Evaluating IoT Device Security

Fraser Dumayne

40205071

Submitted in partial fulfilment of
the requirements of Edinburgh Napier University
for the Degree of

BEng (Hons) Cyber Security & Forensics

School of Computing

April 2019

Authorship Declaration

I, Fraser Dumayne, confirm that this dissertation and the work presented in it are my own achievement.

Where I have consulted the published work of others this is always clearly attributed;

Where I have quoted from the work of others the source is always given. With the exception of such quotations this dissertation is entirely my own work;

I have acknowledged all main sources of help;

If my research follows on from previous work or is part of a larger collaborative research project I have made clear exactly what was done by others and what I have contributed myself;

I have read and understand the penalties associated with Academic Misconduct.

I also confirm that I have obtained **informed consent** from all people I have involved in the work in this dissertation following the School's ethical guidelines

Signed:

Date:

Matriculation no:

General Data Protection Regulation Declaration

Under the General Data Protection Regulation (GDPR) (EU) 2016/679, the University cannot disclose your grade to an unauthorised person. However, other students benefit from studying dissertations that have their grades attached.

Please sign your name below *one* of the options below to state your preference.

~~The University may make this dissertation, with indicative grade, available to others.~~

The University may make this dissertation available to others, but the grade may not be disclosed.

~~The University may not make this dissertation available to others.~~

Abstract

The Internet of Things is a global network of internet connected devices which can perform intelligent sensing and processing. These devices are used in many areas by both general consumers and organizations to carry out various tasks in an efficient manner.

As the world becomes more reliant on technology, it becomes clear that security is playing an ever more important role in protecting devices. Personal data is frequently stored, processed, and transmitted across networks and it is crucial for this to be secured. Vulnerabilities which can be exploited by attackers to gain access to this data are often ignored by manufacturers in an attempt to create cheap and accessible devices.

The purpose of this paper is to provide the reader with a clear overview of the Internet of Things as it is today and what the future is like for IoT devices with the present lack of security standards in many devices. Additionally, this paper will show how attacks can be performed against real world IoT devices utilising various protocols for data transmission. These attacks will prove how simply vulnerabilities can be exploited and what this could mean for both an attacker and a victim.

Contents

List of Tables

List of Figures

Acknowledgements

I would like to thank my supervisor, Liam Bell for his continued support and advice which helped me attain my goals for the project as well as keep my sights aligned through our various meetings.

Finally, I would like to thank Edinburgh Napier University for the support I received through the various facilities and knowledge gained from modules which assisted me in taking on this project.

1. Introduction

1.1 Background

The Internet of Things is an increasingly significant concept in the computing field which defines a world in which everything is connected via the internet. As the number of IoT devices available increases every day, it becomes ever more important to ensure that these devices are secured.

1.2 Aims

The aim of this project is to provide readers with a greater understanding of the Internet of Things from both a technical level and a more general level. This will involve discussions on the topic itself as well as further review of security issues that exist within many devices. Additionally, some analysis and testing will be performed on easily purchasable IoT devices which use a variety of different protocols for communication. These tests should expose vulnerabilities within the devices and allow for an evaluation to be made on the security measures used by the IoT devices. The report will discuss possible solutions to these vulnerabilities and discuss what the consequences of such a lack of security may mean for attackers and victims.

1.3 Objectives

The objectives of this project revolve around providing readers with a greater understanding of the security weaknesses in IoT devices and why a low standard for security may have a hugely negative impact on consumers. The following objectives were created to attain the aims of the project:

1. Create a comprehensive literature review which clearly discusses the concept of the Internet of Things and provides a background to common security issues.
2. Discuss in detail two common IoT protocols so that penetration tests can be performed with greater accuracy.
3. Perform analysis on Bluetooth and Wi-Fi devices by exploiting vulnerabilities using software and hardware.
4. Evaluate results obtained and discuss the impact of these vulnerabilities as well as the simplicity of recreating the attacks.

5. Based on the results obtained during the implementation, provide potential countermeasures to mitigate attacks.

The main objective of this project is to evaluate the current communication protocols used in the Internet of Things and try to gain an understanding of what vulnerabilities exist within these protocols by attempting to perform penetration tests on various IoT devices.

1.4 Scope

The project gives an expansive review on the Internet of Things and the security issues which make it such a volatile subject. The topics covered are limited in detail so as to cover a wide range of topics in an easily understandable fashion. Additionally, two largely used protocols in the Internet of Things – Bluetooth and Wi-Fi – will be covered in greater detail through literature review and by performing a detailed analysis on the IoT devices which use the protocols.

1.5 Limitations

Various limitations exist which may negatively affect aspects of the project such as the final aims. The implementation will require use of a variety of software tools built into Kali Linux which have not been used by the author previously, which may mean that some penetration tests will be less effective. Additionally, the project is limited in time scale which means that a detailed knowledge of the protocols discussed and the Internet of Things in general may be more difficult to acquire.

1.6 Structure

The paper is divided into 6 chapters:

Chapter 1 introduces the concept of the project and an overview of what is expected in terms of results and discussion.

Chapter 2 presents an overview of Internet of Things, including the main usage areas and future possibilities. It also provides a more detailed evaluation of the technologies used and how devices communicate. Additionally, this chapter gives a detailed look at what the most common vulnerabilities in IoT devices are and an overview of the types of current threats. Finally, a detailed review of Wi-Fi and Bluetooth is given to provide background for the practical elements of the project.

Chapter 3 provides a clear outline of how the project will be performed and what the expectations are from the device analysis. This chapter includes an overview of how the lab was set up including both the software and hardware tools required to perform the tasks.

Chapter 4 describes the methods of analysis including how attacks were performed and what data was extracted as a result.

Chapter 5 explains what results were obtained from the tests and gives a clear discussion on how these may affect various types of users. This chapter will also bring into discussion how the vulnerabilities found may be solved by the manufacturer.

Chapter 6 gives a final discussion on what was concluded from the tests and provides a critical analysis of the aims and objectives originally stated at the beginning of the project.

2. Literature Review

2.1 Introduction

This chapter presents a literature review of three subjects which are all connected by the concept of IoT. The first component gives a brief discussion on Internet of Things and its current/future uses, followed by a more in-depth description of the technology and its inner workings. The second part gives a clear summary of common vulnerabilities which exist in IoT devices and the consequences of such attacks by referring to recent incidents involving exploited IoT devices. The final component will discuss the protocols used by IoT devices to transmit data and how they operate.

2.2 The Internet of Things

The Internet of Things is a concept which involves connecting everyday objects (also known as 'things') which are otherwise 'unintelligent' and allowing them to become smart by installing them with devices such as hardware chips, sensors, and software so they can gather data on the world around them. There are thousands of different IoT devices in existence including personal health trackers, intelligent thermostats, and voice controlled personal assistants. These devices tend to be so well integrated into the environment around us that they often go unnoticed. The most important aspect of IoT devices is their ability to connect to the internet as it allows for them share data with each other and even be remotely controlled by their owner.

2.2.1 Usage Areas

Applications for IoT devices can range from home appliances such as smart fridges which allow the owner to remotely monitor groceries using internal sensors and cameras, to wearables which can track various aspects of your physical health and report those statistics to a mobile application. IoT devices are not just aimed at consumers however, some significant growth usage of this technology is redefining how these areas like the agriculture, transport, education, and health sectors operate.

2.2.1.1 Smart Homes

One of the leading IoT consumer markets is the intelligent housing industry which enhances the interactivity and control owners have of their houses. There are many devices for smart homes on the market today such as smart boilers, locks, lights, doorbells, fridges, thermostats, and others as shown in Figure 1. These devices offer the ability for greater control and a better user experience than non-smart versions due to their ability to connect to the internet and allow the owners to communicate with them over mobile or web interfaces without being in the vicinity (Ricquebourg et al., 2006).

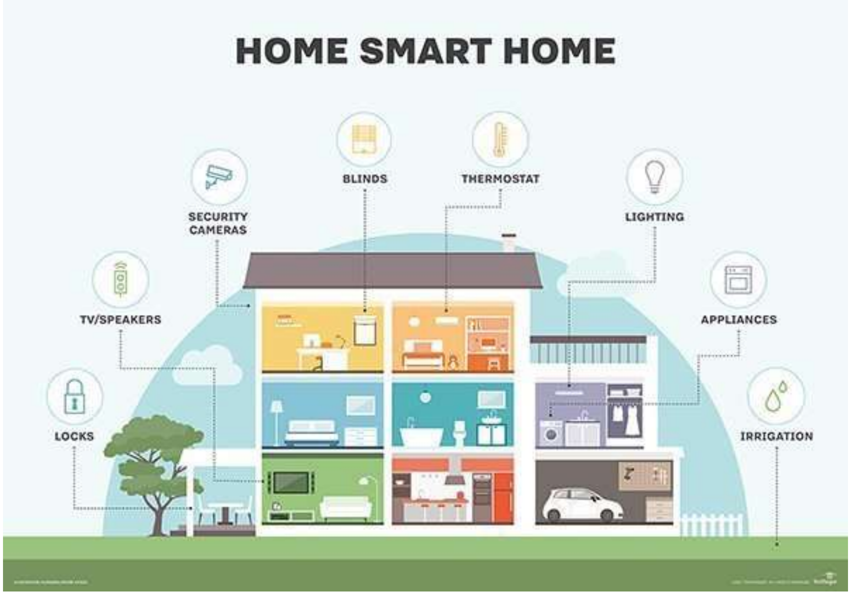


Figure 1 - Smart home example (Napoli, 2018)

2.2.1.2 Smart Factories

In the manufacturing world, factories are evolving to become more and more automated as technology advances. With the help of IoT, performance of equipment can be analysed to ensure they are not being overused, and employee health can be monitored, and efficiency can be improved. One example of IoT in this environment is the use of Wi-Fi capable sensors which allow for products to be monitored in real-time using a performance management system (Castaldi, 2016).

2.2.1.3 Smart Hospitals

Another area of growth within IoT is in hospitals. These facilities tend to require a lot of data to be processed on many different patients which can often be overwhelming when monitored solely by humans. By using IoT devices, the process of collecting patient data can be automated and improve efficiency within the hospital as well as reduce stress on staff. In this smart hospital scenario, internet connected wearables could be worn by patients to read their vitals and report any changes to staff members, resulting in faster response times (Bohlin, Sehlstedt, Kharbanda, & Treutiger, 2016).

2.2.1.4 Smart Cities

On a larger scale, the Internet of Things could potentially result in smart cities which are composed of many interconnected, intelligent devices which could potentially make living in cities much easier for many people. These devices could include intelligent traffic lights which can make decisions on their light phases based on information gathered on traffic from their cameras ultimately resulting in reduced congestion (Marshall, 2018), or even smart bins which can sense when they are becoming full meaning they can be emptied on time by sending notifications to the collectors.

2.2.2 Future of IoT

The proliferation of both consumer and industrial IoT devices is only going to continue as we move towards a more automated society with smart cars, smart homes, and smart cities. In 2018, the number of active IoT devices surpassed 7 billion and is still advancing towards an expected 21.5 billion active devices in 2025 as shown in Figure 2. This increase in devices will result in many more damaging cyber-attacks due to the number of vulnerable devices.

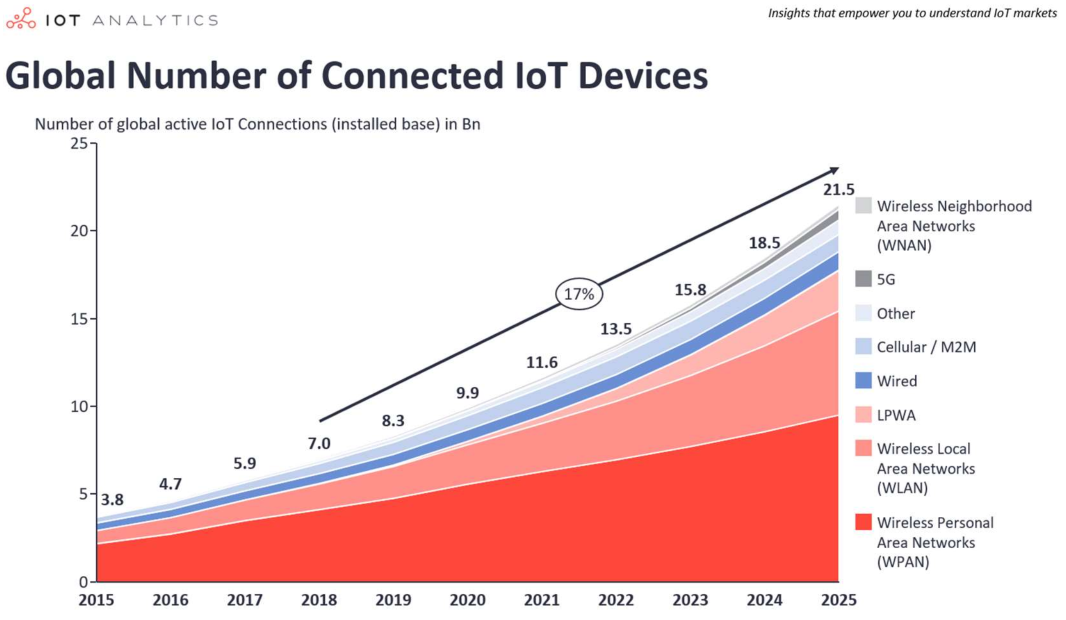


Figure 2 - Graph of the number of IoT devices expected by 2025 (Leuth, 2018)

As IoT devices become more common and technology advances, it is likely that they will become cheaper to purchase and to operate over time, as well as become more powerful than they are now; sensors will be able to become more sensitive, devices will become more intelligent and integrate more complex artificial intelligence for processing large volumes of data quickly.

Currently, consumer devices take up roughly 63% of Internet of Things usage (Meulen, 2017), with industrial usage still rising. As IoT becomes more viable, industrial usage should increase significantly with devices becoming a part of factories and office spaces to increase efficiency.

2.2.3 Challenges in IoT

2.2.3.1 Scalability & Flexibility

The Internet of Things is constantly expanding as more things become connected using sensors, and other hardware. One crucial challenge of IoT is the need for it to be scalable so that both new and old devices can continue communicating and sharing data (Gupta, Christie, & Manjula, 2017). Scalability should cover new devices which are attempting to integrate with current devices so that older devices can remain useful. In a smart home network, it would be important that additional smart devices would seamlessly integrate into the current network by being capable of communicating with other devices on the network.

2.2.3.2 Latency

For most internet connected devices, a low latency is critical to operations. As significant volumes of data are handled and transmitted by these devices, slowdowns can occur especially when many of these devices may operate on the same network such as in a smart home. Due to hardware limitations on some IoT devices, fast response times can be difficult to attain but still must be prioritised. For a reduction in latency, an increase in power is required, which can be difficult for some devices to achieve, especially battery operated IoT devices which must be as power efficient as possible (IEEE, n.d.).

2.2.3.3 Data Management

One functionality of many IoT devices is the storage of data which is captured from sensors to make further decisions and predictions. In some cases, devices will communicate data at certain intervals, meaning that sensed data must be stored somewhere until it is transmitted to a central server. This could potentially result in storage issues when large volumes of data are being stored on the device itself (Vongsingthong & Smanchat, 2015). Data collection offers great benefits to the end users and to the overall quality of the Internet of Things as it can be analysed rather than simply locally storing the data and not using it for anything more. Processing and analysing this data can be a great task for IoT devices especially due to the low processing power that many devices have. Data management solutions are therefore required to ensure that the volume of data gathered can successfully be operated on and stored on a low power device (Abu-Elkheir, Hayajneh, & Ali, 2013).

2.2.3.4 Interoperability

Different IoT devices use different technologies in terms of protocols, hardware, and software to successfully carry out their tasks and communicate. This is generally due to manufacturers independently creating their own solutions for their devices. For devices to communicate effectively in an IoT network it is crucial that they can use standardised models for increased interconnectivity (Uviase & Kotonya, 2018). There are two main business models for IoT deployment; the most common is vertical which involves all of the components – such as devices, gateways, and services – being managed by one overarching controller. Alternatively, a horizontal solution may be implemented which allows various different providers of devices and services to operate in unison by utilising a common framework between the devices (Quinnell, 2013).

2.2.3.5 Resource Constraints

Many sacrifices must be made during the development of IoT devices to allow them to be both affordable and accessible. In some cases, devices utilise batteries so that they can remain portable but results in a lack of useful functionality to retain a long battery life. This mobility may also lead to a reduction in processing power, storage, and networking capabilities (Sehgal, Perelman, Küryla, & Schönwälder, 2012).

2.2.4 IoT Communication

IoT devices utilise a variety of protocols which allow them to communicate over different software and hardware. These protocols can be viewed in the form of the OSI model which shows how data is handled through different layers of the stack, each with its own functionality, as shown in Figure 3.

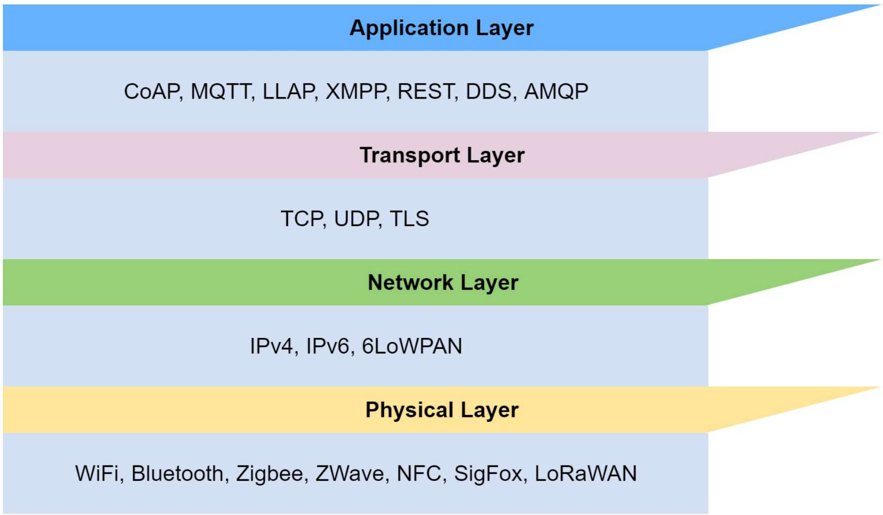

Application Layer
CoAP, MQTT, LLAP, XMPP, REST, DDS, AMQP
Transport Layer
TCP, UDP, TLS
Network Layer
IPv4, IPv6, 6LoWPAN
Physical Layer
WiFi, Bluetooth, Zigbee, ZWave, NFC, SigFox, LoRaWAN

Figure 3 - IoT protocols in OSI stack

The top layer known as the Application Layer acts as an interface between the client device and the application. One example of a protocol used in this layer is CoAP (Constrained Application Protocol) which offers superior functionality on networks lacking in resources (also known as Low Power and Lossy Networks). It operates as HTTP for constrained, low power devices, with low bandwidth and low overhead. This is extremely useful for many IoT devices which operate on batteries or are small and do not have the hardware to maintain proper HTTP connections (Bilal, Rehman, & Ali, 2018). Other examples of such protocols are shown below in Table 1.

Protocol	Description	Environments
CoAP	Designed for constrained devices by using a lightweight design with UDP for low overhead.	Networks with less resources.
MQTT	A lightweight and reliable protocol for use in constrained machine-to-machine environments.	Unreliable networks.
DDS	Used to enable high-end applications such as power grids to operate efficiently and reliably using a publish-subscribe model.	High performance networks.
AMQP	This is a standard for transmitting messages between different applications on an enterprise scale.	Networks requiring safe transport of messages.

Table 1 - Application layer protocols comparison

The transport layer provides a way for devices to communicate using protocols such as the Transmission Control Protocol (TCP) which allows for reliable connections between a client and the server, or the User Datagram Protocol (UDP) for connectionless and lightweight operations. Additionally, IoT networks are prone to attacks on security and therefore require some protection for data being transmitted between end devices and applications. Missing encryption can be a significant privacy issue, especially in easily accessible wireless networks as data being processed is open to an attacker analysing network packets from the victim (Pal, n.d.). The transport layer can provide security features by utilizing protocols such as Datagram Transport Layer Security (DTLS) so that data confidentiality can be sustained.

The Network Layer is responsible for sending data in the form of packets from a client to its destination. One of the most common technologies in this layer is IPv4 which operates by giving each host a unique 32-bit address. Given the small address space and the significant increase in internet connected devices, a new standard had to be created. IPv6 is an updated version of IPv4 which offers improvements such as 128-bit addresses so that more devices can be identified, as well as improvements to security (Sun, 2016). Another protocol used in this layer is IPv6 Low Power Wireless Personal Area Network (6LoWPAN) which is designed for devices with resource constraints which operate in Wireless Sensor Networks. Additional features include header compression so that less overhead is required for processing compared to a normal IPv6 packet.

The Physical Layer gives devices a channel to communicate either between an access point or between other client devices. Many IoT devices make use of wireless protocols such as Wi-Fi, Bluetooth, ZigBee, and NFC to operate with extra mobility. In addition, these protocols offer simpler connection methods due to the lack of physical connection needed (LaBrie, 2017). One example of a protocol used in this layer is Zigbee which is designed for low power wireless operations in personal area networks. This protocol is ideal for smart home automation due to its ability to create a mesh network in which data is transferred between different devices on the network. This is beneficial to the home owner as devices with different purposes can be controlled from a single application (Ludlow, 2018).

