UNIVERSITY of BRADFORD

Faculty of Engineering and Informatics
Department of Computer Science

Internet of Things Hacking: Ethical Hacking of a Smart Camera

Dissertation Project Report

Mohammed Hamza Javed

A thesis submitted in part fulfilment of the degree of BSc (Hons) in Computer Science for Cyber Security

Supervisor: Dr Daniele Scrimieri

9th May 2023

Abstract

The Internet of Things (IoT) has introduced to the world the power of connectivity and automation. Through the use of a smart device, a task as simple as turning on a light bulb can take place directly from a person's mobile phone. Other, more powerful, devices have the ability to record their surroundings, listen to commands, and even guard and protect a house! These are all commendable advancements, but what happens when one of these devices becomes compromised?

An attacker could gain access to an internet-enabled device and become aware of private and sensitive information - useful for taking over other devices on the network, as well as harvesting data for their own malicious purposes. Within this report, common weaknesses and mitigation paths of IoT devices will be discussed, as well as methods and tools that hackers use to attack.

Furthermore, an ethical hacking demonstration will be deployed against an internet-enabled security camera, Wyze Cam V2. Through the use of a firmware attack, root access was gained, allowing the view of confidential folders, containing video and audio files. Certain security methods demand improvement within this device in order to keep its users safe, which will be discussed further in this report.

Key Words: Penetration testing, firmware, vulnerabilities, smart devices, Kali Linux

Acknowledgements

Firstly, I would like to express my gratitude towards my supervisor, Dr Daniele Scrimieri, for his constant guidance and patience towards my questions. His knowledge, feedback and advice were pivotal in helping towards this report and I am very thankful.

I would also like to extend my sincere appreciation to my friends and family. The completion of my report would not have been possible without their never-ending support and encouragement.

Table of Contents

List of Tables

List of Figures

Chapter 1. Introduction

1.1 Problem Definition

Internet of Things devices have become increasingly prominent in modern times, with over 11.3 billion devices connected worldwide in 2021, and a further projected total of 29.4 billion by 2030 (Vailshery, 2022). According to a report from Aviva, an average of 10.3 internet-enabled devices exist within a regular UK household, with this figure increasing to 15.4 in a home with three children (Aviva, 2020).

While this progression is a sign of our technological era advancing, it also brings with it some increasing security concerns. These concerns mainly come in the form of weak security frameworks implemented on smart devices, which can lead to vulnerabilities that could be exploited by an attacker. Many IoT developers may overlook the security aspect due to the associated costs, but in doing so, leave their devices open to attacks that could give a *black hat hacker* full control of the device, and potentially, the entire network. A black hat hacker is a person who uses their hacking skills for malicious purposes, as opposed to an ethical hacker, who uses their skills for good. Certain attacks they could execute include Man-in-the-Middle attacks, malware attacks, injection attacks, firmware attacks and many others to be discussed within the report. As a result, it is safe to conclude that the more Internet of Things devices within a household, the more at risk the household could be.

1.2 Goal

The goal of this report is to analyse, evaluate and assess the security of Internet of Things devices by exploring common vulnerabilities and weaknesses. This will be achieved through various ethical hacking techniques, as well as using specific tools to assist. Additionally, possible remediation methods will be discussed throughout the report. Furthermore, a proof of concept exploit involving a Wyze Cam V2 security camera will be demonstrated via a firmware attack, in order to showcase the ease at which IoT devices can be compromised.

This report aims to answer the following question: *how secure are Internet of Things devices?* Hopefully, as a result, information and awareness can be spread about the growing lack of safety in regards to internet-enabled devices.

1.3 Scope

Throughout the early sections of this report, common and generic Internet of Things devices will be explored, along with their weaknesses and certain mitigations to prevent these. However, during the later sections, the report aims to focus specifically on the penetration testing of a security camera, namely the Wyze Cam V2. In these sections, a demonstration of a proof of concept exploit will be shown.

The following learning objectives will be referred to throughout:

- Use of cyber security concepts and problems - including threat modelling and a study of various cyber-attacks and remediation methods
- Conduct research into relevant books and essays to gain an understanding of real cyber-attacks and case studies
- Explore the legal, social, ethical and professional issues of cyber security - including a look into the laws implemented and ways that they can be improved
- Explore management of security and security tools used to address vulnerabilities - including Nmap, John the Ripper and Binwalk

1.4 Outline

The report is divided into 8 chapters:

- Chapter 1 is the current chapter that introduces the thesis and its aims and objectives.

- Chapter 2 studies the work of other researchers and evaluates their methods and insights. This chapter also provides details on the background of Internet of Things devices, as well as common security vulnerabilities and methods of remediation. Furthermore, reasons as to why security issues exist and their consequences are explored, in addition to the theoretical background of ethical hacking, security protocols and firmware.

- Chapter 3 takes a look at penetration testing in more depth, introducing methodologies, common attacks, tools used by hackers, and methods of protection against attacks.

- Chapter 4 introduces the smart camera that will be hacked as part of the project, namely the Wyze Cam V2.

- Chapter 5 provides a threat evaluation of the Wyze Cam V2, mentioning possible entry points and attack surfaces.

- Chapter 6 demonstrates a proof of concept exploit for the Wyze Cam V2, showing how a potential attacker could gain root access.

- Chapter 7 discusses the results of the attack, ways the security can be improved, legal, social, ethical and professional considerations, and suggestions for future work.

- Chapter 8 concludes the thesis, summarizing what has been discussed throughout.

Chapter 2. Literature Review

2.1 Introduction to Internet of Things Devices, how they work and advantages of their use

The term 'Internet of Things' was coined in 1999 by Kevin Ashton (Lueth, 2014) and is used to describe "pieces of hardware that […] can transmit data over the internet or other networks" (What are IOT devices | n.d.). Although many people remain unaware, the Internet of Things is all around us: in our homes; our places of work; hospitals; supermarkets, and many more. Popular examples of such devices include:

- Smart light bulbs
- Smart plugs
- Smart cameras
- Remote patient monitoring
- Heart rate monitors
- Baby monitoring cameras
- Amazon Alexa
- Smart watches
- And even smart fridges!

All of these devices have the capability of communicating over the internet, which means they have the capability of being attacked.

Internet-enabled devices contain many sensors and actuators, which send the data they collect to a cloud network. Once there, the data is processed and a certain action can be performed. Many common IoT devices are paired with a smartphone application that receives the data and can notify the user of any information that they may need, or even self-adjust based on the conditions set. For instance, if a smart thermometer has sensed that a room has reached a set temperature, it can automatically adjust itself to perform a new action.

From my point of view, the main advantage of IoT devices is the real-time information that they provide. With a few presses of a button, a person is quickly able to monitor a security camera, or measure their current heart rate. This goes a long way in ensuring that changing circumstances can be adapted to almost instantaneously.

Their ease of access is also extremely helpful as it could allow a person to, for instance, immediately turn on a smart plug whenever they need, as long as they have a proficient internet connection.

Internet of Things devices can also increase production through the automation of tasks. For example, if a smart printer detects that an ink cartridge is low, it can automatically notify the user, or even place an online order, based on its settings. Rather than having to manually monitor the ink usage, this system will allow the user to be more productive without wasting time.

As reported by the International Data Corporation, it is estimated that, by 2025, there will be almost 80 zettabytes (equivalent to 80 billion terabytes) of data produced from IoT devices (Hojlo, 2021). With the increasing rate of Internet of Things devices being used, in my

opinion, the number of cyber-attacks will also inevitably grow, mainly due to the challenges further discussed in section 2.3.

2.2 Related Works

The study and related works of other researchers have helped contribute to this report, and serve as a basis for which I hope to expand upon.

The work undertaken by Fredrik Radholm and Niklas Abefelt (2020) into "A Survey on Security of a Smart Refrigerator" introduces the concepts of ethically hacking a Samsung refrigerator using its underlying Operating System. In this thesis, several cases of threat modelling are used in order to evaluate how secure the machine is, which resulted in no exploitable vulnerabilities being found. The use of threat modelling allowed for a more dynamic view of the machine, which was used to illustrate the potential methods an attacker could use to gain access. In the report, investigations into firmware are limited but have the potential to be scrutinized further.

Ivan Gudymenko thoroughly explained the security weaknesses of Internet of Things devices, mainly in regards to wireless communications, such as RFID and Wireless Personal Networks. His essay, titled "Security in the Internet of Things" (Gudymenko, 2011), opens a discussion into the privacy aspects of internet-enabled devices and the challenges they present.

In a blog post titled "Hacking Reolink cameras for fun and profit", author George Hilliard (2020) hacks into a security camera by modifying and reverse engineering the firmware. These modifications allowed the camera to run telnet from boot, which could be used to connect and gain root access.

The work of a security researcher, with the YouTube screen name of Stacksmashing, explains the uses of the Squashfs and JFFS2 filesystems that are used in firmware images. His online video details the steps taken to inject modified firmware into a smart device in order to take control of it (Stacksmashing, 2020).

A lecture given at a Sydney-based University by Jason Ford (2020), goes into detail about security vulnerabilities of Internet of Things devices, as well as a demonstration of how to hack a security camera. This is done through code injection and brute-forcing the admin password.

A report by Xin Liu (2021), titled "Ethical Hacking of a Smart Video Doorbell", expands upon the use of penetration testing tools to explore vulnerabilities within a smart doorbell. Using attacks such as Man-in-the-Middle, SQL Injection, and more, the security of the device was assessed, and suggestions were made for improvement.

