An analysis of the orchestration, maintenance and security of IoT devices.

Charles Hamerston-Budgen
UP2040477

BSc Computer Science
Project Code: PJR40

Supervisor: Dr. Gail Ollis

A dissertation submitted in partial fulfilment
of the requirements for the degree of
Bachelor of Science
of the
University of Portsmouth.

May 2024

Word count: 10,936

Please tick

☑	I give permission for my project to be published in the University library and/or be made available to other students as examples of previous work. (optional).
☑	I confirm that I have read and understood the University Rules in respect of plagiarism and student misconduct.
☑	I declare that this work is entirely my own. Each quotation or contribution cited from other work is fully referenced.

Date: 03/05/2024

Declaration

No portion of the work contained in this document has been submitted in support of an application for a degree or qualification of this or any other university or institution of learning. All verbatim extracts have been distinguished by quotation marks, and all sources of information have been specifically acknowledged.

Date: 2023

Abstract

IoT is utilised in a very wide field of applications ranging from smart home sensors to medical pacemakers, evolving at a very fast pace. Due to the usage and recent growth of IoT, the risk for consumers and organisations has increased, resulting in a growth of cyber-attacks, rising from 32 million in 2018 to 112 million in 2022 (Statista, 2023). Due to a wide variety of reasons, such as complexity or liability, security within IoT has been left behind, resulting in becoming the consumers' responsibility and risk.

This paper explores the use of IoT at both an enterprise level and a standard consumer alongside evaluating how IoT devices can be deployed, managed and maintained. Concurrently a survey was undertaken to capture usage and concerns around IoT as well as any mitigations which have been set up.

The results from the survey which involved roughly 50 participants, both technical and non-technical, showed that on a scale of 1-10 (no concern to high concern), the average score for concern was around 5. The most common theme was privacy, what data left the users' network and how it was protected. Most participants noted that they are selective with the devices they used, e.g. without a camera or microphones regardless of brand mainly due to concerns around privacy, as explored in a follow-up question.

Furthermore, due to the common themes, this project goes on to research the security of an IoT device from a non-network-based perspective, focusing on the hardware and firmware of the device. It was discovered that the firmware from the device could be dumped, decrypted, and dissected to extract sensitive information.

The recommendations in this paper are focused towards mitigating the methods in this paper, by implementing secure crypto processors such as a Trusted Platform Module (TPM) to implement better security practices. Alongside this, a few key improvements are mentioned to improve practices without replacing hardware.

Another mitigation which should be implemented alongside the aforementioned mitigations is the implementation of network segmentation to improve security at a consumer level.

Acknowledgements

I’d like to thank my project supervisor Dr. Gail Ollis for her continued support throughout the entire project.

I’d like to thank my family and friends for their support and encouragement throughout the entire project.

Consent to Share
I consent for this project to be archived by the University Library and potentially used as an example project for future students.

Table of Contents

Table of Figures

Table of Tables

1. Introduction

1.1. Project Context

The use of IoT has grown exponentially, according to a Statista report in 2023 (Statista, n.d.). Figure 1, shows the number of connected IoT devices worldwide, roughly 7.7 billion IoT devices as of 2019 which has increased to 15 billion as of 2023.

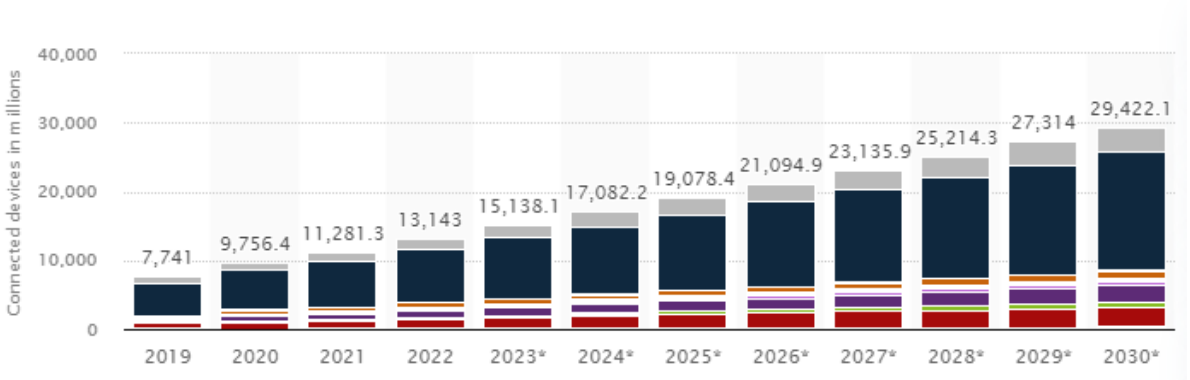


Figure 1: number of connected IoT devices (Statista, n.d.)

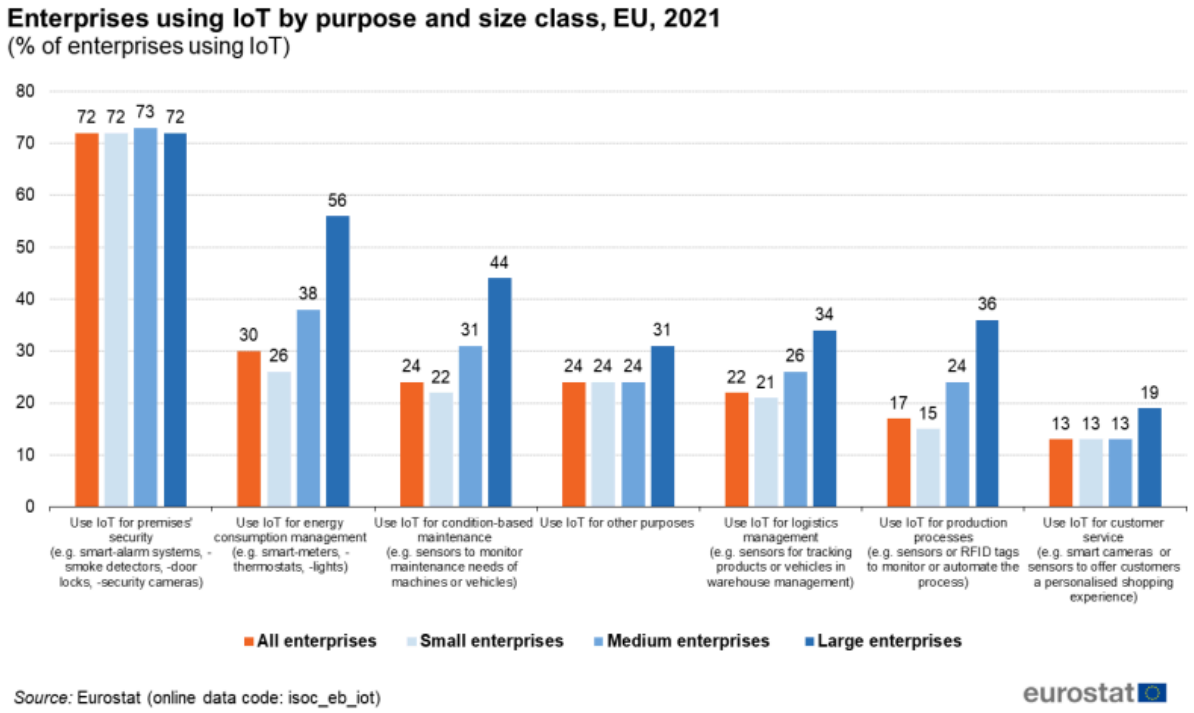


Figure 2: Enterprises Using IoT Devices by purpose (eurostat, 2022)

IoT is used in a wide variety of applications as shown in Figure 2, ranging from the physical security of property to customer service. The ability to utilise low-powered

devices to track a wide variety of parameters has allowed businesses to make informed business decisions utilising a high granularity of data.

Cybersecurity plays a vital role within IoT, from dependence enterprises have on the IoT systems and raw data to the regular consumers' privacy. Utilising 3rd party products may introduce new attack surfaces.

One of the most famous examples, outlined in a SecurityWeek Journal in 2017, was the St. Judes PaceMakers. A vulnerability was discovered within pacemakers, with the ability to crash the devices or drain the batteries. Two separate Common Vulnerability and Exposures (CVE) namely, CVE-2017-12712 (NIST, 2018a) and CVE-2017-12716 (NIST, 2018b) outline the importance of encryption and security within IoT.

A separate CVE (CVE-2017-12716) explains that a separate pacemaker manufacturer transmitted unencrypted patient information via RF communication to other internet-connected units (NIST, 2018).

Another example of the risk of IoT is the Mirai botnet, which was developed in September 2016 to target a security expert. However, the attackers released the source code for the botnet, resulting in other malicious uses such as the Distributed Denial of Service (DDoS) attack on the Dyn provider in October 2016 (Cloudflare, n.d.).

In both cases, the devices were vulnerable due to poor implementation from the manufacturers, resulting in compromise and used with malicious intent.

1.2. Project Significance

The significance of this project stems from the increasing utilisation of IoT and the increased volume of cybersecurity and privacy concerns. The project will explore the security, maintenance and orchestration of these devices at enterprise and consumer levels helping to isolate issues within the IoT space.

Furthermore, the project will help to explain, and share awareness of the risks of utilising IoT devices both on a consumer level as well as at an enterprise scale. The significance of the project derives from improving IoT security for both manufacturers and consumers by helping to investigate the current and common pitfalls for devices with strict restrictions.

I plan to explore how IoT devices are managed by both the manufacturer and consumer, devices that are fully managed by manufacturers often result in restrictions for the consumer around custom applications for interfacing with devices.

1.3. Aims and Objectives

The key aims and objectives for the project are as follows;

- Research and analyse the key issues within deploying, maintaining, utilising and securing IoT devices.

IoT has evolved very quickly, with a wide range of applications with varying security concerns. It's important the methods and tooling used to manage IoT devices stay up to date, this allows enterprises to fully utilise IoT devices. As outlined in an article from Cloudflare, manufacturers have little to no incentive to invest in security for IoT devices, especially for low-powered devices (Cloudflare, n.d.).

The key objectives for the project are as follows;

1. Research IoT communication protocols; evaluating the use of different protocols and summarising the pros and cons of each.
2. Research ‘Home Hubs’ for self-hosted IoT solutions; evaluate the use of home hubs to define what options a consumer has available, alongside the pros and cons.
3. Research the management and maintenance concerns within small and large-scale deployments.
4. Research the deployment of IoT devices at both small and large scales; explore the different challenges which may be faced at scale compared to at the consumer level.
5. Research the security and privacy concerns within IoT; by reverse engineering an IoT device, I can evaluate the security and privacy concerns in a real scenario.

1.4. Project Constraints

One of the major challenges within my project is to ensure all tests, exploits, and other vulnerabilities do not interact with any networks or servers that I do not own. To fulfil objective 5 I will investigate a single IoT device. The differences in manufacturing standards and implementations introduce complexity to reverse engineer, requiring investigation and tools to be built from scratch.

1.5. Legal, Ethical and Professional Issues

Legal:

1. Data protection (GDPR) (United Kingdom Government, 2018).
2. In the case of proprietary devices, the solution(s) shouldn’t manipulate the communication to/from remote servers.
3. Adhere to not using any copyrighted, protected designs.
4. Adhere to the use of the software licences.

Ethical:

1. Adhering to any privacy in terms of user data.

2. Any data should be protected/anonymized.
3. Respecting intellectual property.
4. The devices could create e-waste as they are single-purpose devices.
5. Must comply with the University of Portsmouth ethics requirements.

Professional:

1. Any potential solutions/recommendations should follow the IEEE standards for secure transmissions (ISO, 2019).

Social:

1. This research will explore giving users back the ownership of their devices.

It's important to note, that any testing will occur strictly within a network environment I own, with only devices I own making sure nothing extends out of the network.

1.6. Report Structure

Chapter 1 - Introduction

- This chapter is dedicated towards outline the project aims, outcomes and general context.

Chapter 2 - Literature Review

- This chapter is dedicated to researching prior literature, providing more context towards the research questions.

Chapter 3 - Research

- This chapter is dedicated to further providing context around IoT attacks, as defined by the previous chapter.

Chapter 4 - Methodology

- This chapter is dedicated to a discussion of the approach and reasoning behind the data collection.

Chapter 5 - Data Analysis

- This chapter is dedicated to a discussion of the results and potential reasoning from the survey.

Chapter 6 - Discussion

- This chapter is dedicated to the actual investigation of an IoT device.

Chapter 7 - Results

- This chapter outlines the results from the discussion compared to the initial objectives.

Chapter 8 - Conclusion

- This chapter is dedicated towards the conclusions and recommendations from the discussion. I will also discuss how the outcomes link back to the original objectives.

2. Literature Review

2.1. Introduction

Throughout the literature review, I will explore the use of IoT from both the consumers, and enterprises, points of view. I plan to investigate the variety of communication protocols alongside the use cases and concerns with each. I also plan to investigate how IoT is managed, deployed and maintained through custom means such as open-source solutions to enterprise-focused solutions.
The main focus will be IoT devices which communicate and transmit over the internet, due to the rapid growth over the previous decade in this space (Statista, n.d.).

2.2. Hardware

There are two types of hardware which I will be exploring through this research, enterprise and domestic hardware.

2.2.1. Enterprise Hardware

The application may vary per business, however, enterprise deployments of IoT devices may need to be more resistant to environments which the standard consumer may not encounter, for example in high temperatures or hazardous environments. Another requirement of business IoT devices may be that the devices need to support high throughput, or additional infrastructure e.g. an edge network to keep devices connected 24/7. Enterprises also require hardware which is low-cost and scalable to many thousands in some cases. The IoT devices also require much stricter cybersecurity standards, as they may sometimes be deployed in critical e.g. power and nuclear applications.

2.2.2. Domestic Hardware

Home applications of IoT may include more general-purpose devices compared to enterprise applications e.g. light bulbs and heating systems. The application of home devices may be managed by a platform (also called a hub) for example, Google Home (Google, n.d.-a), or Apple HomeKit (Home App, n.d.), although proprietary systems there are a few open-source solutions such as Home Assistant (Home Assistant, n.d.) which will be explored later on.

2.3. Communication Methods

2.3.1. Wi-Fi

Wi-Fi is a very common communication method, Wi-Fi is very well suited for LAN-based applications due to the multiple standards which are supported (802.11ax, 802.11ac, 802.11n, etc.) (WiFi Alliance, n.d.). A disadvantage of Wi-Fi is due to the large range of communication protocols, power consumption is increased which becomes a limiting factor.

2.3.2. Advanced Message Queuing Protocol (AMQP)

The Advanced Message Queuing Protocol (AMQP) lives on the Application Layer of the OSI model, allowing for interoperability between clients and brokers. AMQP defines how the brokers should operate, as well as how the networking takes place. AMQP is used where reliability is of importance, as well as rapid delivery (Ivan Lee, 2024).

This protocol is useful where devices may fall offline, and need the ability to fetch data at a later time when a connection is available. The AMQP system distributes the messages to multiple brokers and queues, unlike other systems which use a first in first out (FIFO) approach. AMQP consists of 4 parts as shown in Figure 3 below;

- The Broker - An application which receives requests from the Producer.
- Message - A block of data.
- A Consumer - An application which receives data from a queue.
- The Producer - An application which pushes requests to the queue.

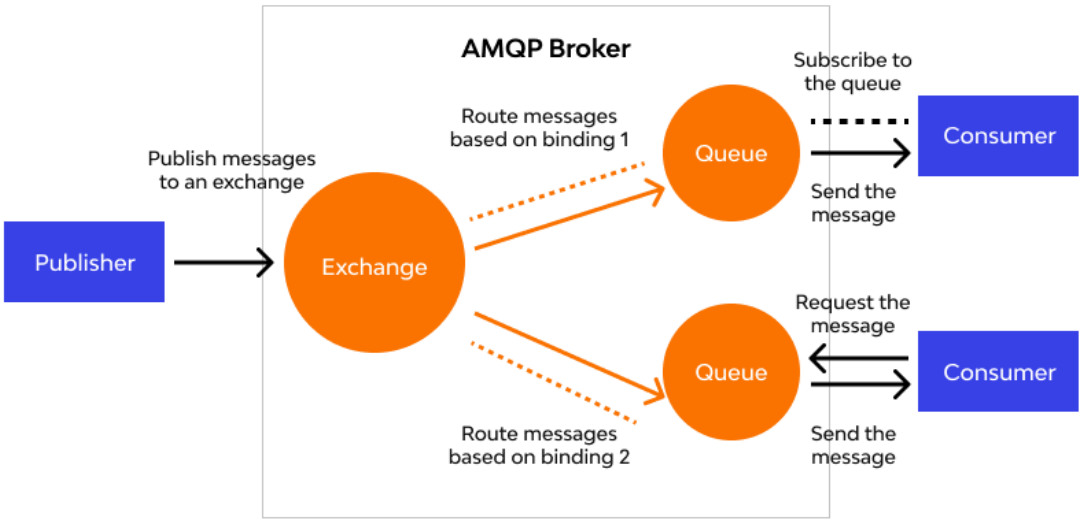


Figure 3: AMQP Protocol (Ivan Lee, 2024)

2.3.3. Bluetooth

Bluetooth provides a low power, low range but high bandwidth connection to devices. A parent-child model is used to connect the devices. The parent-child model is a

design which allows one device e.g. a smartphone to connect to multiple devices e.g. thermostat or a smart home. Bluetooth provides the capability to create a mesh network of devices, which allows for the network to be self-healing in the event a device loses power or gets disconnected (Marcel, 2024).

2.3.4. Cellular

The use of pre-existing Cellular networks within IoT communication provides a low-cost connectivity platform to build upon. Cellular networks also provide short and long-range connectivity with a larger bandwidth, which is suitable for a much broader range of applications (Wedd, 2020).

2.3.5. LoRa / LoRaWAN

The LoRa protocol is a wireless modulation technique composed of small radio waves, which are robust and resistant to disturbances and can be transmitted over long distances. LoRaWAN is built upon the LoRa protocol and utilises the Media Access Control (MAC) layer. LoRaWAN supports low-power but long-range transmissions, as described in the paper (Zourmand et al, 2019).

The spectrum which LoRaWAN operates within is an unlicensed range. LoRaWAN supports end-to-end encryption and can be updated over the air which is beneficial for IoT devices. As the devices can seamlessly switch between networks, it allows for a much wider range of applications compared to Wi-Fi such as space applications, airport tracking, and IoT-enabled farms (LoRa Alliance, n.d.).

2.3.6. XMPP

The Extensive Messaging and Presence Protocol (XMPP) allows for near-real-time messaging, conserving bandwidth to ensure fast processing of chronologically ordered messages. XMPP uses a client-server methodology as shown in Figure 4 below, passing small structured chunks of XML to other clients through intermediary servers. XMPP supports de-centralised hosting, which is suitable for domestic users (XMPP, 2024).

