A Context-Aware IoT Security Framework For Elderly Care

Mohamed Abdurazak Hassan, Ahmad Ryad Soobhany

Heriot-Watt University, Dubai, U.A.E
mah2002@hw.ac.uk, r.soobhany@hw.ac.uk

Abstract

The use of IoT networks in healthcare has led to the adoption of smart home environments for users in their place of abode and one such example is smart home for elderly care. Sensors and devices in the network capture data that can be stored or transmitted and this can lead to security issues. The security solutions for securing the IoT network are usually applied blanket-wise. On the other hand, context-aware security approaches consider contextual data when applying security solutions. In this paper, we identified critical weaknesses in popular data transmission protocols as well as the interactions of devices communicating with them in a simulated environment. To mitigate the identified vulnerabilities, a Context-Aware IoT Security Framework was developed that dynamically adjusts security measures based on environmental and contextual factors in smart home environments for elderly care. By adopting a user-centric approach, the proposed framework minimizes the risk of unauthorized access, data manipulation, and network-based attacks, and ensures that the solution aligns with regulatory standards such as GDPR and HIPAA. The evaluation of the proposed framework demonstrated encouraging improvements to the protection and robustness of the system.

Introduction

With rapid technological advancements, cities worldwide, for example Dubai, are transitioning into smart cities that leverage technology to connect and upgrade existing infrastructures such as traffic systems, utilities, and public safety networks. These transformations have been made possible through the integration of IoT (Internet of Things) devices and networks. These networks of interconnected devices assist in performing and automating tasks, as well as provide insights without the need of human intervention.

Within the broader concept of smart cities lies smart homes and smart healthcare, which have gained significant traction as key applications for IoT devices, bringing automation and security to their use in private residences or hospitals. These IoT devices have enhanced the quality of life for the elderly community by providing access to healthcare solutions that aid in their safety, improve health monitoring, and support independent living.

The rapid integration of IoT devices in elderly care has raised cybersecurity concerns that need to be addressed, ranging from data breaches and unauthorized access to even potential physical harm due to device compromise. The security solutions for securing IoT networks are usually designed for generalised networks, where the contextual usage of the sensors and devices are not taken into consideration. Context-awareness in IoT occurs when sensors and devices in the system cope with changes in location, time of use or number of devices in use or network configurations (Sylla 2021). User behaviour can also have an impact on context-awareness. Context-aware security in IoT aims to emphasise user-centric methods to implement, for example, access control models, improve privacy and trust. The contribution of this paper is to assess the vulnerabilities in IoT devices and sensors, followed by the design and implementation of a novel technical context-aware IoT security framework within a smart home environment for elderly care.

Background

The older someone gets, the more care they might need to continue performing their daily activities. Technological progress has led to enhancements in the quality of life, allowing the elderly to take care of themselves independently and extend their lifespan by reducing the risks that are posed to their well-being in ways previously unimaginable (Panigrahi 2019). The devices associated with the technological progress need security. A pacemaker, for example, is a device attached to the patient that helps regulate their heartbeat. It only sends data out to other receivers, and its user depends on it to live a comfortable life; however, cyberattacks can occur that would put their life in jeopardy. By exploiting a vulnerability, such as intercepting and changing the recorded data, attackers could endanger the pacemaker's user by preventing the information from reaching its intended destination, i.e., the hospital (Chacko 2018). This example considers an attack vector for a healthcare-oriented IoT device, also known as Medical Internet of Things (MIoT) device.

"Smart homes in healthcare" means two things in the context of this paper: the smart residence of an elderly patient and a modern hospital. The smart residence would include IoT devices that function by capturing, storing and sometimes transmit data. In a hospital setting, several IoT devices

are at play, with the majority serving as a "crutch" for the patient's well-being, such as wearable sensors, ambient sensors, and more (Uslu 2020).

Selection of IoT devices in Elderly Care

There is a plethora of healthcare-oriented IoT devices when it comes to smart homes and in this paper the focus is on:

Wearables and Health Monitoring Devices: These devices go from smart watches that track blood pressure and heart rate to others that are implanted into the body such as pacemakers, ICDs (Implantable Cardioverter-Defibrillator), insulin pumps, and defibrillators. These devices are carried by the patient most of the time and provide continuous monitoring of the user's state. (Chacko 2018).

Sensors and Alerts Systems: These are devices that are external to the user. An example of these devices are fall detection, motion detection, and abnormal activity detection devices, which fall under the IoT infrastructure of a "smart home" (Mocrii 2018) but are connected to a medical backend (Chacko 2018).

Data Transmission in IoT

An IoT device usually collects data using built-in sensors, processes or stores it locally (if necessary), and then transmits it to a central server or cloud service for analysis or storage (Wheelus 2020). For example, a pacemaker has a sensor on the patient that continuously monitors them and records changes at set intervals. Some devices transmit this data through the internet using an access point (i.e. a router) to a server, while others send it to nearby medical devices using Bluetooth for processing (Chacko 2018). Some devices even perform micro-processing prior to sending data to reduce its noise (Selvaraj 2019).

