View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by Research Online @ ECU

Edith Cowan University

Research Online

Australian Information Security Management Conference

Conferences, Symposia and Campus Events

2018

XMPP architecture and security challenges in an IoT ecosystem

Muhammad Imran Malik
Edith Cowan University, muhammad.malik@ecu.edu.au

Ian Noel McAteer
Edith Cowan University, imcateer@westnet.com.au

Peter Hannay

Syed Naeem Firdous
Edith Cowan University, n.syed@ecu.edu.au

Zubair Baig

Follow this and additional works at: https://ro.ecu.edu.au/ism

 Part of the Information Security Commons

Recommended Citation

Malik, M. I., McAteer, I., Hannay, P., Firdous, S., & Baig, Z. (2018). XMPP architecture and security challenges in an IoT ecosystem. DOI: https://doi.org/10.25958/5c52735166690

DOI: 10.25958/5c52735166690
Malik, M.I, McAteer, I.N., Hannay, P., Syed, N.F., & Zubair, B. (2018). XMPP architecture and security challenges in an IoT ecosystem. In *proceedings of the 16th Australian Information Security Management Conference* (pp. 62-73). Perth, Australia: Edith Cowan University.
This Conference Proceeding is posted at Research Online.
https://ro.ecu.edu.au/ism/219

XMPP ARCHITECTURE AND SECURITY CHALLENGES IN AN IOT ECOSYSTEM

Muhammad Imran Malik[1], Ian Noel McAteer[1], Peter Hannay[1,2], Syed Naeem Firdous[1,], Zubair Baig[3]
[1]School of Science, Edith Cowan University, [2]Asterisk Information Security, Perth, Australia
[3]Data 61, CSIRO, Melbourne, Australia
muhammad.malik@ecu.edu.au, i.mcateer@ecu.edu.au, peter.hannay@asteriskinfosec.com.au,
n.syed@ecu.edu.au, zubair.baig@data61.csiro.au

Abstract

The elusive quest for technological advancements with the aim to make human life easier has led to the development of the Internet of Things (IoT). IoT technology holds the potential to revolutionise our daily life, but not before overcoming barriers of security and data protection. IoTs' steered a new era of free information that transformed life in ways that one could not imagine a decade ago. Hence, humans have started considering IoTs as a pervasive technology. This digital transformation does not stop here as the new wave of IoT is not about people, rather it is about intelligent connected devices. This proliferation of devices has also brought serious security issues not only to its users but the society as a whole. Application layer protocols form an integral component of IoT technology stack, and XMPP is one of such protocol that is efficient and reliable that allows real-time instant messaging mechanism in an IoT ecosystem. Though the XMPP specification possesses various security features, some vulnerabilities also exist that can be leveraged by the attacking entity to compromise an IoT network. This paper will present XMPP architecture along with various security challenges that exist in the protocol. The paper has also simulated a Denial of Service (DoS) attack on the XMPP server rendering its services unresponsive to its legitimate clients.

Keywords

Internet of Things (IoT), Extensible Messaging and Presence Protocol (XMPP), Cyber Security, Cyber-Physical Systems (CPS), IoT Architecture

INTRODUCTION

> *"When the winds of change blow, some people build walls and other build windmills" (Chinese proverb)*

Kevin Ashton first coined the term Internet of Things (IoT) in 1998/1999 (Rose, Eldridge, & Chapin, 2015; Swamy, Jadhav, & Kulkarni, 2017; Wu, Lu, Ling, Sun, & Du, 2010) and represented the concept that all electronic and non-electronic things stand connected in real time with the ability to be managed, controlled, and monitored remotely. Ashton explained the concept behind IoT by illustrating the power of Radio-Frequency Identification (RFID) tags in a corporate supply chain system that counts and track goods without human intervention (Rose et al., 2015). At the time of this writing, the market has already seen a broad array of IoT devices in numerous sectors of our society ranging from homes, healthcare, agriculture, industry, and so on. Due to this wide implementation, technology research companies have projected 50 billion IoT devices by 2020 (AT&T, 2016; Cisco, 2016; TrendMicro, 2014) which makes it an attractive target for adversaries.

IoT uses several protocols that are broken into eight layers of infrastructure, identification, transport, discovery, data protocol, device management, semantic and multi-layer frameworks. At the application/data layer, numerous protocols are currently in use which includes Extensible Messaging and Presence Protocol (XMPP), Message Queuing Telemetry Transport (MQTT), Constrained Application Protocol (CoAP), Simple Sensor Interface (SSI) and Advanced Message Queuing Protocol (AMQP), to name a few. This paper will, however, focus only on Extensible Messaging and Presence Protocol (XMPP) which is successfully used in various domains and considered as one of the most successful protocol since its inception.

XMPP evolved through an open development within the open-source Jabber community, is an application layer protocol for real-time data exchange and request/response services between multiple entities on a network. Developed in 1999 with the name Jabber, XMPP has extensively been used as a communication protocol which primarily was designed for Instant Messaging (IM) services like Google Hangouts and WhatsApp Messenger, etc. Over the years, this protocol has seen its implementation in services like Voice Over Internet Protocol (VoIP), gaming, etc. Of late, XMPP has seen wide implementation in IoT applications with its lightweight versions like

XMPP-IoT. This protocol is usually implemented using a client-server architecture where clients and servers communicate over a TCP connection. Being open source and widely implemented, XMPP is considered as a reliable and secure protocol for use in IoT applications, particularly in scenarios where devices need two-way communication with the servers or where two remotely connected devices need to talk to each other.

