



DEGREE PROJECT IN COMPUTER SCIENCE,
SECOND CYCLE, 30 CREDITS
STOCKHOLM, SWEDEN 2021

Threat Modeling and Penetration Testing of an IoT-product

A Survey on the Security of a Yanzi IoT Network

Diyala Isabar

Authors

Diyala Isabar, diyala@kth.se
Electrical Engineering and Computer Science
KTH Royal Institute of Technology

Place for Project

Stockholm, Sweden

Examiner

Robert Lagerström
Stockholm
KTH Royal Institute of Technology

Supervisor

Emre Süren
Stockholm
KTH Royal Institute of Technology

Abstract

Today, our society is more technology-reliant than ever, whether it is the individual or a small/large corporation, they all rely on computer systems in one way or another. Particularly Internet of Things has become increasingly popular with both industries and private consumers over the years. While there are several advantages with IoT solutions, they have also introduced new security threats that can not be overlooked. It is not uncommon that IoTs contain unknown vulnerabilities, which could potentially be exploited by cybercriminals to obtain sensitive information. Such vulnerabilities can easily be avoided if the security of the IoT is tested before it is launched. This degree project aims to assess the security aspects of a connected IoT device known as the "Yanzi IoT Network", designed for smart office environments. Different solutions will be proposed for each found vulnerability. The assessment is divided into three phases: planning, penetration testing and evaluation. The radio communication between a gateway and a few sensors is the targeted attack surface. Due to some technical issues, not all suggested attacks were possible to perform. Out of two that were possible to perform, one of them revealed a security flaw. The attacks were performed with a HackRF One as a transceiver, GNU Radio (Companion) to build different flow graphs, GQRX for frequency checks and Universal Radio Hacker (URL) for different purposes.

Keywords

Security, Internet of Things, penetration testing, threat assessment, ethical hacking, vulnerabilities, radio communication, sensor, smart office, gateway, Yanzi, 6LoWPAN, HackRF, GNU Radio

Abstrakt

Idag är vårt samhälle mer tekniskt beroende än någonsin, oavsett om det är individen eller ett litet / stort företag, de litar alla på datorsystem på ett eller annat sätt. Särskilt Internet of Things har blivit allt populärare hos både industrier och privata konsumenter genom åren. Även om det finns flera fördelar med IoT-lösningar har de också infört nya säkerhetshot som inte kan förbises. Det är inte ovanligt att IoTs innehåller okända sårbarheter som potentiellt kan utnyttjas av kriminella för att få tag på känslig information. Sådana sårbarheter kan lätt undvikas om en IoT enhets säkerhet testas innan den lanseras. Detta examensarbete syftar till att bedöma säkerhetsaspekterna hos en ansluten IoT-enhet som kallas "Yanzi IoT Network", designad för smarta kontorsmiljöer. Olika lösningar kommer att föreslås för varje funnen sårbarhet. Processen är indelad i tre faser: planering, penetrationstest och utvärdering. Radiokommunikationen mellan en gateway och några sensorer är den valda attackytan. På grund av vissa tekniska problem var det inte möjligt att utföra alla föreslagna attacker. Av de två som var möjliga att utföra avslöjade en av dem ett säkerhetsfel. Attackerna utfördes med en HackRF One som sändtagare, GNU Radio (Companion) för att bygga olika flödesdiagram, GQRX för frekvenskontroller och Universal Radio Hacker (URH) för olika ändamål.

Nyckelord

Säkerhet, "Internet av saker", penetrationstestning, hotbedömning, etisk hacking, sårbarheter, radiokommunikation, sensor, smart kontor, gateway, Yanzi, 6LoWPAN, HackRF, GNU Radio

Acknowledgements

I would like to thank my supervisor Emre Süren and examiner Professor Robert Lagerström for their support and participation in this thesis project. A big thank you to research engineer Johannes Olegård, who quickly provided me with both technical and practical assistance, and has given me valuable advices throughout the whole thesis.

I would also like to thank my supervisor from the partner company Coor, Mattias Wahlgren, for providing me with information and contacts whenever needed, and a thank you to Oriol Piñol at Yanzi for providing me with technical assistance.

My final thank you goes out to my friends and family that have followed and supported me throughout this whole journey.

Acronyms

Contents

1	Introduction	1
1.1	Background	2
1.2	Problem statement	2
1.3	Objective	3
1.4	Methodology	4
1.5	Delimitations	5
1.6	Benefits, Ethics and Sustainability	5
1.6.1	Benefits	5
1.6.2	Ethics	5
1.6.3	Sustainability	6
1.7	Outline	7
2	Background	8
2.1	Smart office solutions	8
2.2	Cloud service	9
2.2.1	Software as a Service	9
2.2.2	Common attacks	9
2.3	Radio communication	11
2.3.1	6LoWPAN	12
2.3.2	Common attacks	14
2.4	Encryption	17
2.4.1	Elliptic-Curve Cryptography	17
2.4.2	Advanced Encryption Standard	17
2.4.3	Transport Layer Security/Secure Sockets Layer	17
2.5	Meta Attack Language	18
2.5.1	securiCAD	19

2.6	Related Work	19
3	Methodology	20
3.1	Planning	20
3.1.1	Threat modeling	20
3.2	Penetration testing	24
3.2.1	Tools & Environment	25
3.3	Evaluation	26
4	Yanzi IoT system	27
4.1	Yanzi System	27
4.1.1	Yanzi Cirrus Software as a Service	28
4.1.2	Gateways	29
4.1.3	Sensors	30
4.1.4	Yanzi security overview	31
5	Planning	32
5.1	Threat model	32
5.1.1	Identified assets	32
5.1.2	Architecture overview	33
5.1.3	Decomposition of the IoT device	35
5.1.4	Identified threats	37
5.1.5	Documented threats	38
5.1.6	Rated and selected threats	40
6	Penetration testing	43
6.1	Planning	43
6.2	Discovery	43
6.3	Testing	45
6.3.1	Examining of packets	45
6.3.2	Replay attack	46
6.3.3	Denial of service	50
7	Results	53
8	Discussion	54

9 Conclusion	57
References	58

Chapter 1

Introduction

Today, our society is more technology-reliant than ever, and digitalization will only continue, showing no signs of slowing down. Whether it is the individual or a small/large corporation, they all rely on computer systems in one way or another. Particularly Internet of Things (IoT) have become increasingly popular with both industries and private consumers over the years. Not only have they improved and simplified both work and life quality, but they have also contributed to economic sustainability. It has been forecasted that the number of connected devices around the world will be roughly 14 billion in the year 2022 ¹.

One particular field that the booming of IoT solutions continues to create endless possibilities for is smart offices. Several different IoT devices are connected in an office environment to create a better workplace and enables a better, faster and smarter working approach. Benefits that increasingly more corporations want to take advantage of and by the year 2023, it is expected that the smart office market will be valued to USD 46.11 Billion ².

However, the increase of connected devices has introduced the threat of an expanding attack surface, which in turn has lead to the rise of cyberattacks ³. One effective approach to prevent these attacks is to perform ethical hacking on a device to test its security. Ethical hacking is performed with an organization's knowledge and involves repeated attempts to gain unauthorized access to the device. The purpose is to expose

¹<https://blog.bosch-si.com/internetofthings/market-size-and-connected-devices-where-the-future-of-iot/>

²<https://www.peerbits.com/blog/how-iot-is-building-the-smart-offices-of-tomorrow.html#>

³<https://www.cisco.com/c/en/us/solutions/internet-of-things/future-of-iot.html>

vulnerabilities that later could be exploited by a hacker [1].

1.1 Background

The term "Internet of Things" was coined by Kevin Ashton in 1999 that describes the collection of objects that are connected to the Internet and embedded with different technologies such as sensors and actuators. The goal of IoT is to collect and provide information about objects around us by unifying people, "things," places, and processes under a universal infrastructure [2]. Information exchange between systems and devices is automated and performed over the Internet with specific communication technologies [3].

Some common technologies utilized in IoTs are showcased below [4]:

IoT Technologies
Communication Technologies IEEE802.15.4, Zigbee, Bluetooth, WiFi, Ethernet etc.
Prototype Hardwares Raspberry Pi, Arduino One, Hackberry, Rascal etc.
Identification Techniques RFID, IPv6, Barcode, QR Code etc.
IoT Architectures OpenIoT, 5-layer, 3-layer, IoT-A, BeTaaS etc.
Operating Systems Mantis, Contiki, LiteOS, Tiny OS etc.
Protocols 6LoWPAN, IPV6, NanoIP, UDP etc.

The application of IoT can be found in several different fields, where some common ones are smart homes, smart buildings, smart offices, traffic monitoring, healthcare, smart grids etc [3].

1.2 Problem statement

While there are several advantages with IoT solutions, they have also introduced new security threats that can not be overlooked. In [3], [2] and [4], security and privacy

issues related to IoT devices are described, and in this blog ⁴ the ten most significant challenges for IoTs are numbered. The first issue stated is insufficient testing and updating. One main problem with corporations that manufacture devices of this type is their carelessness in device-related security. When an IoT device is manufactured, there is an eagerness to produce and deliver it as quickly as possible, resulting in security negligence. An absence of security testing of a device before it is launched into the market and regular firmware updates leave customers with devices that could contain vulnerabilities, which could easily be exploited.

Security issues in IoT devices are a matter of concern wherever they are utilized. However, it is incredibly worrying that they are present in many homes and offices, making them prone to a home invasion. The IP address related to the device could uncover sensitive information, such as the address of the place where the device is located, and then potentially be sold on the illegal market. Furthermore, incorporating IoT in security systems could compromise the building/home further ⁵. Moreover, it is essential to raise awareness of this problem since it further extends to national security when considering that these devices are also utilized in government offices and can therefore pose a significant threat to the nation.

Because of the security issues in IoT devices, several successful attacks have been conducted, resulting in severe damages. One of them is the infamous Mirai botnet [5], where a DDoS attack was performed by exploiting vulnerable IoT devices. Another example is when two security researchers managed to hijack a Jeep car of the model Cherokee over the Internet. They took control over the car's brakes and accelerator, but also less essential parts, such as the radio and the horn ⁶.

1.3 Objective

This degree project aims to assess the security aspects of a connected IoT device known as the "Yanzi IoT Network", designed for smart office environments. In this scope, any vulnerabilities found are reported, and different solutions to how they can be managed are proposed and discussed.

The research question(s) that this thesis aims to answer is/are: *What vulnerabilities*

⁴<https://www.peerbits.com/blog/biggest-iot-security-challenges.html>

⁵See footnote 4

⁶<https://www.kaspersky.com/blog/remote-car-hack/9395>

can be identified in the Yanzi IoT product? If vulnerabilities are identified, how can they easily be managed in future office environments?

The research question was broken down into six objectives:

- Information gathering and literature research
- Threat modeling (use cases, use case diagrams, architecture map)
- Vulnerability analysis, (previously known vulnerabilities, most common and trending vulnerabilities, searching in CWE)
- Risk assessment (scoring each found threat to compare and prioritize the severity of risk presented by each threat).
- Penetration testing on radio communication
- Propose solutions to found vulnerabilities
- Evaluation of obtained results

1.4 Methodology

The project methodology is divided into three sequential phases, following the proposed method in [6], to conduct security tests on the Yanzi IoT system systematically. The first phase is planning, which consists of literature research, information gathering, and threat modeling, where potential threats are identified using the STRIDE model. An architecture overview of the system is showcased in both draw.io and SecuriCAD. This is performed to fully understand the systems underlying technologies and the threats it faces. The second (and largest) phase consists of penetration testing, where a modified version of the NIST model will be used. Testing is conducted, taking into consideration solely the most critical threats derived from the first phase. Threats are scored and prioritized using the DREAD model. Furthermore, some literature was also used to find possible and/or common weaknesses that would also be taken into consideration when test cases were selected. Tests are performed to test the security of the product distinctly. As for the last phase, an evaluation of the results from the penetration testing is conducted. The information from this phase is used for the discussion and conclusion regarding the security of the product.

1.5 Delimitations

Solely one attack surface will be investigated in this thesis; the radio communication. Hardware, firmware, web application (cloud) and mobile application are excluded due to the time constraint of the degree project, but also because some are not part of the scope of the education.

1.6 Benefits, Ethics and Sustainability

1.6.1 Benefits

The company Coor is one of the leading facility management suppliers in the Nordics, where one of the services that they provide is smart solutions (IoT devices) for smart facilities. One such solution is the Yanzi IoT network, which is manufactured by a software company named Yanzi. Therefore, it is not peculiar that they want to know if the products they sell contain vulnerabilities that people with malicious intent can exploit.

Since testing is performed on the Yanzi IoT network, whether vulnerabilities are found or not, both companies will benefit from this degree project.

1.6.2 Ethics

The word "hacker" is often associated with someone who performs illegal and unethical actions on computer systems and has gotten a negative hype around it. However, it is well known to the more informed person that this perception is not completely true. A hacker is often referred to as a White hat, Black hat, or Grey hat, depending on the hacking approach in terms of followed intentions [7]. This thesis follows a white hat approach, where testing is performed with consent, and the intention is to spot weaknesses and offer solutions to enhance the product's security. Furthermore, different Swedish laws and regulations that concern security are also taken into consideration when testing is performed. The following Swedish laws are taken into consideration:

- The Swedish Criminal Code (1962:700); 4 chap.8 § states that anyone who illegally tries to obtain access of a message that is transmitted in an electronic

communications network is sentenced to a fine or imprisonment for a maximum of two years. Law (2012:280)

- The Swedish Criminal Code (1962:700); 4 chap.9c § states that anyone who illegally tries to obtain access of a resource that is processed automatically or illegally tampers, change, block or obliterates such resource is sentenced to data breach, either to a fine or up to two years in prison. This also applies if anyone illegally through a similar measure interfere or prevent the use of such a resource.