When data is ready for transmission, it must adhere to a transmission protocol, which may include: **BLE** (Bluetooth Low Energy), **Zigbee**, **MQTT** (Message Queuing Telemetry Transport), **CoAP** (Constrained Application Protocol),**LoRaWAN** (Long Range Wide Area Network).

Extremely low powered devices tend not to encrypt data during transmission or use inadequate authentication mechanisms to save as much power as possible, thus making it possible for an attacker using a network sniffer tool to intercept and read the data, and potentially alter its sensitive information or use man-in-the-middle (MITM) attacks (Meneghello 2019). These types of risks could also become a gateway for the attacker into the network by leveraging a vulnerable MIoT device in a hospital and using it to further increase the attack scope and threat (Elhosney 2018). Many MIoT devices use low-frequency transmission protocols for energy efficiency but do not always employ secure protocols like TLS on top of them (Mocrii 2018). Data replay attacks, where an attacker captures legitimate data and replays it to the server to simulate real-time activity, can cause significant errors by disrupting accurate monitoring.

Data Storage in IoT

Data at rest, whether in the IoT devices or on the server, are still under threat due to specific constraints. For example, IoT devices would likely use the protocols mentioned in the previous section, as they typically do not store large amounts of data for extended periods of time. In comparison, servers use protocols like HL7 (Health Level 7) and FHIR (Fast Healthcare Interoperability Resources) to meet industry standards.

Additional practices for securing data at rest mentioned by Lee (2021) include encrypting the data, implementing firewalls, regular patching and monitoring systems for abnormal database activity, anonymizing or pseudonymizing health data in storage to protect personally identifiable information (PII), even if accessed without authorization, using redundancy and regular data backups to prevent data loss from accidental deletion, hardware failure, or cyberattacks like ransomware, and maintaining detailed access logs that track all data interactions and modifications.

Data should always comply with regulations such as HIPAA (Health Insurance Portability and Accountability Act), GDPR (General Data Protection Regulation), and HITECH (The Health Information Technology for Economic and Clinical Health), which outline various strict requirements for data storage security in healthcare.

Lack of encryption or inadequate access control and authentication mechanisms can lead to internal misuse or external breaches that can be used to extract confidential data (Meneghello 2019). Many IoT devices rely on the cloud storage because of its scalability and to gain access to large datasets. However, cloud storage solutions can be vulnerable to attacks if they lack strong encryption, multi-factor authentication, and/or secure API configurations (Selvaraj 2019). Furthermore, weak firmware and software security on IoT devices allow attackers with physical access to an IoT device, such as a health monitor, to bypass software restrictions and extract stored data, potentially compromising patient privacy. These attacks are particularly relevant if the device temporarily stores patient data before transmission to the server (Meneghello 2019).

Context-Aware Security

Tailoring security mechanisms based on real-time contextual information is the foundation of Context-Aware Security. This contextual information includes user location, behavior, device status, and environmental conditions. This type of security aims to create systems that not only collect data but also process and interpret it based on its context, dispatching appropriate security responses (de Matos 2020). Context-Aware IoT security uses the same core principles. Additionally, Sylla (2021) elaborated on how dynamic context is, highlighting its ability to range from physical and network conditions to user-specific risk profiles and how it can be integrated into security protocols to mitigate emerging threats in real time.

The selected context for this paper is IoT devices that aid in elderly care within a smart home environment. The IoT security framework is built using this context as the container. This approach enables the system to shift from a "one size fits all" security model to one that is sensitive to the nuances of the typical environments and user conditions, which is the essence of Context-Aware Security.

IoT Security Frameworks

There is a set of cybersecurity practices aimed at addressing security challenges, protecting data, and ensuring the reliable operation of interconnected devices in any environment. The collection of these practices forms a security framework (Lee 2021). An IoT security framework is built on five main pillars:

- Device Authentication and Authorization
- Secure Communication Protocols
- Monitoring and Intrusion Detection
- Data Integrity and Privacy
- Incident Response and Recovery

An example is the IoT security framework developed by IOT Security Foundation (IoTSF) known as the IoTSF IoT Security Assurance Framework, which details best practices and risk assessment guidelines (IoTSF 2021).

Related Work

The work by Lee (2021) evaluated their proposed model, termed the Layered Cloud-Edge-IoT Model, against existing IoT models like the generic and stretched IoT models. Their main talk points were about the privacy, security, and data protection of existing infrastructures to assess the effectiveness of their proposed infrastructure. Their findings showcased impressive results, achieving a security efficiency of 94%, compared to 82% for the generic model and 91% for the stretched model (Lee 2021). Similarly, the paper by Darwisha (2017) leveraged the UK HMG IS1 methodology, which is a structured approach to risk assessment that has been made with the restrictions that come with Medical IoTs in mind. They categorize threats into static and dynamic types in which static threats concern newly introduced devices, and dynamic threats affect existing devices when new equipment is added. To demonstrate, they used a case study based on the Technology Integrated Health Management (TIHM) system for dementia care that allows the team to manage evolving threats systematically, ensuring that data integrity, privacy, and system resilience are maintained as the network expands (Darwisha 2017).