All of these research pieces present interesting and unique ideas. One of special interest to me was George Hilliard's blog post, in which he used Wireshark to sniff the data traffic between a security camera and its PC client, as part of his enumeration of the device. From my experience, this is a resourceful method that can be used to gain sensitive information about a device and the servers it communicates with, as well as leading down other useful paths.

As well as this, Gudymenko's thorough explanation of the security issues in IoT devices that use wireless communications was highly informative. The use of wireless networks to hack a device revealed to me the increasingly large dangers of IoT devices.

Although they were all very educative and enlightening, one area that I found limited in the reports I read would be the use of case studies and examples. Delving in to more relevant details of real life events would provide extra context to the reader and ensure that they are following along. Hopefully, this can be developed further within this report.

2.3 Security Issues and Vulnerabilities

With all of the advantages that Internet of Things devices bring, they also have some disadvantages. Some developers of IoT devices may not implement strong security functions within the device, which can lead to vulnerabilities that an attacker could exploit. These include:

2.3.1 Hardcoded and Default Credentials

When an IoT device is manufactured, it may come with default credentials, such as admin:admin or admin:password. These are basic login credentials that are very simple and easy to remember (which are the characteristics of a weak password!), with the expectation being that the consumer changes these to their own unique, secure password. Many users, however, may not know how to change their credentials or may be ignorant of the fact that they can even be changed, causing them to keep the weak default credentials!

By keeping the default credentials, an attacker could brute-force the details within seconds, or could even do a quick Google search for the device name and password. This is one of the easiest ways that an attacker could enter a device and it remains so because users do not make their details unique and difficult to guess.

2.3.2 Large Attack Surface

During development, some common services that the developers may use include SSH, Telnet and certain debug interfaces, which may have ports that are left open at the time of sale. While these may be critical for the developers, they are not as useful to the average consumer and could instead provide a path for attackers to exploit.

The more internet-connected services that are open on the device, the more potential gateways an attacker has to attack. From there, the attacker could gain access to the device in question or they could pivot and attack another device on the network (Langkemper | n.d.).

Closing unnecessary ports is paramount in reducing the size of the attack surface and helping secure the device, as it means that the attacker has less pathways to enumerate the device and execute an attack.

2.3.3 Lack of Encryption

When devices communicate with each other in plain-text, their messages can easily be read by someone with malicious intentions. This is where encryption comes into use. Encryption transforms readable, plain-text data into an unreadable format, with the help of cryptographic algorithms. This ensures that any information cannot be interpreted by an attacker.

One common problem in this area is using an insecure protocol, such as HTTP or FTP, rather than the encrypted version of the respective protocol, HTTPS or FTPS.

For devices that communicate in plain-text, an attacker could execute a Man-in-the-Middle attack where the attacker sits between two communicating devices and intercepts their data, unknown to the devices. This can lead to sensitive data exposure if any personal information is transmitted whilst the attacker is listening.

2.3.4 Lack of update patches

Devices with vulnerabilities require a software patch to alleviate the issue. If a developer neglects the release of patches to vulnerabilities in their device, their entire customer base could be at risk of exposure to an attack, which is why it is important for software to have an update functionality so that developers are able to push patches at any time, in the case of a system weakness (Guest, 2022).

After a software update is released, the update should be pushed onto new devices at the time of sale so that new users are not at risk of using outdated software, with the vulnerability still hiding within.

2.3.5 Use of software with known vulnerabilities

Rather than reinventing the wheel each time a new IoT device is made, many manufacturers resort to using open source software to create their product. This could include libraries or other popular frameworks that may be used in hundreds of other devices. While this may help them to release their product faster, the software could have many vulnerabilities hidden within, and if an attacker were to find just one vulnerability, it could bring down all of the devices that use that same software (Cloudflare, 2020).

An example of this happening is the 2021 Log4j attacks that left millions of devices exposed around the world. This was due to all of the devices using the same Java-based logging and monitoring tool, where a single security weakness lead to a remote control execution exploit being available within a few days. Many attacks were issued due to the sheer amount of devices that used Log4j.

2.3.6 Injection Attacks

One of the easiest and most common attacks that can impact the Internet of Things is an injection attack. This is where an attacker exploits a user input field (such as a username or password field) and enters malicious code that is processed and sent to the back end of the device, where it is executed; essentially giving the attacker ‘control’ of the device. It can also be used to reveal private and sensitive information, that may be stored in a database, and return these to the attacker.

By sanitizing the user input, i.e. preventing unsafe characters from being entered, this type of vulnerability can be avoided. In addition to this, validation requires that the user input be entered in the correct format that is expected, and as such, any code that is entered will be ignored (Cloudflare, 2020).

2.4 Reasons for security issues

In my opinion, one of the main reasons for weak IoT devices is: because of the increased cost of production, many IoT developers tend to overlook the security aspect of their devices. IoT devices are produced on a mass scale, and an average manufacturer could produce hundreds of thousands of the same product per week. Many of these developers may add a simple security function, but ultimately a well-implemented security framework will increase the company's costs and time spent per product, eating into their profit margins. As a result, developers would rather spend their time working on the functionality of the product, rather than its security.

Similarly, having more than basic security is not much of a selling point for internet-enabled devices. Many consumers care more about how the product works and the various different features it has, compared to evaluating the security functions of it. As well as this, having strong security features could run the risk of limiting the usability of the device. For example, having to enter a password upon every use could annoy the user, making it less desirable. From a business perspective, since security is not driving sales, it makes more sense to focus on the functionality of the device. Consequently, security is more of an afterthought rather than a need.

Furthermore, some developers believe that basic security (i.e. default credentials) is enough for the product - instead, shifting the responsibility of security onto the consumer. While I believe that it is correct that consumers should take action to increase their knowledge about attacks and to protect themselves, developers must also ensure that they have done as much as they can to secure their devices. For instance, having default credentials is a form of security, but it is too simple, and not every user can be expected to know that they must change their login details. Instead, this can be improved upon by supplying users with unique tokens to access their admin portal. Shifting responsibility onto consumers should not excuse developers from applying more effort to secure the device.

One of the areas I believe that security could be improved in is the implementation of more protective laws and legislations. There is currently not much regulation regarding the security of Internet of Things devices, with the exception of a new law, proposed by the UK government in April 2021. This particular law ensures that "virtually all smart devices meet new requirements:

- Customers must be informed at the point of sale the duration of time for which a smart device will receive security software updates
- A ban on manufacturers using universal default passwords, such as 'password' or 'admin', that are often preset in a device's factory settings and are easily guessable
- Manufacturers will be required to provide a public point of contact to make it simpler for anyone to report a vulnerability." (Department for Digital, Culture, Media & Sport 2021)

Since "only 12 per cent of organisations review the cyber security risks coming from their immediate suppliers" (Department for Digital, Culture, Media & Sport, 2022), this law was eventually implemented in order to provide regulation for the manufacturers of IoT devices. This is a big step forward as it eliminates the default credentials issue and makes it easier for vulnerabilities to be reported, which can then be patched.

2.5 Consequences of Vulnerable Devices

While all vulnerable devices have the potential to cause damage, some devices tend to have more severe consequences than others. This is certainly true for internet-enabled machines such as vehicles, smart fire alarms, heart rate monitors, and other medical sensors. If these were to become compromised, private information could be stolen, certain protective settings could be changed, or even worse; lives could be lost!

Weaknesses in these devices lend themselves easily to cyber-attacks. “A cyber-attack is any attempt to gain unauthorized access to a computer, computing system or computer network with the intent to cause damage” (Pratt, 2022).

2.5.1 Medtronic Pacemaker Vulnerability

At an annual black hat event in Las Vegas, hosted in 2018, security researchers, Billy Rios and Jonathon Butts demonstrated a vulnerability in a Medtronic pacemaker that could leave thousands of patients at risk. The vulnerability would allow attackers to control the rate at which electronic impulses are sent to the heart, essentially altering the way the heart beats!

The researchers discovered that the radio-frequency protocol, called Conexus, lacks any form of authentication, meaning that, if they were within a 20 feet radius of the pacemaker, an attacker could simply connect and “inject, replay, modify, and intercept the telemetry data.” (Tung, 2019). The Food and Drug Administration (FDA) issued a warning against the use of these pacemakers, and while no lives were lost, the consequences of this vulnerability remain strikingly severe.

2.5.2 Remote Jeep Cherokee Hack

In July 2015, a team of researchers were able to expose weaknesses in a Jeep Cherokee vehicle, allowing them to remotely take control of various modules in the car. The researchers, Charlie Miller and Chris Valasek, were able to send messages to the vehicle’s Control Area Network (CAN) to control functions of the car, including altering the speed, the steering wheel, and even the emergency brake. This prompted them to demonstrate these vulnerabilities by hacking one of their own Jeep vehicles, driving onto a highway, and performing an emergency stop at speed (Brewster, 2016)!

The pair of researchers reported the bugs to Fiat Chrysler, but it just shows the ease at which even vehicles can be taken over. While Miller and Valasek rightfully didn’t share these vulnerabilities publicly, it is frightening to think of what would happen if they ended up in the wrong hands.

2.6 Mitigation of Security Issues

As mentioned previously, issues in security can lead to distributed denial-of-service attacks, private information being intercepted and stolen, or even a device being controlled by a black hat hacker! For these reasons, both the users and manufacturers should follow certain best practices to ensure that their device is not compromised.