Another ethical aspect to consider is how to proceed whenever a vulnerability is found. Full disclosure and responsible disclosure are two common approaches used to bring awareness of such a finding when it has been detected ⁷. When a full-disclosure approach is followed, the vulnerability is immediately made public, e.g. on social media, without contacting the corporation responsible for the product. Thus, giving the corporation no time to fix it. One could argue that this is the superior approach since product users are quickly notified about the problem and can instantly take actions to protect themselves. However, such an announcement also makes adversaries aware of the problem simultaneously, enabling them to exploit the vulnerability for malicious intents before a patch is available. As for the responsible disclosure approach, the corporation is contacted and notified about the vulnerability before disclosing it to the public. Often an agreed time frame is set, where the expectation is that an investigation is performed to validate and patch the finding. Since the testing in this thesis project is completed with consent and the corporations wish to test the product to see any vulnerabilities present, a responsible disclosure approach is followed.

1.6.3 Sustainability

Conducting a security assessment of an IoT device from a sustainable perspective has not significantly impacted social or ecological sustainability. With a trivial exception, namely the electrical power needed to power the gateway and one of the sensors. Furthermore, performing such an assessment could potentially contribute to economic sustainability as well as save a lot of time. Vulnerabilities found and disclosed responsibly could be fixed before anyone with malicious intent exploits them,

⁷https://cheatsheetseries.owasp.org/cheatsheets/Vulnerability_Disclosure_Cheat_Sheet.html#methods-of-disclosure

preventing costly and time-consuming damages.

1.7 Outline

This report is divided into nine chapters; the introduction and then the following eight chapters:

- Chapter 2 Background: introduces the subject, presents related work and relevant theoretical background.
- Chapter 3 Methodology: describes the project methodology and utilized tools.
- Chapter 4 Yanzi IoT system: presents the system that is under consideration.
- Chapter 5 Planning: covers the planning phase.
- Chapter 6 Penetration testing: covers the conducted penetration tests.
- Chapter 7 Results: presents the obtained results from the penetration testing.
- Chapter 8 Discussion: covers the discussion of the obtained results.
- Chapter 9 Conclusion: covers the thesis conclusion and future work suggestions.

Chapter 2

Background

Since this research focuses on an IoT solution for smart offices, the following chapter will first present information about different smart office solutions currently on the market, followed by relevant theoretical background, common attacks as well as related work.

2.1 Smart office solutions

Integrating technologies into work environments to enhance daily working life for employees is becoming a rapidly evolving concept. IoT devices such as sensors are often connected to a web or mobile application that functions as a management tool, enabling, for example, process automation and monitoring of employee occupancy. One example of such an application is **Optimize Workspace**¹. It is developed by the company Rapal with the purpose of improving the workplace experience and collecting IoT data for continuous workplace management. The smart office solution is designed to be used with any IoT sensor, enabling tracking of space availability, indoor air quality, etc., if desired. **Cobundu**² is another smart office platform developed by Spacewell, a solution similar to Optimize, manufactured to enhance comfort and use of spaces and resources but also to assist users in real-time. A third solution, called **Empathic Building**³, is an application that collects data from facilities intending to help employees co-operate and perform better, and in that way enhance daily working

¹<https://www.rapal.com/smart-office>

²<https://spacewell.com/sv/varumarken/cobundu-smart-byggnad/>

³<https://haltian.com/newsandpress/iot-house-haltian-acquires-new-empathic-building-service-to-complete-its-smart-building-solutions/>

life. Copies of the facilities in the form of virtual maps are included in the platform, enabling users to view available workspaces and rooms.

2.2 Cloud service

IoT devices are usually connected to some cloud component, to which collected data is sent, and the Yanzi IoT device is no exception. The cloud itself is accessed via APIs. Users can in this manner monitor a device and all information related to it, such as data analytics, control permissions, etc [6].

To meet the need of a wide variety of users, three different cloud service models [8] are offered: Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS) . This study will only focus on the latter.

2.2.1 Software as a Service

In this model, the user/consumer does not control any technology or code related to the application, such as the infrastructure of the network, operating system, or hardware. Thus, the user only accesses the application through a regular web browser. The model is very cost-effective since it allows users to rent cloud-based applications rather than paying to purchase them. Because of this reason, companies that want to deploy their businesses have become very fond of it. [8]

There are several security fears related to the SaaS model. However, one of the more important ones is data loss. Each cloud model use and store data in a different way, for example in SaaS business, data is processed and customer's data is stored in databases with the help of applications. However, all three have a common denominator, which is that data can be accessed by unauthorized employees that work at the company intentionally or accidentally, and external hackers. Session hijacking and network channel snooping are two examples of hacking techniques that hackers can utilize to gain access to databases in cloud environments. [8]

2.2.2 Common attacks

It is not uncommon that web applications contain serious vulnerabilities and weaknesses, often without the developers' knowledge. Such vulnerabilities could

potentially be exploited by criminals to obtain access to sensitive information or gain unauthorized access to the system. They are exploited by cybercriminals performing different attacks, including SQL Injection and Cross-site WebSocket hijacking. Both attacks are described in more detail down below.

SQL Injection

A SQL injection⁴ attack is a type of injection attack that consists of insertion ("injection") of SQL commands into some data-input kind of box, e.g., one for entering passwords. The purpose is to affect the execution of SQL commands that are already defined in order to gain access to the database.

If the attack succeeds, the attacker is able to modify stored data (insert/update/delete), read sensitive data but also execute administration-related operations, such as closing down the DBMS. Furthermore, since the attacker is also allowed access to the DBMS file system, he/she can recover the content of a specific file present in the file system and, in rare cases, issue commands that require administrative privileges, such as communication with the operating system.

Cross-site WebSocket hijacking

A Cross-site WebSocket hijacking attack⁵ is similar to a CSRF attack but with a small difference; the CSRF vulnerability is on the WebSocket handshake. Such a vulnerability emerges when the identification of the user that made the WebSocket handshake request is solely based on HTTP cookies as well as the request not containing any unpredictable values or CSRF tokens. The vulnerability allows an attacker to disguise itself as the victim user in order to perform unintended/unauthorized actions, such as send messages to the vulnerable application's server as well as retrieve and read sensitive data that is sent back from the server to the user. This two-way interaction with the vulnerable application is another difference from the regular CSRF.

An attacker usually creates a malicious web page (malicious.com) and thereafter with the help of social engineering/phishing tries to deceive the victim into opening it in his/her browser. If the victim is logged into another web page (victim.com),

⁴https://owasp.org/www-community/attacks/SQL_Injection

⁵<https://portswigger.net/web-security/websockets/cross-site-websocket-hijacking>

malicious.com can attempt to establish a cross-site WebSocket connection to the victim's server by utilizing the session from the connection with victim.com.

2.3 Radio communication

It is today very common for IoT devices to interact with each other to share and exchange information, and often use wireless/radio communication to achieve this. Wireless/radio communication is a way of using electromagnetic waves to transmit data from source to destination over the air communication medium. [7]

For a better understanding of this section, essential terminologies are presented below:

- *Wavelength* - indicates the distance between two subsequent high points or two subsequent low points in the waveform.
- *Frequency* - refers to how frequently an event occurs.
- *Signal-to-Noise ratio* - a measure that compares the level of the desired signal with the level of background noise.
- *Gain* - the ratio of the new processed signal divided by the original signal, describing how much they differ after a signal has been processed.
- *Digital Modulation* - the process of encoding a digital signal into a carrier waves amplitude, frequency or phase.

Before any digital data (in bits) is transmitted, it is first converted into an electrical signal and then modulated with a carrier wave, which is later transmitted to the receiver [7] . At the receiver side, the modulated wave is demodulated to recover the original data that was transmitted. There are different types of digital modulation techniques, including Frequency-shift Keying (FSK), Amplitude-shift Keying (ASK), On-Off Keying (OOK) and Phase-shift Keying (PSK). PSK refers to when information is encoded as variations in the instantaneous phase of a carrier wave, and the 2.4GHZ frequency band specifically relies on a special variant of it, called Offset Quadrature Phase-Shift Keying (O-QPSK), which encodes two bits per symbol.

IoT devices work on various radio protocols, including Zigbee, Bluetooth Low Energy (BLE), SDR, and 6LoWPAN (IEEE 802.15.4) [6, 7]. Because the IoT device tested in

this paper utilizes 6LoWPAN to communicate, the following subsection will focus on that protocol.

2.3.1 6LoWPAN

6LoWPAN [9], an acronym for IPv6 over Low-power Wireless Personal Area Networks, is a concept developed to enable the participation of small and low-power devices with limited processing capabilities in the Internet of Things. To make this possible, an adaption layer was defined, and the IPv6 protocol functionality was simplified by defining encapsulation and more compact header formats, allowing IPv6 packets to be transmitted over IEEE 802.15.4 based networks. Figure 2.3.1 illustrates a typical 6LoWPAN protocol stack, divided into six layers.

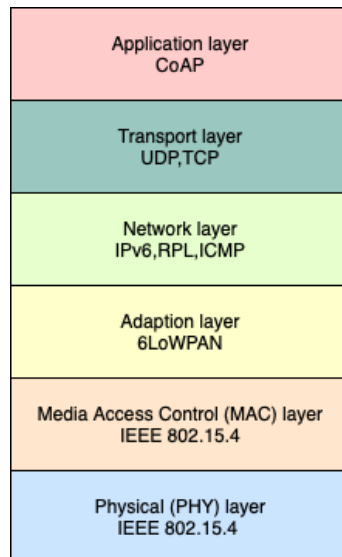


Figure 2.3.1: 6LoWPAN stack overview

The Internet connection to the physical world enables several interesting applications where 6LoWPAN could be applicable, ranging from personal health sensor monitoring to large-scale facility monitoring. More specifically, it can be applied in home and building automation, personal health and fitness, real-time environmental monitoring and forecasting, healthcare automation and logistics, tracking, maintenance etc. [9]

Adaption layer

The maximum transmission unit for IPv6 is 1280 bytes, which is greater than the largest possible packet size for 802.15.4. Depending on overhead, the 802.15.4

protocol data unit (PDU) will be of different sizes. Overhead is usually additional data that is necessary to send the payload, such as headers but also implemented security, such as AES. Available MAC payload can sometimes be as little as 81 bytes, which is unquestionably below the minimum IPv6 requirements. An adaption layer where fragmentation and reassembly is performed below the network layer is therefore needed. In addition, the IPv6 header itself is 40 bytes, affecting the rest of the payload sizes up in the stack, resulting in a need of header compression as well.[10]

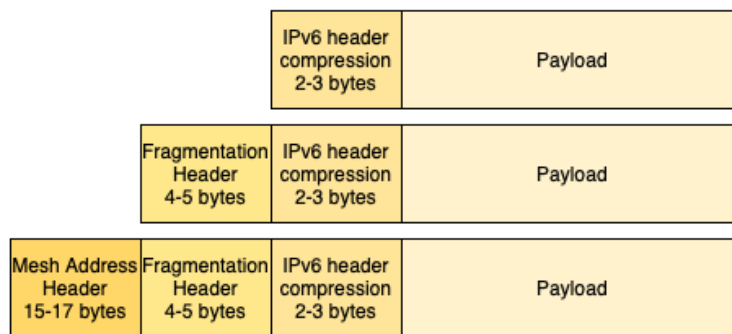


Figure 2.3.2: 6LoWPAN Encapsulation Header Stack

The adaption layer is located between the Network layer and the MAC layer, as illustrated in the figure 2.3.1, and this is where encapsulation of IPv6 datagrams is performed. All encapsulated LoWPAN datagrams transmitted via IEEE 802.15.4 are preceded by an encapsulation header stack (see figure 2.3.2), that contains at least one header, followed by zero or more header fields. The first byte, called the dispatch byte, of each header in the header stack, indicates the type of the header. [10]

IEEE 802.15.4

The IEEE 802.15.4 standard defines the protocol for the operation of low-rate wireless personal area networks (LR-WPAN) [9]. 16 channels on the 2.4 GHz spectrum are supported by the protocol, with 250 kbps per channel and 2 MHz in bandwidth for each channel ⁶. Furthermore, it describes the specifications for the Physical (PHY) layer and Medium Access Control (MAC) sublayer for wireless connectivity with devices that requires limited battery consumption [11]. The MAC and PHY layers are considered the base and are therefore the main target for a majority of the cyber attacks [12].