Lee (2021)'s model showcases it by layering cloud and edge processing to minimize centralized vulnerabilities, while Darwisha (2017) introduce composability to respond to dynamic threats. Lee (2021)'s approach prioritized structure data processing to improve security efficiency which may lack adaptability for all IoT sectors in contrary to the theme. On the other hand, Darwisha (2017) creates a highly adaptable model by focusing on medical IoTs whose complexity may lower its performance and usability in sectors requiring real-time responses.

Methodology

The chosen methodology for this project is centered on simulating an IoT environment within its respective context. The approach is justified by the need for a controlled environment that mimics that of a real-world environment to run repeated and precise tests on the IoT devices and protocols rigorously under realistic conditions. The methodology is built on simulating two environments with similar characteristics, a smart home designed for the elderly and a smart hospital. Key tools are explored and selected based on their compatibility with Kali Linux, and their ability to simulate complex IoT networks. The tools explored are GNS3, VirtualBox, Cooja, Cisco Packet Tracer, IoTIFY, NS-3, and MATLAB, each offering unique advantages in network configuration, scalability, and integration with Kali Linux. The use case of the methodology was explored using an example scenario illustrating a Man-in-the-Middle (MITM) attack simulation on the smart home's vital sign monitor, showed how insecure data transmission could expose sensitive health information.

Attack Scenarios

Using the task scenario mentioned above, attacks have been conducted from an attacking Kali Linux machine. Each attack was specific to a certain protocol. The immediate result of each attack is mentioned with its respective attack, but a deeper analysis of each attack and its results can be found in the Results and Analysis section.

Attack 1: Gaining unauthorized access to the MQTT server.

The attacker first uses a technique like ping sweep, which involves sending pings to an IP range to identify which hosts exist on the network. Then, with the recorded list of active hosts within an IP range, the attacker can use Nmap, a free and open-source utility for network discovery and security auditing. Knowing that the default port number for MQTT communications is 1883, the attacker can use this information to check the activity of each port within the given range (Dinculeană 2019). If successful, the attacker can find the MQTT server and proceed with more direct attacks.

Attack 2: Causing a Denial of Service (DOS) on the MQTT server.

Using hping3, which is a tool used to test firewall rules and test network performance using different protocols, packets can be flooded to the port of the server. Its effects can be observed from other machines. For example, when the medical dispenser tries to connect to the Health Hubs server, its connection fails due to a time out (Alaiz-Moreton 2019).

Attack 3: Sending spoofed messages to the CoAP Server by utilizing's a Man-In-The-Middle (MITM) attack.

The attacker aims to use the fall detection sensor's address and send fake messages to the CoAP server, i.e., the Health Hub, causing it to send an emergency request to the relevant authorities. The attacker first conducts a Man-In-The-Middle attack by sending spoofed ARP (Address Resolution Protocol) packets by mapping their MAC address to the IP address of the CoAP server using the tool arpspoof.

After the success of the MITM Attack, the attacker uses Wireshark to sniff the packets being transmitted. Since the CoAP packets were unencrypted, the attacker could easily see the structure of the fall detected packet. The attacker now forges a packet mimicking the format of the previous one and sends it to the CoAP server.

Attack 4: Replaying a message by utilizing a Man-In-The-Middle (MITM) attack.

If building a spoofed packet failed, the attacker utilizes the previously captured packet to stage a replay attack, which is simple when the attacker replays a previously transmitted message after recording it. Using the Scapy library, a small Python program can be used to reconstruct and send the packet. The attacker has the advantage by reusing packets instead of crafting one from scratch.

Attack 5: Jamming LoRaWAN communication generated by the Wearable Emergency Button.

LoRaWAN communication runs on low-powered radio frequencies, typically ranging from 433 MHz to 915 MHz. The HackRF One (Öst 2018) could be used, but due to the lack of hardware, the wearable emergency button uses the paho.mqtt library instead. The equivalent of this jamming attack would be a DoS attack, covered in Attack 2.

Attack 6: Attacks on encrypted BLE payload.

To simulate BLE communication, specialized hardware is needed. Additionally, GNS3 does not support BLE communications, so the wearable heart rate monitor and the Bluetooth gateway use MQTT for communication. Their communication patterns mimic those of actual BLE communication. The devices communicating via BLE are connected using a "BluetoothSwitch", which acts as a device that can simulate the functionality of Ubertooth One (Nagrare 2023), enabling the attacker to sniff and analyze the traffic generated by the two BLE devices.

