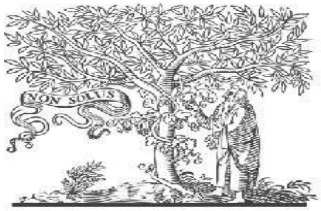
International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

COPY RIGHT



ELSEVIER SSRN

IJIEMR Transactions, online available on 26th Dec 2021. Link :http://www.ijiemr.org/downloads.php?vol=Volume-10&issue=Issue 12

10.48047/IJIEMR/V10/ISSUE 12/23

TITLE: IOT Application Layer Protocols & Its Security Vulnerabilities
Volume 10, ISSUE 12, Pages: 148-169

Paper Authors **Dr. Vadhri Suryanarayana, Dr. Satyabrata Dash, Y. Nagendra Kumar, Sujata Chakarvarty, Dr. T.M Usha**

USE THIS BARCODE TO ACCESS YOUR ONLINE PAPER

To Secure Your Paper As Per UGC Guidelines We Are Providing A Electronic Bar Code

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

IOT Application Layer Protocols & Its Security Vulnerabilities

Dr. Vadhri Suryanarayana, Dr. Satyabrata Dash, Y. Nagendra Kumar, Sujata Chakarvarty, Dr. T.M Usha

Professor, Department of Computer Science and Engineering, Ramachandra College of Engineering, Eluru, AP, INDIA vs@rcee.ac.in

Associate Professor ,Department of Computer Science and Engineering, Ramachandra College of Engineering, Eluru, AP,INDIA Satyabrata.cse@rcee.ac.in

Assistant Professor, Department of Computer Science and Engineering, Ramachandra College of Engineering, Eluru, AP, INDIA nagendrayakkala@rcee.ac.in

Professor Department of Computer Science and Engineering, Centurion University of Technology & Management, Odisha, INDIA sujata.chakravarty@cutm.ac.in

Department of Computer Science and Engineering, Ramachandra College of Engineering, Eluru, AP, INDIA Ushawin2020@gmail.com

Abstract

The network security challenges to the Internet of Things (IoT) vulnerabilities issues provide a platform for protecting and securing the communication networks connecting IoT devices through the Internet. Which is more challenging than traditional network security because there is a wider range of communication protocols, standards, and device capabilities. Internet of Thing is generally made up of three-layer architecture, namely Perception, Network and Application layers. A lot of security constraints should be enabled at each layer for proper and efficient working of these applications. This chapter mainly focuses on the common IoT application layer protocols ie MQTT, AMQP ,XMPP and CoAP and DDS. It also includes the explanations on the security challenges in application layer protocols. Security is still one of the most critical challenges in IoT platforms and, hence, a lot of standards, drafts and research work has been proposed. There exist some security features within IoT protocols, however, that is not enough to fully secure the IoT systems so a proper analysis can help for the counter measure. Almost all security problems that arise are related to the state in which the protocol works due to the lack of common standards like Lack of authentication, Lack of authorization, Lack of confidentiality, Lack of integrity. So these most common security issues that need to be solved through proper protocol configurations.

Keywords: Machine-to-Machine Communication, IOT Application layer Protocol, MQTT, AMQP ,XMPP and CoAP and DDS , Security Vulnerabilities

1. Introduction

Internet of Things (IoT) is a global infrastructure for the information society, enabling advanced services by interconnecting (physical and virtual) things based on existing and evolving interoperable information and communication technologies. All the present-day devices around the globe with internet connectivity have the limitless future possibilities towards utility.[1] And utility needs communication between the devices for accuracy and efficiency, whereas communication needs implementation of protocols to maximize the security and minimize the data loss. It's a challenge to implement various protocols to different platforms in Internet of Things (IoT). In our research the common IoT application layer protocols like MQTT, AMQP, CoAP, and DDS are taken into discussion. Each of them is derived from traditional Internet protocols and further adapted to the IoT specifications to make them suitable for application involving constrained devices[1].

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

The things in IoT is any kind of device integrated with any kind of sensors which has a built in ability to collect and convey data over a network whereas have an additional capacity of partially process the data collected. Since IoT allows devices to be controlled remotely across the internet, thus it created opportunities to directly connect & integrate the physical world to the computer-based systems using sensors and internet. The interconnection of these multiple embedded devices will be resulting in automation in nearly all fields and also enabling advanced applications. IoT has helped a lot in improving technologies and making them better. It also encourages automating devices and making surrounding smarter. It improves customer experience by automating the action. IoT provides real-time information, which helps us in improvisation of knowledgebase, automation, prediction system and many more**.** "Things" in the IoT sense, is the mixture of hardware, data, software and services. Internet is used for connectivity and machine to machine communication. Sensors and actuators are enabled with computing devices and internet for automation of services to make the devices and surrounding smart.[2][4]

Internet: Internet connectivity for communication
+
Things: Embedded Devices with various sensors and actuators

2. The IoT reference model

The International Telecommunication Union-Telecommunication has defined a reference model for IoT. This model is divided into the four layers: application layer, service support and application support layer, network layer and device layer. Each one of these layers also includes management and security capabilities. As shown in the figure 1 these capabilities have both generic and specific capabilities that can cut across multiple layers.

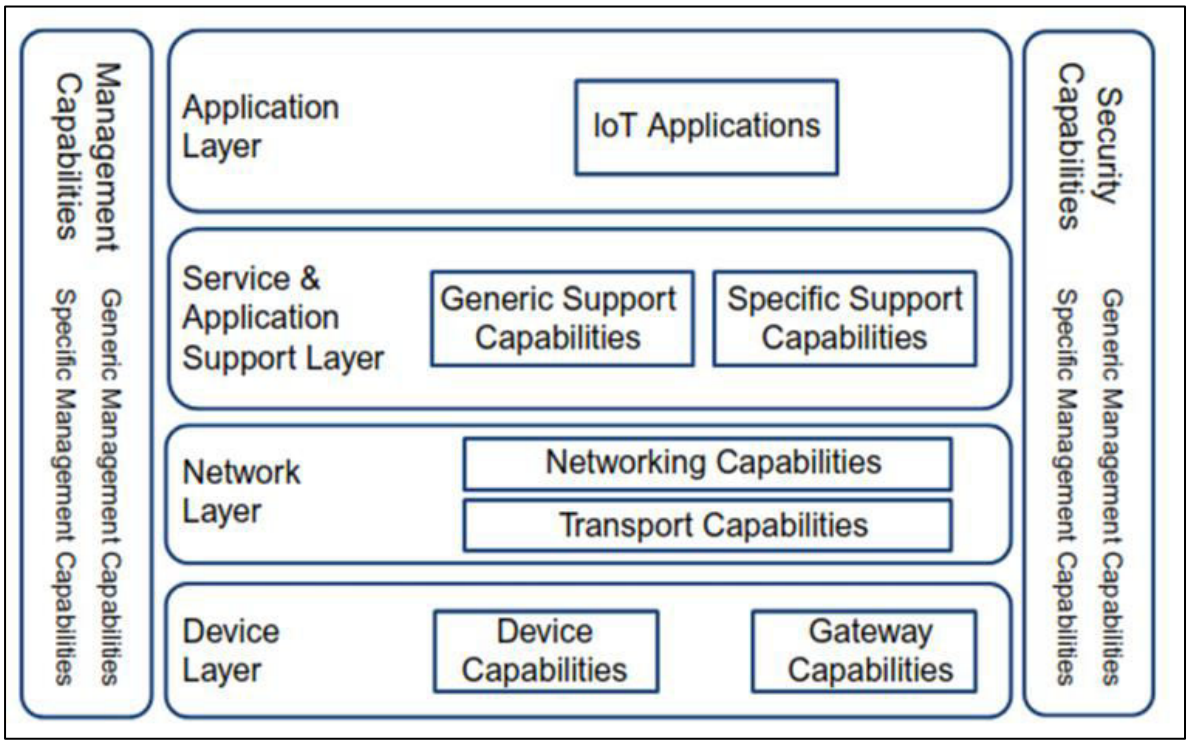


Figure 1 Reference model for IoT

The application layer contains IoT applications which require certain support capabilities from the underlying layer to function. The service and application support layer consists of generic support capabilities which can be used by IoT applications, examples of such capabilities could be data processing or storage. The specific support capabilities are those other than the generic capabilities which are required to create support for diversified applications [4][9]. The network layer is divided into networking and transport capabilities. The networking capabilities provide relevant control functions for network connectivity, while the transport capabilities focus on the transport of IoT service and application specific data. At the bottom of the model, there is the device layer in which the device capabilities include direct and indirect interaction with the communication network. Unlike direct interaction, indirect interaction requires a gateway to be able to send and receive information via the network. Two other capabilities are ad hoc networking and sleeping and waking up which enable devices to connect in an ad hoc manner and saving energy (respectively) [9].