The **MAC** sublayer (see figure 2.3.3) includes both the MAC Management Service and the MAC Data Service. The latter allows MAC protocol data units (MPDUS) to be

⁶<https://yanzi.dev/#/manual/index?id=ieee-802154-overview>

transmitted and received over the PHY data service. Tasks that the MAC sublayer is responsible for include amongst others, channel access, guaranteed time slots (GTS) management, frame delivery acknowledgement and beacon management. [11]

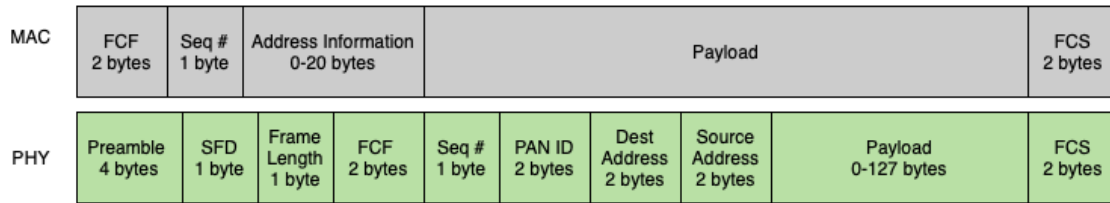


Figure 2.3.3: MAC and PHY packet overview

The **PHY** layer [10] includes the Physical Layer Management Entity (PLME), which provides the layer management service interface, used to invoke layer management functions. It also provides an interface between the physical radio channel and the MAC layer, through radio frequency (RF) hardware and RF firmware. Furthermore, the **PHY** layer is responsible for several tasks, including channel selection, Energy Detection (ED) for the current channel, radio transceiver activation and deactivation, link quality index (LQI) for received packets, but also receiving and transmitting packets traversing the physical medium [11]. A MAC packet with its different fields is illustrated in the figure 2.3.3.

2.3.2 Common attacks

Wireless communication is today a very attractive form of communication, which can quickly and easily be established, and at the same time offer a high level of security. However, a high level of security does not always equal secure communication. A small flaw or sometimes a complete lack of security enables the possibility for cyberattacks, where the attacker can eavesdrop on traffic or manipulate data that is transmitted to obtain sensitive information, amongst others. Some of the more common attacks on radio communication are described below.

Denial of Service

A Denial of Service attack (DoS) ⁷ [CWE-400] is typically accomplished by flooding a target/resource (application, server or site) with several requests in order to make it unavailable for the purpose it was constructed. However, there are also other ways

⁷https://owasp.org/www-community/attacks/Denial_of_Service

to make a resource unavailable by for e.g manipulating code, network packages or resources that manages vulnerabilities, but it could also be that a service/system might stop execute if a vulnerability has been exploited. Furthermore, if an attacker wants to access sensitive information or execute commands on the server, he/she can try to inject and execute arbitrary code while performing the attack.

The result of a DoS attack displays a significant degradation in the experience and service quality for users, which is expected since amongst others it interrupts services and causes large response delays, and therefore impacts the availability directly.

Jamming

A jamming attack [13] is a form of Denial of Service attack where the adversary deliberately disrupts or blocks data transmission between a transmitter and a receiver. The data exchange is not directly affected, but rather that the exchange rate is slowed down, thus threatening data availability. There exist several ways to perform this type of attack, which oftentimes involves a 'jammer'. A jammer is a transmitter that is exploited by a perpetrator to achieve the set-out goal. The jammer emits one or several radio signals called Radio Frequency (RF), which corresponds to the meaningless data that all communicating nodes receive. A *Radio jamming* attack [12] specifically is a jamming attack on the PHY layer, with the purpose to block or jam wireless communication. The goal is to disrupt the reception of messages at the involved nodes and is achieved by sending out radio signals intentionally to decrease the Signal to Noise Ratio (SNR) of ongoing radio communication. One approach to this attack is Pulse-Band Denial, also known as Single-Channel Pulse Jamming and the target, in this case, is a single channel within the related frequency band. A variant of this approach is Constant jamming, which is when radio signals is continuously transmitted over the target channel.

A *Link-Layer Jamming* attack [12] is similar to a Radio Jamming attack but targets the MAC layer instead. The attack involves disrupting message exchange between two nodes and is achieved by sending out packets rather than signals.

Replay

A Replay attack ⁸ [CWE-294] is the malicious act where an adversary eavesdrops and intercepts a protected network communication, with the purpose of either delay the traffic or resend it. By doing so, the attacker deceives the victim into performing actions that he/she desires, without the victim's knowledge. This attack does not require any advanced expertise within cryptography, since the adversary is not required to decrypt traffic for the attack to be successful. It is sufficient to simply resend the captured traffic to the receiver and is therefore why this attack is so dangerous.

Man-in-the-Middle

A Man-In-the-Middle attack (MitM)[14] [CWE-300] is a cyberattack where an adversary secretly positions himself in the middle of a communication between two parties. They believe that they are directly communicating with each other over a secure channel, unaware of the intruder. The intention of the attack is to intercept traffic and later possibly manipulate data (modification) or to exfiltrate information (eavesdropping/sniffing). Packet modification/corruption would be possible if the transmission protocol does not contain a mechanism to verify the integrity of the transmitted data, or it does not validate or incorrectly validates the integrity of a message's (header or payload) checksums [CWE-924-353- 354- 347]. Eavesdropping would be possible if there is a lack of encryption during data transmission [CWE-319]. Performing a MitM attack can therefore compromise both data/information integrity and confidentiality.

Spoofing

Spoofing ⁹ [CWE-290] refers to the act where a cybercriminal impersonates a trusted device or user to perform actions that are beneficial to the criminal. Such actions include accessing information, malware spreading, access control bypassing etc.

There are several types of spoofing attacks, including email spoofing, web spoofing, GPS spoofing, text message spoofing, IP spoofing etc. The latter is a widely known attack that primarily targets a network. An attacker attempts to gain unauthorized access to the system by sending a message with a forged or "spoofed" IP address to

⁸<https://www.kaspersky.com/resource-center/definitions/replay-attack>

⁹<https://www.kaspersky.com/resource-center/definitions/spoofing>

create the impression that the message came from a trusted source. This is achieved by taking the legitimate IP address of a source and modifying the header of the packet sent by the attacker's own system to make it appear to be from the trusted source.

2.4 Encryption

2.4.1 Elliptic-Curve Cryptography

Elliptic-Curve Cryptography (ECC) [15] is a public-key cryptosystem that utilizes the structure of mathematical elliptic curves over finite fields to encrypt data. It allows for the creation and use of smaller keys (in comparison to other encryption algorithms), and a fast generation of a key pair. The foundation of the algorithm is rather complex but simply put: the generation of a key pair is not performed through the traditional method where the product of very large prime numbers is used, instead ECC utilizes properties of the elliptic curve equation. Except for encryption of data, ECC can also be used for digital signatures and key-agreement.

2.4.2 Advanced Encryption Standard

Advanced Encryption Standard (AES) [16] is an encryption standard established by the National Institute of Standards and Technology (NIST) in 2001. It is a subset of the Rijndael block cipher, with a fixed block size of 128 bits, and a cipher key length that can be 128, 192 or 256 bits. The AES algorithm utilizes a symmetric-key, which indicates that data is encrypted and decrypted with the same key. The input of the algorithm is a plaintext in which rounds of different transformations are performed on. The number of rounds is dependent on the length of the cipher key; 128-bits will have 10 rounds, 192-bits 12 rounds and 256-bits 14 rounds. Each round includes four transformations: 1) byte substitution, 2) row shifting, 3) column mixing and 4) adding of the round key. However, in the final round, the third transformation is excluded.

2.4.3 Transport Layer Security/Secure Sockets Layer

Secure Sockets Layer (SSL) and Transport Layer Security (TLS) [17] are two cryptographic protocols that work on top of the transport layer protocol, with the purpose of providing secure end-to-end communication between a server and a

client over a computer network, and in that way ensure confidentiality, integrity and availability (CIA). A negotiation process called SSL/TSL handshaking is used for authentication and to create a session. It includes an exchange of several different parameters such as session ID, random number, cipher suite etc. The latter is used to encrypt the communication.

Both SSL and TSL can be utilized with the same different protocols (ex. HTTP, FTP etc) and they have the same architecture. However, since TLS is an upgraded version of SSL, changes have been made to strengthen the security. Such modifications include a new pseudo-random function, changes in digital signatures and key block, but also in security parameters and the MAC computation.

2.5 Meta Attack Language

The Meta Attack Language (MAL) [18] is a framework used to design domain-specific attack languages, which enables cyber threat modeling and attack simulations for particular systems in a domain, such as power grids ¹⁰.

The aim of attack simulations is to identify vulnerabilities in a system, with the time it takes to compromise each attack step as a result. Dependencies and relationships between different attack steps that might be performed by a malicious person are represented by attack graphs. For e.g an attacker may 1) perform SQL injection, 2) reveal admin password and username, 3) access site and perform malicious activities, etc. With such an attack simulation tool present, the security assessor can focus on information gathering that is needed for simulations instead of focusing on what information to collect about the system and the analyzing of such information to find weaknesses that could be exploited. [18]

coreLang [19] is a domain-specific language (DSL) based on MAL. It is designed for the IT domain and enables analysis of weaknesses associated to common attacks and modeling of IT infrastructures. The six fundamental assets that are included in coreLang are: **System, Vulnerability, User, Identity and Access Management, Data resources, Networking**. For a more detailed overview, readers are referred to the original paper.

¹⁰<https://mal-lang.org>

2.5.1 securiCAD

securiCAD [20] is a CAD tool developed for corporation cybersecurity management. First, the user models the architecture of an existing system or one under development, and then securiCAD simulates potential attacks and highlights possible weaknesses with the help of attack graphs.

The tool includes 23 types of assets, each one containing a few defenses and/or attacks. An example of an asset is data flow, which contains six attack types, e.g replay and eavesdrop, but no defenses. Another example is the protocol asset, which has three defenses, such as cryptographic authentication, but no attacks. The total number of attacks in securiCAD is 59 while the number of defenses is 58.

A user is provided with a heat map, showing different colors that represent an attacker's likelihood of reaching that asset utilizing a specific type of attack. Green indicates that it is not very likely to occur, while red states that it is. The three colors green, yellow and red can appear in different shades on the map. Another functionality provided by securiCAD is that a user can investigate the steps an attacker has to take to reach a certain point, which is shown with the help of red arrows. For a more detailed overview of securiCAD, see the original paper.

2.6 Related Work

To give more perspective on penetration testing and security in IoT devices that utilize radio communication, this section will highlight results obtained from sources related to the work in this paper.

In [21], the authors present existing vulnerabilities in a wireless sensor network, and then identifies and perform attacks that can possibly exploit such vulnerabilities. Eavesdropping was performed by capturing packets using the RTL2832U hardware as a sniffer together with GNU Radio and then analyzed in Wireshark. For the replay attack, the HackRF One hardware and GNU Radio were utilized. Similarly, [22], vulnerabilities in wireless IoT devices are studied using SDR. GQRX is used to identify the frequency the target operates on, HackRF One and GNU Radio to capture traffic and to perform attacks such as signal decoding, signal jamming and packet replaying. Furthermore, to mitigate such attacks, several different schemes are proposed.

Chapter 3

Methodology

This chapter describes the three phases of the project methodology more detailed. Tools are defined together with their purpose of use, and utilized methods are described step by step.

Noteworthy, a few modifications have been made to the different methods that are used. Some steps/sub-steps have switched place, are excluded or diverged from, and some merged together depending on relevancy and delimitations.

3.1 Planning

The intention of this phase is to evaluate the entire infrastructure, with the purpose of increasing knowledge about the product's functionality. This will subsequently provide a thorough threat estimation, i.e presence of vulnerabilities, for different components. [6] See chapter 6 for more details.

3.1.1 Threat modeling

Threat modeling¹ has become an important practice that assists system developers to understand security threats that systems and applications designed by them might encounter. It is described as a common approach to identify and understand threats with the purpose of developing mitigation strategies for potential vulnerabilities and in that manner increase the security of a system. Furthermore, threat modeling helps

¹<https://owasp.org/www-project-web-security-testing-guide/latest/2-Introduction/README.html#threat-modeling>

the developer to focus (often) limited resources and attention to the parts of the system that are most exposed. A threat model is a recommended standard practice for all applications/products/systems and should be created as early as possible in the development process.

The threat modeling approach that is followed in this paper is provided by Microsoft ², and is partitioned into a six-stage process: **Identify assets**, **Create an architecture overview**, **Decomposition of the product**, **Identify threats**, **Rate threats** and **Document threats**. This approach ensures a reviewing and assessment of possible attacks that are likely to occur in terms of risk, impact and likelihood that a vulnerability will be exploited. Each stage is described more in detail down below.

Identify assets

Reconnaissance is the very first step in the process where valuable assets are identified and documented. The purpose is, from a time perspective, to understand where to focus more probable attacks. Furthermore, a lot of time is saved during exploitation of the system if identified assets contain public vulnerabilities.[7]

Create an architecture overview

In the second stage of the process, an extensive documentation review of the products architecture (including physical architecture), subsystems, functionality etc, as well as identification of different technologies is performed. This helps with the visualization on how to attack the product to gain access in an unexpected way. The desired outcome is to discover design and implementation flaws. [7]

Three tasks are performed in this step ³:

- **Documentation of functionalities and features** - use cases are documented to help with the understanding of the device's functionalities and features, how it is operated, as well as to work out how the device can be misused.
- **Create a diagram of the product architecture** - a diagram that describes the architecture of the product, such as structure and composition, but also

²[https://docs.microsoft.com/en-us/previous-versions/msp-n-p/ff648644\(v=pandp.10\)?redirectedfrom=MSDN](https://docs.microsoft.com/en-us/previous-versions/msp-n-p/ff648644(v=pandp.10)?redirectedfrom=MSDN)

³See footnote 2

subsystems and physical traits.

- **Identify the technologies in use** - distinct technologies that are used in the implementation are identified, with the purpose to steer the focus on more technology-specific threats further into the process. This will also help to determine the most appropriate and accurate mitigation methods.

