School of Computer Science and Engineering

Faculty of Engineering

The University of New South Wales

Requirements to theses submitted in the Faculty of Engineering

by

Andrew Jin-Meng Wong

Thesis submitted as a requirement for the degree of

Bachelor of Engineering in Computer Engineering

Submitted:	December 2021
Supervisor:	Prof. Richard Buckland
Student ID:	z5206677
Topic ID:	”Smart” Vacuum Cleaners

An Audit Into The Security and Integrity of IoT Systems

Table of Contents

Abbreviations

- (e)MMC - (Embedded) Multi Media Card - i.e. flash storage
- IoT - Internet of Things
- LAN - Local Area Network
- MITM - Man In The Middle
- PII - Personally Identifiable Information
- SoC - System on Chip
- SSL - Secure Sockets Layer
- UGC - User Generated Content
- WAN - Wide Area Network (i.e. the internet)

Chapter 1

Introduction

Consumer grade Internet of Things (IoT) devices have become widely adopted with continuously growing demand. With demands for such devices growing by 12% each year (Research & Markets 2021), this AU$130B industry has cordially invited thousands of households to invest in smart devices such as light bulbs, fans, televisions and fridges. Giving the abundance and affordability of these products, IoT devices have become an integral part of many homes, where 4 in 5 consumers would be more inclined to choose a property over another if the former were to have such technologies (Brown 2015).

Although convenient, these devices come with hidden costs and risks. Behind the seemingly ‘simple’, ‘smart’ and ‘secure’ product features that attract the general populus lies a hidden complex network of services and devices, where functionality is often obscured and private. Without the transparency of what data is being sent, and of where that data is being sent to, consumers inevitably pay for convenience with not only their money but with their privacy and security (Mehic et al. 2019).

Whilst manufacturers and vendors claim to be secure and/or confidential in how they treat **UGC** and **PII**, it lies evident from various incidents that we cannot completely trust such claims. From involuntary exposure of leaked Facebook user data (Abrams 2021), to rumours of corporations monetising user data without consent (Jones 2017), there lies an equal need for consumers to understand the terms of service to which they agree to, but additionally for companies to be audited against those very same terms of service.

The infrastructural security and product security of IoT devices must also be scrutinised, given the rapid product lifecycle of IoT developments (Giese 2021). As security is often not a sellable feature in contrast to new products and most mistakenly, convenience, proper and wholistic security precautions are often overlooked by companies who are more concerned with profits and high return on investments. Consequently, the prevalence of malicious actors in the cyberworld is alarming, where the overall lack of security awareness between consumers invites target devices to be easily accessed with default passwords or through unpatched vulnerabilities[1].

The black-box nature of IoT network communication raises both privacy and security concerns that may often be overlooked or trivialised in exchange for convenience. This this will involve the audit of an internet-connected robotic vacuum cleaner (Roborock S6) to assess the internal operations and nature of data that is transmitted, as to investigate any potential vulnerabilities that may render the device insecure.

The outline of this thesis is as follows: In Chapter 2, we study the motivations behind wanting to audit the privacy and security of IoT systems. In Chapter 3, we will review existing research, results and methods that comprise the current state of the art of security research on robot vacuum cleaners. Gaps in existing research will then be formalised through the thesis plan in Chapter 4. Finally we will conclude this report in a discussion of preliminary results that will be carried forwards into the later stages of research

[1] https://www.shodan.io/search?query=webcam

Chapter 2

Background

2.1 The widespread availability of IoT

The consumer market has experienced a large influx of IoT devices, largely attributed to the presence of IoT manufacturers who offer white-label partnerships with resellers to provide “custom” products. Through these partnerships, vendors buy into the IoT manufacturer’s ecosystem - namely being the product itself, the companion smartphone application, and the cloud infrastructure supporting backend communications - all without requiring vendors to posses any knowledge or understanding of how to design, develop nor manufacture the IoT products that they sell.

There are concerns regarding the privacy and ownership of user data that is transmitted, as vendors are often not in control of what information is transmitted, nor of how that information is used. This concern is potentially serious, as vulnerabilities within an IoT infrastructure would imply that devices and data from customers of vendor products would too be vulnerable. Furthermore, the lifetime of a vendor business is not guaranteed. With the constant opening and closing of IoT vendors, the closure of the business from which an IoT product was purchased from might result in the eventual in-operability of the said device.

In the event that an IoT infrastructure suffers downtime or service instability, all white-labelled products too will also be inherently affected. Great trust must be placed in the infrastructure’s availability and reliability. However in conjunction with aforementioned privacy and security concerns, many concerned users have turned to internet-less and self-hosted automation systems such as HomeAssistant and OpenHAB. As evident in later reviewed works (See

Chapter 3), concerns for privacy and security has been a driving force for developers and hackers to research and develop software, to replace the out-of-the-box internet-dependent software, effectively decoupling devices from vendor services.

2.2 About The Company

Roborock is a Chinese company founded in Beijing that develops robotic cleaning appliances for households. In 2014, partnering with Xiaomi shortly after the opening of their business, Roborock releasedd a line of both affordable and premium smart robotic vacuum cleaners, with their first iteration the “Mi Home Robotic vacuum Cleaner” being released in September 2016. They have since released eleven other robotic vacuum cleaner models, each model offering new and improved features - such as the addition of a mop functionality and improved spatial object detection via LIDAR technologies.