Attack 7: Disabling the Motion Sensor

In this attack, the attacker aims to disable the motion sensor by capturing its traffic and modifying its payload to only send "0"s, indicating that no motion is currently being recorded. The attacker first impersonates the Zigbee Hub by creating a proxy device that sits between both devices, achieving a MITM attack. Then, with the captured traffic, the attacker can modify it using a tool like ZBOSS, a Zigbee protocol stack that allows the development of Zigbee applications, and forward the modified packets to the Zigbee Hub, which will later reach the MQTT Server in the Health Hub. Similar to the MITM attack in Attack 4.

Simulation Experimentation

A simulated environment was set up to mimic a smart home populated with relevant IoT devices. The devices interactions with the environment were recorded for a period of 24 hours. This served as base the normal activity when the system is secure and excerpts from this recorded time was used during the analysis. The differences in activity are described in the results and analysis section. This approach is supported by previous studies Darwisha (2017) and Chacko (2018). This methodology has facilitated a structured analysis of the vulnerabilities while also contributing insights to the broader IoT healthcare security literature.

The Environment

The Smart Home Simulation was created in GNS3. Multiple virtual machines (using VMware Workstation Pro) were integrated into the created topology, simulating the IoT devices of this environment. Each device in this topology is connected to a virtual machine that uses Ubuntu Live Server as its OS. The reason for this choice is that it is a lightweight OS designed to run in the background. Each device is made to mimic the functionalities of an IoT device.

The choices of the devices varies in the protocols they use to communicate. This has been done to further cover multiple protocols and to build an environment that is as close as possible to its real-life counterpart. For example, a Bluetooth device would need a Bluetooth gateway that listens to its communication and transforms it from Bluetooth to the protocol used by the Health Hub. A similar device is also used for Zigbee communications. Table 1 provides a list of each device, the protocol they communicate in, and a description of their functionalities.

Device Configuration

Each device was configured to use a static IP address to make the simulation predictable and replicable. Each simulated device was then made to mimic the communication of its respective IoT device, which was achieved by writing python scripts using the paho.mqtt.client library, if the device was to communicate with the Health Hub (Dinculeană 2019). The Bluetooth gateway and the Zigbee Hub work in a way similar to the Health Hub, which starts an MQTT server to which the devices connect (Makwana 2017). These devices send their payloads into topics that are then placed in a database using InfluxDB. Ultimately, the Zigbee devices and the Bluetooth devices are using MQTT to communicate due to the limitations of the simulation tools, and this is a shortcoming of this study. However, to address this shortcoming, their communication patterns and behaviors have been replicated.

Task Scenario

An example scenario was orchestrated to enable the functionality of each device, which represents the normal activity of the environment. Therefore, any attack against the environment would have to disrupt and force it to deter from its normal activity for the attack to be considered as a successful attack (in most cases).

The scenario is as followed: The motion sensor and the wearable heartrate monitor are active throughout the scenario illustrating that the elderly person is within limit of a motion sensor and is wearing their heart rate monitor. Every 30 mins, the medical dispenser releases medications simulating the need for the elderly person to take their medications. After one hour, the fall detection sensor goes off detecting a fall and in about 10 seconds, the wearable emergency button also goes off alerting that the elderly person is now in danger and requires immediate assistance. This scenario puts all devices to use and showcases their communications, data storage, and data processing. An attack in any aspect of this scenario should lead to unfortunate events.

Proposed IoT Security Framework

The proposed framework aims to adhere to the 5 pillars of IoT security framework by adopting robust standards that the devices and the network have to adhere to:

Table 1: List of the Devices

Device	Description	Protocol
HealthHub	It is the main device that stores the data generated by every other device in a dataset. It processes and forwards data across the internet in the event of an emergency.	MQTT/CoAP
Fall detection Sensor	It sends an emergency signal to the Health Hub in the event of detecting a fall.	CoAP
Medical Dispenser	It sends a message to the Health Hub with the amount of dispensed pills.	MQTT
Wearable Emergency Button	It sends an emergency signal to the Health Hub.	LoRaWAN
Motion Sensor	It constantly sends 0 or 1 messages to the Zigbee Hub indicating whether motion has been detected.	Zigbee
Wearable Heart Monitor	It constantly sends the current heart rate of its user to the Bluetooth Gateway.	BLE
ZigbeeHub	Translates the Zigbee payload sent by the motion sensor into MQTT and forwards it to the Health Hub.	Zigbee/MQTT
Bluetooth Gateway	Translates the Bluetooth payload from the wearable heart monitor into MQTT and forwards it to the Health Hub.	BLE/MQTT

- Segment the network to isolate IoT devices from critical systems.
- Implement Role-Based Access Control (RBAC) to ensure that only authorized devices can interact with a limited number of devices they are permitted to communicate with, further isolating the network in case of an attack.
- Utilize multi-factor authentication (MFA) where applicable, especially for administrative access.
- Enforce the use of TLS 1.3, which is the latest version of TLS that includes all of the best security practices, for MQTT communication to prevent eavesdropping and replay attacks.
- Implement frequency hopping and cryptographic session keys for LoRaWAN to prevent jamming and eavesdropping.
- Ensure all Zigbee and BLE communications use AES-CCM, an encryption mode that allows authentication and encryption with dynamic key rotation.
- Deployment of intrusion detection (IDS) and prevention systems (IPS) tailored for IoT environments.
- Ensure that the data is encrypted and stored using secure encryption algorithms (e.g. AES-256).
- Ensure the utilization of the zero-trust architecture principles, ensuring that each access request is authenticated and verified, regardless of origin.
- Ensure the use of machine-learning (ML) based anomaly detection to identify suspicious activities.
- Define and implement incident response playbooks for handling detected threats tailored for your environment.