Decompose the product

In this stage of the process, a breakdown of the product architecture is performed with the purpose of creating a security profile based on common vulnerability areas ⁴. Entry points, trust boundaries and data flow are also identified in this step. The aim is to gain as much knowledge as possible about the product's mechanics in order to uncover threats.

Identify threats

In this step, a threat assessment based on found entry points (attack surfaces) is performed, where threats that could possibly affect the system and compromise the assets are identified ⁵. Different methods and/or models can be utilized for this, whereas one of them is the well known and established **STRIDE** model. The STRIDE model's goal is to ensure that important security directives, such as the CIA triad amongst others are met. The word is an acronym and stands for **Spoofing, Tampering, Repudiation, Information disclosure, Denial of Service** and **Elevation of privilege**. Each category is describe in the following way ⁶:

- **Spoofing** - *Spoofing* is a type of attack with the goal of gaining access to a system, which can be accomplished by using illegally obtained user credentials or a forged IP address.
- **Tampering** - *Tampering* is the act of modifying data maliciously, for example data that flows between two computers over a network or data stored in a database.
- **Repudiation** - *Repudiation* is an act associated with users who denies

⁴See footnote 2

⁵See footnote 2

⁶See footnote 2 and [https://docs.microsoft.com/en-us/previous-versions/commerce-server/ee823878\(v=cs.20\)?redirectedfrom=MSDN](https://docs.microsoft.com/en-us/previous-versions/commerce-server/ee823878(v=cs.20)?redirectedfrom=MSDN)

performing specific actions or transactions, without adequate auditing present to prove otherwise.

- **Information disclosure** - *Information disclosure* threats involves unwanted exposure of private data to individuals. An example of such scenario is when a user reads data in transit over a network or when reading a file that they are not authorized to open.
- **Denial of Service** - *Denial of Service* is a type of attack that aims to make a system unusable or unavailable. This is achieved by for example bombarding a server with multiple requests to the point where all available resources have been consumed.
- **Elevation of privilege** - *Elevation of privilege* is the act where an unprivileged user manages to elevate his/her privilege to a higher level with the purpose of gaining privilege access to a system.

Document threats

In the penultimate stage of the threat modeling, a documentation based on a set of significant attributes for each of the chosen threats is performed. The set of attributes in each threat document paper consists of: **threat description**, **threat target**, **attack techniques** and **countermeasures**. Furthermore, literature studies will be performed to identify possible weaknesses or other threats that should be taken into consideration.

Rate and select threats

A risk assessment is the final stage of the threat modeling process where all identified threats are rated base on the risks they pose⁷. This allows the developer to focus on the threats that present the most risk first and then continue with the rest. It can sometimes also be the case that threats that do not pose a great risk will be completely ignored since addressing all identified threats may not be economically or timely viable.

Different rating systems can be utilized when performing a risk assessment, whereas in this paper the **DREAD** model will be used⁸. DREAD is, similar to STRIDE, an acronym

⁷See footnote 2

⁸See footnote 2

that stands for **Damage potential**, **Reproducibility**, **Exploitability**, **Affected users** and **Discoverability**. The model uses a simple scoring scheme with three scales: low (1) risk, medium (2) risk and high (3) risk.

The risk rating of a threat is calculated by answering the following five questions⁹:

- **D:** "How great is the damage if the vulnerability is exploited?"
- **R:** "How easy is it to reproduce the attack?"
- **E:** "How easy is it to launch an attack?"
- **A:** "As a rough percentage, how many users are affected?"
- **D:** "How easy is it to find the vulnerability?"

The next step is to calculate the total score for each threat, where the highest result is 15 and lowest is 5. If the result ends up between 12-15, the threat is considered as a high risk, between 8-11 as a medium risk and 5-7 as a low risk. Test cases will be selected based on the threats with the highest scores and the information found during the literature study.

3.2 Penetration testing

In this phase, all the chosen threats from the previous section are tested. A standardized methodology for penetration testing, developed by NIST is utilized. It is divided into four stages: **Planning**, **Discovery**, **Attack** and **Report** [14, 23].

1. *Planning* - in this stage, it is decided on how the engagement is accomplished. This includes identification of rules and establishing of testing goals.
2. *Discovery* - this stage is partitioned into two sub-stages: 1) information gathering/scanning and 2) vulnerability analysis. However, the second phase is excluded. The first phase has partially already been conducted in the threat modeling stage. The difference is that the reconnaissance performed here hopefully uncover additional necessary information or information that might have been missed before, such as protocols or ports in use, utilized operating system etc.

⁹See footnote 2

3. *Attack* - the attack is the heart of the penetration test. Matching exploits for each vulnerability and suitable tools are found to perform such exploits. If a target is successfully compromised, the vulnerability is verified and mitigation techniques are identified.
4. *Report* - this phase occurs simultaneously as the other three phases. Examples of reports are logs and periodic reports written during the attack phase, but also a *threat traceability matrix* (see chapter 7, where results obtained from the penetration testing are documented).

Furthermore, the type of testing utilized is called **black box** [14], meaning that the assessment is performed with little to no prior knowledge about the targeted technology. Conducted penetration tests are described in chapter 6.

3.2.1 Tools & Environment

To test the system without affecting the application and other technology, an employee at Yanzi relocated the location that the product was associated with to a test environment. A real account was used but the location was fabricated. The penetration testing platform Kali Linux ¹⁰ was used during testing, together with multiple different tools, such as:

- **Hackrf One** ¹¹ - a Software Defined Radio peripheral, developed by Great Scott Gadgets, and is able to transmit and receive radio signals from 1 MHz to 6 GHz.
- **GNU Radio\GNU Radio Companion**¹² - GNU Radio is a software development toolkit used to implement software radios by using signal processing blocks. Supports development of software-defined radios with the help of SD hardware but also simulation-like environments, without any use of external hardware . GNU Radio Companion is GNU Radio's graphical user interface, used to build different flow graphs with the underlying GNU Radio components.
- **GQRX** ¹³ - a SDR receiver operated by the Qt graphical toolkit and the GNU Radio. SD hardwares such as HackRf, Funcube Dongles, rtl-sdr etc are all

¹⁰<https://www.kali.org>

¹¹<https://greatscottgadgets.com/hackrf/one/>

¹²<https://www.gnuradio.org/about/>

¹³<https://gqrx.dk>

supported by gqrx. Several valuable features are offered, including discovering of devices connected to the computer, FFT and waterfall plots, a spectrum analyzer, processing of I/Q data from devices that are supported, streaming of audio output over UDP, changing of frequency etc.

- **Wireshark** ¹⁴ - a network protocol analyzer that allows the user to keep track of all the actions that occur on the network at a microscopic level. It contains a rich feature set, where the user amongst other can capture live data and perform offline analysis as well as read live data from different network and communication protocols.
- **Nmap** ¹⁵ - a scanning tool utilized for security auditing as well as network and single host discovery. By using raw IP packets, nmap can discover and determine available hosts on a network together with associated characteristics, such as type of packet firewalls/filters used, running operating system (and version), available services that are offered by the host etc.
- **draw.io** ¹⁶ - an online tool for making process diagrams, ER, network diagrams, flowcharts etc .
- **Universal Radio Hacker** ¹⁷ - an open source software tool for wireless protocol investigation . It allows easy demodulation of signals and identification of bits and bytes that fly over the air, offers customizable decodings of decoded data, as well as features for protocol reverse-engineering and a fuzzing component.[24]

3.3 Evaluation

An evaluation of the results (see chapter 7) obtained from the penetration testing and the pre-study is conducted and later utilized to draw a conclusion regarding the security of the Yanzi IoT network. See chapter 8 for a detailed evaluation and chapter 9 for the conclusion.

¹⁴<https://www.wireshark.org>

¹⁵<https://nmap.org>

¹⁶<https://drawio-app.com/product/>

¹⁷All source code can be downloaded from <https://github.com/jopohl/urh>

Chapter 4

Yanzi IoT system

This chapter introduces the system under consideration. Information about supported features, hardware and overall functionality as well as technologies, such as protocols, in use are presented.

4.1 Yanzi System

The IoT platform delivered by Yanzi Networks is a fast, all-IP and end-to-end solution, consisting of gateways, sensors and a cloud platform named Cirrus ¹. Cirrus is bundled into a SaaS solution, called Yanzi Lifecycle. The application functions as a management tool for gateways/access points and sensors in smart buildings, and is provided to real estate owners and facility management companies who wish to enter the smart building industry.

The communication between a sensor and gateway is encrypted, regardless if it is wired or wireless ². An individual key is associated with each sensor and it is possible to revoke the key remotely. Besides from communicating with sensors, gateways also communicate and send data to the Cirrus cloud platform. Links between the cloud and a gateway are authenticated with client/server certificates, whilst communication is made using SSL encryption.

All IP technology is cost-efficient by leveraging on energy-saving protocols such as Ethernet, 4G/LTE, Wi-Fi and IEEE 802.15.4, where the latter allows for a long battery

¹<https://www.yanzi.se/pdf/product-brief/product-brief-890-07065-cirrus-iot-solution-smart-building-ibm-v01pa3.pdf>

²See footnote 1

lifetime, resulting in a low maintenance time ³.

4.1.1 Yanzi Cirrus Software as a Service

The Cirrus data model enables access to any data that originates from an unspecified location with a uniform interface ⁴. No sensor details need to be understood and are therefore irrelevant. Furthermore, reliable remote access to the gateway is provided by a secure Gateway-to-Cloud connector to various redundant link servers.

Cirrus Open API

Cirrus API⁵ is a secure open API that utilizes communication-based on JSON over web socket, with a connection that is both authenticated and SSL encrypted. Both hot (live data) and cold (historical data) paths are supported by the API. For hot paths, integration to 3rd party IoT frameworks is simplified with the support of Publish/Subscribe mechanism whereas, for cold paths, data recovery after a connectivity loss is enabled with the support of Request/Response mechanisms.

Yanzi Lifecycle

The **Yanzi Lifecycle**⁶ cloud service allows a user to maintain the whole Yazi infrastructure and connected sensors during the entire lifecycle. Maintenance includes monitoring of a system, changes, redeployments and new installations.

There are three levels in the permission and access hierarchy: Users, Groups and Locations⁷. The first level, **User**, is the term for someone that signs in to the Lifecycle. The table 4.1.1 demonstrates the different roles and levels of access a User can have to one or more Groups.

- **Read data** - allows a user with reading access to a Group to view data related to the sensors and aggregated Locations connectivity within the group.
- **Set outputs** - allows a user to interact with device functionalities, such as toggling of a Yanzi Plug.

³See footnote 1

⁴See footnote 1

⁵<https://www.yanzi.se>

⁶<https://yanzi.dev/#/manual/index?id=introduction>

⁷See footnote 6

Level	Managing users	Configure system	Install sensors	Set outputs	Read data
Admin	Yes	Yes	Yes	Yes	Yes
Manager	No	Yes	Yes	Yes	Yes
Installer	No	No	Yes	Yes	Yes
Writer	No	No	No	Yes	Yes
Reader	No	No	No	No	Yes

Table 4.1.1: Access rights for different User roles

- **Install sensors** - allows a user to install gateways and new sensors to the network.
- **Configure system** - allows a user to modify system configuration, such as creating and removing assets and uploading floorplans.
- **Manage users** - allows a user to modify (add, change and remove) access rights for all users that belong to a particular group.

The second level, **Groups**, assembles Locations and can usually be an approach to divide customers ⁸. The relationship between a Group and a User is that a User can have different rights to different Groups. Furthermore, whenever a User registers a new gateway, that User will also assign it to a Group.

Locations, representing the last level, is automatically created when a gateway is registered to a Group by a User ⁹. This implies that each distinct gateway in the system is represented by a Location, which can be identified with a unique Location ID.

4.1.2 Gateways

Yanzi IoT Gateways are the bridge between the Smart Building and the cloud in terms of communication, using either Ethernet or Wi-fi (IEEE 802.11b/g/n) ¹⁰. Communication is initiated by the gateway, which will open a TCP/IP connection to any IP over port 443 and/or 4445 ¹¹. The gateway is provided with an IP address from a DHCP server that also returns a valid DNS server. If the Internet connection is ever down, the gateway will ensure autonomous operations, sensor data and important local storage included. The gateways can be combined with Yanzi Access Points for larger

⁸See footnote 6

⁹See footnote 6

¹⁰See footnote 1

¹¹<https://yanzi.dev//#/manual/index?id=introduction>

installations that scale to thousands of IP-based sensors, including benefits such as fast installation, secure supplying of real-time data in Cirrus, and a secure network that consists of all-IP sensors in facilities.

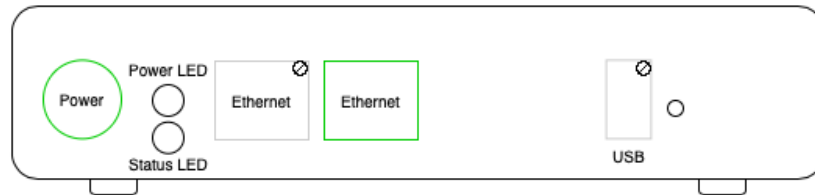


Figure 4.1.1: Overview of gateway ports

The **Yanzi IoT Gateway 2**¹² is a cost-effective gateway solution for small to medium installations in commercial facilities and buildings. The gateway communicates (using the LR-WPAN IEEE 802.15.4 standard) directly (or via a Yanzi Access Point 2) with the sensors due to its built-in radio and does that periodically to verify their presence and status. All communication is performed over a secure encrypted connection. Furthermore, the Yanzi Gateway can operate hours after a power failure has occurred due to its built-in battery back-up and it connects to the Internet either by using an existing connection or by using a built-in 4G when available.