In June 2019, Roborock released their flagship Roborock S6 vacuum cleaner (the focus of this thesis), which boasted reduced operating noise levels, and better overall cleaning performances. The Roborock S6 is powered by an Allwinner R16 SoC (ARM architecture) alongside an STM32 for auxiliary motor and sensor I/O. Depending on the region and firmware, the robot vacuum cleaner is powered by either Xiaomi Cloud or Tuya Smart cloud infrastructures, both which are market leaders in the consumer IoT industry. Whilst superseded by newer revisions such as the S6 MaxV and the S7, the S6 still remains largely popular and is still actively maintained by Roborock.

Chapter 3

Literature Review

3.1 Broad security study of Tuya-based devices

In 2018 the security research group VTRUST (2018) analysed a line of white-labelled IoT product revisions based from the IoT manufacturer Tuya. Despite claims of 'military-grade security', basic packet logging of network activity concluded that "the analysis of the 'smart' devices using this basic platform is generally frightening", with "serious [...] shortcomings". It was revealed that various **PII**, encryption keys and the device's serial number (used to target a specific device through remote commands) were insecurely transmitted over the network, allowing a malicious user on the same wireless network to eavesdrop on the communication. Furthermore during the initial setup and pairing of the IoT device, wireless credentials were also insecurely transmitted in plain text, allowing wireless network credentials to be observed.

VTRUST commented on the dangers of vendors selling white-label products, where anyone could become a so-called 'IoT company' regardless of whether they had "in-depth technical knowledge of IoT or IT security". As a result of the hands-free approach to security and privacy for both direct and indirect customers of the IoT platform, concerns are raised regarding the ease of possibly distributing maliciously modified devices, where firmwares could be tampered at any stage during the supply chain. It is important to recognise that most custom firmware releases or so-called "hardware hacks" originated from the desire to decouple hardware from online and official cloud services. These ventures effectually disconnect internet-reliant devices from the cloud, and limit their connectivity to a local server where communications are transparent and minimal.

As a result of many Tuya-powered devices using the widely popular Espressif ESP8266 SoC, VTRUST was able to exploit discovered vulnerabilities to perform over-the-air upgrades of custom firmware such as ESPhome and Tasmota. An automated flashing tool (`tuya-convert`) was released, allowing consumers to easily integrate these devices with local home automation software such as HomeAssistant. As a result of VTRUSTS's findings, the overall security posture of modern Tuya-powered devices has since improved[1], with implementations of local flash memory encryption and firmware signing measures during over-the-air firmware upgrades.

VTRUST's technical findings offer insights into methods of network-level security assessment highlighting how easily an individual could their own IoT company, and the possibility of reselling devices with modified firmware with malicious intent.

3.2 Broad security study of Xiaomi-based devices

Giese (2019) performed a security assessment over a broad range of Xiaomi's IoT products to examine the overall ecosystem security. Through different keystroke injection methods, UART commands and hardware fault injection techniques, Giese obtained shell access into various Xiaomi-powered devices. It was concluded that due to the enormous size of Xiaomi's ecosystem, it was difficult to enforce global security policies between the different vendor-provided plugins that continued to supported deprecated functions and APIs that were still being used by legacy devices. From this research, a Xiaomi cloud emulator was built, allowing for complete offline functionality and control over a large range of Xioami devices without internet connectivity. This research also paved the way for other third party software to be developed, such as Valeduto - which provides a feature-rich web interface to operate a robot vacuum cleaner.

He concluded that Xiaomi indeed treats their security concerns seriously, given their quick responses to reported security incidents and vulnerability reports. During this study, the security of the Roborock S6 vacuum cleaner was also assessed - albeit briefly. As such this thesis will not only further explore the security posture of the Roborock S6 since 2019, but to also audit the state of privacy of the device.

[1] https://www.heise.de/newsticker/meldung/Smart-Home-Hack-Tuya-veroeffentlicht-Sicherheitsupdate-4292028.html

3.3 Security study of smartphone applications

Jmaxxz (2016) investigated the security claims of a smart doorlock which boasted in bank-grade security, overall better security over conventional lock and key systems. These claimed were however invalidated, as various flaws within the smartphone application were discovered, allowing control over lock settings that were only protected by client-side checks. Consequently, modified request payloads containing elevated authorisation claims would be naively accepted by the server, allowing lock settings to be modified by a guest or limited user. Furthermore, various debugging menus were present in the production version of the smartphone application, allowing certificate pinning protections to be subverted. In addition, the privacy of the user was also questioned, as it was observed that door lock events and other identifiable information were being transmitted to a logging endpoint.

The vulnerabilities in the smart doorlock's own product security highlight the importance to verify claims that manufacturers may advertise. This study serves as an excellent example a poorly access control system, where novel methods of HTTP request tampering, hardcoded keys and insufficient keyspaces allow for the arbitrary privileged control of the smart doorlock. Subversion of HTTP Strict Transport Security (HSTS) and certificate pinning policies through system-wide tools[2], per-application patching[3] or accessible debug menus furthermore underlines that certificate pinning should not be relied upon to verify identity nor authority.

