. The transmission of unencrypted data across networks opens the door to a variety of potential attacks by malicious entities.

One primary vulnerability is data sniffing. Attackers can eavesdrop on the data being exchanged between users and brokers. In doing so, they can capture sensitive details, including usernames, passwords, and the content of the payload. This not only compromises data confidentiality but also infringes upon user privacy [37].

Furthermore, the MiTM attack is another significant threat. In this type of attack, malicious actors intercept and potentially alter data being transmitted across the network, therefore undermining the integrity and authenticity of the communication. For such an attack, malicious actors can capture data exchanged between the nodes and manipulate message contents to serve their objectives. In practical scenarios, attackers might exploit public or counterfeit WiFi networks to share a network with their targets, making it easier to intercept communications. Alarmingly, real-world implications of MiTM attacks on MQTT can be catastrophic. One study showcased the potential fallout of such attacks by maliciously altering temperature data through a MiTM attack, suggesting possible subsequent malfunctions or shutdowns in critical systems like aircraft and data centers [49]. Beyond MiTM attacks, insecure transmission also paves the way for replay attacks, where

attackers capture and retransmit data to deceive the receiver or gain unauthorized access.

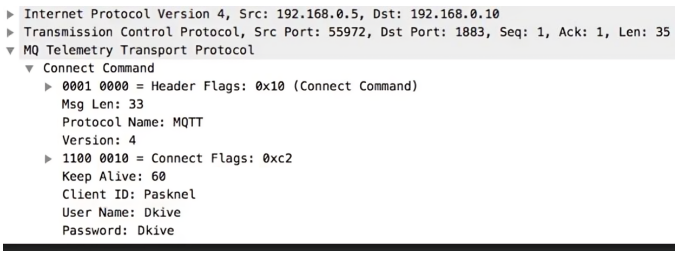

FIGURE 11. Unencrypted Data transmission in MQTT.

4) DoS Attacks

DoS attacks have consistently been a significant security concern across various domains. Broadly, a DoS attack aims to monopolize or exhaust resources—be it time, bandwidth, or storage space—thereby hindering the regular functionality of a system or service for legitimate users.

Within the context of MQTT, researchers have identified three primary categories of DoS attacks [26]:

- TCP-based DoS Attacks: Given that the MQTT protocol operates atop TCP, it is inherently susceptible to certain TCP-centric DoS attacks. A prime example is the TCP SYN flooding attack [41], [50]. In this method, attackers inundate the target with a barrage of SYN packets, leading to the creation of numerous half-opened TCP sessions. This surge in sessions drains the system's resources. Beyond TCP SYN flooding, other TCP-based DoS attacks have been documented [51], [52] where two notable methods, SlowITe and SlowTT, have been designed specifically to exploit vulnerabilities in the MQTT protocol. SlowITe operates by monopolizing all network connections for a predetermined duration, leveraging the KeepAlive parameter. In contrast, SlowTT's strategy is to maintain connections indefinitely. It manipulates specific parameters and network configurations within MQTT to sidestep the need to re-establish connections or set conspicuously high KeepAlive values.
- Payload-based DoS Attacks: In this type of attack, adversaries dispatch payloads exceeding 256 MB, the maximum permissible payload size in MQTT. This oversized payload strains the system's resources. Additionally, attackers can inundate the broker with a multitude of CONNECT packets, further depleting resources and barring genuine users from accessing the broker [50].
- QoS-based Attacks: QoS levels in MQTT can also be exploited for DoS attacks. Specifically, QoS level 2, which ensures the most reliable message delivery, is more resource-intensive compared to QoS levels 0 and 1. Malicious actors can exploit this by overwhelming the system with high QoS-level requests, thereby draining resources more rapidly [50].
- Publish Message-based Attacks: The process of handling MQTT Publish messages requires a brief processing delay. However, when the payloads of these messages are encrypted, the processing time increases, potentially leading to congestion and may result in DoS attacks [105].
- Will Message-based Attacks: When a client closes its connection abruptly from the broker, a *Will message* as mentioned above will be sent to all the subscribers. However, attackers can leverage this feature by setting a well-crafted payload with increased length to initiate the DoS attacks [108].

In essence, the multifaceted nature of DoS attacks, combined with the inherent vulnerabilities of MQTT, underscores the need for robust security measures to safeguard systems and services.

B. BROKER SOFTWARE

Message broker software is pivotal in the development of MQTT-based IoT applications, serving as the backbone for implementing the MQTT protocol. Developers leverage these tools to effortlessly construct brokers on servers. However, the security of the broker is intrinsically tied to the security of these tools. Hence, it is imperative to be vigilant about potential vulnerabilities and threats that might emerge from their usage.

To shed light on the current threat landscape targeting broker software, we embarked on an analysis of the most prevalent message broker software in the market, including Mosquitto, ActiveMQ, RabbitMQ, VerneMQ, and HiveMQ. A meticulous examination of their Common Vulnerabilities and Exposures (CVE) vulnerabilities spanning the last decade was undertaken.

The data sourced from CVE [109], visualized in Figure 12, unveils a concerning trend: there has been a consistent uptick in reported vulnerabilities from 2010 to 2021. This escalating trajectory underscores the growing challenges and risks associated with ensuring the security of these tools. In addition, Apache ActiveMQ appears to be the most consistently vulnerable software among the ones listed, showing vulnerabilities in almost every year from 2010 to 2023.

Delving deeper, when analyzing the nature of these vulnerabilities among the top five broker software, two types stood out as particularly recurrent over the past decade: Cross-Site Scripting (XSS) and DoS. This observation, depicted in Figure 13, emphasizes the need for heightened awareness and mitigation strategies against these specific threats.

In conclusion, while message broker software facilitates the development and deployment of MQTT-based applications, it's crucial to remain cognizant of their associated vulnerabilities. Regular updates, patches, and proactive security measures are essential to safeguard systems and data.

1) System Vulnerabilities

System vulnerabilities typically arise from inadequate validation or parsing of messages, which is also the root cause

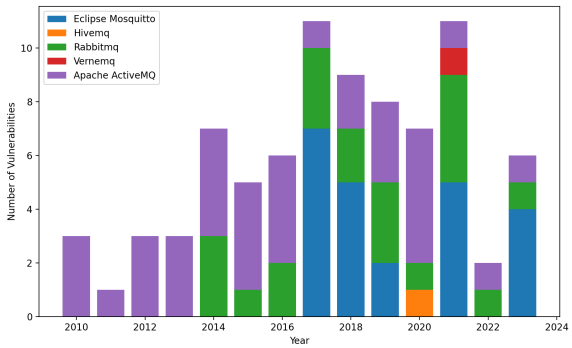

FIGURE 12. **Vulnerabilities Reported for Various Software Products.**

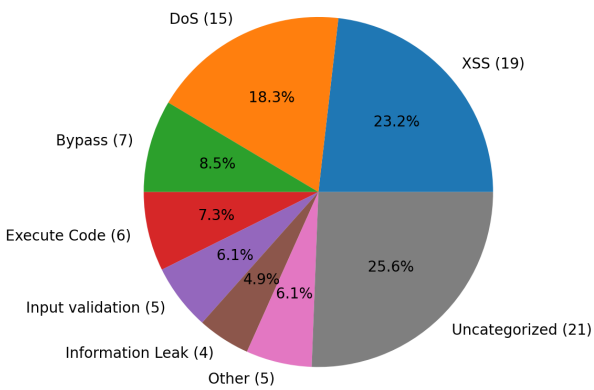

FIGURE 13. **Categories of Vulnerabilities in Brokers**

for many vulnerabilities [53]. These vulnerabilities can have severe repercussions, often culminating in unauthorized command or code execution on the server or even causing the broker software to crash.