By adopting this proposed framework and its standards, changes can be made to the environment to mitigate and defend against the attacks mentioned in the previous section. A change in the topology and the environment has been performed as seen in Figure 1. Firewalls have been installed, and communications between IoT devices have been isolated to lessen the attack scope in case of an attack. Additionally, it can be observed that the adaptive nature of the framework helps it to better protect itself as long as it has the sufficient contextual-data.

All devices encrypt their payload using AES encryption, and all communications are done through TLS. Additionally, the MQTT server requires a username and password, adhering to the authentication requirements of the proposed framework.

The context-aware nature of the framework was developed to make it more adaptive. The road for a machine-learning-based anomaly detection system was paved, which would be unique to the individual user based on their data. To achieve this, a rule-based approach was considered as the initial step to provide sufficient security while enough data is being generated for the ML algorithm to reach a suitable accuracy rate in detecting anomalies. The role of the ML algorithm is to generate more rules and enhance the set rules in the framework. The rules are divided into two categories: **Security Rules**, which deal with securing the system from external and internal attacks, and the context-heavy **Emergency Rules**, which aim to ensure that emergency alerts are sent based on actual emergencies and reduce the number of false positives.

Both rule sets allow for more context-focused decisions. The data generated by the emergency rules are also used by the security rules to detect anomalies. Algorithms 1 and 2 showcase how the rule sets can be implemented. Using the simulation in Figure 1, this rule-based approach was implemented and tested in a 24-hour simulation that included the seven attack scenarios, as well as additional attack scenarios to securely check the efficiency of this approach. Each attack contributed to the collection of metrics specific to one of three cases:

- **Case 1 - Normal Behavior:** The data generated when every alert, triggered by the utilization of a rule (Emer-

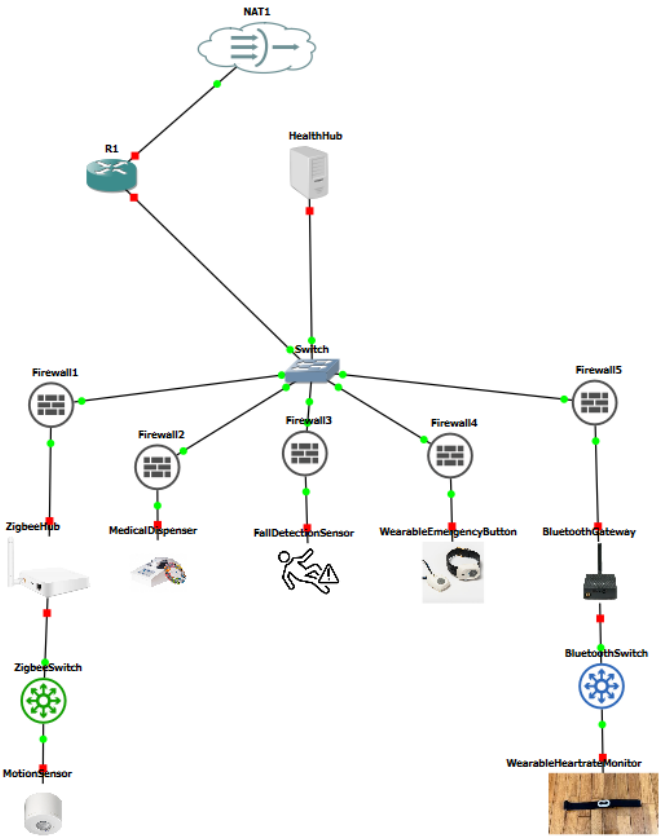

Figure 1: The updated topology after adopting the proposed security framework.

Algorithm 1: CheckSecurityRules

Input: MQTT message m
Global: `REGISTERED_IPS`, `MESSAGE_RATE`, `metrics`
Output: list of security alerts S

$S \leftarrow \{\}$
$ip \leftarrow m.source\_ip$
# Rule 1: isolate unregistered devices
if $ip \notin$ `REGISTERED_IPS` **then**
  $S \leftarrow$ `APPEND: "[SECURITY] Unknown device → Isolate"`
end if
# MORE RULES...
return S

Algorithm 2: CheckEmergencyRules

Input: MQTT message m
Output: list of emergency alerts E

$E \leftarrow \{\}$
# Rule 1: fall detection + abnormal heart rate
if $m.topic =$ `"fall_detection"` **then**
  $hr\_list \leftarrow FetchRecent(\texttt{"heartrate"}, last = 1\,min)$
  if $\texttt{mean}(hr\_list) < 40$ **or** $\texttt{mean}(hr\_list) > 120$ **then**
    $E \leftarrow$ `"APPEND: [EMERGENCY] Fall + Abnormal HR"`
  end if
end if
# MORE RULES...
return E

gency or Security), is a true positive.