3.4 Analysis of similarities in IoT firmwares

Costin et al. (2014) performed a broad static analysis over a large number of firmware images to identify common patterns and similarities between firmwares of different product vendors. During the analysis of the 693 images, 38 new vulnerabilities were discovered, some which were present in the majority of images. A large number of hardcoded keys and credentials were also discovered that could render the IoT device or its infrastructural service vulnerable. To facilitate the similarity analysis of firmware images, where per-byte analysis techniques are nonsensical, tools like binwalk, ssdeep, and sdhash were employed - which helped to facilitate file exploration relative to their file type and architecture. To compare versions of the same binary across different firmwares, a tool called BinDiff was used, which would compare similarities and

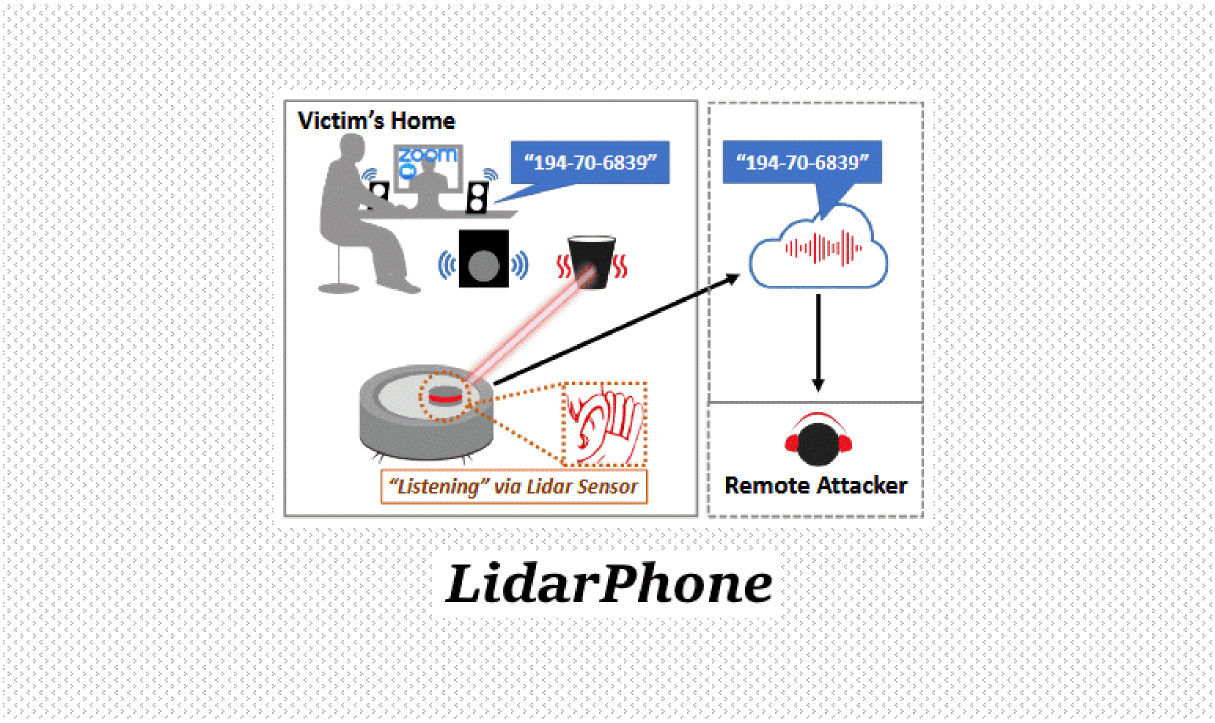
[2] https://github.com/nabla-c0d3/ssl-kill-switch2

[3] https://github.com/shroudedcode/apk-mitm

differences in disassembled code.

A large proportion of images shared similarities in code execution graphs, indicating that many vendors had simply reused and repurposed sample code (often available as part of a SoC's SDK). Whilst sample code itself is not often vulnerable, given the commonality of other vulnerabilities, concern is raised as to the vendor's technical capability and understanding of IoT systems and of security. The tools and methods to perform this firmware study are transferable to the scope of this thesis.

3.5 Side-channel application of LIDAR sensor measurements



As more and more IoT devices become online and sensor data is transmitted around the world, there are growing concerns to thoroughly investigate the extents of what data can be retrieved from the sensors. Given that the outputs of Light Detection and Ranging (LIDAR) sensors are intensity readings and distance measurements, Wei et al. (2015) developed a method to translate the intensity readings from the LIDAR sensor back into audio singles, when the LIDAR sensor was directed towards a surface near an audio source. This allowed speech to be identified from micro-vibrations within objects, prospectively raising concern regarding the privacy and confidentiality of sound in a sound-proof room.

This research has since been continued and tested on robot vacuum cleaners which too incorporate LIDAR sensors intended for spatial mapping. As general off-the-shelf LIDAR sensor units are used within vacuum cleaners, light intensity values were also returned by the sensor, and could be used to in a similar fashion to detect speech and sound (Sami et al. 2020). In the application of a robotic vacuum cleaner, light intensity values are considered a side-channel concern as those readings are not required for the operation of a vacuum cleaner. Despite integration limitations of sampling intensity values on a vacuum cleaner (i.e. accounting for the continuous rotation of the LIDAR sensor and noise floor as a result of the vacuum engine), a classification accuracy of 91% was achieved when extracting sensitive data such as digits of a credit card.