A case in point is the vulnerability identified in the Eclipse Mosquitto broker up to version 1.4.15. This specific vulnerability, labeled CVE-2017-7653 [54], highlighted a significant oversight in the broker's handling of strings. The broker did not have mechanisms in place to reject non-valid UTF-8 strings. Consequently, a malicious client could exploit this oversight by sending a topic string embedded with invalid UTF-8 characters. This would, in turn, cause other clients that are designed to reject such strings to sever their connection from the broker.

The ramifications of such an attack are multifaceted. At the surface level, it disrupts the normal functioning of the affected clients. However, the broader implication is the potential for a cascading effect, leading to DoS for all impacted clients. Such vulnerabilities underscore the importance of rigorous validation mechanisms and continuous security assessments to pre-emptively identify and rectify potential threats in broker software.

2) Web Vulnerabilities

Web vulnerabilities are a pressing concern, especially as many broker software solutions now offer web management consoles. These interfaces, if not designed with rigorous security measures, can become a hotbed for potential exploits, especially when users' input is not adequately validated and sanitized.

A classic example of such a vulnerability can be seen with HiveMQ. While HiveMQ offers a management console for user convenience, certain versions of its broker control center have been found lacking in stringent input data validation. This oversight can be exploited by malicious actors. By crafting a specific "ClientID" parameter within an MQTT packet, attack actors can launch Cross-Site Scripting (XSS) attacks, as evidenced by the vulnerability CVE-2020-13821 [54]. Successful exploitation of this vulnerability allows attackers to steal administrator cookies. With these stolen cookies, malicious actors can gain unauthorized access to the management console, effectively granting them control over the broker.

However, the threat landscape does not end with XSS attacks. Other potential web-based attacks, such as Structured query language (SQL) injection and Cross-Site Request Forgery (CSRF), can also be executed if input validation remains lax [56]. SQL injection can allow attackers to manipulate databases, potentially leading to data breaches or system compromises. CSRF, on the other hand, tricks unsuspecting users into performing actions they did not intend to, potentially leading to account takeovers or data loss.

In essence, while web management consoles offer convenience and ease of use, they also introduce a new vector of potential attacks. Developers and administrators must ensure that these interfaces are fortified with robust security measures, including stringent input validation and regular security assessments.

C. IOT CLOUD PLATFORMS

The proliferation of smart devices across various sectors, combined with the advantages of cloud computing, has led to the rise of IoT cloud platforms. These platforms, such as AWS IoT, Samsung SmartThings, and Apple HomeKit, offer both software and cloud-based services for manufacturers and consoles for users. However, with the convenience and scalability they bring, they also introduce a myriad of security challenges in the context of the MQTT ecosystem.

1) Insecure Design

The inherent security limitations of the MQTT protocol, which primarily offers basic authentication mechanisms, have prompted many IoT cloud platforms to devise their security measures. However, these custom mechanisms are not immune to vulnerabilities. Research indicates that these security measures can be exploited, allowing attackers to remotely

control IoT devices, breach user privacy, and launch DoS attacks [33]. A specific study on Samsung SmartThings unveiled multiple design flaws that could be exploited [57].

2) Inconsistencies

A comprehensive analysis of IoT policies on contemporary cloud platforms revealed inconsistencies between the IoT policy and the MQTT protocol, especially concerning wildcards. While MQTT employs ' + ' and '#' as wildcards, cloud-based IoT policies typically use '*' [31]. This discrepancy can be leveraged by attackers to bypass certain policy restrictions and access MQTT topics. To be more specific, if there is an IoT policy that allows access to $\mathrm{a/b/*}$ but denies $\mathrm{a/b/x/y}$, however, attackers may use $\mathrm{a/b/x/+}$ or $\mathrm{a/b/\#}$ to bypass the policy at the IoT policy level [31]. Such inconsistency is also identified between the delegator and delegatee, where different vendors and users may not be clear about each other's policies, leading to going against the original intention of the delegator or delegatee and bringing security risks for the whole delegation chain [58].

3) Cloud Security

As IoT cloud platforms are inherently built upon cloud technologies, the security of the underlying cloud infrastructure becomes paramount [59], [60]. There are well-documented security risks associated with virtualization technologies [18], [42], such as the potential for virtual machine or container escape [61], [62]. The centralized nature of services provided by IoT cloud platforms amplifies concerns about data and privacy security. For example, FlexBooker, a US digital scheduling platform, compromised 3.7 million records of sensitive user data due to a breached AWS server in 2022 [63]. In addition, Civicom, a company that offers online conference service, disclosed eight terabytes of sensitive files due to their misconfiguration on Amazon Simple Storage Service [64]. Given that these platforms centralize data storage and processing, they become attractive targets for attackers. Ensuring the security of data in these environments is even more critical than in traditional setups.

D. DEVICES

1) Smart Devices

Smart devices play a pivotal role in the MQTT ecosystem's security landscape. A notable attack, termed the "Trampoline-over-the-air" attack, was highlighted in a prior study, emphasizing vulnerabilities from the vantage point of the devices rather than the brokers [32]. In this study, researchers developed a fuzzer to probe potential vulnerabilities. Their findings were alarming: attackers could instigate attacks by dispatching malicious messages to brokers, even without gaining control over the broker server, provided the client devices exhibited flaws in message handling and parsing. The efficacy of this framework was underscored by its ability to uncover 34 previously unknown vulnerabilities (0day) in devices, of which 22 were command injection vulnerabilities.

The implications of these findings are profound. System vulnerabilities in smart devices can serve as gateways for attackers, granting them access to internal networks. Once inside, they can exploit additional internal devices, amplifying the potential damage. Furthermore, firmware vulnerabilities compound the security challenge. Attackers, with the right tools, can extract firmware from devices and reverse-engineer it. This process can potentially reveal hard-coded connection credentials embedded within the firmware, providing attackers with the keys to launch even more sophisticated attacks.

In essence, while smart devices offer convenience and enhanced functionality, they also introduce potential weak points in security into the MQTT ecosystem. Ensuring their robustness and resilience against attacks is paramount for the overall security of the system.

2) Gateway

In the context of a smart home, a multitude of devices, ranging from smart thermostats to security cameras, are interconnected through a central router. This router acts as the primary gateway for these devices to communicate with an external broker, facilitating the publishing or subscribing of topics with other clients. Given the router's pivotal role as its central place in the network, its security becomes paramount [65]. If compromised, it can serve as a launchpad for a myriad of attacks [66], jeopardizing the entire smart home ecosystem.

- Router Attacks: If a router is susceptible to brute force attacks, malicious actors can gain access to the home network. Once inside, they can monitor, intercept, and even manipulate the unencrypted communication between smart devices and the broker. Binary vulnerabilities in the router's firmware or software can be another entry point [66]. If exploited, these vulnerabilities can grant attackers elevated privileges, allowing them to run malicious code directly on the router.
- Fake Router Attacks: An insidious tactic employed by attackers involves setting up a rogue router with an identical SSID to the legitimate home router. By strategically positioning this fake router nearby and simultaneously launching a DoS attack on the original router, they can force smart devices to connect to their rogue device. Once devices are connected to this malicious router, attackers can control the data flow, potentially launching MiTM attacks or other malicious activities, which are known as Evil Twin attacks [67], [68].

