Article

Security of IoT application layer protocols: challenges and findings

Giuseppe Nebbione* and Maria Carla Calzarossa

University of Pavia, Department of Electrical, Computer and Biomedical Engineering, I-27100 Pavia (Italy)
* Correspondence: giuseppe.nebbione01@universitadipavia.it

Version February 28, 2020 submitted to Future Internet

Abstract: IoT technologies are becoming pervasive in public and private sectors and represent nowadays an integral part of our daily life. The advantages offered by these technologies are frequently coupled with serious security issues that are often not properly overseen or even ignored. The IoT threat landscape is extremely wide and complex and involves a large variety of hardware and software technologies. In this framework, the security of application layer protocols is of paramount importance since these protocols are at the basis of the communications among applications and services running on different IoT devices and on cloud/edge infrastructures. This paper offers a comprehensive survey of application layer protocol security by presenting the main challenges and findings. More specifically, the paper focuses on the most popular protocols devised in IoT environments for messaging/data sharing and for service discovery. The main threats of these protocols as well as the Common Vulnerabilities and Exposures (CVE) for their products and services are analyzed and discussed in detail. Good practices and measures that can be adopted to mitigate threats and attacks are also investigated. Our findings indicate that ensuring security at the application layer is very challenging. IoT devices are exposed to numerous security risks due to lack of appropriate security services in the protocols as well as to vulnerabilities or incorrect configuration of the products and services being deployed. Moreover, the constrained capabilities of these devices affect the types of security services that can be implemented.

Keywords: IoT; security; threat; mitigation; application layer protocols; CVE; MQTT; CoAP; mDNS; SSDP; AMQP; DDS; XMPP; good practices

1. Introduction

The IoT ecosystem encompasses a growing number of smart objects connected to the Internet and characterized by diverse capabilities, such as sensing, actuating, processing, storing and communicating [1,2]. These physical objects are becoming pervasive in many industry verticals (e.g., transportation, manufacturing, energy, oil, gas, healthcare), as well as in governments (e.g., smart cities, smart buildings) and in our daily life (e.g., smart homes) [3]. In fact, IoT technologies offer enormous potentials to consumers and industry. More precisely, they improve quality of life, increase operational efficiency and productivity, allow real-time decisions and create new business opportunities. These benefits are leading to an exponential increase of the number of connected devices that is expected to reach tens of billions in the next coming years. According to Gartner's estimates, Internet-connected-things will outnumber humans 4-to-1 by 2020. This expansion will have a strong economic effect. The McKinsey Global Institute predicts that IoT technologies could have an annual economic impact of 3.9 to 11.1 trillion USD worldwide by 2025.

Unfortunately, all these benefits are often coupled with many security risks and challenges. The main problem nowadays is the presence of many insecure IoT objects treated by their designers, manufacturers and even owners as dumb devices that in the hands of malicious hackers can be easily exploited to create serious economic and reputation damages, steal private data and even threaten safety. For example, a security hole on an implanted medical device might pose serious risks to patients. A distributed cyberattack on connected cars might easily gridlock entire cities.

IoT systems integrate and rely on a variety of enabling technologies, e.g., software modules, libraries, middleware, application programming interfaces, protocols, sensor and mobile networks, whose source and nature are often out of the control of organizations or individuals deploying these systems. The diversity of the devices and of the environments where they operate requires specific consideration of the potential security challenges.

In the complex IoT world, application layer protocols play a key role. In fact, they are at the basis of the communications among applications and services running on different IoT devices and on cloud/edge infrastructures. This paper offers a comprehensive analysis of the security risks and challenges affecting the most popular application layer protocols employed in IoT environments. In particular, the paper examines and classifies the potential security threats and attacks outlined in the protocol standards. To gain some further insights of whether/how security threats have materialized and of their actual impact, these threats are also studied under a different perspective, that is by analyzing the Common Vulnerabilities and Exposures (CVE) collected by MITRE for products and services devising the various protocols. Moreover, the paper investigates and discusses the measures and good practices proposed in the literature to enhance security and mitigate the associated risks.

The main contributions of this paper can be summarized as follows:

- Analysis and discussion of the potential security threats and attacks affecting the application layer protocols typical of IoT environments;
- Analysis and discussion of the CVEs affecting products and services based on these protocols;
- Analysis and discussion of good practices and countermeasures that could be applied to mitigate risks and enhance security.

The layout of this paper is as follows. Section 2 presents a general overview of IoT threat landscape, while Section 3 introduces and compares the application layer protocols considered in this paper. Sections 4 and 5 analyze the potential security risks and possible countermeasures of messaging and service discovery protocols, respectively. Section 6 summarizes and discusses the main findings of the analysis. Finally, Section 7 concludes the paper with some remarks.

2. Background

The IoT threat landscape is extremely wide and complex. Gartner predicts that over a quarter of all cyber attacks against businesses will be IoT-based by 2025. Nevertheless, nowadays the market prioritizes convenience and price over security that is seldom built by design. Moreover, there is a general lack of defense in aging firmware or architectures. Similarly, little consideration is given to promoting user awareness and education.

Vulnerabilities of IoT devices are discovered with increasing frequency and their exploitation continues to accelerate and escalate. The evaluations of the security and privacy of consumer IoT devices presented in [4,5] show that most devices display some form of vulnerability, although some devices have a better security posture than others. In 2016 the Mirai botnet used many thousands hijacked IoT devices (e.g., security cameras, DVRs) as attack vectors to engage in a huge Distributed Denial of Service (DDoS) attack whose peak traffic reached as many as 1Tbps. In summer 2019, Armis discovered a batch of 11 zero-day vulnerabilities affecting VxWorks, a very popular real-time operating system used for a wide range of commercial and consumer IoT devices.

Even though large scale attacks cause big damages, small scale attacks can be even more dangerous since they often go unnoticed and undetected for quite a long time. Therefore, it is compelling to strengthen cybersecurity by identifying what needs to be secured and developing countermeasures that take account of the specific characteristics and physical limitations of individual devices.

It is worth noting that IoT security is not only a technical issue. Policy makers have acknowledged its importance for businesses, citizens and the whole society by supporting and pushing the definition of proper safety, security and privacy measures and practices to fight security threats. The European Cybersecurity Act – entered into force in June 2019 – is a response to cybersecurity challenges. The act also envisions rules for EU-wide cybersecurity certification of products, processes and services. Similarly, the US Congress's Internet of Things (IoT) Cybersecurity Improvement Acts 2017 and 2019 specifically leverage the Federal Government procurement power to encourage minimal cybersecurity operational standards for Internet-connected devices purchased by Federal agencies and put forward some recommendations regarding the minimum information security requirements for managing cybersecurity risks associated with such devices.

Another important issue to be addressed in the framework of IoT security refers to user awareness and education regarding the purchase and use of IoT devices. Although the use of default credentials associated with IoT devices represents one of the biggest security weaknesses, many users are not aware of this vulnerability and leave these passwords unchanged. The IOT Consumer TIPS Act of 2017 tries to respond to this issue by requiring the development of specific educational resources.

IoT security has also been extensively analyzed in the literature. Research efforts studied this challenging topic under different perspectives. In recent years, several surveys aimed at reviewing and classifying these efforts have been published (see, e.g., [6–16]). More specifically, Aly et al. [6] consider the layers of the IoT reference models and present a systematic literature review aimed at providing guidelines for researchers and practitioners interested in understanding security issues. The focus of Ammar et al. [7] is the security of IoT frameworks and platforms adopted to develop industrial and consumer applications. The study compares the architectures of the frameworks and discusses the approaches devised for ensuring security and privacy. Mosenia and Jha [10] present a detailed analysis of the vulnerabilities affecting the edge-side layer of IoT (i.e., edge node, communication and edge computing) and outline the possible countermeasures against these attacks. Neshenko et al. [11] offer a multi-dimensional taxonomy of IoT vulnerabilities based on their classification. Zhou et al. [16] propose a set of features that uniquely characterize IoT devices, network subsystems and applications and discuss the potential threats and vulnerabilities associated with each feature as well as solutions and opportunities to tackle the threats.

Let us remark that most of the surveys on IoT security focus on specific aspects of the IoT ecosystem, such as networking infrastructures, deployment environments, whereas to the best of our knowledge, our paper is the first comprehensive survey addressing the security issues affecting application layer protocols.