2.6.1 2-Factor Authentication

One of the easiest ways for users to keep their device secure is by using 2-Factor Authentication where possible. This is a system that not only requires the user’s account

details to login, but also asks for the user to input a special code sent to another linked application, which could include email or text messages. If a remote attacker manages to brute-force their way into a user's account, it is unlikely that they will also have the user's email or phone number to hand, making this method very effective.

2.6.2 Enabling secure communication protocols

A fundamental method of improving IoT privacy is through the use of secure communication protocols. Manufacturers can employ the use of Transport Layer Security (TLS) in order to encrypt the data sent back and forth from the device to the server. Once the data is sent and encrypted, it is signed with a Message Authentication Code (MAC). Through this code, the server is then able to verify that the data has not been tampered with. If the data has been tampered with, the server knows not to trust the message (Das & Samdaria, 2014).

Many browsers are now phasing out the use of HTTP in favour of HTTPS. This is because HTTPS offers the same services but has more advanced security protocols in place to keep data safe.

2.6.3 Activating a next-generation firewall

A firewall is like a security guard; it controls what enters a network. The majority of routers have an active firewall, but these tend to be limited in their features. Next-generation firewalls are part of the third generation of firewalls and offer a range of advanced features such as malware protection; intrusion protection systems (IPS); content filtering; virtual private networks (VPN) and more. As a means to filter out attackers, malware, and viruses, firewalls are essential to users (Goodreau | n.d.).

2.6.4 Rigorous penetration testing

Before an Internet-enabled device is released for sale, the manufacturers should ensure that it has passed many in-depth penetration tests. These are done by a team of cyber security analysts to find any possible vulnerabilities in a device in order to assess where they lead and the consequences that may ensue by exploiting them. The tester then informs the 'Blue' team of any issues that they caught so that they can be patched whilst still in production. A thorough test will lead to the elimination of any flaws so that they are not sent out to customers.

2.6.5 Create a separate network for IoT devices

A way for users to keep themselves, their personal data and their network secure is by creating a new network specifically for their Internet of Things devices. This method of security ensures that if an attacker is successful in compromising a device, the only other path they can pivot to is to attack other IoT devices, rather than personal devices. It is uncommon to enter sensitive information in a smart plug or a smart lightbulb, so if these are compromised, their specific network can be shut down as a last resort, without any additional harm being done to the main network and the equipment connected to it.

2.6.6 Use of a VPN

A VPN is a virtual private network that provides the user with a form of online anonymity by disguising their IP address. These attributes help mitigate vulnerabilities as a user's network traffic is sent to a server provided by the VPN, where the data is encrypted in the process. This will ensure that an attacker will not be able to view any private information this way and hides the user's network traffic.

2.6.7 Other common remediation methods

While the above-mentioned practices can help lower the risk of security issues, there are still many other techniques that users and manufacturers can use to keep their devices and networks secure:

- Ensure that all passwords are kept strong and difficult to guess
- Ensure vulnerabilities are patched as soon as possible
- Correctly configure the router
- Use encryption wherever possible
- Reduce the attack surface by closing unnecessary ports

2.6.8 Reflection

I believe that, for manufacturers of an IoT device, an extremely attentive and thorough penetration test would be the best course of action before selling their products. This is because any vulnerabilities that exist can be hand tested in great detail, to explore the different avenues that an exploit would lead to. Whilst still in production, any errors can be resolved without risk of damage or data loss from an attacker.

On the other hand, for the users of an IoT device, I would recommend strengthening the rules of their firewall as a method of protection for their network. A strong firewall is one that not only filters malicious software from being downloaded, but also prevents malicious attackers or unwanted data connections from reaching the user's device. The majority of the latest computers, laptops and routers come with a dedicated firewall settings menu, where certain rules and adjustments can be made, allowing for more personalized and specific control. In other words, the firewall can be tailored to block specific objects, or allow them based on the rules the user has set. The user could easily make changes according to their preferences and improve the security of their network.

2.7 Theoretical Background

2.7.1 Ethical Hacking

Thus far, the focus has been on black hat hacking and its consequences. An alternative to this, however, is white hat hacking, also known as ethical hacking. Ethical hackers are those who employ the same skillset and exposure of vulnerabilities as black hat hackers, except they operate within the law and follow strict rules and guidelines.

These types of hackers offer their services to organisations in order to assess their products and report any possible security flaws and weaknesses that they find. The term 'ethical hacking' is interchangeable with penetration testing. Organisations rely on ethical hackers to provide a full risk assessment of their business and products to keep them protected.

2.7.2 Protocols

Protocols are a set of rules that determines how data should be formatted, transmitted, and delivered between devices within a network. They can be used in conjunction with ports, where each of the system ports is assigned a specific protocol or service. If a manufacturer of IoT devices leaves unnecessary ports open, it can lead to a larger attack surface, allowing for more possible vulnerabilities to be exposed by a malicious hacker. There are, however,

different uses and types of protocols, including transport protocols, security protocols, and many others. Keeping in scope of this report, two common security protocols include:

- **SSL/TLS** - Secure Sockets Layer (or Transport Layer Security, as it is now known) is a protocol that ensures safety and privacy over the web via the use of encryption and certificates. When a visitor enters a website, their browser looks for the SSL/TLS certificate and performs a 'handshake' to verify that it is real. If this succeeds, an encrypted link is created between the browser and server to safely transport data.

  SSL was deprecated and replaced by TLS in 1999 due to security issues, but since the naming convention is now commonplace, many people still refer to TLS certificates as SSL certificates (Das & Samdaria, 2014).
- **HTTPS** - Hypertext Transfer Protocol Secure is an advanced version of the HTTP protocol that uses SSL/TLS certificates to ensure that data is encrypted when it is transported.

2.7.3 Firmware

Although firmware and software are very similar, there are some important distinctions between the two. Firmware is a piece of software that provides instructions to the hardware and ensures that it works as intended. The firmware is usually embedded in the flash ROM (Read-Only Memory) of a device so that it is not erased when the device is turned off and is stored permanently (Teja, 2021).

In the majority of personal computers/laptops, BIOS is the firmware that acts as an interface for the hardware. Similar to how an Operating System is used to control downloaded software applications, the BIOS is used to control the device hardware and is responsible for booting up the computer.

In the context of Internet of Things devices, firmware updates are either released on a regular basis or not at all, depending on the device. These updates are used to fix bugs and vulnerabilities, improve security to prevent attacks, or to introduce new features. A device such as a smart plug may not require many firmware updates if it is secure enough, whereas a smart watch may have constant updates in order to keep up to date with new functions.

Chapter 3. Internet of Things Attacks & Tools

3.1 Methodology

Before a vulnerability analysis of IoT devices can be carried out, an initial plan must be devised based on how the test will be performed and what the expected result of the test is. This is called a penetration testing methodology. A methodology is chosen based on the type of device that is being tested, as well as the different techniques that will be used.

There are many methodologies available, outlining the different phases of a test. While there are some exceptions, most follow a similar structure:

Phase	Description
Reconnaissance	This phase attempts to gain information (that is usually found online) about the device system, and the underlying weaknesses and vulnerabilities it may have
Enumeration	Similar to above, but gains information about the device's attack surface through network scans and maps, which could reveal the Operating System, software version, and open ports on the machine
Attack	Using the information gathered, the attack takes place - exploiting vulnerabilities and attempting to gain access
Privilege Escalation	Once access is gained into the device, a further attack is carried out to escalate from a low-privilege user to a high-privilege/root user
Maintain Access	When root access is gained, the attacker will want to install a persistent backdoor to the device, allowing for remote access at any time.

Table 1: Common methodology structures/phases

Another popular ethical hacking framework is the OWASP Testing Guide V3, which outlines five phases of testing that can be performed in various phases of the IoT Software Development Life Cycle:

- Information Gathering
- Configuration Management Testing
- Authentication Testing
- Session Management Testing
- Authorization Testing (Kang et al., 2015)

3.2 Common Attacks

Experienced black hat hackers will have an arsenal of tools and attacks ready to be used to compromise an Internet of Things device. Given that a penetration tester must think like an attacker, it makes sense that they would also use the same attacks and skillset in order to uncover vulnerabilities in the device that they are assessing.

The type of weakness that is found dictates the most suitable attack, as using an attack that can't expose the weakness further would only waste time without advancing further towards root access. Therefore it stands to reason that some research should be applied before deploying an attack.

Commonly used attacks that Internet of Things devices are vulnerable to include:

3.2.1 Man-in-the-Middle Attacks

As mentioned previously, Man-in-the-Middle attacks are a form of eavesdropping that are critical in providing an attacker with private information. It occurs when there are no valid security certificates to encrypt the communication between devices, which means that the attacker can trick the victim and the server into believing that they are communicating with each other, when in actuality, the attacker sits in between and intercepts the data being sent. The data that is captured is typically confidential, such as login credentials, credit card details, or even browser cookies, which will give access to a user session.

Man-in-the-Middle attacks take place when there are either weak or no security frameworks in place to encrypt the communication between devices. These attacks would commonly take place in internet-enabled devices that use the HTTP protocol rather than the more secure version, HTTPS. Because HTTP does not use TLS certificates to authenticate the device it is communicating with, it leaves the user exposed to this type of attack.

One real life example of a Man-in-the-Middle attack occurred in 2014, when it was reported that the technology company, Lenovo, preinstalled adware to their computers and laptops. The adware, known as Superfish Visual Discovery, would generate its own root certificate and replace all other SSL certificates displayed on HTTPS websites. In doing so, the company could view and scan the user's online activity and use that against them, as well as inject their own advertisements to the browser (Paul, 2015).