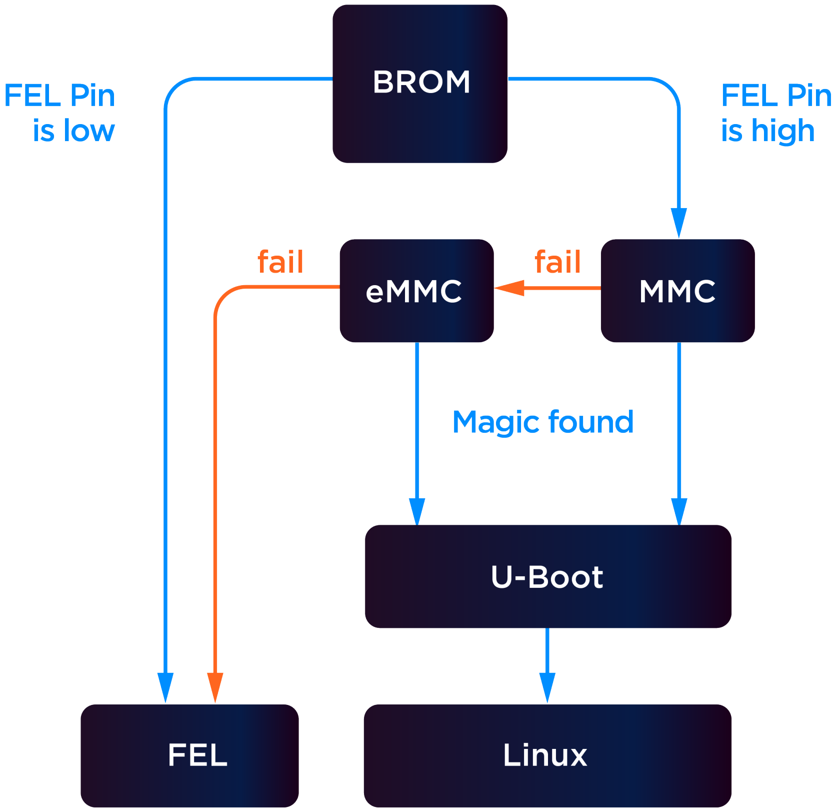
Whilst this thesis will not pursue the exploration of sensor data analysis, these two studies offer potential future research areas on privacy concerns surrounding robot vacuum cleaners, as newer revisions of smart devices become continually equipped with more accurate and feature-rich sensor.

3.6 Shell access via sideloaded media

Often as a necessary preliminary step to further research, modification and integration of proprietary technologies, many device rooting methods (i.e ways to gain elevated access to a device) have been publicly disclosed on the internet. Commonly, devices that may not be expected to have outbound internet connectivity may provide offline firmware upgrade functionality by executing a script, or booting from some form of removable flash memory such as an microSD / SD card. Kotlyar (2017) demonstrated the ability for the inexpensive Xiaomi Dafang WiFi Camera to boot into a custom alternate `u-boot` bootloader flashed onto a microSD card. Through UART headers located on the device's circuit board, the boot environment was modified to start a shell (`/bin/sh`) instead of the original entry-point script, effectively rooting the device. Kotlyar was then able to dump the firmware, later producing a custom firmware release that did not rely on the vendor's cloud infrastructure.

Through the subversion of interrupting the default boot sequence, access to a shell allowed for the development and release of decoupled software. Whilst the exact rooting steps are unlikely to be directly transferable to other devices, the idea of obtaining elevated access via sideloading techniques is an important method to investigate.

3.7 Shell access via BGA pin shorting



For devices that do not automatically boot into removable media, methods have been discovered to force certain SoC's to enter a recovery or fallback mode. Allwinner-based SoCs implement a mode known as "FEL" that can be entered by pulling a certain pin LOW during boot[4], which allows device manufacturers to perform initial image flashing and bootloader configuration. For developers and hardware hackers, FEL mode allows users to modify the boot environment to execute a shell, allowing for further post-exploitation methods and firmware dumping / analysis.

It is noted that FEL mode can also be entered if the SoC fails to successfully launch the bootloader. Giese (2019) identified this fact and exploited the physical pin layout of the Allwinner R16 BGA package, where the data pins connecting the SoC to the (e)MMC chips (where the bootloader is stored) were on the physical perimeter of the SoC. By sliding a piece of aluminium foil (roughly 0.02mm thick) between the circuit board and the solder plane of the SoC (0.3mm), the electrically conductive aluminium foil could momentarily short the data pins long enough to cause the bootloader read operation to corrupt and fail, hence booting into

[4] Generally triggered by pulling FEL pin (`LRADC0`) LOW during boot

FEL mode and eventually gaining shell access. This method is favourable when compared to pulling the FEL pin low during boot - as access to the FEL pin would require the desoldering and removal of the SoC from a circuit board - which can be tedious and potentially destructive.

Through this hardware fault injection technique of shorting data pins during boot, Giese was able to successfully gain access to a shell on Roborock's first robot vacuum cleaner (Mi Robot Vacuum Cleaner). later Giese noted that on the circuit board of the Roborock S7 vacuum cleaner, test pad `TPA17` is connected to the ball grid location corresponding to the FEL pin - allowing FEL mode to be entered without performing a hardware fault injection.