The above scenarios underscore the importance of robust router security in a smart home setting. As the central hub of communication, the router's integrity is crucial. Ensuring it is fortified against potential attacks, regularly updating its firmware, and employing strong, unique passwords are essential steps in safeguarding the smart home environment.

E. HUMAN FACTORS

The role of human behavior in cybersecurity is well-documented and cannot be understated [69], [70]. Particularly

in the realm of the IoT and its industrial IoT (IIoT), the human factor emerges as a significant vulnerability [71], [72]. MQTT, a protocol integral to many IoT and IIoT systems, is not immune to these human-induced vulnerabilities.

- Developer and Programmer Oversights: MQTT's security can be compromised by lapses in developer awareness. As depicted in Figure 14, connection details might be inadvertently exposed in JavaScript code. Such exposure can provide malicious actors with the necessary information to connect to brokers, leading to potential breaches. Another common oversight is the hard coding of sensitive credentials within mobile application source files. This practice can be especially perilous if the code is subsequently uploaded to public repositories like GitHub, making it accessible to anyone. Misconfigurations during development can also introduce vulnerabilities. For instance, leaving default settings unchanged or not enabling security features can make systems easy targets for attackers [73].
- End-User Behaviors: Users, too, can inadvertently compromise MQTT security. Connecting to untrusted or public WiFi networks and conducting sensitive operations can expose them to MiTM attacks or data eavesdropping. Lack of awareness about the importance of strong, unique passwords or the risks of using outdated software can also make users susceptible to breaches.

In essence, while technological measures are crucial for securing MQTT and associated systems, addressing the human factor is equally vital. Continuous education, training, and awareness campaigns targeting both developers and end-users can go a long way in bolstering the overall security of MQTT-based systems.

```
// Create a Client Instance
client = new Paho.MQTT.Client("broker.server", port, "clientId");

// Set Callback Handlers
client.onConnectionLost = onConnectionLost;
client.onMessageArrived = onMessageArrived;

// Connect
client.Connect({
    onSuccess: onConnect,
    userName: "userName",
    password: "password",
    useSSL: true
});
```

FIGURE 14. Information Leakage in JavaScript Code.

F. MQTT SECURITY ASSESSMENT TOOLS

The landscape of MQTT security has seen the rise of offensive tools that aid attackers in exploiting vulnerabilities in MQTT systems. These tools, while designed for penetration testing and security assessment, can be misused for malicious purposes. To understand the range and functionality of these tools, Table 3 provides an overview of existing security tools available for MQTT. MQTT-PWN is a prominent tool in this domain, offering a comprehensive suite for IoT broker penetration testing [74], as illustrated in Figure 15. The tool comes equipped with a range of modules to facilitate various attacks:

- Credential Brute-Forcer: Attempts to gain unauthorized access by trying multiple username-password combinations.
- Topic Enumerator: Discovers available topics on the broker.
- Useful Information Grabber: Extracts valuable data from the broker.
- GPS Tracker: Pinpoints the location of devices.
- Sonoff Exploiter: Targets vulnerabilities specific to Sonoff devices.
- Extensibility: Allows for the addition of custom modules.
- Shodan API Integration: Integrates with the Shodan search engine to find vulnerable MQTT servers.

The versatility of MQTT-PWN means that attackers, even those with limited knowledge of MQTT, can exploit systems on a large scale.

FIGURE 15. MQTT-PWN.

Another tool that has gained attention is MQTTSA, which, similar to MQTT-PWN, also supports insider attacks [75]. While MQTTSA is open-sourced on GitHub, making it accessible for security researchers and developers, this very accessibility poses a risk. Malicious actors can misuse the tool, turning its capabilities against unsuspecting MQTT systems.

IV. COUNTERMEASURES

In this section, we delve into contemporary solutions designed to combat the vulnerabilities previously highlighted in the context of MQTT security. We will touch upon the efficacy of these countermeasures, underscoring both their advantages and potential limitations. In addition, it is important to understand that MQTT is designed to be a lightweight messaging protocol and enables communications between IoT devices under constraint resources, which results in the lack

TABLE 3. Available Tools in MQTT Security

Name	Description	Purpose	Positive Impacts	Negative Impacts	Common Attack Modules	Unique Attack Modules
MQTT-PWN	A suite for IoT broker penetration testing with various modules for attacks and system testing.	Penetration testing and security assessment of MQTT systems.	Enables security professionals to identify and address vulnerabilities in MQTT systems.	Accessibility makes it prone to misuse by malicious actors targeting MQTT systems.	Brute-Force Attack, Information Disclosure	GPS Tracker Extensibility Shodan Integration
MQTTSA	An open-source tool supporting insider attacks.	Testing and researching MQTT system security.	Facilitates understanding and improvement of MQTT system security by researchers and developers.			Sniffing attack Malformed Data Denial of Service

of built-in security mechanisms of this protocol. Even though users can implement Secure Sockets Layer (SSL)/TLS to secure communications, which offers a high level of security, it is computationally expensive and can be challenging to implement on devices with limited resources. These constraints include limited processing power, memory, and battery life, which make it difficult to sustain the overhead introduced by SSL/TLS. Consequently, adopting these protocols without considering the specific limitations of IoT devices could render the proposed countermeasures impractical.

Although SSL/TLS provides robust security, its feasibility on constrained IoT devices requires careful consideration. Studies on alternative solutions such as lightweight encryption, access control, intrusion detection, blockchain, and proactive defense strategies can help implement effective countermeasures without compromising the performance and battery life of IoT devices.

A. ENCRYPTION

Ensuring data security and privacy is paramount in MQTT communications. While MQTT inherently supports the SSL/TLS protocol, offering a layer of encryption to thwart potential MiTM attacks [47], this approach is not without its challenges. Primarily, the added encryption layer can be resource-intensive, posing challenges for IoT devices that are typically constrained in terms of computational resources [76]. Furthermore, while SSL/TLS ensures data in transit remains encrypted, once the data reaches its destination, such as a broker, it is often decrypted, leaving it vulnerable if the broker is compromised [46]. To address these challenges, payload encryption has been introduced alongside the SSL/TLS protocol. This method ensures end-to-end encryption, meaning data remains encrypted even when at rest. Various encryption techniques, both symmetric and asymmetric, have been proposed for this purpose. For instance, a comprehensive security framework for MQTT has been suggested that combines the strengths of both Public-key and Secret-key cryptography. This dual approach not only ensures data confidentiality but also offers granular access control [77]. Another innovative solution leverages the Elliptic Curve Cryptography (ECC) Integrated Encryption Scheme to offer three distinct security levels for MQTT communications. This approach allows for the secure transmission of unaltered, non-sensitive data and ensures that only authenticated publishers can send data to authenticated subscribers [78]. In addition, considering that IoT devices are resource-constrained, a dynamic IoT security system using secured MQTT, ECC encryption, and timestamps o protect against data tampering and Replay attacks is then proposed. By adjusting encryption key strength based on residual energy and using wake-up patterns, it optimizes energy consumption [103]. Complementing this approach, another study proposes a highly secure and efficient Remote User Authentication (RUA) scheme for IoT environments. This scheme also utilizes ECC and secure, lightweight key exchange to ensure confidentiality, integrity, and availability and is validated through formal and informal security analyses [104].