3.2.2 Brute-force Attacks

When default or weak credentials are used in internet-enabled devices, they lend themselves very easily to brute-force attacks. This is an attack where a wordlist of potential usernames and/or passwords is input to a tool, which automatically runs through the list, attempting each one in turn. The more simple the credentials, the easier it is to gain access, which is why it is very important to ensure that all passwords are complex with multiple special characters.

One of the largest wordlists that exists is *rockyou.txt*, which originated from a company that stored all of their user's passwords in plaintext within an unencrypted database. When the database was hacked via an SQL Injection attack, over 14 million usernames and passwords were leaked, which are now used as a basis to attack other user accounts to this day.

3.2.3 Firmware Attacks

As the name suggests, firmware attacks give access to the underlying processes of a device, allowing the attacker to control how the hardware works. These types of attacks are considered 'stealthier' than other attacks due to their unique method of opening a 'backdoor',

which allows them to go unnoticed and become difficult to detect. A backdoor is an "undocumented way of gaining access" (National Institute of Standards and Technology | n.d.) so its use opens up an alarming security risk.

Since the firmware of a device maintains high privileges and is able to bypass common security checks (because security usually works on the software level), such as anti-malware, firmware is a good attack surface for malicious hackers. Many firmware vulnerabilities emerge from a lack of updates to a device, which may leave it unprotected and out of date. Therefore it is always advisable to update devices to the latest official firmware available.

Some specific types of attacks involve the attacker downloading or creating a malicious version of firmware, that is then physically implanted onto a device by delivering it through a USB or SD card. From there, the device can be compromised before it is even able to boot up, reiterating just how dangerous firmware attacks can be (Eclypsium, 2019).

3.2.4 Malware Attacks

Malware attacks remain one of the most common grievances to users of internet connected devices, although they are a very simple attack. It involves an attacker creating a malicious piece of software, hoping to damage or control a person's device/machine. There can be various methods for malware to infect a device, including a user being sent a link by an attacker, posing as someone they trust. When opened, the link would download the malware directly on to their device and be able to attack it from within.

As well as this, similar to firmware attacks, another method that attackers use to spread malware is by packing their malicious software on to a USB or SD card and plugging it in to devices that are left unsupervised. From there, any personal information on the device can be stolen and the device itself can even be controlled.

Many attackers attempt to increase the stakes by encrypting an infected device's files and then contacting the device's owner. The attacker demands a large payment in exchange for the encryption key to their device, essentially holding the owner's personal files to ransom.

3.2.5 Distributed Denial-of-Service Attacks/Botnets

A serious threat to the safety of Internet of Things devices is a Distributed Denial of Service (DDoS) attack. The reason this is so prevalent is because its aim is to shut down a specific website or online resource by sending an overwhelming amount of traffic to the web server, preventing it from handling the amount of requests coming through.

In order to send such a large amount of requests, a network of bots is formed, called a botnet. These are internet connected devices that have been infected by malware and are now in the control of the attacker. When the attacker gives the command, the botnet collectively sends continuous requests and overloads the web server.

A Distributed Denial-of-Service attack was performed in one of the largest cyber incidents of the past decade; the Mirai Botnet attack. In this specific case, the attacker controlled a large botnet, by infecting hundreds of devices with malware, which he used to attack a French hosting firm called OVH. The attacker then leaked the botnet's source code online as a way to keep the attention off of himself, but this action gave way for other cyber criminals to use the botnet for themselves. After a few days, the botnet was used to perform a DDoS attack on a large domain registration service named Dyn, shutting it down. Since Dyn provided the

DNS information for a majority of online websites, many of them were temporarily taken down as well, including large websites such as Twitter and Netflix (Cloudflare, 2016).

3.3 Preventing Attacks

The difference between a security issue and an attack is that a security issue, or vulnerability, is a weakness in a user's network that can be exploited. An attack, however, is an active threat from a malicious actor to cause damage to your system.

To prevent a black hat hacker from attacking an IoT device, certain precautions can be taken beforehand in order to protect your network. These include:

3.3.1 Virtual Private Networks

As mentioned previously, Virtual Private Networks (VPNs) are used as a method to hide online activity, internet traffic, and IP addresses from unwanted third parties. This is done by encrypting a web request and sending it through a data tunnel to the VPN provider's servers. The request is then sent through another tunnel where it is decrypted upon reaching its destination.

Any third party, such as the Internet Service Provider or malicious hackers will no longer be able to view an individual's internet traffic and IP address, and as a result, they will be less likely to become a victim of a Man-in-the-Middle attack.

3.3.2 Strong Wi-Fi Encryption

Having a strong encryption method on Wi-Fi reduces the chances of an attacker hijacking an individual's router, as well as their internet traffic. Of the four standard Wi-Fi encryption protocols, WPA3 is the most recently developed and secure. However, many wireless access points do not support WPA3 as of yet, making WPA2 the best choice. The two protocols left unmentioned, WEP and WPA, are mostly outdated by now and are no longer commonly used as a strong form of encryption as they can expose a network to vulnerabilities (Zaidan, 2021).

3.3.3 Anti-Malware

One very common type of cyber-attack is executed by infecting a device with malicious software, i.e. malware. This is spread through downloadable links, files, and software that is designed to look authentic, but actually contains the malware within. To protect against this, anti-malware is used to detect and eliminate the malware before it can cause any harm to the device. One way this is achieved is through 'sandboxing' the suspicious file and opening it within a virtual environment - if the file attempts to access other system files or programs, it is rejected and removed (Rosencrance, 2021).

Thousands of people fall victim to malware attacks every day. This can be prevented by being skeptical when spotting unknown and untrustworthy links that are usually included in spam emails.

3.3.4 User Access Control

User Access Control (UAC) is a fundamental security principle on Windows machines, used predominantly in corporations. It works by encouraging the use of a high-privilege administrator account, along with several other low-privilege user accounts. The administrator account has the most power and decides what changes can be made to the system and which users & applications can use administrator privileges.

When a user attempts to download software or make any changes to the machine, a prompt is displayed, asking the administrator for permission. If the administrator approves of the change, they can enter their password and the change will be made. However, if they do not approve, they can discard the changes being made.

This control feature is paramount in many businesses and corporations, as if one user unknowingly downloads malware to their device, it can have an impact on all other machines in the network. User Access Control prevents this situation from happening as the download will have to be approved before it can be installed on the machine.

3.3.5 Reflection

For the average user, I believe that using an online VPN service would be one of the best safeguards to protect them and their network. This is because a VPN disguises their IP address online, encrypts their data, as well as preventing their traffic from being tracked. Many commercial companies provide VPN services for a price, including ExpressVPN or NordVPN. More advanced users could be capable of configuring their own VPN server using Cloud Virtual Private Servers, including Amazon Web Services. With all of these attributes in mind, it would be difficult for an attacker to steal or intercept their data or to execute attacks on their device, using their IP address.

Alternatively, a good anti-malware system, such as Norton Security, would help keep a device secure by sandboxing malicious software and preventing it from causing damage to the actual device. As well as this, by following key internet safety rules (i.e. not trusting unofficial links, creating safe and complex password, keeping software up to date etc.) a user is less likely to fall victim to phishing attempts, keyloggers and other malicious software crafted by an attacker.

3.4 Commonly used Tools

In order to gather information about an Internet of Things device and its attack surface, as well as to execute attacks, certain tools can be used to increase the ease of penetration testing.

3.4.1 Kali Linux

Kali Linux is a Debian-based Linux distribution that comes pre-packaged with over 600 industry-standard tools, such as network scanners, hash crackers, packet sniffers, and more. It is an environment used by cyber security professionals and, as an Operating System, is highly customizable (Kali, 2020).

In order to avoid replacing the main Operating System of a machine, many people install a Kali Linux image to VirtualBox. This is another tool used as a virtual machine to create an isolated environment used for the sole purpose of hacking.

3.4.2 Nmap

Nmap (Network Mapper) is used as a tool to scan networks and map out the ‘landscape’ of the device in question. It sends IP packets that return a great deal of information, including the active hosts within a network, the ports open on a device, the Operating Systems, and servers in use. Nmap also contains the use of a wide range of scripts, which allow it to automate certain tasks.

Within a penetration testing methodology, network mapping is commonly one of the primary tasks completed, as part of the information gathering/enumeration phase.

3.4.3 Wireshark

Wireshark is a widely used packet sniffer and analyser. It can be used to monitor packets that travel across a network, as well as capture them and inspect them in further detail. With these, a hacker is able to view communications between devices, which may include unencrypted private details, such as credentials or bank account details. Wireshark can be used as part of the enumeration phase or to aid during the attack phase of the methodology.

3.4.4 Burp Suite

Burp Suite is an extensive tool with many uses. One of its primary modules is used to capture and manipulate web requests before they are sent to a web server, allowing the user to gain access to areas of a website kept off limits. This can be done by modifying cookies or web responses.

Burp Suite can also be used as a brute-force tool; by supplying a wordlist, it will make continuous requests until it finds valid credentials. Other modules within Burp Suite include Repeater, Decoder, Comparer, Sequencer, and Extender.

3.4.5 John the Ripper

John the Ripper is a password-cracking tool, designed to reveal strong, hashed passwords. A hashed password differs from an encrypted password as encryption can be decrypted, whereas a hash is irreversible. John the Ripper works by requesting a wordlist, hashing each word using a specified hashing algorithm, and comparing the result to the hash that is being cracked (Lubeck | n.d.).