2.3 Security in the Internet of Things

2.3.1 Security Challenges in IoT

As so many devices are poorly configured, not up to date, and ultimately vulnerable (Bell, 2018), it becomes very simple for an attacker to infect and exploit IoT devices and make use of them for their own benefit and even steal sensitive data. These devices can easily be configured and updated to prevent further attacks. According to a report by Zingbox, poor user practices including downloading unknown applications and visiting dangerous websites resulted in 41% of IoT issues (Zingbox, 2018). Often times security features are reduced to ensure that devices are simpler and more convenient to use in the real world, or just due to developer laziness as they attempt to rush out hardware whilst IoT remains a popular topic but still relatively unclear to the general public (Nichols, 2015).

2.3.1.1 Outdated Software

One security issue which is prevalent across not only IoT devices but also devices such as desktop computers, networking equipment, and games consoles, is the updating of software (Donnelly, 2017). Patches for devices are generally put out frequently by manufacturers to fix bugs, improve efficiency, and solve any vulnerabilities which could lead to an exploit resulting in devices being susceptible to attacks. Quite often these updates require some time to install or can simply cause devices to stop working optimally. According to a survey commissioned by Google in 2015, only 64% of security experts upgrade software when prompted to, suggesting that those users were still susceptible to any vulnerabilities being exploited, and that normal users without the same expertise are even less likely to update their software (Ion, Reeder, & Consolvo, 2015). One recent example of outdated software being exploited was in the ‘WannaCry’ ransomware attack which affected hundreds of thousands of computers around the globe in May 2017, but most significantly the NHS (Jalil & Lumpur, 2018). The attack used a previously known exploit for Windows XP devices which was not patched in time by the victims, enabling the ransomware to spread. This issue is not always necessarily a user problem and can also be the result of developers dropping support of devices or failing to supply the correct updates in time.

2.3.1.2 Default Configurations

All devices come with some sort of default configuration whether it be something as simple as a date and time zone or a basic user profile. Since IoT has become so relevant, many companies have been pushing out devices to market with very little security testing resulting in weak software configurations. One common attack which is carried out on not just IoT devices, but also general networking equipment, is a default password attack. This attack makes use of the password and username combinations which comes with devices as they are shipped before they are supposed to be altered by end users, by guessing the kind of combinations which may occur based on knowledge of the manufacturers. More recently, the Satori botnet utilised this weakness in manufacturer and end user practices by brute-forcing these combinations on vulnerable IoT devices to create a massive network of devices to operate under the control of an attacker (Zurkus, 2018). Since these devices had no form of authentication to detect who was accessing them other than a password, it was simple for the attacker to gain access and abuse the vulnerability. On top of these issues, many devices come with insufficient security configurations even if the user did decide to alter device settings.

2.3.1.3 Hardware Security

Although potentially not as frequent or as easy to perform as virtual attacks, physical attacks can still occur and be just as dangerous. These attacks involve the attacker gaining physical access to the components inside the vulnerable device such as the processor and storage elements. Since IoT devices often blend into the environment around us, there is not a whole lot of security measures in place to stop an attacker from simply going up to the device and abusing its vulnerabilities. In comparison, hardware such as servers are often stored centrally in locations with a wide variety of security measures in place such as locked server racks, doors locked with physical authentication methods such as fingerprint scanners, and surveillance cameras, all of which are external to the physical devices offering much more reliable possibilities for defending against attackers. Despite the vulnerable state of a lot of IoT devices, many hardware security measures are possible such as Trusted Platform Modules (TPM) which is a dedicated chip stored on end devices which securely stores encryption keys for the host (Kim, 2005), or Trusted Execution Environments (TEE) which is a secure area within the device which protects data by utilizing a minimal OS specializing in security (Borza, 2016).

2.3.1.4 Insecure User Interfaces

For users to interact with their IoT device, they must access some form of interface which could be on a website, a cloud service, or even in the form of an app on a mobile device. Web applications are constantly open to attacks as long as they are available to users as they utilise many technologies to provide a range of services (Whitelegg, 2017). One of the most common web application attacks utilises injection flaws which allows for an attacker to input commands into user input forms, for example an attacker might type in an SQL command into an input box which sends that to an SQL server and bypasses any security systems in place to retrieve sensitive information (Reetz, 2017). Other examples include public facing sensitive data, lack of user authentication, and cross-site scripting which allows attackers to implement scripts onto the web page.

Mobile applications are an extremely useful way of accessing IoT devices due to the portable nature of mobile phones making it easier for users to control their devices. Similar to web applications, mobile apps tend to suffer from a wide range of vulnerabilities which can be leveraged by attackers to cause damage or extract information. Insecure programming is one potential weakness in a mobile application and could mean that an attacker may be able to exploit bugs such as memory leaks, or buffer overflows which may allow an attacker to execute specific actions (Maroš, et al. 2014). In an IoT environment this may allow an attacker to manipulate a device by exploiting weaknesses such as those previously described to extract sensitive data.

2.3.1.5 Insufficient Authentication

Authentication is important to any security solution and is especially required in IoT due to the number of devices and how they are used. For example, in a smart home setting we would want to ensure that the owner is the one changing the temperature of their thermostat and not just some outsider trying to cause damage or frustration. On the other hand, it could be possible that the owner would want more than one user on the smart home application for other family members to access the devices or for suppliers to push software updates (Rossi, 2016). If the owner of an IoT device used weak login details for their smart home mobile interface without two-factor authentication for example, an attacker may simply just login to their account by simply guessing their password. In this attack, there is no way for the device to confirm whether the original owner is logging in or not as there is no form of authentication to prove their identity.

2.3.1.6 Insecure Communication

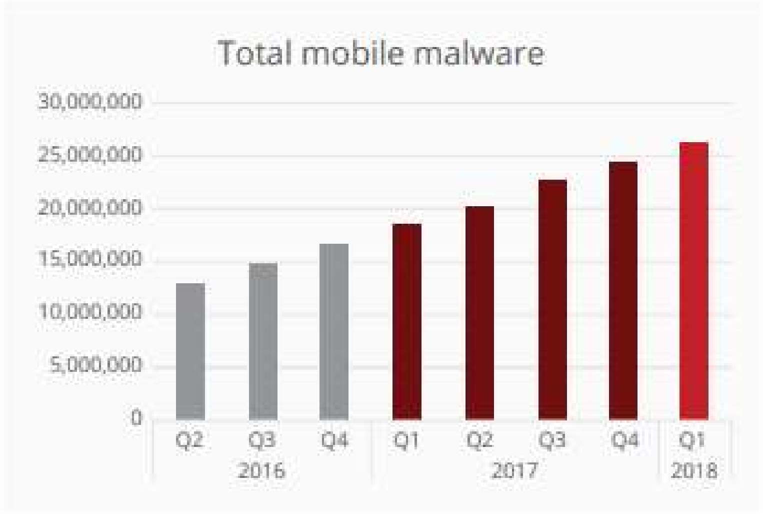
Encryption standards are critical not only in IoT but in general computing applications as they are required to maintain confidentiality during transport of important data. Applications sometimes fail to include encryption meaning that sensitive messages or back end communications containing session tokens can be read if an attacker is observing the traffic. One recent attack on the Z-Wave communication protocol used by many IoT devices meant that two Z-Wave devices being paired together could be fooled into using an out of date standard which used a default encryption key of ‘0000000000000000’, meaning that all traffic could be read by decrypting it with the default key (Cimpanu, 2018). Since IoT devices often collect sensitive data and transmit it to cloud servers or other devices, it is crucial that manufacturers implement encryption to avoid attackers being able to intercept data and use it against the victim.

2.3.1.7 User Data Leakage

Since IoT devices are constantly collecting and transmitting data, large volumes of potentially sensitive information such as GPS locations, health statistics, and personal data which can be used to uniquely identify users has to be stored in the cloud and can sometimes be incorrectly configured meaning that it can be viewed and manipulated by attackers. Recently, an attack on Google’s Home and Chromecast devices showed that GPS locations could be extracted due to the data being transmitted over insecure HTTP connections and accessed by attackers without authentication (Young, 2018). User privacy is one of the most important aspects when dealing with data collection and it is crucial for manufacturers to ensure that proper security configurations are in place so that users cannot be made vulnerable through attacks. Another example of this happening was with an internet-connected toy bear which allowed for parents to send messages to their children. The user credentials themselves were stored on a database that was completely unprotected from attackers allowing them full access to 800,000 users details, as well as messages which were stored in an AWS bucket with no authentication measures (Franceschi-Bicchierai, 2017). In mobile device applications, data can often be stored on the device itself in the main storage or in the cache, meaning that if the owner does not lock their device then their personal data could be accessed, so it is crucial for developers to ensure that sensitive data is encrypted if it is stored on mobile devices.

2.3.2 IoT Threats

As well as potentially improve the lives of many, this increase in devices will have serious consequences for cyber security as we become more reliant on the IoT. The most significant threat to IoT devices is the increasing spread of malware which can abuse known vulnerabilities in unpatched devices. Over the years, malware on mobile devices has drastically increased, as shown in Figure 4. As previously stated, users often fail to update their device with the latest security patch, resulting in a weak device that is susceptible to attack. Once malware infects a device, it becomes simple for it to spread to multiple devices as vulnerable devices connect with other vulnerable devices and gives attackers a variety of further attacks to carry out.



Source: McAfee Labs, 2018.

Figure 4 - Graph showing increase in mobile malware (McAfee, 2018)

2.3.2.1 Distributed Denial of Service

Denial of service attacks are already a frequent occurrence in the present world and are only becoming more devastating and difficult to defend against. According to an article published by TechRepublic, the number of DDoS attacks increased by 91% from Q1 to Q3 of 2017 (Rayome, 2017). Some of the most devastating attacks being carried out today are Distributed Denial of Service (DDoS) attacks which utilise large botnets making them very difficult to stop. Botnets are made up of a main ‘command and control’ server and many of zombie devices which have been infected with malware, making them easy to be remotely controlled by the attacker. With the proliferation of IoT devices, botnets have become more of a danger as many vulnerable devices are exploited – often without the owner’s knowledge – to form part of the botnet. One real world example of this happening is the ‘Mirai Botnet’ which “left much of the internet inaccessible on the U.S. east coast.” (Fruhlinger, 2018). This attack abused weak security practices across a variety of IoT devices such as weak default username and password combinations to create an entire network of infected vulnerable devices used to bring down servers across the world.

2.3.2.2 Data Extraction

One of the main functions of IoT devices is data collection from sensors and the storage of it. This collection of data is happening constantly as the device is active and as previously suggested is being stored insecurely. Ultimately, an attack on data like this may lead to identity theft through combinations of fitness tracking statistics, social media information, and other IoT device data. Data like this can be a stepping stone for an attacker to carry out even further attacks. Recently, a fitness tracking app named ‘Strava’ was shown to give away private military bases as soldiers used fitness trackers on their frequent scouting routes (Berlinger, 2018). This data leaking already poses a significant threat to many, but if it were to happen to individuals, an entire map of their personality could be built up over time and used against them through a variety of their devices.

2.3.2.3 Ransomware

Ransomware attacks have become more high profile in recent years with various attacks such as NotPetya which caused chaos across many businesses by encrypting important files and charging for their recovery (Solon, 2017). Ransomware offers attackers the ability to gain a large sum of money in return for very minimal effort when compared to other attacks. The main function of this attack is to impede functionality of the affected device, meaning that an attack like this on IoT could become even more serious as many unprotected devices used in a variety of areas could risk being damaged with the potential of real-world effects (Ismail, 2017). For example, an attack could be carried out on an IoT insulin pump – which automatically detects when a patient requires insulin and supplies it – by blocking its functionality until payment is supplied, risking the health of those using the pump. Normal ransomware activity is shown in Figure 5

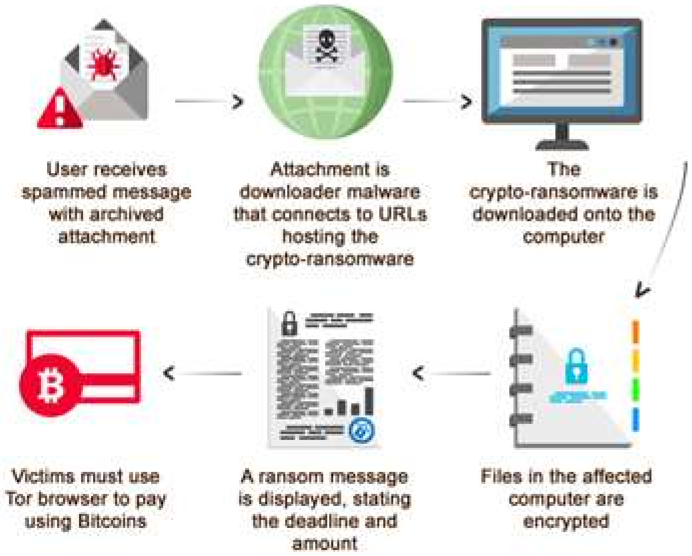

User receives spammed message with archived attachment
Attachment is downloader malware that connects to URLs hosting the crypto-ransomware
The crypto-ransomware is downloaded onto the computer
Files in the affected computer are encrypted
A ransom message is displayed, stating the deadline and amount
Victims must use Tor browser to pay using Bitcoins

Figure 5 - Ransomware activities (Trend Micro, 2015)

2.3.2.4 Remote Surveillance

With IoT devices becoming more integrated into the daily lives of many, the idea of these being exploited becomes even more devastating. Smart homes for example are made up of many different IoT devices which extend the functionality of various household features. If an attacker were to infiltrate a smart home network, then extensive information could be gained through observation of these devices and the personal data which they maintain, allowing an attacker to carry out further attacks in the future. For example, they may wish to exploit security vulnerabilities within internet-connected surveillance cameras to observe activities within private buildings, this could be done by exploiting simple weaknesses such as transmission of plaintext passwords used in insecure software found in cheap camera models (Palmer, 2017).

2.3.3 Mitigating Attacks

Whilst there are many security weaknesses across many IoT devices which cannot be fixed with one solution, there are various aspects which can be improved with broader techniques such as the use of cryptography and authentication mechanisms. One measure of security used across the information security field is the pillars of information assurance (Confidentiality, Integrity, Availability, Authentication, and Non-repudiation) which states that data must not be accessible by outsiders, it must not be able to be tampered with, but it must still remain accessible by the intended recipient, users must prove their identity, and finally that users must be held accountable for all actions (Cherdantseva & Hilton, 2015). Following these standards is an important aspect of IoT due to the collection, transmission, storage, and processing of large volumes of data.

2.3.3.1 Cryptography

The ever-increasing connectivity of devices across the world has been incredibly beneficial but making these communications remain confidential has been a slight challenge. As manufacturers of IoT devices fail to integrate encryption so they can make devices faster and more accessible, it is becoming clearer that a level of respect to customer data needs to be created (Barcena & Wueest, 2015). Increasing usability and accessibility across various hardware and protocols is a challenging aspect when implementing techniques such as encryption but more crucial is the protection of customer data in a world of constant cyber-attacks. Additionally, the lack of processing power and battery power limitations within many IoT devices causes constraints when it comes to including extra security features, making it difficult to create a usable device which also respects user privacy (Chabrow, 2016).

Encryption is a crucial aspect of all computers which handle data, but especially so in IoT due to the constant need to transmit or receive potentially sensitive data across many devices. There are many different encryption standards which offer different benefits based on the protocols, software, hardware, and connectivity of the devices in use. For IoT devices it is useful for the encryption used to be lightweight to accommodate the lack of memory, power, and battery life. Public key encryption is a very secure way of protecting data confidentiality whilst also offering authentication through the use of a public key. Unfortunately, this type of encryption can be very difficult to integrate into IoT devices due to the lack of computational power which is needed to process the keys and ensure that authentication is put in place (SecureRF, 2017). Private key encryption is therefore more likely to be of use in IoT devices to suit the hardware being used. This method involves each party maintaining their own mathematically related private key to encrypt and decrypt communications, which involves less power to utilise but comes with a lack of authentication that may be given by having public keys in place to confirm identities.

Block ciphers such as SIMON and SPECK have been specially created to allow for optimized performance on hardware and software in lightweight applications when using encryption (Assurance & Cryptography, 2018). Additionally, IoT messaging protocols such as MQTT, and CoAP have a variety of inbuilt features to improve security amongst IoT devices such as the integration of authentication features for device-to-device communication and the ability to utilise Transport Layer Security so that data is encrypted end to end and therefore completely unreadable to attackers. Examples of encryption standards are shown below in Table 2.

Cipher	Block Size	Key Size
AES	128	128
Piccolo	64	80
Robin	128	128
Simon	64	96
Speck	64	96

Table 2 - Encryption methods in IoT

Another feature that IoT devices may benefit from is the use of hardware cryptography such as Trusted Platform Modules (TPM) which is an international standard for a secure crypto processor with dedicated cryptographic features such as key management, random number generators, and hash generators which can be used to confirm the identity of the device for improved authentication (Casey, 2016). IoT manufacturers should attempt to integrate modern cryptography tools such as TPM's and disk encryption into their products to meet the high security standards which should be attained to protect customer privacy. Ensuring that the local storage of data on IoT devices is also an important aspect as this may be compromised if the IoT device itself is physically accessed. This involves the use of encryption on data which is at rest by requiring a password from the user to access the data stored.

Hashing is an important aspect in cryptography and one of the most common ways of storing passwords (Hazzard, 2018). A reliable algorithm such as SHA-256 can be used along with salts to store passwords used by the owners of IoT devices to ensure that their login details are secured, and they can remotely access their device. Additionally, hashing can be used to reliably validate firmware updates by giving the bootloader of a device the ability to calculate these values and compare this with the original digest. This is a useful feature that can stop firmware updates which may have been manipulated after they were uploaded (Bigoness, 2018). These signatures mean that the original author of the firmware can guarantee that the firmware is approved of so that attackers cannot manipulate it without the signature changing. Many devices nowadays use a feature called secure boot which ensures that code run on the device is signed and confirmed to be reliable so that integrity is maintained, otherwise unauthorized code is found then the secure boot process prevents the exposure of the device to it (Rane, 2017).

2.3.3.2 Identity Management

Authentication plays a key role in security for the Internet of Things as it proves that users are who they say they are. Since IoT is made up of millions of devices it is important that they can securely establish levels of trust between each other. One way of ensuring devices are authenticated is through the utilisation of Public Key Infrastructure which provides digital certificates to devices proving their identity. Certificates are handed out and signed by reputable Certificate Authorities to authenticate entities, therefore resulting in more reliable communications between devices.

2.4 Communication Protocols

2.4.1 Physical Layer

In the OSI model, the physical layer represents the hardware used by devices to transfer data. In many circumstances, this is the wiring and cabling such as ethernet cables which provide a path for data to travel. Devices in the Internet of Things often use wireless protocols (as shown in Table 3) to transfer data due to the increased mobility rather than cabling (Sharma, Bogale, & Rawat, 2016). This layer is particularly susceptible to attacks as it is the physical transmission of data is possible to be intercepted.

Protocol	Range	Speed	Security Measures
Bluetooth 4.0	Class 1: <100m Class 2: <10m	Up to 25Mbps	Link Key Generation Security Modes Encryption Modes
NFC	<1m	Up to 24Kbps	Power Control Encryption Mechanisms
802.11ac	<50m	Up to 1.3Gbps	Encryption Mechanisms such as WPA2 and WPA3. Authentication Methods
Zigbee	<100m	Up to 250Kbps	Network/Link Keys High Security Mode
ZWave	<100m	Up to 100Kbps	Basic Encryption and Authentication Methods

Table 3 - Comparisons of protocols

2.4.2 Wi-Fi

One of the most common protocols utilised in IoT is Wi-Fi due to its wide coverage and universal support across many devices. It is a wireless protocol which uses a technology referred to as 802.11 for transmitting data over certain frequencies. A Wi-Fi network is usually made up of an access point which Wi-Fi enabled devices connect to, allowing them to then connect to the internet. Since its inception, there have been several iterations of Wi-Fi each with their own improvements over the previous version (Babiker, Abdelrahman, Babiker, Mustafa, & Osman, 2015). Examples of these versions are shown in Table 4.

Version	Description
802.11a	This standard allows for operation at either 5GHz or 3.7GHz frequencies for transmitting data using Orthogonal Frequency Division Multiplexing (OFDM) modulation.
802.11b	Released at the same time as 802.11, this standard provides speeds of up to 11Mbps using a 2.4GHz frequency.
802.11g	Operates at 2.4GHz frequencies to provide a maximum bandwidth of 54Mbps at a distance of up to 140m. This standard utilises special modulation schemes such as OFDM and DSSS.
802.11n	The first standard to utilize multiple antennas to allow for both transmission and receiving of data using 'Multiple Input Multiple Output' technology to allow for simultaneous connections. This standard offers 300Mbps transmission speeds and up to 250m ranges.
802.11ac	The most recent iteration of 802.11. This version allows for simultaneous connections on either 2.4GHz/5GHz bands so there is an option for which band devices can connect to making it much more flexible but also more costly.

Table 4 - Wi-Fi Versions

2.4.1.1 Usage Areas

Wi-Fi is frequently used across many locations such as homes, businesses, and public areas for fast and mobile internet access. The security features, data transfer rates, and communication distances make it useful for various operations. Adoption of Wi-Fi in IoT is extremely high due to the high range and low energy usage compared to other protocols (Parekh, 2017). Wi-Fi IoT devices include mobile devices such as smartphones and laptops which utilise Wi-Fi as a wireless alternative to Ethernet as it offers fast transmissions over the alternatives such as Bluetooth and provides greater range and security features.