Despite all such advantages of IoTs in our lives and to the society, this ecosystem that connects millions of devices has the greater potential of compromises mainly due to security not being given priority from scratch. AT&T (2016) in their report has shown a 458% increase regarding vulnerability scans of IoT devices during the years 2014 and 2015. Such glaring statistics stresses the need of addressing security implications in IoT devices and services to enable the society to accrue maximum benefits out of it instead of its detrimental effects. This research digs deep into XMPP architecture and critically analyse its efficacy regarding its reliability, stability, and fault tolerant behaviour. More importantly, the research has examined XMPP's key security features such as network authentication, encryption algorithms, and data transfer mechanisms along with known vulnerabilities and threats.

Further, the research has endeavoured to undertake simulation/modelling of XMPP in a virtual environment for its tangible analysis. Following is our list of contributions through this research work:

1. In-depth analysis of XMPP architecture.
2. Efficacy of XMPP security features.
3. Security vulnerabilities in the protocol and threats posed by such weaknesses.
4. Simulation of DoS attack on XMPP.

BACKGROUND INFORMATION

Evolution of Connected Devices

Table 1 shows how IoT usage has grown exponentially in recent years and this trend is set to continue in the future:

Table 1. IoT Advancements through the Years

Year	Growth
1984	Integrated Services Digital Networks (ISDN) introduced the concept of networked-connected devices in the digital arena when it replaced the analog phone system (Tomsho, 2016).
1998	Internet Engineering Task Force (IETF) formalised the IPv6 protocol in preparation the anticipation that the IPv4 address space will be used up (Deering, 1998). IPv6 protocol allows an address space of 2^{128} unique IP addresses.
1999	The first usage of the term IoT (Ashton, 2009).
Early 2000s	The Internet refrigerator became the first commercial domestic appliance to utilise IoT technology (Osisanwo, Kuyoro, & Awodele, 2015).
2008	Evans (2011) estimated that the number of Internet-connected devices exceeded the number of people on earth.
2017	Different estimations of the number of IoT devices in existence. Gartner's figure of 8.4 billion (Meulen, 2017) contradicts IHS Markit's figure of 20 billion (Brown, 2017).
2020	Future estimations of the number of IoT devices in existence continue to show disparity. Gartner's figure of 20 billion Meulen (2017) contradicts Juniper Research's figure of 38 billion (Smith, 2017).

Not only is there an exponential growth in the number of connected devices, as shown in the table above, but how IoT devices are being used also becomes more and more diversified. However, the feverish race to be 'first-to-market' by the IoT manufacturers not knowing what technology stack to use for development and deployment for secure operations has resulted in serious issues related to cybersecurity (Gubbi, Buyya, Marusic, & Palaniswami, 2013). The underlying infrastructure that enables IoT devices to function consists of much more than the IoT devices themselves. TechBeacon (2017) defines the two-way communication between an IoT device at one end and a back-end data system at the other as being a four-stage process:

1. Target data acquired by IoT sensors and actuator devices.
2. Target data aggregated and converted from analogue to digital format within the IoT device.
3. Digital data pre-processed and transmitted from an Edge IT system.
4. Data analysis, management, and storage performed by a back-end data system.

While sensor data is transmitted from the IoT device to the back-end system, so too do command and control communications get transmitted via the same sequence back to the IoT device.

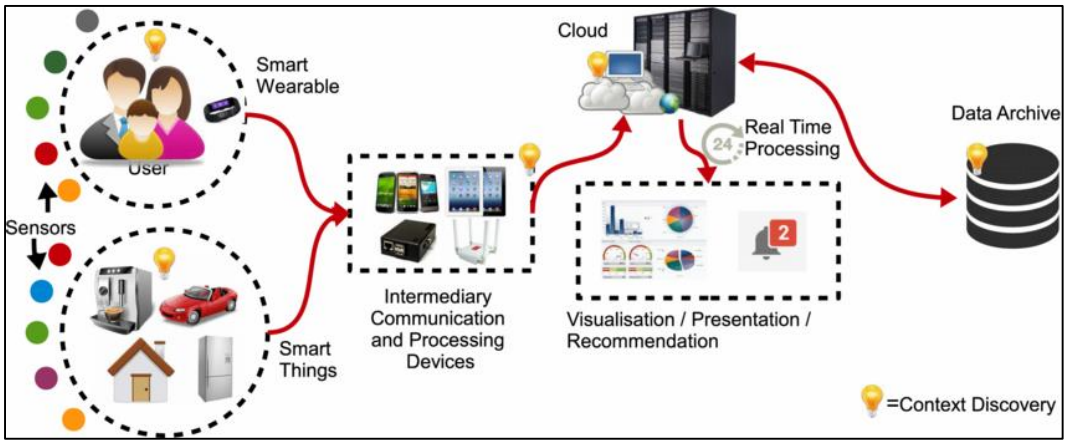

Figure 1. High-Level IoT Data Flow Architecture (Perera, Liu, Jayawardena, & Chen, 2014)

The Relation Between Cyber Security, IoT Security, and Cyber-Physical Systems

Transformation of paper-based systems into computer-based systems paved the way for organisations to develop and deploy solutions at a faster pace with the objective to bring efficiency and transparency into their workflows. Wide expansion of internet on the parallel also attracted such organisations to connect their systems to the internet. These advancements were made without keeping in view the issues of security or data protection. As the dark world started penetrating connected systems and gained access to the critical data, the term cybersecurity was started to come into the picture. Before the world could see the efficacy of IoTs and pervasive computing, the term cybersecurity was relevant to the security of data, servers, network infrastructure, and information security only (Russell & Duren, 2016). The exponential growth of IoTs over the last decade has seen physical devices exchanging data over the networks thus making possible to control such devices digitally. This transformation is termed as Cyber-Physical Systems (CPS) and redefines the conventional cybersecurity with the addition of security aspects of physical resources and machines that process digital data in the physical world (Russell & Duren, 2016). Examples of CPS include but not limited to smart cities, smart grids, medical devices, robotics, etc. (Bartocci, Hoeftberger, & Grosu, 2014). The IoT connects the sensors, actuators, and control/monitoring systems that form integral parts of a Cyber-Physical System, their [IoT] security is considered much more critical as any compromise of such devices may harm human lives or lead to physical destruction. Hence, it can be said that cybersecurity has a much broader dimension to cover when attributed to IoTs and their security.