John the Ripper is commonly used in the attack and privilege escalation phases of many penetration testing methodologies.

Chapter 4. Device under a Scope

4.1 Wyze Cam V2

The Internet of Things device that this thesis will be presenting is a Wyze Cam V2. This is a security camera that provides continuous video recording to aid in monitoring and added protection of a household. The way the Wyze Cam V2 differs from other regular security cameras is its capability of connecting to a network in order to provide the user with real-time footage of what the camera is recording. Through the use of a mobile phone application, as well as a web browser, the user can view a live stream of the footage, as well as play back footage recorded in the past.

4.2 Physical Components

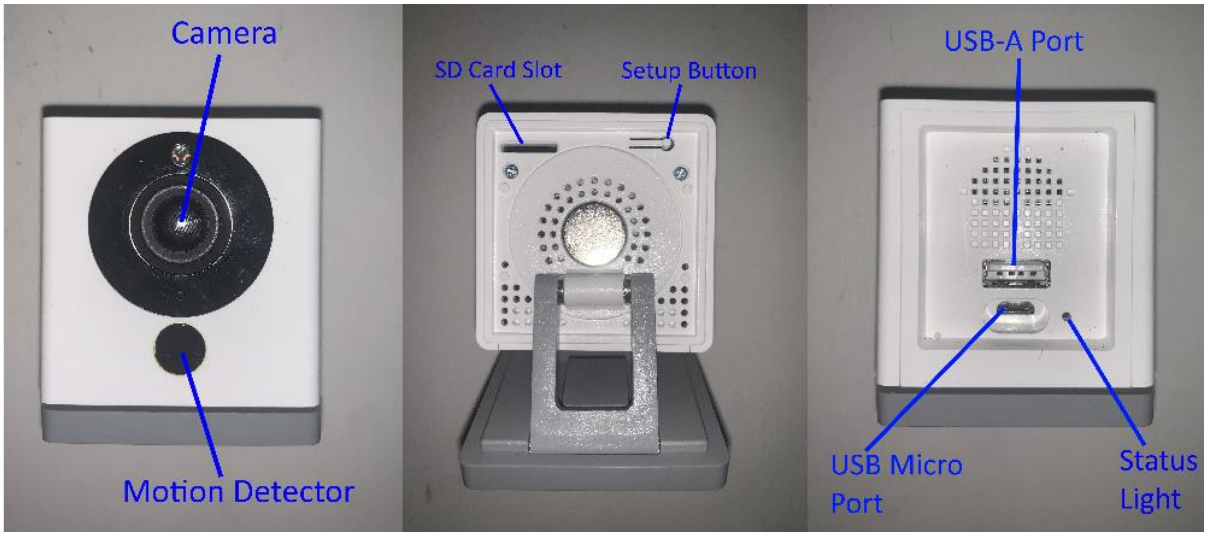

Figure 1: Physical Components of Wyze Cam V2

The model of the camera is WYZEC2. The Wyze Cam V2 offers 8x digital zoom, two-way audio, night vision, and also runs on a Linux 3.10.14 Operating System. As shown in figure 1, the camera comes with a motion detector, two USB ports, and an SD card slot. If not protected properly, these carry the risk of an attacker inserting malicious data directly into the camera, which will be explored in more detail in the next chapter.

Chapter 5. Threat Modelling

At this point, a threat evaluation must take place in order to predict the way the device could be attacked. Using common vulnerabilities and exposures found in other similar internet-enabled security cameras, it is possible to predict certain threats specifically for the Wyze Cam V2.

5.1 Identifying Assets

This section shows the various hardware and software components that make up the Wyze Cam V2, as well as the ports and protocols used.

Asset	Description
Wyze Cam V2	The model of the camera is WYZEC2 . It allows for continuous video recording and includes functions, such as two-way audio, motion detection, night vision, and 8x digital zoom. Technologies supported: TCP/IP, HTTP, HTTPS, Wi-Fi, UDP, WPA/WPA2
Camera	Provides a real-time live stream of captured footage, which is uploaded to a cloud server.
Mobile Application/Web Application	The mobile application is compatible with iOS 14+ and Android 7.0+. It can be used to access live recordings and events, adjust settings, and update the camera. The web application is accessed through a web browser with HTTPS and a TLS 1.2 certificate.
Wireless Communication	The camera uses Wi-Fi to upload recordings to the cloud server. It supports Wi-Fi 802.11 b/g/n 2.4 GHz
Cloud Services	Cloud services are used to store uploaded recordings. Port 8443 is used to connect to the Cloud API and events are uploaded using port 443.
Firmware	The current Wyze Cam V2 firmware (at the time of writing) is 4.9.9.1433. Hardware and system features can be modified through the firmware. A download file to roll back firmware is provided directly on the Wyze website.

Table 2: Identifies the assets of the Wyze Cam V2

5.2 Attack Vectors/Entry Points

This section shows the attack surfaces/entry points of the Wyze Cam V2, as well as possible methods of attack

Attack Vector/Entry Point	Method
Connected mobile application/web browser	Both the mobile and web applications require a Wyze account for authentication. It could be possible for an attacker to use typical default credentials to attempt access into the admin area of the camera. Many possible credentials can be found simply through the use of Google.
Wi-Fi	If the network that the camera is connected to is insecure and unencrypted, it can be very simple to capture and inspect the packets it sends using Wireshark. From here, the information found in the packets could help leverage a path into the camera.
Firmware	Firmware is often overlooked when it comes to security, making it a great attack vector. This type of attack could be used to modify the firmware of the device, opening it up to unauthorized access.

Table 3: Possible attack surfaces for Wyze Cam V2

5.3 Identifying Threats

This section shows the different possible threats that exist and the entry point that they target, as well as possible mitigations and countermeasures towards them.

Threat Description	Unauthorised access to user accounts
Entry point targeted	Mobile/Web Application
Attack Method	The attacker could apply brute-forcing techniques, using tools such as Burp Suite or Hydra, to gain unauthorised access to user accounts
Mitigations	Make passwords secure by setting strong requirements; use two-factor authentication; lock out the user account after multiple failed attempts

Table 4: Threat #1 - Unauthorised access to user accounts

Threat Description	Access to critical files/recordings
Entry point targeted	Mobile/Web Application
Attack Method	The attacker could access files storing valuable information, such as recordings and events
Mitigations	Set separate passwords to view valuable information; use two-factor authentication; delete old and unused files when not needed, with the user's permission

Table 5: Threat #2 - Access to critical files/recordings

Threat Description	Packet sniffing for user data
Entry point targeted	Wi-Fi
Attack Method	The attacker could use a packet sniffer, such as Wireshark, in order to intercept packets and read data

Mitigations	Use a VPN; encrypt wireless access points using WPA2 protocol, which Wyze Cam V2 supports; use HTTPS/TLS certificates when web browsing

Table 6: Threat #3 - Packet sniffing for user data

Threat Description	Creating/downloading malicious firmware
Entry point targeted	Firmware
Attack Method	The attacker could download or create malicious firmware, which can then be flashed onto the Wyze Cam V2, using a microSD card
Mitigations	Restrict physical access to the camera; download official and up-to-date firmware directly from the provider; require authentication when making significant changes

Table 7: Threat #4 - Creating/downloading malicious firmware

Chapter 6. Proof of Concept Exploit

This chapter demonstrates a proof of concept exploit for one of the threats outlined in section 5.3. The method of exploit takes inspiration from the works of Jason Ford (2020), George Hilliard (2020) and YouTube security researcher, Stacksmashing (2020). As such, the method of exploit is a firmware attack.

6.1 Finding Common Vulnerabilities and Exposures

Before any exploit of significance can take place, Common Vulnerabilities and Exposures (CVEs) must be found to identify known weaknesses in devices. These are published online in order to spread awareness of their existence and help improve security. Using the National Vulnerability Database provided by the US Department of Commerce, CVE-2019-9564 can be found. This CVE confirms the existence of "a vulnerability in the authentication logic [of Wyze Cam V2, which] allows an attacker to bypass login and control the devices". This vulnerability affects Wyze Cam V2 versions prior to 4.9.8.1002, which was released in March 2022. With a severity rating of 9.8 and a low complexity, this attack seems promising (National Institute of Standards and Technology, 2022).

Another CVE found is CVE-2019-12266, which is a "Stack-based Buffer Overflow vulnerability" that "allows an attacker to run arbitrary code on the affected device". As before, this CVE affects Wyze Cam V2 versions prior to 4.9.8.1002 and also has a severity rating of 9.8 (National Institute of Standards and Technology, 2022).

6.2 Method

Since the firmware of the device in question was set to the latest version, in order to begin the attack, its firmware had to be downgraded to version 4.9.5.36, released in November 2019. This was done because, as implied by the CVEs found, firmware versions before 4.9.8.1002 were typically weaker, and since not many other firmware downloads were supplied directly by Wyze, this version was the most convenient.

Once the binary file of the firmware was downloaded, binwalk was used to translate information about the file. Binwalk is a tool used to find files, code, and hidden text within binary files and is very useful for firmware images.