The device layer also includes gateway capabilities to support devices connected via different types of wired and wireless technologies by supporting multiple interfaces. In some situations, protocol conversion is needed to support communication between devices using different protocols at the device and network layer [9]. Generic management capabilities include device management (such as remote device activation, de-activation, diagnostics, and firmware or software updates) and local network topology, traffic, and congestion management [9]. The generic security capabilities are independent of the application and include authorization and authentication at the application, network, and device layer. Moreover, all of the layers have their own individual capabilities. These include: At the application layer application data confidentiality and integrity protection, privacy protection, security audit and anti-virus; At the network layer signalling data confidentiality and integrity protection; and At the device layer device integrity validation, access control, data confidentiality, and integrity protection. Both the specific management and security capabilities are closely coupled with application specific requirements, for example mobile payment [9].

3. IOT Protocol Stack:

One of the biggest challenges faced by businesses, architects, and developers while dealing with IoT projects is selecting the technology stack and tools – this stems from the fact that standardization in the IoT protocols is virtually non-existent. The root of the problem is the constrained environment of IoT characterized with low memory availability, low power, low bandwidth requirement, and high packet loss – combined, these do not allow TCP/IP stack and web technologies to be used easily for IoT. However, to solve this challenge, there are hundreds of proprietary protocols in IoT, M2M (Machine to Machine) and Home Automation space such as ZigBee and Z-Wave. Though these protocols are supported by an alliance of product vendors, they are not standardized like TCP, IP, HTTP or SMTP. Although the scenario is still a bit cloudy, a set of open, standardized set of protocols have started to emerge. Most bodies such as IEEE, IETF or W3C have standardized protocols such as 6LoWPAN or CoAP. In the long run, these protocols would emerge successful like the open standardized web standards used by the web today.

IEEE 802.15.4 is a standard for wireless communication that defines the Physical layer (PHY) and Media Access Control (MAC) layers. It is standardized by the IEEE (Institute for Electrical and Electronics Engineers) similar to IEEE 802.3 for Ethernet, IEEE 802.11 is for wireless LANs (WLANs) or Wi-Fi.

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

802.15 group of standards specifies a variety of wireless personal area networks (WPANs) for different applications (For instance, 802.15.1 is Bluetooth). IEEE 802.15.4 focuses on communication between devices in constrained environment with low resources (memory, power and bandwidth).6LoWPAN is the secret sauce that allows larger IPv6 packets to flow over 802.15.4 links that support much smaller packet sizes. 6LoWPAN is the acronym of IPv6 over Low Power Wireless Personal Area Networks. So 6LoWPAN as the name implies is an adaptation layer that allows transport of IPv6 packets over 802.15.4 links. Without 6LoWPAN IPv6, internet protocols would not work in these Low Power Wireless Personal Area Networks that uses IEEE 802.15.4. 6LoWPAN is an open standard defined under RFC 6282 by the Internet Engineering Task Force (IETF), the body that defines many of the open standards used on the internet such as UDP, TCP and HTTP to name a few.

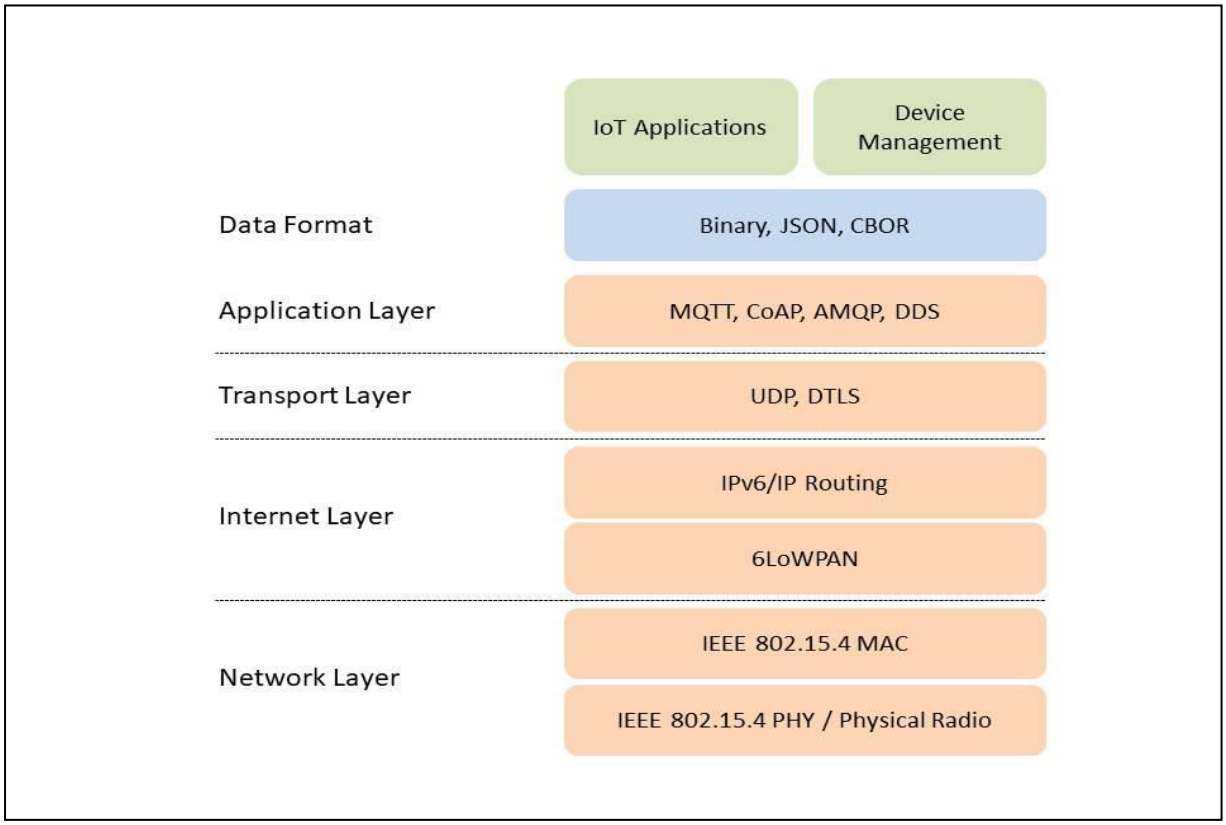


Figure 2: Protocol stack

While TCP is used predominantly in Internet as Transport Layer Protocol (except gaming or video streaming where User Datagram Protocol or UDP is used), most IoT scenarios are well suited for UDP. UDP is a much lighter protocol compared to TCP. UDP is a connection protocol and does not come with resiliency features of TCP, such as guaranteed packet delivery. On the other hand, UDP is much faster than TCP, the header size is much smaller than TCP – making it suitable for the constrained environment of devices and sensors. Higher level Application Layer IoT protocols like CoAP use UDP rather than TCP. [27] DTLS or Datagram Transport Layer Security is a TLS/SSL counterpart that runs on UDP. The way TLS/SSL takes care of security for TCP communication, DTLS provides the same security features on UDP or Datagrams.