B. ACCESS CONTROL

Ensuring that only authorized entities can access specific resources is a cornerstone of cybersecurity. Access control mechanisms play a pivotal role in defining and enforcing these permissions, especially in the context of MQTT. One common method to regulate access is through ACLs, specifying which users or system processes are granted access to specific system objects and what operations are allowed on those objects [79]. A study [80] employed ACLs to encrypt data, store ambiguous data, and limit access only to authorized users. However, the efficiency of this method can vary depending on the nature of the messages being processed. In addition to traditional ACLs, there is a growing interest in more dynamic and flexible access control mechanisms. One such approach is the Attribute-Based Access Control (ABAC) framework [81]. Instead of relying on predefined permissions, ABAC uses attributes to define access control rules and make access decisions. A decentralized ABAC framework tailored for MQTT has been introduced, which is particularly useful for data sharing in MQTT bridging environments. This approach not only offers a more granular level of control but also operates with a relatively low overhead, making it a promising solution for IoT environments [82]. However, it can scale up to thousands of devices in IoT environment, therefore, a scalable authentication and authorization procedure have proposed to address this issue with OAuth2-based authentication and authorization through the ACE-MQTT profile [106]. The implementation includes an Authorization

Server for client registration and policy management, and an MQTT broker to enforce authentication. Practical experimentation demonstrates that this approach is scalable and provides robust security for MQTT systems, with manageable CPU, memory, network, and energy costs, making it suitable for moderately constrained devices [106].

C. INTRUSION DETECTION

Intrusion Detection Systems (IDS) serve as a second line of defense, monitoring network traffic or system activities for malicious actions or policy violations. In the context of MQTT, IDS can be particularly effective in detecting and mitigating threats like DoS, Distributed Denial of Service (DDoS), and buffer overflow attacks [83]. Several datasets tailored for MQTT have been developed to aid researchers in this domain [84]–[86].

A study [83] introduced an IDS equipped with a parsing engine specifically designed for MQTT. While this rule-based approach can effectively detect certain types of vulnerabilities, it might not be as adaptable or comprehensive in identifying new or evolving threats.

With the advent of ML and deep learning, there is potential to enhance the capabilities of IDS. These techniques can be trained to recognize patterns and anomalies in data, making them more adept at identifying malicious activities. A study [87] explored various classification models suitable for IDS applications, suggesting the potential of ML in this domain. Further, a deep learning approach using Deep Neural Networks (DNN) was proposed by [88] to enhance intrusion detection in MQTT-based protocols. Such advanced techniques can offer a more dynamic and adaptive approach to threat detection, ensuring a higher level of security for MQTT systems. Most recently, a method using centralized and federated learning was designed, which achieved over 80% accuracy of detection on the MQTT-IoT-IDS2020 dataset for both centralized and federated models [89].

D. BLOCKCHAIN

Blockchain, with its decentralized and immutable nature, offers a promising solution to many security challenges in the IoT domain [99], which further enhances the data integrity, security, and privacy [100]. Its applications in MQTT security have been explored in recent years [90], particularly in the realm of authentication. In a study [91], an innovative One-Time Password (OTP) authentication mechanism was introduced for MQTT. Utilizing the Ethereum blockchain provides an out-of-band channel for second-factor authentication. This not only ensures the authentication of both local and remote nodes but also bolsters trust and accountability in the system. Moreover, the use of Ethereum's smart contracts ensures that user privacy remains intact, as these contracts can execute predefined operations without revealing sensitive user data. Another research [45] echoes a similar approach, proposing an OTP authentication system for MQTT. By harnessing the power of the Ethereum blockchain, these solutions create a separate, dedicated channel for secondary authentication, bolstering the security of MQTT communications. Akshatha and Dilip Kumar [92] proposed an approach based on Blockchain Sharding and suggested this method outperformed TLS and previous blockchain-based mechanisms regarding bandwidth computation and many other metrics.

E. PROACTIVE DEFENSE

While some researchers have focused on designing and implementing MQTT honeypots with the primary objective of collecting relevant data and gaining insights into attacker behaviors [93], [94], others have created tools and fuzzers intended to actively assess and identify vulnerabilities within the MQTT ecosystem [32], [75], [95], [96]. In addition, researchers designed an application named IoTPenn [97], aiming at performing penetration testing when designing and evaluating the MQTT network. Despite these defensive strategies, there is a noticeable gap in the current research landscape when it comes to proactive defense strategies.

V. CONCLUSION AND FUTURE DIRECTIONS

The IoT has undeniably revolutionized our lives, seamlessly integrating itself into our daily routines. However, this rapid growth has come with a significant challenge: securing the vast and ever-expanding network of connected devices. Traditional IT security approaches struggle to address the unique limitations of resource-constrained IoT devices. MQTT, a lightweight messaging protocol, has emerged as a popular choice for its efficiency, but its widespread adoption has also exposed it to new attack vectors. Malicious actors exploit vulnerabilities in authentication, communication channels, and message delivery to disrupt operations, steal data, or compromise entire systems. This study has delved into the existing body of literature on MQTT security. We have explored the various attack methods employed by malicious actors, analyzed strategies for countermeasures, and discussed the persisting challenges that continue to demand attention. Our contribution includes the introduction of a taxonomy for the components in the MQTT security ecosystem. This framework sheds light on contemporary attack methods and exposes vulnerabilities within the protocol itself, as well as in its implementation and deployment. Additionally, we have provided an in-depth analysis of recent attacks, their potential repercussions, and existing countermeasures. By equipping researchers, developers, and security professionals with a comprehensive understanding of the evolving landscape of MQTT security threats, this paper aims to foster the development of robust defense mechanisms. Securing the ever-expanding IoT ecosystem is crucial to ensure its continued growth and potential in our interconnected world.

Future research in MQTT security must address the evolving threat landscape and the need for advanced protective measures. A critical area of focus is enhancing security testing by integrating fuzzing techniques with static analysis, symbolic execution, and data flow analysis to improve the detection of zero-day vulnerabilities. Additionally, leveraging ML for test case generation can significantly enhance

security testing for MQTT v5.0, which is anticipated to see widespread adoption. This approach will make security testing more robust and effective. Developing lightweight security solutions is another vital direction. Research should prioritize creating efficient security mechanisms that do not compromise the performance of IoT devices. This involves exploring new cryptographic methods and enhancing existing ones to protect against a range of threats, including DoS attacks, identity spoofing, information disclosure, and data tampering. The aim is to ensure that security measures are both strong and resource-efficient. A comprehensive security strategy that combines protocol-level features with supplementary measures such as SSL/TLS, user authentication, and monitoring mechanisms is essential. This holistic approach will ensure data confidentiality, integrity, and availability across diverse IoT environments. Conducting rigorous security evaluations and developing best practices for secure MQTT broker implementations are crucial steps to enhance overall system security.

The advent of quantum computing poses new challenges and opportunities for MQTT security. Research into post-quantum cryptography is crucial to developing encryption methods that can withstand the computational power of quantum computers. This involves creating algorithms that remain secure against quantum attacks, ensuring the longevity and resilience of MQTT communications in the face of future technological advancements.