2.4.1.2 Technical Review

Wi-Fi is based on the IEEE 802.11 standard which defines a LAN structure made up of Basic Service Sets which are zones in which devices within these communicate with an Access Point. Networks with multiple BSS zones are connected by Distributed Systems allowing them to communicate. Wi-Fi uses radio waves of varying frequencies to transmit data over a network (Escobar, 2015).

For a device to connect to a Wi-Fi network it must first associate with the access point. When Wi-Fi is enabled on a device, it automatically probes the area for signals to discover possible networks. Once the probe is received by an access point, it can then confirm whether the device is capable of joining the network and publicises the network name (also known as SSID) and other information. A low level of authentication is then performed by the mobile device with the AP to prove its identity. Following this, the mobile device will then proceed to send an association request which provides information such as encryption methods and other features to be used (Cisco, 2014).

Wi-Fi can operate in two possible modes, most of the time devices operate using ‘Infrastructure’ mode in which all devices communicate through a single access point such as a wireless router. This means that one device has to first transmit towards the access point and then the access point forwards the transmission towards the destination. The second mode is ‘Ad-hoc’ which is a peer-to-peer transmission mode which allows devices to communicate directly rather than through a centralized AP. Ad-hoc is generally easier to set up due to the lack of access point, but infrastructure mode allows for a more reliable and permanent set up. One downside to Ad-hoc is the increase in system resource usage as the connected devices must maintain the connection to each other without an AP (Hoffman, 2016).

The wireless nature of Wi-Fi provides many benefits such as increased mobility, which is very useful in IoT devices as they may need increased movement. However, Wi-Fi suffers from several limitations due to the use of wireless communication which can affect the reliability of transmissions. Interference can occur when other devices which produce wireless signals interfere with other signals nearby (Callisch, 2010). Another weakness in Wi-Fi transmissions is the obstruction of signals due to environmental obstacles such as walls. This can cause transmission speeds and range to be significantly reduced.

Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) is a basic access mechanism which uses intelligent sensing before transmitting to check if the receiver is busy to ensure that others currently communicating on the same network can maintain their transmissions without disruption (Brenner, 1997). Due to the wireless nature of 802.11, it is difficult to implement such a protocol as they are always either transmitting or receiving to maintain connections. Physically connected networks which use Ethernet generally utilise Carrier Sense Multiple Access with Collision Detection (CSMA/CD) which operates differently as it detects collisions and immediately halts transmissions whereas CSMA/CA is unable to deal with packet collisions. In a Wi-Fi network a lack of collision avoidance can lead to the 'hidden node problem' which occurs when two stations can communicate with an AP but not with each other, similar to that shown in Figure 6 (Kapadia, Patel, & Jhaveri, 2010).

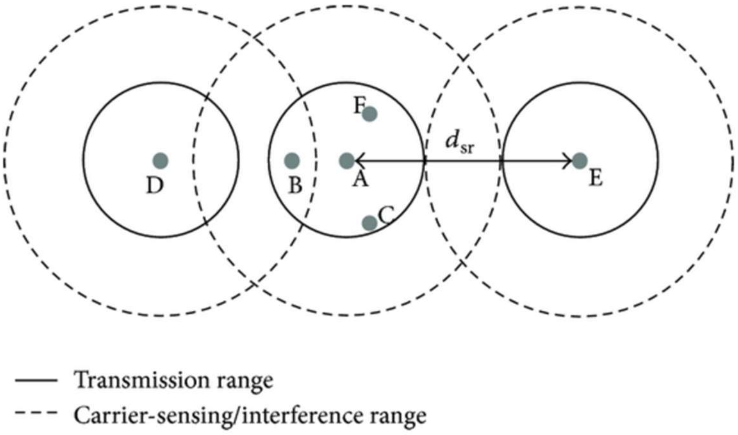


Figure 6 - Hidden node problem (Yoo & Kim, 2013)

2.4.1.3 Security Measures

Wi-Fi offers a selection of security protocols which offer features such as encryption and integrity checks (as shown in Table 5). The first protocol to be introduced was the Wired Equivalent Privacy (WEP) standard which is now largely unused due to the encryption implementation being unreliable. This comes down to the two keys which are used to encrypt data; one of which is the password set up by the user, and the second is a pseudo-random key referred to as an IV. The problem with this method is that the IV has a relatively low number of possible keys which means that it can easily be brute forced (Bulbul, Batmaz, & Ozel, 2008). Despite being a broken standard, it is still used by those who are largely unaware of how the standard works and its weaknesses. This can cause issues when Wi-Fi networks are set up with weak encryption due to insufficient knowledge of security standards, and therefore leave devices vulnerable to attack (Pascucci, 2017). To provide a more secure and reliable alternative to WEP, a new family of standards – named Wi-Fi Protected Access (WPA) – was introduced in 2003. WPA uses Temporal Key Integrity Protocol (TKIP) to provide significant improvements to the encryption of data, more reliable key management, and message integrity checks. Following WPA is WPA2 which provides enhancements over the original with stronger encryption, and additional features such as two separate modes including ‘Personal’ which is used in most home networks due to the easier configuration, and ‘Enterprise’ which utilises additional security measures to ensure that businesses are sufficiently protected against threats (Arana, 2006). This method is the most secure out of the currently available standards but is soon to be followed by WPA3 which offers significant improvements over the previous iterations including, the use of forward secrecy which makes it difficult for an attacker to decrypt old traffic if they gain access to the encryption keys. Additionally, the standard incorporates greater security for public networks by supplying individual users with their own encryption so that data is safe when it is transported even on open networks. Previous standards offer little security for public networks and rely on the user’s own vigilance when it comes to browsing secure web pages (Strain, 2018). WPA3 also reduces the power of brute force attacks by implementing a mechanism for actively managing brute force attempts from unauthorized users. Brute forcing was possible in earlier standards due to the lack of management regarding multiple password attempts (Wiggers, 2018).

Standard	Encryption Method	Features	Status
WEP	RC4	Two Authentication Methods	Insecure
WPA	RC4/TKIP	Wi-Fi Protected Setup	Insecure
WPA2	CCMP/AES	4-Way Handshake, Two Modes	Moderate
WPA3	128-bit Encryption (Personal) 192-bit Encryption (Enterprise)	Simultaneous Authentication of Equals, Forward Secrecy	Secure

Table 5 - Overview of security standards

Authentication techniques are particularly crucial in wireless systems such as Wi-Fi due to the ability for devices to connect without physical access. 802.1x is a standard for authenticating clients who are connecting to an AP which utilizes the Extensible Authentication Protocol (EAP) to support different authentication methods such as the Light Extensible Authentication Protocol (LEAP), EAP Pre-Shared Key (EAP-PSK), EAP Tunnelled Transport Layer (EAP-TTLS), and EAP Subscriber Identity Module (EAP-SIM), all of which add additional authentication mechanisms to the base framework (Mawale, Dakhane, & Pardhi, 2013). The 3 main components to 802.1x are; the client requesting to be authenticated, the server which performs authentication reviews (generally known as a RADIUS server), and finally the authenticator which is the access point that the client is connected to. 802.1x is particularly good for wireless networks due to the low processing power required by the access point for authentication (McNamee, 2013).

2.4.1.4 Common Attacks

Wi-Fi is victim to various types of attacks across all different versions of it. In the context of IoT these attacks may result in smart devices being exploited, data being stolen, and even changes in the physical environment.

Man-in-the-Middle is a common attack which occurs when an attacker intercepts data streams coming from the victim. This generally occurs through the use of eavesdropping using tools such as Wireshark to capture and analyse messages arriving on the interface of a wireless access point. The attacker can then cancel messages and forward their own messages to the original target address (Conti, Dragoni, & Lesyk, 2016). This attack generally occurs when authentication methods are unavailable between the original senders and recipients as they ensure that an attacker cannot imitate anyone. An attacker may utilise this attack in a smart home environment to forward incorrect values to smart devices such as thermometers to alter the temperature of a home causing physical damage.

One method for performing Man-in-the-Middle attacks is with the use of a rogue access point which is set up by an attacker using minimal security measures under the guise of a legitimate looking name such as the name of a local business. Rogue access points can also be used to steal user data such as passwords and other data on websites not using encryption (Alotaibi & Elleithy, 2016). Apart from avoiding suspicious networks, there are various ways to detect and avoid rogue AP's. One method includes the use of an access point whitelist which records trustworthy access points and their relevant MAC addresses so that they can be checked before connecting. This solution comes with the downfall of attackers spoofing their MAC addresses and assuming the identity of a whitelisted AP. Another possible solution is to detect for unusual activity in signal strength as a rogue AP may be situated further away than the authenticated AP so could be detected through such changes of signal (Wu, Gu, Dong, Shi, & Yang, 2018).

In addition to creating rogue access points to capture user data, an attacker may utilise already unsecure access points to gain access to user data. This can come in the form of ‘war driving’ which is the act of roaming around the physical environment with a Wi-Fi enabled device and recording access points and their details such as security levels, GPS location, and whether they use open or restricted access (Priya, Umar, & Sirisha, 2013). The details recorded from this can later be used to attack specific access points.

In some cases, an attacker may not wish to simply steal data from users and may just want to bring down a network to cause disruption. Jamming is the process of spamming a wireless access point with traffic making it difficult for the AP to deal with other wireless clients (Mahoney, 2015). This attack can be performed in various ways to cause different effects on the vulnerable networks; ‘Proactive Jamming’ is the process of consistently sending data packets no matter the environment, and ‘Reactive Jamming’ works by only beginning the process when activity is recorded across the wireless network. Reactive jamming requires a more complex implementation but makes detection of the activity difficult for the victim as only certain messages may be stopped due to jamming (Grover, Lim, & Yang, 2014).

Encryption attacks are also possible but are less likely due to the difficulty, time, and processing power involved. These attacks involve the cracking of encryption keys so that data on a wireless network can be accessed by an attacker. More recently, WPA2 was found to be flawed in an attack known as ‘KRACK’ which allows for attackers to view sensitive information and even manipulate transmitted data (Berghoff, 2017). This attack is less likely to occur due to the design flaw being patched in up-to date devices and the complexity of the attack, however brute forcing attempts can be used to break earlier standards such as WEP which uses simpler encryption methods that can easily be through repetitive login attempts (Davies, 2005).

2.4.3 Bluetooth

Bluetooth is a wireless technology which has been widely used by many electronic devices for short range, low cost, and low energy communications. When it was first invented in 1994, the protocol was intended for data transfer applications but further expanded its uses over several iterations. In addition to the main protocol, two specialised versions were created to suit other applications; Bluetooth Low Energy (BLE) is a more power-efficient version which operates intermittently, making it very useful in IoT devices. The second is Bluetooth Basic Rate/Enhanced Data Rate (BR/EDR) allows for continuous data streams but is more constrained in possible range and power (Nield, 2016).

2.4.2.1 Usage Areas

Bluetooth technology is widespread and has seen much usage in smart phones, laptops, wireless earbuds, smart watches, speakers, cameras, and wireless computer peripherals such as mice and keyboards. Since Bluetooth is of a shorter range than Wi-Fi and other protocols, it can only really be utilised in low-range circumstances such as between a smartphone connecting to a car stereo system or another smart device (Singh, Sharma, & Sharma, 2011). In many cases, the technology is used to form a communication link for devices, for example, wireless peripherals such as earbuds use the technology to transfer audio between a device which it is paired to, and wireless keyboards communicate key presses to the Bluetooth receiver. In certain cases, such as smart phones, the technology offers wider usage such as file transfers between paired devices so users can share documents and media (Hoffman, 2017).

2.4.2.2 Technical Review

Bluetooth operates by using low power, short range frequencies to communicate with other Bluetooth enabled devices. When Bluetooth enabled devices connect to each other they form a piconet in which one device acts as the master and others as the slaves which follow the commands of the master. The master acts as an access point in which other slave nodes communicate through rather than directly to each other. The nodes in each piconet can also be a part of other piconets making them ‘bridge’ nodes so that a ‘scatternet’ can be formed (Lonzetta et al., 2018). Theses nodes can only be active in one piconet at a time but remain parked in the other piconets that they are a part of (Pravin Bhagwat, 2001). Nodes can be tagged in 4 separate states at any one time including:

- **Active** – Node is fully active and communicating
- **Hold** – Nodes go into a sleep state for a period of time, so they cannot communicate at all.
- **Park** – Nodes are temporarily deactivated to conserve energy but can wake up if required.
- **Sniff** – This mode allows for the node to listen to a channel but remain inactive, so they consume less energy.

Bluetooth implements different protocols for creating links between devices. The protocol stack is made up of several layers including the Application Layer, Middleware Layer, and the Transport Layers as shown in Figure 7.

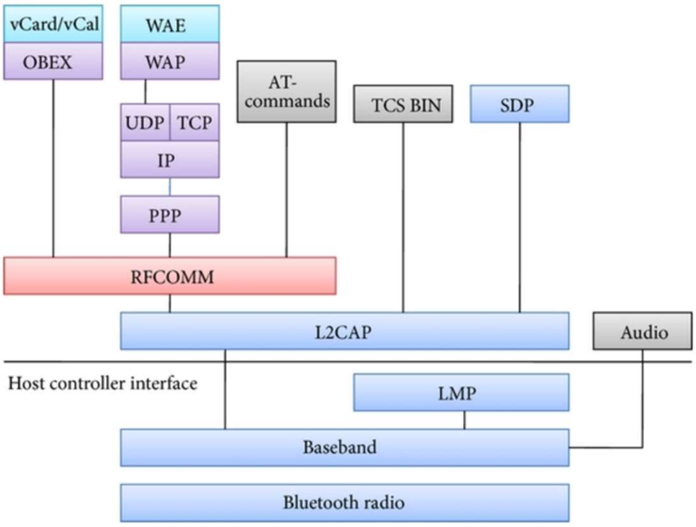


Figure 7 - Bluetooth protocol stack (Park, Mandal, & Park, 2013)

The transport layers are made up of several protocol groups which define how packets are transported including the baseband layer which controls how devices locate and form connections with other devices as well as assigning roles. This layer allows for two link types to be created such as Synchronous Connection Oriented which allows for faster communications when a low latency is needed. This mode of transport does not allow for packets to be resent if an error occurs. The other mode of transport is Asynchronous Connectionless which are much less volatile as retransmissions are allowed if a packet is lost (Haartsen, Naghshineh, Inouye, & Allen, n.d.). The link manager layer controls the properties of the links by setting attributes such as bandwidth, power, and authentication. The middleware layer is made up of both standard and external protocols which allow for applications to communicate over Bluetooth links. Industry standard protocols include Point to Point, Internet Protocol, and Wireless Application Protocol. Finally, the application layers allow for Bluetooth software to utilise the links (Khanpara & Khanpara, 2015).

When Bluetooth devices connect, they utilise a pairing process to authenticate each other and generate a bond. This can occur when a user requests for the process to begin or automatically based on events. This relationship between devices lasts as long as it is maintained but can be removed so that pairing would need to occur again if the devices connected in the future. The pairing process requires for link keys to be established between the connected devices as shown in Figure 8. These keys are generated using cryptographic algorithms and are used to authenticate either device. Once this key is generated, an encrypted link is set up between the devices to secure the transmissions. These keys are maintained by either device so if a key is removed by a device then the bond between devices will be removed (A.Kurawar, A.Koul, 2014).

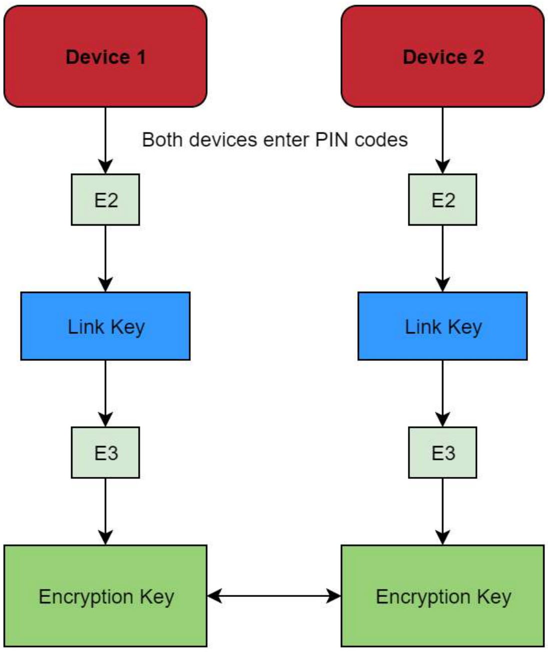


Figure 8 - Link key generation process

2.4.2.3 Security Measures

Bluetooth utilises both software and hardware to implement security controls which make it reasonably difficult to penetrate when compared to other protocols.

Bluetooth addresses allow for devices to be given unique, 48-bit identifiers so that they can be reliably recognised when setting up a connection (Haataja, 2009). These addresses contain values which allow for devices to be recognised based on manufacturer as a vendor ID is given at the beginning of the address.

One feature of Bluetooth is the ability to set the discoverability of the devices to a variety of settings. The first setting is silent which results in the device being uncontactable but allows for traffic to be monitored. The second level is private meaning that the device cannot be discovered unless the unique Bluetooth address is known to the main device. Finally, public mode allows the device to be discovered by any other devices and a connection to be set up (Ošťádal, 2011).

In addition to discoverability settings, there are 4 security modes that can be implemented to provide further security on devices. The first is ‘non-secure’ which offers no further security features, meaning that devices are susceptible to hacking as all authentication and encryption values are ignored. The second security mode utilise a security manager for controlling access to the Bluetooth devices with access control lists. Each device can be given a different trust level, giving it different levels of access to different applications meaning that devices with a higher level of trust can access more sensitive applications. Security mode 3 requires security methods to be set up before channels are created between connected devices. This mode uses both encryption and authentication, making it very secure. The final security mode is a newer introduction into Bluetooth which requires the devices to use Secure Simple Pairing which utilises an Elliptic Curve Diffie Hellman algorithm to generate and share link keys. In this mode several requirements can be set, this can be either an authenticated link key, an unauthenticated link key, or no security at all (Ibn Minar, 2012).

2.4.2.4 Common Attacks

Despite a variety of security measures in place, there still remains weaknesses which can be abused by attackers. In many cases, the easiest way to avoid

Bluetooth attacks is to set the device in use to an undiscoverable status so that malicious entities cannot connect to the potential victim device.

BlueJacking is one possible attack which involves the transmission of unrequested messages between Bluetooth devices which are in a discoverable state. This attack could result in devices being spammed with messages and eventually becoming inoperable, similar to a DoS attack (Bali, 2013).

BluePrinting is a type of attack which can allow an attacker to gain information on a victim’s device. This involves the attacker gaining an understanding of the device details such as the device make and model, and even the current software version. On its own this attack is not particularly devastating, but it could lead to further damage through performing more attacks based on the device details (Mitra, 2017b).

Bluebugging is a type of attack which allows an attacker to listen into conversations by bugging a Bluetooth enabled device.

Another type of attack is the BlueSnarfing attack which involves an attacker gaining access to a Bluetooth device and stealing its contents from memory. This could lead to a loss of sensitive data stored by the user such as contacts, personal data, business-related information, and passwords (Browning & Kessler, n.d.).

3. Methodology

3.1 Introduction

The purpose of this chapter is to present the method by which the analysis and attacks will be performed. A discussion on what is being attempted will be given as well as what is expected from the tests. The tools used to carry out the practical assessments will be reviewed and summarised as to their general usage in penetration testing.

3.2 Problem Statement

As was discussed in previous chapters, Internet of Things devices often fail to achieve a certain level of security which is required to keep consumers and their data safe. Many devices which can be easily purchased and set up by customers contain vulnerabilities right from the point of purchase and can result in massive breaches of privacy. Since IoT devices do not have one particular common use, there are a variety of protocols that are used, each with their own attributes and security flaws.

This experiment aims to perform some in-depth analysis and testing on two particular Internet of Things devices; a Wi-Fi enabled camera and a Bluetooth fitness tracker. Internet enabled cameras are a common feature of many locations such as homes due to their accessibility and ability to enhance the security of the area they are situated (Vena, 2018). Wi-Fi cameras in particular are often cheaper as they can be deployed anywhere with a power source and do not require a complete installation like a normal CCTV camera. Another benefit to these cameras is that they generally come with software which allows the owner to control the camera from a distance as well as back up old footage to the cloud, so camera feeds can be monitored anywhere. Additionally, many of these cameras come equipped with microphones to extend their usage by allowing them to pick up any audio signals being received.

The possibility of an attacker gaining access to a surveillance camera has many potential benefits for the attacker, some scenarios are explained in Table 6. Internet connected cameras are known to be readily accessible due to default passwords which are given when the camera is first manufactured so the customer can easily set up an account on the associated software.

Scenario	Benefits to Attacker
Camera is remotely disabled.	It could be of use to an attacker to disable a camera to either frustrate the victim or possibly make physical attacks easier for them by removing the ability for the victim to review footage of a thief stealing from a house for example.
Camera feed is maliciously monitored.	This could allow an attacker to plan future attacks by learning from what can be seen in the camera feed. If a camera is poorly placed this may even allow an attacker to view passwords typed in on phones or keyboards.
Camera's microphone is accessed.	This attack is could allow an attacker to monitor audio without footage meaning that conversations can be eavesdropped.