3. Application layer protocols

As already discussed, communication protocols at the application layer are a fundamental component of the IoT ecosystem since they are at the basis of all the interactions among IoT devices and among IoT devices and cloud/edge infrastructure [17–19].

The typical functions implemented by these protocols deal with messaging and service discovery. In particular, messaging refers data sharing and exchanges among devices, while discovery refers to detecting devices and services being offered. Table 1 summarizes the main characteristics of the seven standard protocols analyzed in this paper, namely, five messaging protocols (i.e., MQTT, CoAP, AMQP, DDS and XMPP) and two service discovery protocols (i.e., mDNS and SSDP). As can be seen, the protocols differ for many aspects, such as architectural and interaction models and transport protocols. Some protocols use centralized, i.e., client/server, architectures, while others are based on fully distributed architectures. For example, for protocols such as MQTT and AMQP, the broker plays the server role and interacts with clients by receiving and forwarding messages. Message exchanges are in general implemented according to publish/subscribe or request/response models. Similarly, service discovery can be based on request/response or query/response models. It is also worth noting that some protocols offer fully reliable data transfer since they are built on top of the TCP transport protocol, while others – built on top of UDP – are loss-tolerant. In particular, service discovery protocols are based on UDP, whereas messaging protocols on TCP.

Protocol	Standard	Function		Architectural model		Interaction model		Transport protocol	
		messaging	discovery	c/s	decentralized	pub/sub	req/resp	TCP	UDP
MQTT	OASIS	●		●		●		●	
CoAP	IETF	●	○	●		○	●	○	●
AMQP	OASIS	●		●		●	○	●	
DDS	OMG	●	○		●	●	○	●	●
XMPP	IETF	●	○	●		●	●	●	
mDNS	IETF		●		●		●		●
SSDP	UPnP		●	●			●		●

Table 1. Summary of the main characteristics of the most popular application layer protocols for IoT environments. The bullets refer to native features of the protocols, while the circles to additional features supported by the protocols.

The choice of the application protocol depends on the nature of the IoT systems and their requirements. MQTT and CoAP are particularly suitable for services requiring data collection (e.g., sensor updates) in constrained environments. On the contrary, AMQP, DDS and XMPP address specific service requirements, namely, business messaging, instant messaging and online presence detection and real-time exchanges, respectively. In terms of service discovery, mDNS and SSDP are the protocols of choice for IoT environments.

Concerning security services, the solutions that ensure integrity and confidentiality of the exchanges and provide authentication and authorization mechanisms are very diverse. Messaging protocols generally support standard as well as custom security services, whereas service discovery protocols do not support any built-in security service. Therefore, the implementation of appropriate security solutions is left to developers.

As shown in Table 2, encryption mechanisms are available in all messaging protocols. For example, confidentiality is ensured by standard services such as TLS and DTLS, whereas authentication and authorization mechanisms are based on standard (i.e., SASL) or custom solutions.

Protocol	Authentication		Authorization	Confidentiality	
	SASL	Custom	Custom	TLS	DTLS
MQTT		●		●	
CoAP					●
AMQP	●			●	
DDS		●	●	●	●
XMPP	●		●	●	

Table 2. Summary of the security services supported by the messaging protocols.

It is important to outline the lack of security in the protocol design. Moreover, security services are generally considered optional and have to be explicitly enabled by developers. In turn, implementers tend to neglect these services in the development and configuration of their applications. Additionally, end-to-end encryption is often too expensive to cope with the constrained capabilities (e.g., bandwidth, computing power) of many IoT devices. Therefore, as we will discuss in the rest of the paper, devices are frequently exposed to security risks specific of the protocols as well as to risks typically encountered in networked environments.

In what follows, we offer a comprehensive analysis of these security issues. More specifically, for each protocol, our analysis considers the following aspects:

- Potential threats and security attacks;
- Good practices and countermeasures to mitigate the attacks.

The methodological approach followed in our study is based on the examination of the security specifications of the protocol standards and on the analysis of the CVEs collected in the National Vulnerability Database (NVD) over six years since 2014. In addition, we performed an extensive search and analysis of the literature as well as of the good practices proposed by public and private organizations, service providers and cybersecurity companies. In particular, we searched numerous websites and popular digital libraries and databases, such as ACM, IEEE, Springer, Google Scholar, Scopus.

4. Messaging protocols

This section focuses on messaging protocols used in IoT environments. In particular, we analyze in detail MQTT and CoAP because of their popularity and wide acceptance in these environments, while we briefly cover AMQP, DDS and XMPP since they find applications in IoT, even though they are not seen as a typical IoT solution.

4.1. MQTT

Message Queue Telemetry Transport (MQTT) is an open standard messaging protocol that has been around for more than twenty years (OASIS Standard). The protocol – widely used nowadays in the IoT context – is simple, lightweight and ideal for IoT scenarios where saving computing power and network bandwidth is the priority.

As already discussed, MQTT supports various authentication mechanisms as well as encryption based on TLS [20]. Nevertheless, these services are not sufficient to protect MQTT-enabled devices and in particular the broker component. It is worth mentioning that – as reported in the MQTT standard and as demonstrated at DEFCON 24 – many security risks are originated by broker misconfiguration and software vulnerabilities. These threats could be easily exploited for many malicious purposes.

From the analysis of the possible security threats of MQTT-enabled devices, we identified the potentially vulnerable processes and we produced the following classification:

- *Authentication*: the MQTT broker does not properly check the publisher/subscriber identity and does not block repeated authentication attempts. These vulnerabilities could grant an attacker the access to MQTT devices or could overload the broker and eventually make it crash;
- *Authorization*: the MQTT broker does not properly set the publishing/subscribing permissions. This vulnerability could grant an attacker the control over data or functions of MQTT devices;
- *Message delivery*: a publisher sends messages that cannot be delivered because of the lack of subscribers. This vulnerability could lead to significant degradation of broker performance;
- *Message validation*: a publisher sends messages containing disallowed characters that are not properly interpreted by brokers and subscribers. This vulnerability could be exploited to perform many different malicious attacks;
- *Message encryption*: clients and servers exchange messages in plaintext, thus allowing an attacker to eavesdrop and spoof the messages in transit. This vulnerability could be exploited to perform Man-in-The-Middle (MiTM) attacks.

The analysis of the CVEs affecting products and services based on MQTT offers an interesting overview of whether/how security threats have materialized and of their actual impact. More precisely, the NVD database includes 57 CVEs. Many of these vulnerabilities refer to the improper message validation category. In particular, crafted MQTT messages could easily make brokers unresponsive. For example, a malicious MQTT client could cause a stack overflow by simply sending a `SUBSCRIBE` packet containing at least 65,400 "/" characters (CVE-2019-11779). Similarly, a `CONNECT` packet combined with a malformed `UNSUBSCRIBE` request packet can be used to cause a Denial of Service (DoS) attack against the broker (CVE-2019-6241).

Other security issues refer to the authentication and authorization categories, as in the case of clients that set their username to "#", thus bypassing the access control mechanisms and subscribing to all MQTT topics (CVE-2017-7650). Figure 1 depicts the effects of this vulnerability where an attacker can access all information coming from all publishers, including sensitive data with serious consequences on confidentiality.

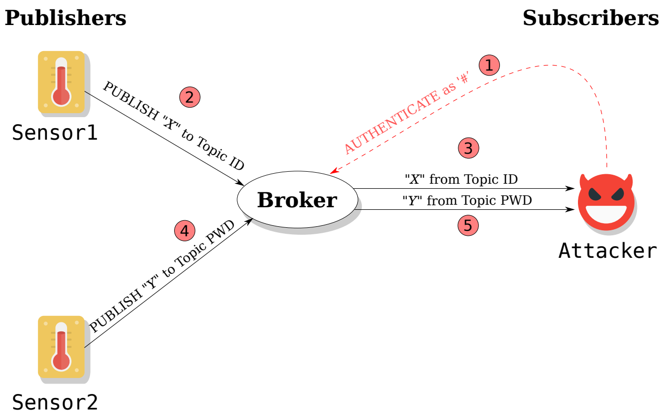

Figure 1. Example of access control vulnerability that allows an attacker to subscribe to all topics and receive all messages being published. The numbers refer to the temporal evolution of the MQTT interactions depicted in the figure.