Using the command *'**binwalk -t**'*, the information in figure 2, below, was displayed, from which we can gather that a Linux Operating System is used with a 3.10.14 kernel image. There are also two Squashfs filesystems in use, as well as one JFFS2 filesystem.

```
┌──(hamza㉿kali)-[~/FYP]
└─$ binwalk -t demo_v2_4.9.5.36.bin

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             uImage header, header size: 64 bytes, header CRC: 0xCDF0042E,
                              created: 2019-11-15 07:00:02, image size: 11075584 bytes, Data
                              Address: 0x0, Entry Point: 0x0, data CRC: 0x869272CE, OS:
                              Linux, CPU: MIPS, image type: Firmware Image, compression
                              type: none, image name: "jz_fw"
64            0x40            uImage header, header size: 64 bytes, header CRC: 0xD3B9E871,
                              created: 2019-02-14 03:00:10, image size: 1859813 bytes, Data
                              Address: 0x80010000, Entry Point: 0x80400630, data CRC:
                              0xE3786CEF, OS: Linux, CPU: MIPS, image type: OS Kernel Image,
                              compression type: lzma, image name: "Linux-3.10.14"
128           0x80            LZMA compressed data, properties: 0x5D, dictionary size:
                              67108864 bytes, uncompressed size: -1 bytes
2097216       0x200040        Squashfs filesystem, little endian, version 4.0,
                              compression:xz, size: 3353204 bytes, 407 inodes, blocksize:
                              131072 bytes, created: 2019-05-21 17:22:45
5570624       0x550040        Squashfs filesystem, little endian, version 4.0,
                              compression:xz, size: 572594 bytes, 12 inodes, blocksize:
                              131072 bytes, created: 2018-08-13 04:50:58
6225984       0x5F0040        JFFS2 filesystem, little endian
```

Figure 2: Properties of the 4.9.5.36 firmware image

Squashfs is a read-only file system that is used to compress files and directories so that space can be saved. JFFS2 (Journaling Flash File System 2) is a “log-structured file system designed for use on flash devices in embedded systems”. It is used to place filesystems on flash drives (Woodhouse, 2003).

6.2.1 Extracting the filesystems

In order to enter and modify the three filesystems found, they must be extracted from the binary file. This could be done using the binwalk tool, but since the filesystems must be extracted and then later repacked, another method must be used.

After researching online, a GitHub repository containing a “wyze_extractor” script was found (Nezza, 2020). The extractor script was used to remove the directories of the two squashfs filesystems and the JFFS2 filesystem (all of which could not be entered at this point) and rewrite them again separately.

Doing this allowed the filesystems to be accessible with their original directories so that they could be analysed.

6.2.2 Finding root password

Since the filesystems could now be accessed, finding the root password depended on entering the most useful one. The JFFS2 and the second squashfs filesystem contained many binary files used by the kernel, which were not helpful towards our goal. The squashfs1 filesystem, however, contained a root directory, so was the most promising in finding the root credentials:

```
┌──(hamza㉿kali)-[~/FYP/extracted/squashfs_1_out]
└─$ ls
backupa  backupk  configs  driver  lib     media  opt     proc  run   sys     thirdlib  usr
backupd  bin      dev      etc     linuxrc  mnt    params  root  sbin  system  tmp       var
```

Figure 3: Root directory found in squashfs1 filesystem

By entering the /etc directory, the file ‘shadow’ was world-readable, meaning that the password for the root user could be viewed as a hash.

Using the cracking tool, John the Ripper, the password was easily revealed with the rockyou.txt wordlist:

```
┌──(hamza㉿kali)-[~/FYP/extracted/squashfs_1_out/etc]
└─$ cat shadow
root:rJ0FHsG0ZbyZo:10933:0:99999:7:::

┌──(hamza㉿kali)-[~/FYP/extracted/squashfs_1_out/etc]
└─$ john --wordlist=/usr/share/wordlists/rockyou.txt shadow
Using default input encoding: UTF-8
Loaded 1 password hash (descrypt, traditional crypt(3) [DES 256/256 AVX2])
No password hashes left to crack (see FAQ)

┌──(hamza㉿kali)-[~/FYP/extracted/squashfs_1_out/etc]
└─$ john --show shadow
root:ismart12:10933:0:99999:7:::

1 password hash cracked, 0 left
```

Figure 4: John the Ripper tool used to crack the root password hash

From figure 4, it can be seen that the root password for the Wyze Cam V2 was *ismart12*. Considering that this is the account with the highest privileges and the most control, it is a very simple password, as proven by the fact that it was easily cracked by the wordlist.

6.2.3 Backdoor

Now that the root password for the camera has been found, it can be used to login and gain access to the highest privileges. Still within the /etc directory, a subdirectory can be found, called ‘init.d’, which holds a file containing the scripts run during the startup of the device. By reading the contents of this file (figure 5, below), it is possible to spot that the camera starts a telnet daemon during boot.

Although it is insecure and not based in encryption, telnet is a protocol used to connect to other computers/devices on the same network. It was replaced by SSH because of its vulnerabilities, which makes it perfect to exploit.

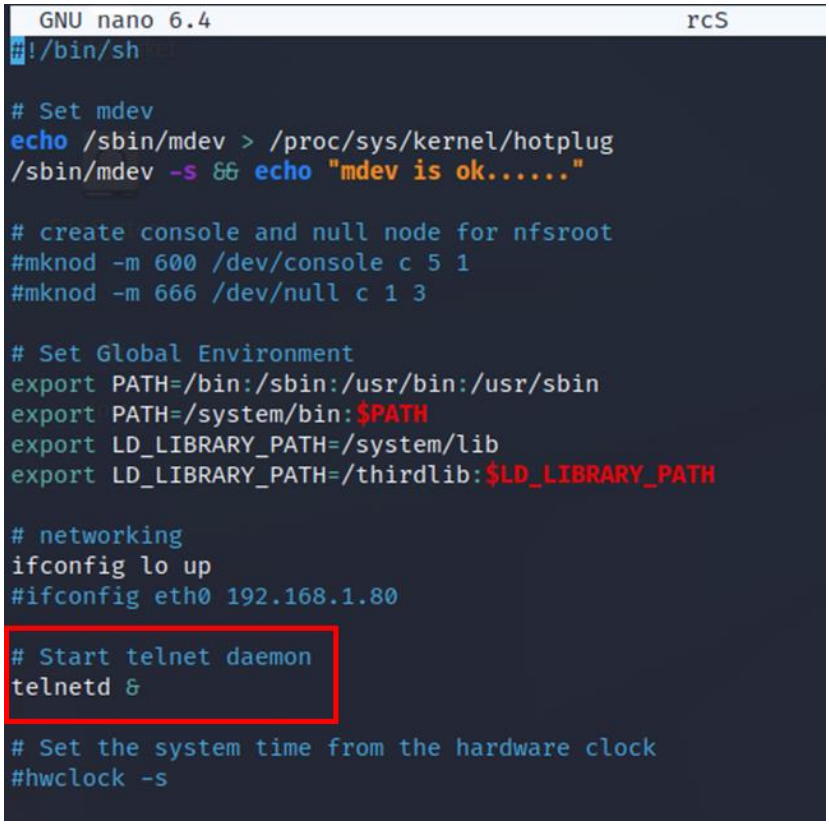

Figure 5: Contents of rcS, a file containing boot scripts for Wyze Cam V2

Upon searching closely for other instances of telnet being used, there is a process being run in a file called 'iCamera' that kills the telnet service. Potentially left as an oversight, it is clear that the developers did not want telnet to be used by their general customers, but rather than uninstalling or removing it from the start-up scripts, their solution was to terminate the process. In doing so, this leaves the camera vulnerable to the creation of a backdoor, simply by modifying the boot script to call BusyBox.

BusyBox is a tool that can perform the action of other Linux tools (Stacksmashing, 2020). For example, since the telnet process gets killed by iCamera when it is run, we can use BusyBox to run telnet instead. By replacing the original command in the boot script file (figure 5) with the new command '***busybox telnet***', telnet will start without being killed.

6.2.4 Creating Firmware Image

Because the squashfs1 filesystem has now been modified, it has to be repacked into a new firmware image for the effects to take place. In order for this to work, a new squashfs1 filesystem was created with the same properties as the old. Using the command '***mksquashfs squashfs_1_out/ squashfs_1_new -comp xz -b 131072***' , the compression and blocksize attributes were kept the same, creating a new squashfs1 filesystem with the modified boot script.

From here, a new binary containing the backdoor was created using the GitHub 'wyze_extractor' script's pack function.

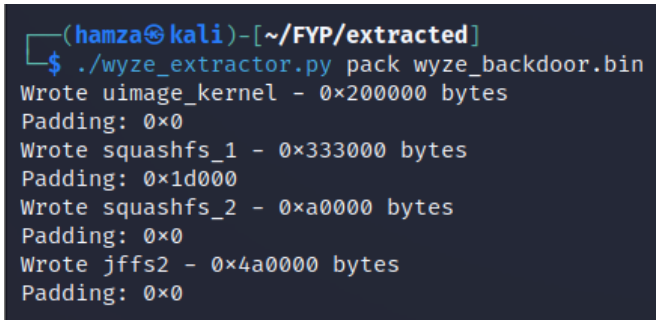

Figure 6: Creating the backdoor binary

As shown in figure 6 above, the binary has been filled with the two squashfs filesystems, along with the JFFS2 filesystem and a uimage kernel. The only file missing is the uimage header, which had to be generated independently.

Once again, the uimage header had to be created with the same properties as the file in the original firmware image (figure 2). With the command '***mkimage -A MIPS -O linux -T firmware -C none -a 0 -e 0 -n jz_fw -d wyze_backdoor.bin wyze_firmimage.bin***', the properties, such as the Operating System, CPU, and firmware name could be matched, creating a new firmware image as seen below.