3.1. Application-Layer Protocols:

3.1.1 COAP:

CoAP (Constrained Application Protocol) is a specialized Web Transfer Protocol for constrained nodes and constrained networks on the IoT. CoAP is an Application Layer protocol in the TCP/IP model (Web uses HTTP as an Application Layer protocol). The term "Constrained" is used because it is designed specifically from the ground up to work well in constrained environments. The devices, sensors and actuators in IoT operate in a constrained environment with low memory, low power, low bandwidth, and high rate of packet failure. HTTP was not designed to work in this sort of environment, so HTTP, which is relatively heavyweight with large header size and text encoding struggles to work in IoT constrained environment. Mphasis 4 This is where CoAP comes to play. CoAP has been standardized by IEFT (The Internet Engineering Task Force) Constrained RESTful Environments (CoRE) Working Group. Think of CoAP as web protocols for devices. CoAP can be transparently mapped to HTTP. [16-22]

The Similarities between CoAP and HTTP are

1. CoAP follows the same request-response pattern used by HTTP that we all are very familiar with. The CoAP client (a smart phone, for example) sends a request to the CoAP server (device/things) and the server then sends a response back.
2. Like the web, devices are addressed using IP address and port number. Access to services exposed by the device is via RESTful URLs.
3. CoAP uses familiar HTTP features like Methods (Get, Post, Put, and Delete), Status Codes, URLs and content type.
4. CoAP supports discovery so that IoT devices/things could be discovered.
5. CoAP provides simple proxy and caching capabilities.

The CoAP has a few differences as follows

1. CoAP runs on UDP as compares to HTTP, which typically uses TCP. UDP is lighter than TCP and has less overhead.
2. CoAP supports only Get, Put, Post and Delete methods. CoAP uses small and reduced set of headers (header size is limited to 4 bytes), and HTTP status codes to be lightweight.
3. CoAP supports confirmable and non-confirmable message types In the example given below, to get the temperature from the thermostat (which acts like a server), the client, which is a smart phone sends a GET request. The URL uses RESTful architecture - clearly indicating the device name, sensor data it is looking for, etc. The thermostat or the server responds back with the current temperature.

3.1.2 MQTT:

MQTT stands for Message Queue Telemetry Transport. MQTT is a publish-subscribe based "light weight" messaging protocol for IoT and M2M (Machine-to-Machine communication). To put it simply, MQTT is the AMQP or JMS for the constrained environment of IoT. Andy Stanford Clark and Arlen Nipper invented MQTT back in 1999, when their use case was to create a protocol for minimal battery loss and minimal bandwidth connecting oil pipelines over a satellite connection. MQTT uses a broker-based pub-sub architecture in the constrained IoT environment similar to other messaging products that exist in the Web and Client Server world.[37]

3.1.3 AMQP:

The Advanced Message Queuing Protocol (AMQP) is an open standard for passing business messages between applications. It connects systems, feeds business processes with the information they need and reliably transmits onward the instructions that achieve their goals. AMQP enables encrypted and interoperable messaging between organizations and applications. The protocol is used in client/server messaging and in IoT device management. AMPQ was conceived by John O'Hara of J.P. Morgan Chase in 2003 and started as a cooperative effort starting with the iMatix Corporation. AMQP arose from the financial industry. It can utilize different transport protocols but it assumes an underlying reliable transport protocol such as TCP [12]. AMQP provides asynchronous publish/subscribe communication with messaging. Its main advantage is its store-and-forward feature that ensures reliability even after network disruptions [13]. Another study shows that comparing AMQP 6 with the aforementioned REST, AMQP can send a larger amount of messages per second [14]. Additionally, it has been reported that an AMQP environment with 2,000 users
spread across five continents can process 300 million messages per day [14].

3.1.4 DDS:

Data Distribution Service (DDS) is an Object Management Group (OMG) standard for real-time systems that addresses data communication between the nodes of a publish/subscribe-based messaging architecture. Released in 2004, DDS serves as middleware architecture for a publish/subscribe messaging pattern. DDS works by providing scalable, high performance and real-time interaction for publishers and subscribers. Data Distribution Service is networking middleware that simplifies complex network programming. It implements a publish–subscribe pattern for sending and receiving data, events, and commands among the nodes. Nodes that produce information (publishers) create "topics" (e.g., temperature, location, pressure) and publish "samples". DDS delivers the samples to subscribers that declare an interest in that topic.The DDS publish-subscribe model virtually eliminates complex network programming for distributed applications. The detail protocol configurations as follows.

4. Security Vulnerabilities In Application Layer Protocol

This includes the explanations on the security challenges in application layer protocols. Security is still one of the most critical challenges in IoT platforms and, hence, a lot of standards, drafts and research work has been proposed. There exist some security features within IoT protocols, however, that is not enough to fully secure the IoT systems so a proper analysis can help for the counter measure.[5][6]

4.1. Security vulnerabilities

Internet of Things (IoT) has been given a lot of emphasis since the 90s when it was first proposed as an idea of interconnecting different electronic devices through a variety of technologies. However, during the past decade IoT has rapidly been developed without appropriate consideration of the profound security goals and challenges involved. This study explores the security aims and goals of application layer protocol of IoT and then provides a new classification of different types of attacks and countermeasures on security and privacy. It then discusses future security directions and challenges that need to be addressed to improve security concerns over such networks and aid in the wider adoption of IoT by masses. Because IoT is a relatively new concept, there is a need to define its security goals. To successfully achieve this we need to understand that IoT is an implementation of

network technologies and an integration of existing network infrastructures (e.g. wireless sensor networks, RFIDs based sensor networks, Cloud Computing, the Internet etc.). Therefore, all of the security challenges and threats of each network technology are passed by default onto the IoT system that utilises these technologies. Further, there is the possibility of additional security threats that arise from the coexistence and collaboration of the different technologies and the open standards and protocols created for the IoT. The most desirable security objective of IoT is to protect the collected data, since the data collected from physical devices may also include sensitive user information. For this reason the security of any IoT system needs to be resilient to data-related attacks and provide trust and data security and privacy[17][18-20] So this will provide future directions for security based on the challenge classification presented earlier. An IoT system consists of three different layers each with vulnerabilities and security attacks. To address these attacks and to successfully protect the IoT system, this section presents a multi-layered security approach that should be structured to give an optimal layered protection at each layer in an IoT system as shown on the next page in Table 1. A detailed description of the table is explained below.

4.2 IoT Physical Layer Security

a) Secure Booting: Authentication and the integrity of the software on the device should be verified using cryptographic hash algorithms, which would provide digital signatures. However, because of the low processing power on most of the devices and their need for ultra-low power consumption most cryptographic hash algorithms cannot be implemented, apart from NH and WH cryptographic hash functions that are optimal for ultra-low power consumption devices [27], [28].

b) Device authentication: When a new device is introduced to the network, it should authenticate itself before receiving or transmitting data, to ensure it is identified correctly before authorisation and keeping malicious devices out of the system.

c) Data integrity: Error detection mechanisms should be provided at each device, to ensure no tampering of sensitive data occurs. Low power consumption mechanisms like Cyclic Redundancy Checks (CRC), Checksum, Parity Bit are preferred, but for more secure error detection method WH cryptographic hash function should be applied [19].

d) Data Confidentiality: All RFID Tags, IDs and data should be encrypted on each device before transmission of data to ensure confidentiality. However, because of the ultralow power consumption, strong cryptographic encryption functions like AES cannot be implemented. Instead Blowfish or RSA have lower power consumption and less processing power and can be successfully implemented on the physical layer devices.

e) Anonymity: In some cases hiding sensitive information like the location and identity of nodes is crucial. Although Zero-Knowledge [20] approach would be the optimal solution for anonymity, it cannot be implemented on low power devices as it is a very strong algorithm and needs a lot of processing power, hence K- anonymity [21] approach best fits the job for low power devices such as the devices used in an IoT system.