In the literature, MQTT security threats have been investigated by Firdous et al. [21] who propose a model to identify the abilities of threat agents in carrying out attacks. Moreover, the paper discusses the possible exploitations of these attacks using realistic scenarios. For example, it shows that a Denial of Service attack – aimed at making a broker unresponsive or even crash – can be carried out by sending big messages or messages with high QoS levels. In addition, unauthorized publishing – aimed at physically damaging or disabling IoT devices – can be performed by means of privileged messages that grant an attacker remote control of these devices. Therefore, as these simple scenarios show, threats could seriously affect MQTT environments and compromise their availability as well as sensitive data being exchanged and stored.

4.1.1. Mitigations

To cope with security threats, the MQTT standard lists the mechanisms that should be included in MQTT implementations, namely:

- Authentication of users and devices;
- Authorization of access to server resources;
- Integrity of MQTT control packets and application data;
- Privacy of MQTT control packets and application data.

For each of these mechanisms the standard provides some general recommendations (e.g., re-authentication of long sessions, prevention of subscription to all topics, usage of VPNs). Nevertheless, it is often up to the developer to choose the mechanisms most appropriate to the specific application requirements. In addition, as pointed out by Perrone et al. [22], the standard mainly refers to simple scenarios and does not discuss details of complex scenarios, such as broker interconnections and synchronization mechanisms between brokers. Therefore, these issues require additional research efforts.

Even though the use of the TLS protocol is strongly recommended by the MQTT standard to ensure secure communication, TLS does not solve all security issues. In fact, it is well known that older versions of TLS, its misconfiguration and the use of weak cipher suites make protocols exposed to security attacks [23,24]. In addition, the implementation of TLS requires a significant computing power and network bandwidth which might not be available on constrained IoT devices.

In the literature, many papers focus on TLS with the objective of devising implementations more suitable to MQTT-enabled IoT devices (see, e.g., [25–33]). For example, to ensure message confidentiality and integrity, Dinculeana et al. [28] propose an approach based on the Blake2 algorithm [34]. This approach – very promising in terms of performance on constrained devices – is particularly appropriate in industrial environments where sensors and controllers exchange predictable data. Singh et al. [32] propose a secure version of MQTT which uses a new control packet, called `Spublish`, to publish encrypted data and takes advantage of the Cipher-text 232 Policy/Key Policy Attribute Based Encryption using lightweight Elliptic Curve Cryptography [35,36].

To introduce an enhanced access control mechanism on constrained devices where TLS is too expensive, Bali et al. [25] developed a lightweight authentication mechanism based on a chaotic algorithm. Similarly, Niruntasukrat et al. [30] propose an MQTT architecture based on a modified version of the OAuth framework [37] where two sets of credentials are used by the devices to access the broker.

Access control is also studied in [38,39]. More precisely, to enforce security policy rules, Neisse et al. [38] developed a connector that intercepts the messages exchanged by the broker and generates proper notifications that might lead to the execution of an enforcement action. Similarly, the mechanism proposed in [39] is based on the use of a proxy that monitors the exchanges between clients and servers.

Another problem addressed in the framework of TLS deals with the proper configuration of TLS-enabled devices. For this purpose, Alghamdi et al. [40] developed an automated software agent based on a state machine model to help the identification of TLS vulnerabilities. In particular, the agent checks possible misconfiguration by means of certificate validation.

In summary, our analysis has shown that the MQTT protocol supports a good number of security services although these services in general do not cope with all possible security risks affecting the protocol.

4.2. CoAP

Constrained Application Protocol (CoAP) is an emerging open web transfer protocol whose latest specifications are defined in RFC 7252 published in 2014 [41]. Although CoAP shares many characteristics with the HTTP protocol, it has been specifically designed for constrained devices with limited energy, processing power, storage space and transmission capabilities.

As already discussed, CoAP supports the usage of the Datagram Transport Layer Security (DTLS) protocol, a UDP implementation of the TLS protocol that provides equivalent security guarantees [42]. The DTLS binding for the CoAP protocol is defined in terms of four security modes that differ by authentication and key negotiation mechanisms and range from no security to certificate based security.

In this framework, it is up to developers to find the best tradeoff between performance/energy constraints and security requirements. Of course, the lack of appropriate security services could allow attackers to easily compromise CoAP environments.

From the analysis of the possible security threats of CoAP-enabled devices, we identified the potentially vulnerable processes and we produced the following classification:

- *Message parsing*: the processing logic of client and server parsers does not properly handle incoming messages. This vulnerability could affect CoAP node availability because of overload conditions and even open the ability to remotely execute arbitrary code on the node under attack;
- *Proxying and caching*: the access control mechanisms of proxies and caches are not properly implemented. This vulnerability could compromise their content, thus breaking confidentiality and integrity of CoAP messages;
- *Bootstrapping*: the setup of new CoAP nodes is not properly implemented. This vulnerability could grant unauthorized nodes the access to a CoAP environment;
- *Key generation*: the generation of cryptographic keys is not sufficiently robust. The usage of these keys could compromise CoAP nodes;
- *IP address spoofing*: by forging the IP addresses of CoAP nodes, an attacker can perform a variety of side attacks including the generation of spoofed response messages and acknowledgments as well as reflection/amplification attacks;

- *Cross-protocol exchanges*: an attacker sends a CoAP node a message with a spoofed IP address and a fake source port number; the response of this node will reach the node under attack and force it to interpret the received message according to the rules of the target protocol.

The analysis of the few CVEs affecting products and services based on CoAP suggests that these vulnerabilties materialize differently. In particular,according to our classification, the most common security issue refers to improper message parsing. For example, some CoAP libraries mishandle invalid options or certain exceptions when receiving specifically crafted messages (e.g., CVE-2018-12679, CVE-2018-12680). Other libraries are affected by overflow vulnerabilities while processing an incoming message (e.g., CVE-2019-17212). The exploitation of these vulnerabilities could have different impacts, such as memory leak, Denial of Service as well as remote code execution, thus leading to serious effects on the entire CoAP system.

The UDP protocol is also a vector used to attack the CoAP-enabled nodes. For example, certain CoAP server interfaces can be exploited for a Distributed Denial of Service attack using source IP address spoofing and traffic amplification. This vulnerability is a consequence of a specific response message mishandling (e.g., CVE-2019-9750).

4.2.1. Mitigations

The CoAP standard provides some general mitigation measures to cope with the types of threats and attacks discussed in the previous section. In particular, the standard strongly encourages the adoption of DTLS for securing CoAP nodes.

In the literature, several works focus on the identification of specific mitigation measures for different scenarios (see, e.g., [43–54]). In particular, the mitigations proposed by these works mainly focus on two aspects:

1. Access control mechanisms;
2. Secure communication.

In the framework of access control, a collection of general use cases for authentication and authorization in constrained environments is presented in [53]. The report identifies the main authorization problems arising during the life cycle of a device and provides a guideline for implementing effective solutions. Pereira et al. [50] developed a service-level access control on low-power devices. The proposed approach is based on the authentication of CoAP nodes and the usage of tickets to grant access to resources.

Another mitigation measure presented in the literature deals with secure node bootstrapping. This process is particularly important and its misconfiguration could compromise the entire network. In fact, it allows a node to collect the information necessary to join a CoAP-enabled network as an authenticated node. In this framework, Bergmann et al. [44] propose a three-step process to bootstrap a new node. The process starts with a discovery phase where the new node is detected. This node is then provided with keys to establish a secure communication channel. Finally, these keys are used to perform the actual configuration of the node itself.

In the framework of secure communication, Iglesias et al. [47] describe and compare the DTLS libraries supported by the CoAP implementations typically encountered in industrial IoT environments. The paper outlines the need to keep an eye to new security developments because of their relevance especially in these environments. Alghamdi et al. [55] compare the security services provided by IPSec and DTLS. This study shows that although both protocols have strengths and weaknesses, in general their overhead could be significant and drain resources of constrained devices. Several papers addressed these issues by focusing on the design of lightweight solutions to secure the communication channel between clients and servers. A header compression scheme for DTLS that leverages the 6LoWPAN standard is proposed in [52], while the problem of reducing the number of DTLS handshakes is addressed in [49]. More specifically, this work presents a group-oriented handshake between a

CoAP client and a group of CoAP servers that reduces the total computational requirements of the DTLS protocol.