Table 6 - Potential attacks on wireless cameras

Fitness trackers are used by many to extend their ability to interact with their mobile device and even perform additional tasks. These devices collect many different types of sensitive data from the user such as location and health statistics (such as blood pressure, heart rate, and step count) which are used by the related mobile application to display the data in useful graphs and give tips to the user regarding their physical performance (Goldstein, 2018). Since Bluetooth is a relatively low-range protocol, it is less likely for an attack on such a device to be as easily carried out and also can be much more difficult as extra hardware can be needed. Although an attack on such a device may be unlikely, it is still a hugely useful target for an attacker due to the information stored. One aspect of the device which is most important is the mobile application which stores the user statistics as it can potentially lead to a leakage of the sensitive data if it is not secured correctly. Table 7 shows what types of attacks could be possible and how they might benefit an attacker in the case of a successful exploit.

Scenario	Benefits to Attacker
Tracker is remotely disabled.	This type of attack does not necessarily benefit the attacker in any significant way other than causing annoyance to the user.
Data is extracted.	Fitness trackers maintain lots of data such as GPS and therefore could allow an attacker to remotely monitor a victim by mapping their location.

Table 7 - Potential attacks on fitness trackers

3.3 Lab Setup

A laptop was set up with a Kali 2019.1 running on Virtualbox to perform the attacks. This software comes with a wide range of tools which are specialised for penetration testing.

The Wi-Fi device being used is the 'iHaven Surveillance Camera' due to its accessible price and popularity when compared to other devices of the same type which suggests that it is owned by a large number of customers making any potential attacks have a much higher impact if an attacker with malicious intent carried out the attacks. The Wi-Fi camera in use was set up according to the instructions given and connected to a 2.4GHz wireless network. This process involved downloading the 'PixPlus' application from the Google Play Store and creating an account on the application. Once this process was completed it was possible to pair the camera with the account using a QR code, resulting in full access to the camera's functionality.

For the Bluetooth tests, the 'iposible Fitness Tracker' was selected as similar to the Wi-Fi camera being tested on, it was of a reasonable price and high popularity. The device was set up by charging it in a USB port and initiating the device once charged. To gain access to the data over Bluetooth, the user must download the 'H-Band 2.0' application from the Google Play store which presents the tracked data in a clearer fashion and offers additional settings.

3.4 Tools

Various tools are required to perform reconnaissance, analysis, and exploitation of the Bluetooth and Wi-Fi devices. Most of the tools required come pre-installed with the latest version of Kali Linux, including:

- **Aircrack-ng** – A software suite which includes tools such as airmon-ng which allows for wireless transceivers to operate in monitoring mode so that data packets can be tracked, and also aireplay-ng which is primarily used to generate traffic which can be used to create a denial of service against a device by bombarding it with many deauthentication frames so that it can no longer respond.
- **BLE Scanner** – An Android application which shows details of nearby Bluetooth devices in a user-friendly GUI. It can detect how close devices actually are and give details on its attributes.
- **Bluetoothctl** – Allows for Bluetooth devices to be directly communicated with via Linux terminal. Operations include pairing and scanning.
- **Gatttool** – A Linux terminal tool which can be used to communicate directly with Bluetooth devices and scan for services or characteristics.
- **Hcitool** – Monitors Bluetooth traffic and gives details on nearby devices via a terminal.
- **Hydra** – Used for cracking logins by passing in values either in the form of a single string or text files containing many possibilities for usernames and passwords. This works by trying every possible combination against a login page so that access can be gained. One downside to this tool is that it struggles when login attempts are limited or when captchas are required.
- **Kismet** – This is a useful wireless network scanning tool which can monitor for access points and clients within range and display them in a useful interface.
- **Metasploit** – Software which is used to scan and exploit hosts using a database of stored vulnerabilities. This can be used to easily configured to target certain hosts and ports as well as set other options based on the vulnerability in use.
- **Nmap** – An open-source network scanner used to discover details such as IPs, MAC addresses, and software by probing hosts. One of the most useful

features of the tool is its ability to perform port scans which can reveal potential vulnerabilities within a host by displaying which ports are open and what services may be running on those ports.

- **nRF Connect** – Another Android application which gives in depth detail of nearby Bluetooth devices but also allows for services to be viewed and explicitly communicated with by sending write or read commands.
- **Spooftooph** – Gives an attacker the ability to alter their Bluetooth interface by changing characteristics such as the Bluetooth address and the name of the device.
- **Wireshark** – This is a frequently used program which is available on both Linux and Windows and is used to eavesdrop on communications across given networks. Much information can be obtained through careful analysis of packets and their headers although this can be useless if traffic is encrypted and there is no way to access it.

4. Implementation

4.1 Introduction

This chapter follows on from the previous by performing the practical tests needed based on the methodology and returning results from the environment described. The discussion will expand on the processes involved in attacking the devices using the software mentioned previously to achieve a variety of results. Different attacks will be performed on each device to understand what the vulnerabilities in each device are and discover the range of possibilities in terms of what can be achieved with a successful attack.

4.2 Wi-Fi Analysis

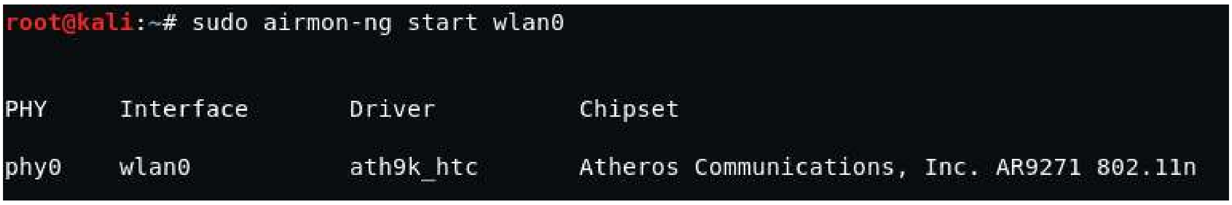
To test the security of the Wi-Fi camera, a number of attacks are attempted based on knowledge gained about IoT security and what common vulnerabilities are used by real world attackers to gain control of devices.

4.2.1 Denial of Service Attack

Surveillance cameras offer great security by offering consumers with the ability to receive a video feed which can be monitored 24/7. The benefits of having surveillance in place can be greatly reduced if an attacker can remotely disable it either temporarily or permanently. One way of doing this is through a denial of service attack which involves spamming the device with packets until it stops operating. The first stage to carrying out this attack is finding out the MAC address of both the camera and the associated access point. One possible way of finding this information can be done through the following process:

1. Start monitoring on the wireless interface.
2. Run Kismet to discover access points and their associated devices.
3. Locate the MAC addresses of the access point and the camera as well as the access point channel.
4. Begin the denial of service process.

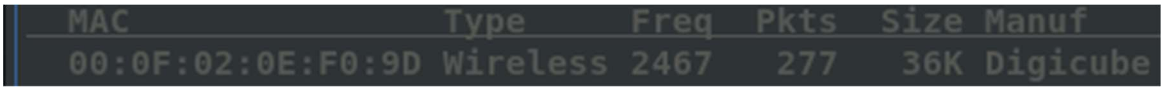
To monitor the wireless interface, the following command (Figure 9) must be run based on the wireless interface you wish to monitor on.



```
root@kali:~# sudo airmon-ng start wlan0

PHY     Interface       Driver          Chipset

phy0    wlan0           ath9k_htc       Atheros Communications, Inc. AR9271 802.11n
```

Figure 9 - WLAN monitoring command

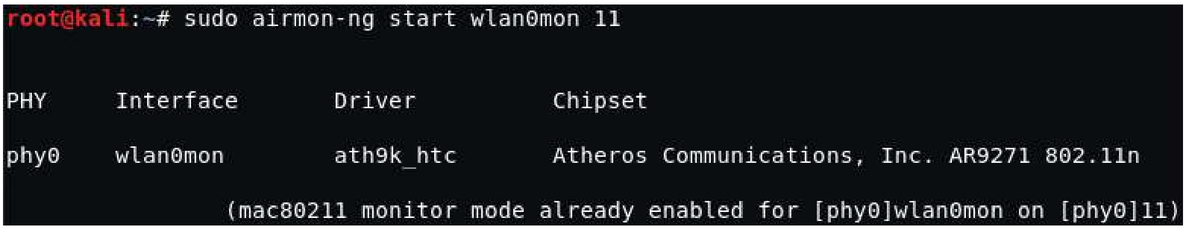
The wireless interface given to the command can be found using ‘ifconfig’. This will set the interface to monitoring mode and it will pick up all wireless signals within range. Kismet can then be used to display this information so that the data needed to perform the DoS is easily visible:



```
 MAC               Type     Freq  Pkts  Size Manuf
 00:0F:02:0E:F0:9D Wireless 2467   277   36K Digicube
```

Figure 10 - Target MAC found in Kismet command

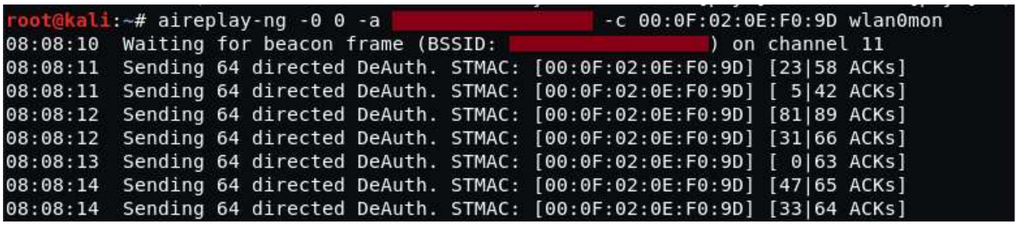
This data can be located by clicking on the target access point in the network details section and observing the client list for the device. If little is known about the target device, then it may be difficult to find straight away so the list must be cut down by ignoring devices which are clearly not the target by looking at the manufacturer column or by searching on the internet for the first 6 values of the MAC address to discover the associated manufacturers. Once the MAC address for both devices and the channel ID for the AP are located, the final stage can be carried out to disable the camera using ‘airmon-ng’ (Thomas, 2011). To pin down the particular channel to attack on the AP the following command must be run:



```
root@kali:~# sudo airmon-ng start wlan0mon 11

PHY     Interface       Driver          Chipset

phy0    wlan0mon        ath9k_htc       Atheros Communications, Inc. AR9271 802.11n

                (mac80211 monitor mode already enabled for [phy0]wlan0mon on [phy0]11)
```

Figure 11 - Final DoS commands

Finally, to execute the denial of service victim must be given by specifying ‘-a’ which sets the target AP MAC address, ‘-c’ which sets the target device MAC address, and the monitoring interface using the ‘aireplay-ng’ command, as seen in Figure 12.



```
root@kali:~# aireplay-ng -0 0 -a [redacted] -c 00:0F:02:0E:F0:9D wlan0mon
08:08:10  Waiting for beacon frame (BSSID: [redacted]) on channel 11
08:08:11  Sending 64 directed DeAuth. STMAC: [00:0F:02:0E:F0:9D] [23|58 ACKs]
08:08:11  Sending 64 directed DeAuth. STMAC: [00:0F:02:0E:F0:9D] [ 5|42 ACKs]
08:08:12  Sending 64 directed DeAuth. STMAC: [00:0F:02:0E:F0:9D] [81|89 ACKs]
08:08:12  Sending 64 directed DeAuth. STMAC: [00:0F:02:0E:F0:9D] [31|66 ACKs]
08:08:13  Sending 64 directed DeAuth. STMAC: [00:0F:02:0E:F0:9D] [ 0|63 ACKs]
08:08:14  Sending 64 directed DeAuth. STMAC: [00:0F:02:0E:F0:9D] [47|65 ACKs]
08:08:14  Sending 64 directed DeAuth. STMAC: [00:0F:02:0E:F0:9D] [33|64 ACKs]
```

Figure 12 - Aireplay-ng output

This results in the camera no longer being able to deal with any other requests as it is flooded with deauthentication frames meaning that the video feed stops responding (as seen in Figure 13) and any attempts to control the camera are denied.

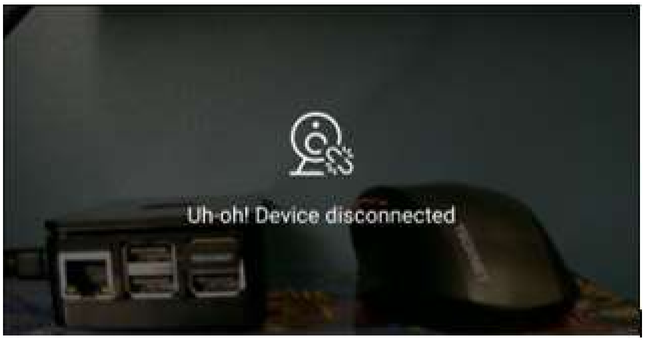
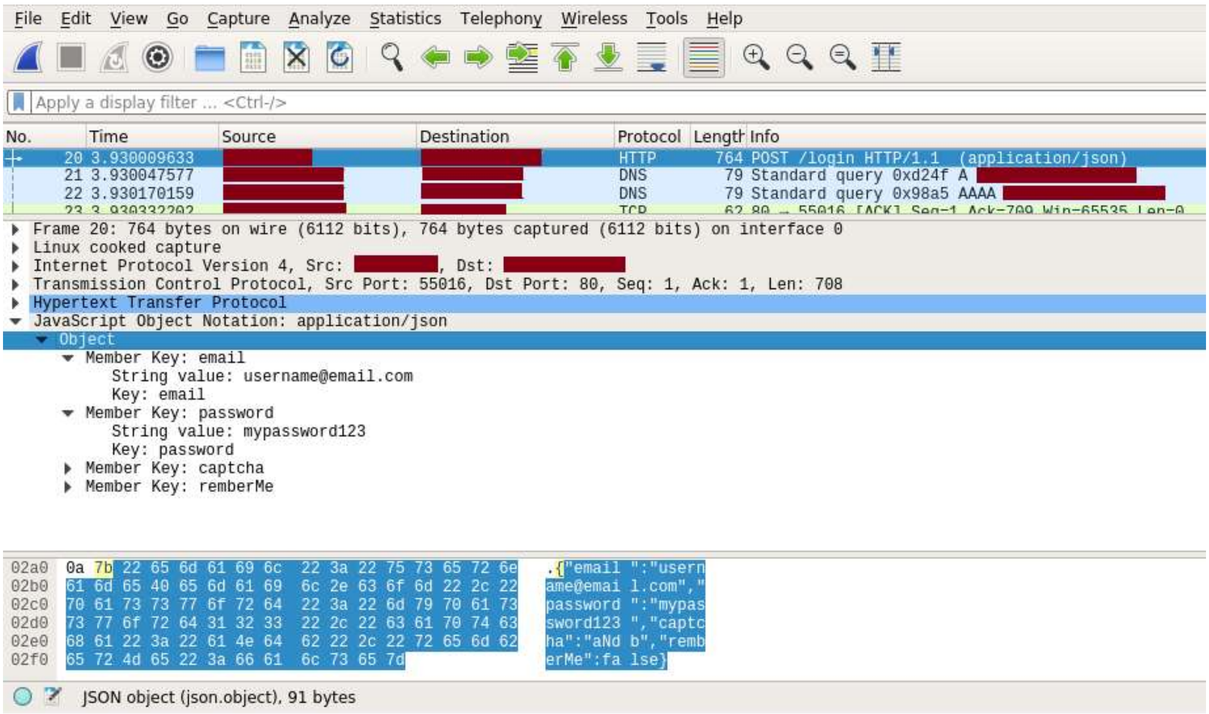


Figure 13 - Video stream no longer responding

4.2.2 Traffic Monitoring

Since it was not possible to easily gain access using default login details through the mobile application, the alternative was to try and discover these values through unencrypted connections. To check if this was possible, a web browser (Firefox) was opened up and the URL for which the camera can be remotely controlled from was entered into the address bar. This led to a login page which used HTTP rather than HTTPS making it possible that the credentials can be stolen using Wireshark. To investigate further, the web page was accessed on Microsoft Edge and also Google Chrome. This investigation showed that unless deliberately specified, the web page will be shown in HTTP except for Chrome which forces the HTTPS version of the web page (Morey, 2018) whilst the HTTP page was shown by default for Edge and Firefox. To take advantage of this lack of secure login on Firefox and Edge, Wireshark was opened, and the wireless channel on wlan0 was monitored. Whilst this was running, login details were entered into the HTTP web page and the Wireshark output was scanned for (as shown in Figure 14), this process can be made simpler by using the ‘tcp.port == 80’ filter to refine the search to only HTTP packets.


File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>

```
No.   Time         Source      Destination  Protocol Length Info
   20 3.930009633  [redacted]  [redacted]   HTTP     764 POST /login HTTP/1.1  (application/json)
   21 3.930047577  [redacted]  [redacted]   DNS       79 Standard query 0xd24f A [redacted]
   22 3.930170159  [redacted]  [redacted]   DNS       79 Standard query 0x98a5 AAAA [redacted]
   23 3.930332202  [redacted]  [redacted]   TCP       62 80 → 55016 [ACK] Seq=1 Ack=709 Win=65535 Len=0
```

```
▸ Frame 20: 764 bytes on wire (6112 bits), 764 bytes captured (6112 bits) on interface 0
▸ Linux cooked capture
▸ Internet Protocol Version 4, Src: [redacted], Dst: [redacted]
▸ Transmission Control Protocol, Src Port: 55016, Dst Port: 80, Seq: 1, Ack: 1, Len: 708
▸ Hypertext Transfer Protocol
▾ JavaScript Object Notation: application/json
  ▾ Object
    ▾ Member Key: email
        String value: username@email.com
        Key: email
    ▾ Member Key: password
        String value: mypassword123
        Key: password
    ▸ Member Key: captcha
    ▸ Member Key: remberMe
```

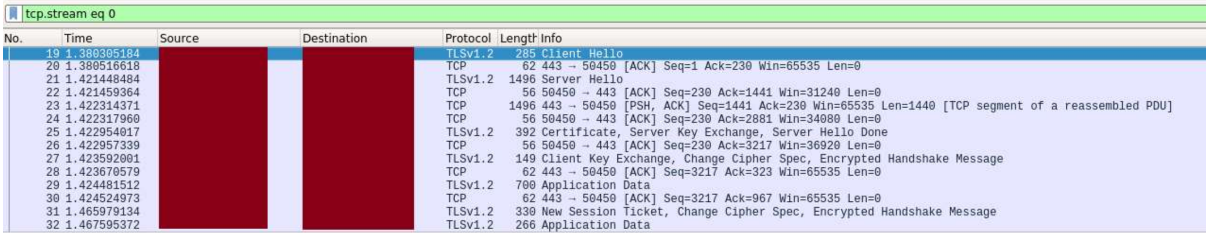
```
02a0  0a 7b 22 65 6d 61 69 6c  22 3a 22 75 73 65 72 6e   .{"email ":"usern
02b0  61 6d 65 40 65 6d 61 69  6c 2e 63 6f 6d 22 2c 22   ame@emai l.com","
02c0  70 61 73 73 77 6f 72 64  22 3a 22 6d 79 70 61 73   password ":"mypas
02d0  73 77 6f 72 64 31 32 33  22 2c 22 63 61 70 74 63   sword123 ","captc
02e0  68 61 22 3a 22 61 4e 64  62 22 2c 22 72 65 6d 62   ha":"aNd b","remb
02f0  65 72 4d 65 22 3a 66 61  6c 73 65 7d               erMe":fa lse}
```

JSON object (json.object), 91 bytes

Figure 14 - Wireshark entry with unencrypted credentials

Once the packet carrying login data (packet 20 in Figure 14) was located, it could then be analysed for login details by searching in the panels below. In a real-world scenario, the attacker could then simply take these details and login as the victim on the vulnerable web page and perform any number of actions.