4.1.3 Sensors

All Yanzi sensors are multi-sensor solutions with a twofold communication and are upgradable from the Cirrus cloud over-the-air¹³. Each sensor uses the low-rate wireless personal area network (LR-WPAN) IEEE 802.15.4 for communication, enabling the sensors which are operated by batteries to last for more than 10 years. Communication between a sensor and gateway is performed every 1 to 10 minutes and each sensor can be connected either directly to the gateway or via Access Points if there are larger installations.

The **Yanzi Presence Mini**¹⁴ sensor is utilized to monitor occupancy in an office space, using motion detection and temperature monitoring. All registered data, such as occupancy and temperature, is available over the Cirrus API and it takes about 200-500 ms after a motion event is detected for it to be provided at the API. A secure

¹²<https://yanzi.dev/manual/product-briefs/product-brief-890-07076-gateway-ap-2-smart-building.pdf>

¹³See footnote 1

¹⁴<https://yanzi.dev/manual/product-briefs/product-brief-890-07075-presence-mini-smart-building.pdf>

connection between the Yanzi Presence Mini and the Yanzi IoT gateway is performed through Yanzi's "Zero-Touch configuration". Communication between the two of them is wireless, using the LR-WPAN IEEE 802.15.4 standard, the IPv6 internet protocol, over 2.4GHz radio frequencies. The security key management is automatic and uses ECC with AES encrypted communication.

The **Yanzi Comfort**¹⁵ sensor is utilized to monitor the air condition indoors. It measures the levels of volatile organic compounds (VOC) and carbon dioxide (CO₂), but also humidity, barometric pressure, temperature and ambient noise. Security and communication between the sensor and the gateway is identical to the one described for the Presence Mini sensor. However, unlike the Presence Mini sensor, the Yanzi Comfort sensor is not operated by batteries. Instead, it receives its power from a USB wall outlet. Another difference is that it also performs as a mesh node for other sensors in the network.

4.1.4 Yanzi security overview

Channels from open cloud API to cloud and from cloud to the gateway is TLS encrypted and authenticated by SSL client/server certificates¹⁶. ECC is utilized for automatic key management between sensor and gateway, and AES application-layer encryption for user data and management data. This also includes firmware updates. Each device has an individual security key that can be revoked if needed. When applicable (Wi-Fi and IEEE 802.15.4), a layer 2 key distribution is used.

¹⁵<https://yanzi.dev/manual/product-briefs/product-brief-890-07066-comfort-smart-building.pdf>

¹⁶<https://yanzi.dev/#/manual/index?id=security>

Chapter 5

Planning

The following chapter covers the first phase of the methodology, where planning in form of threat modeling is performed. An architectural overview of the system, identified assets, entry points and threats etc. are presented here.

5.1 Threat model

5.1.1 Identified assets

5 assets were identified and are described in the following table:

ASSETS

- **Yanzi Gateway:** The device has two Ethernet ports, whereas solely one is in use, and the device, therefore, connects to the local network through an Ethernet cable.
- **Yanzi Sensors (Comfort and Presence):** Each sensor contains several smaller sensors that measure temperature, motion etc. depending on the model of the sensor.
- **Web application:** The Yanzi cloud SaaS platform, where live data sent by the gateway can be viewed by a user. To enable this, the user is required to have a separate username and password and to be a member of a group that has access to a location associated with sensors and gateway.
- **Radio communication:** All traffic between sensors and gateway is transmitted over radio communication.
- **Firmware:** Yanzi Stamp firmware that includes automatic channel selection, automatic security setup, automatic over-the-air update etc.

5.1.2 Architecture overview

Use cases

Use case 1: Create an account

1. User press on "Sign up"
2. User fills in personal details such as email address, first name, last name and last name and then press "Next".
3. User receives an email.
4. User verifies email and then press the link in the received email to create a password.

Use case 2: Install sensor

1. User sign in with username and password.
2. User chose a location

3. User press on "Install" and then "Install sensors"
4. User press on the chosen sensor that
5. User connects sensor, either manually entering the DID or by scanning
6. User turn on sensor
7. User press the "Connect" button

Use case 3: Read information and live data about connected devices

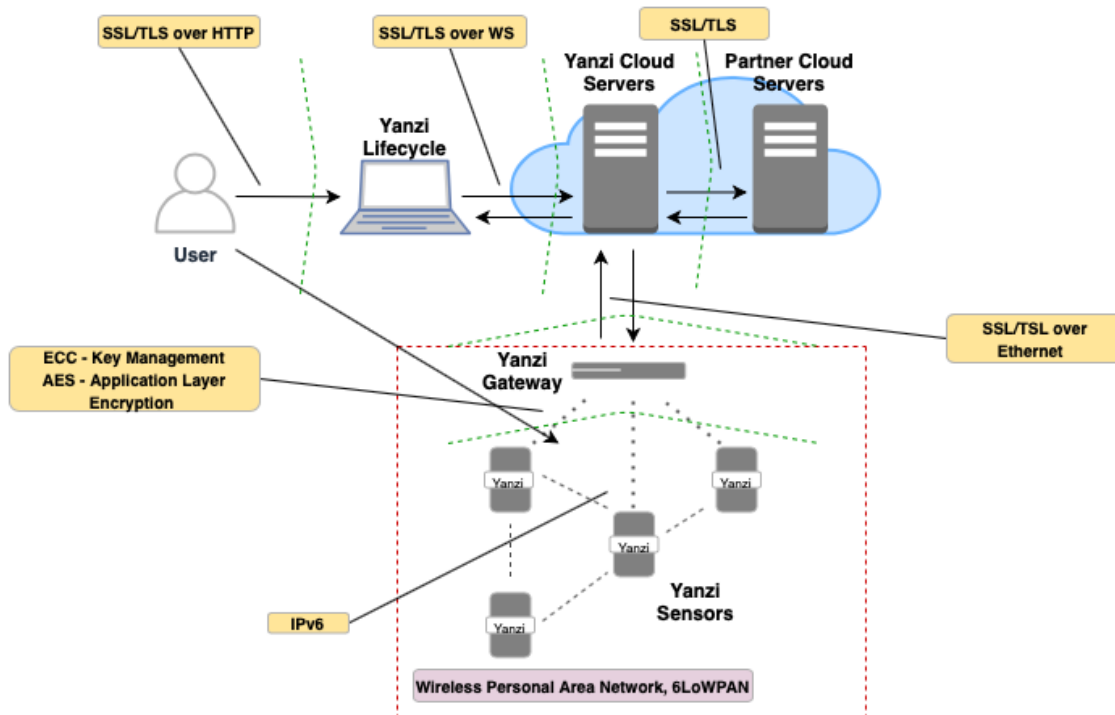
1. User sign in with username and password.
2. User press on the desired location.
3. User press on "List" under "Physical devices".
4. User press the arrow next to the device name for information about the device (except gateway).

Use case 4: Reset a sensor to factory default

1. User press and hold the status button
2. User waits 3 seconds for the LED to start blinking red rapidly
3. User waits 5 more seconds for the LED to start shift between red and green rapidly
4. User release status button
5. User waits a few seconds until the LED starts blinking green

Architecture diagram

An architecture diagram (see figure 5.1.1) was created in *draw.io*, which provides details of the components for the Yanzi Network ecosystem. It demonstrates utilized network and communication protocols together with implemented encryption methods, i.e IPv6 that transports data within the WPAN, where AES encryption is implemented on the application layer. SSL/TLS is implemented over the rest of the protocols that are in use, including WS and HTTP. It also showcases trust boundaries, and how data flows internally and externally within the system (appended after their discovering in the subsequent step, see section 5.1.3).



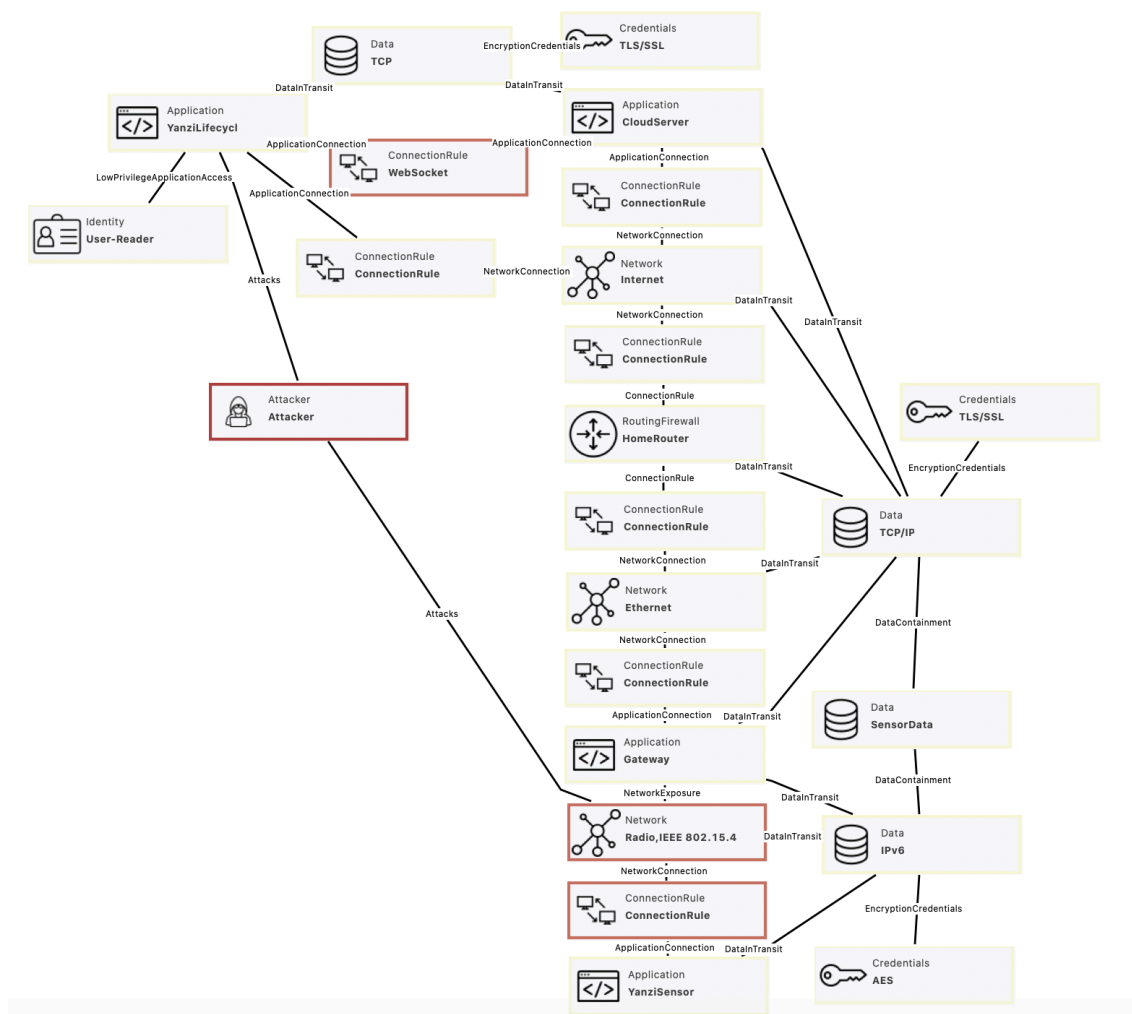


Figure 5.1.2: The Yanzi system modeled in SecuriCAD

Technology	Details
Yanzi Sensors	2.4 GHz, 5-25 m range, 50-100 m line of sight
Yanzi Gateway	Embedded Raspberry Pi, 2.4 GHz, communicates over Ethernet
Firmware: Yanzi Stamp	Firmware for sensor updates, security setup, automatic discovery etc
Communication protocol: HTTPS	Encrypted communication when user access the web application through the web browser
Communication protocol: 6LoWPAN (802.15.4)	RF protocol for communication between gateway and sensors
Communication protocol: IPv6	Network protocol for data delivery between sensors and gateway
Communication protocol: WSS	Encrypted communication between web application and Yanzi cloud servers
Communication protocol: Ethernet	Encrypted communication between gateway and servers
Encryption: AES	Encryption on the application layer
Encryption: SSL/TSL	Encryption over several various protocols
Encryption: ECC	Key management between sensors and gateway

Table 5.1.1: Identified technologies

Another entry point is the gateway itself, since it is connected to the internet and might have ports open that can be exploited. The third entry point is the wireless communication between sensors and gateway, which however is protected by AES encryption, and can therefore be difficult to attack. The last identified entry point is the Yanzi Stamp firmware, which controls several critical aspects of the product, and can therefore be misused if it falls into the wrong hands.

5.1.4 Identified threats

As mentioned in section 1.5, most of the entry points are excluded in this thesis due to time constraints and the nature of the scope. Threats were therefore only identified based on the included entry point.

Identified threats for radio communication according to the STRIDE model:

Spoofing - a sensor or gateway can be spoofed to deceive the other part into thinking that it communicates with an authenticated sensor/gateway.

Tampering - there is a possibility for tampering of data in packets that are sent

between sensors and gateway.