- **Case 2 - Unaccounted User Behavior:** The data generated when there are scenarios for rules that were not accounted for, increasing the chances of false negatives and false positives.
- **Case 3 - Attack on Security:** This comprises cyber attacks. It includes the seven attack scenarios as well as additional ones.

Results and Analysis

This section aims to test the effectiveness of all attacks done in the simulation and evaluating them based on:

- Attack success rate.
- Data integrity check to analyze the data modification or corruption during transmission.
- Latency analysis for attacks that introduce time delays.
- False alert rate.

The effect of using the framework is then explained at the end of each attack subsection.

Attack 1 Figure 2 showcases the findings: out of the 100 IPs scanned, only 7 were active hosts, and of those 7, only 6 had port 1883 open for MQTT communications. Among them was the MQTT Server. The success rate of this attack without narrowing down the potential hosts was 0.4%, but after narrowing it to just 6 hosts, it rose to 97.647% per host. This attack was a passive attack aimed at setting up future attacks, which is why there are no implications on data integrity, latency, or a rise in false alerts.

The framework uses ”subnetting”, limiting the number of hosts that can be detected per the number of scanned IP addresses. Furthermore, all communications are conducted over TLS, which encrypts the traffic to and from the MQTT server, guarding it against sniffing and can protect its users from attacks that against the MQTT Server.

Attack 2 The success rate of this attack was measured by attempting to connect to the MQTT server from the medical dispenser at different times during the attack. A script was created that connects to the MQTT server, records the time, and disconnects, repeating this process for a total of 60 seconds. The normal behavior indicated that in 60 seconds, 15,154 connections could be made to the server, totaling an outstanding 252.57 connections per second. During the attack, in which the attacker transmitted 27,142,256 packets, only 1,223 successful connections to the MQTT server from the medical dispenser could be made, averaging 20.38 connections per second. Figure 3 showcases the results and distinguishes the patterns. At the start of the attack, it was easy to connect to the server; however, as the attack intensified,

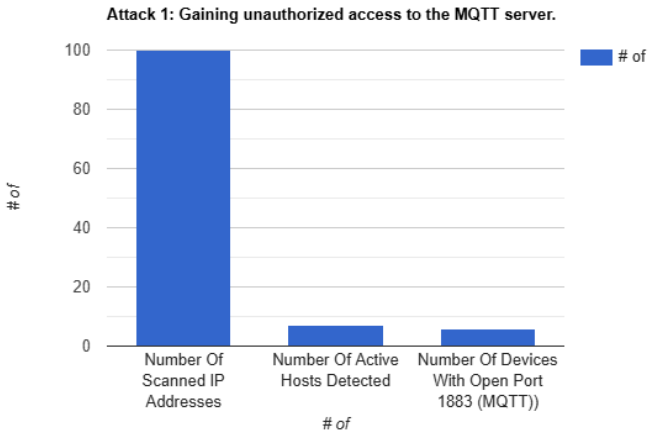

Figure 2: Attack 1 unauthorized access to the MQTT server.

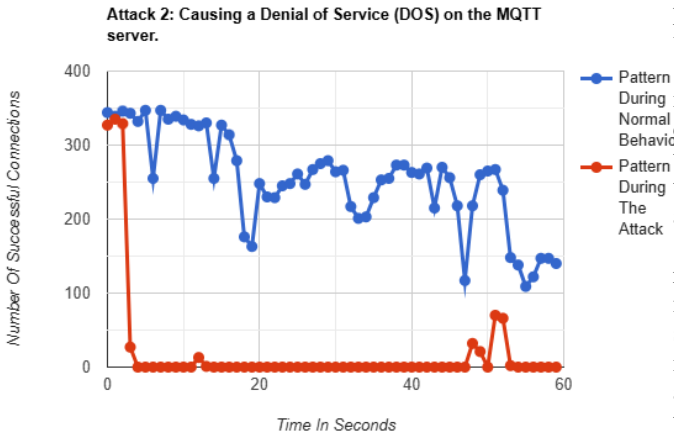

Figure 3: Attack 2 Causing a DoS on the MQTT server

the connection rate dropped to zero, except for some rare cases, but never exceeded a hundred connections per second. There was no direct attack on data integrity; however, no data could be transmitted if a connection to the MQTT server was not established. Additionally, this attack did not introduce latency issues or raise any false alerts. With the framework, the network can be configured to stop the spam of packets. Additionally, the MQTT server can be configured to detect a DoS attack and block the address from which the attack is coming.