3.8 Hardware based extraction of flash memory

In situations where no provisions exist to programmatically extract stored data from a system (i.e. shell access to perform disk imaging), hardware devices known as flash programmers can be used; where they are specially designed to read and write data on flash chips. Flash programmers however present a high cost overhead, as they are rather costly and only work with specific models and/or types of flash chips; rendering it infeasible to own a specific flash programmer for each possible variation of flash chip. Jimenez (2016) points out that a Raspberry Pi could be used as an affordable budget solution when paired with open source flash programming software like `flashrom`.

It is noted that the process of hardware flash chip dumping is not feasible in the scope of this thesis due to resource constraints of not possessing a suitable flash programmer, as well as the risk associated with hardware-based methods being possibly destructive with irreversible damage. Preliminary results (See chapter 5) have however suggested that firmware data can be obtained through programmatic methods, and as such this method will not be pursued.

3.9 Cold-boot attack to dump memory state

Regarding prior investigations of smart robot vacuum cleaners, Ullrich et al. (2019) performed a security analysis on the *Neato BotVac Connected* robot. Through the combination of a cold-boot attack - where a system is rebooted without the volatile memory (i.e. RAM) being cleared - and the booting of a custom bootloader image, the memory state of the system's prior execution was able to be dumped and analysed. This memory dump is of significant value as it would contain the binaries of loaded programs as well as their application state. The proceeding analysis revealed major vulnerabilities and concerns in the vacuum cleaner and more alarmingly, in Neato's cloud infrastructure.

Whilst logs and coredumps were encrypted when transmitted to cloud servers, the encryption keys were discovered to be hardcoded which nullified any assurances of encryption. Authentication and authorisation tokens were all encrypted with the same static RSA key - which left the cloud infrastructure vulnerable to impersonated identities and access. Supposed randomly generated keys were also discovered to be vulnerable, due to the keyspace for entropy being so short that the key was able to be bruteforced within reasonable time. Furthermore, an unauthenticated endpoint on the robot vacuum cleaner's remote port was found to be vulnerable to a buffer overflow, allowing remote code execution on the robot by anyone connected to the same wireless network.

The analysis of a system's memory state is beneficial to the security assessment of a product's firmware as static analysis techniques are unable to account for dynamic data such as response payloads from HTTP communications. It is however, unlikely that a cold-boot attack will be necessary in the scope of this thesis as preliminary results have already concluded that shell access is obtainable, hence other simpler methods could be performed to extract memory states.

Chapter 4

Thesis Plan

4.1 Statement

With the widespread availability of IoT devices, there is large interest in exploring **how manufacturers of IoT / smart home devices have addressed the increasing concerns of digital privacy and product security**. This thesis aims to explore the stance and measures that Roborock has taken regarding privacy policies and security concerns in their Roborock S6 robot vacuum cleaner.

Whilst the Roborock S6 vacuum cleaner has been assessed before (Giese 2019), there has since been significant changes and updates to the firmware and communications protocol - with security vulnerabilities even being reported recently[1]. As such, it is favourable to perform a replication study of Giese's past work, as well as to complete further in-depth security and privacy assessments of the Roborock S6.

[1] https://global.roborock.com/pages/disclosure-security-vulnerability-on-tuya-iot-cloud

4.2 Plan

4.2.1 Overview

During the course of Thesis A - initial product research was performed, and tasks were set out for Thesis B and Thesis C. The software, hardware and tooling that are required to carry out the thesis were also investigated and obtained in preparation for the remainder of the thesis, where a concurrent assessment will be performed on the Roborock S6's digital privacy and product security.

In Thesis B, a security assessment of the device will be performed, where the Roborock S6 firmware will be analysed in order to identify potential security vulnerabilities that may either render the device insecure, or raise concern to other devices connected to the same network (i.e devices on a home network). These vulnerabilities include vulnerable code as well as insecure or static secrets, credentials and configurations. A review of existing security fortifications will also be executed to audit Roborock's security posture and response to reported disclosures. Techniques and IoT forensics checklists will be followed from various sources and handbooks (Yu et al. 2020) to maximise search and analysis coverage. The product security analysis of the Roborock S6 will be the primary milestone for Thesis B, however privacy analysis will also be performed simultaneously.

In Thesis C, an investigation into the nature of networked data will be performed to qualify how important Roborock values the concerns of their customers' digital privacy. This investigation will involve the observation of the device's behaviour, where the frequency, size, content and destination of transmitted data will be analysed to gain heuristic insights of how the data might be used by Roborock. An audit of what **PII** and **UGC** is being transmitted will also be performed to address concerns of tracking and (de)anonymisation. This digital privacy assessment of the Roborock S6 will be the primary milestone for Thesis C, however research into the device's product security will also be continued.

It should be noted research into the security of the Xiaomi Home smartphone application is beyond the scope of this thesis and will only be investigated should it be required, as extensive research on the application's security has already been completed (Mehic et al. 2019; Yu et al. 2020; Giese 2019)

4.2.2 Stretch Goals

Should the major thesis milestones be accomplished ahead of schedule and sufficient time remains, an investigation into the verbosity of sensor data will be performed to explore the extent to which data can be extracted from the sensors, and of whether they are of a privacy concern. This investigation will also entail a search for other side-channel data similar to intensity values from the LIDAR sensor (Sami et al. 2020).