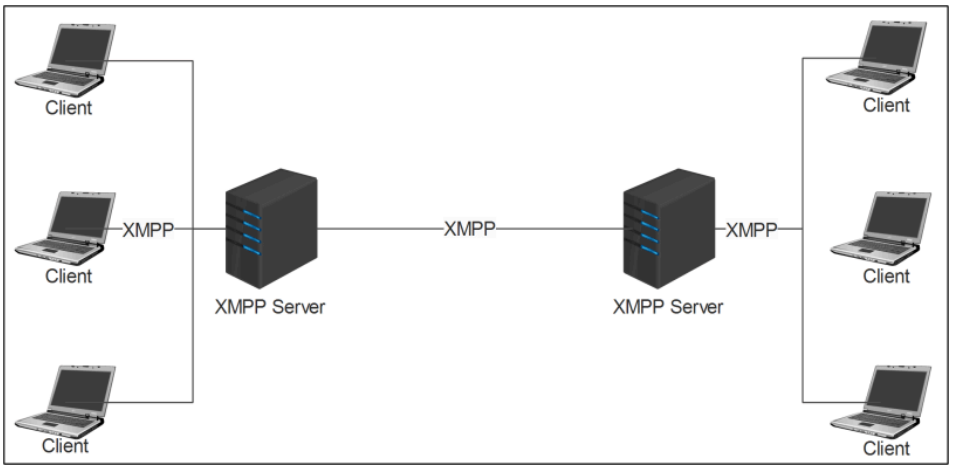


Figure 4: XMPP Network (Arslan, 2016)

2.3.7. Zigbee

Zigbee is a protocol in which smart devices can communicate over a Personal Area Network (PAN). Zigbee is a low-power option for deployed IoT devices but also includes 128-bit AES encryption along with Over The Air (OTA) updates. A large capacity is supported for Zigbee devices (Over 60,000 devices) on a single PAN. Zigbee is built upon the 802.15.4 standard utilising the physical layer of the OSI model, and Media Access Control (MAC) (Connectivity Standards Alliance, n.d.).

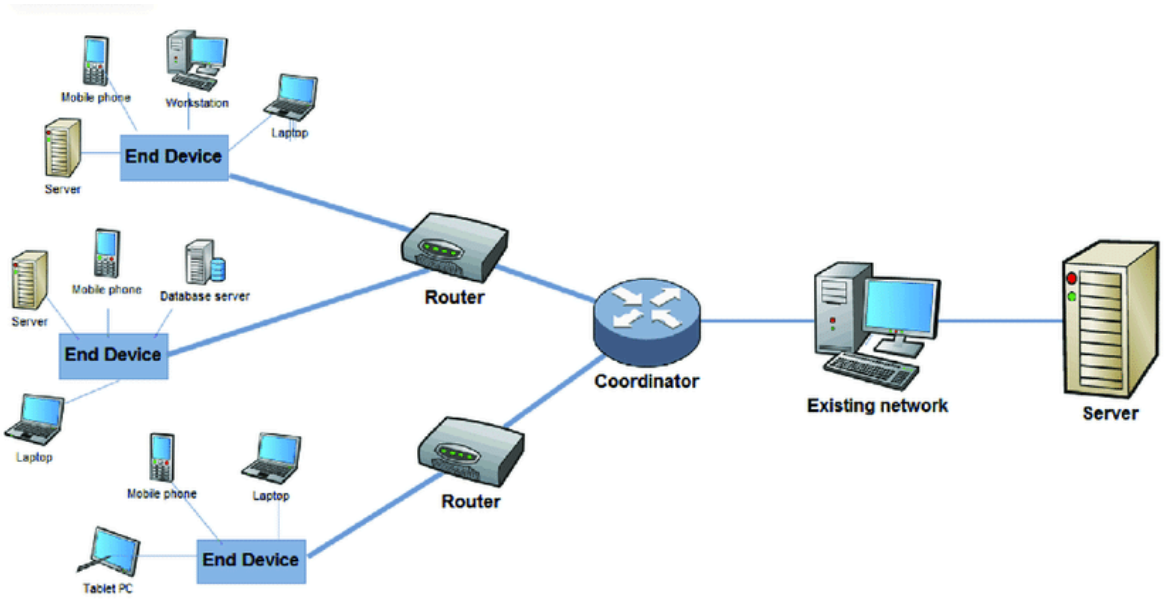


Figure 5: Zigbee Network (Eljiona Zanaj et al., 2021)

A typical ZigBee network will consist of at least 3 components as demonstrated in Figure 5.

- ZigBee Coordinator - A smart home controller, hub or gateway which manages the network itself including connections and security.
- ZigBee Router - A router is used to create the links between the devices, these are typically the devices themselves (light bulbs, plugs, etc) these devices are usually mains powered.
- ZigBee End Device - Basic devices which cannot route, only send or receive data.

Zigbee devices are usually only battery-powered such as a motion sensor.

Zigbee devices form a mesh topology, to provide low latency and high robustness, each node takes one of two possible forms;

1. Full Function Device - Can play all roles.
2. Reduced Function Device - Only an end device.

2.3.8. Z-Wave

Z-Wave is a low-cost and low-power protocol for connecting devices in LANs, the devices become small signal repeaters and a single network can support up to 250 devices. Z-Wave operates on the 868.42 MHz band to avoid any potential interference from Wi-Fi and Bluetooth which operate on the 2.4GHz band. A Z-Wave network consists of a few components:

- Controller - Compiles a routing table of all the devices.
    - Primary controller - Manages the details of the Z-Wave network
    - Secondary controller - Constantly maintains the routing tables.
- Node - This may contain network maps but can receive frames and respond.

Z-Wave has a short range of only 30 metres, which limits the suitable applications (Z-Wave Alliance, n.d.).

2.4. Types of Management

IoT device management is an important aspect of improving user experience (UX). Device management usually covers not only how a user interfaces with a device but also makes it easier to update firmware and change settings about the device.

2.4.1. Proprietary Solutions

SmartThings

SmartThings, managed by Samsung, provides a quick and easy platform for allowing consumers to update the firmware, control, and adjust their IoT products remotely through a server (*Home Automation with the SmartThings App | SmartThings*, n.d.).

Google Home

Google Home allows users to set up Google devices, such as cameras, speakers, and displays. Home also allows users to manage these devices, as well as

non-Google devices. Google's own devices manufactured after 2019 are audited externally by a 3rd party, the nccgroup. The results produced by the nccgroup are published online, this shows a level of responsibility from the manufacturer to protect devices (Google, n.d.-b).

Google also invests in vulnerability programs (*Google Bug Hunters*, n.d.) to enhance the security of their devices, alongside internal security teams.

2.4.2. Open-sourced Solutions

Portainer

Providing both enterprise and domestic versions, Portainer is a platform in which Docker/Swarm/Kubernetes environments can be deployed. Portainer removes the vendor lock-in for cloud-based IoT management.

Home Assistant

Home Assistant is an open-sourced tool designed and supported by a community to integrate IoT devices into local control to support user privacy.
Home Assistant also provides a smart hub called 'Home Assistant Green' supported by a mobile app, allowing non-technical users to connect to their IoT devices with ease. Support from the open-source community means Home Assistant is compatible with over 2,500 types of devices, making it one of the most popular options.

2.4.3. Cloud Solutions

There is a subset of solutions which require cloud-based connectivity or deployments to operate, some of the major cloud providers offer solutions to this e.g. Microsoft and Amazon.

Microsoft - Microsoft Azure IoT

Azure IoT is more focused towards enterprise customers and offers a managed dashboard, with a large ecosystem of connectivity. Azure IoT also offers solutions for securing communication between the chip and the cloud.

Amazon - AWS IoT

AWS IoT is designed to aid users in building, and managing IoT devices, alongside securing the communication to the cloud. A range of features are provided from access control to auditing to management to help organisations meet their needs.

2.5. Device Security

The importance of device security is increasing as we deploy more and more sensitive devices such as cameras, door controls, devices that hold payment info and more critical devices for human safety, such as smoke alarms and CO2 alarms. Homes are integrating more smart devices, ranging from smoke alarms to door and window alarms. Users also can create automation or write custom code to automate tasks, for example, by switching a light off when a person leaves the room automatically.

IoT device security is critical in modern deployments, security should always be an important consideration. Lack of security in devices could lead to critical infrastructure failure in industrial applications, as well as smaller events like data breaches for the standard consumer.

2.5.1. Encryption

One of the more common encryption methods, outlined in (Kaiyuan Yang et al., 2017) is AES. Due to its low complexity, it is suitable for IoT encryption.
Another security issue to consider is Random Number Generation (RNG). There are two types of RNG, Pseudo-random Number Generators (PRNGs) and True Random Number Generators (TRNGs). PRNGs are based on an initial seed and follow a pattern to produce numbers which are not cryptographically random. TRNGs use physical analogue inputs such as resistor values to generate randomness, however, this is much more expensive and requires more power which is not suitable for low-powered devices.

In regards to IoT, it's important to use TRNGs rather than PRNGs to ensure stronger data encryption standards, reducing the change of data decryption without the key.

2.6. Hardware

IoT devices, and more broadly, embedded systems, all share similar concepts in terms of architecture and manufacturing.
A complex IoT device, such as a hub may have a series of segmented blocks, for example;

- A CPU block
- A storage block
- A power block
- Sensor block(s)

- Networking block(s)
- Interface block(s)

In the next section, I will explore why each of these blocks is important to security.

Data within an IoT device is usually written to some form of memory usually stored within the CPU, or dedicated memory module. Without the use of cryptoprocessors, this information is usually unencrypted.

Due to the lack of a cryptoprocessor a wider variety of attacks can be executed. It may also be possible for an attacker to dump the contents of the memory to find out potential encryption keys, this poses a further issue if a company decides to use the same key across multiple products to reduce costs.

2.6.1. The CPU block

The CPU block is responsible for processing and controlling the flow of data between the other components. There can be either of two CPU types within a system, a microcontroller (MCU) or a microprocessor (MPU).
A normal computer will use either an x86 or an x64-based architecture, which has a much greater performance but is more costly and more power-intensive.

MIPS

Microprocessor without Interlocked Pipeline Stages (MIPS) is a 32/64 bit von Neumann architecture which is usually present in devices designed to complete a specific purpose.

RISC-V

RISC-V is an open-sourced architecture with 32/64 bits, based on the von Neumann architecture, commonly used in academia.

PIC

Designed to be low-cost and low-processing micro-controllers, which run 8-bit to 32-bits on a Harvard architecture.

2.6.2. The storage block

RAM

Random Access Memory (RAM) is a volatile but very fast storage solution. RAM is usually closely located to the CPU directly reducing the length of the buses, allowing for faster read/write speeds.

Long-term storage

Long-term memory, such as a Hard Disk Drive (HDD), or Solid State Drive (SSD) are types of non-volatile memory. These types of memory are useful for storing information to keep long term, for example, programs, operating systems (OS), configurations etc.

2.6.3. The power block

The power block in an IoT device distributes power in varying voltages to different components, for example, the power block may take 12v as an input and convert that to 5v for the CPU and 3v for a sensor.

The power regulation for computers in general is important, spikes or fluctuations in the voltage to a CPU may cause unpredictable or unexpected behaviour.

2.6.4. The sensor blocks

Sensors can come in two types, analog and digital.

Analogue

Analogue sensors have a varying voltage range depending on the measurement, for example as the temperature increases the resistance changes. Data received from an analogue input is usually considered untrusted until processed.

Analogue sensors usually require an Analog Digital Converter (ADC), to convert the signal to a numeric value.

Digital

Digital sensors are very similar to analogue sensors, however, they usually have an Analogue to Digital Converter (ADC) built in, freeing up one within the CPU. The data received from the ADC to the CPU is usually considered trusted. This may allow an attacker to manipulate the inputs of an IoT device to perform an unintended action.

2.6.5. The networking block

The networking block is responsible for how the device can communicate e.g. over Bluetooth, Wi-Fi, Ethernet.

The network interface is a common way to update IoT devices, usually an Over The Air (OTA) protocol is used to send the updates directly to the device. This poses a security concern as a potential attacker may listen to the traffic and obtain sensitive information.

The networking block uses the same protocols and technologies as discussed earlier while managing the physical, transport and logic layers of the OSI model.

2.6.6. The interface block

The interface block manages how the device interfaces with the user, for example, screens, LEDs, and audible interfaces. These usually communicate over a bus using serial or USB.

2.6.7. Architectures

The architecture used within a microprocessor can differ, however, the most common two are either the Harvard Architecture or the Von Neumann Architecture. Depending on the architecture used, this may present different attack surfaces towards IoT.

Harvard Architecture

The Harvard architecture is slightly different from the Von Neumann architecture, in the fact it has a separate bus for instructions and data. This version of the architecture is usually utilised within microcontrollers and RISC-based processors, as shown in Figure 6 below.

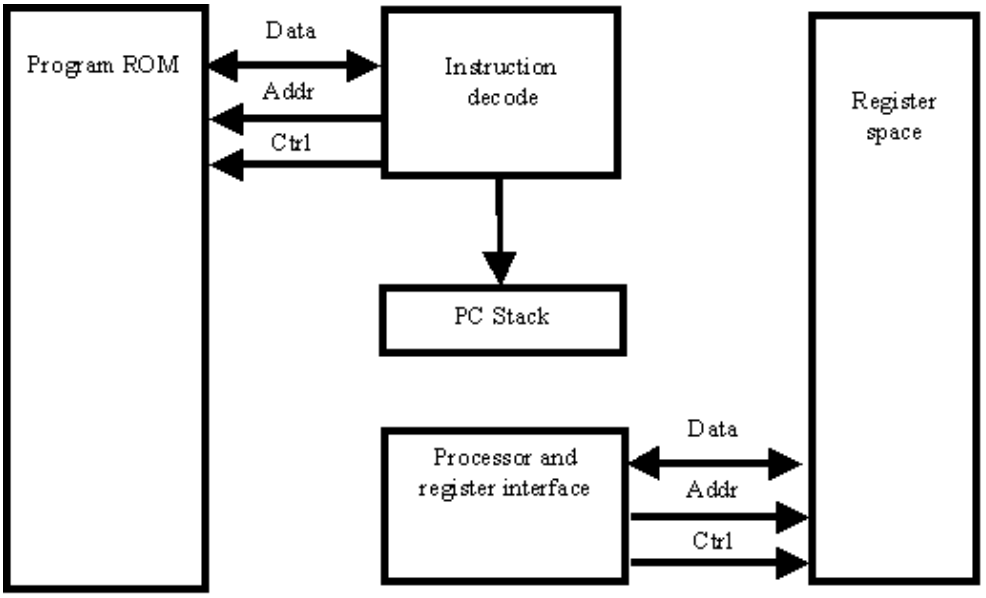


Figure 6: Harvard architecture (Gary Burt, 2004)

Von Neumann Architecture

In the Von Neumann architecture, in comparison to the Harvard architecture, there is a single bus for both the instruction and the data, as shown in Figure 7 below.

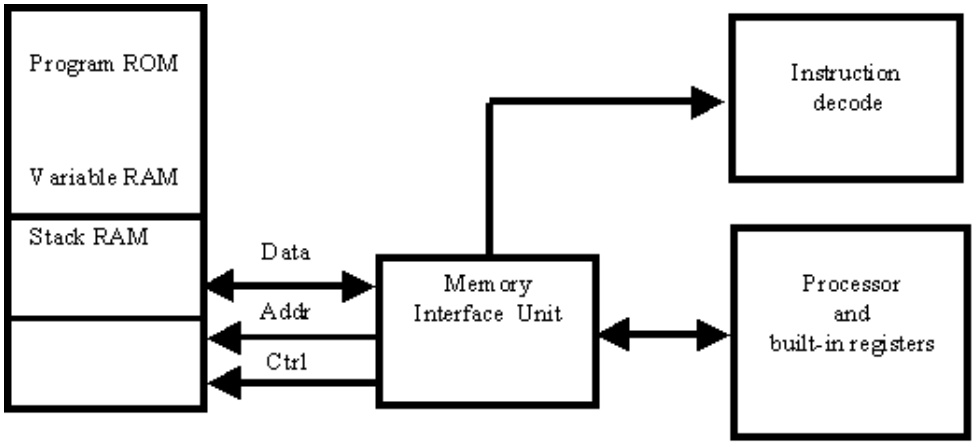


Figure 7: Von Neumann Architecture (Gary Burt, 2004)

2.7. Conclusion

Throughout the research, I've identified that a lot of IoT security is focused on networking and communication, moving forward, I will begin to focus on hardware specifically.

3. IoT Risks

As outlined in the paper written by (Ivan et al., 2016) IoT has a high security risk on the perception layer, due to physical and logical implementations.

The paper demonstrates that while there are security concerns around the networking layer outside of the home network, vulnerabilities may also reside within the hardware and protocols which are used by the device within the home network.

The paper (Tawalbeh et al., 2020) shows us that the main security concerns and challenges in IoT come from the inability to standardise the devices, as well as the scale in which they are used can have a cascading effect on security vulnerabilities being discovered. It is also mentioned that a lot of IoT comprises similar hardware / identical hardware.

Devices that may be a part of a security vulnerability could be difficult to update due to the wide variety of applications and deployments. IoT can take many form factors, ranging from full onboard CPUs to only microcontrollers. The variation generates complexity for the manufacturers to manage.

Due to the varying factors, it's not possible to outline all possible complexities and attack surfaces for IoT. Therefore, the responsibility of IoT security is pushed back to the consumer to ensure security.

3.1. Threat Sources

Agent	Goals	Risk	Work Factor	Methods
Nation States	Disruption or information	Very low	Extreme	Very sophisticated (0 days)
Hacktivists	Disruption or information	Low	High	Known, social engineering
Researchers	Financial, Academia	High	Medium	Known and or unique
Script Kiddies	Fun	Extreme	Very low	Known

Table 1: Threat Sources (David Wheeler et al., 2020)

The above table is a summarised version of the one produced as part of The IoT Architect's Guide (David Wheeler et al., 2020). The table lays out the potential threats a manufacturer will have to consider ranging from entities such as nation states which have a high work factor but shallow risk. The low-risk aspect is a measure of how much risk would the attacker take to find an exploit. In this case, they may be very cautious to ensure they don't reveal themselves.

On the other end of the table, we have "script kiddies" referring to entities who have little to no knowledge, and most likely reuse scripts and exploits which are well-known on vulnerable machines. Unlike nation-state actors which would very likely create new methods and exploits.

This demonstrates the agents a manufacturer may have to take into account when deciding on security.

3.2. Common Attack Vectors

3.2.1. Denial of Service (DOS) / Distributed Denial of Service (DDOS)

A Denial of Service (DOS) attack is an attack aimed towards preventing the device from functioning properly or rendering it fully unusable, the attack consists of flooding a network with useless packets to a specific device and exhausting the device

resources (e.g RAM, CPU, Bandwidth) until it can no longer process the useful packets.

A Distributed Denial of Service (DDOS) attack follows the same principles as the DOS attack however, is usually from a larger number of machines, simultaneously sending traffic to the device to deplete either resources or bandwidth.
The paper (Mikail Mohammed Salim et al., 2019), shows us that methods to mitigate DDOS attacks are mostly theoretical and may require the use of edge networks to direct traffic.

3.2.2. Replay attacks

A replay attack is a common method in which an attacker captures full or partial network packets and can replay these packets back to the device, or another device, with modifications to the data. This can cause unwanted effects on the devices, e.g. turning off smart plugs, lights and cameras.

There are 3 commonly used methods to circumvent replay attacks;

- A timestamp may be used to make sure that the packets that are sent to the devices are current and not old packets replayed.
- A unique string may be used to prevent an attack however this is memory intensive.
- A response challenge may be used which consists of a challenge which needs to be solved before processing the packet, although this requires a common secret to be used.

3.2.3. Brute Force

A brute force attack may be used to crack an authentication code/value which is exchanged as part of a handshake, or other communication between two devices. A brute force attack consists of randomly guessing passwords/key phrases which are commonly used (e.g. a rainbow table) to see if two hashes match.

A common prevention method is to use a salt, to create a new string by prepending the value. As shown in (F. James., 2019) brute force attacks can be mitigated by the use of intrusion detection systems, allowing a controller to drop specific packets based on the sender.
This would prevent an IoT device from becoming inaccessible.