4.3 IoT Network Layer Security

a) Data privacy: Illegal access to the sensor nodes can be prevented, using authentication mechanisms and point to point encryption [22].

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

b) Routing security: Secure routing is vital to the acceptance and use of sensor networks for many applications, but the majority of used routing protocols are insecure [23]. However, security of routing can be ensured by providing multiple paths for the data routing which improves the ability of the system to detect an error and keep performing upon any known of failure in the system [23]. Also, encryption and authentication mechanisms increase the security level of routing data.

c) Data integrity: Using cryptographic hash functions, the integrity of the data received on the other end is confirmed. In case of prove of tampering of data, error correction mechanisms could be introduced to mitigate the problem.

3.4. IoT Application Layer Security

a) Data security: Authentication Encryption and Integrity mechanisms are critical at this level for insuring the privacy of the whole system and protecting against data theft; it prevents unauthorised access to the system and ensures the confidentiality of the system data.

b) Access Control Lists (ACLs): Setting up policies and permissions of who can access and control the IoT system, is a crucial part as this ensures the privacy of the data, and the well being of the system. ACLs can block or allow incoming or outgoing traffic, and give or block access to requests from different users inside or outside of the network.

c) Firewalls: This is an extra effective layer of security that will help block attacks that authentication, encryption and ACLs would failed to do so. Authentication and encryption passwords can be broken if weak passwords were selected. A firewall can filter packets as they are received, blocking unwanted packets, unfriendly login attempts, and DoS attacks before even authentication process begins.

d) Anti-virus, Anti-spyware and Anti-adware: Security software like antivirus or anti spyware is important for the reliability, security, integrity and confidentiality of the IoT system. The IoT application layer security issues is shown in table-1

Table-1 IoT Layer Security Issues

IoT Layer	Counter Attacks for the Specific Layers	Counter Attacks for All Layers
Physical Layer	1) Secure Booting for all IoT devices a) Low power Cryptographic Hash Functions 2) Device Authentication using Low Power Techniques a) Data Integrity b) CRC – Cyclic Reduntancy Check c) Checsum d) Parity Bit e) WH Cryptographic Hash Function	1) a)Risk Assessment b) Finding New Threats c) Applying Updates d) Applying Patches e) Providing Improvements f) Upgrading Systems 2)Intrusion Detection Mechanisms specific to IoT Systems 3)Securing the IoT Premises a) Physical Barriers

	3) Data Confidentiality a) Encryption Algorithms like Blowfish and RSA 4) Data Anonimity a) K- Anonimity	b) Intrusion Detection Alarms c) Monitoring Devices d) Access Control Devices e) Security Personel 4)Trust Management a) Trust relation between layers b) Trust of Security and Privacy at each layer c) Trust betweeenIoT and User
Network Layer	1) Secure Communication between the devices a) Network Authentication – challenge-response mechanisms b) Point-to-Point Encryption for the confidentiality of the transmited Data c) Cryptographic Hash Functions for the Integrity of the transmitted Data 2) Implementation of Routing Security a) Use of Multiple Paths b) Encrypting Routing Tables c) Hashing Routing Tables 3) Secure User Data on the Devices a) Data Authentication b) Data Confidentiality; Encryption Schemes of encrypting the data c) Data Integrity; Cryptographic hash functions	1) a)Risk Assessment b) Finding New Threats c) Applying Updates d) Applying Patches e) Providing Improvements f) Upgrading Systems 2)Intrusion Detection Mechanisms specific to IoT Systems 3)Securing the IoT Premises a) Physical Barriers b) Intrusion Detection Alarms c) Monitoring Devices d) Access Control Devices e) Security Personal 4)Trust Management a) Trust relation between layers b) Trust of Security and Privacy at each layer c) Trust between IoT and User
Application Layer	1) Data Security a) Authentication; biometrics, passwords, etc. b) Confidentiality; Strong Encryption Schemes (AES) c) Integrity; Cryptographic Hash Functions 2) Access Control Lists (ACLs) 3) Firewalls 4) Protective Software	1) a)Risk Assessment b) Finding New Threats c) Applying Updates d) Applying Patches e) Providing Improvements f) Upgrading Systems 2)Intrusion Detection Mechanisms specific to IoT Systems

	a) Anti-virus b) Anti-adware	3)Securing the IoT Premises a) Physical Barriers b) Intrusion Detection Alarms c) Monitoring Devices d) Access Control Devices e) Security Personel 4)Trust Management a) Trust relation between layers b) Trust of Security and Privacy at each layer c) Trust between IoT and User

To insure the continued protection of an IoT system and maintain its trustworthiness, Risk Assessment, Intrusion Detection, Physical Security and Trust Management should be mandatory at all layers in IoT.

5. Security Vulnerabilities in Application layer protocols.

5.1. Security Vulnerabilities in MQTT protocol

Message Queuing Telemetry Transport protocol has various security mechanisms, but most of the securities are not configured and processed by default, such as encryption od users data or authentication of the entity. Authentication such as using the physical address of the device (MAC), exist and are supervised by the user by registering a device's information once it tries to connect. Access authorization can be done by the broker using a mechanism called an Access Control List (ACL). The ACL, as the name implies, contains records of information such as the identifiers and passwords of the different clients that are allowed to access different objects and can also specify what functions the client can perform on these. According to Reference [26,27], confidentiality is a major requirement of a secure system and can be accomplished at the application layer by encrypting the message that needs to be published.[7] This method of encryption can either be implemented as client-to-broker or end-to-end. In a client-to-broker type of encryption, the broker decrypts the information that is being published to a topic and respectively encrypts the values that it needs to send to other clients. In an end-to-end situation, the broker cannot decrypt the information being published to topics and it forwards the cipher ext to other devices. In the latter method, the user ie the broker needs fewer computational resources and less energy as it only functions as a courier and does not require any additional modules that can encrypt/decrypt messages.[7] The additional security issues can also be processed and implemented on lower layers. According to [26], one way to reliably ensure the security of a communication channel at the transport layer is by using Transport Layer Security protocol (for TCP) or even Datagram Transport Layer Security (in the case of UDP). Additionally, encryption at the link layer can be achieved by using one of the many algorithms available, such as Advanced Encryption Standard (AES) in Counter Block Mode or AES in Counter with CBC-MAC mode, also called CCM mode. This type of security mechanism provides some

additional advantages compared to other methods, such as increased efficiency due to the hardware acceleration capabilities found on radio chips.[27]

Message Queuing Telemetry Transport protocol is a publisher/subscriber messaging protocol specifically developed for constrained devices. Message queuing telemetry transport (MQTT) security is based on the TLS/SSL to provide transport encryption. It provides a security against eavesdropping.[7] On the application layer, MQTT application provides client identifier and username/password credentials which can be used for devices authentication. The disadvantage of MQTT security is the use of TLS/SSL which is not optimized for constrained devices. In fact, using TLS/SSL with certificates and session key management for a multitude of heterogeneous devices, is surely cumbersome [24, 25]. For this reasons, a more scalable, lightweight, and robust security mechanism is required. In [22] a secure MQTT (SMQTT) is proposed to increase security features of the existing MQTT protocol and its variants based on lightweight attribute-based encryption (ABE), over elliptic curves.