IoT Architecture

The main features of IoT as discussed by Yang et al. (2011) includes all-out perception, having reliable transmission and intelligent operations. All-out perception and intelligent operations deals respectively with ubiquity and big data analysis and makes IoT an integral part of a Cyber-Physical System. However, ensuring reliable transmission for the exchange of information that fulfils the need for information assurance is a major concern with regards to cybersecurity. Firdous, Baig, Valli, and Ibrahim (2017), and Wu et al. (2010) argues that basic IoT architecture comprises of perception, network, and application layer. The distinction between these layers is discussed in the following table:

Table 2. IoT Layered Architecture (Firdous et al., 2017; Wu et al., 2010)

Layer	Description
Perception	Includes RFID devices, cameras, sensors, etc. that can identify the object, gather information, and process it. Processing of information can be done at the device level or in the cloud.

Layer	Description
Network	Takes information from the perception layer and enable communication between the device, with the cloud or with the gateway depending on the requirement.
Application	Responsible for secure transport of information to other devices or humans and ensuring reliability.

Vandana and Chikkamannur (2016) discussed an enhanced layered architecture for IoT that consists of five layers namely business, application, service management, object abstraction and objects/ things. The two different views about IoT layer architecture is mainly due to lack of standardisation. A brief overview of the 5-layered architecture is presented as under:

Table 3. 5-Layer IoT Architecture (Vandana & Chikkamannur, 2016)

Layer	Description
Objects/Things	Refers to the physical sensors and actuators in an IoT ecosystem. This layer sits at the bottom, digitise the data and forward it to the upper layers.
Object Abstraction Layer	Performs cloud computing and data management functionalities. Refines the data received from the sensors and actuators at the Object/Things layer. Various data transfer protocols/technologies like ZigBee, RFID, Bluetooth, etc. to transfer information to the upper layer also falls under the purview of this layer. Therefore, this layer can also be called as connectivity layer.
Service Management Layer	Allows integration with heterogeneous objects/things by enabling data processing without looking into specific hardware issues and how each device processes the data.
Application Layer	Provides high-quality smart services requested by the end-users such as health, temperature, air humidity measurements, etc.
Business Layer	Manages the overall IoT ecosystem (activities and services). This layer enables decision-making processes through big data (collected from the underlying layers) analysis as well as monitors and manages the other four layers.

Communication Models in an IoT Ecosystem

At the core of IoT resides the concept of how such devices connect and communicate to be able to exchange information. IoT devices use four different communication models under different situations as discussed in the RFC 7452. A succinct overview of these model is tabulated as under followed by their graphical illustration in an IoT ecosystem:

Table 4. Brief Overview of Communication Models in IoT (Tschofenig, Arkko, Thaler, & McPherson, 2015)

Model	Description
Device-To-Device (D2D)	Devices communicating directly with one another without the need of any intermediary application server between them. D2D communication uses protocols like ZigBee, Bluetooth, Z-Wave, etc.
Device-To-Cloud (D2C)	Devices connect to the application server hosted in the cloud by the service provider to exchange data and control messages. D2C communication model uses conventional networking means such as wired or wireless connections to connect a device with the IP network which they connect to the cloud.

Model	Description
Device-To-Gateway (D2G)	Devices exchange information with the cloud using an application-layer gateway in between typically known as ALG model. This involves the use of application software on the gateway device which ensures security and other functionalities such as data or protocol translation (Rose et al., 2015).
Back-End Data-Sharing Model	Firdous et al. (2017) argue that through back-end data-sharing model, an IoT device can communicate with authorised third parties. This model enables IoT devices to export and analyse sensors data from the cloud as well as the data from other devices/sensors, service providers, etc.

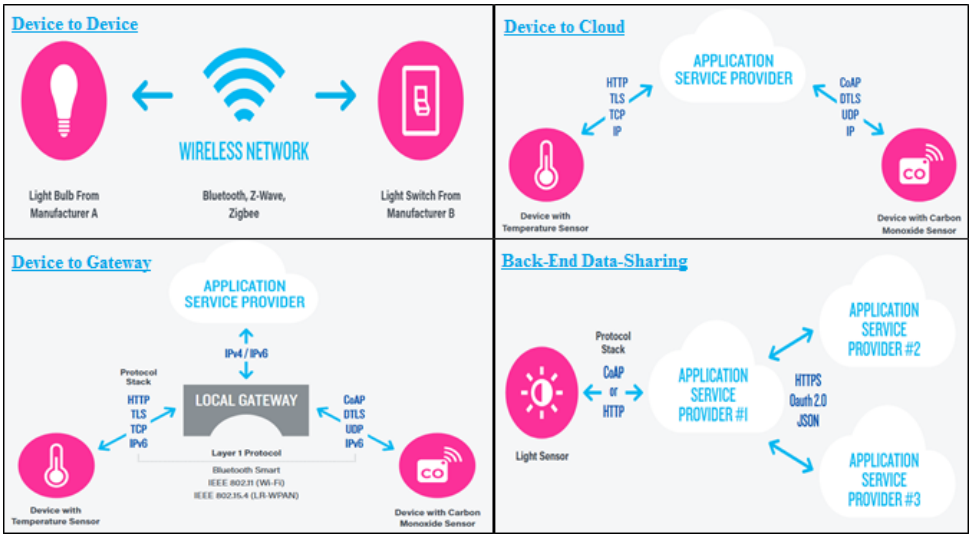

Figure 2. Communication Models in IoT (Rose et al., 2015)