3.2.4. Spoofing

A spoofing attack consists of an attacker disguising an event to create the appearance it originates from a trusted source.

3.3. Common Challenges in Hardware Security

There are multiple factors which play into IoT security, such as software, networking, and in some cases cloud security. One of the most overlooked factors though is hardware security.

Hardware is usually overlooked due to complexities and concerns around external companies infringing on Intellectual Property (IP), reverse engineering proprietary Integrated Circuits (IC) or installing back doors into firmware. Sidhu et al. (2019) explores the different potential vulnerabilities of IoT devices, as well as assessing the potential damage by hardware attacks.

3.3.1. Hardware Trojans (HT)

Hardware Trojans are physical alterations to an existing circuit, for example, to skip security boot checks. All types of IoT devices are vulnerable to HTs, however the behaviour of the trojan cannot change once installed, although this also means they cannot be removed by software updates.

3.3.2. Side-Channel Attacks (SCAs)

Side-channel attacks are cyber forensic hardware-based attacks, for example, analysing power signatures. SCAs are directly related to the physical implementation. SCAs are a threat to IoT security as the attack may exploit information such as cryptographic keys.

There are two types of SCAs, passive and active. A passive SCA follows the process of exploiting the output, where the attacker monitors the outputs without tampering. An active SCA, is the process of exploiting the inputs, directly manipulating inputs to create fault-injection attacks.

3.3.3. Cold Boot Attacks (CBAs)

A cold boot attack is the process of retrieving information stored within volatile memory (SRAM or DRAM). A CBA is a form of side-channel attack where an attacker, with physical access to the memory, can perform a memory dump.

Due to the laws of physics, the electrical charges stored within the memory aren't immediately dissipated when power is lost, due to capacitance.

The original paper from a team at Princeton (Halderman et al., 2009), demonstrated the attack in 2009, however the results were called into question, due to the lack of real executions in real scenarios. It was observed that flash-freezing the memory modules using a compressed liquid to drop the temperature below freezing dramatically slowed the dissipation rate.

A study by Carbone et al. (2011), showed that a CBA would be feasible in the field, however in some cases could degrade the data stored within the memory. If done improperly. The paper concludes that all experiments were conducted successfully and data was extracted from a wide variety of machines every time.

In the case of IoT cold boot attacks can still be effective, due to the firmware being decrypted (if applicable) every boot, meaning the attack could be repeated if unsuccessful the first time.

3.3.4. RowHammer Attacks (RBAs)

Depending on the type of memory used within IoT systems such as Dynamic Random Access Memory (DRAM), it may be vulnerable to RowHammer attacks. RowHammer attacks involve repetitive access to an address, resulting in possible bit flips to neighbouring bits. The vulnerability occurs due to the distance between DRAM modules.

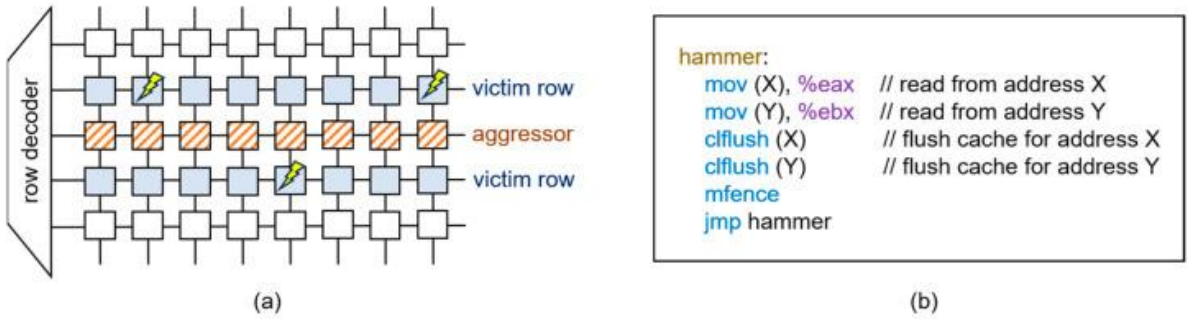


Figure 8: RowHammer attack illustration (Kim et al., 2024)

Figure 8 (Kim et al., 2024), demonstrates the basics for a RowHammer attack. The aggressor row is accessed very quickly, resulting in bit flips in the neighbouring rows. IoT devices may be vulnerable to this type of attack due to the small form factor, this poses a new security concern bypassing traditional security. RowHammer attacks can be executed without physical access to a device.

3.3.5. RamBleed (RB)

RAMBleed is based on the previously mentioned RowHammer attack, however is more focused towards Error Code Correction (ECC) Memory. In the study by Kwon et al., in 2020 it was proved to extract 2048-bit RSA keys from memory. Whereas previously using RowHammer, attackers were unable to directly read memory, only write. RamBleed makes RowHammer not only a threat to integrity but also confidentiality.

3.4. Common Security Practices

3.4.1. Root Of Trust (RoT)

A root of trust provides the foundations for performing secure operations within any device, usually protecting cryptographic keys and other sensitive information. The RoT allows devices to utilise Secure Boot, a process in which upon boot validates drivers and other system functions for malicious changes.

The Root of Trust is a key component in preventing cyber security attacks within IoT, preventing Side channel attacks, processor compromises, and memory extraction alongside other threats.

RoTs are usually implemented in a dedicated hardware module. A Trusted Platform Module (TPM) is a method for providing a cryptoprocessor, which is dedicated to cryptographic functions. A TPM module is usually a separate entry to the main CPU, which can be interfaced with directly.

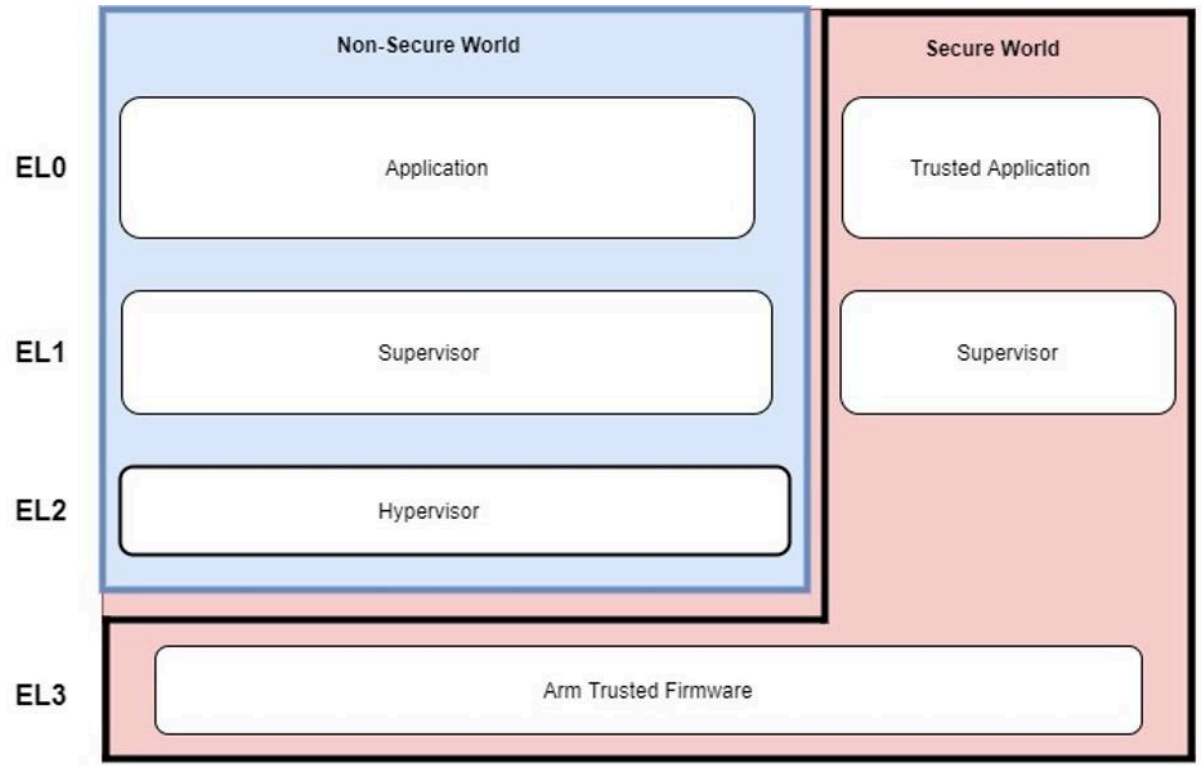


Figure 9: Root Of Trust illustration (Doran, 2022)

Figure 9 illustrates how the RoT is implemented within a systems architecture, demonstrating the isolation between data as well as code, which is controlled by the manufacturer. The secure zones of a TPM module, or TrustZone (ARM Equivalent of TPM) module can usually only be modified with physical access.

Although a hardware Root of Trust is very secure (NIST, 2016), security measures through code are still required.

3.5. Code Signing

Code signing is the process of using certificate-based signatures, enabling a manufacturer or end user to verify the authenticity of a firmware version. The process ensures that any updates come from a validated source and haven't been tampered with, however, this becomes obsolete when an attacker can overwrite both the bootloader and app partitions, removing the validation entirely.

3.6 Public Key Infrastructure (PKI)

The PKI has been utilised throughout the internet for digital certificates. PKI delivers the basics for the CIA triangle, ensuring privacy through encryption and authentication.

The PKI process allows end entities, such as IoT devices, laptops, and mobiles to connect to other devices by sharing certificates to provide their identities. Both entities need to validate the chain of certificates to ensure that they originate from a trusted authority (trusted anchor) which is usually predefined. (National Cyber Security Center, 2020)

However, due to the complexity of PKI, it's difficult to implement into IoT devices which utilise microcontrollers due to the constraints in computation and memory rather than microprocessors.

3.7. Conclusion

Security within IoT plays a vital role in privacy and availability in organisations. It is indicated that security is not properly addressed by many manufacturers and is left to be the responsibility of the consumer due to the complexities. Furthermore, maintenance heavily relies on security, due to Over The Air (OTA) updates being the most common form to update IoT devices.

It is increasingly important to improve security within IoT through both physical and digital implementations through the network and software. Minimising the risk of compromise due to the reduced number of vulnerable devices.

4. Methodology

4.1. Introduction

To gain an understanding of how IoT devices are used, I have created a survey with a range of questions to gain a broader understanding from both a technical perspective and a non-technical perspective.

I will use the information from the survey to gauge what devices are used, how they are used, and what concerns users have about the devices.

4.2. Project Management

This research will be managed by an Agile/Scrum methodology. I will use sprint planning, backlog refinement and periodic reviews throughout my project to make sure the project stays focused on the aim and is on track to complete the objectives. I have removed stakeholder reviews and the sprint reviews as they won't be needed for this project and instead have periodic retrospectives to replace these.

Task	Estimated weeks	Actual weeks
Project startup - Choose supervisor / project	2.5	2.5
Project startup - PID	2	2
Project startup - Ethics	9	3
Literature Review	6	30
Methodology / Data collection	7	20
Analysis	2	3
Design/Implementation	8.5	10
Evaluation	8.5	4
Discussion/Conclusion	8	2
Review	1	1
Technical Investigations	10	26

Table 2: Estimated compared to actual weeks

As shown in Table 2, the actual timeline of my project became quite different from the estimated timeline, this was mainly due to complexities within the device and implementation taking longer than expected. Some sections such as the literature

review became an iterative cycle as I discovered new things through the build. The data collection took longer than planned, this was to reach as many people as possible.

4.3. Research Strategy

As part of my research, I have created a quantitative survey as this method is ideal for reaching a large number of people. The questions must be meaningful and measurable, as well as being suitable to be answered quickly. A survey is also suitable for reaching a wider range of people.

4.4. Research Design

In this section, I will evaluate my research questions as well as how they will be useful in the summary.

4.4.1. Question 1

“What types of devices are used at home? (e.g Smart home, thermostat, cameras, Wi-Fi extenders)”

The first question will help me evaluate what devices are used, and how they are used. This is important because some devices are more sensitive in the event of a compromise. For example a security camera compared to a hoover.

4.4.2. Question 2

“Do you maintain the devices used at home? (e.g. software/firmware updates)”

Keeping the devices updated is important, this question will also help gauge how many devices may be vulnerable to attack.

4.4.3. Question 3

“Are you concerned about the security of the devices?”

This question will aim to provide a sense of how users feel about the devices, which may have interesting correlations to how devices are used as well as how many devices a user may have.

4.4.4. Question 4

“What tools do you use to manage these devices? (e.g Google Home, Home Assistant)“

This question will provide information about how devices may be managed and maintained. This is important as open-sourced solutions such as Home Assistant may not automatically update devices, which could lead to leaving devices open to security vulnerabilities.

4.4.5. Question 5

“Any other notes you’d like to add?”

This question will provide any additional information which may be relevant adding further information to the previous questions.

I have created two sets of questions, the previous questions 1-5 are aimed towards the general user, and the other set is focused towards businesses. I have focused on just the user questions as I expect these to be the majority of my responses to the survey.

4.5. Tooling

For the tooling, I have used Google Forms which will provide me with an easy way to distribute the form, reaching multiple people. Google Forms comes from a well-known brand, as well as meets the GDPR requirements I need to keep the data secure.

To avoid any bias, I have shared on my LinkedIn network to reach a range of people in varying job roles and technical skills.

4.6. Summary

I have created a variety of questions to gain insight into how IoT devices are utilised and managed, and the concerns surrounding IoT. I have considered any potential biases and tried to mitigate these where possible.

5. Data Analysis

5.1. Introduction

In this section, are the results of my primary research and evaluations of the potential reasoning behind the gathered results.

For this research an online Google form was created, targeted towards both technical and non-technical users, to gain a wider understanding of the concerns circling IoT, as well as any mitigations or preventions the end users have established.

The survey starts by gaining the person's confirmation and providing the information sheet, as per university policy, to store the data.

I have separated the enterprise and consumer results by the first question, due to the different requirements between a consumer and an organisation.

5.2. Results

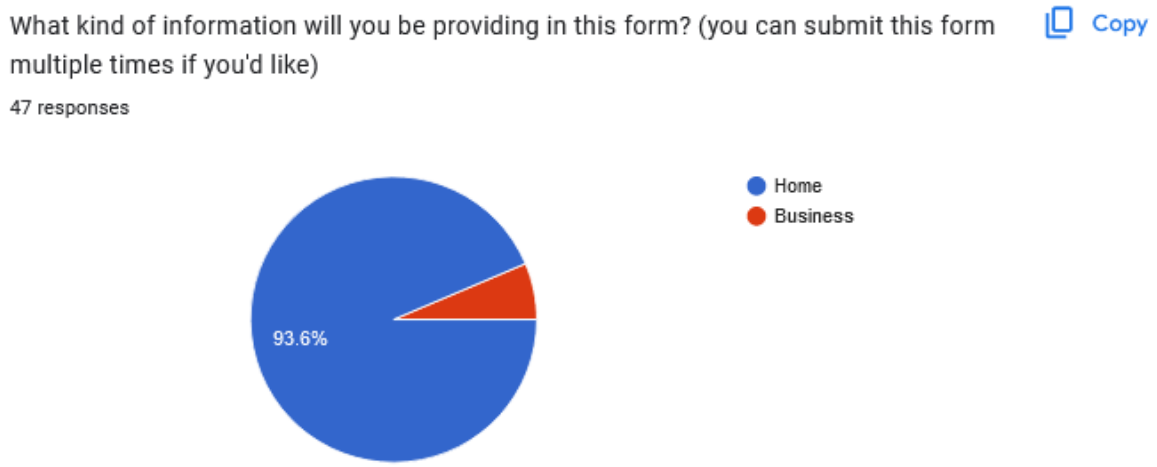


Figure 10: User / Business pie chart

From Figure 10, we can see that the majority of the users who submitted an answer to the survey were home users, which I will focus on for the remainder of the project.

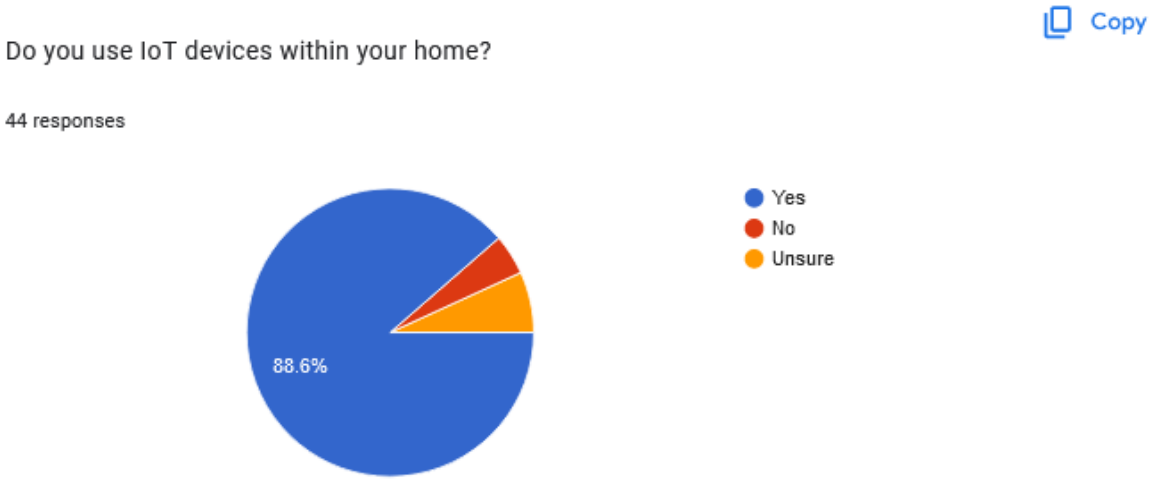


Figure 11: Percentage of users using IoT devices

From Figure 11, 88.6% of the users said yes to using devices within their home, 4.5% said no and 6.8% were unsure. This metric is interesting due to the majority of

the users have IoT devices, however, there is a small percentage (6.8%) who were unsure, meaning these devices may not be maintained and can pose security concerns.

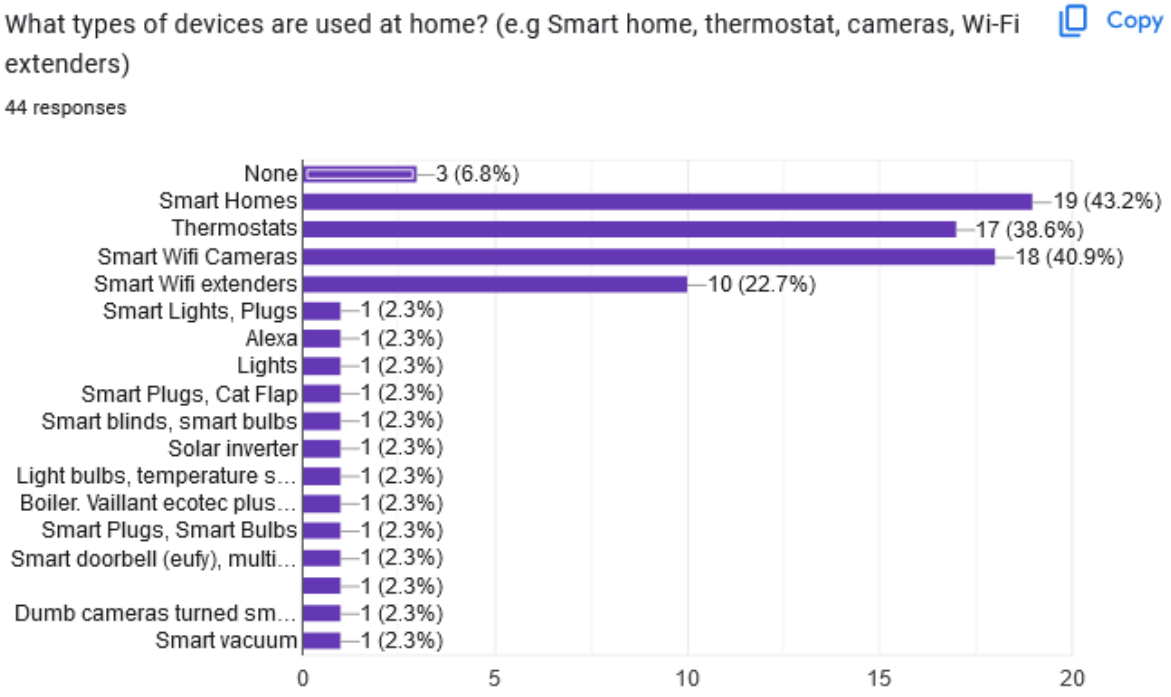


Figure 12: Device distribution

The next question shows us that the majority of users have some form of smart home, wifi setup, thermometer or camera. Figure 12 shows us that IoT devices can be deployed in many instances with varying security risks, e.g. smart cameras and even in some cases smart locks.