Information disclosure - data transmitted between a sensor and gateway can be captured, and if it is not encrypted, sensitive information can be identified. If data is encrypted, different techniques for decryption or decoding can be used to uncover information. There is also a possibility to find an open port on the gateway and exploit it for direct access to information that might be stored in it.

Denial of Service - there is a great possibility for a denial of service, where signals can be blocked or jammed.

5.1.5 Documented threats

The penultimate step of the threat modeling was to document all identified threats based on four parameters (see table 5.1.2 - 5.1.8), as mentioned in section 3.1.1.

Threat description	The attacker obtains the keys used for authentication and then uses one of the keys to impersonate a sensor/gateway
Target	Traffic, sensors, gateway
Attack techniques	The attacker can sniff/eavesdrop on traffic and then analyze the packages in Wireshark searching for the keys
Countermeasures	Encrypt all traffic that is transmitted

Table 5.1.2: Threat #1 documentation

In addition to the identified threats, [6] presents 11 possible weaknesses in radio protocols that were also used to decide what tests that should be conducted. The 11 weaknesses are the following:

- Sensitive data exposure
- Lack of transport encryption
- Interception and modification
- Man-In-The-Middle attack

Threat description	The attacker modifies either data payload or e.g the sequence number in the MAC packet
Target	Packets (data) in transmission
Attack techniques	The attacker can perform a MITM attack or just simply capture the traffic, and then use Wireshark to modify packets
Countermeasures	Some type of message integrity check or payload "checksum"

Table 5.1.3: Threat #2 documentation

Threat description	The attacker obtains sensitive information that is sent in clear text
Target	Data in packets
Attack techniques	The attacker captures traffic and analyzes the packets in Wireshark for any unencrypted information
Countermeasures	Encrypt all traffic that is transmitted

Table 5.1.4: Threat #3 documentation

- Replay attack
- Jamming attack
- Spoofing attack
- Denial of Service
- Lack of payload verification
- Lack of message integrity check
- Fuzzing custom protocols

Threat description	The attacker can decrypt encrypted data and obtain sensitive information
Target	Packets in transmission
Attack techniques	The attacker could with different possible techniques break the encryption algorithm
Countermeasures	If the attacker successfully breaks the encryption algorithm, it indicates that it is not strong enough and must be strengthened

Table 5.1.5: Threat #4 documentation

5.1.6 Rated and selected threats

The final step of the threat modeling was to evaluate and prioritize discovered threats and weaknesses. Each discovered threat was scored according to the DREAD model, see table 5.1.9.

After the scoring and evaluation of the possible weaknesses, the following was decided:

- Traffic will be captured to expose if there is a lack of transport encryption, which could result in sensitive data exposure, if for example encryption keys are transmitted in plain text
- A Denial of Service in the form of a Jamming attack will be performed
- Because a replay attack is relatively easy to perform, it will be feasible to test it. A successful attack indicates that there is a large flaw in the security.
- Traffic will be intercepted (MITM or sniffing) and information modified to verify if there is a lack of message integrity check as well as a lack of payload verification.

In this case, threat #6 is treated like a grey zone. It did not receive high enough scores to be classified as high risk, but will still be taken into consideration. The scoring of it is more complicated than the others, due to some unknown facts, such as if there are any ports that can be tested. Port scanning is part of the information gathering step (see section 6.2) in the penetration testing chapter, and will reveal if any ports are open or not. Naturally, testing will only be performed if there are any ports open.

Threat description	The attacker can decode signal data and obtain valuable information
Target	Signals
Attack techniques	The attacker can RF reverse engineer the signal by the use of URH to go from signals to meaningful bits, and then use different decoding techniques to find useful information
Countermeasures	Use standardized encryption algorithms on the whole packet and not only the payload

Table 5.1.6: Threat #5 documentation

Threat description	The attacker obtains possible saved information, such as encryption keys, in the gateway by accessing open ports
Target	Ports on the gateway
Attack techniques	Port scanning with Nmap to identify open ports that could be exploited
Countermeasures	Have ports closed as often as possible and if they must be open, implement a security mechanism to secure them

Table 5.1.7: Threat #6 documentation

Threat description	The attacker bombards the communication with packets/signals to threaten data availability
Target	Radio communication
Attack techniques	Perform a DoS in the form of constant jamming by sending out noise signals
Countermeasures	Frequency hopping and other assisting tools to measure, locate and identify RF signals

Table 5.1.8: Threat #7 documentation

Threat	D	R	E	A	D	Total
#1	3	1	2	2	2	10
#2	2	3	2	2	3	12
#3	2	3	3	2	3	13
#4	2	1	1	2	1	7
#5	2	1	1	1	1	6
#6	3	2	2	2	1	10
#7	3	3	3	2	3	14

Table 5.1.9: Threats evaluated according to DREAD

Chapter 6

Penetration testing

This chapter covers the second and largest phase of the methodology. Newly discovered information is presented, together with testing rules and goals, as well as conducted penetration tests.

6.1 Planning

A set of rules and one goal were established for a clearer structure and more accurate testing. One of the more relevant rules was that if an attack does not succeed, it should be performed at least ten to twenty more times before any conclusions could be drawn. Either with the same values or changed values, if possible. Another rule was that if an attack succeeds, it should be performed a few more times to be certain that it was not a coincidence (if that was a possibility). The goal was to obtain as much information as possible about the security of the communication to uncover possible flaws.

6.2 Discovery

Before any testing was conducted, essential information such as the precise frequency used for communication and the protocol used. It was also important to scan the gateway to discover if there are any open ports that can potentially be exploited.

Information gathering

The IP address used by the gateway was scanned to identify if there were any open ports that could be exploited. The commands *sudo nmap -sv <network address>/24*

and `sudo nmap -Pn <gateway address>` were both issued a number of times and gave different results nearly every time. At one point 586 ports were filtered, 411 closed and ports 21/ftp, 554/rtsp and 7070/realserver opened. Another time it was stated that 995 ports were closed and those ports that were filtered were displayed in the terminal with associated information. Other occasions resulted in all ports closed and/or filtered.

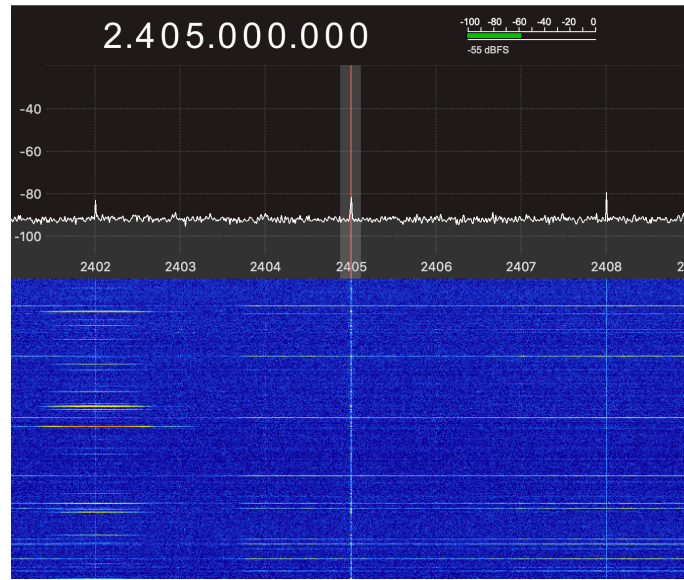


Figure 6.2.1: A graph in GQRX, showcasing a peak at the frequency 2.405 GHz

The documentation for the Yanzi devices does not disclose the precise frequency that sensors and gateway communicate on, it only states that they operate on the 2.4-2.4835 GHz band. There are 16 channels available for the 802.15.4 protocol, and it was, therefore, necessary to find out which of the channels is utilized. At first, the GQRX spectrum analyzer and the HackRF were used in an attempt to find a peak rising higher above the other peaks, e.g. when moving a sensor closer to the HackRF. This unfortunately did not yield any necessary information. However, after some consulting with an employee at Yanzi, it was disclosed that frequency band and channel can be found in the Yanzi lifecycle tool, under Network topology and the 2D map section, by clicking on one of the sensors for information. "Channel 2.1.11" is displayed, where 2.1 corresponds to the 2.4 GHz frequency band and 11 the 802.15.4 channel utilized in that band. With a quick search on the web, it is found that channel 11 corresponds to the 2.405 GHz frequency. To have this confirmed, GQRX was once again utilized. A peak was indeed found at 2.405 GHz, see figure 6.2.1.

The documentation of the Yanzi IoT states that IPv6 and IEEE 802.14.5 are used, which

indicates that the 6LoWPAN protocol and an adaption layer are present. However, since it is not clearly stated, it was not completely safe to conclude that 6LoWPAN is used. The initial idea was to capture the traffic with Wireshark, which would clearly display the protocol that is utilized. However, problems were encountered and it was, therefore, necessary to obtain this information from a Yanzi employee instead.

6.3 Testing

This section presents the process of each performed attack. During the testing process, a few problems were encountered, which complicated and ruined some of the test cases. This is described in section 6.3.1. Furthermore, because the Yanzi Presence Mini sensors transmit data to the gateway whenever they sense any motion in the room, the **Motion** parameter is, in contrast to other parameters, more regularly updated (see figure 6.3.1). For this reason and that the motion sensor is the easiest to trigger, the motion data was used for testing and validation.





Unit name	Status	Moti...	Tem...	Hum...	CO ₂ ...	VOC...	Sou...	Pres...
 Gateway 63A6	Up							
>  Area Comfort 39F8	Up		24	43	836	352	51	998
▼  Area Presence 804C	Up	2h	23					
Motion 804C-4 inputMotion		<u>Latest motion today at 1:03 PM.</u>						
Temp 804C-3 temp		23.0 °C						
▼  Merkurius Presence 7525	Up	2m	23					
Motion 7525-4 inputMotion		<u>Latest motion today at 3:17 PM.</u>						
Temp 7525-3 temp		23.0 °C						

Figure 6.3.1: A list of connected devices in Yanzi Lifecycle. Latest motion sensed is circled and underlined

6.3.1 Examining of packets

Most of the pen test cases in this thesis include some type of MiTM attack, either passive or active. The initial intention and the very first step was therefore to capture traffic, either live or into a file, and then analyze it using Wireshark. By achieving this, it would have been possible to examine packets sent between sensors and gateway,

and perhaps an altering of data. Examining packets would disclose if there is a lack of transport encryption or not, which could risk sensitive information in clear text to be extracted. A capturing of the key-exchange between sensors and gateway could have potentially resulted in a successful spoofing attack, while modification of information could have possibly resulted in a successful replay attack. In the case of the latter, it would have indicated that there could be a lack of message integrity check.

In the first attempt to capture traffic, HackRF One and GNU Radio were utilized. In section 6.3.2, it is demonstrated how traffic without any issues is captured into a file. The problem was therefore not the capturing of traffic into a file, but how to bridge the gap between GNU Radio and Wireshark. After some searching on the web, an open-source SDR-based IEEE 802.15.4 transceiver testbed for GNU Radio was discovered¹, see figure 6.3.2. With it, traffic can be logged by the SDR in PCAP format, which is a format compatible with Wireshark [25]. The Wireshark connector is compatible with Wi-Fi and Zigbee, and since the latter is based on identical MAC and PHY layers as 6LoWPAN, it was assumed that it would work. However, it did not and after several tries, this solution was abandoned. Error messages such as "MAC: frame too short. Dropping!" was constantly written out in the terminal, causing the packets to be marked as "malformed" in Wireshark and if no such message was displayed, the file was completely empty instead. The remaining attempts included Yanzi's own USB serial radio, named Yanzi Mesh. It was tested several times with both Sensniff² and Sparrow³, both open source, without success. The latter was used by Yanzi a few years ago to sniff traffic while testing the product, but is today unfortunately not compatible with the radio due to updates on the radio.

6.3.2 Replay attack

As mentioned before, a reply attack does not require any knowledge about encryption or capturing of encryption keys to succeeding, and is thus a relatively easy attack. However, simply capturing a signal and then re-sending it did not work in this case. The attack was performed at least 30 times, using GNU Radio Companion and the HackRF One.