IoTs are considered as self-configuring and adaptive networks that are complex and uses several interoperable communication protocols. In an IoT ecosystem, the communication of sensors/actuators can be categorised into three scenarios, i.e. basic, extended and cloud (Zhang, Cho, & Shieh, 2015). The basic scenario involves sensors/actuators communicating within a closed network that can often be referred to as LAN. When data is exchanged outside of a closed network, such a scenario can be termed as the extended one, and it exchanges data using a centralised or a decentralised network configuration. When data storage is done in the cloud, i.e. by using various data storage services provided by the vendors, cloud communication scenarios come into place. Every communication scenario discussed has its associated challenges. However, all of them are used due to the diverse nature of the IoT ecosystem. These pros and cons are tabulated as under:

Table 5. Challenges to Various Communication Scenarios (Zhang et al., 2015)

Scenario	Challenges
Basic	a. Authentication and authorisation to use the LAN b. Countermeasure for eavesdrop over wireless networks.
Extended	a. Authentication and authorisation to use the LAN b. Eavesdrop resistance over wireless networks c. Integrity assurance when using the public network d. Confidentiality assurance when using the public network
Cloud	a. Authentication and authorisation to use the LAN b. Eavesdrop resistance over wireless networks c. Confidentiality and integrity assurance over the Internet d. Authentication and authorization for the cloud service e. Confidentiality and integrity for the cloud storage

XMPP Application Layer Protocol

Communication protocols define a standard way between the sensors/objects to establish a meaningful interaction which allows an effective, legitimate and anticipated behaviour of all involved parties (Wang, 2017). Firdous et al. (2017) state that some protocols exist in an IoT ecosystem that facilitates communication in the scenarios discussed earlier. Russell and Duren (2016) also argue that an IoT ecosystem uses a wide array of protocols to enable message transfer and communication services, however, selection of appropriate IoT stack is often very critical and challenging as it requires an in-depth understanding of the system particularly with regards to its [system's] security requirements. Lack of standardisation in the IoT domain has allowed the use of any technology as long as it facilitates the connectivity and data exchange requirements of the system.

XMPP is one such protocol that is designed on the open technology stack and has the ability for both clients and servers to efficiently agree upon data required to be exchanged (Russell & Duren, 2016). XMPP (2017) argues that this protocol is one of four Instant Message (IM) protocols which has been developed to satisfy the rapidly expanding information society's need for short message services with an open and decentralised framework. XMPP's use of Extensible Markup Language (XML) overcomes prior difficulties in connecting an IM system with a non-IM system. Public IM services, such as LJ Talk, Nimbuzz, and HipChat exclusively use XMPP. Other popular IM applications like WhatsApp, Gtalk, and Facebook Chat also use XMPP on their back-end servers. Apart from instant messaging services, Moffitt (2010) and Ranot (2016) argues that this protocol is also used in multi-party chat, voice and video calls, collaboration, lightweight middleware, and generalised routing of XML data. Key features of XMPP are discussed as under:

1. Data between two devices is exchanged in small sets and structured pieces (Moffitt, 2010) and independent of the operating system in use (Al-Fuqaha, Guizani, Mohammadi, Aledhari, & Ayyash, 2015).
2. The underlying technology is rich in features and helps developers in implementing XMPP as their preferred choice as they can easily focus on unique pieces of the application being developed (Moffitt, 2010).
3. Easily understandable and allows SIP compatible multimedia-signalling for voice, video, data transfer, privacy control, and other applications (Moffitt, 2010; Ranot, 2016).
4. Uses Publish-Subscribe messaging pattern for data syndication and rich presence (Moffitt, 2010; Ranot, 2016).
5. Recognized by IETF through various specifications; RFC 6120 (Saint-Andre, 2011b), RFC 6121 (Saint-Andre, 2011c), and RFC 6122 (Saint-Andre, 2011a), having specifications suitable for IoT particularly due to the provision of Extension Protocols (XEP) that increases the functionality of XMPP (Wang, 2017). RFC 6120 was updated with RFC 7590 that introduced the use of Transport Layer Security (TLS) in the protocol (Saint-Andre & Alkemade, 2015).
6. A client-server protocol that works over TCP and allows either side to send data asynchronously with persistent connections (Jain, 2014; Moffitt, 2010).
7. Suitable for individually tailored messages, real-time or bidirectional communication (Waher, 2015).
8. Apart from the publish/subscribe pattern, XMPP supports request/response and push communication models (Al-Fuqaha et al., 2015; Saint-Andre, 2011b; Waher, 2015).
9. Easily extensible structured XML based messages (Al-Fuqaha et al., 2015; Moffitt, 2010; Waher, 2015).
10. Contains inbuilt security mechanism that includes authentication, authorisation and session encryption (Al-Fuqaha et al., 2015; Moffitt, 2010; Saint-Andre, 2011b; Waher, 2015).

XMPP ARCHITECTURE

The XMPP protocol outlines a format for exchanging data between the two or more communicating devices which in case of IoT ecosystem could be between sensors/actuators (D2D) or between a sensor/actuator and a server/cloud (D2S/C). Moffitt (2010) argues that systems involving XMPP protocol over the Internet are mostly accessible to all and thus such systems form a federated network of interconnected systems. The XMPP system consists of servers, clients, components, and server-plugins. While the concept behind servers and clients is understandable, Moffitt (2010) describe components as external to the servers to which clients can communicate with as a new service. A multi-user chat service is an example of such a component. The purpose of server-plugins

is mostly similar to components, but plug-ins can change the core behaviour of the server with reduced overhead compared to components as well as it can access the data structures being used in the internal server (Moffitt, 2010).