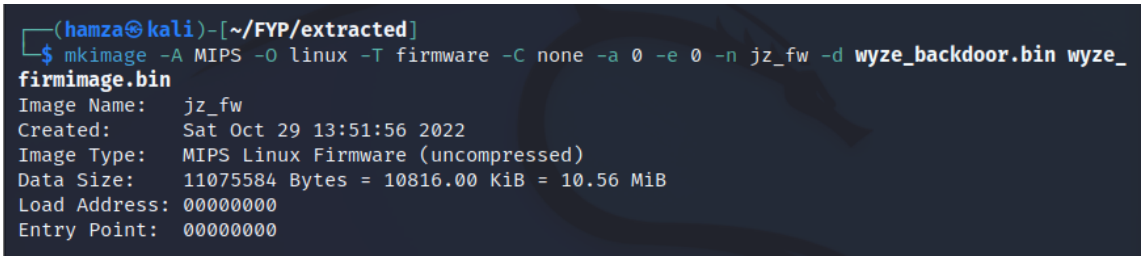

Figure 7: Creating the final firmware image

With the creation of a new, modified firmware image, the camera is almost compromised with root privileges. For the changes to be made and telnet to run, the new firmware image was copied to a microSD card and physically implanted directly to the Wyze Cam V2.

After the firmware had been successfully downloaded to the camera, if a port scan was attempted, the results show that a telnet service is active on port 23, due to the modifications made:

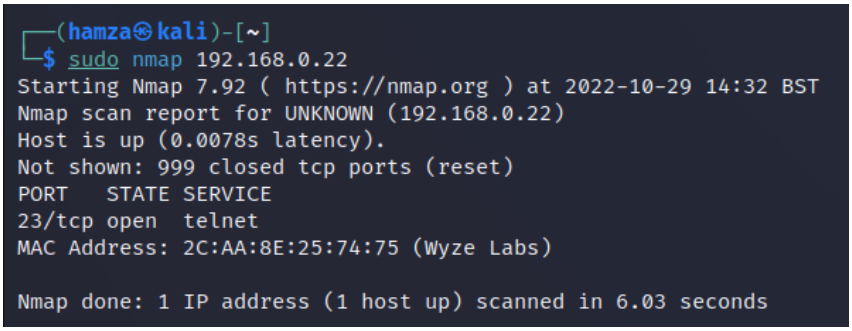

Figure 8: Output of Nmap scan, showing that the telnet service is now active

Since the service is now active, the command '***telnet***' followed by the camera's IP address can be entered. This will prompt for a username, which can be entered as 'root', as well as a password. Using the root password that was cracked previously, 'ismart12', access to the root user has been gained:

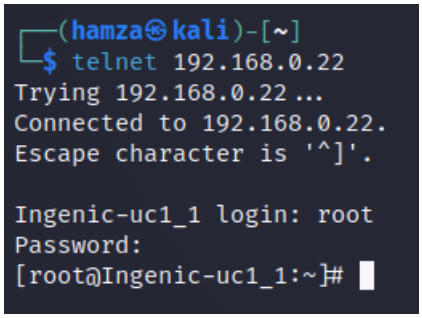

Figure 9: Logging in as root user via telnet

As shown in figure 9 above, the Wyze Cam V2 has now been hacked, with full root access and control. From here, confidential user data can be viewed, as well as modules that allow an attacker to interact directly with the camera. One such function is the two-way audio feature which allows an attacker to project their voice out of a user's device. As identified previously, the ability to take control of the Wyze Cam V2 shows that it is in desperate need of security upgrades.

Chapter 7. Results & Discussion

7.1 Results

To summarise, the Wyze Cam V2 was hacked by cracking the simple root password hash, which was world readable in the /etc/shadow file. A slight modification was then made to the camera's boot script to allow the telnet service to persist, and once the firmware image was repacked, it was ready to download to the device.

Holding root privileges, many private files and directories could now be entered via the command line. These include a USB storage directory, as well as an audio directory; both giving a potential attacker deeply sensitive information that could be harvested and used to gain access elsewhere, or potentially hold the data for ransom. Other files allow an attacker to access the live camera feed and even output their voice using the two-way audio feature.

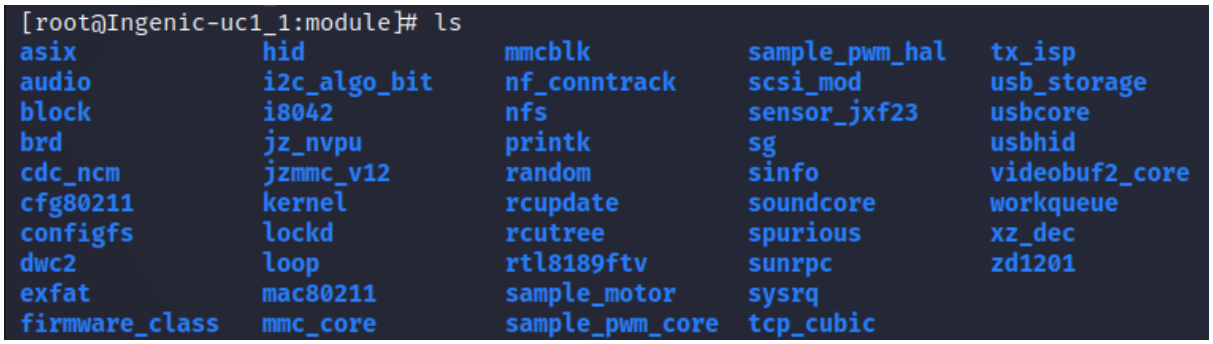

Figure 10: Private modules that can be accessed as the root user

Overall, the ease of hacking the Wyze camera is very straightforward and effortless. After some research, an experienced black hat hacker would be able to compromise this camera within minutes, which represents a vulnerability in itself! The security measures and the modules left behind after development definitely need to be reviewed and improved in the Wyze Cam V2.

One method that the developers can use to enhance security includes using authentication methods to verify a firmware upgrade (or any other significant structural change to the device). Similar to how Apple devices ask users to enter their password before a software update, the Wyze Cam V2 can implement the same validation method to ensure that the actual device owner is requesting the change, rather than an unwanted third party.

As well as this, by leaving a tool within the backend of the device, past its development uses, the Wyze Cam V2 was able to be exploited. Rather than uninstall the service, telnet's process was being killed upon startup, and as demonstrated, it was able to be modified to persist. Developers who leave modules and services active past development give attackers a larger attack surface to exploit the device, and as a result, should be more careful.

Furthermore, users of the device should be more concerned for their own security as well. By restricting physical access to the camera (such as keeping it in a cage if left outside, for example) and by not trusting unknown USBs or SD cards, the integrity of the device can be preserved.

7.2 Legal, Social, Ethical and Professional Issues

Cybercrime is on the rise. This report is not intended to provide black hat hackers with a means to hack into devices unlawfully, but instead to spread awareness on the general lack of security in Internet of Things devices and how these must be improved. This report was written from an educational point of view and should not be used to launch attacks on others' devices.

The difference between ethical hackers and black hat hackers is the side of the law that they operate on. Ethical hackers must work within the law and ensure that all devices or applications tested on are their own, or they have received express, written permission to analyse security on behalf of a company. Laws and legislations can be different depending on the country the ethical hacker is based in, and as a result, must be observed with care to stay within boundaries.

Black hat hackers operate outside the boundaries of the law. This means they maliciously hack into machines or networks that are not their own, or they gather private information that can be used to harm. Many black hat hackers are motivated by financial gain, causing disruption, or for recognition. Some punishments for cybercrimes can range from a fine to prison sentencing, depending on the damage caused.

All devices, tools, and accounts used as part of this project are owned, first and foremost, by the author. The Local Area Network that the device was part of belonged to the author. No hacking was conducted on the devices of others, as doing so would break the law. Furthermore, no attacks that could affect other users or the Wyze servers and internal structures were conducted.

In terms of morality, the Wyze Cam V2 used throughout the report was not handed to other parties or resold, so as not to deceive or capture any private information of others, since the device was hacked with the creation of a backdoor. Instead, the device has been safely disposed of.

It is expected that all white hat hackers follow a code of ethics in order to conduct a safe, legal and professional standard of penetration test. Without this, there is no difference between a white hat hacker and a black hat hacker.

Some of the rules they may follow include:

- Staying within the limits of the network - penetration testers may have full access to a client's network, however, most clients will have strict rules on which areas of the network they would like to be tested. It is critical that the penetration tester does not go beyond the boundaries of the network, as it could cause severe damage to the client's infrastructure.
- Maintain clear communication with the client - this ensures that the client is given all of the relevant information, which can be used to improve their systems.
- Do not discuss the results and outcomes of the penetration test with other third parties - all of the information gained within the assessment must be kept private and classified (Johansen, 2017).

7.3 Future Work

Due to the scope of this thesis and certain limitations, it was advised to demonstrate a single exploit. A way to improve upon this would be to explore in more detail the other components of the device, as well as to test more vulnerabilities and evaluate their security. More work can be done on vulnerability testing the mobile application, for example, as well as investigating further into the cloud services offered.

Due to technical errors in the Wyze Web View (the area of viewing recordings within the web application), further research had to be abandoned in this regard, which would be an interesting avenue to explore in the future.