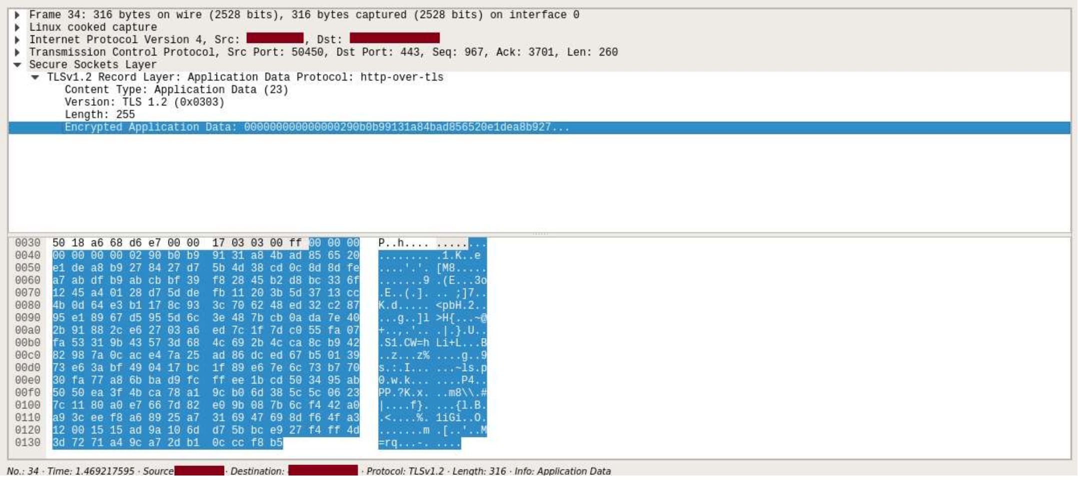
When the rest of the web page was viewed it was clear that none of the pages were encrypted so any user activities could be viewed in the Wireshark feed. Another possible scenario was that the video stream data was delivered on an unencrypted channel. This was once again tested by viewing the page which displayed the video stream and then using Wireshark to search for packets carrying plaintext data (as seen in Figure 15) which could then be converted and viewed.



```
tcp.stream eq 0

No.   Time         Source      Destination  Protocol Length Info
   19 1.380305184  [redacted]  [redacted]   TLSv1.2  285 Client Hello
   20 1.380516618                           TCP       62 443 → 50450 [ACK] Seq=1 Ack=230 Win=65535 Len=0
   21 1.421448484                           TLSv1.2 1496 Server Hello
   22 1.421459364                           TCP       56 50450 → 443 [ACK] Seq=230 Ack=1441 Win=31240 Len=0
   23 1.422314371                           TCP     1496 443 → 50450 [PSH, ACK] Seq=1441 Ack=230 Win=65535 Len=1440 [TCP segment of a reassembled PDU]
   24 1.422317960                           TCP       56 50450 → 443 [ACK] Seq=230 Ack=2881 Win=34080 Len=0
   25 1.422954017                           TLSv1.2  392 Certificate, Server Key Exchange, Server Hello Done
   26 1.422957339                           TCP       56 50450 → 443 [ACK] Seq=230 Ack=3217 Win=36920 Len=0
   27 1.423592001                           TLSv1.2  149 Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
   28 1.423670579                           TCP       62 443 → 50450 [ACK] Seq=3217 Ack=323 Win=65535 Len=0
   29 1.424481512                           TLSv1.2  700 Application Data
   30 1.424524973                           TCP       62 443 → 50450 [ACK] Seq=3217 Ack=967 Win=65535 Len=0
   31 1.465979134                           TLSv1.2  330 New Session Ticket, Change Cipher Spec, Encrypted Handshake Message
   32 1.467595372                           TLSv1.2  266 Application Data
```

Figure 15 - Wireshark encrypted stream output

If an attacker wished to view this data in its raw format, all that would be shown is scrambled messages (Figure 16) which would ultimately give them no useful information.



```
▸ Frame 34: 316 bytes on wire (2528 bits), 316 bytes captured (2528 bits) on interface 0
▸ Linux cooked capture
▸ Internet Protocol Version 4, Src: [redacted], Dst: [redacted]
▸ Transmission Control Protocol, Src Port: 50450, Dst Port: 443, Seq: 967, Ack: 3701, Len: 260
▾ Secure Sockets Layer
  ▾ TLSv1.2 Record Layer: Application Data Protocol: http-over-tls
      Content Type: Application Data (23)
      Version: TLS 1.2 (0x0303)
      Length: 255
      Encrypted Application Data: 0000000000000000290b0b99131a84bad856520e1dea8b927...
```

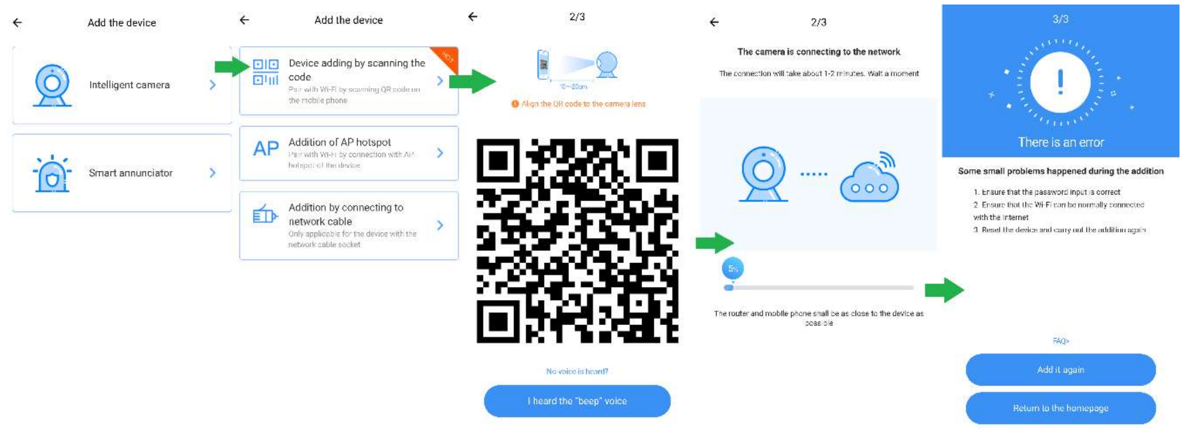
```
0030  50 18 a6 68 d6 e7 00 00  17 03 03 00 ff 00 00 00   P..h.... ........
0040  00 00 00 00 02 90 b0 b9  91 31 a8 4b ad 85 65 20   ........ .1.K..e 
0050  e1 de a8 b9 27 84 27 d7  5b 4d 38 cd 0c 8d 8d fe   ....'.'. [M8.....
0060  a7 ab df b9 ab cb bf 39  f8 28 45 b2 d8 bc 33 6f   .......9 .(E...3o
0070  12 45 a4 01 28 d7 5d de  fb 11 20 3b 5d 37 13 cc   .E..(.]. .. ;]7..
0080  4b 0d 64 e3 b1 17 8c 93  3c 70 62 48 ed 32 c2 87   K.d..... <pbH.2..
0090  95 e1 89 67 d5 95 5d 6c  3e 48 7b cb 0a da 7e 40   ...g..]l >H{...~@
00a0  2b 91 88 2c e6 27 03 a6  ed 7c 1f 7d c0 55 fa 07   +..,.'.. .|.}.U..
00b0  fa 53 31 9b 43 57 3d 68  4c 69 2b 4c ca 8c b9 42   .S1.CW=h Li+L...B
00c0  82 98 7a 0c ac e4 7a 25  ad 86 dc ed 67 b5 01 39   ..z...z% ....g..9
00d0  73 e6 3a bf 49 04 17 bc  1f 89 e6 7e 6c 73 b7 70   s.:.I... ...~ls.p
00e0  30 fa 77 a8 6b ba d9 fc  ff ee 1b cd 50 34 95 ab   0.w.k... ....P4..
00f0  50 50 ea 3f 4b ca 78 a1  9c b0 6d 38 5c 5c 06 23   PP.?K.x. ..m8\\.#
0100  7c 11 80 a0 e7 66 7d 82  e0 9b 08 7b 6c f4 42 a0   |....f}. ...{l.B.
0110  a9 3c ee f8 a6 89 25 a7  31 69 47 69 8d f6 4f a3   .<....%. 1iGi..O.
0120  12 00 15 15 ad 9a 10 6d  d7 5b bc e9 27 f4 ff 4d   .......m .[..'..M
0130  3d 72 71 a4 9c a7 2d b1  0c cc f8 b5               =rq...-. ....
```

No.: 34 · Time: 1.469217595 · Source [redacted] · Destination: [redacted] · Protocol: TLSv1.2 · Length: 316 · Info: Application Data

Figure 16 - Encrypted Wireshark data

4.2.3 Device Reauthentication

During the setup process of the camera, it became clear that there were a number of potential vulnerabilities which could result in an attacker being able to remove the camera from the original owners account and reauthenticating it to the attackers own account. An attacker may take this route because the camera can only be accessed by logging in to the device owners account, which is only possible when you create an account on the application. To test this, a second account was created, and the setup process was attempted once again. To authorize a new camera on an account, the attacker must physically press the reset button then simply display a QR code on their phone to the camera and it will automatically attach to their account (as seen in Figure 17).


Add the device
Intelligent camera
Smart annunciator
Add the device
Device adding by scanning the code
Pair with Wi-Fi by scanning QR code on the mobile phone
HOT
Addition of AP hotspot
Pair with Wi-Fi by connection with AP hotspot of the device
Addition by connecting to network cable
Only applicable for the device with the network cable socket
2/3
10~20cm
Align the QR code to the camera lens
No voice is heard?
I heard the "beep" voice
2/3
5%
3/3
There is an error
FAQ›
Add it again
Return to the homepage

The camera is connecting to the network

The connection will take about 1-2 minutes. Wait a moment

The router and mobile phone shall be as close to the device as possible

Some small problems happened during the addition

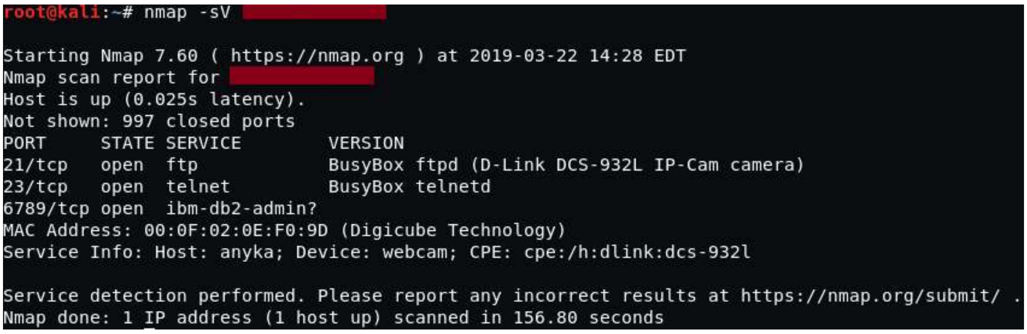
1. Ensure that the password input is correct
2. Ensure that the Wi-Fi can be normally connected with the internet
3. Reset the device and carry out the addition again

Figure 17 - Camera setup process

This process proved that it is not possible to have one device authenticated on more than one account but the process of setting up does potentially pose other threats which will be discussed further on.

4.2.4 Default Credential Attack

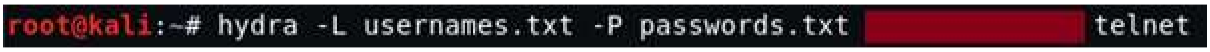
One of the most frequent attacks on IP cameras occurs by taking advantage of the default password which is used to secure the camera. To test what ports were available on the device, the 'nmap' tool was utilised to gain an insight into how the camera operates over a network by performing a version scan on the IP of the camera:



```
root@kali:~# nmap -sV [redacted]

Starting Nmap 7.60 ( https://nmap.org ) at 2019-03-22 14:28 EDT
Nmap scan report for [redacted]
Host is up (0.025s latency).
Not shown: 997 closed ports
PORT      STATE SERVICE        VERSION
21/tcp    open  ftp            BusyBox ftpd (D-Link DCS-932L IP-Cam camera)
23/tcp    open  telnet         BusyBox telnetd
6789/tcp open  ibm-db2-admin?
MAC Address: 00:0F:02:0E:F0:9D (Digicube Technology)
Service Info: Host: anyka; Device: webcam; CPE: cpe:/h:dlink:dcs-932l

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 156.80 seconds
```

Figure 18 - Nmap scan of camera

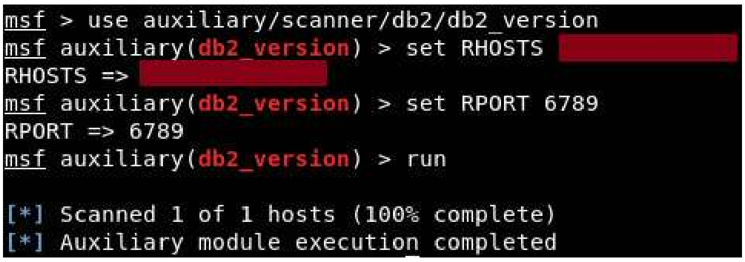
This scan showed that 3 ports of interest were open; FTP (21) which is used for file transfers, Telnet (23) which is an unencrypted communication tool, and ibm-db2-admin (6789) used for maintaining IBM-DB2 databases (IBM, n.d.). The first action to be performed based on this information was to try and brute-force the FTP and Telnet ports to see if access to the underlying storage could be gained due to weak credentials. This was done by downloading plaintexts file with common usernames and passwords to enter into 'Hydra'. To perform the brute force process, Hydra was passed the values of the username and password files as well as the IP of the device and port to attack:



```
root@kali:~# hydra -L usernames.txt -P passwords.txt [redacted] telnet
```

Figure 19 - Hydra password attack

After running this attack on FTP and Telnet, no results were found, and the credentials were unable to be brute forced. Finally, to understand more about the 'ibm-db2-admin' port, an internet search was made using the port name. The first recommendation was 'ibm-db2-admin exploit' which suggests that the service may be vulnerable to some exploits. However, the only exploits available on Metasploit for db2 that could have been of use included one from 2004 which was unlikely to work. To try and gain information on the version, Metasploit was utilised using a specialised module as shown:



```
msf > use auxiliary/scanner/db2/db2_version
msf auxiliary(db2_version) > set RHOSTS [redacted]
RHOSTS => [redacted]
msf auxiliary(db2_version) > set RPORT 6789
RPORT => 6789
msf auxiliary(db2_version) > run

[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

Figure 20 - Scanning ibm-db2 with Metasploit

The search for version returned few results though as the database was secured with credentials. To try and bypass these, a number of educated guesses (with varying capitalisations and combinations) were made using login details such as:

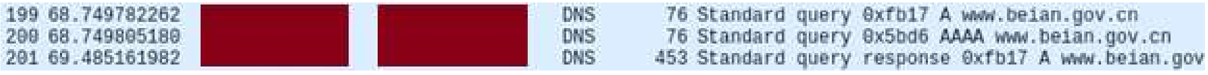
Usernames	Passwords	Result
admin	admin	Unsuccessful
administrator	user	Unsuccessful
user	root	Unsuccessful
root	test	Unsuccessful
test	password	Unsuccessful
ibm-db2	123456	Unsuccessful
ihaven	qwerty	Unsuccessful
username	abcdef	Unsuccessful

Table 8 - Credential usernames and passwords

Ultimately the credentials did not work, and the database could not be accessed or exploited.

4.2.5 Data Handling

Whilst performing Wireshark analysis on the packets shown in previous sections, some information regarding how packets were handled brought up some privacy uncertainties. Several DNS requests were sent all the way to Chinese servers as shown below in the Wireshark trace:



```
199 68.749782262 [redacted] [redacted] DNS     76 Standard query 0xfb17 A www.beian.gov.cn
200 68.749805180                       DNS     76 Standard query 0x5bd6 AAAA www.beian.gov.cn
201 69.485161982                       DNS    453 Standard query response 0xfb17 A www.beian.gov
```

Figure 21 - Suspicious sever queries

Whilst these may not include direct data transfers and are simply DNS queries, it does cause some slight worries regarding user privacy as external transmissions are made without the user realising.

4.3 Bluetooth Analysis

To test the security of the Wi-Fi camera, a number of attacks are attempted based on knowledge gained about IoT security and what common vulnerabilities are used by real world attackers to gain control of devices.

4.3.1 Bluetooth Reconnaissance

The first step in testing the security of the device was to check how easily visible it is to a variety of Bluetooth scanning tools. This can be done using an Android application such as ‘BLE Scanner’ which displays nearby Bluetooth devices and how close they potentially are which is useful when trying to find the device.

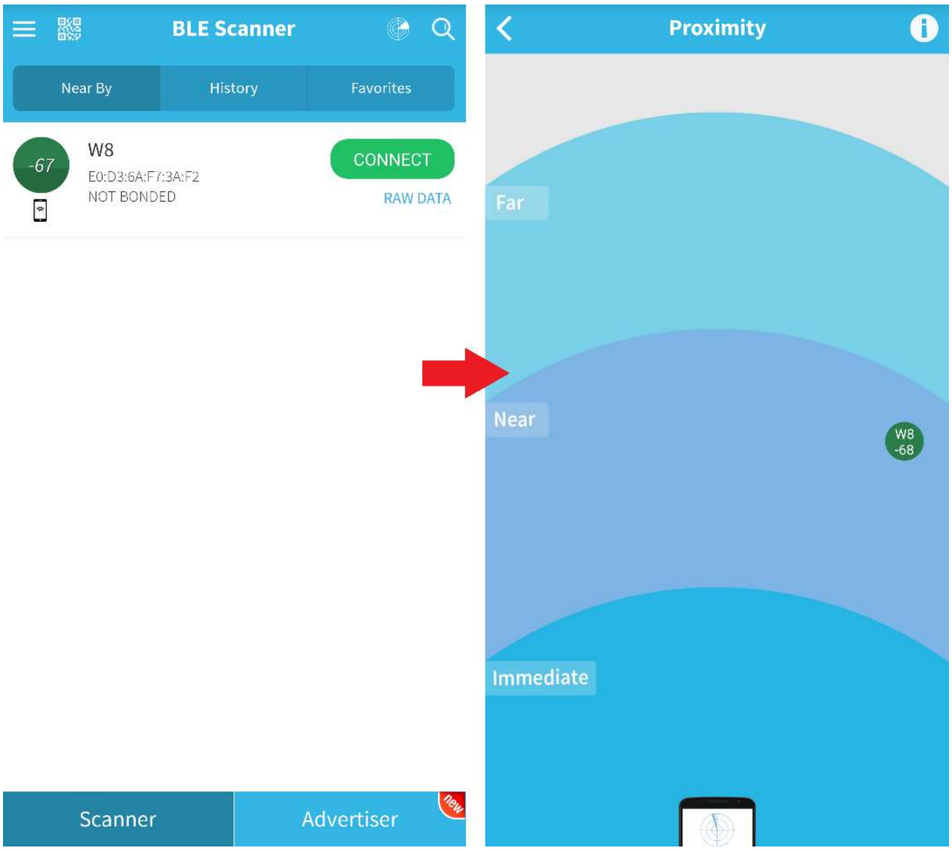
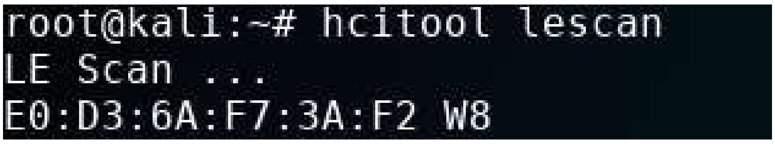


Figure 22 - Bluetooth proximity scan

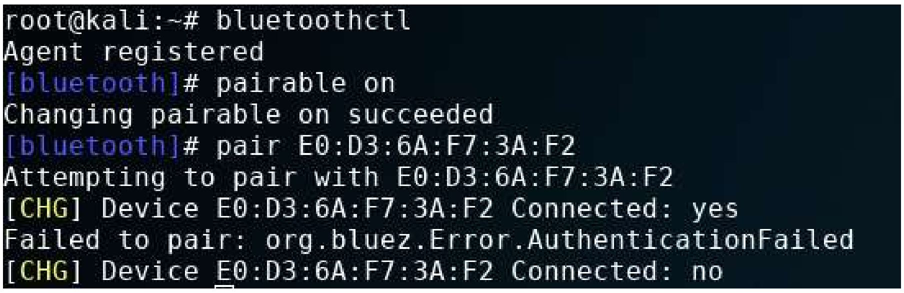
There are several Linux based tools which can be used to try and find further information on the device and its services. To scan for nearby devices, the ‘hcitool’ command was used. This monitors nearby channels for Bluetooth devices which are currently transmitting:



```
root@kali:~# hcitool lescan
LE Scan ...
E0:D3:6A:F7:3A:F2 W8
```

Figure 23 - Scanning for nearby Bluetooth devices

The ‘bluetoothctl’ command can then be used to attempt to pair with the device. This involves setting the Bluetooth interface and its pairable attribute to ‘on’, then pairing with the target using its Bluetooth address:

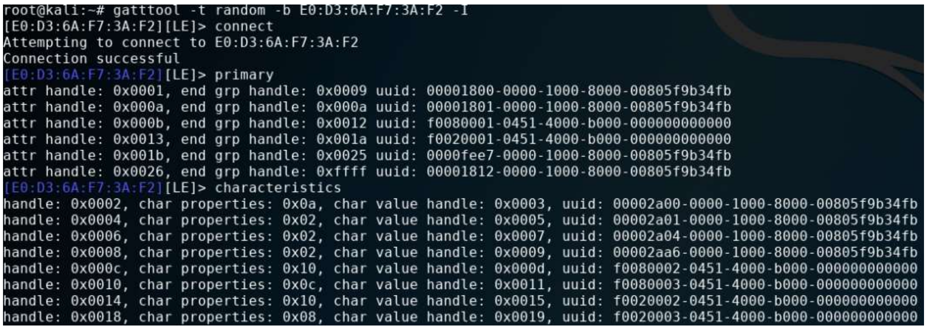


```
root@kali:~# bluetoothctl
Agent registered
[bluetooth]# pairable on
Changing pairable on succeeded
[bluetooth]# pair E0:D3:6A:F7:3A:F2
Attempting to pair with E0:D3:6A:F7:3A:F2
[CHG] Device E0:D3:6A:F7:3A:F2 Connected: yes
Failed to pair: org.bluez.Error.AuthenticationFailed
[CHG] Device E0:D3:6A:F7:3A:F2 Connected: no
```

Figure 24 - Attempting to pair with device

4.3.2 Service Modification

To try and communicate with the fitness tracker directly, it is possible to use ‘gatttool’ (as seen in Figure 25). The command uses the phrase ‘-t random’ which is crucial to communicate to the device as it is programmed to only communicate with mobile devices and this flag manages to bypass that filter (Bedi, 2018). Once the tool is started, the command ‘connect’ can then be sent to directly communicate with the device and scan it for services with ‘primary’.

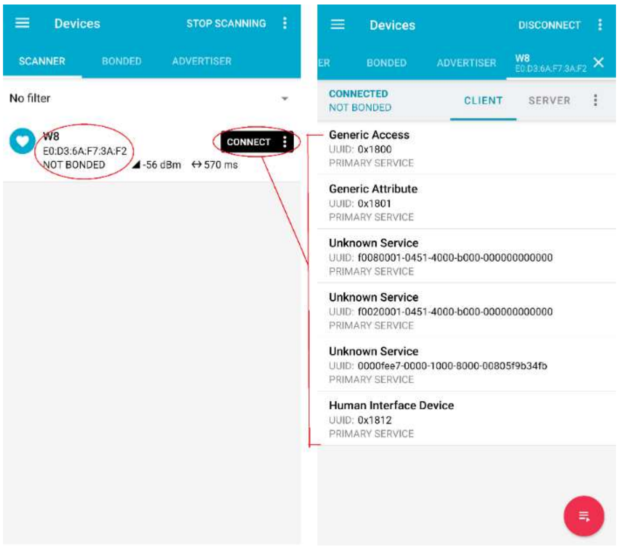