While this review provides a comprehensive analysis of the security challenges and mitigation strategies associated with the MQTT ecosystem, it is essential to acknowledge certain limitations. Firstly, the scope of the study is inherently constrained by the rapidly evolving nature of IoT technologies and security threats. As a result, some emerging vulnerabilities and the latest mitigation techniques might not be fully covered. Additionally, the research predominantly relies on existing literature and available data, which may not capture the most recent advancements or unpublished findings in the field.

REFERENCES

[1] K. Ashton *et al.*, "That 'internet of things' thing," *RFID journal*, vol. 22, no. 7, pp. 97–114, 2009.

[2] R. Minerva, A. Biru, and D. Rotondi, "Towards a definition of the internet of things (iot)," *IEEE Internet Initiative*, vol. 1, no. 1, pp. 1–86, 2015.

[3] P. Bellini, P. Nesi, and G. Pantaleo, "Iot-enabled smart cities: A review of concepts, frameworks and key technologies," *Applied Sciences*, vol. 12, no. 3, p. 1607, 2022.

[4] X. Wang, "The impact of iot on news media in the smart age," *Mobile Information Systems*, vol. 2022, 2022.

[5] D. Kumar, R. K. Singh, R. Mishra, and S. F. Wamba, "Applications of the internet of things for optimizing warehousing and logistics operations: A systematic literature review and future research directions," *Computers & Industrial Engineering*, p. 108455, 2022.

[6] P. Singh, Z. Elmi, V. K. Meriga, J. Pasha, and M. A. Dulebenets, "Internet of things for sustainable railway transportation: Past, present, and future," *Cleaner Logistics and Supply Chain*, vol. 4, p. 100065, 2022.

[7] N. N. Dlamini and K. Johnston, "The use, benefits and challenges of using the internet of things (iot) in retail businesses: A literature review," in *2016 international conference on advances in computing and communication engineering (ICACCE)*. IEEE, 2016, pp. 430–436.

[8] I. Cvitić, D. Peraković, M. Periša, and M. Botica, "Novel approach for detection of iot generated ddos traffic," *Wireless Networks*, vol. 27, no. 3, pp. 1573–1586, 2021.

[9] N. Waheed, F. Khan, S. Mastorakis, M. A. Jan, A. Z. Alalmaie, and P. Nanda, "Privacy-enhanced living: A local differential privacy approach to secure smart home data," in *2023 IEEE International Conference on Omni-layer Intelligent Systems (COINS)*. IEEE, 2023, pp. 1–6.

[10] S. Laghari, S. Manickam, S. Karuppayah, A. Al-Ani, and S. U. Rehman, "Cyberattacks and vociferous implications on secs/gem communications in industry 4.0 ecosystem," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 7, 2021.

[11] A. T. Ebrahem, M. M. M. Al-Hatab, E. Y. Abd Al-Jabbar, W. H. Alkhaled, and Z. H. Al-Sawaff, "Using iot technology for monitoring alzheimer's and elderly patients," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 31, no. 2, pp. 986–994, 2023.

[12] S. H. Talib, L. A. Abdul-Rahaim, A. J. Alrubaie, and I. M. Raseed, "Design smart hospital system based on cloud computing," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 29, no. 2, pp. 797–807, 2023.

[13] 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 *2016 13th international conference on new technologies for distributed systems (NOTERE)*. IEEE, 2016, pp. 1–6.

[14] S. Elhadi, A. Marzak, N. Sael, and S. Merzouk, "Comparative study of iot protocols," *Smart Application and Data Analysis for Smart Cities (SADASC'18)*, 2018.

[15] 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)*. IEEE, 2016, pp. 290–295.

[16] M. M. A. Muslam, "Enhancing security in vehicle-to-vehicle communication: A comprehensive review of protocols and techniques," 2023.

[17] "Shodan Search Engine," https://www.shodan.io/dashboard, accessed: May 17, 2023.

[18] M. Mijwil, O. J. Unogwu, Y. Filali, I. Bala, and H. Al-Shahwani, "Exploring the top five evolving threats in cybersecurity: An in-depth overview," *Mesopotamian journal of cybersecurity*, vol. 2023, pp. 57–63, 2023.

[19] S. U. A. Laghari, S. Manickam, A. K. Al-Ani, S. U. Rehman, and S. Karuppayah, "Secs/gemsec: A mechanism for detection and prevention of cyber-attacks on secs/gem communications in industry 4.0 landscape," *IEEE Access*, vol. 9, pp. 154 380–154 394, 2021.

[20] A. Jaisan, S. Manickam, S. Laghari, S. U. Rehman, and S. Karuppayah, "Secured secs/gem: A security mechanism for m2m communication in industry 4.0 ecosystem," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 8, 2021.

[21] A. Qashlan, P. Nanda, and M. Mohanty, "Differential privacy model for blockchain based smart home architecture," *Future Generation Computer Systems*, vol. 150, pp. 49–63, 2024.

[22] W. Zack, "Exposed iot servers let hackers unlock prison cells, modify pacemakers," 2023, accessed: May 13, 2023. [Online]. Available: https://www.zdnet.com/

[23] 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.

[24] P. Oza and D. Kamdar, "A review on security approaches of mqtt protocol with respect to internet of things," 2020.

[25] F. Chen, Y. Huo, J. Zhu, and D. Fan, "A review on the study on mqtt security challenge," in *2020 IEEE International Conference on Smart Cloud (SmartCloud)*. IEEE, 2020, pp. 128–133.

[26] A. J. Hintaw, S. Manickam, M. F. Aboalmaaly, and S. Karuppayah, "Mqtt vulnerabilities, attack vectors and solutions in the internet of things (iot)," *IETE Journal of Research*, vol. 69, no. 6, pp. 3368–3397, 2023.

[27] D. B. Ansari, A.-U. Rehman, and R. Ali, "Internet of things (iot) protocols: a brief exploration of mqtt and coap," *International Journal of Computer Applications*, vol. 179, no. 27, pp. 9–14, 2018.

[28] E. Atilgan, I. Ozcelik, and E. N. Yolacan, "Mqtt security at a glance," in *2021 International Conference on Information Security and Cryptology (ISCTURKEY)*. IEEE, 2021, pp. 138–142.

[29] M. Harsha, B. Bhavani, and K. 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)*. IEEE, 2018, pp. 2244–2250.

[30] J. Kotak, A. Shah, and P. Rajdev, "A comparative analysis on security of mqtt brokers," 2019.

[31] Z. Jin, L. Xing, Y. Fang, Y. Jia, B. Yuan, and Q. Liu, "P-verifier: Understanding and mitigating security risks in cloud-based iot access policies," in *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, 2022, pp. 1647–1661.

[32] H. Xu, M. Yu, Y. Wang, Y. Liu, Q. Hou, Z. Ma, H. Duan, J. Zhuge, and B. Liu, "Trampoline over the air: Breaking in iot devices through mqtt brokers," in *2022 IEEE 7th European Symposium on Security and Privacy (EuroS&P)*. IEEE, 2022, pp. 171–187.

[33] Y. Jia, L. Xing, Y. Mao, D. Zhao, X. Wang, S. Zhao, and Y. Zhang, "Burglars' iot paradise: Understanding and mitigating security risks of general messaging protocols on iot clouds," in *2020 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2020, pp. 465–481.

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

[35] K. Jaikumar, T. Brindha, T. Deepalakshmi, and S. Gomathi, "Iot assisted mqtt for segregation and monitoring of waste for smart cities," in *2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS)*. IEEE, 2020, pp. 887–891.