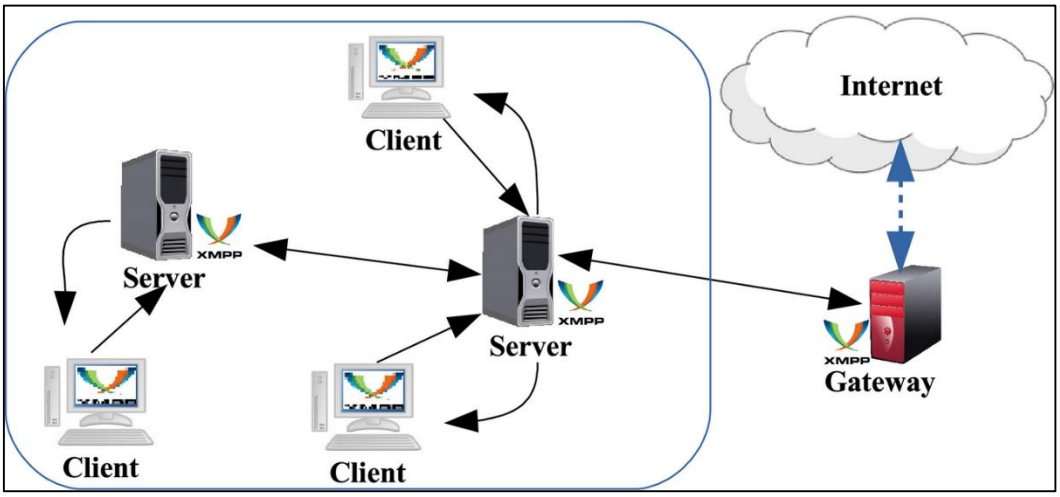

Figure 3. XMPP Communication Model (Al-Fuqaha et al., 2015)

Wang (2017) argues that XMPP uses client-server-server-client model in which clients do not connect directly to other servers. All communication is done through the respective server which then forwards the message to the client located under another server. The clients connect to the server using TCP port 5222 whereas server communicates with each other using port 5269 (Waher, 2015; Wang, 2017). The architecture of XMPP is similar to Simple Mail Transfer Protocol (SMTP) with the difference that XMPP is designed for real-time instant messaging applications with low latency (Waher, 2015). Nastase (2017) argues that IoT implementations that involve XMPP are usually deployed in decentralised client-server architecture and follows the client-server stream and server-server stream. Therefore, two clients cannot communicate directly without an intermediate entity, i.e. a server with some trust level.

XMPP uses a unique identifier called as addresses (also known as Jabber Identifier (JID)) assigned to each device for identification. Jabber IDs looks similar to an email address as it comprises of three parts, i.e. local part (username), the domain, and the resource (Moffitt, 2010; Waher, 2015; Wang, 2017). Moffitt (2010) argues that the domain name portion is the mandatory requirement and it contains the resolvable DNS name of the entity which could be a server, component, or a plug-in name. The full JID consist of all three components whereas an ID that contains local part (username) and the domain is termed as 'bare Jabber ID'.

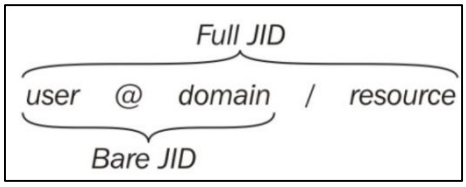

Figure 4. Jabber ID/Address format in XMPP (Waher, 2015)

The rapid and asynchronous exchange of small payload information within the XMPP network is handled by two XMPP entities, i.e. XML streams and XML stanzas (Moffitt, 2010; Saint-Andre, 2011b). The stream is a container that allows the exchange of XML elements between the two entities in a network. Al-Fuqaha et al. (2015) state that XML stanzas enable a client to connect to a server using a code that has three components. These three components form the core XMPP toolset and have following purposes/behaviours (Al-Fuqaha et al., 2015; Wang, 2017):

1. Presence: notifies and updates the status of an entity to the entities that have subscribed to it.
2. Message: send messages from one entity to another using push mechanism including the structures information that includes (source and destination addresses, types and IDs of XMPP entities).
3. Iq (info/query): works on the principle of a request-response mechanism allowing both get and set queries to make a request with 'iq result' or 'iq error' response from the recipient.

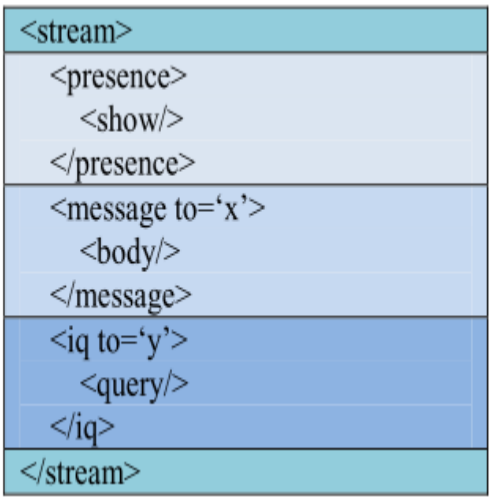

Figure 5. XMPP Stanza Structure (Al-Fuqaha et al., 2015)

The three components of XML stanzas discussed above can perform all the tasks in an XMPP network; however, before that, an authenticated XMPP session needs to be established. Moffitt (2010) argues that other than XML stanzas, connection life cycle in an XMPP protocol include connection, stream set up, authentication, and disconnection phases. These phases have following functionalities (Moffitt, 2010; Waher, 2015):

1. Connection: establishes communication with the XMPP server typically by utilising the Domain Name System (DNS) mechanism by querying the appropriate server records (DNS-SRV).
2. Stream Set Up: After establishing the connection, the client sends the XMPP stream to the server, and the server responds with the response stream.
3. Authentication: using Simple Authentication and Security Layers (SASL) protocol. The server can support plain text or MD5 authentication with some having the capability of using Kerberos or special tokens.
4. Disconnection: session is terminated and disconnected after information exchange is completed. This is achieved by sending the 'unavailable' presence type in the XML stanza that terminates the stream with the client.