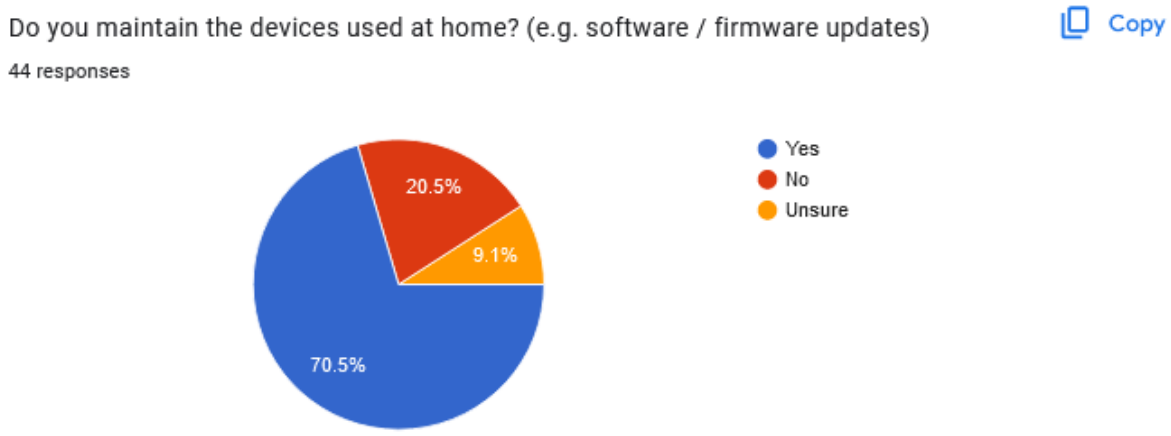


Figure 13: Percentage of maintained devices

From Figure 13, it's important to note that roughly 30% of users are unsure or do not update their IoT devices.

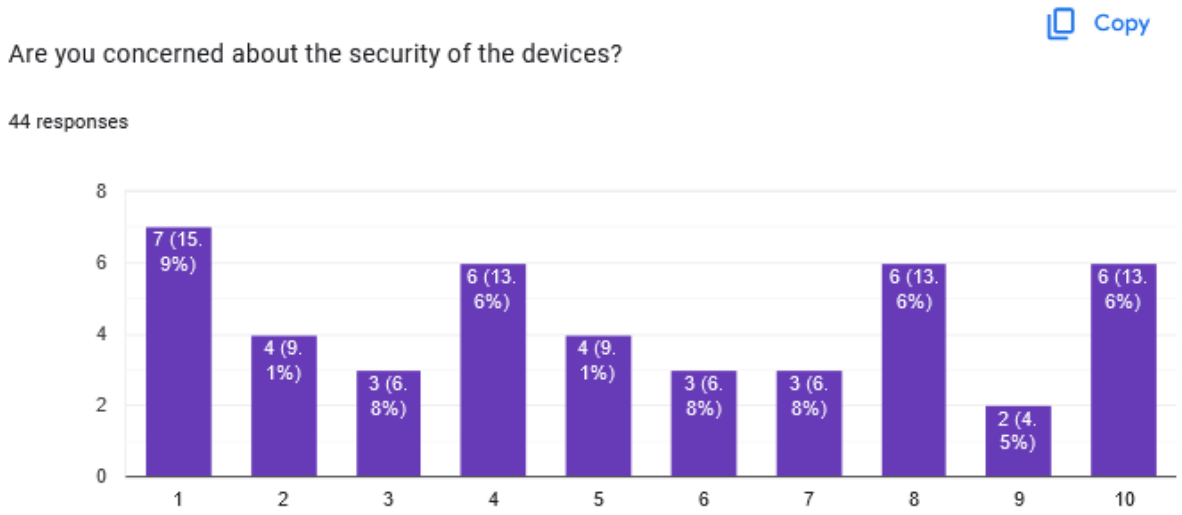


Figure 14: Scale of concern around IoT

The average level of concern falls around 4.95 as shown in Figure 14, where 1 is not concerned and 10 is very concerned, this shows us that there is some concern.

“What tools do you use to manage these devices? (e.g Google Home, Home Assistant)”

The results for this question range quite a bit between no solutions, to home deployed solutions such as home assistant, to fully managed services e.g. Google Home.

Quite a few of the results refer back to the device’s in-built security features or rely on the service provider, e.g. Hive/Google/Amazon. This is an interesting factor since different companies have different standards depending on their operating regions e.g. Google, globally. Due to the different standards, certain companies may undergo stricter processors and certifications and provide longer-term security updates. Another example of this is devices within Europe most likely have to comply with GDPR.

The final question “Any other notes you’d like to add?” adds quite a bit of insight into why IoT devices are not used.

The main points from the final question:

1. Security - A large number of the responses to this question mention that security is one big factor as to why they are cautious of what IoT devices are present in their homes.

2. Privacy - Another factor is privacy, how is the data managed and does it all stay within the user's network.
3. Maintenance - Another factor is that open-sourced solutions are not simple to set up and maintain adding overhead to all the devices connected to the solution.

Overall, there is a trend towards users being concerned about what devices they use and set up, primarily around security and privacy. The data showed that the most common devices are smart homes, usually produced by reputable manufacturers, for example, Google or Microsoft. The overall concern of IoT devices fell roughly at 5, on a scale between 1 and 10, indicating there was some concern.

The most common theme for the additional notes was privacy, indicating that users aren't comfortable with how their data may be processed or transmitted over the internet. Another concern was security, for the same reasons as privacy.

6. Discussion

6.1. Introduction and Scope

In this section, I will discuss more information about how an IoT device could be manipulated. It is worth noting that all of the testing and operations were completed within my network which I have permission to use.

I have purchased a smart plug to evaluate the security, to keep the manufacturer anonymous I will refer to the product as "Smart Device".

6.2. Device investigation

The first thing I looked into was how the Smart Device was physically set up, in this case, it was a plastic shell which was sealed together with no screws.

The first step was to install the Smart Device app and register the device. After the setup, I was able to control the Smart Device remotely which in this case was a smart plug.

This poses a few potential attack vectors for the next stage, through the device itself, the mobile application or via a network.

6.2.1. Networking

The first stage was to dig closer into how the device communicated over the network, using Wireshark we can have a closer look at how the device connects, as well as any information it may be sending.

No.	Time	Source	Destination	Protocol	Length	Info
210	3.568613	192.168.0.43	255.255.255.255	UDP	214	52071 → 6667 Len=172
419	8.689300	192.168.0.43	255.255.255.255	UDP	214	52071 → 6667 Len=172
476	13.623433	192.168.0.43	255.255.255.255	UDP	214	52071 → 6667 Len=172
589	18.519493	192.168.0.43	255.255.255.255	UDP	214	52071 → 6667 Len=172
641	23.661680	192.168.0.43	255.255.255.255	UDP	214	52071 → 6667 Len=172
706	28.554436	192.168.0.43	255.255.255.255	UDP	214	52071 → 6667 Len=172
751	33.696644	192.168.0.43	255.255.255.255	UDP	222	52071 → 6667 Len=172
986	38.599105	192.168.0.43	255.255.255.255	UDP	222	52071 → 6667 Len=172
1010	43.731923	192.168.0.43	255.255.255.255	UDP	222	52071 → 6667 Len=172
1029	48.633357	192.168.0.43	255.255.255.255	UDP	222	52071 → 6667 Len=172
1149	53.540304	192.168.0.43	255.255.255.255	UDP	222	52071 → 6667 Len=172
1183	58.681069	192.168.0.43	255.255.255.255	UDP	222	52071 → 6667 Len=172
1213	63.575041	192.168.0.43	255.255.255.255	UDP	222	52071 → 6667 Len=172
1231	68.721725	192.168.0.43	255.255.255.255	UDP	222	52071 → 6667 Len=172
1404	73.610561	192.168.0.43	255.255.255.255	UDP	222	52071 → 6667 Len=172
1416	78.525681	192.168.0.43	255.255.255.255	UDP	222	52071 → 6667 Len=172
1443	83.645566	192.168.0.43	255.255.255.255	UDP	222	52071 → 6667 Len=172
1485	88.560975	192.168.0.43	255.255.255.255	UDP	222	52071 → 6667 Len=172
1546	98.608067	192.168.0.43	255.255.255.255	UDP	222	52071 → 6667 Len=172
1579	103.716147	192.168.0.43	255.255.255.255	UDP	222	52071 → 6667 Len=172
1634	108.654517	192.168.0.43	255.255.255.255	UDP	222	52071 → 6667 Len=172
1697	113.583437	192.168.0.43	255.255.255.255	UDP	222	52071 → 6667 Len=172
1751	118.689350	192.168.0.43	255.255.255.255	UDP	222	52071 → 6667 Len=172
1783	123.596639	192.168.0.43	255.255.255.255	UDP	222	52071 → 6667 Len=172
1800	128.727496	192.168.0.43	255.255.255.255	UDP	222	52071 → 6667 Len=172

Figure 15: Wireshark Network Capture

From the screenshot from Wireshark (Figure 15), we can see that the device sends a call to home, roughly every 5 seconds. Depending on the implementation of the network protocols and authentication, these may be vulnerable to a replay attack.

6.2.2. Physical implementation

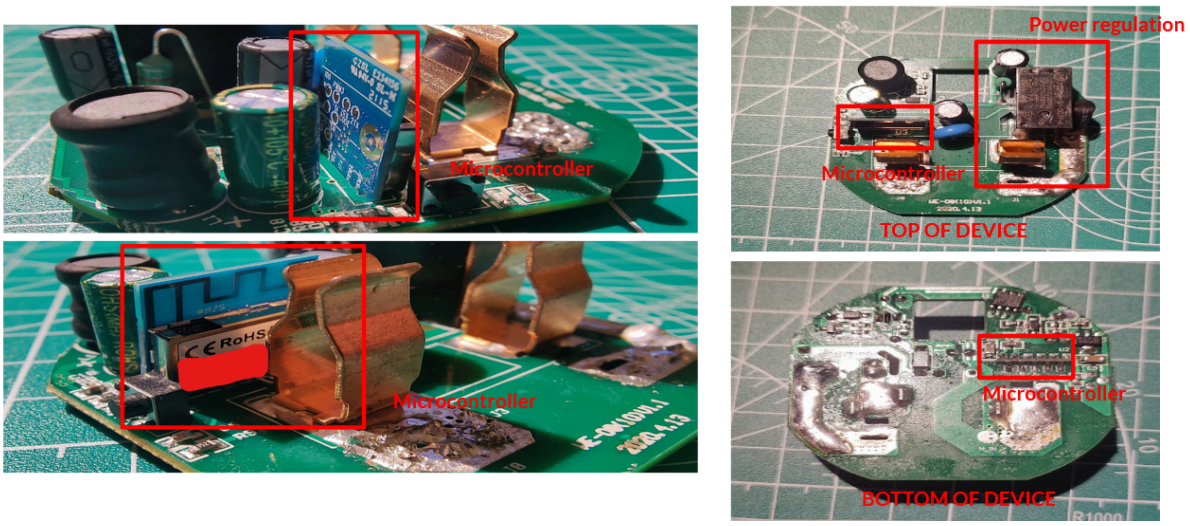


Figure 16: Smart Device main board

Upon closer inspection, after opening the device as shown in Figure 16 the smart plug uses a third-party breakout board to manage all communication. The green board manages power to the device from the socket.

6.3. Dismantling

6.3.1. IC1

The next stage was to isolate the blue breakout board, which I will refer to as IC1. Upon further research, IC1 is a general-purpose IoT board, deployed in multiple products. The IC1 has a wide range of functionality, such as Wi-Fi integration as well as a few GPIO pins allowing these to be installed in many devices.

Due to the metal shielding on the IC1 which protects the main IC underneath we can't see the IC serial numbers. By removing the metal shielding we can identify the microprocessor used.

6.3.2. BK7231T

The Beken 7231T IC is a small chip which supports Wi-Fi and radio frequencies (RF), containing a small 32-bit CPU, 256kbit RAM and a 2MB flash as well as a real-time operating system.

The IC is based on the Arm Cortex M4 microcontroller.

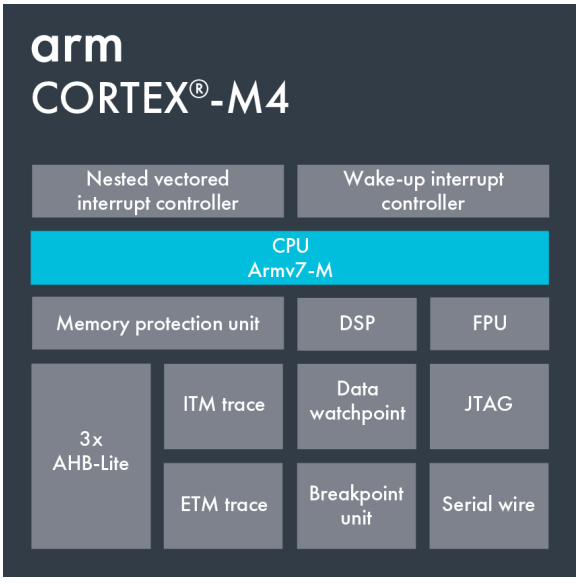


Figure 17: Cortex M-4 Architecture (ARM, n.d.-a)

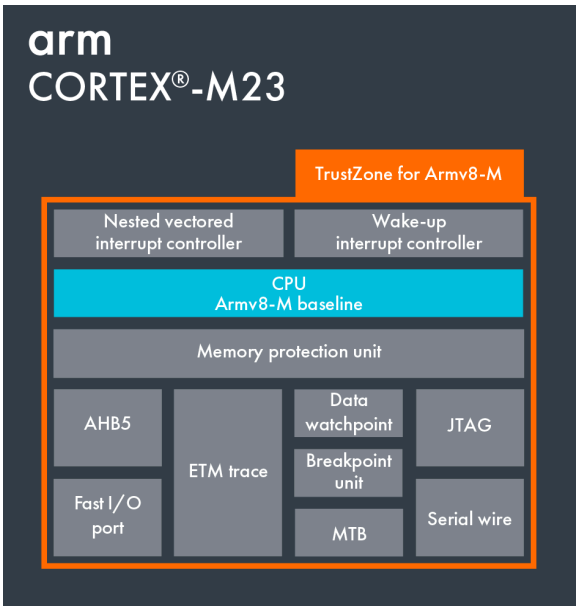


Figure 18: Cortex M-23 Architecture (ARM, n.d.-b)

As seen in the figures above (Figure 17 and Figure 18), the Cortex-M4 IC which is used within BK7231T, isn't part of the TrustZone family of microprocessors as shown in the documentation (ARM, n.d.-a). Due to the lack of a Root of Trust (RoT), this indicates that the bootloader may manage the decryption and store the decryption key in a storage partition located somewhere within the firmware itself.

Furthermore, in the IC1 datasheet, we begin to map out the attack surface. The datasheet identifies 256KB of RAM, 2MB of storage and "extensive peripherals".

Due to the lack of a TrustZone, which would usually manage all of the cryptographic key management and processes, I can assume the key and any sensitive information is within the readable text. however, this key should be retrievable.

Another feature of the Cortex-M4 is the Memory Protection Unit (MPU), which isolates the memory into 8 protected regions enforcing software-defined access levels and isolating processes to use a specific region of memory. Due to the circuit only having one IC we can assume the memory is located on the chip itself. If the manufacturer has assumed the memory is protected, similar to using a processor without a trusted zone, the memory may be vulnerable to memory-focused attacks such as Rowhammer or side-channel attacks.

Real-Time Operating Systems (RTOS)

A Real-Time Operating System (RTOS) is an operating system which processes data and events which have a critical time frame, meaning these OS's are event-driven and preemptive. This allows for task switching based on priority, rather than a clock interrupt.

The IC1 Module is based on RTOS, due to the use of Wi-Fi and TCP/IP protocols.

It is recommended that RTOS systems should have sufficient capabilities to store data securely utilising encryption and partitioning.

6.4. Breakout boards

The next step was to isolate the IC1 module, this allows for easier interfacing reducing the risk of the main power from the main board. I've created a basic schematic to interface with the IC1 module, adding an LED for power as well as one for the status of the plug emulating if the switch is powered or not.

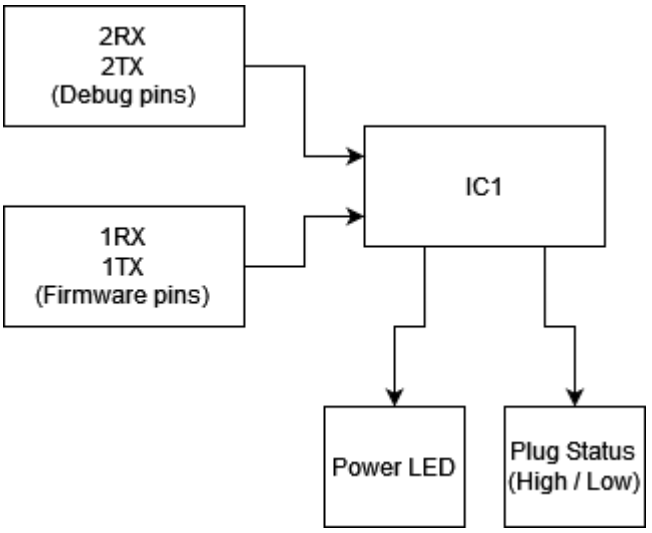


Figure 19: IC2 Block diagram

Figure 19, shows a basic block diagram of the inputs and outputs to interface with IC1.