[36] H. G. Hamid and Z. T. Alisa, "A survey on iot application layer protocols," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 21, no. 3, pp. 1663–1672, 2021.

[37] M. Sălăgean and D. Zinca, "Iot applications based on mqtt protocol," in *2020 international symposium on electronics and telecommunications (ISETC)*. IEEE, 2020, pp. 1–4.

[38] S. Andy, B. Rahardjo, and B. Hanindhito, "Attack scenarios and security analysis of mqtt communication protocol in iot system," in *2017 4th International conference on electrical engineering, computer science and informatics (EECSI)*. IEEE, 2017, pp. 1–6.

[39] M. Bender, E. Kirdan, M.-O. Pahl, and G. Carle, "Open-source mqtt evaluation," in *2021 IEEE 18th Annual Consumer Communications & Networking Conference (CCNC)*. IEEE, 2021, pp. 1–4.

[40] G. Kim, S. Kang, J. Park, and K. Chung, "An mqtt-based context-aware autonomous system in onem2m architecture," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8519–8528, 2019.

[41] E. Gamess, T. N. Ford, and M. Trifas, "Performance evaluation of a widely used implementation of the mqtt protocol with large payloads in normal operation and under a dos attack," in *Proceedings of the 2021 ACM Southeast Conference*, 2021, pp. 154–162.

[42] D. Kant, A. Johannsen, and R. Creutzburg, "Analysis of iot security risks based on the exposure of the mqtt protocol," *Electronic Imaging*, vol. 2021, no. 3, pp. 96–1, 2021.

[43] A. Alkhafajee, A. M. A. Al-Muqarm, A. H. Alwan, and Z. R. Mohammed, "Security and performance analysis of mqtt protocol with tls in iot networks," in *2021 4th International Iraqi Conference on Engineering Technology and Their Applications (IICETA)*. IEEE, 2021, pp. 206–211.

[44] A. Poulter, S. Johnston, and S. Cox, "Srup: The secure remote update protocol. 2016 ieee 3rd world forum on internet of things (wf-iot)," pp. 42–47, 2016.

[45] F. Buccafurri and C. Romolo, "A blockchain-based otp-authentication scheme for constrainded iot devices using mqtt," in *Proceedings of the 2019 3rd International Symposium on Computer Science and Intelligent Control*, 2019, pp. 1–5.

[46] N. Menyah, "A real time demonstrative analysis of lightweight payload encryption in resource constrained devices based on mqtt," Ph.D. dissertation, Sakarya Universitesi (Turkey), 2017.

[47] M. Aleesha and C. Laseena, "Mqtt protocol for resource constrained iot applications: A review."

[48] A. Al-Ani, W. K. Shen, A. K. Al-Ani, S. A. Laghari, and O. E. Elejla, "Evaluating Security of MQTT protocol in Internet of Things," in *2023 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*. IEEE, 2023, pp. 502–509.

[49] H. Wong and T. Luo, "Man-in-the-middle attacks on mqtt-based iot using bert based adversarial message generation," in *KDD 2020 AIoT Workshop*.

[50] S. N. Firdous, Z. Baig, C. Valli, and A. Ibrahim, "Modelling and Evaluation of malicious attacks against the IoT MQTT protocol," in *2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*. IEEE, 2017, pp. 748–755.

[51] I. Vaccari, M. Aiello, and E. Cambiaso, "Slowite, a novel denial of service attack affecting mqtt," *Sensors*, vol. 20, no. 10, p. 2932, 2020.

[52] I. Vaccari, M. Aiello, and E. Cambiaso, "Slowtt: A slow denial of service against iot networks," *Information*, vol. 11, no. 9, p. 452, 2020.

[53] F. Yamaguchi, C. Wressnegger, H. Gascon, and K. Rieck, "Chucky: Exposing missing checks in source code for vulnerability discovery," in *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, 2013, pp. 499–510.

[54] "NVD - CVE-2017-7653," https://nvd.nist.gov/vuln/detail/CVE-2017-7653, National Vulnerability Database, accessed: Nov. 21, 2023.

[55] "NVD - CVE-2020-13821," https://nvd.nist.gov/vuln/detail/CVE-2020-13821, National Vulnerability Database, accessed: Nov. 21, 2023.

[56] F. Faisal and H. T. Elshoush, "Input validation vulnerabilities in web applications: Systematic review, classification, and analysis of the current state-of-the-art," *IEEE Access*, 2023.

[57] E. Fernandes, J. Jung, and A. Prakash, "Security analysis of emerging smart home applications," in *2016 IEEE symposium on security and privacy (SP)*. IEEE, 2016, pp. 636–654.

[58] B. Yuan, Y. Jia, L. Xing, D. Zhao, X. Wang, and Y. Zhang, "Shattered chain of trust: Understanding security risks in {Cross-Cloud}{IoT} access delegation," in *29th USENIX security symposium (USENIX security 20)*, 2020, pp. 1183–1200.

[59] Z. Wang, N. Wang, X. Su, and S. Ge, "An empirical study on business analytics affordances enhancing the management of cloud computing data security," *International Journal of Information Management*, vol. 50, pp. 387–394, 2020.

[60] R. Kumar and R. Goyal, "On cloud security requirements, threats, vulnerabilities and countermeasures: A survey," *Computer Science Review*, vol. 33, pp. 1–48, 2019.

[61] H. Abusaimeh, "Virtual machine escape in cloud computing services," *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 7, 2020.

[62] Y. Yang, W. Shen, B. Ruan, W. Liu, and K. Ren, "Security challenges in the container cloud," in *2021 Third IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA)*. IEEE, 2021, pp. 137–145.

[63] P. Paganini, "Over 3.7m accounts were compromised in the flexbooker data breach," https://securityaffairs.com/126409/data-breach/flexbooker-data-breach.html, 2024, accessed: Jan. 29, 2024.

[64] "Report: Conferencing service exposes private customers' meetings," https://www.websiteplanet.com/blog/civicom-leak-report/, Website Planet, 2024, accessed: Jan. 29, 2024.

[65] N.-G. Vasilescu, P. Pocatilu, and M. Doinea, "Iot security challenges for smart homes," in *Education, Research and Business Technologies: Proceedings of 21st International Conference on Informatics in Economy (IE 2022)*. Springer, 2023, pp. 41–49.

[66] A. C. Jose and R. Malekian, "Smart home automation security: a literature review," *SmartCR*, vol. 5, no. 4, pp. 269–285, 2015.

[67] A. Smith, "Strange wi-fi spots may harbor hackers: Id thieves may lurk behind a hot spot with a friendly name," *The Dallas Morning News, Knight Ridder Tribune Business News, Washington, DC: May*, vol. 9, 2007.

[68] A. Bartoli, E. Medvet, and F. Onesti, "Evil twins and wpa2 enterprise: A coming security disaster?" *computers & security*, vol. 74, pp. 1–11, 2018.

[69] E. Metalidou, C. Marinagi, P. Trivellas, N. Eberhagen, C. Skourlas, and G. Giannakopoulos, "The human factor of information security: Unintentional damage perspective," *Procedia-Social and Behavioral Sciences*, vol. 147, pp. 424–428, 2014.

[70] B. M. Bowen, R. Devarajan, and S. Stolfo, "Measuring the human factor of cyber security," in *2011 IEEE International Conference on Technologies for Homeland Security (HST)*. IEEE, 2011, pp. 230–235.