```
root@kali:~# gatttool -t random -b E0:D3:6A:F7:3A:F2 -I
[E0:D3:6A:F7:3A:F2][LE]> connect
Attempting to connect to E0:D3:6A:F7:3A:F2
Connection successful
[E0:D3:6A:F7:3A:F2][LE]> primary
attr handle: 0x0001, end grp handle: 0x0009 uuid: 00001800-0000-1000-8000-00805f9b34fb
attr handle: 0x000a, end grp handle: 0x000a uuid: 00001801-0000-1000-8000-00805f9b34fb
attr handle: 0x000b, end grp handle: 0x0012 uuid: f0080001-0451-4000-b000-000000000000
attr handle: 0x0013, end grp handle: 0x001a uuid: f0020001-0451-4000-b000-000000000000
attr handle: 0x001b, end grp handle: 0x0025 uuid: 0000fee7-0000-1000-8000-00805f9b34fb
attr handle: 0x0026, end grp handle: 0xffff uuid: 00001812-0000-1000-8000-00805f9b34fb
[E0:D3:6A:F7:3A:F2][LE]> characteristics
handle: 0x0002, char properties: 0x0a, char value handle: 0x0003, uuid: 00002a00-0000-1000-8000-00805f9b34fb
handle: 0x0004, char properties: 0x02, char value handle: 0x0005, uuid: 00002a01-0000-1000-8000-00805f9b34fb
handle: 0x0006, char properties: 0x02, char value handle: 0x0007, uuid: 00002a04-0000-1000-8000-00805f9b34fb
handle: 0x0008, char properties: 0x02, char value handle: 0x0009, uuid: 00002aa6-0000-1000-8000-00805f9b34fb
handle: 0x000c, char properties: 0x10, char value handle: 0x000d, uuid: f0080002-0451-4000-b000-000000000000
handle: 0x0010, char properties: 0x0c, char value handle: 0x0011, uuid: f0080003-0451-4000-b000-000000000000
handle: 0x0014, char properties: 0x10, char value handle: 0x0015, uuid: f0020002-0451-4000-b000-000000000000
handle: 0x0018, char properties: 0x08, char value handle: 0x0019, uuid: f0020003-0451-4000-b000-000000000000
```

Figure 25 - Service scan of fitness tracker

The previous figure shows a list of services which the tracker uses to communicate between itself and any connected device. To verify these services, the ‘nRF

Connect’ application can be used by selecting the device and connecting to it (all previous connections must first be disconnected before being able to pair).



W8
E0:D3:6A:F7:3A:F2
NOT BONDED
-56 dBm ↔570 ms

W8
E0:D3:6A:F7:3A:F2

CONNECTED
NOT BONDED

Generic Access
UUID: 0x1800
PRIMARY SERVICE

Generic Attribute
UUID: 0x1801
PRIMARY SERVICE

Unknown Service
UUID: f0080001-0451-4000-b000-000000000000
PRIMARY SERVICE

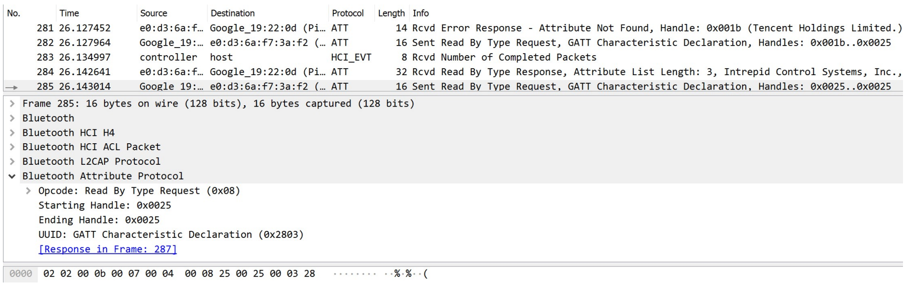
Unknown Service
UUID: f0020001-0451-4000-b000-000000000000
PRIMARY SERVICE

Unknown Service
UUID: 0000fee7-0000-1000-8000-00805f9b34fb
PRIMARY SERVICE

Human Interface Device
UUID: 0x1812
PRIMARY SERVICE

Figure 26 - Services found by nRF Connect

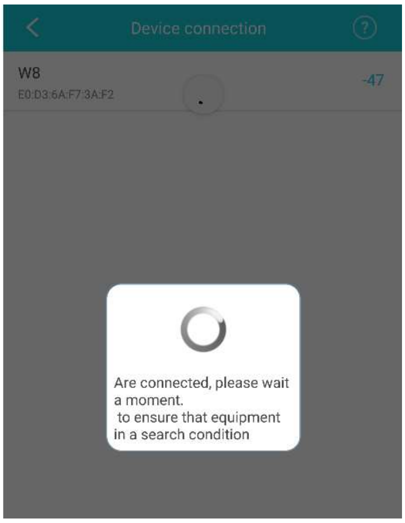
Using this tool, it is possible to view each service and its permissions in a somewhat user-friendly fashion. By selecting each service, it is possible to try and send commands which can lead to the modification of the tracker’s settings (Wolff, 2014). To try and find out what possible values can be used here, the ‘Enable Bluetooth HCI Snoop Log’ option must be enabled on the connected Android device. This will allow for any Bluetooth transmissions to be recorded and stored in a viewable file. To record the data, the ‘Take Bug Report’ option was set in the same section and various actions were carried out in the fitness tracker application and on the tracker itself (Gao, 2016)b. Once this was completed the bug report file was saved and viewed in Wireshark which showed the following communications:



```
No.      Time        Source        Destination          Protocol  Length Info
     281 26.127452   e0:d3:6a:f…   Google_19:22:0d (Pi… ATT           14 Rcvd Error Response - Attribute Not Found, Handle: 0x001b (Tencent Holdings Limited.)
     282 26.127964   Google_19:…   e0:d3:6a:f7:3a:f2 (… ATT           16 Sent Read By Type Request, GATT Characteristic Declaration, Handles: 0x001b..0x0025
     283 26.134997   controller    host                 HCI_EVT        8 Rcvd Number of Completed Packets
     284 26.142641   e0:d3:6a:f…   Google_19:22:0d (Pi… ATT           32 Rcvd Read By Type Response, Attribute List Length: 3, Intrepid Control Systems, Inc.,
→    285 26.143014   Google_19:…   e0:d3:6a:f7:3a:f2 (… ATT           16 Sent Read By Type Request, GATT Characteristic Declaration, Handles: 0x0025..0x0025
> Frame 285: 16 bytes on wire (128 bits), 16 bytes captured (128 bits)
> Bluetooth
> Bluetooth HCI H4
> Bluetooth HCI ACL Packet
> Bluetooth L2CAP Protocol
v Bluetooth Attribute Protocol
  > Opcode: Read By Type Request (0x08)
    Starting Handle: 0x0025
    Ending Handle: 0x0025
    UUID: GATT Characteristic Declaration (0x2803)
    [Response in Frame: 287]
0000  02 02 00 0b 00 07 00 04  00 08 25 00 25 00 03 28   ········ ··%·%··(
```

Figure 27 - Wireshark trace for Bluetooth communications

It is then possible to select communications and view the GATT characteristics. After performing some research on the handles, it was not clear what the purpose of each one was, making the possibility of any further usage more difficult. To test the services which have write permissions, random values were sent to the device using nRF Connect to try and cause the device to stop responding or produce sensitive data. This resulted in the device no longer being able to pair with the mobile application (as seen in Figure 28) as the running services were damaged, however it was still discoverable.

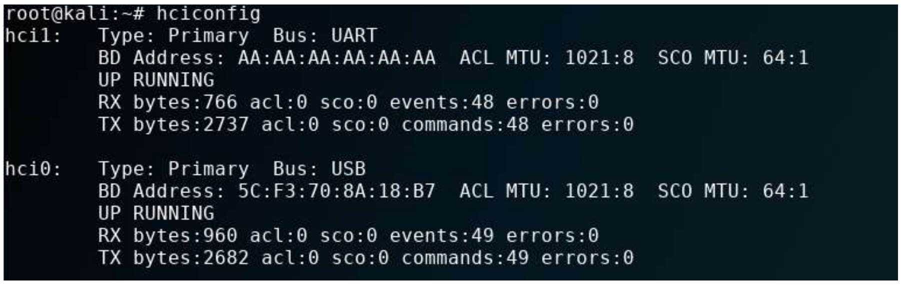


Are connected, please wait a moment.
to ensure that equipment in a search condition

Figure 28 - Failure to pair with fitness tracker

4.3.3 Device Spoofing

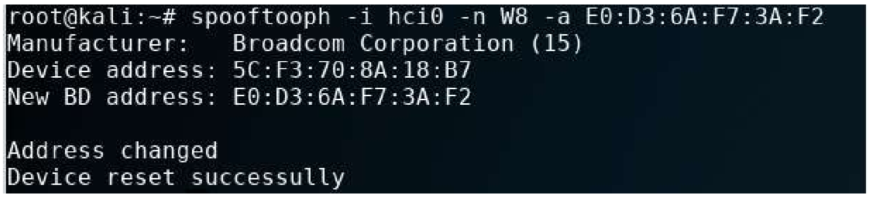
Another way thought to cause issues with the device is through spoofing of the target Bluetooth address. The first step in performing the reconnaissance required using 'hciconfig' to find out the details (seen in Figure 29) of the host Bluetooth interface and ensure that it was active (this will then become the interface used for the spoof attack).



```
root@kali:~# hciconfig
hci1:   Type: Primary  Bus: UART
        BD Address: AA:AA:AA:AA:AA:AA  ACL MTU: 1021:8  SCO MTU: 64:1
        UP RUNNING
        RX bytes:766 acl:0 sco:0 events:48 errors:0
        TX bytes:2737 acl:0 sco:0 commands:48 errors:0

hci0:   Type: Primary  Bus: USB
        BD Address: 5C:F3:70:8A:18:B7  ACL MTU: 1021:8  SCO MTU: 64:1
        UP RUNNING
        RX bytes:960 acl:0 sco:0 events:49 errors:0
        TX bytes:2682 acl:0 sco:0 commands:49 errors:0
```

Figure 29 - Local Bluetooth devices

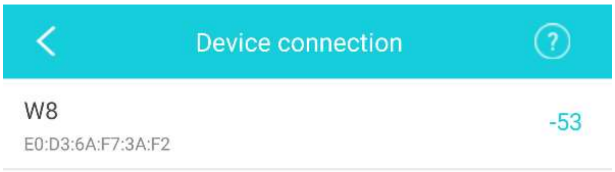
The next stage which involves spoofing (Mitra, 2017a) the device can be done with the 'spooftooph' tool and by specifying the host interface and target Bluetooth device address as seen in Figure 30.



```
root@kali:~# spooftooph -i hci0 -n W8 -a E0:D3:6A:F7:3A:F2
Manufacturer:   Broadcom Corporation (15)
Device address: 5C:F3:70:8A:18:B7
New BD address: E0:D3:6A:F7:3A:F2