The advantage of using ABE is due to its inherent design which supports broadcast encryption (one encryption message delivered to multiple intended users) that make it suitable for IoT applications; moreover, the feasibility of SMQTT approach through simulations and performance evaluation has been validated. In [24], two different types of ABEs, key-policy ABE and cipher text-policy ABE, have been evaluated on different classes of mobile devices including a laptop and a smart phone providing a comprehensive study of ABE techniques and their performances.[7] Compared to the RSA (an asymmetric cryptographic algorithm), ABE is slower and has more data overhead and energy consumption; however, the main advantage to use ABE is to enable a flexible and fine grained access control and to offer scalable key management because senders and receivers are completely decoupled. In IoT world, protection of privacy can be a challenging task because connected objects can generate an enormous amount of data, some of which actually constitute personal data. In addition, it is difficult to control the data flow without having any user interface or adequate tools for the user. An efficient solution to enforce security policy rules in IoT is described in [14-16]. This enforcement solution consists of a model-based security toolkit named SecKit that is integrated within the MQTT protocol. The policy enforcement support for MQTT is based on a custom policy enforcement point (PEP) component implemented in C language. The PEP is a connector that: 1) intercepts the messages exchanged inside the broker with a publish-subscribe mechanism; 2) notifies these messages as events in the SecKit policy decision point implemented in Java; and optionally 3) receives an enforcement action (allow, deny, modify, and delay) to be executed. In addition, this PEP has been embedded in the Mosquitto Broker [23] using security plugin. The following list summarizes advantages of this solution respect to the missing features in current MQTT implementations.

a. Modification of messages and identity obfuscation.
b. Delaying of messages to prevent real-time tracking of devices and users.
c. Enforcement when a message is delivered to a client in addition to enforcement when a client subscribes a topic.
d. Support for reactive rules to notify, log, or request user consent.
e. Misbehaviour checking rules, for DoS attack detection.

The main drawback of this approach is the high overhead when one publisher has many interested subscribers, and a policy needs to be checked for every subscriber.

5. 2. Security Vulnerabilities in Constrained Application Protocol (CoAP)

Constrained Application Protocol: The protocol is an HTTP remarkable version to match the IoT requirements for low overhead. The CoAP uses UDP protocol and encryption is most commonly accomplished using DTLS and sometimes with IPSec. DTLS is applied in the transport layer and the fundamental AES/CCM provides confidentiality, integrity, authentication, and non repudiation. The Californium framework (implemented in Java) provides a set of security capabilities for CoAP. There are four security modes defined for CoAP to implement TLS [26].No security PSK enabled by sensing devices pre-programmed with symmetric cryptographic keys. This mode is suitable for devices that are unable to support the public key cryptography. Raw public key (RPK) for devices that require authentication based on public key. This mode enables a TLS session without certificate.

Certificates to support authentication based on public key where keys are always validated according to a trusted entity known as certificate authority. The drawback of using the certificates is mainly due to heavy data format and fixed costs. A clear advantage, however, is the possibility to revoke certificates if the device is compromised. Key management is a drawback of the CoAP security which is a common issue in almost all protocols. Another problem is the heavy cost of computation and high handshake in the message which causes message fragmentation. Many studies proposed different solutions to compress the DTLS. In fact, a novel DTLS header compression scheme called Lithe has been proposed in [25] with the aim of significantly reducing the energy consumption by leveraging the 6LoWPAN standard without compromising the end-to-end security properties. In addition, the evaluation results show significant gains in terms of packet size, energy consumption, processing time, and network-wide response times when compressed DTLS is enabled.[7] A clear limitation of this solution is that DTLS header compression is applied only within 6LoWPAN networks. In [26], a security analysis between CoAP and MQTT is presented with a particular focus on the transport level protocol used (UDP for CoAP and TCP for MQTT), which inherently enforces the usage of DTLS for CoAP and TLS for MQTT. Moreover a set of security modes and also mandatory-to implement ciphers are supported by CoAP whilst, in contrast, the MQTT specification only enumerates a list of security considerations and does not enforce any kind of implementations. The comparative analysis has been conducted considering the four security modes already described. According to this analysis, RPK is not supported by MQTT, but it represents a mixed security alternative to heavier certificates and lightweight PSKs.

However, the traditional certificates-based authentication and encryption offers the highest level of security. Furthermore, the possibility to revoke certificates, considering illicit usage, makes it more capable to react to different attacks as already been proven with HTTP. In addition, due to different standard security mechanisms, the interoperability issue has a non trivial solution, mostly based on security level negotiation between IoT devices.

5. 3. Security Vulnerabilities in Advanced Message Queuing Protocol (AMQP)

As the Internet of Things expands to encompass billions of devices around the world, the cyber security CIA triad of Confidentiality, Integrity, and Availability becomes as significant as ever. With an exponential growth in the number of IoT devices, so too is there a corresponding exponential growth in the number of lines of communication and data transfer, be they via wired or wireless connections. Indeed, in a situation where every device is capable of communicating with every other device, the number of communications channels equals n(n-1)/2, where n is the number of devices involved. Every IoT communication channel is as vulnerable to potential man-in-the-middle cyber-attack as in a simple email communication between two end-users.

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

The four types of such active attacks are:

Replay : An attack entity replays data between communication sessions to impersonate a user to obtain information.

Masquerade :An attack entity gains access to a system or performs a malicious act by posing as an authorised entity.

Modification : An attack entity performs additions or deletions to the network communication content.

Denial of Service : An attack that inhibits legitimate users from accessing computer services.

Despite AMQP using TLS/SSL-based encryption on an underlying TCP-based transmission protocol, resolute threat entities will still be able to intercept and decipher IoT communications, given sufficient time. Not only are we seeing IoT devices being introduced on the market with insufficient security measures (Arias et al., 2015), but we are also seeing IoT networks being compromised by the introduction of carefully-crafted botnets. An example of such an attack occurred at an unnamed University in the United States early this year (2017) [26, 27]. Cybercriminals were able to crack default or poorly-configured passwords in one IoT device via a brute force attack taking advantage of the device's inadequate security measures. Once this device was under their control, specially designed malware was able to be installed (Palmer, 2017).

The malware then spread from IoT device to IoT device by a botnet which again brute-forced weak or defaults passwords. As the botnet spread, it locked administrators out and repeatedly changed the password on infected devices with each malware update (Moss, 2017). Within a short time frame, all 5,000+ devices were infected, and each device was making hundreds of DNS requests for seafood restaurants (Mezzofiore, 2017) [26].

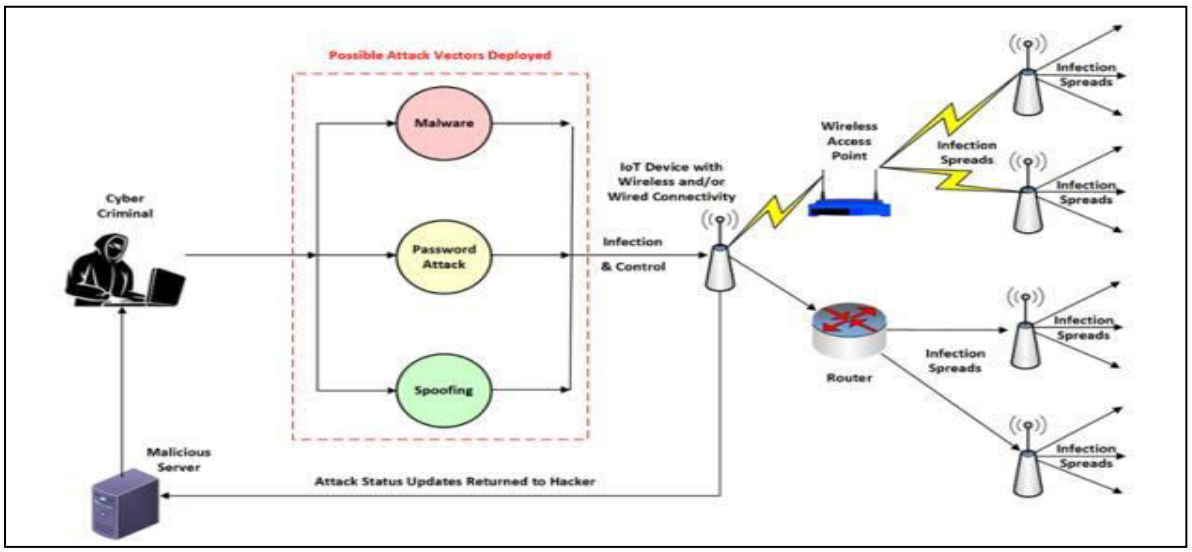


Figure 3: Hypothetical Interpretation of Attack

Figure 3 is a theoretical graphical interpretation of how the botnet attack on the University above may have been initiated and spread. The consequence of this DDoS attack was a severe slowing of the University's Internet access resulting in a loss of availability of resources required by students and staff (Palmer, 2017). What makes this incident particularly interesting is that it is one of the few cases to date which has seen a botnet DDoS attack spread and then directed against the same network on which the infected devices are hosted. If such an attack is to involve the compromise of server-to-server communications hosting AMQP, then the potential would be for multiple IoT networks to be seeded with such internally-spreading infections; causing widespread compromise of AMQP-enabled IoT devices [26].