First, the flowgraph in figure 6.3.3 was executed, and then the motion sensors were

¹All source code can be downloaded from <https://github.com/bastibl/gr-ieee802-15-4>

²All source code can be downloaded from <https://github.com/g-oikonomou/sensniff>

³All source code can be downloaded from <https://github.com/sics-iot/sparrow/wiki>

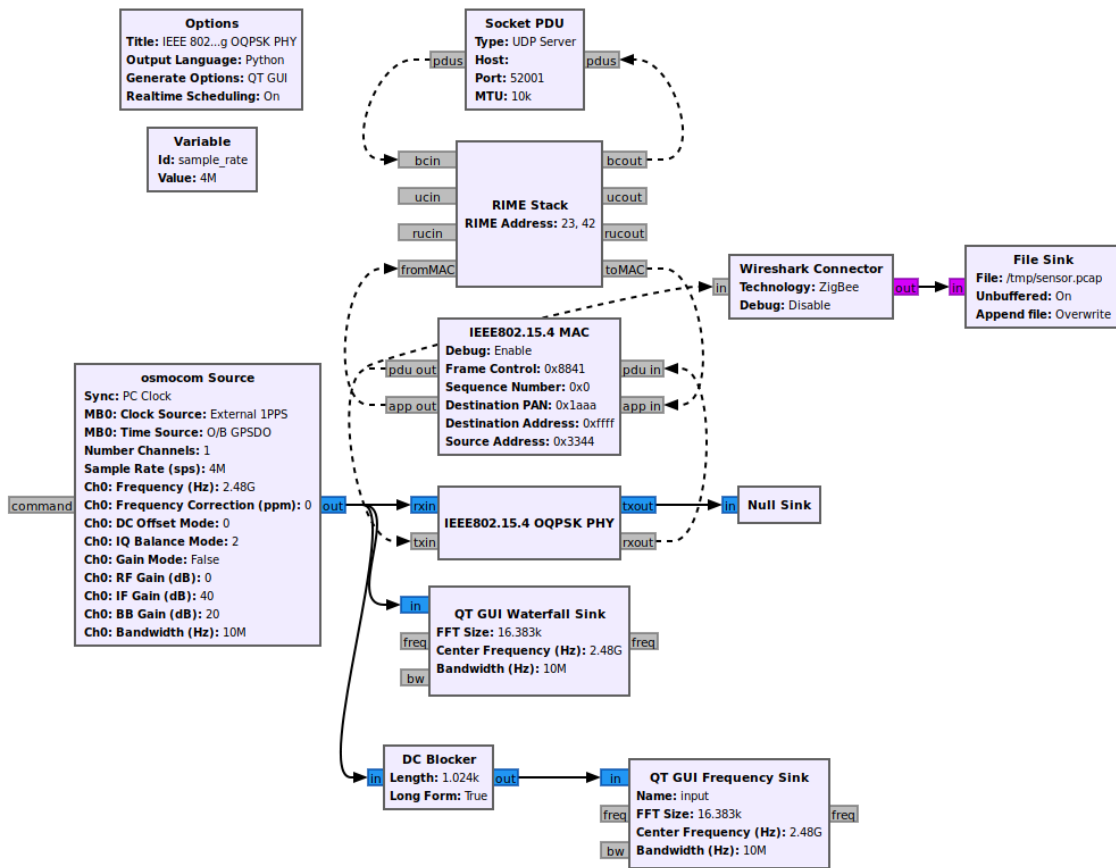


Figure 6.3.2: The layered structure of the SDR transceiver in GNU Radio Companion

triggered. All traffic on the 2.405 GHz frequency was captured and saved into a file. The third step was to retransmit captured signals, which is demonstrated in figure 6.3.4. The **Motion** parameter in the Yanzi Lifecycle was monitored at the same time to see if the value would be updated, without actually triggering the sensors.

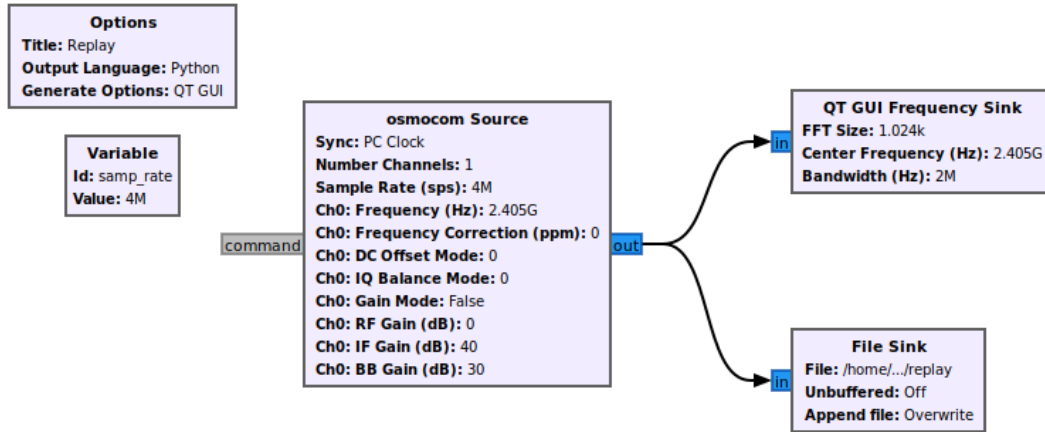


Figure 6.3.3: GNU Radio Companion flowgraph of the replay attack where traffic is captured

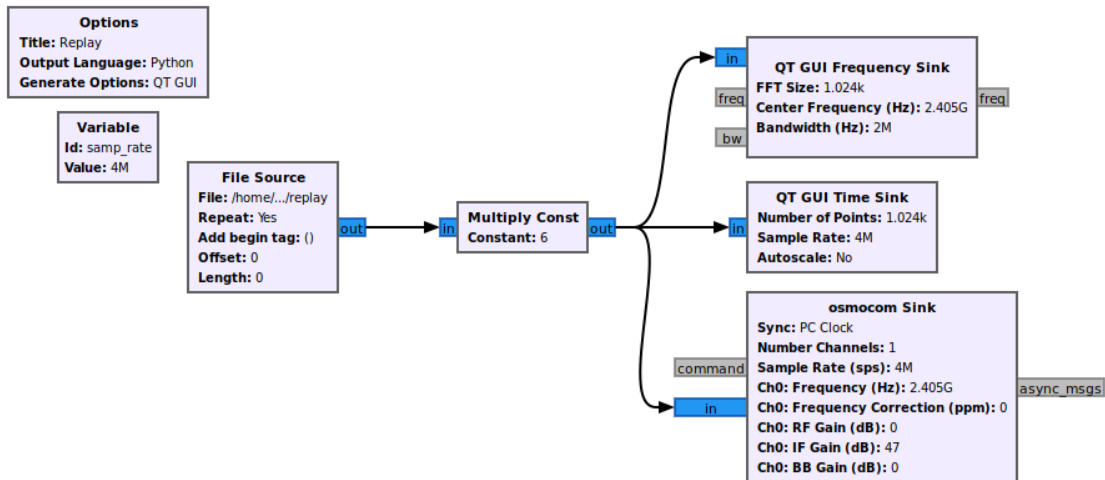


Figure 6.3.4: GNU Radio Companion flowgraph of the replay attack where captured traffic is retransmitted

Table 6.3.1 showcase all parameter values that were used in the attack. In all attempts, the sample rate parameter was set to $4e6$ since the bandwidth of each channel is $2e6$, and according to the Nyquist sampling criterion, the sampling rate should be at least twice the maximum frequency of interest. The RF Gain was set to zero as usual, and the BB Gain parameter in the sink block was also set to zero due to its irrelevance when signals are transmitted. The constant multiplier amplifies the signal in the

digital domain before conforming it to the analog domain, while the IF Gain value increases the transmitting power the larger it is. Values were chosen randomly for both parameters, however for the latter, the maximum possible value that will have an effect on the power is 47. Anything above that will be considered as 47.

IF Gain dB (source)	BB Gain dB (source)	IF Gain dB (sink)	Sample rate (samples/sec)	Constant multiplier
16	16	25	4e6	5
16	16	28	4e6	5
16	16	32	4e6	5
16	16	37	4e6	5
16	16	40	4e6	6
16	16	47	4e6	6
16	16	47	4e6	8
16	16	47	4e6	10
16	16	47	4e6	12
16	16	47	4e6	30
40	62	20	4e6	5
40	62	25	4e6	6
40	62	30	4e6	6
40	62	38	4e6	7
40	62	47	4e6	7
40	30	20	4e6	5
40	30	25	4e6	6
40	30	30	4e6	7
40	30	47	4e6	10
40	30	47	4e6	30

Table 6.3.1: Parameter values used in the attack

To be certain that the IoT is indeed secure from this type of replay attack, it was also performed using the Universal Radio Hacker together with the HackRF One, see figure 6.3.5. The sensors were triggered, and the signals were then captured and saved into a file. The sample rate was set to the same value as the previous attempts and the bandwidth to 2e6. The IF and BB Gain parameters when capturing signals were set to 40 and 16 respectively, and the IF Gain parameter when retransmitting signals to 47. The attack was performed five times.

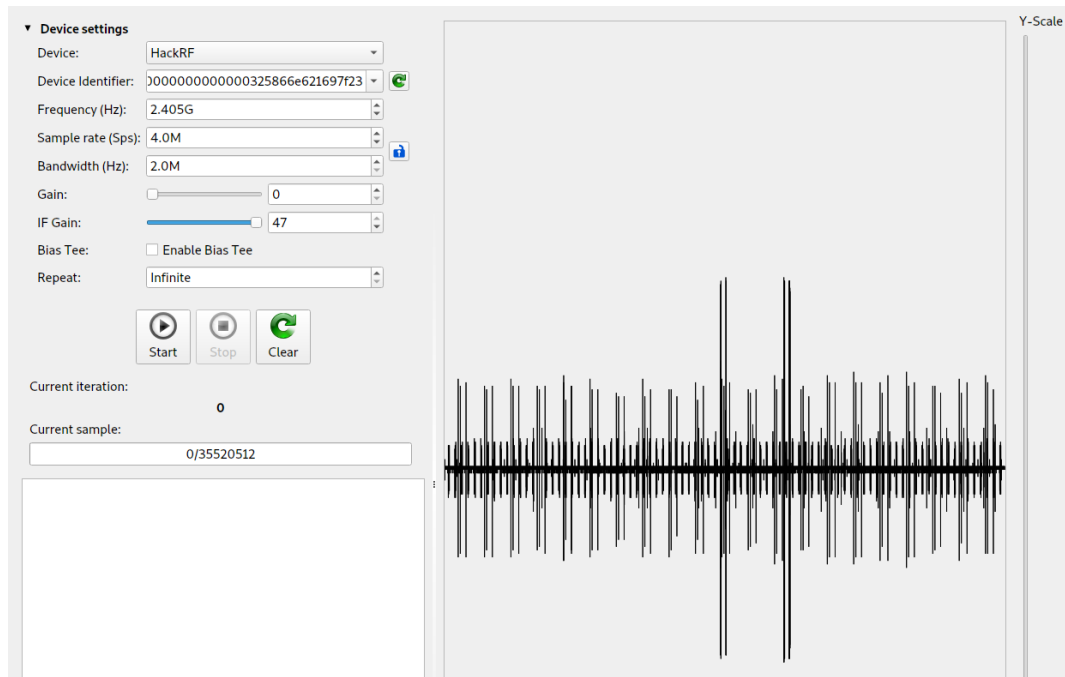


Figure 6.3.5: Captured signals before they are replayed showcased in Universal Radio Hacker

6.3.3 Denial of service

Jamming attack

The jamming attack was performed by repeatedly transmitting (constant jamming) signals in the form of cosine waves over 2.405 GHz, using GNU Radio Companion and the HackRF One, with the aim to block/jam any signals that are sent from the sensors to the gateway. IF Gain was set to 47, which is the maximum possible value for that parameter, while RF Gain as often was set to zero and BB Gain to the default value since it is irrelevant when signals are transmitted. The first try resulted in a great success, where signals were completely blocked. This was verified by monitoring the **Motion** parameter while several attempts to trigger the motion sensor were performed. Before the attack was conducted, it was observed that on a few occasions there were around 10-30 seconds of delay in the update of the motion data in Yanzi Lifecycle whenever a sensor was triggered. Therefore, to be certain that the signals were indeed blocked and not solely a late update, signals were transmitted for at least three minutes while sensors were triggered.

The assigned values to each parameter is displayed in figure 6.3.6. In the replay attack, it was stated that the sample rate should be at least 2x the bandwidth, which was not followed in this case, as the figure showcase. The reason is that nothing is captured

here, only transmitted. Therefore, the least recommended sample rate value was selected to start with, and then if the attack would not succeed, the value would've been set to 4e6 instead. However, as the results demonstrate, the attack succeeded with the sample rate set to only 2e6.

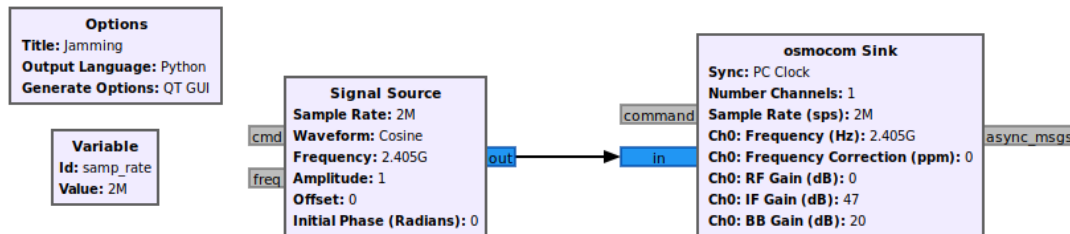


Figure 6.3.6: GNU Radio Companion flowgraph of the jamming attack

To be completely certain that the attack succeeded and was not a coincidence, it was performed five additional times (with the same parameter values). The second time was also successful, with no issues or differences from the first attempt. However, the third time one of the sensors lost connection and was marked as "Down" in the Yanzi Lifecycle. It then connected again by itself, but did not sense any motion when it should. It was therefore necessary to reset it. The fourth and fifth times also succeeded without any issues or noteworthy changes. The sixth attempt was decided upon a few days after the other attempts were performed. The motivation for this was that it would be interesting to see how the sensors and the gateway would react if signals were jammed for more than five minutes. As a result, all three sensors lost connection and were marked as "Down" in the Yanzi Lifecycle. A few minutes after the jamming of signals was stopped, one sensor after the other connected and was marked as "Up" again.

At a certain point in time, the whole system was relocated from channel 11 to 26, and to make sure that the attack would be possible on this channel also, it was performed an additional time. Two sensors were used in this test case, one Presence Mini and one Comfort. As it was suspected, jamming of signals was possible on this channel as well. As soon as noise signals were transmitted, Yanzi Lifecycle stopped updating the motion parameter and after a few minutes, both sensors were marked as "Down".