Improvements of the DTLS protocol have also been studied from the perspective of the cryptographic algorithm. In particular, as shown in [43,45], the integration of DTLS over CoAP based on Elliptic Curve Cryptography helps in minimizing the computation overhead and ROM occupancy.

In summary, our analysis has shown that DTLS ensures confidentiality in CoAP environments. Nevertheless, lightweight solutions are to be sought to cope with the capabilities of constrained devices.

4.3. AMQP

Advanced Message Queuing Protocol (AMQP) is an open protocol for business messaging (OASIS Standard). The protocol offers sophisticated functionalities and is widely used nowadays in many scenarios where a reliable asynchronous communication between endpoints is needed.

Concerning security, AMQP supports the Simple Authentication and Security Layer (SASL) framework [56] for client authentication and TLS for ensuring integrity and confidentiality of communication. Let us remark that, unlike MQTT and CoAP, these security services are generally enabled by default, thus reducing the potential security risks. Nevertheless, according to the NVD database, a large variety of vulnerabilities have been discovered in the past six years in products and services based on AMQP. These vulnerabilities mainly involve the broker component and affect processes, such as access control, message and identity validation, message queue management. The effects of these vulnerabilities include privilege escalation, information disclosure, Denial of Service attacks, authentication and authorization bypass, remote code execution, traffic hijacking. More specifically, several vulnerabilities refer to the lack of hostname and certificate validation whose exploitation allows attackers to spoof identities and intercept traffic for MiTM attacks (e.g., CVE-2018-11087, CVE-2018-8119, CVE-2016-4467). Similarly, the lack of access control in the message queues reported by CVE-2019-3845 allows attackers to execute privileged commands. In addition, several CVEs suggest that the use of specifically crafted AMQP messages and of exposed shutdown commands makes it possible to achieve a Denial of Service attack (e.g., CVE-2015-7559, CVE-2017-15699, CVE-2015-0224, CVE-2015-1499).

Other security risks affecting AMQP environments are related to broker configuration. In fact, AMQP brokers are very complex and despite the presence of a web user interface their setup can be very challenging. Incorrect choices in the setup of message queues, exchanges, producers and consumers might lead to serious vulnerabilities. Moreover, the user interfaces might be affected by vulnerabilities typically encountered in the web domain (e.g., CVE-2015-0862, CVE-2016-0734, CVE-2017-4965). We finally outline that a simple – although very common – misconfiguration refers to the use of default login credentials that can be abused by an attacker to take control of a publicly exposed broker administrator interface and of the entire AMQP environment.

4.4. DDS

Data Distribution Service (DDS) is a data-centric standard protocol defined by the Object Management Group. The protocol is generally used to manage data exchanges between lightweight devices and large high-performance sensor networks as well as the cloud. While not being a typical IoT solution, DDS finds its application in some industrial deployments, such as air-traffic control, smart grid management, autonomous vehicles, transportation systems and healthcare services.

Concerning security, the DDS protocol offers a rich variety of mechanisms. Similarly to other messaging protocols, DDS supports both TLS and DTLS. Moreover, for ensuring confidentiality, integrity and authenticity of the exchanges, the newest OMG DDS security specification defines an architecture based on a set of built-in plugins. For example, plugins offer mechanisms for authentication and authorization of DataWriters and DataReaders, thus avoiding unauthorized publication and subscription. Nevertheless, both specification and plugins are affected by vulnerabilities. In particular, the handshake protocol used for permission attestation sends clear text information about participant capabilities, thus allowing attackers to discover potentially sensitive reachability information on a DDS network (CVE-2019-15135). As White et al. [57] reported, this vulnerability breaches the confidentiality of the connection and allows attackers to collect information that could be used for malicious purposes.

It is also important to point out that plugins per se do not ensure security of DDS environments. In particular, the two vulnerabilities discovered for the Access Control plugin could lead to unauthorized or unintended connections between participants (CVE-2019-15136, CVE-2019-15137).

Finally, it is worth mentioning that not every DDS product and service are compliant to the security specification and even compliant implementations might be affected by vulnerabilities. In fact, as shown in [58], node misconfiguration can be abused to perform malicious activities inside a DDS environment.

4.5. XMPP

Extensible Messaging and Presence Protocol (XMPP) is an open XML technology for real-time asynchronous communication between two or more entities. XMPP latest specifications are defined in RFCs 6120 [59] and 6121 [60].

The XMPP protocol provides robust security services by supporting SASL for the authentication process and the TLS for ensuring data confidentiality and integrity. Note that these services are built into the core specifications of the protocol, thus enabled by default. Nevertheless, the lack of end-to-end encryption support makes the protocol vulnerable to various types of threats. For example, an attacker could modify, delete, or replay stanzas or gain an unauthorized entry to a server. In addition to the security issues of the protocol, numerous vulnerabilities affect products and services based on XMPP. More specifically, slightly less than 100 CVEs – mainly referring to the authentication and message validation processes – have been discovered in the past six years. Frequent issues deal with insufficient controls on memory operations and inappropriate certificate verification as well as the presence of hard-coded accounts (e.g., CVE-2019-1845, CVE-2019-12855, CVE-2014-3451, CVE-2018-15720, CVE-2016-1307). These vulnerabilities allow a large variety of attacks with different effects, such as making the services unavailable, obtaining sensitive information or gaining access to XMPP servers.

Other vulnerabilities are associated with custom functionalities that can be easily built on top of the XMPP protocol. As discussed in [61] implementations of an extension used for communicating user avatar information allow attackers to breach data location.

A number of practices to mitigate security threats has been developed as extensions of XMPP in its XEP series. More precisely, XEP-0205 presents measures aimed at discouraging DoS attacks, while XEP-0178 focuses on the proper usage of certificates for SASL authentication. Nevertheless, several XEPs contain vulnerabilities related to the incorrect implementation of the XEPs themselves (e.g., CVE-2016-10376, CVE-2017-5602, CVE-2019-1000021). By exploiting these vulnerabilities, attackers could gain access to private data or impersonate users and perform social engineering attacks.

5. Service discovery protocols

This section focuses on the service discovery protocols typical of IoT environments, namely, mDNS and SSDP.

5.1. mDNS

Multicast Domain Name System (mDNS) is an open protocol widely used nowadays for service discovery and name resolution on local links [62]. This protocol, coupled with DNS-based Service Discovery (DNS-SD) [63], offers the flexibility required by environments where it is necessary to automatically integrate new devices and perform DNS-like operations without the presence of a conventional DNS server.

Unlike messaging protocols, the mDNS protocol does not provide any built-in security service. As a consequence, similarly to DNS, mDNS environments are exposed to security attacks. Recent efforts to improve DNS security, such as DNSSEC [64] and DNS over TLS [65], are in general too complex for self-configuring networked environments.

From the analysis of the potential security threats of mDNS, we identified and classified the attacks as follows:

- *Denial of Service attacks*: attackers flood mDNS-enabled nodes with messages that exploit specific characteristics of the protocol. These messages could make nodes unresponsive or unavailable by invalidating cache entries or blocking the probing process;
- *Poisoning attacks*: attackers spoof mDNS response messages and advertise fake services frequently exploited for further attacks towards unaware nodes;
- *Remote attacks*: attackers exploit mDNS-enabled nodes responding to queries from outside to abuse services for various purposes, e.g., Distributed Denial of Service reflection attacks, collection of sensitive information.

To understand the vulnerabilities that might be behind these attacks, we analyzed the 29 CVEs affecting products and services based on mDNS. This analysis reveals that nodes that inadvertently respond to unicast queries with source addresses outside the local link allow attackers to cause Denial of Service or obtain potentially sensitive information via UDP packets over port 5353 (e.g., CVE-2015-1892, CVE-2017-6519, CVE-2017-6520). Similarly a Denial of Service attack can be performed by sending malformed or maliciously crafted packets (e.g., CVE-2015-0650).

Moreover, the multicast nature of the communications and the lack of any encryption mechanism might lead to security and privacy issues that often remain undetected. In fact, messages frequently disclose personally identifiable information as well as sensitive information about the nodes of the network and the services being provided. For example, Könings et al. [66] show that in their Wi-Fi campus network, the majority of mDNS-enabled devices include as part of their identifiers the real names of the users. This information could be easily used for any malicious purpose. Therefore, it is necessary to increase awareness of privacy risks associated with service announcements that contain sensitive information.