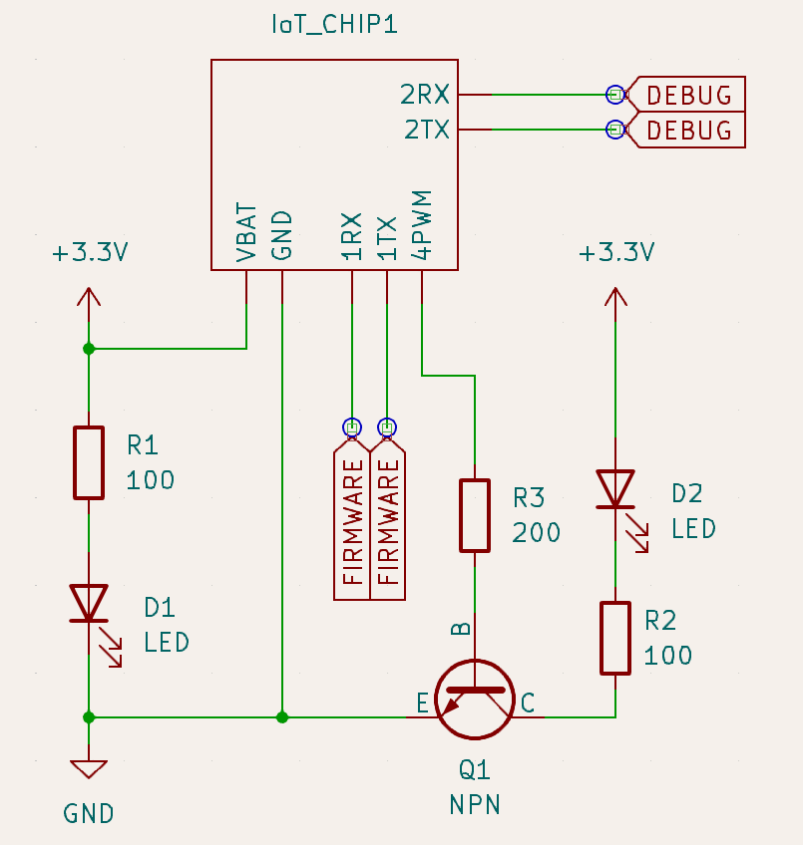


Figure 20: Custom circuit to interface with IC1

Figure 20, shows the basic schematic for interfacing and providing access to 2 DEBUG pins, 2 FIRMWARE pins, power and status. This will allow me to interface with the IC directly, in a safer environment.

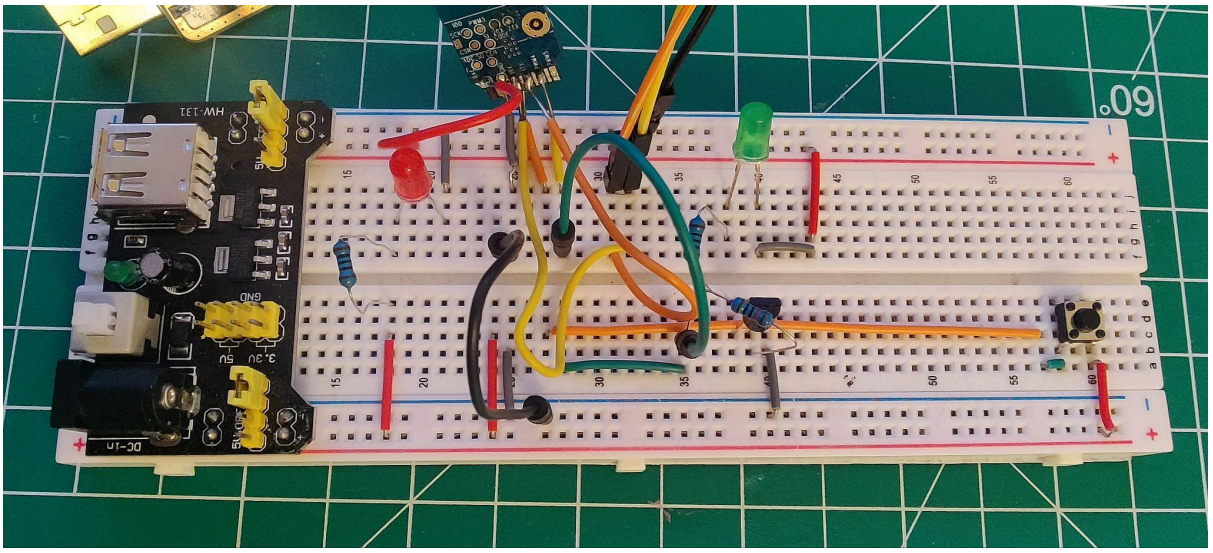


Figure 21: Physical implementation

Figure 21, shows the physical implementation of Figure 20, connected to IC1

6.4.1. Raspberry PI Setup

In Figure 20, there are two sets of ports, 1RX/1TX and 2RX/2TX, focusing on 2RX/2TX, from the datasheet we can expect a serial response from these ports.

Initially, I tried to connect the transmitter directly to a Raspberry Pi (Model 3B) serial GPIO pin. However, I didn't get the response I expected, instead of anything readable I got a large string of random bytes. I tried a variety of different encodings and baud rates however I still was unable to find any useful information.

Due to the potential issues of interference, I purchased a basic USB to TTL converter for £3, reducing the chance of error in the designs, and also removing the Raspberry Pi. Upon adding the converter the response was much clearer, providing useful information which I will explore in the next section.

6.5. Debug Ports

6.5.1. Listening to serial ports

The debug ports provide useful information to system manufacturers to aid in troubleshooting devices. Through some trial and error, I was able to find a suitable baud rate for reading the debug ports. There may be cases where a developer has left information, similar to printing data in a console, in the debug logs. I can use this information to extract information about the device during startup.

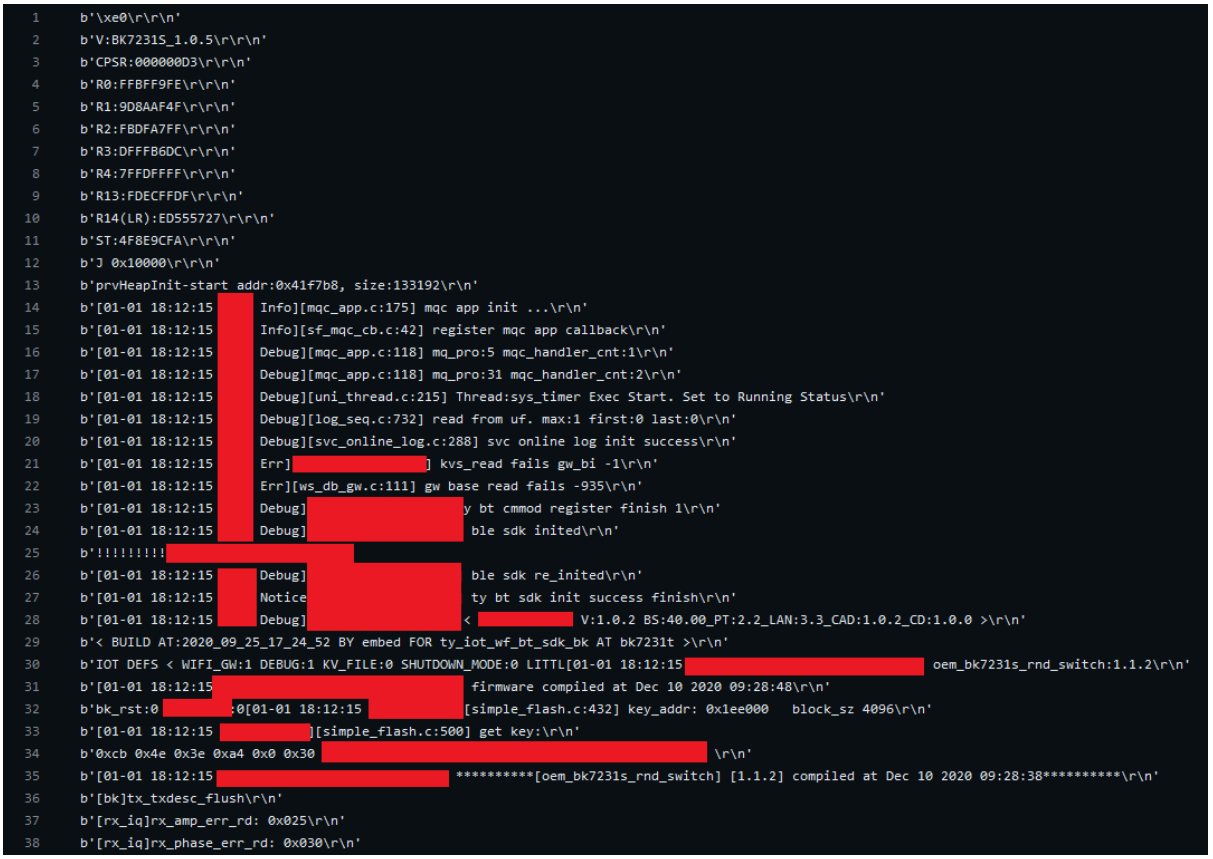


Figure 22: Debug Log (All sensitive/identifiable information removed)

From collecting the debug logs, shown in Figure 22, I could investigate what happened during the initial boot process. From Figure 22, we can see some information about the device. I've added a few bits of information that are not in the screenshot, due to the log containing 422 lines into Table 3.

Property	Value	Description
Firmware version	b'V:BK7231S_1.0.5\r\r\n'	The firmware version
Register values	b'R0:FFBFF9FE\r\r\n' b'R1:9D8AAF4F\r\r\n' b'R2:FBDFA7FF\r\r\n' b'R3:DFFFB6DC\r\r\n' b'R4:7FFDFFFF\r\r\n' b'R13:FDECFFDF\r\r\n' b'R14(LR):ED555727\r\r\n'	The information stored within the registers may give sensitive information.
Firmware compile date	b'[01-01 18:12:15 Notice][device.c:216] firmware compiled at Dec 10 2020 09:28:48\r\n'	The date on which the firmware was compiled.

Device ID	Devid:bfbXXXXXXXXXX	The ID of the device itself
Security information	b'security2cipher 2 2 16 16 security=5\n' b'cipher2security 2 2 16 16\n'	Information about the cipher protocols

Table 3: Extracted Information

Another output states ‘b'CREATE DB SUCCESS\r\n'’ Indicating a database stored on the device.

Quite a few of the values provided in the table will be unique to a device and can’t be easily read by the consumer.

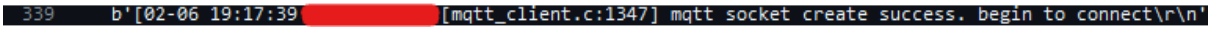


Figure 23: Debug Log snippet

From this extract from the debug log, Figure 23, we can see MQTT is utilised for network communication.

6.6. Firmware Dumping

The first step was to connect to the firmware ports, rather than debug, as I had set this up through a breadboard I could easily switch the pins. After connecting the device to the Raspberry PI physically I had to connect via USB.

The next step was to repeatedly boot the device, listen to the transmitter (1TX) and begin collecting the information from the IC. Initially, I tried to write this code myself, but in the process found a tool instead (LibreTiny, n.d.), that also provided useful information.

6.7. Entropy

Binwalk is a tool used for searching binary images for files and code. (*Binwalk | Kali Linux Tools*, n.d.)

Entropy is a measure of disorder or randomness, we can use the features within binwalk to estimate how much of our firmware image is encrypted or compressed.

Figure 12 shows that there is a high entropy through offset 0-1.1, this suggests that the bits have a high randomness indicating they may be encrypted, or compressed.

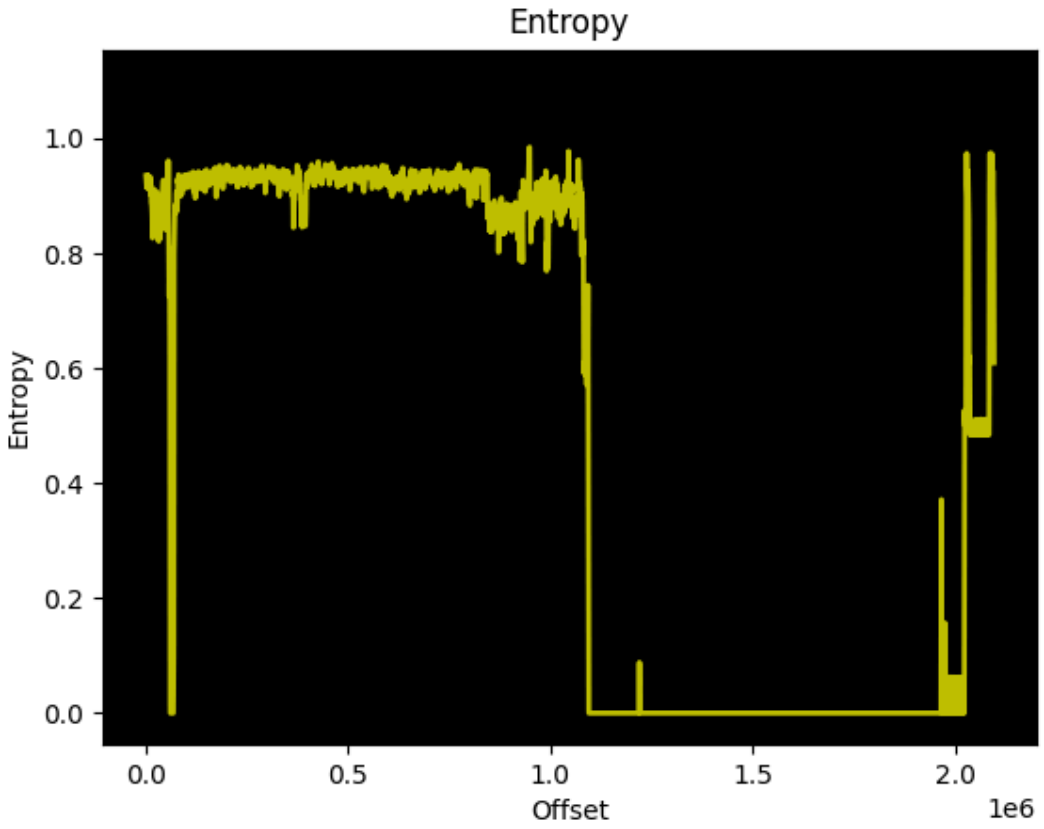


Figure 24: File entropy result

Figure 24 also shows that between offset 1.1 and 1.9, there is a very low entropy suggesting that these bits are not encrypted, however, this may be padding between the blocks.

Running binwalk over the firmware image provides no information about the headers as shown in Figure 25 below, if the file was not encrypted or compressed, data about the headers and firmware itself would be displayed here. Reinforcing the previous observation that the image is compressed or encrypted.

```
charlie@DESKTOP-JKCS7TJ:~/fyp-code$ binwalk dump-copy.bin

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
```

Figure 25: Binwalk result

6.8. Encryption

Initially, I started to investigate and dissect the firmware to try and discover any keys from the bootloader logs. I also tried to look through the raw encrypted version of the firmware file. I was unable to find any information of use.

I discovered a repository focused on the BK7231T chip specifically (ghsecuritylab, 2020/2020).
Upon reviewing the code, it appears that the programmer either dumped memory from the RAM or connected to the bootloader upon boot to find out how the file is decrypted.

This is possible due to the lack of the TrustZone, mentioned earlier alongside the lack of protected memory modules, meaning all cryptographic functions are performed by the bootloader upon boot.

The memory modules of IC1 reside on the IC, which makes it more complex for an attack to extract the contents during a cold boot attack, however, this is still achievable using the lasers to remove the top layer of the IC, to expose the modules within.

Reverse searching the keys, found within the repository using GitHub code search provides another 19 instances of these keys being shared on public GitHub repositories. From what I can observe, the same keys are used across the BK7231X family, affecting multiple ICs.

The key, consisting of 4 blocks of 32-bit binary integers, is quite secure however this is overshadowed by the fact that the same keys are used across multiple products, which is generally bad practice. The security of multiple chips is at risk if the shared key is exposed, rather than a single device.

Upon using the decryption script, provided by the tysdk_for_bk7231t repo (ghsecuritylab, 2020/2020), I can assume the data is no longer encrypted and I can begin to dissect the firmware into its partitions.

6.9. Firmware Dissecting

Initially, I began by digging through the source code which I had decrypted, this led to identifying the chip information, alongside the start blocks for both the bootloader and app partitions.

Through the sections of dissecting, I will write helper functions which I will attach to the appendix.

Figure 26 below shows a few strings, upon reverse searching the values I later discovered that they are commonly used for OTA update encryption using AES (*Finding Encryption Keys - LibreTiny*, n.d.).

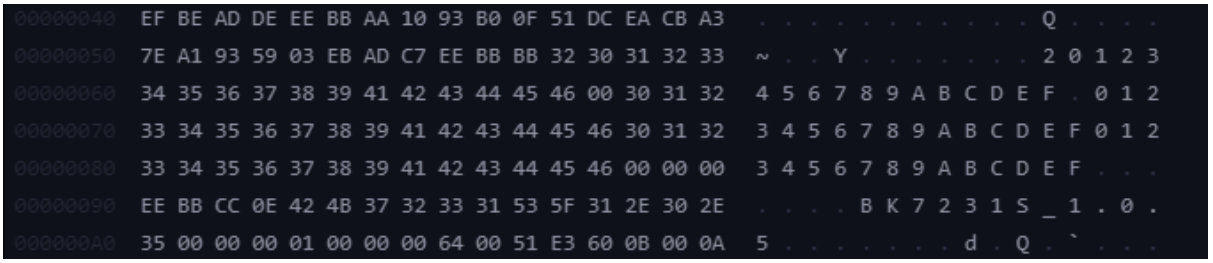


Figure 26: Firmware dump header content

While not explored as part of this paper, hypothetically, an attacker could be able to send an update in the correct format encrypted with this key to perform malicious updates.

I then tried to determine partitions for the bootloader and application, to do this I searched the firmware dump for keywords such as “boot loader”, “bootloader” “app”, “application” etc.

From Figure 27 below, we can identify the start of the bootloader partition, prefixed by ROM BootLoader (RBL) followed by a null (\xFF) byte.

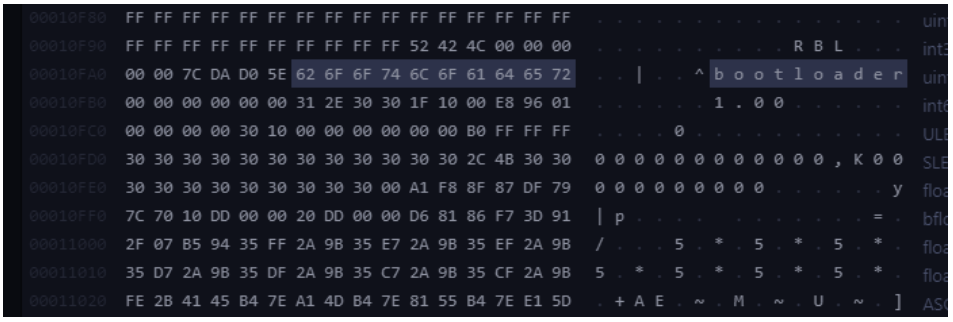


Figure 27: Firmware dump bootloader content

Searching the raw bytes file for the b"RBL\x00", I was able to extract the RBL start indexes, resulting in two values, [69530, 1220362].

The first partition, the bootloader most likely contains a header to identify the size of the partition and other metadata.

From Figure 28 I can identify the most likely end bytes to the block, being the last two null bytes. I can estimate the full header size is roughly 94 bytes of data, however, due to the fact null bytes are used for padding we can assume the full size is 96 bytes.