5. 4.Security Vulnerabilities in Data Distribution Service (DDS)

Data Distribution Services or DDS (Object Management Group, 2015) is an open standard primarily intended for peer-to-peer inter-device communications. This protocol defines a data-centric publish/subscribe model and is focussed on low latency communications between devices, rather than between a device and a server or between two servers. The specification defines DDS as: "a Data-Centric Publish-Subscribe (DCPS) model for distributed application communication and integration. This specification defines both the Application Interfaces (APIs) and the Communication Semantics (behaviour and quality of service) that enable the efficient delivery of information from information producers to matching consumers." (Object Management Group, 2015, p. 1).

Industrial Control Systems has viewed as a concern for cause in recent times (Harp & Gregory-Brown, 2016) and also several legacy systems uses a pre defined standards, protocols and applications and implemented the same when the attack was primarily physical based, due to very less interaction between devices. However, in an interconnected world, ICS are gaining attention from cyber adversaries. For example, in 2015 Ukraine's power grid was attacked (Lee, Assante & Conway, 2016) and availability severely compromised after attackers gained access to SCADA systems and shut down parts of the grid. This was one of the first known successful cyber-attacks on power infrastructure, highlighting the growing threat of sophisticated attack operations against cyber-physical infrastructure.

DDS has found uses in many critical environments, such as amongst the energy and aerospace industries, as well as the military. Wang et al. (2008) explored the use of DDS in network-centric operations and warfare systems, demonstrating the increased use of these protocols in environments where security is essential. This is unsurprising as the DDS protocol has broad usage in military applications, having originally been developed by Thales (2015) for use in their TACTICOS Combat Management System. This usage has been one of the primary drivers for the high performance and resilient design requirements of DDS. DDS defers to TLS to provide the bulk of security rather than providing security at the application layer. However, reliance on TLS is clearly not sufficient, given the creation of a standardised post-protocol ratification security specification (aptly named DDS Security). This additional specification provides "authentication, authorization, non-repudiation, confidentiality and integrity" (Object Management Group, 2016) to DDS implementations. He & Liang (2015) have analysed the DDS specification for security issues and put forward a scenario where unauthorised publishers or subscribers may be able to inject data into the DDS network or receive data not intended for the legitimate recipient. They present a high-level overview of theoretical attacks on DDS and it is these types of attacks that DDS Security has been designed to

mitigate. Unfortunately, at this point there appears to be limited research on the effectiveness of the DDS Security specification in mitigating the defined theoretical attacks.

Given the range of vulnerable network protocols in use in the IoT, and the associated cost of data breaches; further research is necessary to reduce the attack surface of critical infrastructure installations. The following section describes a series of laboratory experiments undertaken which aims to test a subset of vulnerabilities specific to the DDS protocol.

6. Vulnerabilities And Issues in MQTT Protocol

This outlines about most widely used application layer protocol MQTT and its security issues and countermeasures. Although research on MQTT security is still scant, some incipient work has been already done about its security issues. Almost all security problems that arise are related to the state in which the protocol works due to the lack of common standards like Lack of authentication, Lack of authorization, Lack of confidentiality, Lack of integrity.

6.1 Issues& Threats in MQTT Protocol

MQTT features different security mechanisms, but most of them are not configured or provided by default, such as data encryption or entity authentication. Authentication mechanisms, such as using the physical address of the device (MAC), exist and are controlled by the broker by registering a device's information once it tries to connect. Access authorization can be done by the broker using a mechanism called an Access Control List (ACL). The ACL, as the name implies, contains records of information such as the identifiers and passwords of the different clients that are allowed to access different objects and can also specify what functions the client can perform on these. According to Reference [26,27], confidentiality is a major requirement of a secure system and can be accomplished at the application layer by encrypting the message that needs to be published. This method of encryption can either be implemented as client-to-broker or end-to-end. In a client-to-broker type of encryption, the broker decrypts the information that is being published to a topic and respectively encrypts the values that it needs to send to other clients. In an end-to-end situation, the broker cannot decrypt the information being published to topics and it forwards the cipher ext to other devices.[23] In the latter method, the broker needs fewer computational resources and less energy as it only functions as a courier and does not require any additional modules that can encrypt/decrypt messages. Nonetheless, additional security mechanisms can also be implemented on lower layers. According to Reference [26][27]one way to reliably ensure the security of a communication channel at the transport layer is by using Transport Layer Security protocol (for TCP) or even Datagram Transport Layer Security (in the case of UDP). Additionally, according to [27], encryption at the link layer can be achieved by using one of the many algorithms available, such as Advanced Encryption Standard (AES) in Counter Block Mode or AES in Counter with CBC-MAC mode, also called CCM mode. This type of security mechanism provides some additional advantages compared to other methods, such as increased efficiency due to the hardware acceleration capabilities found on radio chips.[36]

Message Queuing Telemetry Transport protocol is a publisher/subscriber messaging protocol specifically developed for constrained devices. Message queuing telemetry transport (MQTT) security is based on the TLS/SSL to provide transport encryption. It provides a security against eavesdropping. On the application layer, MQTT application provides client identifier and username/password credentials which can be used for devices authentication. The disadvantage of MQTT security is the

use of TLS/SSL which is not optimized for constrained devices. In fact, using TLS/SSL with certificates and session key management for a multitude of heterogeneous devices, is surely cumbersome [28]. For this reasons, a more scalable, lightweight, and robust security mechanism is required. In [28] a secure MQTT (SMQTT) is proposed to increase security features of the existing MQTT protocol and its variants based on lightweight attribute-based encryption (ABE), over elliptic curves. The advantage of using ABE is due to its inherent design which supports broadcast encryption (one encryption message delivered to multiple intended users) that make it suitable for IoT applications; moreover, the feasibility of SMQTT approach through simulations and performance evaluation has been validated. In [27], two different types of ABEs, key-policy ABE and cipher text-policy ABE, have been evaluated on different classes of mobile devices including a laptop and a Smartphone providing a comprehensive study of ABE techniques and their performances. Compared to the RSA (an asymmetric cryptographic algorithm), ABE is slower and has more data overhead and energy consumption; however, the main advantage to use ABE is to enable a flexible and fine grained access control and to offer scalable key management because senders and receivers are completely decoupled. In IoT world, protection of privacy can be a challenging task because connected objects can generate an enormous amount of data, some of which actually constitute personal data.[27]. In addition, it is difficult to control the data flow without having any user interface or adequate tools for the user. An efficient solution to enforce security policy rules in IoT is described in [18, 21, 22]. This enforcement solution consists of a model-based security toolkit named SecKit that is integrated within the MQTT protocol. The policy enforcement support for MQTT is based on a custom policy enforcement point (PEP) component implemented in C language. The PEP is a connector that: 1) intercepts the messages exchanged inside the broker with a publish-subscribe mechanism; 2) notifies these messages as events in the SecKit policy decision point implemented in Java; and optionally 3) receives an enforcement action (allow, deny, modify, and delay) to be executed. In addition, this PEP has been embedded in the Mosquitto Broker [23] using security plugin. The following list summarizes advantages of this solution respect to the missing features in current MQTT implementations.