Protection against jamming

Since the performed jamming attack targets the PHY layer of the protocol, anti-jamming techniques will be presented based on that. One often used anti-jamming technique is Frequency hopping (FH), which refers to the act where the carrier frequency of a radio signal changes regularly [26]. In other words, hop in frequency between different bands. In this way, the signal is spread over a wider band, which hides the signal from both intentional and unintentional interference. There exists different FH techniques that can be utilized, including Frequency-Hopping Spread Spectrum (FHSS) and Direct Sequence Spread Spectrum (DSSS). If any such anti-jamming technique is implemented, to successfully jam the signals in transmission, the jammer would have to either deduce or guess the DSSS sequence or the FHSS pattern [27]. Therefore, depending on the computational ability of the jammer, it can only interfere with signals in transmission with a certain probability.

Other more general countermeasures include RF detection, spectrum analysis and direction finding equipment [28]. RF signals that may interfere and affect communication can be identified by RF detection devices, and the frequency and received strength of these signals can then be determined by spectrum analysis devices. Such devices can furthermore be utilized to recognize frequencies free of interference, which is beneficial when RF jamming mitigation tactics such as FH are used. Lastly, RF direction finding equipment can assist in locating the sources that transmits disrupting RF signals, by measuring and triangulate the direction from which the signals were transmitted.

Chapter 7

Results

This chapter presents findings from the testing in section 6. A summarize of the results are presented in a traceability matrix, see figure 7.0.1. Unfortunately, due to the technical problems which were encountered during testing, some results are not applicable (N/A).

Surface	Component	Weakness	Attack	Impact	Severity	Result
Radio	PCAP (Data in packets)	Sensitive data exposure	Sniffing/Eavesdropping	Access to sensitive information	N/A	N/A
Radio	PCAP (Data in packets)	Lack of transport encryption	Sniffing/Eavesdropping	Access to information	N/A	N/A
Radio	Radio (Packets (Data) in transmissio	Interception and modification	Interception and modification	Data modification	N/A	N/A
Radio	Radio (Packets in transmission)	Man in the middle attack	Man in the middle attack	Access to information	N/A	N/A
Radio	Radio	Replay attack	Capture signal and resend	Forged values	Low	Fail
Radio	Radio	Denial of service (DoS)	Constant jamming	Compromised data availability	High	Success
Radio	PCAP	Lack of payload verification	Interception and modification	Data modification, forged values	N/A	N/A
Radio	PCAP	Lack of message integrity check	Interception and modification	Data modification	N/A	N/A

Figure 7.0.1: Threat traceability matrix with obtained results.

Chapter 8

Discussion

Because of the AES application layer encryption and the use of ECC key exchange, the expectations on the security of the product were quite high. The first performed attack, where traffic was captured and then replayed, did not succeed, which is a good sign considering that this type of replay attack often is easy to perform. There are several reasons to why it did not work, including the application encryption (?), timestamps on each packet and/or the protocol containing a sequence number, which prevents packets from being accepted more than once. However, there is a possibility that other types of replay attacks would succeed, since they were not conducted in this thesis. One would be to target a specific packet, in such way that it is first jammed, then captured and finally replayed. Another would be to capture the packet, change i.e the sequence number and then replay it.

As the results displayed, it was possible to perform a jamming attack where signals (noise) were constantly transmitted over a chosen target channel (11 and 26 in this particular case). Unrelated to the product itself, but also worth mentioning is that it would probably also be possible to target all the channels at the same time (Wide-Band Denial) rather than solely a single channel, and in that way block signals on the entire 2.4 GHz band. Performing a jamming attack of this type was rather easy, and it is therefore not completely wrong to assume that a novice within hacking would be able to perform it if the right tools are used. HackRF One can be purchased by whomever interested in radio, and GNU Radio is a free open source radio ecosystem, available for everyone. Finding the correct frequency that is used is usually not difficult, and often solely requires a spectrum analyzer. However, finding the right

frequency, in this case, was not that straightforward since several other protocols share these channels, but also because no clear peaks could be seen whenever a sensor was triggered. Instead, the frequency was identified through the Yanzi Lifecycle, which is only possible if the hacker has an account and access to a certain location that is associated with the Yanzi system. Nevertheless, the attack uncovered a flaw in the system. By continuously transmitting signals for a short period of time, the gateway could not distinguish between real signals and noise signals, resulting in no update of the motion parameter and a lost connection for all sensors. It is without a doubt a threat to data availability.

Often times, it is desired to mitigate all types of interference in a wireless communication. However, it is important to differentiate between intentional and unintentional interference. The latter refers to signals that are typically transmitted by other electronic devices that operates on the same radio frequency, while the first refers to deliberate act of jamming. Naturally, the unintentional interference is a greater threat, and must be therefore be mitigated. FH is a great solution that is often used for this purpose. However, one main disadvantage is that FH solutions require a much larger bandwidth than the original signal. Furthermore, the frequency hopping rate must be quite rapid, in order to avoid that the jamming power at a certain frequency reaches the receiver before the radio signal shifts frequency. Another disadvantage could be that the attacker targets the whole frequency band instead of solely one channel, which would then defeat the whole purpose of FH. As for the other solutions, there are both advantages and disadvantages. Because the spectrum analysis devices can determine both frequency and received strength of interfering signals, this information can be utilized to determine the amount necessary to increase the strength of the communicating devices' signal in order for it to be well above the attacker's signal strength. The spectrum analysis devices would also be useful in combination with FH, since they can recognize frequencies free from interference, , which is information that could be used to select the channel with least traffic. Additionally, locating sources that transmits intentional interference signals enables the possibility to implement direct mitigation strategies to overcome the effects of such interference. Nevertheless, despite the usefulness of these solutions, it should be mentioned that they can be quite expensive. Here it is important to consider the importance and impact of the found flaw, the urge to fix it as well as economic restrictions. Naturally, the optimum would be to find more inexpensive alternatives that functions at least as well.

With the amount of time invested to be able to analyze traffic in Wireshark, it is unfortunate that it did not go as planned. 6LoWPAN appears to be a commonly used protocol, but it was noticed that it is not taken into consideration as much as other wireless protocols, such as Wi-Fi and Zigbee, which was indeed very time-consuming and frustrating. However, since this IoT is not alone utilizing this protocol when communicating, there probably exists other solutions that others have used when testing their product. The Sewio Open Sniffer is an example of such a solution, which would have been the optimal option if there was more time. The people at Yanzi also recommended another sniffer that they have used, called Zolteria Firefly, which is supported by both Contiki-NG and Sparrow. However, these and other alternatives were order items, and would either take time to arrive or were out of stock. Moreover, there is no guarantee that they would work.

Chapter 9

Conclusion

It was possible to perform a jamming attack, which exposed a flaw in the system that certainly must be taken care of. A few solutions were presented, but an advantageous start would be to begin with a more simple and less expensive solution at first, such as frequency hopping, and then perhaps advance if necessary. Furthermore, it is difficult to say whether the IoT product is secure or not due to the limited testing, but the successfulness of the jamming attack indicates that it is not as secure as it should be. Finally, this thesis creates a possibility for future research and experiments based on documented findings, where a more comprehensive evaluation can be performed. There is a great potential for testing of attack surfaces/entry points that were not a part of this study, which could uncover other vulnerabilities in the system. This also applies to the tests that were part of the study, but which could not be performed. Different technologies, both used and not used, are mentioned in the report that can be used in future testing.

Bibliography

- [1] Patil, Sonali et al. “Ethical hacking: The need for cyber security”. In: *2017 IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI)*. IEEE. 2017, pp. 1602–1606.
- [2] Borgohain, Tuhin, Kumar, Uday, and Sanyal, Sugata. “Survey of security and privacy issues of internet of things”. In: *arXiv preprint arXiv:1501.02211* (2015).
- [3] Yang, Yuchen et al. “A survey on security and privacy issues in Internet-of-Things”. In: *IEEE Internet of Things Journal* 4.5 (2017), pp. 1250–1258.
- [4] Aqeel-ur-Rehman, Sadiq Ur Rehman et al. “Security and privacy issues in IoT”. In: *International Journal of Communication Networks and Information Security (IJCNIS)* 8.3 (2016), pp. 147–157.
- [5] Kolias, Constantinos et al. “DDoS in the IoT: Mirai and other botnets”. In: *Computer* 50.7 (2017), pp. 80–84.
- [6] Süren, E., Heiding, F., and Lagerström, R. “PATRIoT: A systematic and agile vulnerability research process for IoT”. Unpublished paper. 2021.
- [7] Guzman, Aaron and Gupta, Aditya. *IoT Penetration Testing Cookbook: Identify vulnerabilities and secure your smart devices*. Packt Publishing Ltd, 2017.
- [8] Mozumder, Deba Prasead et al. “Cloud computing security breaches and threats analysis”. In: *International Journal of Scientific & Engineering Research* 8.1 (2017), pp. 1287–1297.
- [9] Shelby, Zach and Bormann, Carsten. *6LoWPAN: The wireless embedded Internet*. Vol. 43. John Wiley & Sons, 2011.
- [10] Abbasi, Naveed A et al. “6LoWPAN: IPv6 for battery-less Building Networks”. In: *TU Eindhoven, Aug 31* (2009).

- [11] “IEEE Standard for Low-Rate Wireless Networks”. In: *IEEE Std 802.15.4-2020 (Revision of IEEE Std 802.15.4-2015)* (2020), pp. 1–800. DOI: 10 . 1109 / IEEESTD . 2020 . 9144691.
- [12] Amin, Yasmin M and Abdel-Hamid, Amr T. “A comprehensive taxonomy and analysis of IEEE 802.15. 4 attacks”. In: *Journal of Electrical and Computer Engineering* (2016).
- [13] Jaitly, Sunakshi, Malhotra, Harshit, and Bhushan, Bharat. “Security vulnerabilities and countermeasures against jamming attacks in Wireless Sensor Networks: A survey”. In: *2017 International Conference on Computer, Communications and Electronics (Comptelix)*. IEEE. 2017, pp. 559–564.
- [14] Baloch, Rafay. *Ethical hacking and penetration testing guide*. CRC Press, 2017.
- [15] Fang, Xianjin and Wu, Yanting. “Investigation into the elliptic curve cryptography”. In: *2017 3rd International Conference on Information Management (ICIM)*. IEEE. 2017, pp. 412–415.
- [16] Daemen, Joan and Rijmen, Vincent. “AES proposal: Rijndael”. In: (1999).
- [17] Satapathy, Ashutosh and Livingston, J. “A Comprehensive Survey on SSL/TLS and their Vulnerabilities”. In: *International Journal of Computer Applications* 153.5 (2016), pp. 31–38.
- [18] Johnson, Pontus, Lagerström, Robert, and Ekstedt, Mathias. “A meta language for threat modeling and attack simulations”. In: *Proceedings of the 13th International Conference on Availability, Reliability and Security*. 2018, pp. 1–8.
- [19] Katsikeas, Sotirios et al. “An attack simulation language for the it domain”. In: *International Workshop on Graphical Models for Security*. Springer. 2020, pp. 67–86.
- [20] Ekstedt, Mathias et al. “Securi cad by foreseeti: A cad tool for enterprise cyber security management”. In: *2015 IEEE 19th International Enterprise Distributed Object Computing Workshop*. IEEE. 2015, pp. 152–155.

- [21] Bravo-Montoya, Andrés F, Rondón-Sanabria, Jefersson S, and Gaona-García, Elvis E. “Development and Testing of a Real-Time LoRawan Sniffer Based on GNU-Radio”. In: *TecnoLógicas* 22.46 (2019), pp. 130–139.
- [22] Hung, Phan Duy and Vinh, Bui Trong. “Vulnerabilities in IoT devices with software-defined radio”. In: *2019 IEEE 4th International Conference on Computer and Communication Systems (ICCCS)*. IEEE. 2019, pp. 664–668.
- [23] Scarfone, Karen et al. “Technical guide to information security testing and assessment”. In: *NIST Special Publication 800.115* (2008), pp. 2–25.
- [24] Pohl, Johannes and Noack, Andreas. “Universal radio hacker: a suite for wireless protocol analysis”. In: *Proceedings of the 2017 Workshop on Internet of Things Security and Privacy*. 2017, pp. 59–60.
- [25] Bloessl, Bastian et al. “A GNU radio-based IEEE 802.15. 4 testbed”. In: *12. GI/ITG KuVS Fachgespräch Drahtlose Sensornetze (FGSN 2013)* (2013), pp. 37–40.
- [26] Feng, Zhutian and Hua, Cunqing. “Machine Learning-based RF jamming detection in wireless networks”. In: *2018 Third International Conference on Security of Smart Cities, Industrial Control System and Communications (SSIC)*. IEEE. 2018, pp. 1–6.
- [27] Lu, Zhuo, Wang, Wenye, and Wang, Cliff. “From jammer to gambler: Modeling and detection of jamming attacks against time-critical traffic”. In: *2011 Proceedings IEEE INFOCOM*. IEEE. 2011, pp. 1871–1879.
- [28] Shahid, Hasan. *Radio Frequency Detection, Spectrum Analysis, and Direction Finding Equipment: Market Survey Report*. Report. National Urban Security Technology Laboratory (NUSTL), 2019.

Appendix - Contents

A First Appendix	62
B Second Appendix	63

Appendix A

First Appendix

This is only slightly related to the rest of the report

Appendix B

Second Appendix

this is the information