[71] M. Ahmed, S. Jaidka, and N. I. Sarkar, "Security in decentralised computing, iot and industrial iot," *Industrial IoT: Challenges, Design Principles, Applications, and Security*, pp. 191–211, 2020.

[72] C. Ataç and S. Akleylek, "Iot Çağında güvenlik tehditleri ve Çözümleri Üzerine bir araştırma," *Eur. J. Sci. Technol.*, pp. 36–42, Mar 2019.

[73] B. Eshete, A. Villafiorita, and K. Weldemariam, "Early detection of security misconfiguration vulnerabilities in web applications," in *2011 Sixth International Conference on Availability, Reliability and Security*. IEEE, 2011, pp. 169–174.

[74] D. Abeles and M. Zioni, "Mqtt-pwn documentation," 2018.

[75] A. Palmieri, P. Prem, S. Ranise, U. Morelli, and T. Ahmad, "Mqttsa: A tool for automatically assisting the secure deployments of mqtt brokers,"

in *2019 IEEE World Congress on Services (SERVICES)*, vol. 2642. IEEE, 2019, pp. 47–53.

[76] M. H. Elgazzar, "Perspectives on m2m protocols," in *2015 IEEE Seventh International Conference on Intelligent Computing and Information Systems (ICICIS)*. IEEE, 2015, pp. 501–505.

[77] L. Bisne and M. Parmar, "Composite secure mqtt for internet of things using abe and dynamic s-box aes," in *2017 Innovations in Power and Advanced Computing Technologies (i-PACT)*. IEEE, 2017, pp. 1–5.

[78] L. Malina, G. Srivastava, P. Dzurenda, J. Hajny, and R. Fujdiak, "A secure publish/subscribe protocol for internet of things," in *Proceedings of the 14th international conference on availability, reliability and security*, 2019, pp. 1–10.

[79] R. S. Sandhu and P. Samarati, "Access control: principle and practice," *IEEE communications magazine*, vol. 32, no. 9, pp. 40–48, 1994.

[80] Y. Upadhyay, A. Borole, and D. Dileepan, "Mqtt based secured home automation system," in *2016 Symposium on Colossal Data Analysis and Networking (CDAN)*. IEEE, 2016, pp. 1–4.

[81] V. C. Hu, D. Ferraiolo, R. Kuhn, A. R. Friedman, A. J. Lang, M. M. Cogdell, A. Schnitzer, K. Sandlin, R. Miller, K. Scarfone *et al.*, "Guide to attribute based access control (abac) definition and considerations (draft)," *NIST special publication*, vol. 800, no. 162, pp. 1–54, 2013.

[82] P. Colombo, E. Ferrari, and E. D. Tümer, "Regulating data sharing across mqtt environments," *Journal of Network and Computer Applications*, vol. 174, p. 102907, 2021.

[83] M. Husnain, K. Hayat, E. Cambiaso, U. U. Fayyaz, M. Mongelli, H. Akram, S. Ghazanfar Abbas, and G. A. Shah, "Preventing mqtt vulnerabilities using iot-enabled intrusion detection system," *Sensors*, vol. 22, no. 2, p. 567, 2022.

[84] H. Hindy, E. Bayne, M. Bures, R. Atkinson, C. Tachtatzis, and X. Bellekens, "Machine learning based iot intrusion detection system: An mqtt case study (mqtt-iot-ids2020 dataset)," in *International networking conference*. Springer, 2020, pp. 73–84.

[85] I. Vaccari, G. Chiola, M. Aiello, M. Mongelli, and E. Cambiaso, "Mqttset, a new dataset for machine learning techniques on mqtt," *Sensors*, vol. 20, no. 22, p. 6578, 2020.

[86] H. Siddharthan, T. Deepa, and P. Chandhar, "Senmqtt-set: An intelligent intrusion detection in iot-mqtt networks using ensemble multi cascade features," *IEEE Access*, vol. 10, pp. 33 095–33 110, 2022.

[87] H. Alaiz-Moreton, J. Aveleira-Mata, J. Ondicol-Garcia, A. L. Muñoz-Castañeda, I. García, C. Benavides *et al.*, "Multiclass classification procedure for detecting attacks on mqtt-iot protocol," *Complexity*, vol. 2019, 2019.

[88] M. A. Khan, M. A. Khan, S. U. Jan, J. Ahmad, S. S. Jamal, A. A. Shah, N. Pitropakis, and W. J. Buchanan, "A deep learning-based intrusion detection system for mqtt enabled iot," *Sensors*, vol. 21, no. 21, p. 7016, 2021.

[89] A. Omotosho, Y. Qendah, and C. Hammer, "Ids-ma: Intrusion detection system for iot mqtt attacks using centralized and federated learning," in *2023 IEEE 47th Annual Computers, Software, and Applications Conference (COMPSAC)*. IEEE, 2023, pp. 678–688.

[90] M. Katende, "Combining mqtt and blockchain to improve data security," in *3rd International Workshop on Emerging Trends in Software Engineering for Blockchain (WETSEB)*, 2020.

[91] F. Buccafurri, V. De Angelis, and R. Nardone, "Securing mqtt by blockchain-based otp authentication," *Sensors*, vol. 20, no. 7, p. 2002, 2020.

[92] P. Akshatha and S. D. Kumar, "Mqtt and blockchain sharding: An approach to user-controlled data access with improved security and efficiency," *Blockchain: Research and Applications*, vol. 4, no. 4, p. 100158, 2023.

[93] H. Shimada, K. Ito, H. Hasegawa, and Y. Yamaguchi, "Implementation of mqtt/coap honeypots and analysis of observed data," in *SECURWARE 2019, The Thirteenth International Conference on Emerging Security Information, Systems and Technologies*, vol. 10, 2019.

[94] M. Wang, "Understanding security flaws of iot protocols through honeypot technologies: Thingpot-an iot platform honeypot," 2017.

[95] S. Hernández Ramos, M. T. Villalba, and R. Lacuesta, "Mqtt security: A novel fuzzing approach," *Wireless Communications and Mobile Computing*, vol. 2018, pp. 1–11, 2018.

[96] B. Pearson, Y. Zhang, C. Zou, and X. Fu, "Fume: Fuzzing message queuing telemetry transport brokers," in *IEEE INFOCOM 2022-IEEE Conference on Computer Communications*. IEEE, 2022, pp. 1699–1708.

[97] A. Roets and B. L. Tait, "Iot-penn: A security penetration tester for mqtt in the iot environment," in *Cybersecurity in the Age of Smart Societies: Proceedings of the 14th International Conference on Global Security, Safety and Sustainability, London, September 2022*. Springer, 2023, pp. 141–157.