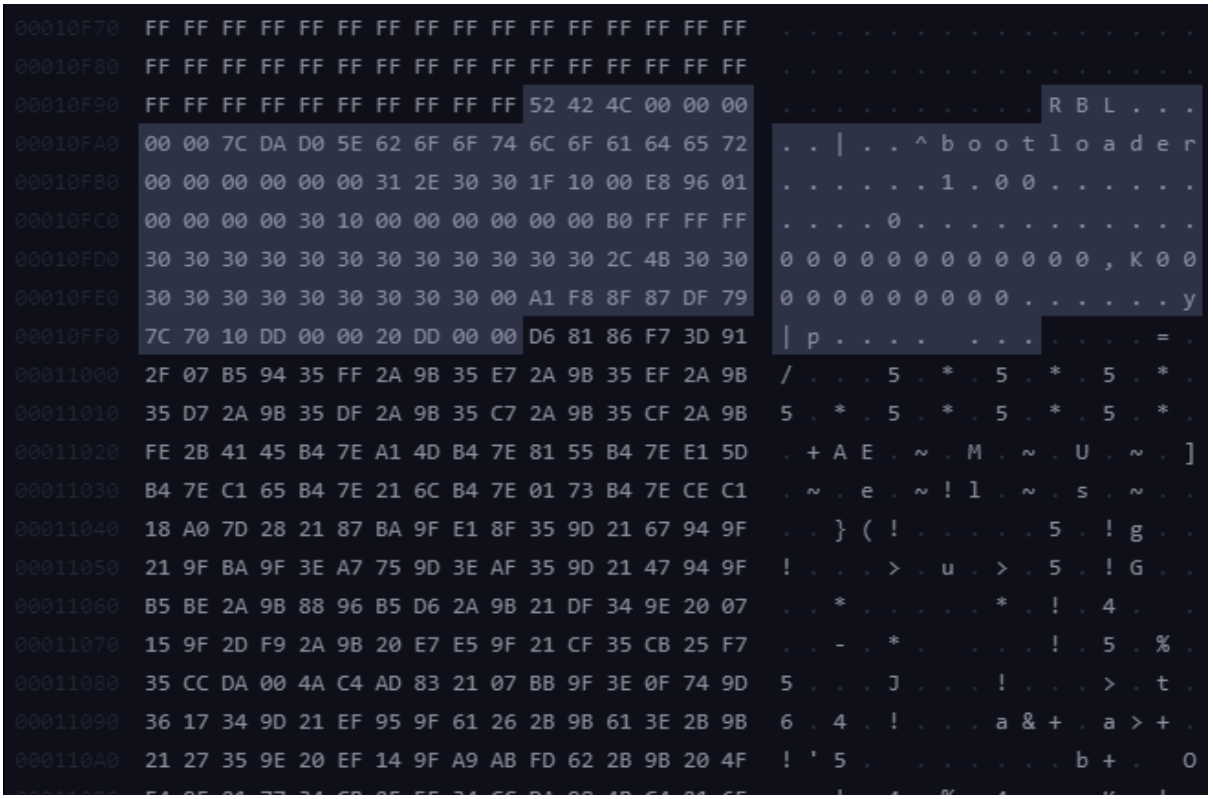


Figure 28: Bootloader header

Figure 29 shows the estimated locations for the metadata which can be used to isolate the blocks into their respective parts.

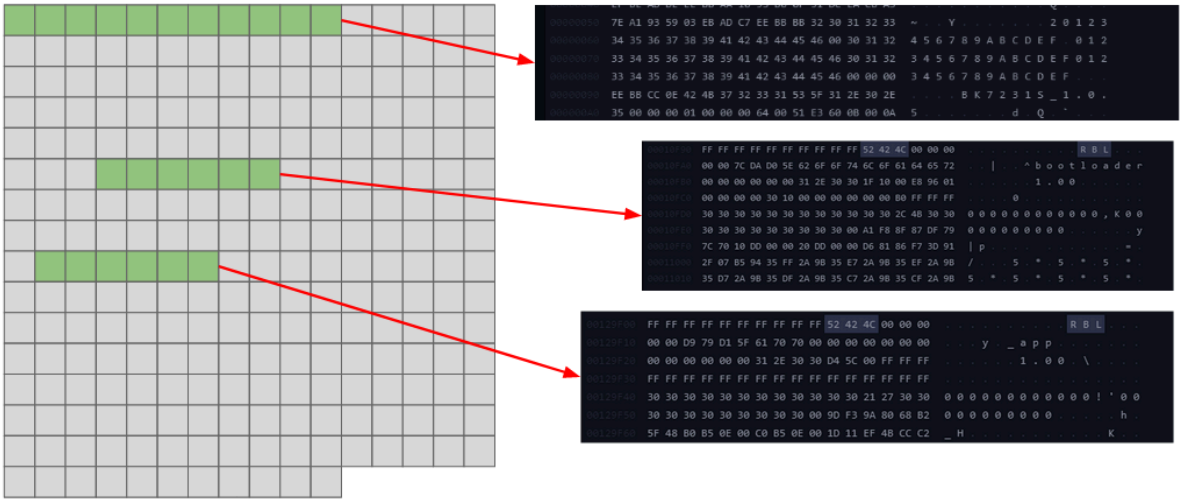


Figure 29: Estimated partition locations

From the metadata, I was able to decode the starting index as well as the size, as demonstrated in Figure 30.

Figure 30: Header locations

Through dissecting the source code I discovered that the null bytes (‘\xFF’) are used to pad the space between partitions. I can use this to identify the start and end of partitions.

I tried decompiling the source code for both the app and bootloader partitions to extract any API keys or sensitive information, however, I was unable to find anything of importance.

`Strings` is a Linux command to print all the printable strings within a file, upon running `strings` over the firmware file I was able to extract printable text.

Upon breaking the firmware into separate partitions, I noticed there was one block of encrypted data remaining as illustrated in Figure 31.

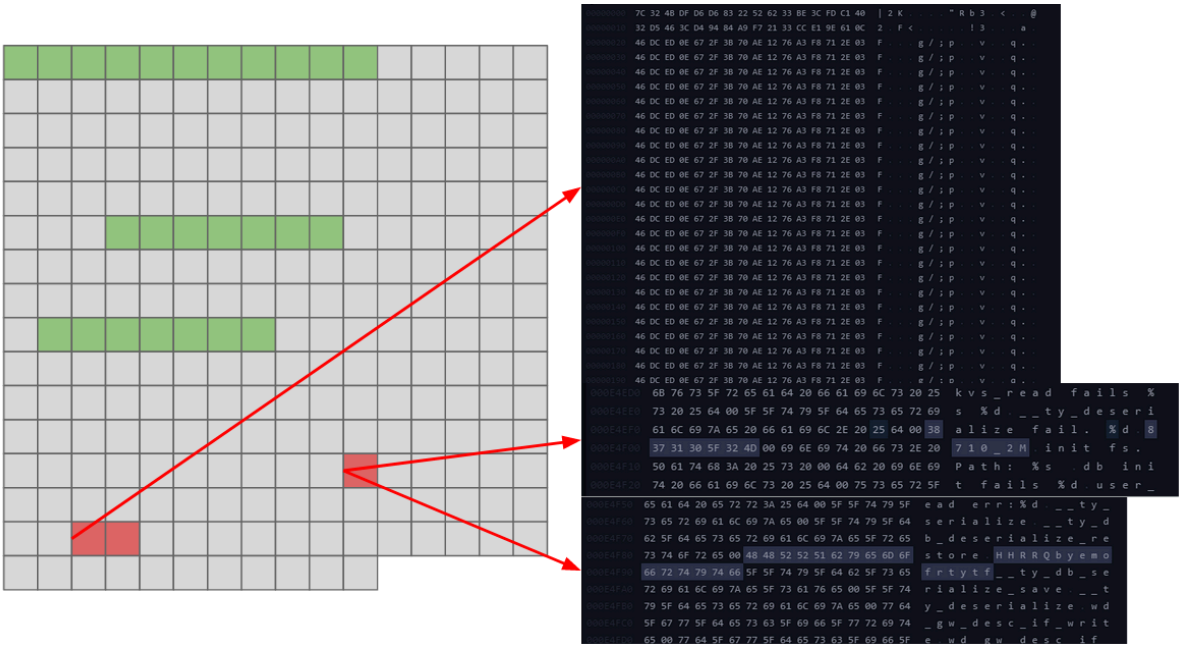


Figure 31: Storage partition

Due to data being present after a large block of null bytes, this indicates the location of the storage partition as no application data has been found so far. This refers back to the database which was discovered as part of the debug port investigation.

Utilising keyword searches such as “db”, “db init”, and “user”, I was able to find 2 additional keys used to protect the storage partitions.

The figure below (Figure 19) illustrates the final storage partition, the keys to decrypt this were stored near the start of the block. This isn’t a good security practice to store keys in plaintext and the keys are also vulnerable to memory dumps.

6.10. Mounting storage

Storage is an important part of an IoT device, in this case storing API URLs, API keys, device information, SSID information and other debug data.

Upon isolating the chunk of data from the firmware file, I tried to decrypt the data.

Initially, I tried decrypting with only the keys found previously from the start of the storage, however, this was unsuccessful.
I tried decrypting the block with the original key, this led to a small chunk of text. I tried a few variations of mixing the 3 keys and keys on their own to decrypt the full data, however, I was unsuccessful.

Upon mixing all 3 keys, in an alternating fashion, for example;

KEY_1[i] + KEY_2[i] + KEY_3[i]

I was finally able to decrypt the entire block.

Figure 32: Key output

Figure 33: Key output continued

For this paper, I have removed any identifiable information, alongside any specific URLs, APIs, or API keys and only shown a subsection of the data (shown in Figures 32, and 33).

7. Results

This project contained a series of questions and research topics.

Objective 1: Research IoT communication protocols; evaluate the use of different protocols and summarise the pros and cons of each.

I discussed objective 1 as part of my literature review, and discussed the pros and cons of the different protocols, alongside the additional infrastructure, if required.

From my findings, there is a wide range of IoT protocols, each of which needs to be evaluated independently to ensure confidentiality, integrity and authenticity.

The wide range of protocols also introduces new attack surfaces, for example, SIM cloning in the case of cellular communication, which will need to be investigated.

Objective 2: Research ‘Home Hubs’ for self-hosted IoT solutions; evaluate the use of home hubs to define what options a consumer has available, alongside the pros and cons.

Objective 3: Research the management and maintenance concerns within small and large-scale deployments.

Objective 4: Research the deployment of IoT devices at both small and large scales; explore the different challenges which may be faced at scale compared to at the consumer level.

Objectives 2, 3, and 4, became quite broad topics, which I was unable to address within this paper due to the number of varying factors. I believe that these are still important research questions, as they aim to provide users ownership of their devices on both a small and large scale, as well as maintaining the devices.

Objective 5: Research the security and privacy concerns within IoT; by reverse engineering an IoT device, I can evaluate the security and privacy concerns in a real scenario.

Objective 5 became the main topic for this paper and was the main discussion point from the survey results, I chose to continue this objective to provide a thorough investigation into the concerns.

From my findings, I discovered that in the case of the Smart Device, the concerns were valid, as encryption keys were reused, although not explored as part of this paper it may be possible for an attacker to provide malicious Over The Air updates to a device. The importance of this is significant, as it has the potential to leak data for example live video feeds from hijacked cameras using malicious firmware.

I was also able to retrieve API keys for the manufacturers' servers, which could be used maliciously by an attacker. Attacks like this could cause major disruption for an enterprise.

I also discovered there was a lack of a Root Of Trust, resulting in any cryptographic operations or keys having to be managed by the main IC itself. The importance of this is to protect the keys used within devices.

I believe that the investigation throughout the discussion explores objective 5 sufficiently to provide insight into IoT security from a hardware perspective for a consumer and or enterprise.

8. Conclusion

8.1. Implications

Throughout the project, I have evaluated and dissected the firmware from an IoT device, the majority of content relates to objective 5.

A few key security practices seem to have been broken, for example, the reuse of encryption keys, lack of the TrustZone or Root of trust both pose significant security concerns. The main concern is an attacker being able to reverse engineer the firmware to obtain sensitive keys.
A secondary concern is the use of non-distinct Over The Air (OTA) decryption keys, increasing the chances of an attacker having the ability to trigger an update to a malicious firmware without even reverse engineering a device.

I believe that security isn't properly addressed within IoT from a consumers view and needs support from manufacturers to ensure suitable standards and regulations.

During my survey, I identified a range of IoT devices consumers were using, ranging from smart cameras to smart appliances, for example, boilers. If these devices were to be attacked using the methods outlined in this paper it could create large privacy implications, such as leaking security camera footage. Another concern would be the SSID and password, this would allow an attacker a greater level of access to a network and could potentially target other devices.

IoT devices must be secure, devices that are deployed in life-critical situations such as pacemakers or environmentally hazardous conditions, must be reliable and ensure the confidentiality of the data.

In the case of the Smart Device evaluated within this paper, an attacker could create malicious firmware, flash the device Over The Air (OTA), issue unintended actions, or use the Smart Device as an entry point to the network.

The lack of secure memory modules and cryptoprocessors almost guarantees that encryption happens within memory and could be reverse-engineered by an attacker.

For enterprises, data protection and GDPR are important, a company must know how data is being processed, especially when personal data is involved. An attacker could cause significant damage to a company breaching an IoT device that manages sensitive data.

8.2. Recommendations

8.2.1. Manufacturer Recommendations

My main recommendation to manufacturers would be to implement the use of a TrustZone or similar, as mentioned before the TrustZone introduces a cryptographic processor alleviating the cryptographic functions from the main IoT device to a dedicated subprocessor. This would improve the security per device, enforcing access control through protected memory.

8.2.2. Consumer recommendations

My main recommendation to consumers would be to isolate the IoT devices from the main network utilising VLANS to mitigate potential damage. This does not resolve any data extraction concerns but isolates the devices to reduce the chance of devices such as mobile phones, and laptops becoming compromised too.

8.3. Future Work

As for future work building off this project, I believe that there is still a lot of work to be done surrounding hardware security within IoT, focusing on standards, and regulations.

I have only begun to thoroughly explore objective 5 of my aims, I believe it's important to explore the other IoT solutions in more depth, providing a better understanding of IoT ecosystems as a whole.

I think it's important to dive deeper into cryptoprocessors, ensuring that the vulnerabilities apparent in IoT devices without cryptoprocessors don’t exist in the cryptoprocessors themselves. Ensuring stronger cryptographic functionality.

I also believe it important to further investigate the ICs themselves by using industrial lasers for a full analysis of how the ICs operate, including how they communicate with memory. I believe it will also be beneficial to evaluate how the ICs handle trusted digital inputs and explore how they can be exploited.

Although briefly mentioned in this paper, I think further code analysis into the bootloader and application partitions would be beneficial.

8.4. Personal Reflection

This project is one of the hardest projects I’ve undertaken. There were many setbacks, and complications, throughout the journey. However, I am very satisfied with the result and findings and I have learned a considerable amount, as I was not very familiar with IoT devices before this project.

If I could redo the project, I'd stick to a stricter schedule, I was unaware of the complexities coming into the project which caused a lot of frustration and was very time-consuming. I didn't get to the final level of investigation I had hoped, however, I was happy with the result regardless. I would have liked to reach out to more contacts and companies including the university to further enhance my knowledge and increase the data available.

9. References

ARM. (n.d.-a). *Cortex-M4*. Retrieved May 3, 2024, from https://developer.arm.com/Processors/Cortex-M4

ARM. (n.d.-b). *Cortex-M23*. Retrieved May 3, 2024, from https://developer.arm.com/Processors/Cortex-M23

Arslan, H. (2016). *Figure 1. XMPP client-server architecture*. ResearchGate. https://www.researchgate.net/figure/XMPP-client-server-architecture_fig1_312243632

Binwalk | Kali Linux Tools. (n.d.). Kali Linux. Retrieved May 3, 2024, from https://www.kali.org/tools/binwalk/

Carbone et al. (2011, January). *An In-Depth Analysis of the Cold Boot Attack: Can It Be Used for Sound Forensic Memory Acquisition?* https://apps.dtic.mil/sti/citations/ADA545078

Cloudflare. (n.d.). *What is the Mirai Botnet?* Retrieved April 25, 2024, from https://www.cloudflare.com/learning/ddos/glossary/mirai-botnet/

Connectivity Standards Alliance. (n.d.). *Zigbee | Complete IOT Solution*. CSA-IOT. Retrieved May 2, 2024, from https://csa-iot.org/all-solutions/zigbee/

David Wheeler, JC Wheeler, & Damilare Fabemi. (2020). *The IoT Architect's Guide to Attainable Security and Privacy*. Routledge & CRC Press. https://www.routledge.com/The-IoT-Architects-Guide-to-Attainable-Security-an

d-Privacy/Fagbemi-Wheeler-Wheeler/p/book/9781032475233

Doran, M. (2022, October 24). *Understanding And Implementing Hardware Root of Trust*. DornerWorks. https://www.dornerworks.com/blog/hardware-root-of-trust/

Eljiona Zanaj, Giuseppe Caso, Luca De Nardis, & Alireza Mohammadpour. (2021). *Figure 2. Zigbee Network Architecture.* ResearchGate. https://www.researchgate.net/figure/Zigbee-Network-Architecture_fig1_350282737

eurostat. (2022, May). *File:Enterprises using IoT by purpose and size class, EU, 2021 (% of enterprises using IoT).png*. https://ec.europa.eu/eurostat/statistics-explained/index.php?title=File:Enterprises_using_IoT_by_purpose_and_size_class,_EU,_2021_(%25_of_enterprises_using_IoT).png

Finding encryption keys—LibreTiny. (n.d.). Retrieved May 3, 2024, from https://docs.libretiny.eu/docs/platform/beken-72xx/keys/

Gary Burt. (2004). *Processor Architectures*. https://redirect.cs.umbc.edu/courses/undergraduate/CMSC391/summer04/burt/lectures/arch/architectures.html

ghsecuritylab. (2020). *Ghsecuritylab/tysdk_for_bk7231t* [Computer software]. https://github.com/ghsecuritylab/tysdk_for_bk7231t (Original work published 2020)

Google. (n.d.-a). *A home that knows how to help.* Retrieved May 2, 2024, from https://home.google.com/intl/en_uk/welcome/

Google. (n.d.-b). *Security updates and security validation results for Google Nest devices—Help*. Retrieved May 2, 2024, from

https://support.google.com/product-documentation/answer/10231940

Google Bug Hunters. (n.d.). Retrieved May 2, 2024, from https://bughunters.google.com/

Halderman, J. A., Schoen, S. D., Heninger, N., Clarkson, W., Paul, W., Calandrino, J. A., Feldman, A. J., Appelbaum, J., & Felten, E. W. (2009). Lest we remember: Cold-boot attacks on encryption keys. *Communications of the ACM*, *52*(5), 91–98. https://doi.org/10.1145/1506409.1506429

Home App. (n.d.). Apple (United Kingdom). Retrieved May 2, 2024, from https://www.apple.com/uk/home-app/

Home Assistant. (n.d.). *Home Assistant*. Home Assistant. Retrieved May 2, 2024, from https://www.home-assistant.io/

Home automation with the SmartThings App | SmartThings. (n.d.). Home Automation with the SmartThings App | SmartThings. Retrieved May 2, 2024, from https://partners.smartthings.com/smartthings-app

ISO. (2019). *ISO/IEC 27701:2019(en), Security techniques—Extension to ISO/IEC 27001 and ISO/IEC 27002 for privacy information management—Requirements and guidelines*. https://www.iso.org/obp/ui/#iso:std:iso-iec:27701:ed-1:v1:en

Ivan, C., Vujic, M., & Husnjak, S. (2016). Classification of Security Risks in the IoT Environment. In B. Katalinic (Ed.), *DAAAM Proceedings* (1st ed., Vol. 1, pp. 0731–0740). DAAAM International Vienna. https://doi.org/10.2507/26th.daaam.proceedings.102

Ivan Lee. (2024). *What is AMQP Protocol? All you need to know*. https://www.wallarm.com/what/what-is-amqp/

Kim, D., Park, H., Yeo, I., Lee, Y. K., Kim, Y., Lee, H.-M., & Kwon, K.-W. (2024).