5.1.1. Mitigations

As already pointed out, mDNS does not provide any built-in security feature. Therefore, since the protocol is affected by various threats, the development of effective mitigation measures is of paramount importance. The solutions could rely on simple measures often provided by operating systems or on more sophisticated measures provided by the services built on top of the mDNS protocol. More specifically, simple measures – mainly mitigating DDoS attacks – could focus on the following aspects:

- Reduction of attack surface by disabling mDNS services whenever not needed;
- Block of the traffic from/to outside the local link by disabling the mDNS UDP port 5353.

In fact, mDNS protocol is often enabled by default on most devices, but users might not be aware of this protocol running on their devices. Moreover, although mDNS has been designed for local link, sometimes services are openly accessible from the Internet.

More sophisticated measures ensure the following security requirements:

- *Authenticity*: query and response messages should be signed by the sender to allow the recipients to verify the sender's identity;
- *Confidentiality*: query and response messages should be encrypted to prevent any possible abuse of their content.

Privacy is a major challenge for mDNS environments. Some research works propose solutions to mitigate this risk. More specifically, the works of Kaiser and Waldvogel [67,68] focus on a privacy-aware mechanism that protects multicast communication by encrypting all data, including potentially sensitive information. In addition, to reduce the network traffic, the mechanism limits the usage of multicast communications by proposing the concept of trusted devices that securely exchange unicast messages.

To cope with the lack of built-in authentication mechanisms, some papers [69–71] propose specific solutions for robust authentication. In particular, Wu et al. [71] develop protocols for private mutual authentication and service discovery that could be deployed over mDNS.

5.2. SSDP

Simple Service Discovery Protocol (SSDP) is an open protocol widely used nowadays for service discovery and advertisement in residential or small business networks (UPnP Forum). The protocol – included in the Universal Plug and Play (UPnP) architecture – makes it possible to transparently plug-and-play devices without the need of any manual configuration.

Concerning security, similarly to mDNS, the SSDP protocol is very weak because it does not provide any built-in mechanism. Therefore, various security risks affect SSDP-enabled devices. These risks generally exploit service discovery features and its multicast nature. A major threat affecting SSDP nodes is represented by *amplification/reflection Distributed Denial of Service attacks* aimed at making devices unresponsive and services unavailable. These attacks exploit the characteristics of the UDP and SSDP protocols as well as device misconfiguration. More precisely, an attacker could create an `M-SEARCH` message with the spoofed IP address of the node under attack (see Fig. 2). This message will be sent to a set of vulnerable SSDP devices that in turn will flood the node target of the attack with response messages with an high amplification potential.

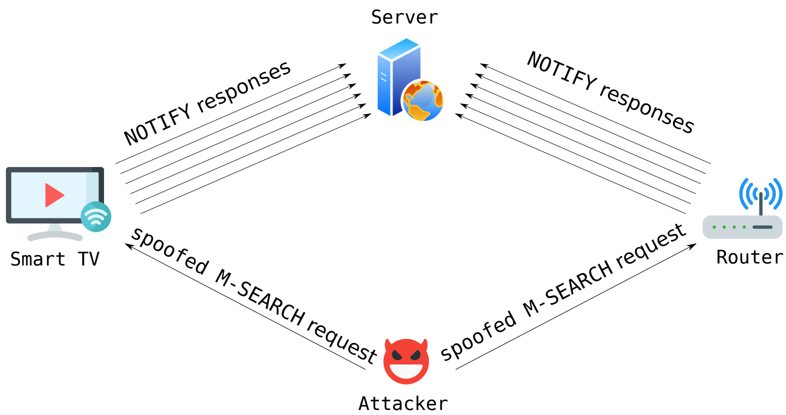

Figure 2. Example of amplification/reflection DDoS attack toward a server.

A more sophisticated variant of amplification/reflection attacks takes advantage of the abnormal behavior of devices that use ephemeral random source ports for sending their response messages instead of the standard port number 1900, thus making the detection of the attack more difficult.

Another security threat affecting SSDP-enabled nodes is represented by *passive attacks* performed by *eavesdropping* the multicast messages exchanged as plaintext over the network. This threat might grant the access to sensitive information without any alert, thus leading to serious consequences for privacy and confidentiality.

SSDP-enabled nodes are also exposed to the following security issues:

- *Poisoning attacks* where attackers advertise fake services using `NOTIFY` request messages. These services are frequently exploited for further attacks towards unaware nodes;
- *Device reconfiguration* where attackers exploit vulnerabilities of misconfigured devices to gain access to internal network resources or use the devices to conduct further malicious activities.

The analysis of the CVEs has shown that numerous vulnerabilities affect products and services based on SSDP. More precisely, 81 vulnerabilities have been detected in the past six years. A common vulnerability is represented by buffer overflow that allows attackers to remotely execute arbitrary code or crash an SSDP node (e.g., CVE-2019-14323, CVE-2019-14363). Other relevant security issues are related to the rules and functions associated with device configuration. In particular, it has been shown that weak authentication and authorization mechanisms allow remote attackers to change device configuration or reboot/shutdown devices (e.g., CVE-2014-5406, CVE-2015-4051).

In the literature, SSDP security challenges have been explored by Liu et al. [72] who analyze the Belkin WeMo home automation ecosystem with the objective of discovering its vulnerabilities. In particular, the paper demonstrates that it is possible to remote control these devices by leveraging the sensitive information being exchanged. Similarly, Lyu et al. [73] quantify the DDoS attack capability of consumer IoT devices and show that devices even behind gateways can be exposed to this type of attacks.

5.2.1. Mitigations

As already pointed out, the lack of built-in security services exposes SSDP-enabled nodes to threats and attacks. Hence, proper countermeasures have to be sought. In particular, it is important to take account of the peculiarities of SSDP. In fact, this protocol is typically deployed on a local network and relies on UDP transport protocol on port 1900. Therefore, as a mitigation measure towards conventional DDoS attacks, it might be necessary to block this type of incoming traffic. In fact, it is known that open SSDP is already a vulnerability. Of course these measures are not effective to mitigate DDoS attacks that leverage on SSDP nodes using random source ports.

At the level of individual nodes, SSDP services should be disabled whenever not needed, since they are often enabled by default on most devices. In addition, unicast `M-SEARCH` request messages should be treated carefully and possibly blocked because of the abnormal usage of this type of messages.

It is also worth mentioning that encryption mechanisms able to ensure *authenticity* and *confidentiality* of the exchanges and avoid possible abuse of their content, must be implemented at the level of the services built on top of the SSDP protocol, rather than at the level of the protocol itself.

Various solutions for securing smart home IoT appliances based on SSDP have been proposed in the literature (see, e.g., [74,75]) In particular, Notra et al. [74] highlight that security and privacy of these appliances can be easily compromised and propose a solution based on access restrictions at the network level. In [75] it has been shown that a flow-based monitoring solution is effective for detecting security threats.

6. Discussion

Our analysis has highlighted that ensuring security of IoT products and services that leverage application layer protocols is not straightforward. In fact, the IoT threat landscape is extremely diverse and complex. The open nature of application layer protocols makes them exposed to a wide range of malicious attacks that exploit their peculiarities as well the characteristics of networked environments. Moreover, despite their potential vulnerabilities, IoT devices and services are often being developed and deployed without specific security consideration.

Since IoT devices are being an integral part of our everyday life, it is compelling to protect these devices by properly identifying potential security risks and by devising adequate mitigation measures. As reported in Table 2, application layer protocols provide some common built-in security services although the constrained capabilities of these devices make their deployment quite challenging or even impossible. In addition, security services are often optional and have to be explicitly enabled and configured by developers.

The major security risks affecting the protocols analyzed in this paper are summarized in Table 3. In general, as main findings of this investigation, we discovered that frequent sources of risks refer to the lack of appropriate security services or to their incorrect configuration. In particular, mDNS and

Protocol	Authentication service	Authorization service	Encryption service
MQTT	△	△	△
CoAP	□	△	△
AMQP	△	△	△
DDS	△	△	△
XMPP	△	△	△
mDNS	□	□	□
SSDP	□	□	□

Table 3. Summary of the major security risks affecting the application layer protocols analyzed in this paper. The squares refer to the lack of the security service, while the triangles to its incorrect configuration.