Attack 3 While the attack was not a success, the MITM attack used to achieve the attack's main goal was successful, enabling grounds for passive approaches like eavesdropping on the packets or more active approaches like dropping all packets and not forwarding them to their destination. It did not introduce significant latency issues to be included and analyzed, nor any false alerts. With the Framework, similar practices to those used in attack 1 can also enhance overall security. Additionally, MITM is a common attack that the

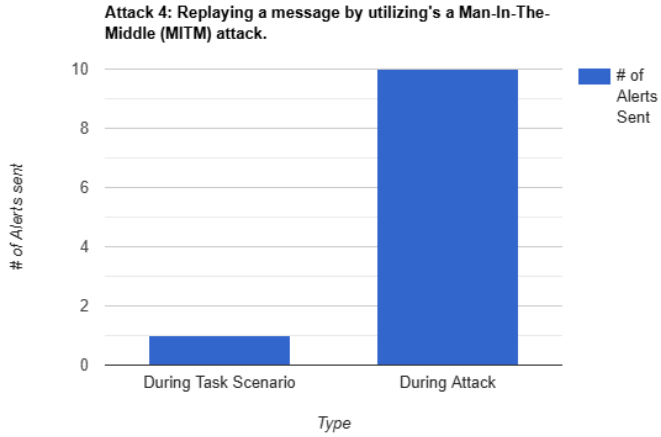

Figure 4: Attack 4 Replacing a message bu using MiTM

proposed framework already accounts for and can be prevented using strong authentication methods.

Attack 4 The attacker utilized the same approach used in the attack 3, the MITM approach, and captured a packet containing the necessary information from the Fall Detection Sensor to trigger an alert. The attacker then replayed the packet whenever they wanted to trigger the alert. The attack was a success, an alert was sent outside the network.

Figure 4 showcases the difference observed in the environment during the task scenario, which is considered normal behavior, and during an attack. The number of alerts sent increased by 10 times the normal behavior, resulting in a false alert rate of 1000%. The attack success rate was also 100% because replaying a captured packet never failed to send the alert. No latency issues arose during this attack however the data integrity was compromised, considering that it was not original data but captured data being retransmitted in the network. With the Framework, it is similar to attack 3, and the prevention methods used by attack 3 apply here as well.

Attack 5 This attack aims to jam the communication generated by the wearable emergency button using the tool "HackRF One". Using the tool, it is possible to transmit interference signals that will disrupt the communication. The equivalent of this attack in MQTT is a DoS attack, which has been covered before in Attack 2 and the analysis of Attack 2. The proposed framework protects against jamming. It enforces the need to implement frequency hopping and cryptographic session keys for LoRaWAN to prevent jamming and eavesdropping. These two methods will not stop the jamming attacks but will mitigate it to the best of their abilities lowering its success rate significantly.

Attack 6 The results demonstrated that due to the nature of the encryption used, which was Fernet encryption employing AES-128, it was impossible to break the encryption and read the actual payload. The attacker could only either stop forwarding the packets from the Wearable Heart Rate Monitor to the Health Hub, delay them, or remain passive by merely sniffing and recording the packets. Another pos-

sible attack is to stop forwarding packets which will prompt the Wearable Heart Rate Monitor to run for a few seconds before disconnecting with a "Timeout" error. The success rate of the "blocking" attack is 100%. All the attacker has to do is stop forwarding the packets for it to be successful. This introduces latency issues as the "attacked" device now faces no viable means to connect to the server. This attack does not produce false errors nor does it affect existing data. However, data integrity is compromised as new data that should be recorded are not being accounted for.

Attack 7 The success rate of this attack was proportional to the number of times it was conducted. However, issues arose in duplicating the packet. The final solution was to swap the payload with another payload and resend the packet, increasing the success chance to a 100%. Intercepting and changing the packet's payload introduced latency issues; however, they were not significant enough to warrant further investigation. This attack was a major threat to data integrity as it altered the packet's payload during transmission before it reached its intended destination. Additionally, this attack targeted the false alert rate by removing true alerts, compromising the integrity and safety of the system and its users.

Using the framework for Attack 6 and 7, with the addition of strong authentication practices, no attack could be used on BLE and Zigbee communications as it would require the attacker to bypass the authentication mechanism put in place. The timeout attack and the "disabling the motion sensor" attack discussed previously, works if the attacker authenticates and listens to the traffic, which will be prevented using the proposed framework.

For the to the 24-hour simulation, the metrics collected were:

- Total Messages
- Emergency alerts count
- Security alerts count
- False positives / false negatives / true positives / true negatives
- Database write count

Each case generated different results. Case 1 had a 100% security rate, which makes it a good base template. The normal message rate was recorded for each device.

- Medical Dispenser: 1 message /30 mins
- Fall Detection Sensor: 1 message / 3.7 hrs
- Wearable Heartrate Monitor: 1 message /10 secs
- Motion Sensor: 1 message /10 secs
- Wearable Emergency Button: When clicked (Only for emergencies)

Case 2 showcased how some activities performed by the user can lead to false emergency alerts. E.g., the user triggered the fall detection sensor by doing a series of push-ups, due to their close proximity to the floor. They also had a high heart rate since they were exercising. This scenario triggers the Emergency Alert, which makes it a false positive. No security alerts were triggered, meaning that the database update occurred due to the lack of security issues. The rule-based approach lacks the foresight needed to mitigate false positives in this way, which is why the ML approach, when sufficient data is collected, will tackle these gaps in security.