Moffitt (2010), (Saint-Andre, 2011b), Al-Fuqaha et al. (2015), and Waher (2015) mention the following security features in XMPP:

1. Transport Layer Security (TLS) to ensure encryption while data is on the move.
2. Authentication using Simple Authentication and Security Layers (SASL) protocol.
3. Access control.
4. Privacy measurement.
5. Client and server certificates generation and validation.

CYBER SECURITY CHALLENGES

IoT Security Challenges

IoTs, at its core, is an extensive ecosystem of interconnected devices performing data gathering and data processing/analytics activities. Decisions based on the analytics coming from these connected devices could lead to catastrophic consequences if data gets corrupted through an attacking entity and thus present new and unique security challenges. Non-secure IoT devices, as well as services, allows cybercriminals to launch attacks that lead to exposure of critical information. One such example, in this case, is the rapid launch of IoT devices in the market with inadequate security features that have allowed the cybercriminals to easily launch carefully-crafted botnets that can seriously affect the performance of the IoT network. This makes information assurance pillars of maintaining Confidentiality. Integrity, and Availability more significant. Therefore, challenges to overcome security loopholes are increasing with the rapid deployment of IoT devices because of more sophisticated attack techniques used by nefarious people. Some of the active attacks associated with the use of IoTs that could be exploited by the cybercriminals discussed by Russell and Duren (2016) and TrendMicro (2014) are as under:

1. Protocol attacks
2. Sniffer/ Eavesdropping attacks
3. Denial of Service (Dos) or Distributed Denial of Service (DDoS) attacks
4. Cryptographic algorithm and key management attacks
5. Spoofing and masquerading attacks
6. Password-based attacks
7. Man-in-the-Middle attacks
8. Physical security attacks
9. Access control attacks
10. Wired and wireless scanning and mapping attacks

XMPP Security Challenges

The IETF specification on XMPP written by (Saint-Andre, 2011b) discusses the implementation of XMPP authentication mechanism using SASL and transport security using TLS. While SASL supports a set of methods that could be employed by the client for authentication, Nastase (2017) argues that there exists a possibility that a weak mechanism can be chosen from the available methods. By default, SASL uses a Base64 encoding which helps in hiding the easily recognised information such as passwords. However, it fails to provide any computational confidentiality (Nastase, 2017). The XMPP specification, therefore, proposes to use authentication mechanisms such as SCRAM-SHA-1 or SCRAM-SHA-1-PLUS that provide channel bindings and thus protect XMPP sessions from man-in-the-middle, spoofing and unauthorised access attacks (Nastase, 2017; Saint-Andre, 2011b).

XMPP uses TLS protocol with STARTTLS extension for channel encryption which protects the stream from tampering and eavesdropping (Nastase, 2017). Before SASL is used to maintain the confidentiality of credentials, a complete shake hand of TLS session needs to be complete. However, Anantharaman, Locasto, Ciocarlie, and Lindqvist (2017) argues that XMPP is vulnerable to crafted messages during a session that may well lead to exploits known as shotgun parsers. Further, Saint-Andre (2011b) states that before successful negotiation of TLS session, an attacker can tamper with the information such as ‘from’ and ‘to’ addresses that are exchanged in the initial stream header.

The XML stanza within the XMPP can travel along the multiple streams, and there is a strong likelihood that some of the streams are not protected using the TLS protocol (Nastase, 2017). Nastase (2017) further argues that this presents a major vulnerability in the XMPP and the only solution to this issue is to have a robust end-to-end encryption mechanism implemented that could ensure confidentiality and integrity of the stanzas travelling along the communication path through multiple hops.

In addition to the threats discussed above, Saint-Andre (2011b) and Nastase (2017) have presented following attacks that can be launched against XMPP systems:

1. Sniffing Passwords
2. Breaking passwords through dictionary attacks
3. Discovering passwords through dictionary attacks
4. Replaying, inserting, deleting, or modifying stanzas
5. Denial of Service (DoS) or Distributed DoS attacks
6. Privilege escalation attacks
7. Gaining control over on-path servers

A search on the Common Vulnerabilities and Exposures (CVE) databases for the known vulnerabilities of XMPP was made using ‘XMPP IoT’ search criteria which presented a total of 43 vulnerabilities in the last five years (since 2013) let alone 15 in 2017 at the time of writing. This not only confirms an increase in the use of IoT devices but also rise in the attacks launched by nefarious entities thus exposing XMPP architecture.

SIMULATION AND ANALYSIS

Security challenges associated with XMPP architecture in an IoT ecosystem as discussed in the above section confirm that the threat entities can launch massive attacks. This section will discuss the simulation of a Denial of

Service (DoS) attack which has the potential to either degrade the performance of the network or bring it down completely. Arış, Oktuğ, and Yalçın (2015) argue that DoS attacks are the most perilous threat to IoT ecosystems. The simulation has been performed by setting up virtual machines followed by a collection of results in the form of network graphs. The DoS attack was launched on a standalone XMPP server running on one of the virtual machines.