[98] Nagaraj, S., Kathole, A., Arya, L., Tyagi, N., Goyal, S., Rajawat, A., Raboaca, M., Mihaltan, T., Verma, C. & Suciu, G. Improved secure encryption with energy optimization using random permutation pseudo algorithm based on Internet of Thing in wireless sensor networks. *Energies*. 16, 8 (2022)

[99] Kathole, A., Katti, J., Dhabliya, D., Deshpande, V., Rajawat, A., Goyal, S., Raboaca, M., Mihaltan, T., Verma, C. & Suciu, G. Energy-Aware UAV Based on Blockchain Model Using IoE Application in 6G Network-Driven Cybertwin. *Energies*. 15 (2022), https://www.mdpi.com/1996-1073/15/21/8304

[100] Patil, S., Kathole, A., Kumbhare, S., Vhatkar, K. & Kimbahune, 5. A Blockchain-Based Approach to Ensuring the Security of Electronic Data. *International Journal Of Intelligent Systems And Applications In Engineering*. 12, 649-655 (2024), https://www.ijisae.org/index.php/IJISAE/article/view/4486

[101] Kathole, A., Vhatkar, K., Kumbhare, S., Katti, J. & Kimbahune, V. IoT-Based Smart Agriculture for Onion Plant Disease Management: A Comprehensive Approach. *International Journal Of Intelligent Systems And Applications In Engineering*. 12, 472 - (2024), https://ijisae.org/index.php/IJISAE/article/view/4612

[102] Kumbhare, S., Ubale, S., Dharmale, G., Mhala, N. & Gandhewar, N. IoT-Enabled Agricultural Waste Management for Sustainable Energy Generation. *International Journal Of Intelligent Systems And Applications In Engineering*. 12, 477 - (2024), https://ijisae.org/index.php/IJISAE/article/view/4613

[103] De Rango, F., Potrino, G., Tropea, M. & Fazio, P. Energy-aware dynamic Internet of Things security system based on Elliptic Curve Cryptography and Message Queue Telemetry Transport protocol for mitigating Replay attacks. *Pervasive And Mobile Computing*. 61 pp. 101105 (2020), https://www.sciencedirect.com/science/article/pii/S1574119219304705

[104] Patel, C. & Doshi, N. Secure Lightweight Key Exchange Using ECC for User-Gateway Paradigm. *IEEE Transactions On Computers*. 70, 1789-1803 (2021)

[105] Potrino, G., Rango, F. & Santamaria, A. Modeling and evaluation of a new IoT security system for mitigating DoS attacks to the MQTT broker. *2019 IEEE Wireless Communications And Networking Conference (WCNC)*. pp. 1-6 (2019)

[106] Michaelides, M., Sengul, C. & Patras, P. An Experimental Evaluation of MQTT Authentication and Authorization in IoT. *Proceedings Of The 15th ACM Workshop On Wireless Network Testbeds, Experimental Evaluation & CHaracterization*. pp. 69-76 (2021), https://doi.org/10.1145/3477086.3480838

[107] "Fofa search engine," https://en.fofa.info/, accessed: May 22, 2024.

[108] Syed, N., Baig, Z., Ibrahim, A. & Valli, C. Denial of service attack detection through machine learning for the IoT. *Journal Of Information And Telecommunication*. 4, 482-503 (2020)

[109] "Common Vulnerabilities and Exposures," https://www.cvedetails.com/, accessed: May 22, 2024.

SHAMS UL ARFEEN LAGHARI is a computer scientist and dedicated researcher, currently working as a Postdoctoral Researcher at the National Advanced IPv6 Centre (NAv6), Universiti Sains Malaysia (USM). He holds a Ph.D. in Internet Infrastructures Security from NAv6 at USM, consistently demonstrating an unwavering commitment to academic excellence. Dr. Laghari's academic journey includes institutions such as the University of Sindh, Jamshoro, Pakistan, where he earned a B.Sc. (Hons.) and M.Sc. in computer science, and PAF-KIET in Karachi, Pakistan, where he obtained an M.S. in computer science. Over his 24-year career in teaching and research, he has contributed at both undergraduate and graduate levels. He has notably held positions as the Head of the Department of Computer Science and as an Additional Director of the Quality Enhancement Cell (QEC) at Hamdard University in Karachi, Pakistan. Dr. Laghari's research interests span various domains, including cybersecurity, IoT, Industry 4.0, distributed systems, cloud computing, and mobile cloud computing.

WENHAO LI (Graduate Student Member, IEEE) received his Bachelor of Engineering degree in Information Security from Chengdu University of Information Technology, China, in 2019, Master of Computer Science degree in Cybersecurity from Arizona State University and MBA degree from Webster University, US, in 2022, respectively. He is currently pursuing a Ph.D. degree in Cybersecurity with the National Advanced IPv6 Centre, Universiti Sains Malaysia. He is also the CEO of Chengdu Meetsec Technology Co., Ltd., where he has successfully led many cybersecurity services, projects, and developments. His current research interests include a wide range of cybersecurity topics, including Anti-Phishing, Web Security & Privacy, Cloud Security and IoT Security.

SELVAKUMAR MANICKAM is the Director of the National Advanced IPv6 Centre and an Associate Professor specializing in cybersecurity, the Internet of Things, industry 4.0, cloud computing, big data, and machine learning. He has authored and coauthored more than 220 articles in journals, conference proceedings, and book reviews. He has graduated 18 Ph.D. students in addition to master's and bachelor's students. He has given several keynote speeches and dozens of invited lectures and workshops at conferences, international universities, and industry. He has given talks and training on Internet security, the Internet of Things, industry 4.0, IPv6, machine learning, software development, and embedded and OS kernel technologies at various organizations and seminars. He also lectures in various computer science and IT courses, including developing new courseware in tandem with current technology trends. He is involved in various organizations and forums locally and globally. Previously, he was with Intel Corporation and a few start-ups working in related areas before moving to academia. While building his profile academically, he is still very involved in industrial projects involving industrial communication protocol, robotic process automation, machine learning, and data analytics using opensource platforms. He also has experience in the building IoT, embedded, server, mobile, and web-based applications.

PRIYADARSI NANDA is a Senior Lecturer at the University of Technology Sydney (UTS) with more than 27 years of experience specialising in research and development of Cybersecurity, IoT security, Internet Traffic Engineering, wireless sensor network security and many more related areas. His most significant work has been in the area of Intrusion detection and prevention systems (IDS/IPS) using image processing techniques, Sybil attack detection in IoT based applications, intelligent firewall design. In Cybersecurity research, he has published over 80 high quality refereed research papers including Transactions in Computers, Transactions in Parallel Processing and Distributed Systems (TPDS), Future Generations of Computer Systems (FGCS) as well as many ERA Tier A/A* conference articles. In 2017, his work in cyber security research has earned him and his team the prestigious Oman research council's national award for best research. Dr. Nanda has successfully supervised 8 HDR at UTS (5 Ph.D. + 3 Masters), and currently, supervising 8 Ph.D. students.

AYMAN KHALLEL AL-ANI earned his Ph.D. in Advanced Computer Networks from Universiti Sains Malaysia (USM). Currently, he holds the position of Senior Lecturer in the Department of Cybersecurity Engineering Technology at Al Hikma University College in Baghdad, Iraq. His ongoing research focuses on various areas, including malware detection, web security, intrusion detection systems (IDS), intrusion prevention systems (IPS), network monitoring, the Internet of Things (IoT), IPv6 security, artificial intelligence, machine learning, data mining, and optimization algorithms.

SHANKAR KARUPPAYAH (Member, IEEE) received the B.Sc. degree in computer science from Universiti Sains Malaysia (USM), Malaysia, the M.Sc. degree in software systems engineering from KMUTNB, Thailand, and the Ph.D. degree in cyber security from Technische Universität Darmstadt, in 2016. He is currently a Senior Lecturer and a Researcher with the National Advanced IPv6 Research Centre (NAv6), USM. His main research interests include P2P botnets, distributed systems, and cyber security. He has authored and co-authored many articles in journals, workshops, and conference proceedings. He is a reviewer of many esteemed networks and security journals.