Rowhammer Attacks in Dynamic Random-Access Memory and Defense Methods. *Sensors (Basel, Switzerland)*, *24*(2), 592. https://doi.org/10.3390/s24020592

LibreTiny. (n.d.). *Libretiny-eu/ltchiptool* [Python]. LibreTiny. Retrieved May 3, 2024, from https://github.com/libretiny-eu/ltchiptool (Original work published 2022)

LoRa Alliance. (n.d.). What is LoRaWAN® Specification. *LoRa Alliance®*. Retrieved May 2, 2024, from https://lora-alliance.org/about-lorawan/

Marcel, J. (2024, February 6). *The Role of Bluetooth Technology in the Ambient IoT*. Bluetooth® Technology Website. https://www.bluetooth.com/blog/the-role-of-bluetooth-technology-in-the-ambient-iot/

NIST. (2016, August 31). *Roots of Trust | CSRC | CSRC*. CSRC | NIST. https://csrc.nist.gov/Projects/Hardware-Roots-of-Trust

NIST. (2018a). *NVD - CVE-2017-12712*. https://nvd.nist.gov/vuln/detail/CVE-2017-12712

NIST. (2018b). *NVD - CVE-2017-12716*. https://nvd.nist.gov/vuln/detail/CVE-2017-12716

Sidhu, S., Mohd, B. J., & Hayajneh, T. (2019). Hardware Security in IoT Devices with Emphasis on Hardware Trojans. *Journal of Sensor and Actuator Networks*, *8*(3), Article 3. https://doi.org/10.3390/jsan8030042

Statista. (n.d.). *IoT connected devices worldwide 2019-2030 | Statista*. Retrieved April 25, 2024, from https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/

Tawalbeh, L., Muheidat, F., Tawalbeh, M., & Quwaider, M. (2020). IoT Privacy and Security: Challenges and Solutions. *Applied Sciences*, *10*(12), Article 12.

https://doi.org/10.3390/app10124102

United Kingdom Government. (2018). *Data Protection Act 2018* [Text]. Statute Law Database. https://www.legislation.gov.uk/ukpga/2018/12/contents

Wedd, M. (2020, February 21). What is Cellular IoT? *IoT For All*. https://www.iotforall.com/what-is-cellular-iot

WiFi Alliance. (n.d.). *Internet of Things | Wi-Fi Alliance*. Retrieved May 2, 2024, from https://www.wi-fi.org/discover-wi-fi/internet-things

XMPP. (2024). *XMPP*. https://xmpp.org/

Z-Wave Alliance. (n.d.). *Learn About Z-Wave*. Z-Wave. Retrieved May 2, 2024, from https://www.z-wave.com/learn

Appendix

Appendix A: Project Initiation Document

School of Computing
Final Year Research Project

Project Initiation Document

Charles Hamerston-Budgen

BSc Computer Science

An analysis of the orchestration, maintenance and security of IoT devices.

1. Basic details

Student name:	Charles Hamerston-Budgen
Draft project title:	An analysis of the orchestration, maintenance and security of IoT devices.
Course and year:	Final Year BSc Computer Science
Project supervisor:	Dr Gail Ollis
Client organisation:	N/A
Client contact name:	N/A

2. Degree suitability

My project is suitable for my degree as IoT is a rapidly evolving field. I believe that this project is relevant as aspects of IoT such as device implementation, security and networking overlap with modules such as distributed systems, ethical hacking, security and cryptography and networking.

I also believe that as IoT devices are becoming more complex with advancements such as artificial intelligence, that this project is suitable for a computer science degree.

3. The project environment and problem to be solved

I am aiming this report at the technical users of IoT devices, as the research will be focused around the use of IoT within homes, workplaces and also in other industrial applications. I aim to research how devices communicate, proprietary, non-proprietary and DIY (do it yourself) devices.
I also aim to research how the devices can be used, maintained or deployed without vendor lock-in, as well as across brands/models.

I'd like to also explore the current state of IoT devices in different applications as well as the usage of IoT at scale, along with the problems that come with it, such as device management and maintenance.

IoT devices are becoming more popular, for example within a home there may be smart-meters, bulbs, thermometers, assistants as well as other sensors. As the industry and demand are increasing, there is larger buy-in from major manufacturers such as Arm not only for device production, but also management for larger enterprises.

I'd also like to explore the use of AI within these devices (AIOT). AI is advancing quickly, there have been a large number of AI applications such as real-time monitoring, supply chain management, predictive maintenance and autonomous driving.

These new areas bring a new set of issues especially in security, before they can become safe and sustainable for long term usage.

4. Project aim and objectives

Aim:
Research and analyse the key issues within deploying, maintaining, utilising and securing a mixture of IoT devices.

The background for this aim is that IoT devices are relatively complex devices, which can cause frustration and other issues to the end users.

Furthermore, manufacturers don't follow the same communication/security/maintainability standards, which can cause devices to only work with additional services such as cloud subscriptions.
IoT devices may also be vulnerable to attackers due to the lack of protections and encryption built in, as well as relying on the user to update any firmware / other software which cannot be done remotely over the internet.
IoT devices may lack repairability and support from the vendor which would create more e-waste.

Key Objectives:

- Research how IoT devices (proprietary, non-proprietary, DIY) communicate with each other.
- Research 'home hubs' for self hosted IoT solutions.
- Research the complexities with maintaining IoT devices at both small and large scales.
- Research the complexities in deploying IoT devices at both small and large scales.
- Create POC (Proof of concept) for DIY IoT solutions.

Additional Objectives:

- Explore the usage of AI/ML models within the IoT space to further analyse maintenance and security of devices.
- Design / Implement POCs (Proof of concept) for utilising machine learning models to improve network security.
- Design / Implement POCs (Proof of concept) for device management.

5. Project constraints

- Project report submission date.
- Proprietary device using proprietary communication/hardware measures.
- Finding out business utilisation of IoT devices.

6. Facilities and resources

- Computing resources
    - Laptop

- Networking resources
- IoT Devices
  - Proprietary
  - Non-proprietary
  - DIY
    - Raspberry PI
    - Sensors
- Software
  - Open-source solution
  - Commercial solution

7. Log of risks

No	Description	Likelihood (high, medium, low)	Impact	Mitigation/Avoidance
1	Loss of IoT devices.	Medium	Delay in the project timeline.	Multiple IoT devices.
2	Laptop failure.	Medium	Delay in the project timeline.	Backup to the cloud.
3	DIY device issues.	Medium	Cannot research DIY.	Start as soon as possible to find issues sooner.
4	Unable to access business data.	High	Cannot talk about business usage of IoT.	Remove the business sections/reference other data.
5	Lack of participants.	Medium	Unable to get recent data.	Reference previous data.
6	Life issues/Illness.	Low-High	Delays to the project.	Speak to Gail and come up with an alternative plan.

8. Project deliverables

- Primary research of users, device manufacturers alongside the final report.
- Secondary research of previous work, statistics alongside the final report.
- Final report/write up.
- Analysis of different types of communications.
- Datasets (Distribution of devices, types of devices, data they provide).
- Proof of concepts (if applicable).
- DIY devices (if applicable).

9. Project approach

My project will be managed by an Agile/Scrum methodology. I plan to use sprint planning, backlog refinement and periodic reviews throughout my project to make sure the project stays focused on the aim and is on track to complete the objectives.

I plan to remove stakeholder reviews and the sprint reviews as they won't be needed for this project and instead have periodic retrospectives to replace these.

I also plan to explore the use of Kanban and possibly have a hybrid between these depending on what works best for my project.

For my primary research I will collect data from surveys about how devices are used and how many devices are deployed and the type of devices deployed.

For my secondary research I plan to use previous research from tools such as Google Scholar, as well as previous reports.

10. Project tasks and timescales

No	Stage	Dates	Main Tasks
1	Project startup	18/09/23 - 06/10/23	Choose topic and supervisor
2	PID	06/10/23 - 20/10/23	Complete PID
3	Ethics	06/10/23 - 08/12/23	Complete Ethics review
4	Literature review	20/10/23 - 01/12/23	Complete Literature review
5	Methodology/Data collection	15/11/23 - 01/01/24	Complete data collection
6	Analysis	01/01/24 - 15/01/24	Refine problem
7	Design/Implementation	01/01/24 - 01/03/24	Complete a MVP/POC
8	Evaluation	01/02/24 - 01/04/24	Write up the evaluation
9	Discussion/Conclusion	01/03/24 - 25/04/24	Write up conclusion
10	Review	25/04/24 - 03/05/24	Review and proofread
11	Technical investigations	20/10/23 - 01/01/24	Investigate current technical products/solutions

I plan to refine and triage the objectives after starting the research on these around stage 5 (Methodology and data collection).

11. Supervisor meetings

I plan to have weekly, 30 minute meetings throughout the duration of the project with my supervisor.

12. Legal, ethical, professional, social issues

Legal:

- Data protection (GDPR).
- In the case of proprietary devices, the solution(s) shouldn't manipulate the communication to / from remote servers.
- Adhere to not using any copyrighted, protected designs.
- Adhering to the use of the software licences.

Ethical:

- Adhering to any privacy in terms of user data.
- Any data should be protected / anonymized.
- Respecting intellectual property.
- The devices could create e-waste as they are single purpose devices.
- Must complete the university ethics requirements.

Professional:

- Any potential solutions/recommendations should follow the IEEE standards for secure transmissions.

Social:

- This research will explore giving users back the ownership of their devices.

13. Permission

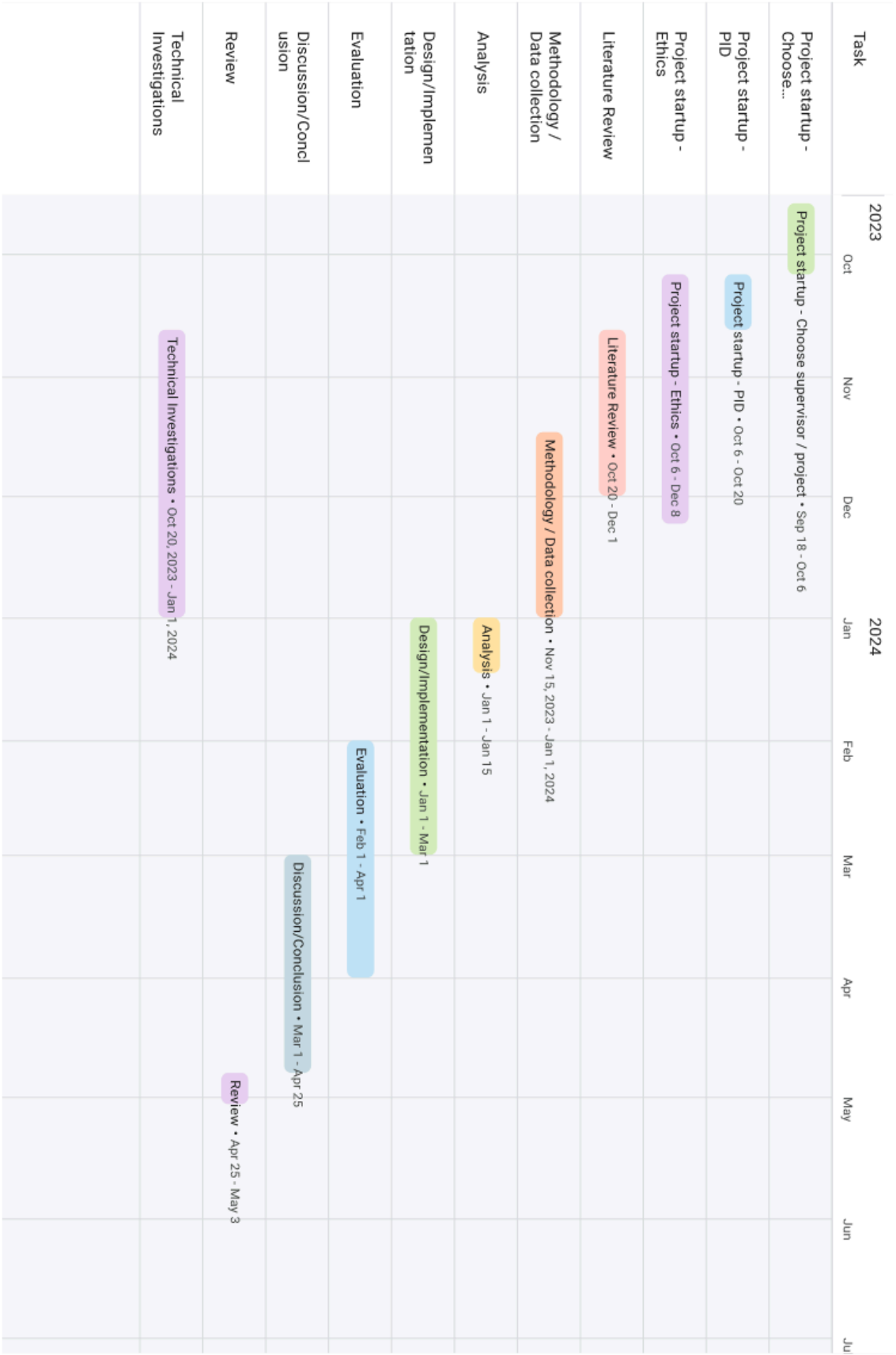
Please tick

☑ I give permission for my PID to be made available to other students as examples of previous work.

☐ I do not give permission for my PID to be made available to other students as examples of previous work.

Date: 19/10/2023

Appendix A: Gantt chart



Appendix B: Ethics Certificate

Certificate of Ethics Review

Project title: An analysis of the orchestration, maintenance and security of IoT devices.

REMOVED

You must download your referral certificate, print a copy and keep it as a record of this review.

The FEC representative(s) for the **School of Computing** is/are **Elisavet Andrikopoulou, Kirsten Smith**

It is your responsibility to follow the University Code of Practice on Ethical Standards and any Department/School or professional guidelines in the conduct of your study including relevant guidelines regarding health and safety of researchers including the following:

- University Policy
- Safety on Geological Fieldwork

It is also your responsibility to follow University guidance on Data Protection Policy:

- General guidance for all data protection issues
- University Data Protection Policy

Which school/department do you belong to?: **School of Computing**
What is your primary role at the University?: **Undergraduate Student**
What is the name of the member of staff who is responsible for supervising your project?: **Gail Ollis**
Will you gather data about people (e.g. socio-economic, clinical, psychological, biological)?: No
Will you gather data from people about some artefact or research question (e.g. opinions, feedback)?: Yes Confirm whether and explain how you will use participant information sheets and apply informed consent.: For interviews, participants will have a information sheet, consent form and opportunity to ask questions, and to withdraw up to anonymity.

Online surveys, will also provide consent and information sheets.
Confirm whether and explain how you will maintain participant anonymity and confidentiality of data collected: Anonymous data will be stored, the data will be stored by a password.
Will the study involve National Health Service patients or staff?: No
Do human participants/subjects take part in studies without their knowledge/consent at the time, or will deception of any sort be involved? (e.g. covert observation of people, especially if in a non-public place): No
Will you collect or analyse personally identifiable information about anyone or monitor their communications or on-line activities without their explicit consent?:

No
Does the study involve participants who are unable to give informed consent or are in a dependent position (e.g. children, people with learning disabilities, unconscious patients, Portsmouth University students)?: No
Are drugs, placebos or other substances (e.g. food substances, vitamins) to be administered to the study participants?: No
Will blood or tissue samples be obtained from participants?: No
Is pain or more than mild discomfort likely to result from the study?: No
Could the study induce psychological stress or anxiety in participants or third parties?: No Will the study involve prolonged or repetitive testing?: No
Will financial inducements (other than reasonable expenses and compensation for time) be offered to participants?: No
Are there risks of significant damage to physical and/or ecological environmental features?: No
Are there risks of significant damage to features of historical or cultural heritage (e.g. impacts of study techniques, taking of samples)?: No
Does the project involve animals in any way?: No
Could the research outputs potentially be harmful to third parties?: No

Could your research/artefact be adapted and be misused?: Yes
Identify any risks associated. How do you plan to minimise risks?: Potential misuse of any security vulnerabilities that are discovered through the research involving business and other domestic devices. Vulnerabilities are not the main point of research, but are flagged for in case any are found. If they are I won't provide specific details for exploitation such as code, methods.
Will your project or project deliverables be relevant to defence, the military, police or other security organisations and/or in addition, could it be used by others to threaten UK security?: No
Please read and confirm that you agree with the following statements: I confirm that I have considered the implications for data collection and use, taking into consideration legal requirements (UK GDPR, Data Protection Act 2018 etc.), I confirm that I have considered the impact of this work and and taken any reasonable action to mitigate potential misuse of the project outputs, I confirm that I will act ethically and honestly throughout this project

REMOVED

Appendix C: Participant Information Sheet

REMOVED

Participant Information Sheet

Name and Contact Details of Researcher(s): REMOVED

Name and Contact Details of Supervisor: REMOVED

1. Invitation

I am a student at the University of Portsmouth undertaking a research final year project based around the organisation, management and security of IoT devices. I'd like to find out more about what devices are used, how they are used, and any concerns about them.

I would like to invite you to take part in our research study.

Joining the study is entirely up to you, before you decide I would like you to understand why the research is being done and what it would involve for you.

Taking part will take about 10 minutes.

Please ask/contact us if you have any questions.

2. **Study Summary**

2.1 Survey

The purpose of this study is to collect data around IoT devices, and how they are currently being utilised across both home and business settings, you may be asked to:

- Answer some questions about:
    - The usage of IoT devices.
    - The maintenance of IoT devices.
    - What devices are used / their purpose.
    - Any other concerns you may have about the devices.

2.2 Interviews

You may be asked for a further interview, provided on the survey, or by a direct contact. This will allow us to further discuss the points raised within the survey, or to ask the questions contained in the survey.

The interview is **optional**. (For further information please refer to section 3)

The interview may take between 30 and 60 minutes and will cover questions from the survey in more detail.

The interview may also be audio recorded, if this is the case you will be asked to fill out the consent form before the interview, the recording will be deleted upon completion of the project, and you will remain anonymous throughout.

The interview may be in-person or online, depending on the location of the interviewer and interviewee.

3. What data will be collected and / or measurements taken?

I would like to collect information about IoT devices (e.g maintenance, security concerns, frustrations, usage), I'd also like to collect the environment in which the devices are used (home or business).

All data stored about you will be fully anonymised. Data will be stored securely on a google drive that can only be accessed by members of the research team. It will be deleted after the end of the project.

During an interview (if applicable) more in depth data will be collected about the deployment of IoT devices, security, architectures, usage, and any other information which may be relevant to the research this will be used to gain a further understanding of how these devices are utilised.

All of the information collected during an interview will remain anonymous.

4. Do I have to take part?

No, taking part in this research is entirely voluntary. You can withdraw any time during the study for any reason.

5. Expenses and payments

There is no payment for taking part

6. What if there is a problem?

If you have a query, concern or complaint about any aspect of this study, in the first instance you should contact the researcher if appropriate. Please contact the supervisor listed if you have a complaint.

Thank you

Thank you for taking time to read this information sheet and for considering volunteering for this research.

Appendix D: Consent Form

CONSENT FORM

Title of Project: An analysis of the orchestration, maintenance and security of IoT devices.