The simulation was set up through the deployment of Openfire version 4.1.6 which is a Real-Time Collaboration (RTC) server and use XMPP for instant messaging (ignite-realtime, 2017). Openfire server was set up on Ubuntu 16.04 machine with specifications of 1 CPU, 40 GB hard disk and 1 GB of RAM. The DoS attack was launched from another virtual machine running Ubuntu operating system by launching a TCP SYN flood attack to consume resources on the targeted server (running Openfire) and rendering it unresponsive. In SYN flood attacks, the threat entity repeatedly sends SYN packets to any, or a designated port of the targeted system to make the service(s) on offer either slow or stop them altogether. For the simulation and subsequent analysis in our case, SYN flood was launched using HPING3 tool (HPING3, n.d.) to flood Openfire server by sending SYN packets on port 9090.

Launching TCP SYN flood attack on the targeted system on port 9090 initially degraded the response time of the Openfire server against the legitimate requests. After a little while, the server automatically logged out thus becoming unresponsive. The screenshot below is taken from the Ubuntu's (target system) system monitor utility depicts that attack launched through HPING3 did not affect the CPU and memory performance but successfully exhausted the server bandwidth that increased up to 1.5MiB/s as compared to its normal usage of less than 20.0 KiB/s.

Figure 6. Ubuntu's System Monitor Depicting Server Performance during SYN Flood Attack

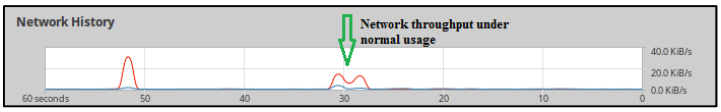

Figure 7. Server's Network Throughput during Normal Working

The above simulation presents vulnerability of XMPP server from TCP SYN flood attacks which are only required at the start when a session between client and server or server and server is established. Even though CPU and memory of the targeted system remained unaffected, the network throughput of the server was increased exponentially as compared to its normal behaviour leading to degraded system performance. Understanding this attack holistically presents that attack entities can launch SYN flood attacks from multiple devices within an IoT ecosystem and successfully bring down XMPP services. Firdous et al. (2017) argue that adequate firewall rulesets can detect such attacks and block malicious entities from achieving success in their evil aims.

CONCLUSIONS

"If you think technology can solve your security problems, then you don't understand the problems, and you don't understand the technology" (Bruce Schneier)

Cyber-attacks, in recent times, have seen a much higher degree of sophistication as the tactics and techniques employed by the attackers have become difficult not only to detect but also challenging to investigate and remediate. While organisations are continuously trying to improve their security posture, the attackers are also refining their attack methodologies instigating unprecedented levels of disruption. Increased dependence of critical infrastructures has seen a broad array of IoT devices enabling the use of automated systems. Some protocol

options are available for use in the IoT network. However, the challenge is to ensure that the correct protocol is utilised in an appropriate environment. To overcome the potentially devastating impact on the digital economy, IoT manufacturers need to secure their devices from scratch.

This research attempts using a theoretical methodology in which various information sources were referred, and available literature was analysed in detail. XMPP is a powerful and flexible real-time communication protocol for instant messaging which has a great potential for implementation in the IoT ecosystems. However, various threats that could impact the otherwise efficient behaviour of XMPP protocol also exist and have been discussed at length. With the help of a simulation depicting the vulnerability of XMPP server from DoS attacks using TCP SYN flood packets, the impact of threat was also quantified. Possible countermeasures to the threats have also been briefly discussed, where appropriate.

As part of the future work, the researchers aim to simulate other malicious attacks on XMPP discussed in this paper and analyse their impact with possible countermeasures to reduce the likelihood of such attacks from being effective.

REFERENCES

Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., & Ayyash, M. (2015). Internet of Things: A survey on enabling technologies, protocols, and applications. *IEEE Communications Surveys & Tutorials, 17*(4), 2347-2376. doi:10.1109/COMST.2015.2444095

Anantharaman, P., Locasto, M., Ciocarlie, G. F., & Lindqvist, U. (2017). *Building Hardened Internet-of-Things Clients with Language-theoretic Security.* Paper presented at the Security and Privacy Workshops (SPW), 2017 IEEE.

Arış, A., Oktuğ, S. F., & Yalçın, S. B. Ö. (2015). *Internet-of-Things security: Denial of service attacks.* Paper presented at the Signal Processing and Communications Applications Conference (SIU), 2015 23th.

Ashton, K. (2009). That 'internet of things' thing. Retrieved from http://www.rfidjournal.com/articles/view?4986

AT&T. (2016). *Exploring IoT security*. Retrieved from https://www.business.att.com/cybersecurity/docs/exploringiotsecurity.pdf

Bartocci, E., Hoeftberger, O., & Grosu, R. (2014). Cyber-Physical systems: Theoretical and practical challenges. *ERCIM NEWS, 97*, 8-9. Retrieved from https://ercim-news.ercim.eu/images/stories/EN97/EN97-web.pdf

Brown, P. (2017). 20 billion connected Internet of Things devices in 2017, IHS Markit says. Retrieved from http://electronics360.globalspec.com/article/8032/20-billion-connected-internet-of-things-devices-in-2017-ihs-markit-says

Cisco. (2016). Securing the Internet of Things: A proposed framework. Retrieved from https://www.cisco.com/c/en/us/about/security-center/secure-iot-proposed-framework.html#4

Deering, S. E. (1998). Internet protocol, version 6 (IPv6) specification. Retrieved from https://tools.ietf.org/html/rfc2460

Evans, D. (2011). The internet of things [Infographic]. Retrieved from http://blogs.cisco.com/diversity/the-internet-of-things-infographic

Firdous, S. N., Baig, Z., Valli, C., & Ibrahim, A. (2017). *Modelling and Evaluation of Malicious Attacks against the IoT MQTT Protocol.* Paper presented at the 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 IEEE International Conference on.

Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2013). Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Gener. Comput. Syst., 29*(7), 1645-1660. doi:http://10.1016/j.future.2013.01.010

HPING3. (n.d.). HPING 3. Retrieved from http://www.hping.org/hping3.html

ignite-realtime. (2017). Openfire. Retrieved from https://www.igniterealtime.org/projects/openfire/

Jain, J. (2014). What is XMPP and how does it work? [Blog comment]. Retrieved from https://www.quora.com/What-is-XMPP-and-how-does-it-work

Meulen, R. v. d. (2017). Gartner says 8.4 billion connected "Things" will be in use in 2017, up 31 percent from 2016. Retrieved from https://www.gartner.com/newsroom/id/3598917

Moffitt, J. (2010). *Professional XMPP Programming with JavaScript and jQuery*: John Wiley & Sons.

Nastase, L. (2017). *Security in the Internet of Things: A survey on application layer protocols*. Paper presented at the Control Systems and Computer Science (CSCS), 2017 21st International Conference on, Bucharest, Romania http://ieeexplore.ieee.org/document/7968629/

Osisanwo, F., Kuyoro, S., & Awodele, O. (2015). *Internet refrigerator – A typical internet of things (IoT)*. Paper presented at the 3rd International Conference on Advances in Engineering Sciences & Applied Mathematics (ICAESAM'2015), London (UK). http://iieng.org/images/proceedings_pdf/2602E0315051.pdf

Perera, C., Liu, C. H., Jayawardena, S., & Chen, M. (2014). A survey on Internet of Things from industrial market perspective. *IEEE Access, 2*, 1660-1679.

Ranot, I. (2016). What is XMPP and how does it work? [Blog comment]. Retrieved from https://www.quora.com/What-is-XMPP-and-how-does-it-work

Rose, K., Eldridge, S., & Chapin, L. (2015). *The Internet of Things: An overview*. Retrieved from https://www.internetsociety.org/wp-content/uploads/2017/08/ISOC-IoT-Overview-20151221-en.pdf

Russell, B., & Duren, D. V. (2016). *Practical Internet of Things security*. Birmingham, U.K.: Packt Publishing Ltd.

Saint-Andre, P. (2011a). Extensible Messaging and Presence Protocol (XMPP): Address Format. Retrieved from https://tools.ietf.org/html/rfc6122

Saint-Andre, P. (2011b). Extensible messaging and presence protocol (XMPP): Core. Retrieved from https://tools.ietf.org/html/rfc6120

Saint-Andre, P. (2011c). Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence. Retrieved from https://tools.ietf.org/html/rfc6121

Saint-Andre, P., & Alkemade, T. (2015). Use of Transport Layer Security (TLS) in the Extensible Messaging and Presence Protocol (XMPP). Retrieved from https://tools.ietf.org/html/rfc7590

Smith, S. (2017). 'Internet of Things' connected devices to almost triple to over 38 billion units by 2020. Retrieved from https://www.juniperresearch.com/press/press-releases/iot-connected-devices-to-triple-to-38-bn-by-2020

Swamy, S. N., Jadhav, D., & Kulkarni, N. (2017). *Security threats in the application layer in IOT applications.* Paper presented at the I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC), 2017 International Conference on.

TechBeacon. (2017). The 4 stages of an IoT architecture. Retrieved from https://techbeacon.com/4-stages-iot-architecture

Tomsho, G. (2016). *Guide to Networking Essentials* (7th ed.). Boston, MA: Cengage Learning.

TrendMicro. (2014). The Internet of Everything: Layers, protocols and possible attacks. Retrieved from https://www.trendmicro.com/vinfo/us/security/news/internet-of-things/ioe-layers-protocols-and-possible-attacks

Tschofenig, H., Arkko, J., Thaler, D., & McPherson, D. (2015). Architectural considerations in smart object networking.

Vandana, C. P., & Chikkamannur, A. A. (2016). IOT future in edge computing. *International Journal of Advanced Engineering Research and Science (IJAERS), 3*(12), 148-154. doi:dx.doi.org/10.22161/ijaers/3.12.29

Waher, P. (2015). *Learning internet of things*: Packt Publishing Ltd.

Wang, M. (2017). *Understanding security flaws of IoT protocols through honeypot technologies.* (Master of Science), Delft University of Technology, Netherlands. Retrieved from https://repository.tudelft.nl/islandora/object/uuid%3Af4be5514-e9df-499a-8eea-f78c510d3346?collection=education

Wu, M., Lu, T.-J., Ling, F.-Y., Sun, J., & Du, H.-Y. (2010). *Research on the architecture of Internet of Things.* Paper presented at the Advanced Computer Theory and Engineering (ICACTE), 2010 3rd International Conference on.

XMPP. (2017). XMPP. Retrieved from https://xmpp.org/

Yang, Z., Yue, Y., Yang, Y., Peng, Y., Wang, X., & Liu, W. (2011). *Study and application on the architecture and key technologies for IOT.* Paper presented at the Multimedia Technology (ICMT), 2011 International Conference on.

Zhang, Z.-K., Cho, M. C. Y., & Shieh, S. (2015). *Emerging Security Threats and Countermeasures in IoT*. Paper presented at the Proceedings of the 10th ACM Symposium on Information, Computer and Communications Security, Singapore, Republic of Singapore.