1. Modification of messages and identity obfuscation.
2. Messages delay to check real-time tracking of IoT enabled devices and users.

SLNO	Vulnerability/ Challenges	Problem Description
1	DoS Attack	Deceiving node to breach defensive system
2	Sphear Phishing Attack	Luring emails for adversary gains
3	Sniffing Attack	Introduction of a sniffer application into the system
4	Overwhelm Attack	Undue consumption of energy by nodes and bandwidth
5	Insecure web interface & Data Privacy	Log and keys leakage at IoT end-node, illegitimate malicious nodes feeding contaminating data and/or accessing critical information (Malicious Code Injection due to end user hacking techniques)
6	Insecure mobile interface &	Unsecured apps, no Device Lockout, In-

	Cloud Interface	Cloud data leakage, Cross site scripting, poorly configured SSL/TSL
7	Insecure Remote Security Configuration	Fails to implement security measures @ interfaces, IoT end -node, end-device, end-gateway, no security logging, lack of granular permission model, lack of add-on password security options, lack of comprehensive security management
8	Insecure Software/Firmware	Threats to system from pirated softwares, malware installations, unencrypted update files, inability to receive timely security patches
9	Insufficient Authentication/Authorization	Lack of multifactor authentication, unsecure password recovery mechanism, Account Enumeration, lack of Role based access, No account Lockout
10	Risk Assessment/Trust Management	Lack of convenient tools for real time risk expectancy, threat detection and security reporting, absence of global and standard trust policies
11	Lack of Protocol Standardization	Lack of global standards and policies guiding development of security protocols, failure of existing policies to provide 100% protection from threats
12	Existing protocols coping with newer & stronger threats	Network bottlenecks are still prevalent in existing security protocols which are only relatively successful (like CoAP)

Table-2 Threats and Vulnerabilities

Acknowledgement when a message is succefully delivered to a client in addition to intimation when a client subscribes a topic.

3. Support for reactive rules to notify, log, or request user consent.
4. Misbehaviour checking rules, for DoS attack detection.

The major problems of this approach is the maximum overhead when one publisher has many interested subscribers, and a policy needs to be checked for every subscriber. This overhead introduces a small latency of a few in terms of ms. However, we restrict our discussion to protocol based security authentication, especially at the Application layer ie MQTT protocol in particular. Achieving end-to-end security triggers network challenges due to the discrepancy between the high demand for security standards and the available envisioned constrained hardware. Unprotected protocols (without security based implementations) are often vulnerable to various network attacks, eavesdropping, spoofing etc. Having SSL/TLS, IPSec, DTLS or any other security mechanism still does not assure the protocol of flawless security. In fact, IPSec faces Network Address Translation

(NAT), Port Address Translation (PAT) and multicast communication issues. DTLS does not support multicast communications since it lacks group key management. Both IPSec and DTLS have an incompetent QoS, Access Control and network trust and rely upon out-of-the-box extra protocols like Extensible Authentication Protocol (EAP) and Internet Key Exchange (IKE). SSL/TLS is expensive to be used in constrained device.[27]

Vulnerabilities are the weaknesses of a system due to poor design which allow the network to be hacked illegally. An attacker may bank upon improperly maintained network access and permissions, buffer overflow, cross site scripting, error configurations, data tampering and poor data authentication mechanisms. The table-2 and Table-3 provides a classification for security threats and counter measures related to all most all protocol in AL including MQTT. [57]They are Attacks,, Insecure web interface & Data Privacy , Insecure mobile interface & Cloud Interface , Insufficient Authentication/Authorization, Privacy Leak, DoS Attack, Malicious Code and Social Engineering etc.

Table-3 Solutions and Counter measure

SLNO	Vulnerability/ Challenges	Solutions Proposed
1	DoS Attack	1 Dynamic threat anticipation ASTM 2 Adaptive learning technique with changing internal parameters 3 Risk transfer mechanism based security systems 4 Support for Software Defined Networks (SDNs) architectures
2	Sphear Phishing Attack	1 Dynamic threat anticipation ASTM 2 Adaptive learning technique with changing internal parameters 3 Risk transfer mechanism based security systems 4 Support for Software Defined Networks (SDNs) architectures
3	Sniffing Attack	1 Dynamic threat anticipation ASTM 2 Adaptive learning technique with changing internal parameters 3 Risk transfer mechanism based security systems 4 Support for Software Defined Networks (SDNs) architectures
4	Overwhelm Attack	1 Dynamic threat anticipation ASTM 2 Adaptive learning technique with changing internal parameters 3 Risk transfer mechanism based

			security systems
		4	Support for Software Defined Networks (SDNs) architectures
5	Insecure web interface & Data Privacy	1	Preference Based Privacy
		2	Protection Method - Third party evaluation, report to service provider and appropriate security level based sensed preferences
6	Insecure mobile interface & Cloud Interface	1	Stronger passwords
		2	Testing the interface against the vulnerabilities of software tools (SQLi and XSS)
		3	Using https along with firewalls
7	Insecure Remote Security Configuration	1	Remote safe configuration
		2	Scalable security enhancement system of the SMC model for distributed resources – SMSC
		3	Simplified security management of network security teams
8	Insecure Software/Firmware	1	Encryption with validation
		2	Anti-virus, anti-adware, firewalls, Real Time Intrusion Detection Systems (IDS)
		3	Security patches
		4	Code with languages such as JSON, XML, SQL and XSS needs to be tested carefully
9	Insufficient Authentication/Authorization	1	Cross-layer authentication and authorization Sensitive information isolation/Data leakage protection
		2	Administrator/Identity Manager authentication Effective Key coordinate sharing, frequent key coordinate updates
		3	Identity Authentication and Capability based Access Control (IACAC)
		4	Strong Encryption schemes
		5	Cryptographic Hash functions & Feature Extraction
		6	Decentralized control of authentication using user-dependent security context
10	Risk Assessment/Trust Management	1	Security quantified in terms of incident and asset loss – CCM

		2	Mutual trust for inter-system security
		3	Agent-based and weight-based trust models
11	Lack of Protocol Standardization	1	Smart Object Lifecycle Architecture for Constrained Environments (SOLACE)
12	Existing protocols coping with newer & stronger threats	1	TLS/DTLS and HTTP/CoAP mapping
		2	Mirror Proxy (MP) and Resource Directory
		3	TLS-DTLS tunnel and message filtration using 6LBR .

Despite of the amount of work and standards on IoT, developing a successful IoT application is still not an easy task due to multiple challenges. These challenges include: mobility, reliability, scalability, management, availability, interoperability, cost and energy harvesting. So Protecting and securing the network connecting IoT devices to back-end systems on the Internet is more challenging than traditional network security because there is a wider range of communication protocols, standards, and device capabilities. A lot of security principles should be enabled at each layer for proper and efficient working of these applications. The most widely used application layer protocol is MQTT. The security issues and threats for Application Layer Protocol MQTT is particularly selected and analysed. Almost all security problems that arise are related to the state in which the protocol works due to the lack of common standards like Lack of authentication, Lack of authorization, Lack of confidentiality, Lack of integrity was discussed. So its very clear that these most common security issues that need to be solved through proper protocol configurations to increase the trust in IoT.

7. Conclusion

In this chapter we have provide a comprehensive survey of protocols for IoT and their security issues. The main objective is to give an insight to developers and service providers of different layers of protocols in IoT and the study related to the Application Layer IOT protocols ie MQTT, AMQP ,XMPP and CoAP and DDS, Study of MQTT protocol in details and also analysis of vulnerabilities in MQTT protocol with the counter measure for identification and prediction of better configuration mechanism for MQTT protocols.