Name and Contact Details of Researcher(s): REMOVED

Name and Contact Details of Supervisor (if relevant): REMOVED

University Data Protection Officer: Samantha Hill, 023 9284 3642 or information-matters@port.ac.uk

Ethics Committee Reference Number: REMOVED

	Please initial box
1. I confirm that I have read and understood the information sheet v1.0 dated 17/11/2023 for the above study. I have had the opportunity to consider the information, ask questions and have had these answered satisfactorily.	☐
2. I understand that my participation is voluntary and that I am free to withdraw at any time without giving any reason up to anonymity.	☐
3. I understand that data collected during this study will be processed in accordance with data protection law as explained in the Participant Information Sheet v1.0 dated 17/11/2023.	☐
4. I consent for my interview to be audio recorded. The recording will be transcribed and analysed for the purposes of the research any audio recordings will be destroyed at the end of the project.	☐
5. I consent to verbatim quotes being used in publications; I will not be named but I understand that there is a risk that I could be identified.	☐
6. I agree to take part in the above study.	☐

Name of Participant: **Date:** **Signature:**

Name of Researcher: **Date:** **Signature:**

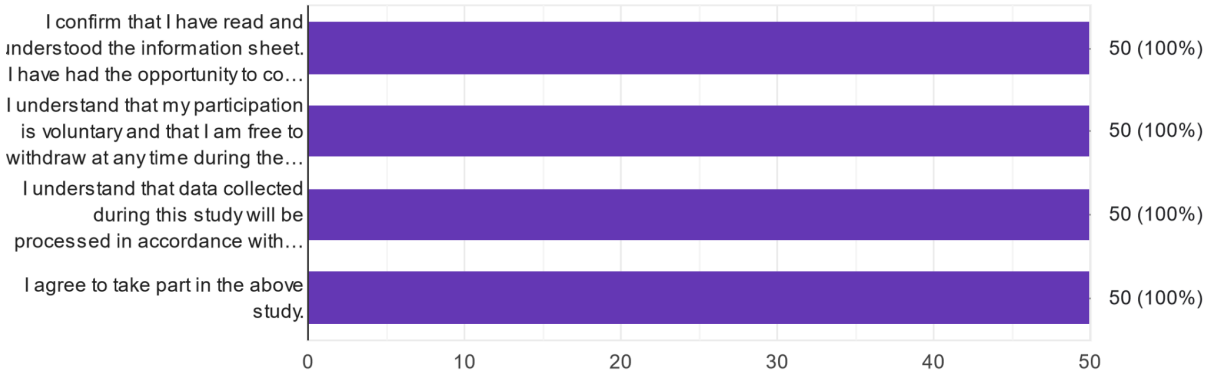
***Note**: When completed, one copy to be given to the participant, one copy to be retained in the study file*

Appendix E: Survey Data

Consumer questions

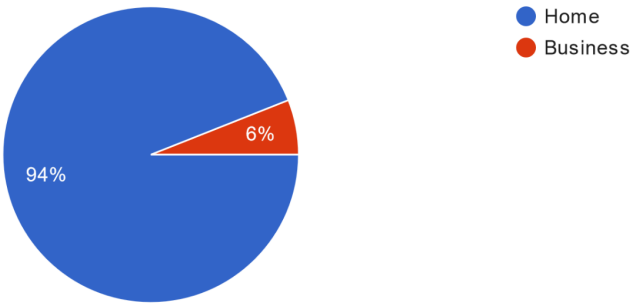
I confirm that I have read and understood the information sheet. I have had the opportunity to consider the information, ask questions and have had these answered satisfactorily.

50 responses



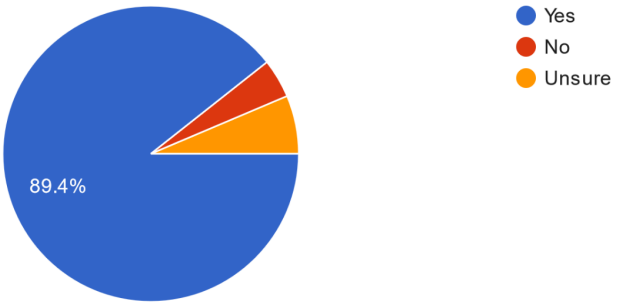
What kind of information will you be providing in this form? (you can submit this form multiple times if you'd like)

50 responses



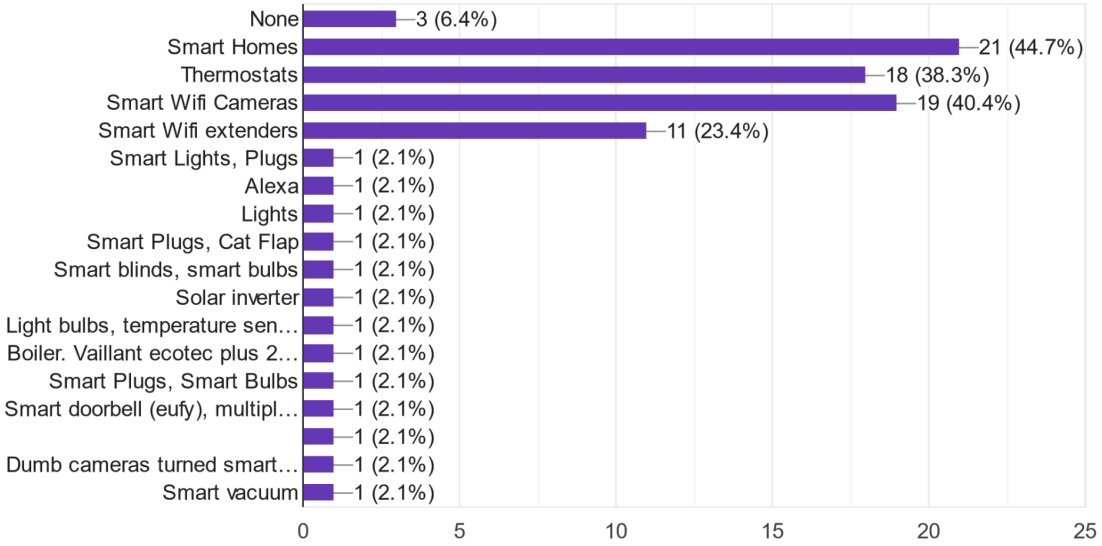
Do you use IoT devices within your home?

47 responses



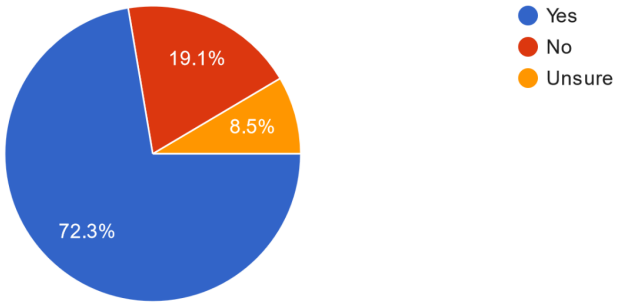
What types of devices are used at home? (e.g Smart home, thermostat, cameras, Wi-Fi extenders)

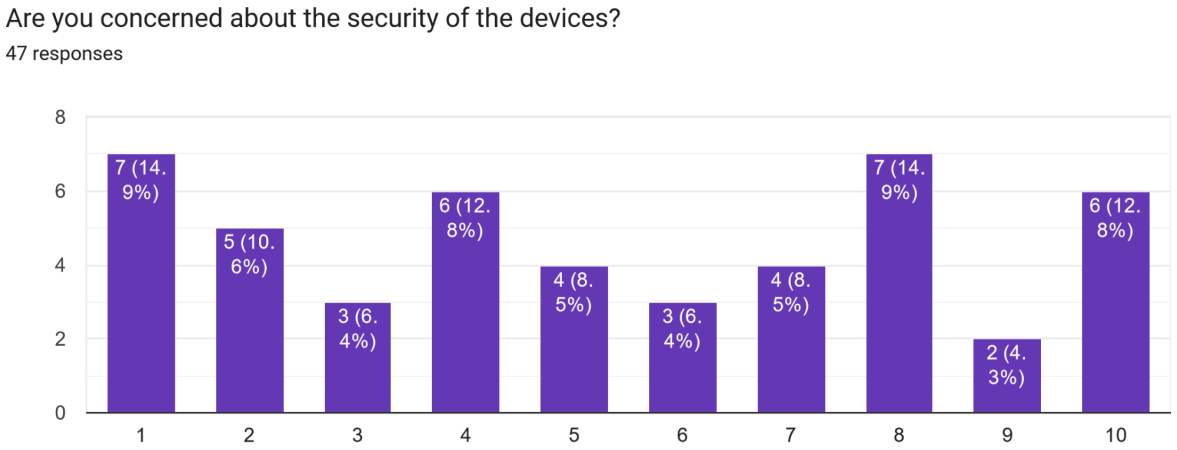
47 responses



Do you maintain the devices used at home? (e.g. software / firmware updates)

47 responses





What tools do you use to manage these devices? (e.g Google Home, Home assistant)

Home assistant
Alexa assistant
Amazon Alexa
Unifi and smart home
Google Home, App associated with smart device
Google
In built app updates etc
Home assistant
N/A
N/A
Amazon echo dot
Alexa
Unsure
Kasa (TP-Link's smart home management app), Nest app and Sureflap app for cat flap.
In built security features

Home assist
Amazon app
Google Home
Hive
Norton
Norton
None

No tools
Apps from vendors. Google Home. Amazon Alexa app.

Tahome, Ring, Wiz v2
Hive and Google Home
My phone
Google Assistant/Arduino
Google home and home assistant
Google Home, Alexa and the Govee app
Windstream
Proprietary interface (web based and mobile app) (Arlo security cameras, doorbell and hub).
Apple Home app
Phone and google home
Google home
Google Home, Ikea Home Smart, Apple Home
Eufy app
Google home
Apple home, Amazon
Samsung smart things
HomeKit, HomeBridge Hub, Scrypyed
Home assistant
Mobile app
Google home, individual apps

Inbuilt ubiquity dashboard
Alexa, Home Assistant
Google Home

Any other notes you'd like to add?

Alexa devices
No

"I do not use IoT devices in my home expressly for the cyber security concerns of cameras / microphones in my home. I Would however consider a VLAN setup with something like HomeAssistent - provided it was 100% FOSS and never left my network"

As far as I'm concerned, IoT devices are just a liability and a potential complete breach of privacy. You really have no idea what is happening in these devices, or how exactly they are connecting to the internet. They also aren't exactly particularly useful; I have no need for "smart light bulbs" for example, and simply pressing a

switch is far more convenient then having to open a specific application on my phone.

None

None

There is no way I would use Alexa or Google Home in my home.

None

Given the performance of both the prduct (hardware, firmware and software), I'm inclined to doubt that they have enough focus / resources dedicated to security.

I’m a software dev myself but prefer to have a dumb home. Happy to have the boiler connected but it all works without the internet, including the non smart boiler thermostat.

Would love to set up Home Assistant but it is a lot of work, especially to integrate with Google Assistant/etc

Using home assistant gives me piece of mind because the services I use are local to my home network and not cloud dependent.

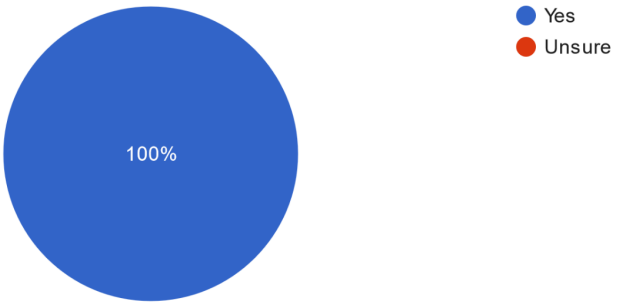
Nope :)

I use separate VLAN/Wifi for IoT stuff with firewalling between that and the main network.

Business questions

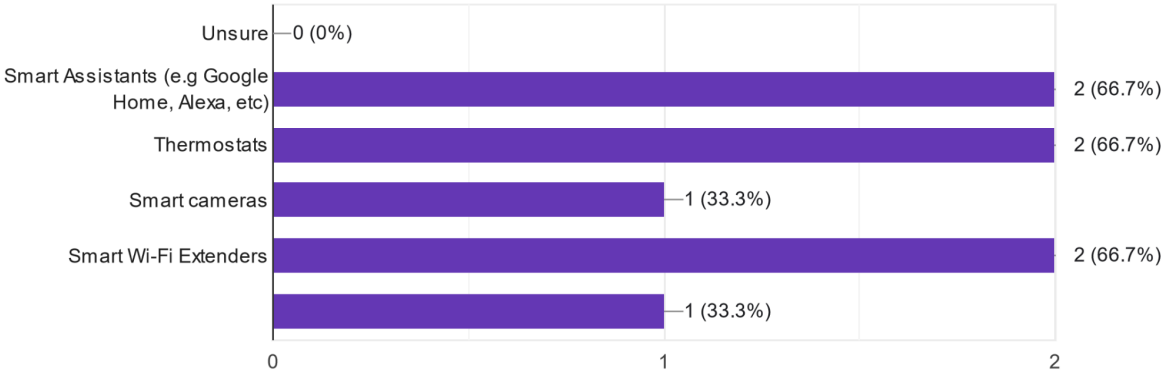
Do you use IoT devices within your workplace?

3 responses



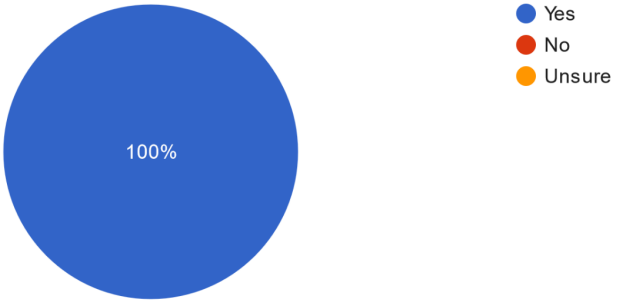
What types of devices are used at your workplace? (e.g Smart home, thermostat, cameras, wifi extenders)

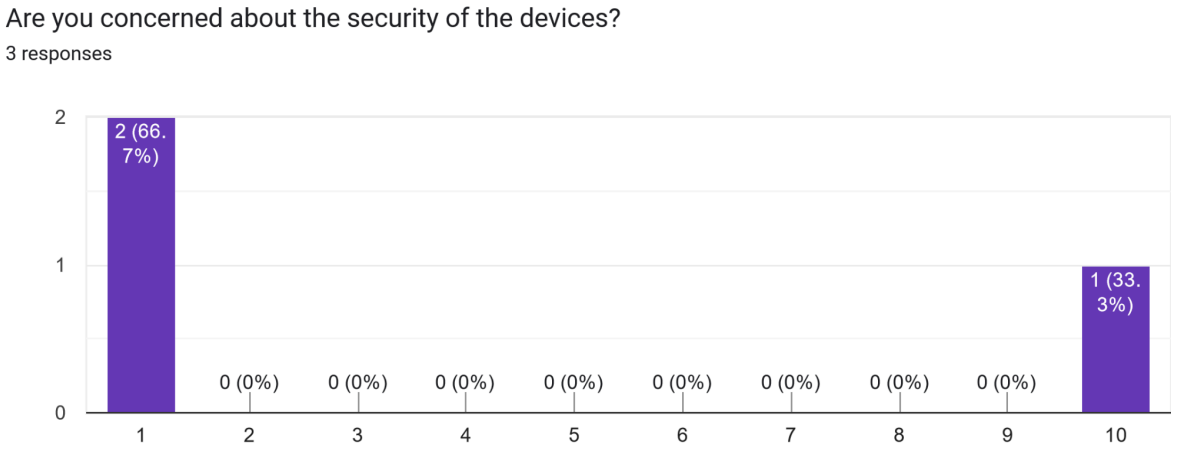
3 responses



Are the devices maintained in the workplace? (e.g. software / firmware updates)

3 responses





What tools do you use to manage these devices? (e.g Google Home, Home assistant, proprietary solutions)

As it's a business I have no control and so have nothing in place

Google home system and space

Home (iOS) + Xiaomi Smart Home Hub 2

Any other notes you'd like to add?

None

Appendix F: Helper Functions

```
import os
from Crypto.Cipher import AES

BLOCK_BYTES = 32

# Pad data to block size with null bytes
def pad_data(data):
    data_remainder = len(data) % BLOCK_BYTES
    print(data_remainder)
    if data_remainder != 0:
        remainder = BLOCK_BYTES - data_remainder
```

```
        data += b'\xFF' * remainder
    return data

dump = ''
with open('dump.bin', 'rb') as f:
    dump=f.read()

# Find the RBL locations within the raw dump
def find_containers_indexes():
    with open('./dump.bin', 'rb') as f:
        HEADER = b"RBL\x00" # Start of a block
        locations = []
        data = f.read()
        last_pos=0
        while True:
            try:
                index = data.index(HEADER, last_pos)
            except:
                break
            if(index != -1):
                locations.append(index)
                last_pos = index+1
        return locations

# Find the name, and size of the partition from header
def find_name_size(header):
    dec_arr = []
    for index in range(0,len(header),4):
        dec_arr.append(header[index:index+4])
    name = ""
    for index in range(2,12):
        name += dec_arr[index].decode('utf-8', 'ignore')
    name = name.replace('\x00', '')
    size = int.from_bytes(dec_arr[-2], 'little')
    return name, size

# Pretty print containers, decrypt partition
def container_out(containers):
    with open('dump.bin', 'rb') as f:
        raw_data = f.read()
        print('Containers: ')
        index = 1
        for container in containers:
```

```
            header = raw_data[container:container+96] # 96 bytes for
header
            name, size = find_name_size(header)
            print(f'Container: {index}, index: {container}, name:
{name}, size: {size} ')
            index += 1
        raw_data_length = len(raw_data)
        end = raw_data_length - 256
        while(end > 0):
            if(raw_data[end] != 0xFF):
                break
            end -= 1
        raw_data_length = end
        temp = b''
        for block in range(0, raw_data_length, 34):
            temp += raw_data[block:block + 32]
        with open('dump_temp.bin', 'wb') as temp_file:
            temp_file.write(temp)
        os.system("./encrypt dump_temp.bin")


# Returns a location and subset of the raw data which is the storage
block.
# Uses the data offset to capture the block and index of block
(location)
def find_storage():
    with open('./dump.bin', 'rb') as f:
        dump = f.read()
        aes_obj = AES.new(key="",mode=AES.MODE_ECB)
        magic_value = aes_obj.encrypt(b'\xFF' * 16)
        location = dump.index(magic_value) - 32
        offset = location + 32769
        return location, dump[location:offset]

# Uses the data offset to capture the block from the index of block
(location)
def dump_storage():
    with open('./dump.bin', 'rb') as f:
        dump = f.read()
        aes_obj = AES.new(key="",mode=AES.MODE_ECB)
        magic_value = aes_obj.encrypt(b'\xFF' * 16)
        location = dump.index(magic_value) - 32
        with open('dump_storage.bin', 'wb') as out:
```

```
            offset = location + 32769
            out.write(dump[location:offset])

# Key mix, a,b,c max length, roll over 256 as byte is 0,256
def create_key(key: bytes):
    new_key = bytearray(16)
    for i in range(0, 16):
        new_key[i] = ((KEY1[i] + KEY2[i]) + key[i]) % 256
    return AES.new(key=new_key, mode=AES.MODE_ECB)

# Decrypt storage, print values
def decrypt_storage(data, aes):
    data_blocks = []
    for i in range(0,len(data),4096):
        data_blocks.append(data[i:i+4096])
    data_block = data_blocks[0]
    first_decrypt = aes.decrypt(data_block)
    key = first_decrypt[8:24]
    aes = create_key(key)
    count = 0
    while True:
        count+=1
        enc_block = data_blocks[count]
        dec_block = aes.decrypt(enc_block)
        print(dec_block)
        break
```