SSDP are very weak because they do not offer any built-in security service. On the contrary, although messaging protcols offer various security services, they suffer from the incorrect configuration of these services. In addition, the lack of built-in authentication/authorization mechanisms or the use of weak mechanisms make IoT devices vulnerable to unauthorized accesses. Similarly, the incorrect configuration of TLS or the use of weak cipher suites make devices vulnerable to the disclosure of sensitive data.

These findings have been confirmed by the analysis of the CVEs of products and services based on the protocols considered in this paper. More precisely, many vulnerabilities refer to improper message validation/parsing (e.g., buffer overflow, option/exception validation) and to weak authentication/authorization mechanisms (e.g., username/hostname validation, certificate verification). Our investigation has also shown that vulnerabilities are appearing with an increased frequency, although with differences from protocol to protocol (see Figure 3).

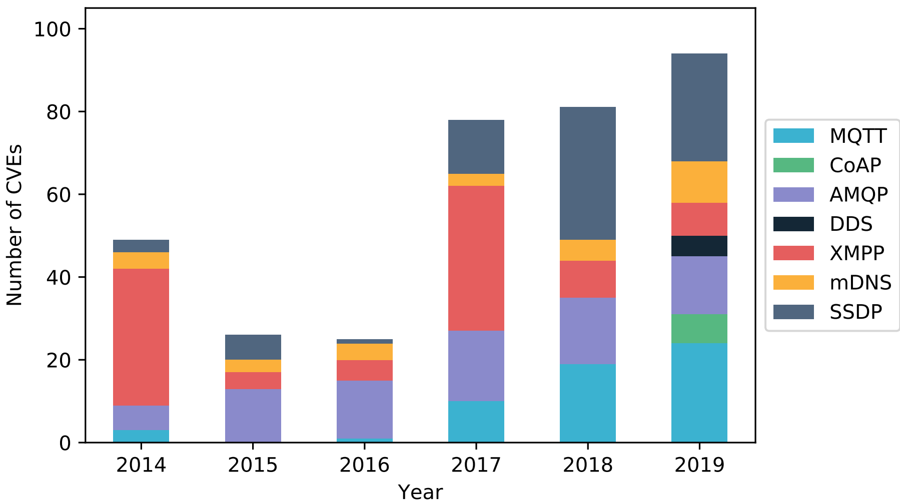

Figure 3. Breakdown of the CVEs per year and protocol.

Moreover, these CVEs are characterized by different severity ratings (see Table 4). The Common Vulnerability Scoring System (CVSS), at the basis of these ratings, provides a numerical score and the corresponding qualitative representation, i.e., Low, Medium and High, reflecting the CVE severity. For each protocol, Table 4 reports the breakdown of the number of CVEs according to their severity as well as the overall CVSS score. Note that our analysis is based on CVSS version 2 since the scores for the latest CVSS version 3.1 were unavailable for some of the analyzed CVEs.

It is also important to outline that security risks and vulnerabilities expose IoT devices to a wide range to threats and attacks (see Table 5) that could have very serious effects. We notice that

Protocol	Severity			CVSS2 Score
	Low	Medium	High	
MQTT	3	42	12	5.6
CoAP	0	5	2	6.6
AMQP	11	50	17	5.2
DDS	0	5	0	5.0
XMPP	5	70	19	5.6
mDNS	0	16	13	6.4
SSDP	5	49	27	5.9

Table 4. Per protocol breakdown of the number of CVEs according to their severity and overall CVSS2 score.

Protocol	Eavesdropping attacks	IP spoofing attacks	DoS/DDoS attacks	MiTM attacks	Poisoning attacks
MQTT			•	•	
CoAP		•	•	•	
AMQP			•		
DDS			•		
XMPP			•	•	
mDNS	•	•	•	•	•
SSDP	•	•	•	•	•

Table 5. Summary of the major attacks affecting the application layer protocols analyzed in this paper.

constrained devices are especially vulnerable to DoS and DDoS attacks mainly because of their limited capabilities or of an incorrect configuration. Attackers can easily cause temporary or permanent failures of a service by flooding a device with connection attempts that drain its battery or by performing amplification/reflection attacks that simply exploit device vulnerabilities. It is also important to outline that the UDP transport protocol is the main attack vector for application layer protocols, such as CoAP, mDSN and SSDP.

Good practices and measures aimed at mitigating the security risks and reducing the attack surface have been proposed by several papers.

Table 6 presents the breakdown of the papers appeared in the literature as a function of the protocol and security service. We notice that most works focused on MQTT and CoAP protocols

Protocol	Authentication service	Authorization service	Encryption service
MQTT	[25,38,39]	[30,38–40]	[22–33]
CoAP	[50,53,54]	[44,50,53]	[43,45,46,48,49,51,52,54]
mDNS	[69–71]		[66–68]
SSDP		[74,75]	

Table 6. Breakdown of the papers focusing on good practice and mitigation measures as a function of the protocol and of the security service.

and in particular on the development of lightweight encryption mechanisms able to cope with the constrained characteristics of IoT devices. On the contrary, despite the serious security risks affecting service discovery protocols, little research efforts have been dedicated to mitigate the potential attacks. We also outline that our search did not produce any relevant paper proposing mitigation measures for the AMQP, DDS and XMPP protocols.

7. Conclusion

The increased proliferation and ubiquity of IoT devices have also increased security issues. Many devices are treated by their designers, manufacturers and owners as dumb objects that in the hands of hackers can be easily exploited to create all sort of damages.

In this paper we analyzed the security of a set of application layer protocols widely accepted in the IoT ecosystem. In particular, we focused on messaging and service discovery protocols and discussed their characteristics as well as their potential vulnerabilities and security risks. Our investigation has shown that vulnerabilities make IoT devices an ideal target of attacks with serious consequences for the services being deployed. Good practices and measures have been developed to mitigate threats and attacks. These measures mainly focused on lightweight solutions that cope with the capabilities of constrained devices.

To properly secure IoT devices, many research and practical challenges are still to be investigated. In particular, research efforts should be directed towards security and privacy of service discovery protocols. Moreover, solutions for end-to-end security of complex systems consisting of many interconnected devices have to be investigated. Finally, it is compelling to increase user awareness towards potential security risks associated with the ownership and use of IoT devices.

References

1. Gubbi, J.; Buyya, R.; Marusic, S.; Palaniswami, M. Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Generation Computer Systems* **2013**, *29*, 1645–1660.
2. Hanes, D.; Salquiero, G.; Grossetete, P.; Barton, R.; Henry, J. *IoT Fundamentals: Networking technologies, Protocols, and Use Cases for the Internet of Things*; Cisco Press, 2017.
3. Miller, M. *The Internet of Things: How smart TVs, Smart Cars, Smart Homes, and Smart Cities are changing the world*; Pearson Education, 2015.
4. Loi, F.; Sivanathan, A.; Gharakheili, H.H.; Radford, A.; Sivaraman, V. Systematically Evaluating Security and Privacy for Consumer IoT Devices. Proc. of the Workshop on Internet of Things Security and Privacy (IoTS&P). ACM, 2017.
5. Alrawi, O.; Lever, C.; Antonakakis, M.; Monrose, F. SoK: Security Evaluation of Home-Based IoT Deployments. Proc. of the IEEE Symposium on Security and Privacy (S&P), 2019, pp. 1362–1380.
6. Aly, M.; Khomh, F.; Haoues, M.; Quintero, A.; Yacout, S. Enforcing security in Internet of Things frameworks: A Systematic Literature Review. *Internet of Things* **2019**, *6*, 100050.
7. Ammar, M.; Russello, G.; Crispo, B. Internet of Things: A survey on the security of IoT frameworks. *Journal of Information Security and Applications* **2018**, *38*, 8–27.
8. Jing, Q.; Vasilakos, A.V.; Wan, J.; Lu, J.; Qiu, D. Security of the Internet of Things: perspectives and challenges. *Wireless Networks* **2014**, *20*, 2481–2501.
9. Macedo, E.L.C.; de Oliveira, E.A.R.; Silva, F.H.; Mello, R.R.; França, F.M.G.; Delicato, F.C.; de Rezende, J.F.; de Moraes, L.F.M. On the security aspects of Internet of Things: A systematic literature review. *Journal of Communications and Networks* **2019**, *21*, 444–457.
10. Mosenia, A.; Jha, N.K. A Comprehensive Study of Security of Internet-of-Things. *IEEE Transactions on Emerging Topics in Computing* **2017**, *5*, 586–602.
11. Neshenko, N.; Bou-Harb, E.; Crichigno, J.; Kaddoum, G.; Ghani, N. Demystifying IoT Security: An Exhaustive Survey on IoT Vulnerabilities and a First Empirical Look on Internet-Scale IoT Exploitations. *IEEE Communications Surveys & Tutorials* **2019**, *21*, 2702–2733.
12. Nguyen, K.T.; Laurent, M.; Oualha, N. Survey on secure communication protocols for the Internet of Things. *Ad Hoc Networks* **2015**, *32*, 17–31.
13. Noor, M.b.M.; Hassan, W.H. Current research on Internet of Things (IoT) security: A survey. *Computer Networks* **2019**, *148*, 283 – 294.
14. Radoglou Grammatikis, P.I.; Sarigiannidis, P.G.; Moscholios, I.D. Securing the Internet of Things: Challenges, threats and solutions. *Internet of Things* **2019**, *5*, 41–70.
15. Riahi Sfar, A.; Natalizio, E.; Challal, Y.; Chtourou, Z. A roadmap for security challenges in the Internet of Things. *Digital Communications and Networks* **2018**, *4*, 118–137.