Alternately, the functionality of the Micro USB / ADB port located on the chassis of the Roborock S6 will be investigated. Whilst there has been discussion regarding replacing the custom Android Debugging Bridge (ADB) binary with a complete version, there is little to no information available concerning the functionality of the original binary. This investigation will likely involve a binary disassembly of the custom ADB binary in addition to an exploration of the ADB USB protocol.

4.2.3 Contingency

Contingencies have been provisioned in the event that the proposed research plans cannot be executed. Should it not be possible to gain access to the Roborock S6's firmware in any way (hence prohibiting a binary firmware analysis), focus will turn towards a black-box assessment of the vacuum cleaner's network behaviour. Network activity will be then be observed for a longer period of time over the course of Thesis B and Thesis C as to detect long-term patterns and possibly changing behaviours.

Alternatively, the network interactions of the Xiaomi Home smartphone application may be assessed, where local communications between the vacuum cleaner and the smartphone application can be observed under both scopes of privacy and security. This approach will entail the observation of network packets to look for questionable data and weak security measures.

4.3 Project Preparations

4.3.1 Acquisition of Software, Hardware and Tooling

It is estimated that a minimal amount of hardware and physical resources will be required for the completion of this project.

In observing network behaviour in a secure manner, an isolated wireless network be needed - which requires the acquisition of a wireless access point, a network switch capable of port mirroring and an active internet connection. To analyse network packets, the free Wireshark network protocol analyser will be used in conjunction with a Man-In-The-Middle proxy like Burp Suite or mitmproxy. All of the required software and hardware to perform networking monitoring of the Roborock S6 has already been obtained, with preliminary results recorded in Chapter 5.

To perform a security analysis of the robot vacuum cleaner's firmware, physical entry into the vacuum cleaner must first be performed in order to gain access to the UART pins on the device's circuit board, from which further post-exploitation and analysis can then occur. In order to interface with the UART pins, spare jumper wires and a soldering iron will be required as the UART pins are exposed as test pads rather than pin headers. Additionally a UART interface (i.e. USB to TTL UART) is also required, however all tools have already been obtained. Whilst hardware such as flash chip programmers and microsoldering stations may be beneficial to creating an image of the Roborock S6's firmware, preliminary testing had successfully allowed for a root shell to be obtained as later discussed in Chapter 5. As consequence, these additional devices will not be required. To perform firmware and binary analysis, tools like Binary Ninja, binwalk and bindiff have been obtained.

4.3.2 Upskilling and Learning

Prior to the security analysis of binary images found in the Roborock S6 firmware, sound understanding of the ARM processor instruction set (ISA) must be obtained in order to understand the disassembled instructions. In-depth understanding of ARM-specific and embedded systems-specific protection mechanisms such as processor modes, protection rings, OPTEE and RPMB may be required to be learned as subtle vulnerabilities are discovered.

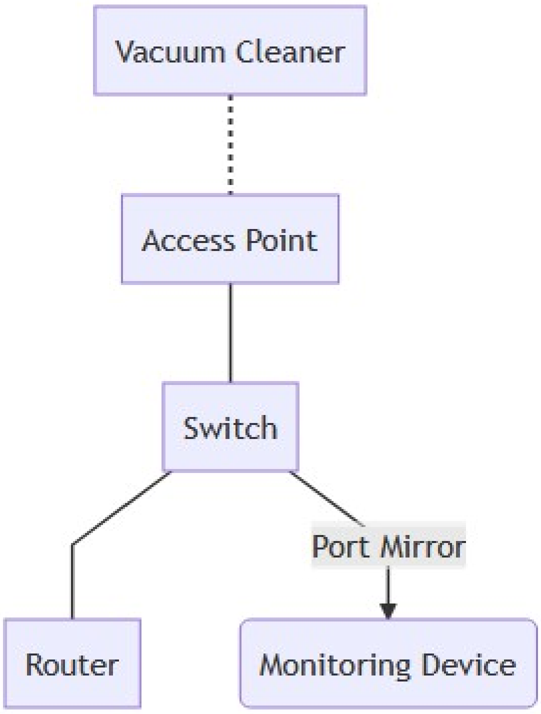
It will also be beneficial to be familiarised with both the common and uncommon paths that should be looked at in a Linux system forensics analysis, as relevant to the research performed by Yu et al. (2020).

Chapter 5

Preliminary Results

5.1 Privacy Assessment

5.1.1 Network Setup



An isolated wireless network was set up to securely observe the wireless network activity of the vacuum cleaner.

A DHCP server was set up to serve address leases on the `10.10.10.0/24` network, where the IPv4 address of the vacuum cleaner is dynamically issued to simulate a general home network. A TP-LINK TL-SG105E switch was configured to port mirror any data sent and received through the wireless access point to the Roborock S6.

An Android smartphone with the Xiaomi Cloud application was also connected to the isolated network so that intra-device connectivity could be observed. This phone was configured to relay all of its connections through a MITM SSL proxy, so that HTTPS payloads could be decoded.

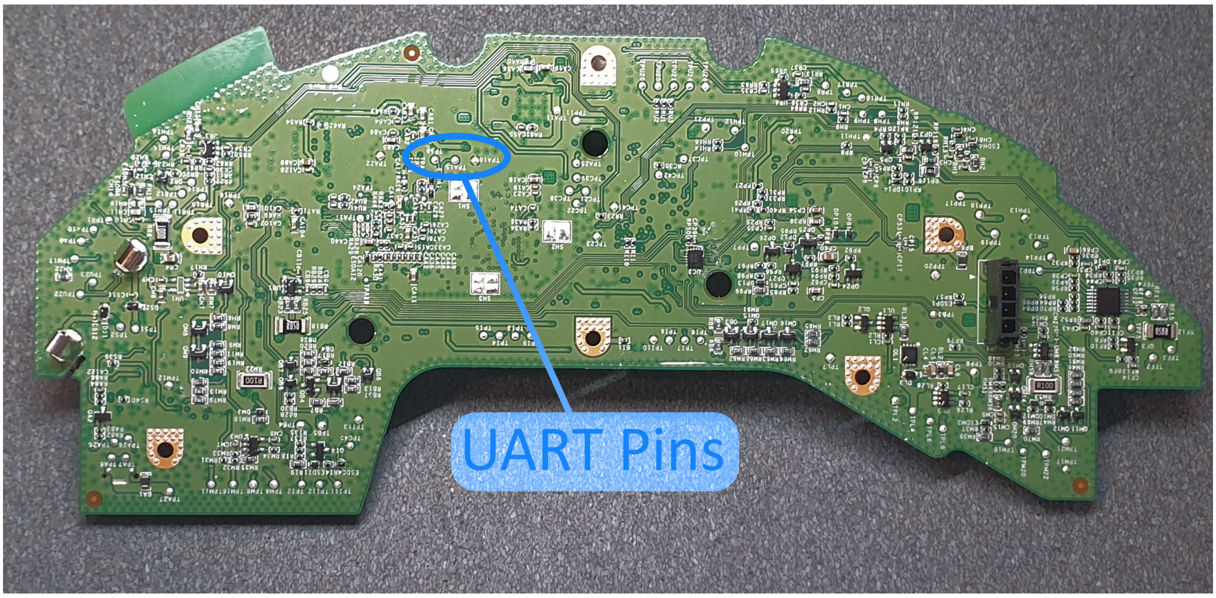
5.1.2 Network Activity

Initial network observations indicated a high volume of network activity between the Xiaomi Cloud application to their IoT servers.
It was also observed that the Roborock S6 periodically replied to an incoming request roughly every 2 seconds - this is presumably either a keep-alive / command and control mechanism. Detailed payload analysis has not yet been attempted.

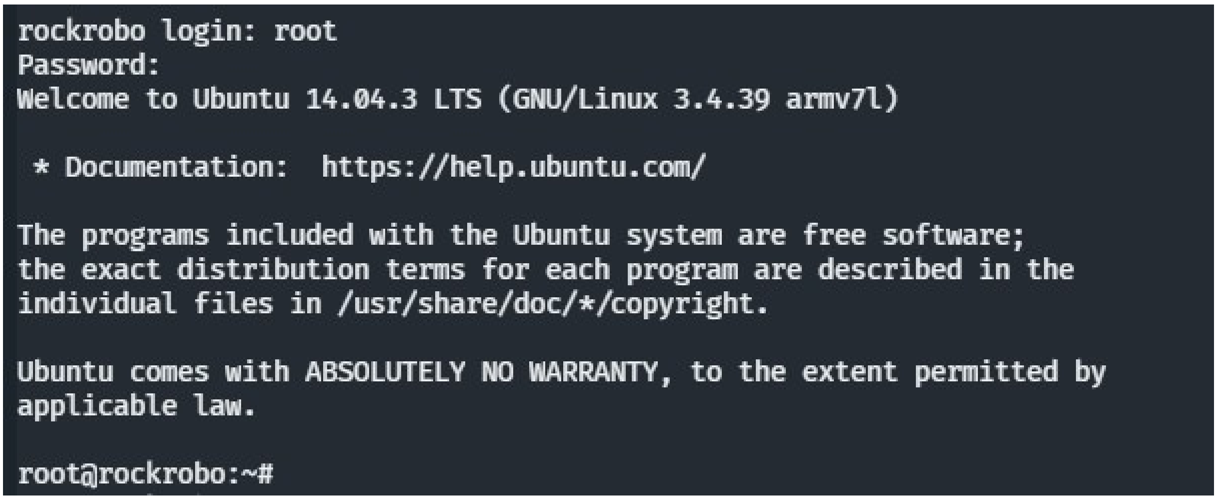
It is worthwhile to note that the network was initially set up incorrectly when packet logging took place. As a result of port mirroring being set up on the router instead of on the network switch, only inbound and outbound WAN packets were being forwarded. This misconfiguration had prevented the logging of LAN packets transmitted between the Xioami Cloud smartphone application and the Roborock S6 whilst remote control operation was in place. The correct networking monitoring environment will be provisioned for future network activity monitoring.

5.2 Security Assessment

Prior to starting the firmware security analysis, the firmware image first had to be be obtained. However as there are no public images of the firmware available online, physical disassembly of the vacuum cleaner was required in order expose the UART pins, as outlined by Giese (2019).



Once the circuit board was removed from the vacuum cleaner chassis, UART pins could be attached to test pads `TPA15`, `TPA16` and `TPA18`. When the vacuum cleaner was powered on and a serial connection was established a baud rate of `115200`, the bootloader and system serial interfaces were able to be interacted with.



```
rockrobo login: root
Password:
Welcome to Ubuntu 14.04.3 LTS (GNU/Linux 3.4.39 armv7l)

 * Documentation:  https://help.ubuntu.com/

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

root@rockrobo:~#
```

Through research into the bootloader source code[1], it was revealed that the injection of the `s` character during boot would trigger the bootloader to enter a shell. Which granted access to read and decrypt the root password that was encrypted in a file called `vinda`. No attempts to further to explore the file system nor to create an image of the firmware has yet been made.

[1] https://github.com/allwinner-zh/bootloader/blob/master/u-boot-2011.09/board/sunxi/board_common.c#L843-L847

5.3 Plans for Future Research

Preliminary security assessment results were satisfactory, as obtaining root access was a critical step prerequisite to analysing the Roborock S6's firmware without expensive equipment like a flash chip programmer - which would have its own inherent risks at outlined in Chapter 3. Moving forwards, the next steps in carrying out the product security assessment will involve the dumping of the flash chip content into an image that can externally analysed offline. It will also be worthwhile to also perform live system forensics, as dynamic behaviours will be hard to detect during a static analysis.

Whilst not the immediate focus for Thesis B, a reconfiguration of the network monitoring setup is required to properly begin the privacy assessment of the product. Additionally, network activity during initial device pairing as well as general network activity will require to be recaptured to observe the LAN packets that were previously omitted during network capture.

References

Abrams, L., 2021. 533 million facebook users' phone numbers leaked on hacker forum. Available at: https://www.bleepingcomputer.com/news/security/533-million-facebook-users-phone-numbers-leaked-on-hacker-forum/.

Brown, R., 2015. Smart homes can pay off when it's time to sell. Available at: https://www.cnet.com/home/smart-home/what-happens-when-you-sell-your-smart-house/.

Costin, A. et al., 2014. A large-scale analysis of the security of embedded firmwares. In *23rd USENIX security symposium (USENIX security 14)*. San Diego, CA: USENIX Association, pp. 95–110. Available at: https://www.usenix.org/conference/usenixsecurity14/technical-sessions/presentation/costin.

Giese, D., 2019. Security analysis of the xiaomi IoT ecosystem. Available at: https://dontvacuum.me/thesis/Security_Analysis_of_the_Xiaomi_IoT_Ecosystem.pdf.

Giese, D., 2021. Smart home security & privacy.

Jimenez, J.C., 2016. Practical reverse engineering - dumping the flash. Available at: https://jcjc-dev.com/2016/06/08/reversing-huawei-4-dumping-flash/.

Jmaxxz, 2016. Backdooring the frontdoor.

Jones, R., 2017. Roomba's next big step is selling maps of your home to the highest bidder. Available at: https://gizmodo.com/roombas-next-big-step-is-selling-maps-of-your-home-to-t-1797187829.

Kotlyar, E., 2017. Xiaomi DaFang hacks. Available at: https://github.com/EliasKotlyar/Xiaomi-Dafang-Hacks.

Mehic, M., Selimovic, N. & Komosny, D., 2019. About the connectivity of xiaomi internet-of-things smart home devices. In *2019 XXVII international conference on information, communication and automation technologies (ICAT)*. pp. 1–6.

Research & Markets, 2021. Insights on the smart homes global market to 2026 - featuring ABB, acuity brands and emerson electric among others. Available at: https://www.prnewswire.com/news-releases/insights-on-the-smart-homes-global-market-to-2026---featuring-abb-acuity-brands-and-emerson-electric-among-others-301425 html.

Sami, S. et al., 2020. LidarPhone: Acoustic eavesdropping using a lidar sensor: Poster abstract. In *Proceedings of the 18th conference on embedded networked sensor systems*. SenSys '20. New York, NY, USA: Association for Computing Machinery, pp. 701–702. Available at: https://doi.org/10.1145/3384419.3430430.

Ullrich, F. et al., 2019. Vacuums in the cloud: Analyzing security in a hardened IoT ecosystem. In *13th USENIX workshop on offensive technologies (WOOT 19)*. Santa Clara, CA: USENIX Association. Available at: https://www.usenix.org/conference/woot19/presentation/ullrich.

VTRUST, M.S. -, 2018. Smart home - smart hack - wie der weg ins digitale zuhause zum spaziergang wird. Available at: https://media.ccc.de/v/35c3-9723-smart_home_-_smart_hack.

Wei, T. et al., 2015. Acoustic eavesdropping through wireless vibrometry. In *Proceedings of the 21st annual international conference on mobile computing and networking*. MobiCom '15. New York, NY, USA: Association for Computing Machinery, pp. 130–141. Available at: https://doi.org/10.1145/2789168.2790119.

Yu, M. et al., 2020. A survey of security vulnerability analysis, discovery, detection, and mitigation on IoT devices. *Future Internet*, 12(2). Available at: https://www.mdpi.com/1999-5903/12/2/27.