Furthermore, since this report was written from a black box testing perspective, i.e. without full knowledge of the internal system and structures, an improvement would be to conduct white box testing, where a more informed evaluation could be conducted.

Chapter 8. Conclusion

Throughout this report, the topics of:

- Security concerns that exist in IoT devices,
- Mitigation and remediation methods for users and manufacturers,
- Identifying vulnerable devices,
- Penetration testing attacks and methods,
- And a proof of concept exploit for an IoT device

have been explored and discussed.

To conclude, we must answer the question proposed in the goal of this text (section 1.2): '*how secure are Internet of Things devices?'* The answer to this is, 'there is room for improvement'.

The many vulnerabilities discussed throughout this report, as well as the exploit demonstration, show the ease at which internet-enabled devices can be hacked into and used to capture private information. Using the Wyze Cam V2 specifically, an attack was able to take place by making changes to the firmware and physically implanting them onto the device. Certain frameworks of security must be in place to prevent these weaknesses and restrict unlawful access to devices and the compromise of networks.

As mentioned previously, many manufacturers of Internet of Things devices feel the need to skip on security due to profit margins and as a result, put their customers at risk. Through the information deliberated in this report, I am hopeful that improvements can be made.

Bibliography

Aviva (2020) *Tech Nation: Number of internet-connected devices grows to 10 per home, Aviva plc*. Available at: https://www.aviva.com/newsroom/news-releases/2020/01/tech-nation-number-of-internet-connected-devices-grows-to-10-per-home/ (Accessed: October 15, 2022).

Brewster, T. (2016) *How Jeep hackers took over steering and forced emergency stop at high speed, Forbes*. Forbes Magazine. Available at: https://www.forbes.com/sites/thomasbrewster/2016/08/02/charlie-miller-chris-valasek-jeep-hackers-steering-brake/?sh=1563e1cf63f4 (Accessed: October 27, 2022).

Cloudflare (2016) *What is the Mirai botnet?*, Cloudflare. Available at: https://www.cloudflare.com/learning/ddos/glossary/mirai-botnet/ (Accessed: December 11, 2022).

Cloudflare (2020) *What is the OWASP Top 10 Vulnerabilities?*, Cloudflare. Available at: https://www.cloudflare.com/learning/security/threats/owasp-top-10/ (Accessed: December 14, 2022).

Das, M.L. and Samdaria, N. (2014) *On the security of SSL/TLS-enabled applications, ScienceDirect*. King Saud University. Available at: https://www.sciencedirect.com/science/article/pii/S2210832714000039 (Accessed: December 2, 2022).

Department for Digital, Culture, Media & Sport (2021) New cyber security laws to protect smart devices amid pandemic sales surge *Gov.uk* [Preprint]. Available at: https://www.gov.uk/government/news/new-cyber-security-laws-to-protect-smart-devices-amid-pandemic-sales-surge (Accessed: October 30, 2022).

Department for Digital, Culture, Media & Sport (2022) New laws proposed to strengthen the UK's resilience from Cyber Attack, GOV.UK. Available at: https://www.gov.uk/government/news/new-laws-proposed-to-strengthen-the-uks-resilience-from-cyber-attack (Accessed: December 14, 2022).

Eclypsium (2019) *Anatomy of a firmware attack, Security Boulevard*. Available at: https://securityboulevard.com/2019/12/anatomy-of-a-firmware-attack/ (Accessed: November 17, 2022).

Ford (2020) *How To Hack IoT Cameras - Vulnerability Demonstration*. Available at: https://youtu.be/jiYv-bQ2UX8 (Accessed: October 22, 2022).

Goodreau, T. (no date) *7 tips to secure your smart home: IEEE Computer Society*. Available at: https://www.computer.org/publications/tech-news/trends/7-actionable-tips-to-secure-your-smart-home-and-iot-devices (Accessed: October 22, 2022).

Gudymenko, I. (2011) *Security in the Internet of Things*. rep. Available at: http://mhutter.org/papers/Gudymenko2011SecurityInThe.pdf (Accessed: November 16, 2022).

Guest, T. (2022) *Top IOT security risks and vulnerabilities*, BeyondTrust. Available at: https://www.beyondtrust.com/blog/entry/top-iot-security-vulnerabilities (Accessed: December 14, 2022).

Hilliard, G. (2020) "Hacking Reolink cameras for fun and profit," *Thirtythreeforty.net*, 16 May. Available at: https://www.thirtythreeforty.net/posts/2020/05/hacking-reolink-cameras-for-fun-and-profit/ (Accessed: November 17, 2022).

Hojlo, J. (2021) *Future of industry ecosystems: Shared data and Insights, IDC Blog*. Available at: https://blogs.idc.com/2021/01/06/future-of-industry-ecosystems-shared-data-and-insights/ (Accessed: October 22, 2022).

Johansen, R. (2017) Ethical hacking code of ethics: Security, risk & issues, Panmore Institute. Available at: https://panmore.com/ethical-hacking-code-of-ethics-security-risk-issues (Accessed: March 19, 2023).

Kali (2020) *What is Kali Linux?: Kali linux documentation*, *Kali Linux*. Available at: https://www.kali.org/docs/introduction/what-is-kali-linux/ (Accessed: February 17, 2023).

Kang, Y.-S. *et al.* (2015) *Comparative Study of Penetration Test Methods* . rep. Available at: https://web.archive.org/web/20180603043546id_/http:/onlinepresent.org/proceedings/vol87_2015/8.pdf (Accessed: December 3, 2022).

Langkemper, S. (no date) *The most important security problems with IOT devices, Eurofins - Cyber Security*. Available at: https://www.eurofins-cybersecurity.com/news/security-problems-iot-devices/ (Accessed: November 4, 2022).

Liu, X. (2021) *Ethical Hacking of a Smart Video Doorbell*. rep. Available at: http://kth.diva-portal.org/smash/get/diva2:1637549/FULLTEXT01.pdf (Accessed: October 15, 2022).

Lubeck, T. (no date) *Distributed password cracking with john the ripper , Tufts University*. Available at: https://www.cs.tufts.edu/comp/116/archive/fall2013/tlubeck.pdf (Accessed: February 20, 2023).

Lueth, K.L. (2014) *Why the internet of things is called internet of things: Definition, history, disambiguation, IoT Analytics*. Available at: https://iot-analytics.com/internet-of-things-definition (Accessed: October 30, 2022).

National Institute of Standards and Technology (2022) *CVE-2019-12266 Detail*, *NVD*. Available at: https://nvd.nist.gov/vuln/detail/cve-2019-12266 (Accessed: December 8, 2022).

National Institute of Standards and Technology (2022) *CVE-2019-9564 Detail*, *NVD*. Available at: https://nvd.nist.gov/vuln/detail/CVE-2019-9564 (Accessed: December 8, 2022).

National Institute of Standards and Technology (no date) *Glossary - Backdoor, National Institute of Standards and Technology*. Available at: https://csrc.nist.gov/glossary/term/backdoor (Accessed: November 4, 2022).

Nezza (2020) *Github*. Available at: https://github.com/ghidraninja/wyze_scripts/blob/master/wyze_extractor.py (Accessed: December 5, 2022).

Paul, I. (2015) *Lenovo Preinstalls Man-in-the-middle adware on new pcs, PCWorld*. Available at: https://www.pcworld.com/article/431959/lenovo-preinstalls-man-in-the-middle-adware-on-new-pcs.html (Accessed: March 25, 2023).

Pratt, M.K. (2022) *What is a cyber attack?, TechTarget*. Available at: https://www.techtarget.com/searchsecurity/definition/cyber-attack (Accessed: November 21, 2022).

Radholm, F. and Abefelt, N. (2020) *Ethical Hacking of an IoT-device: Threat Assessment and Penetration Testing*. rep. Available at: http://kth.diva-portal.org/smash/get/diva2:1472577/FULLTEXT01.pdf (Accessed: October 17, 2022).

Rosencrance, L. (2021) *What is antimalware?, TechTarget*. Available at: https://www.techtarget.com/searchsecurity/definition/antimalware (Accessed: November 3, 2022).

Stacksmashing (2020) *IoT Security: Backdooring a smart camera by creating a malicious firmware upgrade*. Available at: https://youtu.be/hV8W4o-Mu2o (Accessed: December 1, 2022).

Teja, R. (2021) *Firmware vs software: Difference between software and firmware, Electronics Hub*. Available at: https://www.electronicshub.org/firmware-vs-software/ (Accessed: March 1, 2023).

Tung, L. (2019) *FDA warning: Scores of heart implants can be hacked from 20ft away, ZDNET*. Available at: https://www.zdnet.com/article/fda-warning-scores-of-heart-implants-can-be-hacked-from-20ft-away/ (Accessed: October 30, 2022).

Vailshery, L.S. (2022) *IOT connected devices worldwide 2019-2030, Statista*. Available at: https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/ (Accessed: October 17, 2022).

What are IOT devices (no date) *Arm*. Available at: https://www.arm.com/glossary/iot-devices (Accessed: October 17, 2022).

Woodhouse, D. (2003) *JFFS2: The Journalling Flash File System, version 2, JFFS2: The journalling Flash File System, version 2*. Available at: https://sourceware.org/jffs2/ (Accessed: December 6, 2022).

Zaidan, D.T. (2021) *Analyzing Attacking methods on Wi-Fi wireless networks pertaining (WEP, WPA-WPA2) security protocols*. rep. Available at: http://pen.ius.edu.ba/index.php/pen/article/view/2545/1022 (Accessed: November 20, 2022).