16. Zhou, W.; Jia, Y.; Peng, A.; Zhang, Y.; Liu, P. The Effect of IoT New Features on Security and Privacy: New Threats, Existing Solutions, and Challenges Yet to Be Solved. *IEEE Internet of Things Journal* **2019**, *6*, 1606–1616.
17. Cabrera, C.; Palade, A.; Clarke, S. An evaluation of service discovery protocols in the Internet of Things. Proc. of the Symposium on Applied Computing (SAC '17). ACM, 2017, pp. 469–476.
18. Dizdarević, J.; Carpio, F.; Jukan, A.; Masip, X. A Survey of Communication Protocols for Internet of Things and Related Challenges of Fog and Cloud Computing Integration. *ACM Computing Surveys* **2019**, *51*, 116:1–116:29.
19. Naik, N. Choice of effective messaging protocols for IoT systems: MQTT, CoAP, AMQP and HTTP. Proc. of the IEEE Int. Systems Engineering Symposium (ISSE), 2017.
20. Rescorla, E. The Transport Layer Security (TLS) Protocol Version 1.3. RFC 8446, RFC Editor, 2018.
21. Firdous, S.N.; Baig, Z.; Valli, C.; Ibrahim, A. Modelling and Evaluation of Malicious Attacks against the IoT MQTT Protocol. Proc. of the IEEE Int. Conf. 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), 2017, pp. 748–755.
22. Perrone, G.; Vecchio, M.; Pecori, R.; Giaffreda, R. The Day After Mirai: A Survey on MQTT Security Solutions After the Largest Cyber-attack Carried out through an Army of IoT Devices. Proc. of the 2nd Int. Conf. on Internet of Things, Big Data and Security (IoTBDS). SciTePress, 2017, pp. 246–253.
23. Ivanov, O.; Ruzhentsev, V.; Oliynykov, R. Comparison of Modern Network Attacks on TLS Protocol. Proc. of the Int. Scientific-Practical Conf. on Problems of Infocommunications, Science and Technology (PIC S&T). IEEE, 2018, pp. 565–570.
24. Sheffer, Y.; Holz, R.; Saint-Andre, P. Summarizing known attacks on Transport Layer Security (TLS) and Datagram TLS (DTLS). RFC 7457, RFC Editor, 2015.
25. Bali, R.S.; Jaafar, F.; Zavarasky, P. Lightweight Authentication for MQTT to Improve the Security of IoT Communication. Proc. of the 3rd Int. Conf. on Cryptography, Security and Privacy (ICCSP '19). ACM, 2019, pp. 6–12.
26. Bisne, L.; Parmar, M. Composite secure MQTT for Internet of Things using ABE and dynamic S-box AES. Proc. of the Innovations in Power and Advanced Computing Technologies (i-PACT). IEEE, 2017.
27. Calabretta, M.; Pecori, R.; Veltri, L. A Token-based Protocol for Securing MQTT Communications. Proc. of the 26th Int. Conf. on Software, Telecommunications and Computer Networks (SoftCOM). IEEE, 2018.
28. Dinculeană, D. Vulnerabilities and Limitations of MQTT Protocol Used between IoT Devices. *Applied Sciences* **2019**, *9*, 848.
29. Malina, L.; Srivastava, G.; Dzurenda, P.; Hajny, J.; Fujdiak, R. A Secure Publish/Subscribe Protocol for Internet of Things. Proc. of the 14th Int. Conf. on Availability, Reliability and Security (ARES '19). ACM, 2019.
30. Niruntasukrat, A.; Issariyapat, C.; Pongpaibool, P.; Meesublak, K.; Aiumsupucgul, P.; Panya, A. Authorization mechanism for MQTT-based Internet of Things. Proc. of the IEEE Int. Conf. on Communications Workshops (ICC), 2016, pp. 290–295.
31. Shin, S.; Kobara, K.; Chia-Chuan Chuang.; Weicheng Huang. A security framework for MQTT. Proc. of the IEEE Conf. on Communications and Network Security (CNS), 2016, pp. 432–436.
32. Singh, M.; Rajan, M.A.; Shivraj, V.L.; Balamuralidhar, P. Secure MQTT for Internet of Things (IoT). Proc. of the 5th Int. Conf. on Communication Systems and Network Technologies. IEEE, 2015, pp. 746–751.
33. Yerlikaya, O.; Dalkılıç, G. Authentication and Authorization Mechanism on Message Queue Telemetry Transport Protocol. Proc. of the 3rd Int. Conf. on Computer Science and Engineering (UBMK). IEEE, 2018, pp. 145–150.
34. Aumasson, J.P.; Neves, S.; Wilcox-O'Hearn, Z.; Winnerlein, C. BLAKE2: Simpler, Smaller, Fast as MD5. In *Applied Cryptography and Network Security*; Jacobson, M.; Locasto, M.; Mohassel, P.; Safavi-Naini, R., Eds.; Springer, 2013; Vol. 7954, *Lecture Notes in Computer Science*, pp. 119–135.
35. Hankerson, D.; Menezes, A.J.; Vanstone, S. *Guide to Elliptic Curve Cryptography*; Springer, 2004.
36. Schneier, B.; Kohno, T.; Ferguson, N. *Cryptography engineering: design principles and practical applications*; Wiley, 2013.
37. D. Hardt. The OAuth 2.0 Authorization Framework. RFC 6749, RFC Editor, 2012.