Security is still one of the most critical challenges in IoT platforms So on the other hand, to prevent attacks, and to reduce vulnerabilities the organizations must ensure that they educate their consumers about the correct security procedures to be followed while using an IoT system. It is evident that successful attackers are smart since their success is based on knowledge. But it is also true that for successful IoT projects, the designers must be smarter, in other words be at least one step in front of any smart attacker. It is a continuous competition between the two parties and will forever be like that, since none is truly wise, meaning know and understand everything. For that, like in any domains, the IoT research has to continue forever, sooner or later any reasonable technological barrier that can be imagined nowadays has to be broken. Future work will be aimed at implementing all these protocols and obtain an experimental and quantifiable comparison among them ie the Comparison of

Application Layer Protocols (MQTT, AMQP ,XMPP and CoAP and DDS) including the security and deployed parameters Via Experimentation.

8. References

[1] P. N. Mahalle, N. R. Prasad, R. Prasad – "Object classification based context management for identity management in internet of things", International Journal of Computer Applications, vol. 63, no. 12, 2013

[2] Fei Hu – "Security and Privacy in Internet of Things (IoT).Models, Algorithms and Implementations", CRC Press, 2016

[3] Thamer A. Alghamdi – "Security Analysis of the Constrained Application Protocol in the Internet of Things", IEEE (2013)

[4] C. Hongsong, F. Zhongchuan, Dongyan – "Security and Trust Research in M2M System", Vehicular Electronics and Safety (ICVES), 2011 IEEE International Conference

[5] VasileiosKaragiannis, PeriklisChatzimisios, Francisco Vazquez-Gallego, Jesus Alonso-Zarate – "A Survey on Application Layer Protocols for the Internet of Things", Transaction on IoT and Cloud Computing 2015

[6] N.D. Lane, E. Miluzzo, H. Lu,D. Peebles,T.Choudhury, andA. T. Campbell, "A survey of mobile phone sensing," *IEEE Communications Magazine*, vol. 48, no. 9, pp. 140–150, 2010.

[7] Dan Dinculeană and Xiaochun Cheng, "Vulnerabilities and Limitations of MQTT Protocol Used between IoT Devices", Appl. Sci. 2019, ISSN 2076-3417 9, 848; doi:10.3390/app9050848

[8] A. Schmidt and K. Van Laerhoven, "How to build smart appliances?" *IEEE Personal Communications*, vol. 8, no. 4, pp. 66–71, 2001.

[9] W. Z. Khan, Y. Xiang, M. Y. Aalsalem, and Q. Arshad, "Mobile phone sensing systems: a survey," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 1, pp. 402–427, 2013.

[10] ITU Telecommunication Standardization Sector, "ITU-T Recommendation database," 2012. [Online]. Available: http://handle.itu.int/11.1002/1000/11559-en?locatt=format:pdf&auth. [Accessed 13 April 2015].

[11] H. Ning and Z. Wang, "Future internet of things architecture: like mankind neural systemor social organization framework?" *IEEE Communications Letters*, vol. 15, no. 4, pp. 461–463, 2011.

[12] Kun Wang JianmingBao and Meng Wu Weifeng Lu "Research on Security Management for Internet of Things" 201O International Conference on Computer Application and System Modeling Vol-I5 pg- 133 to 137. [2] SarfrazAlam, Mohammad M. R. Chowdhury and Josef Noll "Interoperability of Security-Enabled IoT" Wireless Commun (2011) vol-61:567–586 .

[13] HuiSuo, Jiafu Wan, CaifengZou and Jianqi Liu "Security in the Internet of Things: A Review" 2012 International Conference on Computer Science and Electronics Engineering pg- 648 to 651.

[14] Wang Chen "AN IBE-BASED SECURITY SCHEME ON INTERNET OF THING 2012 Proceedings of IEEE CCIS pg- 1046 to 1049.

[15] Stefan Poslad, Mohamed Hamdi and HabtamuAbie" International Workshop on Adaptive Security & Privacy management for the Internet of Things" 2013 pg-373 to 378

[16] MirzaAbdurRazzaq, Muhammad Ali Qureshi, SajidHabib Gill, SaleemUllah "Security Issues in the Internet of Things (IoT): A Comprehensive Study" International Journal of Advanced Computer Science and Applications, Vol. 8, No. 6, 2017.

[17] A.Arun Raja, R.Naveedha, G.Niranjanadevi and V.Roobini "AN INTERNET OF THINGS (IOT) BASED SECURITY ALERT SYSTEM USING RASPBERRY PI" ASIA PACIFIC INTERNATIONAL JOURNAL OF ENGINEERING SCIENCE Vol. 02 (01) (2016) 37–41

[18] HafsaTahir, Ayesha Kanwer and M. Junaid "Internet of Things (IoT): An Overview of Applications and Security Issues Regarding Implementation" INTERNATIONAL JOURNAL OF MULTIDISCIPLINARY SCIENCES AND ENGINEERING, VOL. 7, NO. 1, JANUARY 2016.

[19] Mario BallanoBarcena and Candid Wueest "Insecurity in the Internet of Things" http:// www.symantec.com Version 1.0 – March 12, 2015.

[20] J. Sathish Kumar and Dhiren R. Patel "A Survey on Internet of Things: Security and Privacy Issues" International Journal of Computer Applications (0975 – 8887) Volume 90 – No 11, March 2014 pg-20 to 26.

[21] Suchitra.C and Vandana C.P "Internet of Things and Security Issues" IJCSMC, Vol. 5, Issue. 1, January 2016, pg.133 – 139 .

[22] KANG Kai1, PANG Zhi-bo and WANG Cong "Security and privacy mechanism for health internet of thing" December 2013, 20(Suppl. 2): pg-64–68 www.sciencedirect.com/science/journal/10058885.

[23] Kai Zhao and LinaGe "Survey on the Internet of Things Security" 2013 Ninth International Conference on Computational Intelligence and Security. pg-663 to 667

[24] PallaviSethi and Smruti R. Sarangi "Internet of Things: Architectures, Protocols, and Applications," Hindawi Journal of Electrical and Computer Engineering Volume 2017, Article ID 9324035, 2017.

[25] http://en.wikipedia.org/wiki/Advanced Message Queuing Protocol, cited 28 Jul 2014.

[26] Frank T. Johnsen, Trude H. Bloebaum, Morten Avlesen, SkageSpjelkavik, BjørnVik, Evaluation of Transport Protocols for Web Services, Military Communications and Information Systems Conference (MCC), 7-9 Oct. 2013, pp. 1-6.

[27] Joel L. Fernandes, Ivo C. Lopes, Joel J. P. C.Rodrigues, Sana Ullah, Performance Evaluation of RESTful Web Services and AMQP Protocol, Fifth International Conference on Ubiquitous and Future Networks (ICUFN), 2-5 July 2013, pp. 810-815.

[28] Angelo P. Castellani, MattiaGheda, Nicola Bui, Michele Rossi, Michele Zorzi, Web Services for the Internet of Things through CoAP and EXI, IEEE International Conference on Communications Workshops (ICC), 5-9 June 2011, pp. 1-6.

[29] SyeLoongKeoh, Sandeep S. Kumar, HannesTschofenig, Securing the Internet of Things: A Standardization Perspective, Internet of Things Journal IEEE (Volume: 1, Issue: 3), June 2014, pp. 265-275.

[30] Maria Rita Palattella, Nicola Accettura, Xavier Vilajosana, Thomas Watteyne, Luigi Alfredo Grieco, GennaroBoggia, MischaDohler, Standardized Protocol Stack for the Internet of (Important) Things, Communications Surveys & Tutorials IEEE 15(3), 2013, pp. 1389-1406.