Case 3 generated the same findings as mentioned for the seven attack scenarios.

Discussion

The findings suggest that conventional, static security measures are insufficient for securing dynamic and changing environments. They require a security system that can adapt to contextual changes, such as the proposed framework, which has practical benefits when integrated into existing IoT environments. It demonstrates a reduction in the attack surface and improved data integrity under simulated attack conditions. These outcomes suggest that static approaches, like those of Darwisha (2017) and Chacko (2018), are not as enticing as the context-sensitive approaches showcased in this dissertation. The proposed framework offers better protection for vulnerable populations, such as elderly patients relying on smart home healthcare devices.

Context-Aware Security not only focuses on data collection but also includes the ability to modulate security responses automatically, and enhances traditional security frameworks that consider only static environments. Our findings align with models such as Lee (2021)'s model, which advocates for distributed and adaptive security strategies, as well as with composable security frameworks proposed in other recent studies.

Conclusion

This paper addressed the security vulnerabilities inherent in IoT devices deployed for elderly care in smart home environments. The main achievements include the identification of several critical vulnerabilities in data transmission protocols, highlighting the need for robust encryption, adaptive authentication, and network segmentation to safeguard sensitive healthcare data. The creation of a Context-Aware IoT Security Framework that adjusts to the context of the given environment, specifically a smart home for elderly care, goes beyond traditional, static measures. It successfully reduces the risk of unauthorized access, data manipulation, and network-based attacks. Additionally, with a focus on elderly care within smart home environments, this paper aligns with real-world needs by proposing a framework that can potentially improve patient safety and ensure compliance with regulatory standards like GDPR and HIPAA.

Future work includes the framework to be deployed and tested in real-world environments. Additionally, incorporating hardware that supports communication protocols such as BLE, Zigbee, and LoRaWAN would more accurately replicate real-world IoT scenarios. Finally, adding more robust trust management mechanisms (e.g. Blockchain) would enhance the framework's security.

References

Alaiz-Moreton, H. 2019. Multiclass Classification Procedure for Detecting Attacks on MQTT-IoT Protocol. 2019: 6516253.

Chacko, A. 2018. Security and Privacy Issues with IoT in Healthcare. *EAI Endorsed Transactions on Pervasive Health and Technology*, 4: 8182–8201.

Darwisha, S. 2017. Towards Composable Threat Assessment for Medical IoT (MIoT). *Procedia Computer Science*, 113: 627–632.

de Matos, E. 2020. Context information sharing for the Internet of Things: A survey. 166: 106988.

Dinculeană, D. 2019. Vulnerabilities and Limitations of MQTT Protocol Used between IoT Devices. *Applied Sciences. Appl. Sci.*, 9: 848.

Elhosney, M. 2018. Secure Medical Data Transmission Model for IoT-Based Healthcare Systems. 6: 20596–20608.

IoTSF. 2021. *IoT Security Assurance Framework*. IoT Security Foundation.

Lee, C. C. K. 2021. Improving Internet Privacy, Data Protection and Security Concerns. *International Journal of Technology, Innovation and Management (IJTIM)*, 1: 18–33.

Makwana, A. 2017. A SURVEY ON MQTT: A PROTOCOL OF INTERNET OF THINGS(IOT). In *ResearchGate*.

Meneghello, F. 2019. IoT: Internet of Threats? A survey of practical security vulnerabilities in real IoT devices. 6: 8182–8201.

Mocrii, D. 2018. IoT-based smart homes: A review of system architecture, software, communications, privacy and security. *Internet of Things*, 1-2: 8182–8201.

Nagrare, T. 2023. BLE Protocol in IoT Devices and Smart Wearable Devices: Security and Privacy Threats. *Electronics and Telecommunication Engineering Department, VJTI, Mumbai, India*.

Panigrahi, C. R., ed. 2019. *Progress in Advanced Computing and Intelligent Engineering*. springer.

Selvaraj, S. 2019. Challenges and opportunities in IoT healthcare systems: a systematic review. *Springer Nature Switzerland AG 2019*, 2: 139.

Sylla, T. 2021. Context-Aware Security in the Internet of Things: A survey. *HAL Open science. HAL Id: hal-03184941*, 14: 1.

Uslu, B. C. 2020. Analysis of factors affecting IoT-based smart hospital design. *Journal of Cloud Computing*, 9: 67.

Wheelus, C. 2020. IoT Network Security: Threats, Risks, and a Data-Driven Defense Framework. *Cyber Security and Privacy in IoT*, 1: 259–285.

Öst, A. 2018. Evaluating LoRa and WiFi Jamming. *MID SWEDEN UNIVERSITY - Department of Information Systems and Technology*.