38. Neisse, R.; Steri, G.; Baldini, G. Enforcement of security policy rules for the Internet of Things. Proc. of the IEEE 10th Int. Conf. on Wireless and Mobile Computing, Networking and Communications (WiMob), 2014, pp. 165–172.
39. Colombo, P.; Ferrari, E. Access Control Enforcement Within MQTT-based Internet of Things Ecosystems. Proc. of the 23nd ACM on Symposium on Access Control Models and Technologies, 2018, SACMAT '18, pp. 223–234.
40. Alghamdi, K.; Alqazzaz, A.; Liu, A.; Ming, H. IoTVerif: An Automated Tool to Verify SSL/TLS Certificate Validation in Android MQTT Client Applications. Proc. of the 8th ACM Conf. on Data and Application Security and Privacy (CODASPY), 2018, pp. 95–102.
41. Z. Shelby.; K. Hartke.; C. Bormann. The Constrained Application Protocol (CoAP). RFC 7252, RFC Editor, 2014.
42. E. Rescorla, N.M. Datagram Transport Layer Security Version 1.2. RFC 6347, RFC Editor, 2012.
43. Albalas, F.; Al-Soud, M.; Almomani, O.; Almomani, A. Security-aware CoAP Application Layer Protocol for the Internet of Things using Elliptic-Curve Cryptography. *Int. Arab Journal of Information Technology* **2018**, *15*, 550–558.
44. Bergmann, O.; Gerdes, S.; Schäfer, S.; Junge, F.; Bormann, C. Secure bootstrapping of nodes in a CoAP network. Proc. of the IEEE Wireless Communications and Networking Conf. Workshops (WCNCW), 2012, pp. 220–225.
45. Capossele, A.; Cervo, V.; Cicco, G.D.; Petrioli, C. Security as a CoAP resource: An optimized DTLS implementation for the IoT. Proc. of the IEEE Int. Conf. on Communications (ICC), 2015, pp. 549–554.
46. Harish, M.; Karthick, R.; Mohan Rajan, R.; Vetriselvi, V. Securing CoAP Through Payload Encryption: Using Elliptic Curve Cryptography. Proc. of the Int. Conf. on Communications and Cyber Physical Engineering (ICCCE 2018); Kumar, A.; Mozar, S., Eds. Springer, 2019, Vol. 500, *Lecture Notes in Electrical Engineering*, pp. 497–511.
47. Iglesias-Urkia, M.; Orive, A.; Urbieta, A.; Casado-Mansilla, D. Analysis of CoAP implementations for industrial Internet of Things: a survey. *Journal of Ambient Intelligence and Humanized Computing* **2019**, *10*, 2505–2518.
48. Kwon, H.; Park, J.; Kang, N. Challenges in Deploying CoAP Over DTLS in Resource Constrained Environments. In *Information Security Applications*; Kim, H.W.; Choi, D., Eds.; Springer, 2016; Vol. 9503, *Lecture Notes in Computer Science*, pp. 269–280.
49. Park, Y.j.; Lee, K.h. Constructing a secure hacking-resistant IoT U-healthcare environment. *Journal of Computer Virology and Hacking Techniques* **2018**, *14*, 99–106.
50. Puñal Pereira, P.; Eliasson, J.; Delsing, J. An Authentication and Access Control Framework for CoAP-based Internet of Things. Proc. of the 40th Annual Conf. of the IEEE Industrial Electronics Society. IEEE, 2014, pp. 5293–5299.
51. Randhawa, R.H.; Hameed, A.; Mian, A.N. Energy efficient cross-layer approach for object security of CoAP for IoT devices. *Ad Hoc Networks* **2019**, *92*.
52. Raza, S.; Shafagh, H.; Hewage, K.; Hummen, R.; Voigt, T. Lithe: Lightweight Secure CoAP for the Internet of Things. *IEEE Sensors Journal* **2013**, *13*, 3711–3720.
53. Seitz, L.; G. Selander.; M. Mani.; S. Kumar. Use Cases for Authentication and Authorizationin Constrained Environments. RFC 7744, RFC Editor, 2016.
54. Ukil, A.; Bandyopadhyay, S.; Bhattacharyya, A.; Pal, A.; Bose, T. Auth-Lite: Lightweight M2M Authentication reinforcing DTLS for CoAP. Proc. of the IEEE Int. Conf. on Pervasive Computing and Communication Workshops, 2014, pp. 215–219.
55. Alghamdi, T.A.; Lasebae, A.; Aiash, M. Security Analysis of the Constrained Application Protocol in the Internet of Things. Proc. of the 2nd IEEE Int. Conf. on Future Generation Communication Technologies (FGCT), 2013, pp. 163–168.
56. A. Melnikov, K.Z. Simple Authentication and Security Layer (SASL). RFC 4422, RFC Editor, 2006.
57. White, R.; Caiazza, G.; Jiang, C.; Ou, X.; Yang, Z.; Cortesi, A.; Christensen, H. Network Reconnaissance and Vulnerability Excavation of Secure DDS Systems. Proc. of the IEEE European Symposium on Security and Privacy Workshops (EuroS&PW), 2019, pp. 57–66.

58. Michaud, M.; Dean, T.; Leblanc, S. Attacking OMG Data Distribution Service (DDS) Based Real-Time Mission Critical Distributed Systems. Proc. of the 3th Int. Conf. on Malicious and Unwanted Software (MALWARE). IEEE, 2018, pp. 68–77.
59. Saint-Andre, P. Extensible Messaging and Presence Protocol (XMPP): Core. RFC 6120, RFC Editor, 2011.
60. Saint-Andre, P. Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence. RFC 6121, RFC Editor, 2011.
61. Ferreira, R.; Aguiar, R. Breaching location privacy in XMPP based messaging. Proc. of the IEEE Global Communications Conf. (GLOBECOM), 2012, pp. 917–922.
62. S. Cheshire, M.K. DNS-Based Service Discovery. RFC 6763, RFC Editor, 2013.
63. S. Cheshire, M.K. Multicast DNS. RFC 6762, RFC Editor, 2013.
64. Arends, R.; Austein, R.; Larson, M.; Massey, D.; Rose, S. DNS Security Introduction and Requirements. RFC 4033, RFC Editor, 2005.
65. Hu, Z.; Zhu, L.; Heidemann, J.; Mankin, A.; Wessels, D.; Hoffman, P. Specification for DNS over Transport Layer Security (TLS). RFC 7858, RFC Editor, 2016.
66. Könings, B.; Bachmaier, C.; Schaub, F.; Weber, M. Device Names in the Wild: Investigating Privacy Risks of Zero Configuration Networking. Proc. of the IEEE 14th Int. Conf. on Mobile Data Management, 2013, Vol. 2, pp. 51–56.
67. Kaiser, D.; Waldvogel, M. Adding Privacy to Multicast DNS Service Discovery. Proc. of the IEEE 13th Int. Conf. on Trust, Security and Privacy in Computing and Communications, 2014, pp. 809–816.
68. Kaiser, D.; Waldvogel, M. Efficient Privacy Preserving Multicast DNS Service Discovery. Proc. of the IEEE Int. Conf. on High Performance Computing and Communications, Proc. of the IEEE 6th Int. Symposium on Cyberspace Safety and Security, Proc. of the IEEE 11th Int. Conf. on Embedded Software and Systems (HPCC,CSS,ICESS), 2014, pp. 1229–1236.
69. Bai, X.; Xing, L.; Zhang, N.; Wang, X.; Liao, X.; Li, T.; Hu, S. Staying Secure and Unprepared: Understanding and Mitigating the Security Risks of Apple ZeroConf. Proc. of the IEEE Symposium on Security and Privacy (S&P), 2016, pp. 655–674.
70. Bai, X.; Xing, L.; Zhang, N.; Wang, X.; Liao, X.; Li, T.; Hu, S. Apple ZeroConf Holes: How Hackers Can Steal iPhone Photos. *IEEE Security & Privacy* **2017**, *15*, 42–49.
71. Wu, D.J.; Taly, A.; Shankar, A.; Boneh, D. Privacy, Discovery, and Authentication for the Internet of Things. In *Computer Security – ESORICS*; Askoxylakis, I.; Ioannidis, S.; Katsikas, S.; Meadows, C., Eds.; Springer, 2016; Vol. 9879, *Lecture Notes in Computer Science*, pp. 301–319.
72. Liu, H.; Spink, T.; Patras, P. Uncovering Security Vulnerabilities in the Belkin WeMo Home Automation Ecosystem. Proc. of the IEEE Int. Conf. on Pervasive Computing and Communications Workshops (PerCom Workshops), 2019, pp. 894–899.
73. Lyu, M.; Sherratt, D.; Sivanathan, A.; Gharakheili, H.H.; Radford, A.; Sivaraman, V. Quantifying the reflective DDoS attack capability of household IoT devices. Proc. of the 10th ACM Conf. on Security and Privacy in Wireless and Mobile Networks (WiSec '17). ACM, 2017, pp. 46–51.
74. Notra, S.; Siddiqi, M.; Habibi Gharakheili, H.; Sivaraman, V.; Boreli, R. An experimental study of security and privacy risks with emerging household appliances. Proc. of the IEEE Conf. on Communications and Network Security, 2014, pp. 79–84.
75. Sivanathan, A.; Sherratt, D.; Gharakheili, H.H.; Sivaraman, V.; Vishwanath, A. Low-cost flow-based security solutions for smart-home IoT devices. Proc. of the IEEE Int. Conf. on Advanced Networks and Telecommunications Systems (ANTS), 2016.