Address changed
Device reset successully
```

Figure 30 - Bluetooth spoofing with spooftooph

To check if this process of spoofing managed to trick the fitness tracker application, the device list was checked while the tracker was disconnected to find any additional nearby devices:



Device connection

W8
E0:D3:6A:F7:3A:F2 -53

Figure 31 - Device list on mobile application

Figure 31 shows that only the original fitness tracker is shown, and no further devices were found suggesting that it is not so simple to trick the application.

5. Evaluation

5.1 Introduction

This chapter shows the results obtained from data collected in the previous chapter and presents overall discussions on what these suggest. The results are based on the device analysis performed on the IoT devices and take into account various factors to give fair analysis and evaluation.

5.2 Wi-Fi

The aim of analysing this device was to find out if it was possible to gain access or remotely disable the camera with as little effort as possible. There were a number of attacks which were performed to present a variety of potential scenarios and results which may occur in a successful attack on the device.

5.2.1 Results

After much analysis across the device from the setup process to logging into the web application, many vulnerabilities were discovered but some positive results were found which suggested that the device may not be as insecure as previously thought. After many attempts to break into the open FTP and Telnet ports on the device, it was clear that the credentials were likely more secure than expected which was not expected. Another positive was that the login page for the web application was secured with captcha so brute force attempts could not be attempted as easily but there were issues with HTTPS not being enforced on some browsers.

5.2.2 Discussion

Before carrying out the practical tests, it was expected that the device would suffer from a lot of common vulnerabilities such as common default passwords which makes access to the device itself very simple. After much testing it became obvious that this was not as simple as first expected as it was not possible to brute force access to the open ports or exploit any weaknesses in the services running. Whilst this was a positive overall, it does not necessarily suggest that the device is actually safe from brute forcing as the password and username lists used may have just not been given enough data to perform a successful login. If this were to be successful, an attacker would possibly be able to remotely control the camera and change settings without having to physically contact it which is incredibly dangerous. Another weakness regarding passwords was found in the setup process in which the user login for the application and web page had to be under 26 characters which severely restricts the security of the passwords. With password managers becoming more common (Cao, 2018) it is more likely now that passwords will go beyond these character limits and therefore these limits should be removed to enable greater security from brute force attempts or even hash attacks if the hash was successfully leaked. One problem with the web login process was that it was unprotected in terms of how many attempts could be made but fortunately a captcha system was implemented which made this process more difficult as a third randomly generated image value was required. The mobile application did not contain a captcha or any further authentication beyond login credentials but brute forcing this is less likely than the web application.

The setup process was very simple which makes it easy for those who do not necessarily know how to configure these devices which was a positive for usability but also lead to the device being susceptible to other attack vectors. As mentioned in the methodology, to authenticate the camera to an account the user must first connect to the network they wish to operate on and then display a given QR code to the camera. This process of authenticating very easy and even possible to do from range by simply presenting the code. If an attacker were to have physical access to the camera for even 10 seconds, they may be able to reset the camera and show a QR code to the device which would authenticate it to their account, and they would therefore have control. This could be made even easier if they were to set up their attacking device as a hotspot so that they would not need to be connected to nearby wireless networks. Whilst this attack may be less likely due to requiring physical access and line of sight, it is still a highly possible and dangerous threat which could easily be counteracted by adding additional stages of authentication so that it cannot be so easily transferred between accounts.

Other than the setup process, the mobile application itself seemed secure as no unencrypted transmissions were detected. One feature of the application was the ability to sign up for cloud storage and other features by paying a monthly subscription, this was initially a worry as credentials could easily be transmitted in this process but after briefly attempting this process, the application linked to PayPal which uses encrypted payments making it unlikely for payment information to be stolen. The application itself offers very little in terms of extra layers of authentication as no PIN code is required to access the device list and there was no option to force a login attempt each time it is opened. This could be an issue if an attacker were to gain physical access to the phone and it had no primary authentication measures.

The web application showed a greater weakness as it failed to force HTTPS across web browsers other than Google Chrome. This could lead to users being forwarded unknowingly into an unencrypted web page if Chrome is not used and result in credentials being stolen. Attackers can then use these credentials to log in and perform any number of actions from any location. This issue likely could be fixed with simple additions to the code which ensure that the HTTPS version of the page is forced as it is easily viewable if it is deliberately stated in the URL. The attack which involved using Wireshark to view these credentials is somewhat unlikely unless a victim is specifically targeted and logs into the camera feed during this time but could still be an issue if an attacker were to go to these lengths. Fortunately, during further testing it was discovered that TLS 1.2 was used to ensure that the video stream and so on was encrypted during transport so that attackers could not gain access to it and view the stream. Additionally, actions on the web page are limited other than viewing the device so the potential for attackers to monitor user's activity is unlikely. As well as some encryption issues, there was some odd activity recorded regarding communications with distant servers during activity on the web page which came across as safe due to it just being DNS requests but still prompted some worry regarding user data being recorded and being sent elsewhere. Whilst this may come down to a simple error, such communications could possibly be resolved by relying on servers closer to the user's location so there is greater respect for customer privacy.

Lastly, a denial of service attack was attempted on the device to check if it was susceptible. After some testing it was successfully carried out and the device was no longer operable as the DoS caused it to become busy and therefore unresponsive. As suggested in earlier chapters, this attack is quite common and difficult to counteract as it is not always easily predictable. Therefore, it is unlikely that the manufacturer could implement a solution to the attack but there is still some possibility that IPs could be blocked, or a filter could be applied to incoming traffic which is not from the server which allows the stream to be viewed online or the AP.

From a difficulty standpoint, this was very simple to carry out and is entirely possible for an attacker with little knowledge to perform. This kind of attack may be used to disable the camera to allow for further activities such as theft to occur without the camera recording the crime. The technique used however, only prevents the camera from working for as long as a connection is held, and the command is running.

5.3 Bluetooth

The tests on the Bluetooth device were important as they proved whether or not sensitive data could be interpreted from the fitness tracker and showed to what end it could be remotely controlled.

5.3.1 Results

The results of the testing showed that the fitness tracker used a variety of techniques to remain safe against attackers which ultimately hindered the ability to gain further results as access was restricted. The testing began with some reconnaissance which showed that the device attributes were easily available but only if it was not currently connected to the application. Additionally, the services which were used to control the device were kept to read-only mode, making it very difficult to remotely manipulate the data on the device. However, once the services with write values were bombarded with values, the device stopped being able to pair with the application. As well as this, the mobile application was intelligent enough to not be fooled by spoofed Bluetooth addresses.

5.3.2 Discussion

The Bluetooth fitness tracker proved to be much more secure than the Wi-Fi camera as very few attacks could be performed due to reasonable authentication measures. The first attempt to understand the security was to try and understand how easily it can be located and paired with by non-mobile devices such as the Kali Linux VM. This process proved that once a pairing process was attempted via the bluetoothctl tool, authentication was denied, and the connection was shut down. This security measure ensures that an attacker cannot simply stand nearby to the device and connect to it. As well as this, the device remained undiscoverable once paired with another device so that the connection could not be interrupted as easily which meant that any attacker would need to wait for the tracker to no longer be paired with the victim’s mobile phone before performing an attack.

The next test which was performed was on the services which the device uses to communicate with the host and perform a variety of actions which could not be fully understood from the investigation. This may be partially down to the manufacturer using custom characteristics rather than default ones used in many devices. These services were found to mostly be in read-only mode so that communication with them was impossible which was a positive for the security aspect of the device, but with repeated communications to the writeable services, the device no longer responded to the mobile applications request for pairing. This could be stopped by ensuring that any sensitive services are read-only rather than allowing an attacker to write values to the device. If the handles were researched for longer, it may have been possible to send the device some request transmissions to extract stored data which could contain data about the user's current activity levels. However, after some analysis of the device, it was found that most of the data stored was mostly to do with the user's heart rate, step count, and activity details. If an attack were to occur on the services, it may not reveal anything of use as much of the data such as location were recorded on the paired mobile device and stored using the application. This could be considered a benefit of the device as GPS data is only recorded on the mobile device and is not susceptible to attack from monitoring communications.

The final attack which was attempted involved spoofing the Bluetooth address of the target device using the Kali VM. This was done to verify that the mobile application could not be fooled as otherwise the victim may accidentally connect to the attacker's device and this would therefore allow them to maliciously communicate with the victim's mobile phone. Since this was not possible, it proved that the device pairing process was complimented by more data than just the device address showing that more security measures were in place.

Overall, it would be very unlikely for an attacker to perform an attack on such a device successfully as the protocol contains many security modes which make discovering devices or pairing with devices which are busy difficult. As well as this, Bluetooth is a low range protocol so connecting to such a device in a discrete manner may be difficult as it would require getting close to the device and possibly staying in range for the length of the attack. Furthermore, features such as frequency hopping make attacks like jamming very difficult without some bulky equipment which may be expensive.

6. Conclusions

6.1 Overall Conclusions

In conclusion, the Internet of Things has a security problem which needs to be solved as soon as possible before too much reliance is placed on smart technology. Many people already interact with such devices in their day to day lives, sometimes without realising it and end up putting their personal security at risk. As analysis was performed on a variety of sources in Chapter 2 in which much knowledge was gained about the Internet of Things and a broad overview of security was given which showed what the common weaknesses are in many devices using real world examples. The review gave a clear understanding of what environments IoT devices are used and what data may therefore be at risk. Understanding the types of attacks was a key component and was discussed throughout the literature review to give readers a greater knowledge on what a hacker may try to do. Overall, the literature review proved that IoT is currently suffering from a huge security problem as cheap devices – and even some high-end devices – with no security standards are put to market with very little consequences.

The results of the experiment in this paper showed that devices that are sold at an accessible price on popular online retailers can easily be attacked and manipulated to an attacker’s benefit but in some cases still offer some security features to protect user privacy. The Wi-Fi camera proved to be easily exploited through weaknesses in both its web and mobile application and resulted in personal privacy being broken due to either missing security knowledge or a lack of motivation by the manufacturer to effectively test their device and implement solutions before going to market with their product. The Bluetooth fitness tracker proved to be more reliable as it maintained several good security practices such as making itself not discoverable when already paired with another device and ensuring many of its services are in read-only mode, so an attacker cannot alter them. One positive from the tests is that as some vulnerabilities become more frequent such as default passwords which are often used to enlist IoT devices into botnets, manufacturers make more attempts to stop them from being exploited. Testing of the Wi-Fi camera showed that it was not possible to use default credentials to gain access to the device meaning that there may be more suitable passwords in use. Another worry was that sensitive data may have been leaked by the fitness tracker application, this could not be confirmed as communications were not be completely decoded but much of the data stored on the device from testing was simple data such as heart rate and activity level which would not be of much use to an attacker.

Further discussion showed that some solutions could still be put in place by manufacturers to ensure that their devices are completely secure, particularly in the Wi-Fi camera. The fitness tracker already implemented several security practices and was a lot harder to attack but this was largely due to the difference in protocol functionality and a lack of tools available to attack Bluetooth when compared to Wi-Fi.

6.2 Objective Evaluation

The objectives set for the project as discussed in section 1.3 gave the project a clear focus on what was expected so that reliable conclusions could be drawn from results gained. The objectives were to:

1. Create a comprehensive literature review which clearly discusses the concept of the Internet of Things and provides a background to common security issues.
2. Discuss in detail two common IoT protocols so that penetration tests can be performed with greater accuracy.
3. Perform analysis on Bluetooth and Wi-Fi devices by exploiting vulnerabilities using software and hardware.
4. Evaluate results obtained and discuss the impact of these vulnerabilities as well as the simplicity of recreating the attacks.
5. Based on the results obtained during the implementation, provide potential countermeasures to mitigate attacks.

6.2.1 Objective 1 – Produce a Comprehensive Literature Review

This objective was achieved by using as many sources as possible to build up a bigger picture of the topics discussed so that a clear presentation of IoT, security, and protocols was given to readers in a way that can be understood by those without much knowledge and also provide those with IT skills useful information as well. As well as produce a review of how the Internet of Things works, real-world examples of devices and scenarios were given to show how these attacks could occur.

To gain a broader understanding of the Internet of Things industry and security in general IT, it was crucial to perform a lengthy literature review which tried to cover as much as possible. Since IoT is such an expansive and ever-changing world, it was important to make sure that new references were utilised as well as modern examples of technologies and attacks. Information security terms such as ‘hacking’ and so on often have an unclear definition to those who are less knowledgeable about IT, so it was crucial to give a clear overview of what this really means by delving into common attacks and defence mechanisms and present a range of vulnerabilities which cause attacks to be possible in the first place.

6.2.2 Objective 2 – Gain a Clearer Understanding of IoT Protocols

The two protocols selected were Wi-Fi and Bluetooth as they were easily found in every day devices and many people are likely to have interacted with both of these protocols at some point in the past making the thought of an attack on devices which utilise these more relatable. This objective was completed by using a variety of reliable sources to learn about the selected protocols so that they could be reviewed in a technical fashion whilst also providing material regarding attacks and security measures which can be understood by as many people as possible.

One possible improvement to the literature review could have been to provide more detail on how the protocols work but due to time constraints this was not possible. Both Bluetooth and Wi-Fi were discussed to a reasonable depth without becoming too cumbersome but the addition of another protocol for discussion such as ZigBee may have been useful. Another possibility could have been to discuss many different types of protocols at a lower level, so more ground was covered whilst maintaining the accessibility of the material.

6.2.3 Objective 3 – Perform Analysis on IoT Devices

Several tests were performed on both devices to ensure that as many vulnerabilities could be discovered as possible and to give a clear review on what positive aspects existed. Overall this objective was completed well as the tests covered a lot of ground by covering different areas such as the web application, mobile application, and a variety of attacks such as sniffing and denial of service. Both device tests included as wide a range of tools as possible to try and provide the reader with greater knowledge on the different possibilities of an attack and how complete a hacker's arsenal can be in a real-world situation. The Bluetooth device was much more difficult to perform attacks against due to a lack of practical knowledge regarding the protocol but still proved to be somewhat successful whilst the Wi-Fi attacks turned out to be very successful as more was known about the protocol and the tools used before carrying out the experiment.

This objective could have been improved by taking more time on the practical attacks and learning more about Bluetooth attacks in advance by cutting down time spent on the literature review. A wider range of protocols may have been used if more time was given so that more reliable results could be made. As mentioned earlier in the project, cheaper products often lead to weaker security, this could have been more reliably proven by also testing higher end Bluetooth and Wi-Fi devices which are known to be secure so that a comparison to the cheaper devices could be made.

6.2.4 Objective 4 – Evaluate Results with Regards to Real World Possibilities

To complete this objective, the results gained from the implementation were discussed by explaining how an attacker may use the security vulnerabilities in a real-world situation and how likely such an event would be in the first place. Using common sense and previously gained knowledge on security threats and breaches, brief scenarios were discussed to explain how likely an attacker may be able to carry out an attack and what would ultimately be gained. The results of this objective could have been improved on further by providing detailed examples of the attacks performed in the real world or by extending the discussion even further with a more detailed fictional scenario.

6.2.5 Objective 5 – Provide Potential Countermeasures for Attack

As weaknesses were discovered in both devices, several solutions were mentioned in the discussion stages to potentially counteract attacks on those vulnerabilities. Whilst these solutions were not presented in great detail, some insight was still offered into the right direction that a manufacturer may need to take to secure their device. The solutions for the Wi-Fi device ranged from enforcing HTTPS to creating a stronger authentication process which are relatively simple solutions that can easily be implemented. The Bluetooth device offered greater security than the Wi-Fi camera, so countermeasures were not discussed in as great detail but the defences in place already were mentioned.

One potential improvement for this could be to perform a case study on devices which successfully avoid the vulnerabilities found in the implementation so that further discussion could be had on how the issues covered are solved by other manufacturers.

6.3 Project Evaluation

Overall the project managed to meet the objectives set at the beginning due to some successful planning despite some issues such as time management. The literature review aspect of the project went well as a large area was covered whilst also trying to maintain as wide an audience as possible without making those with less knowledge in the field of IoT feel overwhelmed.

Time management was one difficult area as it was difficult to clearly define how long should be spent on each sub-task of the project. Some early areas were attempted for too long and later stages were hindered as a result of poor time management. One fault in the project plan was that too much time was dedicated to the literature review with very little overlap in the different stages of the project. If more time was allocated to the experiment rather than the literature review, it may have been possible to perform analysis of a third device. This could easily be improved by spending a shorter amount of time on the literature review but also trying to learn even more about the potential practical attacks during this stage so that the process of analysing the devices was simpler. To prevent this from happening, more thought could have gone into the earlier stages of planning the entire project by ensuring that lost time could be made up for and that certain stages did not go on for longer than was needed. The project plan method of a Gantt chart was slightly unreliable as it did not as easily allow for dynamic changes to the timespan of the project so perhaps an additional method could have been used to improve the reliability of the project. Whilst the more time-consuming stages of the project were harder to plan ahead for, the smaller subtasks were easier to perform in advance which gave increased flexibility for the more difficult stages.

The experiment carried out was also a success as many weaknesses were found in the devices and overall lead to a coherent discussion on what the results were, how they may be used by an attacker, and what countermeasures can be put in place to avoid the vulnerabilities. The Bluetooth fitness tracker showed fewer results due to a lack of experience testing Bluetooth devices prior to the investigation but examples of good and bad security practices were still discussed. This aspect of the project could have been improved by practicing with other devices before beginning the main project and taking this practice stage into account during the planning phase. However, the data which was discovered was very insightful and the implementation part of the project for both devices offered a lot of discoveries which lead to a greater knowledge of both protocols. This would mean that future work in these areas could be performed at a higher standard as previous knowledge is held regarding the tools which can be used, and how the protocols and devices themselves work. Another issue regarding the Bluetooth attacks was the lack of tools available to use and the complexity of the protocol meaning that hardware would be needed to perform some attacks such as jamming.

6.4 Further Work

Overall, this project covered only an extremely small portion of devices within the IoT world but could easily be expanded on by testing more types of devices which use different protocols to gain a wider understanding of the damage possible in the event of attacks on these devices. Every day there are new devices and discoveries in the Internet of Things and there are therefore many possibilities for expanding on the research.

One possible route for expanding on the research could be to try and perform more complex attacks on a wider range of devices. Since IoT is such an expansive area, it would be more reliable to take into account as many different devices as possible to gain a greater understanding as to what the most common security flaws are and why they occur. As seen in the Wi-Fi enabled camera, trends such as default passwords are bypassed by using features such as QR codes, so it would be of interest to see how security flaws such as a weak default password are solved by device manufacturers.

Another potential avenue for this research could be in the malware attacks that affect so many devices. This paper focused mainly on the ability for devices to be exploited using basic tools and analysis but with a focus on malware this could result in even more damage occurring. There are many types of malware in IoT and it could be useful to analyse and compare the different possibilities.

References

A.Kurawar, A.Koul, P. V. T. P. (2014). Survey of Bluetooth and Applications. *International Journal of Advanced Research in Computer Engineering & Technology*, *3*(8), 2832–2837. Retrieved from http://ijarcet.org/wp-content/uploads/IJARCET-VOL-3-ISSUE-8-2832-2837.pdf

Abu-Elkheir, M., Hayajneh, M., & Ali, N. A. (2013). Data management for the Internet of Things: Design primitives and solution. *Sensors (Switzerland)*, *13*(11), 15582–15612. https://doi.org/10.3390/s131115582

Alotaibi, B., & Elleithy, K. (2016). Rogue Access Point Detection: Taxonomy, Challenges, and Future Directions. *Wireless Personal Communications*, *90*(3), 1261–1290. https://doi.org/10.1007/s11277-016-3390-x

Arana, P. (2006). Benefits and Vulnerabilities of Wi-Fi Protected Access 2 (WPA2). *Global Journal of Computer Science and Technology*, *612*, 1–6. https://doi.org/10.1016/s0144-2449(05)80306-9

Assurance, I., & Cryptography, N. L. (2018). Notes on the design and analysis of Simon and Speck, (January), 1–23.

Babiker, R., Abdelrahman, M., Babiker, A., Mustafa, A., & Osman, A. A. (2015). A Comparison between IEEE 802.11a, b, g, n and ac Standards. *IOSR Journal of Computer Engineering Ver. III*, *17*(5), 2278–2661. https://doi.org/10.9790/0661-17533034

Bali, R. (2013). Bluejacking Technology : Overview , Key Challenges and Initial Research. *Ijett*, *4*(7), 3020–3024.

Barcena, M. B., & Wueest, C. (2015). Insecurity in the Internet of Things.

Bedi, V. (2018). The Practical Guide to Hacking Bluetooth Low Energy. Retrieved March 11, 2019, from https://blog.attify.com/the-practical-guide-to-hacking-bluetooth-low-energy/

Bell, L. (2018). Half a billion enterprise IoT devices vulnerable to DNS rebinding attacks. Retrieved October 28, 2018, from https://www.theinquirer.net/inquirer/news/3036359/half-a-billion-iot-devices-in-the-office-vulnerable-to-dns-attacks-warns-armis

Berghoff, T. (2017). KRACK attacks against Wifi encryption: here's what you need to know. Retrieved from https://www.gdatasoftware.com/blog/2017/10/30126-krack-attacks-against-wifi

Berlinger, J. (2018). US military reviewing security practices after fitness app reveals sensitive info - CNNPolitics. Retrieved December 1, 2018, from https://edition.cnn.com/2018/01/28/politics/strava-military-bases-location/index.html

Bigoness, E. (2018). Practical Cryptography for the Internet of Things | IoT For All. Retrieved December 2, 2018, from https://www.iotforall.com/cryptography-for-iot/

Bilal, D., Rehman, A.-U., & Ali, R. (2018). Internet of Things (IoT) Protocols: A Brief Exploration of MQTT and CoAP. *International Journal of Computer Applications*, *179*(27), 9–14. https://doi.org/10.5120/ijca2018916438

Bohlin, N., Sehlstedt, U., Kharbanda, V., & Treutiger, J. (2016). Building smart hospitals. Retrieved November 3, 2018, from https://pharmaphorum.com/views-and-analysis/building-smart-hospitals/

Borza, M. (2016). Hardware roots of trust for IoT security. Retrieved November 3, 2018, from http://www.techdesignforums.com/practice/technique/hardware-roots-of-trust-for-iot-security/

Brenner, B. P. (1997). A Technical Tutorial on the IEEE 802 . 11 Protocol Director of Engineering. *Portal*. https://doi.org/10.1098/rspb.2002.2207

Browning, D., & Kessler, G. C. (n.d.). Bluetooth Hacking: A Case Study Dennis Browning. *Digital Investigation*.

Bulbul, H. I., Batmaz, I., & Ozel, M. (2008). Wireless network security: comparison of WEP (Wired Equivalent Privacy) mechanism, WPA (Wi-Fi Protected Access) and RSN (Robust Security Network) security protocols. *Proceedings of the 1st International ICST Conference on Forensic Applications and Techniques in Telecommunications, Information and Multimedia*, (Icv), 1–6. https://doi.org/10.4108/e-forensics.2008.2654

Callisch, D. (2010). Coping with Wi-Fi's biggest problem: interference | Network World. Retrieved December 19, 2018, from https://www.networkworld.com/article/2215287/tech-primers/coping-with-wi-fi-s-biggest-problem--interference.html

Cao, P. (2018). LastPass sees 50% increase in usage thanks to iOS 12 password autofill feature - 9to5Mac. Retrieved March 1, 2019, from https://9to5mac.com/2018/12/19/lastpass-increased-usage-ios-12/

Casey, H. T. (2016). What Is a TPM? How This Chip Can Protect Your Data. Retrieved December 2, 2018, from https://www.laptopmag.com/articles/tpm-chip-faq

Castaldi, C. (2016). IoT In The Warehouse. Retrieved November 3, 2018, from https://www.manufacturing.net/article/2016/07/iot-warehouse

Chabrow, E. (2016). Encrypting the Internet of Things - BankInfoSecurity. Retrieved December 2, 2018, from https://www.bankinfosecurity.com/encrypting-internet-things-a-9382

Cherdantseva, Y., & Hilton, J. (2015). Understanding Information Assurance and Security, *44*(0).

Cimpanu, C. (2018). Z-Shave Attack Could Impact Over 100 Million IoT Devices. Retrieved November 7, 2018, from https://www.bleepingcomputer.com/news/security/z-shave-attack-could-impact-over-100-million-iot-devices/

Cisco. (2014). 802.11 Fundamentals, 1–4.

Conti, M., Dragoni, N., & Lesyk, V. (2016). A Survey of Man in the Middle Attacks. *IEEE Communications Surveys and Tutorials*, *18*(3), 2027–2051. https://doi.org/10.1109/COMST.2016.2548426

Davies, B. (2005). Brute force Attacks with WEPAttack against Static WEP Protected Access Points, (Security 504).

Donnelly, L. (2017). Cyber attack: NHS ordered to upgrade outdated systems as disruption continues. Retrieved October 14, 2018, from https://www.telegraph.co.uk/news/2017/05/15/cyber-attack-nhs-ordered-upgrade-outdated-systems-disruption/

Escobar, E. (2015). How Does Wi-Fi Work? - Scientific American. Retrieved December 16, 2018, from https://www.scientificamerican.com/article/how-does-wi-fi-work/

Francesca Marshall. (2018). Smart traffic lights which always turn green to be trialled on Britain's roads. Retrieved October 25, 2018, from https://www.telegraph.co.uk/news/2018/05/22/smart-traffic-lights-always-turn-green-trialled-britains-roads/

Franceschi-Bicchierai, L. (2017). Internet of Things Teddy Bear Leaked 2 Million Parent and Kids Message Recordings - Motherboard. Retrieved November 7, 2018, from https://motherboard.vice.com/en_us/article/pgwean/internet-of-things-teddy-bear-leaked-2-million-parent-and-kids-message-recordings

Gao, V. (2016). Debugging Bluetooth With An Android App | Bluetooth Technology Website. Retrieved March 7, 2019, from https://blog.bluetooth.com/debugging-bluetooth-with-an-android-app

Goldstein, R. (2018). Council Post: The Benefits Of Fitness And Activity Trackers In The Workplace. Retrieved March 11, 2019, from https://www.forbes.com/sites/forbeslacouncil/2018/09/07/the-benefits-of-fitness-and-activity-trackers-in-the-workplace/#2405879569cf

Grover, K., Lim, A., & Yang, Q. (2014). Jamming and anti-jamming techniques in wireless networks: a survey. *International Journal of Ad Hoc and Ubiquitous Computing*, *17*(4), 197. https://doi.org/10.1504/IJAHUC.2014.066419

Gupta, A., Christie, R., & Manjula, P. R. (2017). Scalability in Internet of Things : Features , Techniques and Research Challenges. *International Journal of Computational Intelligence Research*, *13*(7), 1617–1627.

Haartsen, J., Naghshineh, M., Inouye, J., & Allen, W. (n.d.). Bluetooth : Vision , Goals , and Architecture, 1–8.

Haataja, K. (2009). *Security Threats and Countermeasures in Bluetooth-Enabled Systems*.

Hazzard, K. (2018). Best Practices for Managing Customer Passwords. https://doi.org/DOI10.4207/PA.2011.ART40 Francesco d'Errico at CNRS-University of Bordeaux

Hoffman, C. (2016). What's the Difference Between Ad-Hoc and Infrastructure Mode Wi-Fi? Retrieved December 18, 2018, from https://www.howtogeek.com/180649/htg-explains-whats-the-difference-between-ad-hoc-and-infrastructure-mode/

Hoffman, C. (2017). More Than Headsets: 5 Things You Can Do With Bluetooth. Retrieved February 23, 2019, from https://www.howtogeek.com/165845/more-than-headsets-5-things-you-can-do-with-bluetooth/

IBM. (n.d.). IBM Db2 – Data management software – IBM Analytics. Retrieved March 3, 2019, from https://www.ibm.com/analytics/us/en/db2/

Ibn Minar, N. B. N. (2012). Bluetooth Security Threats And Solutions: A Survey. *International Journal of Distributed and Parallel Systems*, *3*(1), 127–148. https://doi.org/10.5121/ijdps.2012.3110

IEEE. (n.d.). Why We Need Low-Power, Low-Latency Devices | IEEE Innovation at Work. Retrieved January 4, 2019, from https://innovationatwork.ieee.org/why-we-need-low-power-low-latency-devices/

Ion, I., Reeder, R., & Consolvo, S. (2015). ...No one Can Hack My Mind: Comparing Expert and Non-Expert Security Practices. *Eleventh Symposium On Usable Privacy and Security (SOUPS 2015)*, 327–346. Retrieved from https://www.usenix.org/conference/soups2015/proceedings/presentation/ion

Ismail, N. (2017). Ransomware of IoT: the new security nightmare - Information Age. Retrieved December 1, 2018, from https://www.information-age.com/ransomware-iot-new-security-nightmare-123470151/

Jalil, B., & Lumpur, K. (2018). a Review of Latest Wannacry Ransomware : Actions and Preventions, 24–33.

Josh Fruhlinger. (2018). The Mirai botnet explained: How IoT devices almost brought down the internet | CSO Online. Retrieved October 25, 2018, from https://www.csoonline.com/article/3258748/security/the-mirai-botnet-explained-how-teen-scammers-and-cctv-cameras-almost-brought-down-the-internet.html

Kapadia, V., Patel, S., & Jhaveri, R. (2010). Comparative Study of Hidden Node Problem and Solution Using Different Techniques and Protocols, (November).

Khanpara, P., & Khanpara, P. (2015). BlueJacking, *4*(May), 220–227.

Kim, R. (2005). Trusted Platform Module and Privacy : Promises and Limitations.

LaBrie, G. (2017). 6 Benefits of Wireless Networking + Wireless Networking Solutions. Retrieved December 17, 2018, from https://blog.wei.com/6-benefits-of-wireless-networking-wireless-networking-solutions

Leuth, K. L. (2018). State of the IoT 2018: Number of IoT devices now at 7B – Market accelerating. Retrieved November 3, 2018, from https://iot-analytics.com/state-of-the-iot-update-q1-q2-2018-number-of-iot-devices-now-7b/

Lonzetta, A., Cope, P., Mohd, B., Hayajneh, T., & Campbell, J. (2018). Security Vulnerabilities in Bluetooth Technology as Used in IoT. *Journal of Sensor and Actuator Networks*, *7*(3), 28. https://doi.org/10.3390/jsan7030028

Ludlow, D. (2018). What are Z-Wave, Zigbee and other smart home protocols? | Trusted Reviews. Retrieved January 31, 2019, from https://www.trustedreviews.com/opinion/z-wave-zigbee-smart-home-protocols-3426057

Mahoney, J. (2015). Hacking and jamming WiFi networks – Jack Mahoney – Medium. Retrieved January 30, 2019, from https://medium.com/@jackmahoney/hacking-and-jamming-wifi-networks-d2a6ec51f0c2

Maroš, B., Ivan, H., Matej, K., & Petr, H. (2014). Detection of network buffer overflow attacks: A case study. *Proceedings - International Carnahan Conference on Security Technology*, (March 2015). https://doi.org/10.1109/CCST.2013.6922067

Mawale, K. R., Dakhane, D. M., & Pardhi, R. L. (2013). Authentication Methods for Wi-Fi Networks, *2*(3), 356–360.

McAfee. (2018). McAffee Labs Threats Report, (June), 1–27.

McNamee, M. (2013). Why 802.1X is the Best Choice for Large Scale Wireless Network Design. Retrieved January 30, 2019, from https://www.securedgenetworks.com/blog/Why-802-1x-is-the-Best-Choice-for-Large-Scale-Wireless-Network-Design

Meulen, R. van der. (2017). Gartner Says 8.4 Billion Connected &quot;Things&quot; Will Be in Use in 2017, Up 31 Percent From 2016. Retrieved November 3, 2018, from https://www.gartner.com/en/newsroom/press-releases/2017-02-07-gartner-says-8-billion-connected-things-will-be-in-use-in-2017-up-31-percent-from-2016

Mitra, A. (2017a). MAC Address Spoofing for Bluetooth - The Security Buddy. Retrieved March 5, 2019, from https://www.thesecuritybuddy.com/bluetooth-security/mac-address-spoofing-bluetooth/

Mitra, A. (2017b). What is BluePrinting in Bluetooth Security ? - The Security Buddy. Retrieved January 8, 2019, from https://www.thesecuritybuddy.com/bluetooth-security/what-is-blueprinting/

Morey, R. (2018). Google's Enforcing HTTPS - Is Your Site Ready for Chrome 68? Retrieved March 09, 2019, from https://wp-rocket.me/blog/googles-enforcing-https-website-ready-chrome-68/

Napoli, B. (2018). Hacking Smart homes &amp; your personal data... - Cyber Talks. Retrieved November 12, 2018, from https://cybertalks.co.uk/articles/bruno-napoli-writes-hacking-smart-homes-your-personal-data/

Nichols, S. (2015). Lazy IoT, router makers reuse skeleton keys over and over in thousands of devices – new study • The Register. Retrieved December 29, 2018, from https://www.theregister.co.uk/2015/11/26/lazy_iot_skeleton_keys/

Nield, D. (2016). What is Bluetooth? Retrieved January 24, 2019, from https://www.techradar.com/uk/how-to/computing/what-is-bluetooth-1323284

Ošťádal, R. (2011). Evaluation of bluetooth security.

Pal, A. (n.d.). The Internet of Things (IoT) – Threats and Countermeasures - CSO | The Resource for Data Security Executives. Retrieved January 30, 2019, from https://www.cso.com.au/article/575407/internet-things-iot-threats-countermeasures/

Palmer, D. (2017). 175,000 IoT cameras can be remotely hacked thanks to flaw, says security researcher | ZDNet. Retrieved November 30, 2018, from https://www.zdnet.com/article/175000-iot-cameras-can-be-remotely-hacked-thanks-to-flaw-says-security-researcher/

Parekh, J. (2017). WiFi's evolving role in IoT | Network World. Retrieved December 12, 2018, from https://www.networkworld.com/article/3196191/lan-wan/wifi-s-evolving-role-in-iot.html

Park, N., Mandal, B. K., & Park, Y. H. (2013). Sensor protocol for roaming bluetooth multiagent systems. *International Journal of Distributed Sensor Networks*, *2013*(April 2013). https://doi.org/10.1155/2013/963508

Pascucci, M. (2017). Why companies still use the insecure WPA and WEP protocols. Retrieved January 13, 2019, from https://searchsecurity.techtarget.com/answer/Why-companies-still-use-the-insecure-WPA-and-WEP-protocols

Pravin Bhagwat. (2001). Bluetooth : Technology for Short-Range Wireless Apps. *Ieee Internet Computing*, (June). https://doi.org/10.1109/4236.935183

Priya, C. S., Umar, S., & Sirisha, T. (2013). The Impact of War Driving On Wireless Networks.

Quinnell, R. (2013). Vertical vs. horizontal: Which IoT model will thrive? | Embedded. Retrieved February 5, 2019, from https://www.embedded.com/electronics-blogs/other/4422131/Vertical-vs--horizontal--Which-IoT-model-will-thrive-

Rane, A. (2017). IoT security starts with secure boot. Retrieved December 2, 2018, from http://www.embedded-computing.com/embedded-computing-design/iot-security-starts-with-secure-boot

Rayome, A. (2017). DDoS attacks increased 91% in 2017 thanks to IoT - TechRepublic. Retrieved November 30, 2018, from https://www.techrepublic.com/article/ddos-attacks-increased-91-in-2017-thanks-to-iot/

Reetz, S. (2017). SQL Injection - Technical White Paper. *Matrix*, (September), 1–7. https://doi.org/10.1021/bi049888k

Ricquebourg, V., Menga, D., Durand, D., Marhic, B., Delahoche, L., & Logé, C. (2006). The smart home concept: Our immediate future. *2006 1st IEEE International Conference on*

E-Learning in Industrial Electronics, ICELIE, (January), 23–28. https://doi.org/10.1109/ICELIE.2006.347206

Rossi, B. (2016). Privacy and authentication in the Internet of Things - Information Age. Retrieved November 4, 2018, from https://www.information-age.com/privacy-and-authentication-internet-things-123461082/

SecureRF. (2017). Challenges of Cryptography for Low-energy Devices in the IoT - SecureRF. Retrieved December 2, 2018, from https://www.securerf.com/challenges-of-cryptography-for-low-energy-devices-in-the-iot/

Sehgal, A., Perelman, V., Küryla, S., & Schönwälder, J. (2012). Management of resource constrained devices in the internet of things. *IEEE Communications Magazine*, *50*(12), 144–149. https://doi.org/10.1109/MCOM.2012.6384464

Sharma, S., Bogale, T., & Rawat, D. (2016). Physical Layer of Wireless IoT: Enablers and Issues | EAI Blog. Retrieved February 23, 2019, from https://blog.eai.eu/physical-layer-of-wireless-iot-enablers-and-issues/

Singh, P., Sharma, D., & Sharma, A. (2011). A Modern Study of Bluetooth Wireless Technology, *2*(3), 295–307.

Solon, O. (2017). “Petya” ransomware attack: what is it and how can it be stopped? | Technology | The Guardian. Retrieved December 1, 2018, from https://www.theguardian.com/technology/2017/jun/27/petya-ransomware-cyber-attack-who-what-why-how

Strain, L. (2018). WPA3 will secure Wi-Fi connections in four significant ways in 2018 - Malwarebytes Labs | Malwarebytes Labs. Retrieved January 30, 2019, from https://blog.malwarebytes.com/security-world/2018/01/wpa3-will-secure-wi-fi-connections-in-four-significant-ways-in-2018/

Sun, C. (2016). No IoT without IPv6 | Computerworld. Retrieved January 30, 2019, from https://www.computerworld.com/article/3071625/internet-of-things/no-iot-without-ipv6.html

Thomas, S. (2011). Step-by-step aircrack tutorial for Wi-Fi penetration testing. Retrieved March 05, 2019, from https://www.computerweekly.com/tip/Step-by-step-aircrack-tutorial-for-Wi-Fi-penetration-testing

Trend Micro. (2015). Ransomware: What It Is and How You Can Protect Yourself - Security News - Trend Micro USA. Retrieved December 10, 2018, from https://www.trendmicro.com/vinfo/us/security/news/cybercrime-and-digital-threats/ransomware-what-it-is-and-how-you-can-protect-yourself

Uviase, O., & Kotonya, G. (2018). IoT Architectural Framework: Connection and Integration Framework for IoT Systems. *Electronic Proceedings in Theoretical Computer Science*, *264*, 1–17. https://doi.org/10.4204/EPTCS.264.1

Vena, M. (2018). Why Today’s Security Cameras Are Already Obsolete. Retrieved March 02, 2019, from https://www.forbes.com/sites/moorinsights/2018/08/16/why-todays-security-cameras-are-already-obsolete/#b29c792559ec

Vongsingthong, S., & Smanchat, S. (2015). A Review of Data Management in Internet of Things. *KKU Research Journal*, *20*(2), 215–240. https://doi.org/10.14456/kkurj.2015.18

Whitelegg, D. (2017). Scan your app to find and fix OWASP Top 10 - 2017 vulnerabilities. Retrieved November 6, 2018, from https://www.ibm.com/developerworks/library/se-owasp-top10/index.html

Wiggers, K. (2018). What is WPA3, why does it matter, and when can you expect it? | VentureBeat. Retrieved January 30, 2019, from https://venturebeat.com/2018/05/19/what-is-wpa3-why-does-it-matter-and-when-can-you-expect-it/

Wolff, J. (2014). Get Started With Bluetooth Low Energy. Retrieved March 10, 2019, from https://www.jaredwolff.com/get-started-with-bluetooth-low-energy/

Wu, W., Gu, X., Dong, K., Shi, X., & Yang, M. (2018). PRAPD: A novel received signal strength–based approach for practical rogue access point detection. *International Journal of Distributed Sensor Networks*, *14*(8), 155014771879583. https://doi.org/10.1177/1550147718795838

Yoo, H., & Kim, D. (2013). Dynamic channel coordination schemes for IEEE 802.11p/1609 vehicular networks: A survey. *International Journal of Distributed Sensor Networks*, *2013*. https://doi.org/10.1155/2013/827317

Young, C. (2018). Google's Newest Feature: Find My Home. Retrieved November 7, 2018, from https://www.tripwire.com/state-of-security/vert/googles-newest-feature-find-my-home/

Zingbox. (2018). *Medical devices*. *Zingbox*. https://doi.org/10.1007/BF02695761

Zurkus, K. (2018). Default Passwords Aid Satori IoT Botnet Attacks. Retrieved November 4, 2018, from https://www.infosecurity-magazine.com/news/default-passwords-aid-satori-iot/

Appendix 1 – Project Overview

Initial Project Overview

SOC10101 Honours Project (40 Credits)

Title of Project: Evaluating IoT Device Security

Overview of Project Content and Milestones

The Internet of Things is an increasingly relevant technology used to enhance daily life by giving ‘non-smart’ objects such as kitchen appliances the ability to gather and share data by connecting to the internet. While the IoT can provide significant benefits to people, it also suffers from significant security weaknesses which can cause substantial damage to other computing systems, in the form of DDoS attacks for example.

The first part of the report will discuss the subject of IoT and give a technical overview of how it works. Secondly, extensive research will be undertaken into the communication protocols most frequently used by IoT devices to share their data, such as Wi-Fi and Bluetooth.

The final phase will involve reproducing known attacks – such as brute forcing pin codes, device spoofing, and replay attacks - on IoT devices by exploiting weaknesses in their respective communication protocols to make conclusions on the viability for each device protocol in terms of security. Being able to reproduce these attacks will show how simple it is to carry out and for comparisons to be made between protocols, as well as review whether these vulnerabilities still exist on more up to date devices or if they have been solved. The ability to attack these devices will show how important security is in either consumer or non-consumer applications were there may be greater damage caused.

The Main Deliverable(s):

An overview of IoT and its current uses as well as an analysis of what attacking these devices could mean for IoT devices. Additionally, a step by step analysis of how IoT devices using protocols such as Wi-Fi and Bluetooth can be attacked using software.

The Target Audience for the Deliverable(s):

The target audience will be IoT researchers and developers, as well as security engineers. The research could assist those working with IoT devices in ensuring that they are secure by showing how outdated software can result in vulnerabilities being exploited and to ensure that they are aware of the vulnerabilities which exist.

The Work to be Undertaken:

Several common IoT attacks will be recreated and reviewed on physical IoT devices which are susceptible to attack to try and gain an understanding of where the weaknesses are and how they can be solved. Particular protocols such as Wi-Fi and Bluetooth will be researched and understood so that some penetration tests can be carried out on devices using these protocols. For example, one attack which may occur could be a brute force attack on a Bluetooth lock to bypass the pin and unlock the device.

Additional Information / Knowledge Required:

Research into the communication protocols used in IoT devices such as Bluetooth, and Wi-fi will need to be undertaken to gain a clear understanding of how they can be manipulated and used to affect IoT devices. Various software tools will also need to be used so an understanding of these will be required for the practical component of the project to work. For the Bluetooth testing, tools integrated into the Kali platform such as ‘hcitool’, ‘bluesnarfer’, and ‘BlueMaho’. Finally, for Wi-Fi there is a wide range of tools available on both Linux and Windows for vulnerability testing such as ‘Aircrack-ng’, ‘Kismet’, and ‘AirSnort’ which will be used.

Information Sources that Provide a Context for the Project:

A variety of academic papers from quality sources – such as IEEE, and Oxford Academic – may be used and articles from reliable publishers detailing examples of IoT attacks and statistics. Additionally, various textbooks on IoT will be used, including:

- Practical Internet of Things Security – Russell, B. (2016)
- Internet of Things for Architects - Lea, P. (2018)
- Security & Privacy in Internet of Things – Hu, F. (2016)

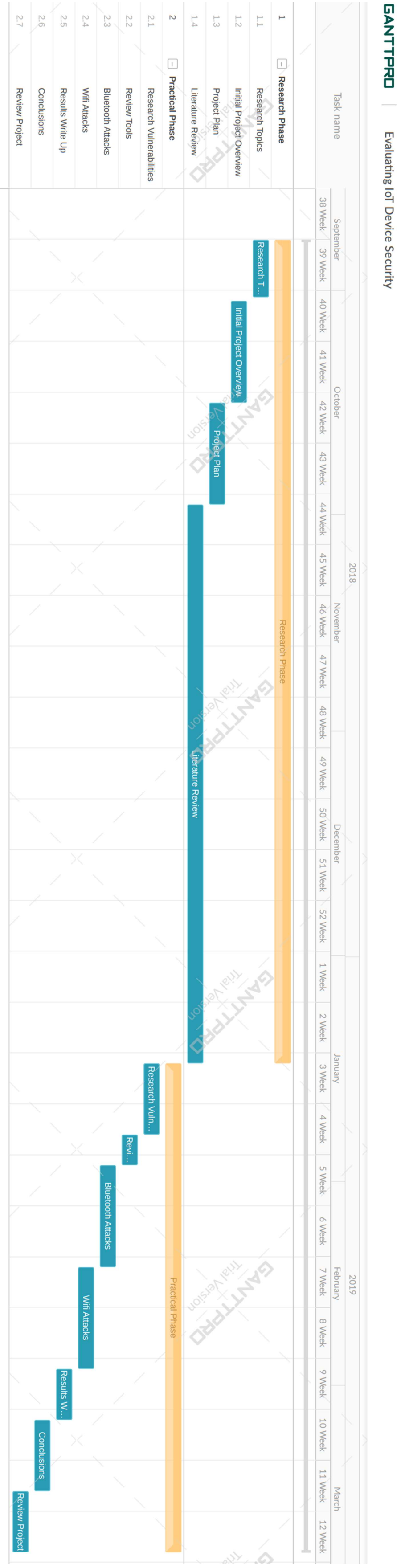
The Importance of the Project:

Security is often overlooked across many areas in the field of computing and it is important now more than ever to remain informed on the vulnerabilities of key technologies. This project will ensure that those working with IoT devices are aware of the possible attacks that could potentially be carried out. The penetration tests on the communication protocols will produce results that will give those with IoT devices a clear understanding of why it is important to keep devices updated with the latest security patches by gaining access to devices with weak security.

The Key Challenge(s) to be Overcome:

Some easily available IoT devices – such as a Bluetooth locks and Wi-Fi cameras – will be tested on meaning that new hardware will have to be understood as well as learning how to utilise tools such as Kali Linux to exploit vulnerabilities in each device used in the project. Additionally, results will have to be converted into useful solutions to provide to those trying to secure their devices.

Appendix 2 – Project Management & Diary Sheets



GANTTPRO | Evaluating IoT Device Security

	Task name
1	**Research Phase**
1.1	Research Topics
1.2	Initial Project Overview
1.3	Project Plan
1.4	Literature Review
2	**Practical Phase**
2.1	Research Vulnerabilities
2.2	Review Tools
2.3	Bluetooth Attacks
2.4	Wifi Attacks
2.5	Results Write Up
2.6	Conclusions
2.7	Review Project

EDINBURGH NAPIER UNIVERSITY

SCHOOL OF COMPUTING

PROJECT DIARY

Student: Fraser Dumayne **Supervisor: Liam Bell**

Date: 11/10/2018 **Last Diary date: N/A**

Objectives:

The main objective for this meeting was to finalise the project idea so that the IPO could be completed as well as briefly discuss the next steps of the project.

Progress:

The main idea for the project was finalised – Security in IoT. The research will take place on various popular communication protocols used by IoT devices such as WiFi and Bluetooth.

Supervisor's Comments:

EDINBURGH NAPIER UNIVERSITY

SCHOOL OF COMPUTING

PROJECT DIARY

Student: Fraser Dumayne **Supervisor: Liam Bell**

Date: 18/10/2018 **Last diary date: 11/10/2018**

Objectives:

This meeting was to give a clearer understanding of the direction that should be aimed for with the literature review as well as discuss the planning which should be involved.

Progress:

A better understanding of what work needs to be completed for the literature review was given. A discussion on what each chapter in the literature review should be made up of as well as how much detail should be put into the security portion and how much should be a general overview. Finally, a discussion was had on what planning should be completed for the entire project, including the possibility of a Gantt chart.

Supervisor's Comments:

EDINBURGH NAPIER UNIVERSITY

SCHOOL OF COMPUTING

PROJECT DIARY

Student: Fraser Dumayne **Supervisor: Liam Bell**

Date: 25/10/2018 **Last diary date: 18/10/2018**

Objectives:

The objective of this week's meeting was to improve my academic writing style by analysing the beginning of my literature review and deciding on improvements to be made.

Progress:

In this meeting we discussed how I could more effectively lay out my chapter plan by having IoT in the first and security in the second. As well as this, some small issues in my writing style were resolved.

Supervisor's Comments:

EDINBURGH NAPIER UNIVERSITY

SCHOOL OF COMPUTING

PROJECT DIARY

Student: Fraser Dumayne **Supervisor: Liam Bell**

Date: 01/11/2018 **Last diary date: 25/10/2018**

Objectives:

The objective of this week's meeting was to review the updates to my literature review and referencing style. Additionally, some discussion for what the next steps in the project are was needed.

Progress:

A discussion was had regarding the future of the project such as what kind of evaluation should be made from the research and a brief overview on what attacks would be possible/useful depending on the data gathered. As well as this, we discussed referencing reports with no authors.

Supervisor's Comments:

EDINBURGH NAPIER UNIVERSITY

SCHOOL OF COMPUTING

PROJECT DIARY

Student: Fraser Dumayne **Supervisor: Liam Bell**

Date: 22/11/2018 **Last diary date: 01/11/2018**

Objectives:

The purpose of this week's meeting was to prepare for the Interim report and to present the current status of my work.

Progress:

Some minor changes were discussed that should be completed such as adding in more diagrams and tables to my literature review, as well as including more citations for some comments.

Supervisor's Comments:

EDINBURGH NAPIER UNIVERSITY

SCHOOL OF COMPUTING

PROJECT DIARY

Student: Fraser Dumayne **Supervisor: Liam Bell**

Date: 24/01/2019 **Last diary date: 22/11/2018**

Objectives:

The objective of this meeting was to review my progress since the last meeting and clean up my literature review.

Progress:

In this meeting we discussed the progress made on my literature review and found some faults such as poor academic writing, and citations of diagrams. We also discussed potential Wi-Fi and Bluetooth devices and tools to be used for the practical experiments.

Supervisor's Comments:

EDINBURGH NAPIER UNIVERSITY

SCHOOL OF COMPUTING

PROJECT DIARY

Student: Fraser Dumayne **Supervisor: Liam Bell**

Date: 14/02/2019 **Last diary date: 24/01//2019**

Objectives:

The objective of this meeting was to review the devices to be used within my implementation and discuss my methodology.

Progress:

In this meeting we reviewed the possible options for devices which could be used in my implementation. This included Bluetooth wearables such as fitness trackers, and Wi-Fi remote cameras as there is a wide selection of possible devices within each category. Additionally, we discussed the possibility of using a smart plug due to the poor security practices known to be found in these.

Supervisor's Comments:

EDINBURGH NAPIER UNIVERSITY

SCHOOL OF COMPUTING

PROJECT DIARY

Student: Fraser Dumayne **Supervisor: Liam Bell**

Date: 14/03/2019 **Last diary date: 14/02/2019**

Objectives:

The objective of this meeting was to discuss the methodology and implementation of my project as well as discuss alterations that could be made to the overall document.

Progress:

In this meeting we discussed the additions I could make to my project such as adding further chapters. We also discussed the poster and changes I could make to that.

Supervisor's Comments:

EDINBURGH NAPIER UNIVERSITY

SCHOOL OF COMPUTING

PROJECT DIARY

Student: Fraser Dumayne **Supervisor: Liam Bell**

Date: 21/03/2019 **Last diary date: 14/03/2019**

Objectives:

The objective of this meeting was to discuss the progress made on the final stages of the project.

Progress:

In this meeting we discussed the practical stage of the project such as what attacks were carried out and reviewed my final chapters including implementation, evaluation, and conclusions.

Supervisor's Comments:

EDINBURGH NAPIER UNIVERSITY

SCHOOL OF COMPUTING

PROJECT DIARY

Student: Fraser Dumayne **Supervisor: Liam Bell**

Date: 28/03/2019 **Last diary date: 21/03/2019**

Objectives:

The objective of this meeting was to review my project.

Progress:

In this meeting we discussed the next stages regarding the poster presentation and briefly discussed my project

Supervisor's Comments: