Developing a Framework for Cybersecurity Evaluation in the Consumer Internet of Things

Abstract

Title	Developing a Framework for Cybersecurity Evaluation in the Consumer Internet of Things
Date	20.05.2022
Authors	Kristoffer Dahl Jan Olaf Storeng Jørgen Oliver Strøm Nikolai Svidal
Supervisor	Andrii Shalaginov, Associate professor in the Department of Information Security and Communication Technology
Employer	Eirik Stephansen, Telenor Norge AS, Tech Lead IoT-Mobile
Keywords	IoT, Security, Evaluation, Framework
Pages	82
Attachments	9
Availability	Open

The field of IoT consists of a wide variety of tools, technologies and methods and is constantly under development. Through sensors and usage of application, the device collects, processes, produces and transfers information that could be sensitive to the user. Securing such devices is proven to be a challenge as they are made to utilize a minimum amount of processing power. As a part of this thesis the group made a framework that depicts an approach of how to evaluate the general security of an IoT-device. Based on industry security standards from ETSI, and requirements from GDPR have developed a set of criteria that gives an indication to how secure the device is and what weaknesses the device possesses. Through an iterative process the group have tested and improved the framework by using it to evaluate commonly used IoT-devices. Evaluation for each criteria is built upon the process of risk management, where the risk is estimated as a sum of how the criteria has been implemented and how severe the consequence could be, upon misuse of weakness in criteria. Accompanying the framework are descriptions of what technical tools can be used to test and monitor the behavior of the devices. The end result of the project was a framework on how to evaluate the security of IoT-devices in regards to general security measurements and how sensitive and personal data is treated.

Sammendrag

Tittel	Utvikling av Rammeverk for Digital Sikkerhets Evaluering av Forbruker Internet of Things
Dato	20.05.2022
Deltakere	Kristoffer Dahl Jan Olaf Storeng Jørgen Oliver Strøm Nikolai Svidal
Veildeder	Andrii Shalaginov, Førsteamuensis ved fakultetet for informasjonsteknologi og elektroteknikk
Oppadragsgiver	Eirik Stephansen, Telenor Norge AS, Tech Lead IoT-Mobile
Stikkord	IoT, Sikkerhet, Evaluering, Rammeverk
Antall sider	82
Antall vedlegg	9
Publiseringsavtale inngått	Åpen

Fagfeltet IoT består av en mengde verktøy, teknologier og metoder som er under utvikling. Gjennom sensorer og bruk av applikasjoner, samler, prosesserer, genererer og forflytter enheten informasjon som kan være sensitiv ovenfor brukeren. Å sikre slike enheter har vist seg å være en utfordring ettersom de er bygget opp for å bruke minst mulig strøm og prosessorkraft. Som en del av denne oppgaven er det laget et rammeverk som beskriver en fremgangsmåte for hvordan sikkerheten av IoT-enheter kan evalueres. Basert på industristandarder fra blant annet ETSI og krav GDPR setter, har gruppen utviklet et sett av kriterier som vil kunne gi et inntrykk av sikkerheten til enhetene og hvilke sårbarheter de har. Gjennom en iterativ prosess er det utviklet og forbedret rammeverket ved å bruke og teste det opp mot IoT-enheter. Den endelige evalueringen er basert på risikostyrings-prosessen, hvor risiko blir satt som en sum av hvor godt kriteriet har blitt implementert og hvor alvorlig konsekvensen av satt kriterie kan bli. Satt sammen med rammeverket er beskrivelser av hvilke verktøy og metoder som er brukt for å monitorere og teste enhetene. Sluttresultatet av prosjektet er et rammeverk for hvordan man kan gå frem for å sikre enhetene i henold til den generelle sikkerheten og hvordan sensitiv og personlig data blir behandlet.

Preface

We would like to thank various people that have helped with the thesis, i.e. people that have proof read the thesis etc.

We would like to thank the people involved with our thesis for their help and support. The first person we would like to thank is Andrii Shalaginov, our supervisor for the thesis, for good resources, advises, as well as proofreading our thesis.

We would also like to thank our contact person within Telenor Norge AS, Eirik Stephansen, which provided us with the task and the tools necessary during our research. As well as good advises given throughout the thesis.

A big thanks to all the different professors and lecturers that have helped us during our Bacheleor's degree. A special mention goes too Erik Hjelmås that has coordinated the Bachelor's program, and for all the small advice's during our thesis. Others we would like to thank our fellow students for proofreading our thesis, and giving constructive feedback, and giving us a good study environment.

Thanks to Robert Colvin - Telenor Connexion AB and Mette Kristine Kanestrøm - Telenor Connexion AB for a great and informative presentation and tips.

Introduction of the employer

The task we are presenting in this bachelor thesis has been provided by Eirik Stephansen, Tech Lead of IoT-Mobile for Telenor Norge AS which is the largest telecommunication company in Norway.

Telenor is looking into the possibilities of IoT-devices and have therefore shown interest in IoT security and a method of evaluating such devices.

Contents

Figures

Tables

Glossary

best practice Best practice means that the technique used is superior to any alternative, or as no known weaknesses. It also does not have any indication of feasible attacks with current readily available techniques. , 1

black-box "Black-box testing, which is also known as functional testing, is the testing that ignores the internal mechanism of a system or component and focuses solely on the outputs generated in response to selected inputs and execution conditions" [1]. , 4

black-hat hacker "A black hat hacker is someone with objectives of studying and using cyber security techniques and tools for personal or private gain through malicious or threat activity" [2]. , 1

botnet "Botnet is a term for a collection of software robots, or bots, which run autonomously and automatically" [3]. , 1

controller A controller is the body determines the purposes and means of the processing of personal data [4].

cyberspace "A global domain within the information environment consisting of the interdependent network of information systems infrastructures including the Internet, telecommunications networks, computer systems, and embedded processors and controllers" [5]. , 2

DevOps DevOps is the combination of cultural philosophies, practices, and tools that increases an organization's ability to deliver applications and services at high velocity: evolving and improving products at a faster pace than organizations using traditional software development and infrastructure management processes. This speed enables organizations to better serve their customers and compete more effectively in the market. [6]. , 38

GDPR General Data Protection Regulation, an EU law regarding data protection and privacy [7].

IMRaD Introduction, Material and method, Result and Discussion [8]. It is a template for how to structure scientific articles.

IoT Internet of things, a common used phrase to describe the inclusion of internet, software, and other smart technology into physical things, like vacuum cleaners and light bulbs.

personal data "Personal data' means any information relating to an identified or identifiable natural person ('data subject')" [9]. , 1

personal sensitive data Data whose disclosure has a high potential to cause harm to the individual [10].

README A readme file – often created as readme.txt or readme.md – usually contains important information about the respective system, project or software. To ensure users can find the file straight away, it should ideally be placed in the top directory level [11]. , 38

reverse engineer To disassemble and examine or analyze in detail (a product or device) to discover the concepts involved in manufacture usually in order to produce something similar [12]. , 4

risk matrix A risk matrix is a commonly used method for evaluation of risk assessment. , 14

scoring system The scoring system is the system used in the framework for evaluating the devices weaknesses, and showing its improvements for the criteria.

sensitive data Sensitiv data is classified information that must be protected and is unauthorized for outside parties [13]. , 1

spoof "Spoofing attacks arise when the attacker is capable to root a user or a device on a system to believe that a piece of information came from a source from which it actually did not initiate" [14]. , 4

threat actor An individual or a group posing a threat [15]. , 12

Z-Wave A wireless technology that is primarily used with smart home products [16]. , 3

Zigbee Lightweight suite for IoT devices to be able to communicating across a mesh network [17]. , 3

Acronyms

AI Artificial Intelligence. , 10

CoAP Constrained Application Protocol. , 16

CoPCSE@NTNU Community of Practice in Computer ScienceEducation at NTNU. , 6

CVE Common Vulnerabilities and Exposures. , 73

CVSS Common Vulnerability Scoring System. , 73

DDoS Distributed Denial-of-Service. , 1

DUT Device under testing. , 27

GDPR General Data Protection Regulation. , 1

HTTP Hyper Text Transfer Protocol. , 4

IFTTT If-This-Then-That. , 79

IMRaD Introduction–Method–Results–and–Discussion. , 6

IoT The Internet of Things. , 1

IRC Internet Relay Chat. , 4

MITM Man-in-the-Middle. , 4

MQTT Message Queuing Telemetry Transport. , 16

NTNU Norwegian University of Science and Technology.

NTNU Norges teknisk-naturvitenskapelige universitet.

RTSP Real Time Streaming Protocol. , 4

SSN Semantic Sensor Network. , 16

SSW Semantic Sensor Web. , 16

UP Unified Process. , 18

XMPP Extensible Messaging and Presence Protocol. , 16

Chapter 1

Introduction

1.1 Problem

The Internet of Things (IoT) has throughout the decade taken a bigger part of the daily life. Smart homes, surveillance, medical equipment and vehicles are dependent on being connected to the internet to function properly. There are many benefits with the devices, everything from controlling temperature to monitoring your house can be done through an application on your phone. However the disadvantages and risk in using such devices are not as clear. Through the use of an IoT-device, the device make use of, share and communicate data that is collected through sensors and usage of applications. If done insecurely, personal and sensitive data might get in the hands of unwanted people.

Connected IoT-devices are rapidly increasing and it is estimated that there will be over 27 billion IoT connections within 2025 [18]. The scale and number of devices also makes the devices attractive for black-hat hackers, making them a part of a botnet. Vulnerable IoT-devices have been the primary source of the biggest Distributed Denial-of-Service (DDoS) attacks such as Mirai [19]. These types of attacks are growing in severity, occurring more frequently and in greater size [20].

Implementing security into IoT-devices is a challenge as the devices are made from components with limited resources and power. Best practice and secure encryption is therefore hard to implement as it requires a larger amount of processing power and thereby also require bigger batteries. The IoT ecosystem is based upon a rapid production development life cycle, where low-cost components are used to make the products affordable and in big quantities [19].

Due to the data that IoT-devices collect, the manufacturer is also responsible of following the General Data Protection Regulation (GDPR). This means that the user has some specific rights when it comes to their data, such as knowledge of what data is collected, how that data is processed and whom this data is shared with. The user also has the right to revoke the organizations rights of that data.

Loss of control of such data would be a security concern for the user. It is therefore important that such organization follow the requirements of GDPR.

IoT-devices are a huge challenge within cyberspace as they are hard to secure, distributed in great numbers and contain and collect sensitive information. It was therefore necessary to develop a framework for testing the security and find out if the devices are trustworthy.

1.2 Task

The task was to create a framework for how to evaluate the security of IoT-devices, with respect to how much GDPR has been considered by the manufacturer of the device. The thesis would therefore have to answer the question, "how can the group build a framework to evaluate the security of the cybernetics, and processing of personal and sensitive data in IoT-devices?". To help answering this question, the group was provided with a set of modern IoT-devices to facilitate the study. The devices was used on a various level of testing and development of the framework. Some of them was used to test the whole framework and would set examples for usage of the framework, whilst other was used for specific areas of development. The devices used are:

- "D-link compact full hd wi-fi camera"[1]
- "Ledvance Smart+ Wi-Fi Plug"[2]
- "Ecovacs Deebot roomba"[3]
- "Adax Neo WiFi panel oven, hvit"[4]
- "Nedis Smartlife WiFi Smart Plug" [5]
- "D-Link WiFi Smart Plug " [6]
- "Nedis Smartlife WiFi Smart Surveillance camera" [7]
- "Mill WiFi-Socket 3 Plug-in-termostat"
- "TP-link Kasa Smart WiFi LED-Bulb"
- "TP-Link Tapo Smart RGB Led-bulb"
- "Cleverio Smart Smoke and heat-detector"

It was chosen to aim this framework towards businesses and organizations that

[1] D-link camera evaluation: `https://github.com/janstrng/Evaluating-IoT-devices/blob/main/Devices/Framework_2/Dlink_camera.md`

[2] Ledvance plug evaluation: `https://github.com/janstrng/Evaluating-IoT-devices/blob/main/Devices/Framework_2/Ledvance.md`

[3] Ecovacs roomba evaluation: `https://github.com/janstrng/Evaluating-IoT-devices/blob/main/Devices/Framework_2/Ecovacs.md`

[4] Adax evaluation: `https://github.com/janstrng/Evaluating-IoT-devices/blob/main/Devices/Framework_2/AdaxOVN.md`

[5] Nedis plug evaluation: `https://github.com/janstrng/Evaluating-IoT-devices/blob/main/Devices/Framework_2/Nedis_Plug.md`

[6] Nedis plug evaluation: `https://github.com/janstrng/Evaluating-IoT-devices/blob/main/Devices/Framework_2/Dlink_plug.md`

[7] Nedis plug evaluation: `https://github.com/janstrng/Evaluating-IoT-devices/blob/main/Devices/Framework_2/Nedis_Camera.md`

import, deliver and sell IoT-devices, and users of such devices with an interest for security. It is a goal for the framework to give the target audience a simple way to evaluate and test the devices before distributing, selling or using them. The idea for the framework is to be clear and simple to use, and make the result of the process easy to understand. By the end of an evaluation, it should be clear what weaknesses the evaluated device possess.

1.3 Scope

Based on the target audience for the framework, there had to a be a limited number of methods that would require specialized knowledge. Thereby giving the framework a low-entry threshold. Both to make the framework accessible, as well as not to end up with an uncontrollable scope creep. Therefore the framework is a tool to perform a high-level evaluation of IoT-devices, making use of free and publicly available tools. The goal of the framework is to fill the limitations when it comes to gathering information of how an IoT-device functions, and if it is safe for use. The framework is based upon the testing and evaluations performed on devices that are commonly found in households. These devices are commonly connected with Wi-Fi and collect and process data before communicating with an external server. The devices collect and make use of telemetry data that is collected from sensors such as thermometer, moisture meter, cameras, microphones and electric meter.

1.4 Delimitation

Security within cyberspace requires many specializations from a wide variety of skills. The framework is therefore not a methodology of how to perform penetration-test or find new vulnerabilities in IoT-devices. This framework is also not a replacement for other frameworks or requirements. Therefore passing this framework does not mean it passes regulations such as GDPR. The framework gives an indication on how security has been taken into consideration by the manufacturer when developing the product. However, this framework does not evaluate or secure the whole supply chain, as it would be infeasible to collect such information as a regular consumer. Subjects such as code-review and vulnerability-hunting/research have been taken out of the scope as these are huge subjects that would require, and already have, frameworks of their own. The main focus of the framework has been devices that use Wi-Fi as their primary method of communication towards external servers on the internet. Therefore it does not include criteria or methods to evaluate devices that uses Zigbee or Z-Wave as communication protocol.

1.5 The groups background and competence

The participants of the thesis, consists of four students, that all have taken the NTNU's Digital Infrastructure and Cybersecurity program. The group have different levels of experience with certain aspects of the project and have throughout the study program used risk management to evaluate and manage risk within a company and are familiar with Linux command line, operating systems, network communication and configuration. The group is also familiar with the process of teamwork and collaboratory projects in courses such as DCSG1002 and PROG1004.

Prerequisite knowledge

The group needed to learn how to use the tools, to perform the various attacks that was performed on the IoT-devices. This was learnt by watching YouTube videos and following guides and cookbooks. The group also needed to freshen up on some things like basic Linux commands and how to use GitHub again, because there had been some time since the last time a group member had used GitHub or a Linux based machine extensively. The approach to the devices that the group were to evaluate, would be what is called a black-box. Unlike "white-box" or "gray-box" the attacker have no knowledge of what the internal structure of the device look like or functions. The approach would therefore be from a reverse engineer standpoint and would influence what tools and methods that had to be used. The tools the group wanted to immerse into had to be publicly available and preferably free, luckily there are many such tools.

Kali Linux[8] is an operating system designed for penetration-testing and attacking data-systems. Embedded with the operating system are many pre-configured tools that could be found useful for testing.

Wireshark[9] is one such tool the group had to get familiar with. As this is a comprehensive tool for analysing and monitor network traffic. The group focused on learning the basics of monitoring and reading packages.

Burp Suite[10] is a tool-kit for analyzing web packages and web security testing. The group wanted to learn how to catch, edit and send packages towards a web interface.

Nmap[11] is a tool used to scan devices for their interfaces and open ports, which would help with detecting attack-vectors. This meant that the group also had to look into different types of services that run on the devices such as HTTP, RTSP and IRC.

Ettercap[12] is a powerful tool for Man-in-the-Middle (MITM)-type attacks. Potential attacks the group would be able to do with the suit, would be to spoof the

[8] https://www.kali.org/

[9] https://www.wireshark.org/

[10] https://portswigger.net/burp

[11] https://nmap.org/

[12] https://www.ettercap-project.org/

communication towards the device and gather its firmware.

Binwalker[13] is a tool for extracting files and executable code from a binary image. If able to gather the firmware, the group would attempt analysing it with the use of this tool.

1.6 Structure

During the project the work structure was affected by the pandemic and difficulties around getting a meeting room for internal meetings within the group. This had some affect on the work structure, but with a good structure set at the beginning of the project, did the group manage a good work flow.

1.6.1 Time allocation

Since the field of IoT is a huge subject, we had to define the scope of our task to best utilize the time we had for the project. In accordance with our employer the focus of the framework was to evaluate the general security of the devices and how GDPR and the users privacy was handled by the manufacturer. To specify the scope further we chose not to go into subjects that would require specialized knowledge as mentioned in section 1.4.

To prevent stress and too little time on the project, it was planned out a GANTT chart that depicted at what stages it would be worked on at the different aspects. To ensure progress the group planned to have daily meetings on week days starting from 10:00 to 14:00, before working separately at tasks that were assigned to each member. It was planned that once a week the group would conduct a meeting with the supervisor and task giver, however it ended up being closer to every other week as all group members had one other subject besides the thesis and as there would be more progress and prepared questions to discuss during the meetings.

As depicted in the GANTT the group needed to be defined a timeline of the thesis and specified deadlines for aspects of the project, such as when to do research, complete testing and finish drafts of the final thesis. Besides this, there was declared deadlines for other specific tasks during the project, that individuals was working on, like the different iterations and aspects of the framework.

1.6.2 Practical surroundings

During the work on the project, the group use a combination of communicating in "Discord", or meeting up in person. When the group members could not meet in person due to sickness or a lack of meeting rooms, the meeting would be held online in "Discord". The biggest differences between these meetings, and the physical meetings would be that the physical meetings would contain a more collaboration, discussions, whilst digital meetings would contain more of planning

[13] `https://www.kali.org/tools/binwalk/`

and delegation of work. When working together in a meeting room, the group can discuss multiple topic at the same time, because of the flexibility of meeting in person against digitally.

The meeting with the task giver was done on "Google Meet", and was structured for updating the task giver on the progression since last meeting, and asking questions that was prepared. These meetings were better to have online, rather then in person because of travelling, time, and logistics. Every meeting with the task giver were around half an hour, and consisted more of presentations and questioning than discussions.

The meetings with the groups supervisor had a similar structure to the meetings with the task giver. These meetings were held on "Microsoft Teams", and consisted of updating the supervisor on the progression, and asking questions that came during the process. These meetings were also better to hold online due to the structure of these meetings and the practicalities for the group and the supervisor.

1.6.3 Work organizing

As mentioned earlier in the thesis, the group had two different methods of progressing the project, either separately or as a group. During the project the group used both methods, as the group worked individually with progressing the work within a assigned task, like writing a sector in the thesis, or testing a device for parts of the framework.When the group needed to have digital meetings, the meetings involved more of delegating assignments, and planning future work than collaboration on testing or development. The individual work was done between meetings, and follows unified process that is described in section 3.2.1.

1.6.4 Technical

During development of the framework, when researching, writing and creating drafts, "Google Disk" was used to share research papers and write on documents together. "Discord" was also heavily used to share files and conduct meetings whenever there was a virtual meeting within the group. "GitHub" would be the main platform for the framework. "GitHub" is a commonly used platform for sharing and hosting software and development assets as well as blogs, frameworks and projects. It is a subsidiary of "Microsoft" and primarily uses the technology Git, with its own features on top. Git itself is a software that is used primarily for collaborations with multiple developers. Git lets its users track changes in a repository.

The structure of the thesis is based on a template that was made available for the group by NTNU. The template is a made in LaTeX and it was recommended to use it in "Overleaf". The template is created by Community of Practice in Computer ScienceEducation at NTNU (CoPCSE@NTNU). In addition, the group also used Introduction–Method–Results–and–Discussion (IMRaD) for the general structure of the thesis.

1.7 Other Roles

1.7.1 Task giver

The taskgiver for the thesis was Eirik Stephansen from Telenor Norge AS, Tech Lead IoT-Mobile.

1.7.2 Supervisor

The supervisor for the thesis was Andrii Shalaginov, Associate professor in the Department of Information Security and Communication Technology.

1.8 Structure of the thesis

Chapter 1: Introduction, introduces the reader to the problem and task presented in this thesis.
Chapter 2: State of the Art, Discusses the field and background of the problem that is represented and fundamental theory that covers the rest of the thesis.
Chapter 3: Research methodology, explains the method and how the group went about creating the framework. Here the thesis discuss requirements for the product, choices the group made for the methodology and shows what the development looked like through the different iterations of the framework.
Chapter 4: Result and analysis, shows the final framework, as well as the tools-file and method-file that follows it, plus a reasoning for every criteria of the framework and how the group evaluated.
Chapter 5: Conclusion, discusses the final product, challenges met and other possibilities. The thesis also discuss future improvement and work that can be done, as well as limitations of the product.

Chapter 2

State of the Art

This chapter of the thesis covers the field of study, and theory of the subject.

2.1 Field of study

"Cybersecurity is the practice of protecting critical systems and sensitive information from digital attacks" [21]. The practice works against crimes and attacks that makes a threat to organizations, companies or individuals. The field of cybersecurity is constantly under development. New exploits and vulnerabilities are developed and discovered, and as preventions are implemented the focus and targets of attacks also change. As the technology changes and new methods are introduced, so are the methods needed to defend and attack such interfaces. New trends such as remote work [22], 5G and artificial intelligence [23], are some examples of things that has changed how security must be adapted.

2.1.1 Previous work

ETSI

There are already a number of regulations and requirements that are demanded upon the security of IoT-devices. One such document is the baseline requirements that ETSI has created, more specifically, "ETSI TS 103 645 V2.1.2" [10] that lists requirements for consumer Internet of Things. This is a comprehensive document, specifying high-level security and data protection provisions for devices connected to the internet. The documents intent is to protect against elementary attacks on fundamental design weaknesses. Performing tests on devices, based on these regulations require that the tester also has a comprehensive testing-lab, specialized knowledge and tools to perform the tests. It also works as a checklist for developers of IoT-devices to evaluate their own security and how to implement best-practise.

GDPR

The "General Data Protection Regulation" [4] is a privacy and security law that are imposed onto organizations that are targeting and collecting data on people related to EU. The regulations sets demands based on the organizations practice of lawfulness and transparency. How data is stored and collected securely and how accessible the data collected is to its owner and rights to opt-out is implemented, are some of many demands. GDPR has the right and power to fine any organization that does not fulfill the requirements that the regulation sets [7].

Mozilla; *Privacy Not Included

"*Privacy Not Included" [24] is an initiative by "Mozilla" for consumers that help with buying safe technological products. The rating of each product is based on the level of *Privacy*, *Security* and in what way Artificial Intelligence (AI) is used. This website makes it easy to see how a device is scored using their scale, "How creepy is this?", that is rated by the users, after the user has read and gotten an introduction to how data is processed within the device.

ENISA; Guidelines for Securing the Internet of Things

The European Union Agency for Cybersecurity, ENISA, has a report on how IoT supply chain should be secured. This report includes best practice implementation based on threats and types of attacks that the devices are exposed to. The report covers the whole supply chain of the devices; design, production (semiconductors and components), assembly, firmware, software programming as well as platform development, distribution, management and recovery [25].

Papers

There has been made a lot of papers and research within IoT. Amongst them are papers that describe the process of attacking specific devices to gain unauthorized access to its functionalities. These papers has functioned as an introduction to penetration-testing IoT-devices and how to go forward checking them for vulnerabilities that are commonly found. For example, Rami Achkoudir and Zainab Alsaadi evaluates the security of a smart plug by performing penetration tests towards the device [26]. The same goes for Christopher Robberts and Joachim Toft writing about finding vulnerabilities in smart locks [27].

2.1.2 CIA-triad

The CIA-triad is a model of the three core concepts within cybersecurity.
The main goal of cybersecurity is to achieve all sides of the CIA-triad; Confidentiality, Integrity and Availability [28]. The components of the CIA-triad is something that must always be taken into consideration when reviewing digital security, and

will therefore be a core pillar during development of the framework.

- Confidentiality; must be achieved to secure that only authorized personnel has access to files, areas and information that is confidential. For different reasons passwords, business-secrets, personal and sensitive data must be kept confidential to keep security intact.
- Integrity; ensures that the data is correct and accurate and can be trusted. This includes securing that the data stays intact during transit and that unauthorized personnel cannot alter that data.
- Availability; makes sure that the data is accessible and available for the authorized personnel when needed. Often measured in uptime where as little downtime as possible is the goal. If data is lost, there is preferably methods for recovery and backup to ensure accessibility.

2.2 Background

Telenor Norge AS provided the task to create a framework for them to use for evaluating the security of IoT-devices efficiently. As discussed in the introduction, IoT-devices, could make a huge impact on the security of its users. The field of IoT is also a huge and comprehensive subject within cybersecurity. Therefore a framework was needed to orderly test all aspects and functionalities of the devices in an effective and thorough manner.

2.3 Theory of subject

In this section of the thesis some concepts and required knowledge will be discussed to give background to the subject. The concepts that are presented are relevant to what is further discussed in the thesis.

2.3.1 IoT-architecture

The IoT-architecture is structured in three primary layers [29, 30]. The “Perception and Hardware”-layer, includes all of the sensors on the device and is where data is collected and constructed. Data is being collected from the environment around the device and is being processed into meaningful information that can be analyzed. The “Network and Communication”-layer establishes a gateway, for the data that was captured to be transferred through. The data is moving through the application, and between the various devices and in the end, data is sent to the back-end service. The “Application”-layer is where data is sent to a data center for final analysis and viewing purposes for the end-user.

In between the "Network and Communication"- and the "Application"-layer there is often depicted a "Management Service"-layer [31, 32]. This layer functions as an interface between the two adjacent layers. This layer is responsible for the information management, capturing and handling large amounts of raw data and extracting the relevant information.

Each layer of the IoT-architecture is vulnerable to different sets of attacks and represents an attack vector for a threat actor. It is therefore important that every layer of the architecture is protected and secured. The “Application”-layer is for instance vulnerable to injection-type attacks while in the “Network and Communication”-layer there must be implemented features that prevent eavesdropping or spoofing.

2.3.2 IoT supply chain

Development and production of an IoT-device requires work from several different industries. Every actor involved in the production of putting an IoT-device together must be secured and trusted to create a secure device. Keeping the IoT supply chain secure is a difficult task as potential threat actors might have a lot of knowledge and resources to take control of such devices. It is also a difficult task for the providers of the devices to control and validate each step of the supply chain. The supply chain by ENISA is separated into six stages [25].

figures/enisa/enisa_iot_supplychain.png

Figure 2.1: IoT Supply chain [25].

The supply chain is composed of two main aspects; the physical and the logical. The physical aspect relates to all of the physical components and objects that the devices is made of, as well as the manual processes, such as assembly of devices and distribution.

The logical process refers to its software development and deployment, as well as network-based communication and virtual interactions between the device and supply chain stakeholders.

- During the conceptual phase of the product, the goal is to establish the security foundation. This will make it less costly to implement the functionalities later during production and making the implementation of feature more embedded within the application.
- The development phase consists of several stages that must be included in the processes of make the device. It includes the production and assembly of semiconductors and component as well as development of software and operational platform. This is one of the more critical steps of the supply chain as many of the risks and weaknesses of devices arise from poor decisions during the development.
- The production phase depicts mass production, distribution and logistics of the IoT-devices. Since production usually use multiple different manufacturers for development, this step could often become complex.
- The utilization phase involves all of the tasks that are required to get the device up and running at the end-users location. This usually involves physical installation, initial device set-up and establishment towards remote server.
- The support phase describes repairing, fixing or replacing damaged units. Another important goal of this phase is constant supervision of the devices security. This information is critical for further developments, maintenance and updates.
- The last phase of the IoT supply chain is the retirement phase. During this phase it is ensured that the device is disposed securely. This includes removal of information that is embedded within the device and recycling of mechanical components in an environmentally friendly way.

2.3.3 Risk management

Risk management is the process for identifying and assessing risks based on vulnerabilities and assets that are possessed. Following the NIST risk management framework, the procedure consists of a process in seven steps; "prepare", "categorize", "select", "implement", "assess", "authorize" and "monitor" . This process is performed to mitigate and avoid the risk of potential damage within an organization, as well as protecting individuals. This includes equipment failure, environmental disruptions, machine or human error and purposeful attacks [33].

Finding vulnerabilities and threats to analyze likelihood and consequence can be done by creating scenarios that describe the unwanted events. The scenarios described could be events that make an impact on; physical-, electronic-, human- and organizational assets [34].

Risk matrix

For every scenario, a value of likelihood and consequence is given to the scenario, usually from 1 to 4 or 5. Describing the likelihood of a scenario, 1 indicates that there is a low chance for the event to occur while 4 indicates that the chance is very high. The level of consequence describes how critical the event would be for the organization, thereby loss of confidentiality, integrity, availability and reputation. This is often measured in the amount of money that has been lost to recovery, assets or in intellectual property. The values of a risk matrix with five levels of consequences can be described as; Negligible, Minor, Moderate, Significant and Severe. Finally the value of risk is measured with $consequence * likelihood$. Thereby a scenario where a consequence would have critical impact on the company, by the likelihood of it happening is very small, the value of risk would be $4 * 1 = 4$. Having a scenario where both the level of consequence and likelihood is high the measure of risk will be $4 * 4 = 16$ and indicate that this scenario has a very high risk to the company.

figures/Framework_developement/riskmatrix5x5.png

Figure 2.2: Example of 5x5 risk matrix [35].

Threat actors

The likelihood of an event to occur within a company will be dictated by what type of threat actors are operating within the field and what motivations they have for the attack. In each scenario a threat actor should be described by their relationship to the company. Whether the actor is external or internal to the company, what access does the actor have to the company, their level of resources, knowledge and capacity as well as their intention and motive for the attack [34]. Different groups and types of threat actors can be segmented in a pyramid(as seen bellow).

figures/enisa/trusselaktÃÿrer.png

Figure 2.3: Pyramid of threat actors [36].

The framework[1] discusses each level of the threat actors pyramid, their capabilities and resources. For the framework it have chosen to utilize a 4x4 Risk matrix for the risk management, it is discussed why in section 5.2.2. There is also a more detailed explanation of the threat actors pyramid in the framework.

2.3.4 IoT protocols

Even though IoT-devices utilize different types of technologies and few standards, there are some methods that are used more frequently than others. Frequently utilized protocols for each layer, are ordered by the IoT architecture, and listed below [37].

Perception layer

The perception layer of the IoT-architecture as mentioned before, described as components that collect and process the first instant of information. Many IoT-applications perform real-time analysis and measurements. The processing of raw

[1] Framework for evaluating IoT-devices : `https://github.com/janstrng/Evaluating-IoT-devices/blob/main/Framework.md`

sensory data that is first collected from sensors is processed in the perception layer. This type of technology i described as a Semantic Sensor Web (SSW) or Semantic Sensor Network (SSN)-ontology [37, 38]. Examples of such technologies are "IoT-lite" and "SOSA".

Network- and Communication-Layer

Wi-Fi (802.11) is the most commonly used network protocol and is found in most households, airports, restaurants, etc... Wi-Fi does have a high power requirement, but works well for applications that require high bandwidth and low latency. Wi-Fi has the opportunity to work with a wide variety of common profiles that IoT-devices make use of [39].

Zigbee over IEEE 802.15.4. support low powered monitoring and control over devices specified for low datarate. Despite low power consumption, Zigbee still provide rage up to 150 meters [40].

Message Queuing Telemetry Transport (MQTT) was created to be able to accurately send data over low bandwidth and long network delays. MQTT is especially suitable for constrained IoT environments due to its simplicity, open source code and support for devices with limited bandwidth [41].

Management Service-Layer

Thread is an IoT protocol released in 2015 and is used on IoT-devices with low battery power, it is used to secure and future-proof the device. It is built to be an open standard and is compatible with a wide variety of home application and automation. NEST thermostats and many ZigBee devices has compatible hardware for thread [42].

Application-Layer

Constrained Application Protocol (CoAP), is the application layer protocol designed as replication and compression of the HTTP to support devices with limited processing power [43, 44]. CoAP is preferred over other protocols when document transfer is a requirement.

Extensible Messaging and Presence Protocol (XMPP), is well established and used on the internet. XMPP is highly extensible and allows extensions of protocol [45]. XMPP supports real-time communication with a range of applications such as instant messaging, voice and video calls and general routing of XML data [46].

MQTT can also be placed in the application-layer as it supports a web socket. Together with CoAP it is the leading messaging protocol in the IoT market and is preferred over CoAP when sending information constantly [37, 47].

Chapter 3

Research Methodology

In this section of the thesis the requirements of the framework, and the process of developing is described. The result of the method was heavily influenced by research of what could result in a good product.

3.1 Requirement specification

Based on the target audience, the framework would have to be clear and easy to use for a person with limited technical knowledge. For the framework to be of use, it would require enough technical methods , to perform a thorough evaluation for it to reach a middle ground to both please the targeted audience and make a satisfying result. Usage of tools would therefore be an important part of the framework as it would only need to teach the user how to use the necessary functionalities to test the device, instead of teaching them the technical knowledge necessary to get these results manually. The framework would therefore require an introduction to the tools and a description of how those tools were used towards the device under testing.

The framework would need to follow a structure that would make a logical approach to testing functionalities of a device. The approach is based on how the group went about evaluating the devices, and the methodology of testing, that is explained in 3.2.6. Accompanying the framework and the criteria, is a description and explanation of how to test these criteria. Each criterion of the framework is written in a way that makes them easily identifiable, readable and compact. A more in depth explanation of each criterion and method of evaluation is located in a separate document referring to their respective criterion.

As an addition to the framework there are created examples of evaluations of certain IoT-devices. This will clarify what the framework is attempting to evaluate and show what a finished evaluation is supposed to look like. By showing a practical example, the user can easily find out if this is the type of framework that is needed for their use case. Adding a reasoning for why the evaluation of the devices was given the value it was given would also give the end user a reference of how to test, and what to look for when evaluating for themselves.

3.2 Methodology of development

For this part of the thesis we will describe our method for developing the framework. It depicts how tools would be utilized for the development and how tests are performed to improve on the framework.

3.2.1 Project plan

The Bachelor-thesis started with creating a project plan, planning how the group should approach the task, organize, and what goals were set for the project.

Unified Process

The process chosen to be used for the method of work was a variant of the Unified Process (UP). The main reason the method chosen was because of the iterative development method. The group are aware that from the start the group would not be able to gather all the information and knowledge needed to develop a framework to evaluate IoT-devices. Therefore it would make sense developing the framework in parallel as the group researched, tested and explored the devices.

The Unified Process consists of four parts that are repeated in sequence:

- In the *Inception* part, the scope is defined of what the group is going to test and how in depth the tests are going to go.
- *Elaboration*, starts with preparing the test environment. This includes setting up the devices, as well as doing research and preparing methods to attack the device based on the scope defined.
- During *Construction*, the group would perform the evaluation and tests of the device, based on the scope of the framework and methods prepared.
- In *Transition*, there would be an evaluation of the results of the evaluation and see what went well and what did not. There will also be an evaluation regarding the criteria that were deemed unnecessary or missing. The group would then go back to the Inception-step and improve and expand the scope of the next evaluation.

The Unified Process is mostly used for software development, but the conclusion from the group was that it would work for developing a framework that would require iterations.

Planned project progression

Phase one of the project consisted of the creation of the project plan and defining the scope of the thesis. Phase two focused primarily on research and creating the first draft of the framework. After phase two would the group start phase three of the project, that together with phase four followed an iterative process. The plan was to do one week on phase three and another on phase four and repeat the process from phase three. In the iterative process the focus shifted to testing the framework, and develop the framework further, this was all in accordance with the GANTT chart. The reasoning behind doing so was that as the group would gather more knowledge of IoT, the project would be able to expand the scope of the framework as the development of it continued. The group would also be able to test what was implemented and figure out if it worked or not. Thereby saving a considerable amount of time instead of making a fully fledged framework, and then discovering that some parts of it either does not work, or need heavily modification.

Some aspect of the previous two phases would not be able to completely be finished within the deadline for this part. These aspects was waiting for a responses to emails regarding the request of the use of GDPR article 15. The manufacturer have 30 days to answer this, and the group waited for more data on the user profile before requesting this. In addition, did the group wait with deleting the profiles until the testing was done. During phase five of the project, the focus shifted to writing the bachelor thesis. This was done partly together in the group for details and discussion, and partly individually to get most of the writing done.

During the project planning the main focus was on the testing process, and how this would turn out. The framework was thought to be developed more around the testing, then the other way around. This was a major shift that happened during the project. The testing was still a part of the project, but the framework was the more important part of the project, and the testing did develop into a supporting role around the development.

3.2.2 Tools

The project plan laid out the tools that would be used. The group ended up using all of them, and added more. The most used tool was "Discord", the primary tool for communicating within the group. "Google Drive" was used for storage of files. The thesis is written in "Overleaf", using "LaTeX", and "GitHub" was the place where the framework and tests got published. Wireshark was the most used tool for scanning network traffic, but in addition to it, there were used several different tools to monitor, attack and scan different parts of the devices. These tools were "Nmap", "Ettercap" and "Burp Suite". Most of the tools are available through "Kali OS", and made it possible to perform most of the tests and attacks.

3.2.3 Creation of framework

The whole point of the project was to look at how IoT-devices use user data, and especially in comparison with GDPR. To do this the framework is based upon existing methods that are proven. The group used *ETSI TS 103 645 v2.1.2*[1] for the more technical and security aspect of the framework and GDPR[2] for safety of user data. ETSI has many standards that are alike, but the chosen standard was "*ETSI TS 103 645 v2.1.2*" as opposed to "*ETSI EN 303 645 V2.1.1*", because TS defines the technical specifications and its purpose is to test against the provisions of "EN 303 645" [48].

GDPR was a logical choice as it is the toughest security and privacy law in the world [7]. The law provides obligatory criteria for any organization that collect data on people related to the EU has to follow. Our framework gives the user an opportunity to score IoT-devices, and therefore see what the unit does well and not so well.

The group decided that creating a framework would make more sense based on how other requirements within the field are portraying their requirements and criteria of security. There was a question on whether the group should make the content of the framework from scratch or base it on existing frameworks and supporting texts. Due to the limited experience in the group, the best course of action was to base it on existing frameworks, regulations and supporting texts. These frameworks also served as a guideline, but they needed modifications and additions. The existing materials are not applicable for the target audience, because of their depth and complexity. A framework for the target audience would be efficiently created by changing and modifying existing frameworks.

The ETSI framework is largely based on having access to a well supplied IoT testing lab, this means that many of the criteria are not suitable for our framework and target audience. GDPR is a law, not a framework, and therefore it is not necessarily made in a way fit for evaluation. The law includes rules that can not be acted upon by the consumer.

3.2.4 Data collection

The initial plan for the project was to familiarize ourselves with the field of IoT, researching the state of the art. Researching commonly found vulnerabilities and methods of attacking IoT-devices was found in research papers and books. Online tutorials and "cookbooks" of practical examples of attacks would also give a clear guide to what tools would be useful and how they are used. Reconnaissance is a big part of penetration-testing and a lot of time was used to research the manufacturers of the device, as well as what components the device was built from. With knowledge of internal components and what systems the device ran, further research and findings of vulnerabilities of the specific components would be possible. Most of the data collected would come from the tools used as referenced here. Searching for the device and then "vulnerabilities" would often result in write-ups of previous vulnerabilities that had most likely been fixed. However,

[1] ETSI TS 103 645 V2.1.2 [10]

[2] `https://gdpr-info.eu/`

this would increase the group members understanding of the device and demonstrate potential attacks on other devices. A part of the framework and the tests performed on the device, was to request information of what knowledge was gathered from the device. This would simply be done by finding the contact-point of the manufacturer and exercise article 15 of the GDPR.

3.2.5 Testing the framework

The framework was tested by using it to test the devices. Using the framework, there was findings on what worked and did not work. By doing so the framework changed dramatically from the first version to the last. The method of work on the framework was an iterative way, by always going back to improve it.

3.2.6 Testing the devices

The methodology for testing the devices was that everyone was responsible for setting up their own testing environment, except the Kali machine due to lack of available machines that could host it. The reasoning behind this was that everyone could perform their own tests, to get to know the device and how to approach the testing. This allowed the group to test multiple IoT-devices simultaneously and also give everyone the opportunity to learn how to use such tools as "Wireshark" and "Burp Suite". This also allowed for greater flexibility and better time management. The work was done individually on the simpler IoT-devices, i.e smart plugs and smart surveillance cameras. For the more complex IoT-devices such as a smart vacuum cleaner and smart electrical heater, the decision was that the best course of action was to work together. The smart vacuum cleaner would have more functions, be more complex and is also more difficult to set up. In addition, the vacuum cleaner needed space to operate in, so that was also something that needed to be planned for.

Testing the device, the person doing the testing would follow each criteria described by the framework in sequence, using the method and tools as described. To be able to find or discover potential criterion that was missing, reconnaissance of the devices was also performed. This would include searching for previously found vulnerabilities of the device and the hardware it ran on and found attack-vectors that has been proven to work on similar devices.

3.2.7 Changes to methodology

The distribution was changed from "Ubuntu" to "Kali" because "Kali" includes a lot of useful software and tools including "Nmap" and "Ettercap". This meant it was easier for the group, since it did not require installation of any software. "Kali" also advertises that it works great on a USB flash drive. As a consequence there was no longer need for a dedicated computer with a "Linux" distribution installed or dual boot "Windows" and "Ubuntu"/"Kali".

3.3 Test environment

For performing tests towards the devices, there was established a basic test environment to simulate a regular network that utilize the device under testing. One computer would be used to enable a "hotspot" from that computer, playing the role of a router. Connected to this computers network, the device under testing and any phone that uses the devices application. There would be performed tests on devices such as "D-link compact full hd wi-fi camera"[3], "Ledvance Smart+ Wi-Fi Plug"[4] and "Ecovacs Deebot roomba"[5]. While connected, there would be performed tests on the functionalities of the application that was used to control the devices, "mydlink", "LDV WiFi" and "ECOVACS HOME" for the devices mentioned.

On the computer hosting the network of the devices there would be a connected machine running on "Kali OS". It is on this machine where most of the tests and attacks would be performed towards the device, located on the same network. Here it would be utilized tools[6] such as "Nmap", "Binwalker" and "Ettercap". Setting up the environment this way would help with seeing the IP of the device and have simple access to the device from the attacking computer and little to no noise when monitoring the behaviour of the devices with tools such as "Wireshark".

3.4 Framework Development Process

The development of the framework was done in iterations by performing tests on the devices, researching and improving the framework, as described in the unified process. There were in total seven different versions of the framework, with improvements and upgrades between each of the versions. The chosen name for the official version of the framework will be "Framework 1.0". Every version of the framework before this is called "Framework 0.X", starting with "Framework 0.0". The reason this method of naming was chosen is because this is a common way of naming different versions of an application in software development. The first iteration of the framework is named "0.0" as it was just a summary criteria from ETSI's framework that was found to be relevant. Even though it is not the first version of the framework, it shows how the development started.

3.4.1 Framework 0.0

When researching for the framework, ETSI was found to be the best standard with most similarities to what was planed for the framework. ETSI is a framework with

[3] D-link camera evaluation: https://github.com/janstrng/Evaluating-IoT-devices/blob/main/Devices/Framework_2/Dlink_camera.md

[4] Ledvance plug evaluation: https://github.com/janstrng/Evaluating-IoT-devices/blob/main/Devices/Framework_2/Ledvance.md

[5] Ecovacs roomba evaluation: https://github.com/janstrng/Evaluating-IoT-devices/blob/main/Devices/Framework_2/Ecovacs.md

[6] Descriptions of tools from framework: https://github.com/janstrng/Evaluating-IoT-devices/blob/main/Tools.md

figures/Framework_developement/ETSIoppsum.png

Table 3.1: Framework 0.0

background in IoT security made by ETSI Technical Committee of Cyber Security. ETSI's framework is made to cover every IoT-device, from the most complex devices, to the simple devices, see more on ETSI in 2.1.1. All of these devices utilized personal data for different reasons. Therefore security is necessary for a user to trust the device. When personal and sensitive data is at stake, there may lead to serious consequences if the security mechanisms and routines are not proper.

When starting the development process, the group started by reading through ETSI's framework to find criteria for the project's framework. Only the criteria found relevant was included in the framework, and the criteria similar to each other were combined. These criteria were distributed into different tables with the same order as ETSI's framework used. This made up to a total of 14 tables with different number of criteria.

The 14 tables in the framework were divided into three columns; the requirement, then a method of evaluating, and a checkbox or a comment regarding the requirement. The first column, the requirement, is the criteria inspired of ETSI, like "*Default password is unique per device*", or "*Support period is published*".

The second column, the method of evaluation, is a suggested method of testing, or evaluating the criterion. At the early stages of the framework these points were more predictions of what subjects there would have to be done more research on. Examples of these suggestions were "*The device require the user to create a new password on startup or is enabled with an auto generated password*" or "*Look in the box, or on internet for Support period*".

The final column of the framework was a checkbox, or any comments regarding the criteria. The suggested checkbox where made as a check for the testing process. The thought was that every point would either get a pass or fail evaluation, and thereby a check for every approved point. The possible comment at the final column was for possible angularities with the criterion, method of testing, or any other reasoning for worrying.

3.4.2 Framework 0.1

figures/Framework_developement/Rammeverk0.1.png

Table 3.2: Framework 0.1

Framework 0.1 is the first version of the framework with more than just the ETSI criteria. This framework included new criteria from GDPR[7] and ENISA[8] in addition to the old criteria. The framework had the same design as earlier, with an individual table for every part of ETSI, ENISA and GDPR. ENISA and GDPR were included into the framework to make the framework more solid.

The criteria inspired by GDPR were taken from GDPR's article 1 through 20, in addition to GDPR's article 42 and 46. These articles stems from chapter 1 to 5 of GDPR. The main reason for excluding a chapter or an article was because it wasn't necessarily relevant for IoT-devices, the scope of the framework, the target audience, or because it was already represented by the ETSI's framework. An example of one such point, can be see in GDPR article 58 in chapter six of GDPRs regulations. This articles talks about the powers a supervisory authority shall have. Although this can be relevant for IoT-devices, it is not relevant for the scope, and

[7] General Data Protection Regulation [4]

[8] European Union Agency for Cybersecurity; Secure supply chain for IoT [25]

the target audience of the framework. ENISA were represented in the framework with the criterion "*Device is made of adequate physical material for its use case*" which covers, and backs up a hole in the framework after ETSI and GDPR were included.

3.4.3 Framework 0.2

figures/Framework_developement/Rammeverk0.2.png

Table 3.3: Framework 0.2

The main upgrades to the framework between framework 0.1 and framework 0.2 is the structure and the design of the framework. Where as in the framework 0.1 the structure were a bit messy, with the content structured after where it was gathered from. In Framework 0.2 the order and grouping of the criteria is based on what area it covers, and where in the process of setting up and testing a device, it becomes relevant. The framework consist of 12 tables after the restructuring. The criteria are grouped into the tables:

- Set up user
- Registration
- Authentication
- Defaults
- Updating
- Security
    Cryptographics
- Interfaces
- Erasure
    Manipulation of personal data
- Personal data
- Disclosing of vulnerabilities

In both cases where a group is within a bigger group of criteria, the smaller group consist of one criterion. The first of these criterion is "*DUT remains operating locally functional after loss of network.(ETSI 5.9)*". This criterion means that a device can, if necessary, be taken off the internet and still function, where the only functionalities missing when offline is functionalities linked to internet. An example of this is an IoT-plug, that, if not online, functions as a normal plug, with an on/off button. This is necessary for the framework because no other criteria covers this scenario, and this can be crucial, if the subnet is hacked. This criterion is put into its own group because it is a security criterion, but is also the only security criterion not needing cryptographics. The other criterion alone in a group is "*The data subject shall have the right to rectify the data without undue delay, and also complete uncompleted data about subject. (GDPR 16)*". This means that the user can edit, or complete, data that is wrong or uncompleted. An example of this can be a change of password, timezone, or change of email. This is not covered by any other criteria, and can be crucial if, for example, the used email is hacked.

The only new addition to the framework is the inclusion of a reference to where the criteria comes from, as seen in the examples above in figure 3.3. These references are kept to the final version, so users always knows where criteria comes from.

3.4.4 Framework 0.3

When updating to framework 0.3, the main focus was on making a proper scoring system. The idea that was worked from regarding scoring the devices, was a formula with the rating of the consequences, and the implementation, of a given criterion. The last column changed to represent the value of consequence, which represents the severity of a criterion for the devices security, or user friendliness. The values given are from one to five, where one is an suggested criterion, and five is a critical criterion. These values should then be calculated with a value for figures/Framework_developement/Rammeverk0.3.png

Table 3.4: Framework 0.3

the quality of the implementation of the criterion.

See more on the scoring system in 3.2.9

3.4.5 Framework 0.4

figures/Framework_developement/Rammeverk0.4.png

Table 3.5: Framework 0.4

At this point of the development, the framework is starting to take shape. From framework 0.3 to framework 0.4 there were three main changes, the addition of a column for the "value of implementation", and the fact that every criteria are put into one table, with the same order as the previous versions. The column for the "value of implementation" were also divided into three, where the criteria based on GDPR were marked with a "G", every security related criteria outside of GDPR was marked with a "S", and every criteria marked with "B" is related to both security and GDPR. The groups were thought to be used for scoring the devices in two scores, one for every GDPR criteria, and one for security criteria. The criteria with a "B" was thought to be in both score.

The latest addition to this version of the framework is the addition of the "Nr" column to the main table. This inclusion was necessary to maintain order, and a reference to which topic the criterion falls under. There is also an additional table of content for the "Nr" column.

3.4.6 Framework 0.5

figures/Framework_developement/Rammeverk0.5.png

Table 3.6: Framework 0.5

There is a change of values in the "value of consequence" column, which earlier was a scale from one to five, to now being a one to four scale. This is in context with the change from a mathematical formula to a risk matrix in the scoring system. The "value of implementation" will also have a score from one to four.

See more on the scoring system in 3.2.9

3.4.7 Framework 1.0 - Final framework

figures/Framework_developement/Rammeverk1.0.png

Table 3.7: Framework 1.0, See the whole framework in section 4.3

This is the final version of the framework and there are more significant changes from the earlier versions of the framework. The first change is the fact that the framework is not one table anymore, but back to tables segmented into topics. Therefore the "Nr" column has a different usage than earlier, by now being an identification for each criteria. It makes it possible to reference an individual criterion later, at other points in the framework. This is done under each table, where there are a reasoning for every criteria not valued to the best score at "value of implementation". The criteria is not divided into GDPR, and security anymore, as this seems a bit unnecessary when each criterion are already referenced in the risk matrix.

The groups the criteria is divided into are:

- Set up user and device
- Authentication
- Security and communication
- Disclosing of vulnerability
- Default password
- Erasure
- Personal data
- Updating and support
- Physical
- Further tests that require specialized knowledge

The last table is a table with criteria that were found relevant, and needed to be included, but will need some sort of special knowledge or equipment to be evaluated. The reason the topics for the tables are changed is because the title for the tables better explain what the content that is in the tables.

figures/exampleID.png

Table 3.8: Identification usage of a criteria

As mentioned earlier the "Nr"-column has changed to an identification for each criteria. This identification is then used below every table as a reasoning for the evaluation and for reference in the "Method" file, the "Tools" file, and in the risk matrix. An example of this is from the evaluation from the testing of the "D-link compact full hd wi-fi camera"[9]. A reasoning for "value of implementation" is put in for a couple of reasons. Firstly so the user should explain what they judge to not be best practice. Secondly for the manufacturer, so they know where their devices have improvements regarding security and GDPR.

3.4.8 Scoring systems

figures/Framework_developement/Scoring.png

Table 3.9: Scoring System

There were some different ideas for a scoring system for the framework, but there was no idea that revealed the weaknesses of a device as good as a risk matrix. A risk matrix is way to show risk based on "probability" and "consequence". There are many different sizes to a risk matrix, but in the framework it is a 4x4 size. See more here.

The normal formula for using a risk matrix is $probability * consequence$. When relating this to the framework, "probability" is represented by "value of implementation" because a bad implementation relates to higher probability, and "consequence" is represented by "value of consequence". The matrix should be read in a way that the bottom left corner has lowest risk, and the top right has the highest risk.

[9] D-link camera evaluation: `https://github.com/janstrng/Evaluating-IoT-devices/blob/main/Devices/Framework_2/Dlink_camera.md`

3.4.9 Values of rating

The final versions of the framework[10] included two values for evaluating the devices, namely "value of consequence" and "value of implementation".

Value of consequence

The most consistent value, "value of consequence", is a value suggested based on every device. This value is given to a criterion based on the criticality of the criterion. The possible values given in the framework to a criterion is 1, 2, 3, or 4. 1 is the least critical value for the security of a device, while 4 is the most critical value given.

1: Not required criteria

There are a couple of reasons for why a criterion has gained 1 as "value of consequence". The groups definition of the value is divided in to three main parts, within security, user privacy and requirements for the device. All these requirements need to be fulfilled for a criterion to gain the value. The security of the device should have "little to none" impact from the criterion. The user privacy should not be impacted, and the device should not be unsafe without the criterion fulfilled. An example of a criterion where this value is given is "*D.2.1: Device is made by adequate physical material for its use case. (ENISA)*".

When talking about the security of a device, and the physical material a device is made of the value might change based on what type of device that is tested. The devices available under the development stages of the framework were not devices that required tests performed on how adequate the physical material is. It does not have an impact of the security, but devices like a lock will need more robust physical materials, and thereby have a higher value of consequence. Moving on to the user privileges where the criterion is "*no impact on users' right to privacy*", where again the physical material do not have any privacy impact. Lastly it is not a criterion that is critical for the device's security, and the device will not be deemed unsafe if the physical material is not adequate.

2: Moderate criteria

Moving on to the value 2 in "value of consequence", where the criteria are a bit bigger. First of all, will a loss of implementation open for potential attack platforms, or a loss of confidentiality, availability or integrity. Secondly the criteria will not affect personal data, but can impact metrics. Lastly, the criteria affect the user friendliness of the environment and can lead to confusions if not fulfilled. An example of a criterion that is valued within these requirements is "*The user is able to check the software version for the device. (ETSI 5.3)*". This criterion can lead to the potential weakness if the current version has weaknesses, and need updating, or changes in settings. The criterion can make for confusion regarding software versions, lastly it can fail in a update affecting some metrics.

[10] `https://github.com/janstrng/Evaluating-IoT-devices`

3: Important criteria

Every criteria where the “value of consequence” are a 3, a potential weakness could lead to a major consequences. Firstly, the criteria have an impact on the device. A bad, or no implementation will weaken the security of the device, and will lead to security gaps and potential for attacks, loss of confidentiality, availability and identity. A lack of full implementation of such criteria could also lead to personal data going astray. In addition, could a lack of implementation of named criterion lead to confusion between customer and company.

An example of a criterion with the value of 3 is, *“A.7: Device look for and initiate updates when first enabled if not latest version.”*. If a device’s software does not look for an update, and do not initiate updates, it can lead to security gaps. A security gap can again lead to loss of confidentiality, integrity, and availability of personal data, as one of the most important security routines is routinely updating software. This is because old software can include old functionalities, errors, or flaws that is covered, updated, or removed in the newest version. This can also lead to confusion between customer and company on new functionalities in latest versions of the software.

4: Critical criteria

The most severe score a criterion can get is the score 4 of “value of consequence”. This value means that it is critical consequence if the criterion is not fulfilled. This means that the criterion should be critical for the devices use, and security. No implementation of such criteria means that all data can be lost or go astray. This means that the device is unsafe and should not be used without any implementation of the criteria. An example of such a criterion is "*C.3.2: All decisions about the user’s personal data are based on consent. (GDPR 7))*".

This is an important criterion for a couple of reasons. It is a requirement from GDPR, and therefore mandatory for the company, but also due to the fact that if a company does not need consent, then the possibilities for misuse of data, metrics and personal data is near endless. One can only imagine what a company can do with such freedom. Therefore should any device retrieve consent before usage.

Value of implementation

Value of implementation represent the likelihood of misuse of the criteria, and is given to show how good a chosen criterion is implemented in the IoT-environment. The reason the name given to the section is "Value of implementation" is because all though it represent the likelihood of misuse, it show the value given for the implementation done to fulfill a criterion. The values given in this section of the the framework also automatically links to the reasoning of the values. It is mandatory to give a reasoning for the value given, if the value given is not 1, which means anything that differs from best practice. An example of a reasoning can you see in table 3.8

Good implementation

If an implementation of a criteria are found to be best practice, and with no weaknesses, the value 1 given to the criteria. This means that the implementation of a given criteria will be deemed as good as it gets. If the criteria gets the "value of implementation" as 1, an in-depth description is not required, but a reason has to be ready. Luckily there are many of examples where the "value of implementation" is 1, like every criteria in table "*A: Set up user and device*". The likelihood for misuse of a criterion where this score is given will be very low, and the possible threat actors will be in tier 5 or 6 of the threat actors pyramid.

See more on Threat actors at 2.3.3.

Implemented

When the criteria are implemented, but not as best practice. The value given to the implementation is 2. This can be achieved in a couple of different ways, one method is to try to implement with best practice, but this have been done with some flaws. Another way to gain this score is to implement a function, or the criteria in a way that is not best practice, but a way that has known weaknesses. An example of such a method is the criterion "*B.2.1: Communication of personal and sensitive data use best practice cryptographic.(ETSI 5.8)*". in the evaluation of an Ecovacs vacuum cleaners. The reason behind the given value is because the device uses the CoAP protocol. "Traffic does not seem to be encrypted, however it does not seem to be able to make use of the format.". When talking about the likelihood of misuse of such criteria is low, and possible threat actors will be from 3, or higher.

See more on Threat actors at 2.3.3.

Barely implemented

The lowest valued implementation, and the second worst value possible implemented. If the criteria gets this value for the implementation of a criterion, it means that the test values the implementation in such a way that there is no effort behind it, or implemented with major flaws, not implemented with best practice. The possibilities for attacks with misuse of the criterion based on the level of implementation is moderate. There are some security mechanisms implemented that does help, but this implementation is not deemed good enough. An example of a criterion where this validation is given are the criterion "*Password is recommended to be at least 8 characters and consists of at least one character from each character group. (big letters, small letters and numbers). (ETSI 5.4)*" for the test D-Link's camera. The application to D-Link only had a 6 character requirement for the passwords for a user, without any requirement for different characters. This means it is possible for threat actors to misuse the implementation with this score, and the actors has to be in tier 2 or above in the threat actor pyramid.

See more on Threat actors at 2.3.3.

No implementation

This is the lowest possible score an implementation can get, which means that there are no implementation to fulfill the criterion at all. The requirements for this score is that the criterion is missing an implementation, which means that likelihood misuse of the criterion is high. In the case of threat actors, will every threat actor in the pyramid have competence and resources to misuse the criterion. An example of a criteria where this value is given is "*A.4: The default value for a decision follows best practice for security. (ETSI 5.12)*" on Ecovacs's vacuum cleaner. The reason for the score was that a lot of values, including option on data collection, access of data, and status of the device, was on by default. This means that a lot of functionalities are set on, in the application, not necessary is a direct way in for an attacker, but can lead to misuse and unauthorized access to data.

3.5 Testing quality control

Throughout the production of the framework there were many test done using it. Most can be found on the Devices. These tests were used to both test the user-friendliness of the framework and the result the framework provided. These tests were done throughout the development phase, and were very helpful in making a usable framework. Most of these tests would result in changes to the framework, either with changes too already existing criteria or additions/subtraction of other criteria.

Chapter 4

Result and analysis

This chapter presents the product of the project. This product consist of the framework developed by the group, a "Tools.md" made to explain the tools used, and a "Method.md" made to explain the method of testing.

4.1 Introduction of the framework

The final framework can be found at the GitHub [1]. This repository includes the three main documents and examples of finished evaluations. Some of these evaluations of the products are, "D-link compact full hd wi-fi camera"[2], "Ledvance Smart+ Wi-Fi Plug"[3] and "Ecovacs Deebot roomba"[4].The three main parts of the repository are the framework, that is found in section 4.3, Tools, found in 4.4, and Method, found in 4.5.

The framework section consist of information around the framework, such as defined scope, definitions of values, the criteria of testing, and the scoring system, in addition to the framework itself. In "Tools.md" file will you find the different tools used for testing of devices and developing of the framework. The "Method.md" file explains how to score and check the criteria in framework, it consist off a general explanation for each group and a more detailed explanation for a few criteria, this file is used mostly to make it easier to understand the criteria in the framework. In the sections below, the final framework can be found.

[1] Framework repository `https://github.com/janstrng/Evaluating-IoT-devices`

[2] D-link camera evaluation: `https://github.com/janstrng/Evaluating-IoT-devices/blob/main/Devices/Framework_2/Dlink_camera.md`

[3] Ledvance plug evaluation: `https://github.com/janstrng/Evaluating-IoT-devices/blob/main/Devices/Framework_2/Ledvance.md`

[4] Ecovacs roomba evaluation: `https://github.com/janstrng/Evaluating-IoT-devices/blob/main/Devices/Framework_2/Ecovacs.md`

4.2 Implementation of the framework

The framework is hosted on Github[5] and consist of one main markdown file with supporting elements. These supporting elements consist off, methods descriptions, tools used, example cases, and a simple README file. These can also be found further down in Result, except the test cases and the README file, the test cases can be found in the appendix and the README file can be found in "GitHub".

The framework is mainly a list of criteria that an IoT-device should meet. By going through the framework with an IoT-device, that device would get a score with a qualitative risk matrix, that you can find in section 3.4.8. There are no download or installation for the framework to work, but a few of the criteria are checked with external programs that require installation. You can get a full list at "Tools.md"[6], some of them are "Nmap", "Wireshark", and "Ettercap". Therefore, to use the framework you will need to either download "Framework.md" from GitHub or you can simply use a web browser and open "Framework.md" and go through it there. The explanation on how to use it will be on the main file, but you may also need the additional information from the files "Method.md" and "Tools.md".

"GitHub" was an obvious choice of use, because off its ease of use under development and implementation of the framework. Earlier in section 1.6.4, it was discussed on publishing the framework, this is another task GitHub can do. It can implement the framework, by hosting on the website as a public project. This will let other see and download the files, they can also directly clone(copy) the repository to their own devices. Another useful functions "GitHub" have is a history function, any file in a project have its history saved, what changed was done to it and by whom. This can be useful, both to simply see earlier versions and if something goes wrong and a file is changed or delete wrongly.

Another option was "GitLab", it is very similar to GitHub and have many of the same uses. Both "GitHub" and "GitLab" are created from "Git", but they are not entirely the same. Some of the more important differences is in how it handles paywalls, Continuous Integration/Continuous Delivery and how its workflow is. Both programs have a subscription, but because of the groups situation paying for these subscriptions are not relevant. Therefore, it was looked at the free functions, for the functions the most important one is the ability to create and manage public repositories, this is something that both have as a free function. "GitLab" also pride itself on being a “complete DevOps platform” and have therefore more built in services than GitHub. On the other hand, "GitHub" can get many of these services through integrating with third party programs. This point is also not that important to the development because the group will mostly make use of the general git functions and the graphical user interface. The general workflow is also slightly different, in "GitHub" you will mostly work on a master branch and merge

[5] `https://github.com/janstrng/Evaluating-IoT-devices`

[6] `https://github.com/janstrng/Evaluating-IoT-devices/blob/main/Tools.md`

other branches into that master branch. In "GitLab" on the other hand, you will have multiple staple branches that you work on in addition to a master branch. This may make testing of code more complex than on "GitHub", but the difference that mattered most in the decision was, that "GitHub" describe itself by being a place where you host open-source projects. The framework were to be open-source and accessible to all, this made "GitHub" stand out to the group and we therefore chose it [49, 50].

4.3 The Final Framework, "Framework.md"

You can find the final version of the framework below. This file is the main part of the project, and the part where you can find the criteria, definitions for values, and explanations of the different aspects around the framework.

The next ten pages is an outtake from "GitHub". Therefore, no links work, and the formatting on the pages will also change.

papers/Framework.pdf

papers/Framework.pdf

papers/Framework.pdf

papers/Framework.pdf

papers/Framework.pdf

papers/Framework.pdf

papers/Framework.pdf

papers/Framework.pdf

papers/Framework.pdf

papers/Framework.pdf

4.4 Framework's "Tools.md"

In the section below is the tools part of the framework, which describes all the tools used in the process of developing the framework. It is made to show the user of the framework which tool that were used, and recommended to be used during testing and usage.

The next six pages is an outtake from "GitHub". Therefore no links will work, and the formatting on the pages will also change.

papers/Tools.pdf

papers/Tools.pdf

papers/Tools.pdf

papers/Tools.pdf

papers/Tools.pdf

papers/Tools.pdf

4.5 Framework’s "Method.md"

In the section below you can find some suggested method of testing for criteria where a description is needed. In the first versions of the framework this was a part of the frameworks own columns, but for ease of use, it was a need to make a file with a better description of the method of testing.

The next four pages is an outtake from "GitHub". Therefore no links will work, and the formatting on the pages will also change.

papers/Method.pdf

papers/Method.pdf

papers/Method.pdf

papers/Method.pdf

4.6 Testing framework

This section encompasses the whole framework and its criteria and goes a bit more in depth on each criterion and how to evaluate them. The earlier section “Framework’s "Method.md"” is meant to explain the few criteria that are more difficult to understand and give a brief description about each table. The criteria that are explained in method will be explained in a similar way in this section. "E: Further tests that require specialized knowledge" is not explained in this section, because it was not a part of the testing the group completed and as the title says it require specialized knowledge.

4.6.1 A: Set up user and device

- A.1 The process is based on consent.
  The first criterion is a general point for the whole process from start to finished is consensual, thereby the user have to agree for things to apply to them and the user understands what they are agreeing to. There should not be any default decision that is not mention for the user and is at the expense of the users consent.
- A.2 The user knows what they accept, the possibility to know what consent means regarding the setup of the user.
  From each consent the user have to accept, you have to check if it is written clearly and is understandable. Here will a "read more"-option be acceptable.
- A.3 Security-relevant user decision is covered by the documentation and a recommendation is given.
  If the device application ask the user for a technical security-relevant question should a recommendation be given. This is because most users will not be able to understand what that question actual ask. If a decision would make the device less safe then it is, the user should also be informed, about the security risk involved.
- A.4 The default value for a decision follows best practice for security.
  If the device or the application for the device have default values, it is important that these follow best practice for security. By going through the device setup and seeing what decisions users must take and what users do not, the tester will be able to find some default values and the tester can see if these follow best practice.
- A.5 Decision taken by the user is understandable for a user with limited technical knowledge
  If a decision is understandable for “a user with limited technical knowledge” or not, is a subjective decision. Also, what exactly is a “user with limited technical knowledge”, this is also a subjective choice. Therefore, it is important to understand where the tester of the device stands, do the tester have limited technical knowledge or is the tester an expert in this field. The group can define “a user with limited technical knowledge” as an average

person that have no education within technology.

- A.6 Every decision taken by the user (prominently requested during setup) is necessary regarding the use of the device.
  The decision the user is asked about should be necessary regarding the use of the device. Here the tester will normally find a few question that ask about email subscriptions, test data, and more. It is preferred that these questions are asked separately from the setup process, and at least that the program makes it clear that these decisions is optional for the use of the device.
- A.7 Device look for and initiate updates when first enabled if not latest version.
  When the device is first initiated it should check for updates, normally the users will only be informed if an update is found, it will then ask for permission to download and update itself. You can normally find update information on the device manufacturer's web page, and some devices let users see what version the device in running on. If the tester are unable to find this information and no update is initiated when stating the device, the tester will be unable to evaluate this criterion.
- A.8 Easy to use and read support material for users with limited technical knowledge. In addition, the model designation should be easy to find by several different methods, i.e. on the product itself, on the box it came in or the app.
  An explanation around "users with limited technical knowledge" can be found in criterion A.5. The support material of the device should not be difficult to understand and read, in addition the model designation shall also be easy to find. The model designation is normally found on the product itself, the box or documents it came with or in the application.

4.6.2 B1: Authentication

- B.1.1 There is no indication that user input fields are vulnerable for injection-attacks and only users with right credentials are given proper access.
  To check if the input fields for the application is protected against attacks, you can try attacking the input fields by attempting to write in SQL, Java or other commands, to try and see if it is protected. The commands should not be executed, and the input field should not be able to recognize that it is a command. One must be careful when attempting such attacks, as a successful attack could have an impact on the manufacturer.
- B.1.2 Every data input validation method is effective for validating the corresponding data input. The validation does not provide an indication that any data input does not protect against the processing of unexpected data inputs.
  To check if the input fields for the device/app is protected against unexpected inputs and only takes the appropriate values. An example should be an email input, should only be able to take an email address. Testers can try

testing the input fields by giving the input SQL, Java or other commands, to try to see if it is protected or not. The commands should not be executed, and the input field should not be able to recognize that it is a command.

- B.1.3 Log-in mechanism is protected from brute-force attacks by making it impractical to execute due to;

  time delay between failed attempts, or
  limited number of failed attempts, or;
  required two factor authentication.

  To try to get into their account, but use the wrong password. Will users be able to continuously try new passwords, or will users be stopped by one of the mechanisms above? If the tester are not stopped the device will fail this criterion. It is important to note other mechanism that may interfere with the testing.
- B.1.4 Process for changing passwords is described in the privacy policy/user agreement and is easy to follow. The password is actually changed.
  If the process is not found in either the privacy policy or user agreement users should still try to change the password to see if it is easy to find the function, if the process easy to use and understand, and if the password actually changed after the process is done. All these questions are part of this criterion. If testers are not able to change the password, the device will completely fail this criterion. If the tester are able to change the password, but with problems or difficulties will the criterion not be fulfilled, the process should be easy to use and find.
- B.1.5 Users must be authenticated towards the device to have access to its functionalities and interfaces.
  Only a user with the right authentication should be able to use and access the device's functionalities and interfaces. This can be tested in numerous ways, but it can help using some of the tools previously introduced. Here have the group used "Kali", "Burp Suite" and a "Wireshark" to scan for functionalities that could have been used without authentication.
- B.1.6 Password is recommended to be at least 8 characters and consists of at least one character from each character group. (big letters, small letters and numbers).
  Try to either change password or create a new account with a purposeful bad password, examples are "ABC", "1234", "password", "AAAAA" and similar. If one of these is possible the device will fail this criterion.
- B.1.7 Administrator/user gets notified about new/unauthorized changes in device software.
  This criterion may be difficult to check, but if the tester are able to change the device setting without proper authorizations, then the device‘s controller and the user should get notified, through mail or the application.

4.6.3 B2: Security and communication

- B.2.1 Communication of personal and sensitive data use best practice cryptographic.
  Communication of personal and sensitive data can be monitored with tools like "Burp Suite", "Wireshark" and "Ettercap". These tools will let the tester monitor the communication theirs device is doing.
- B.2.2 All cryptographic details are configured appropriately, is not known to be vulnerable and considered best practice.
  These cryptographic details may be found on the internet, but most will not publish this information. To see if it is configured appropriately tools like "Burp Suite", "Wireshark" and "Ettercap" can be used.
- B.2.3 The IoT-device can be isolated.
  The device should be able to be isolated and still be able to use its function that are not depended on internet access.
- B.2.4 The level of security and mechanism used is appropriate for the use case of secure communication.
  Tools like "Burp Suite", "Wireshark" and "Ettercap" can be used to monitor the communication the testers device is doing. Here they can find out if the communication is encrypted, is changeable or more.

4.6.4 B3: Disclosing of vulnerability

- B.3.1 The vulnerability disclosure policy of the company/organization is available for anybody/user and contains contact information. (ETSI 5.2)
  The vulnerability disclosure policy of the company should be public and are normally found on their web page. It should also contain the contact information of the controller, for the users, if a vulnerability is found.
- B.3.2 The company/organization is also required to act upon vulnerabilities sent to the user contact in a timely manner (60-120 business days).
  The vulnerability policy of the company should be public and are normally found on their web page and in privacy policy or user agreement. It is also possible to search for earlier vulnerabilities too see how long it took for this company to fix the problem.

4.6.5 C1: Default password

- C.1.1 Default password is generated uniquely per device.
  If the device uses default password, the password must be unique for each device, the password can normally be found on the internet or directly on the device or application.
- C.1.2 Default password does not make use of common patterns or common strings and is not related to public information. Password is recommended to be at least 8 characters and consists of at least one character from each character group (big letters, small letters and numbers(special characters)).

If the device uses default password, the password should be secure and not follow common patterns, this is the same requirement for normal passwords in B.1.6. look at the password and see if it follow patters or is to simple.

4.6.6 C2: Erasure

- C.2.1 User have the right to require erasure of their personal data (GDPR 17).
  By asking for erasure the tester will be able to find out if the company are following GDPR article 17. Normally a specific email is used for these requests, found on their web page and privacy policy.
- C.2.2 When a user requests erasure of data, third-parties of the company that have the data, will also erase the data.
  It will be difficult for a single individual to check this criterion, the company shall have a description on how they use users data, including if any data is sold to third parties and how they will erase the data from the third parties. The erasure process will normally be in order if this is documented correctly and there is no indication they process the data wrongly.
- C.2.3 A clear confirmation is provided after deletions.
  After the user has asked for deletion of their personal data, a clear confirmation that their data have been deleted should be given. There should also be a confirmation about a received request if the deletion process is not automated.
- C.2.4 Users can make use of a simple functionality to erase their user data.
  The processes to delete the personal data should be easy to find and understand, it should be preferable for the users to be able to use a simple functionality to delete their data. For example, a simple request sent through the app should be preferred. By simply trying to find and go through the deletion process testers will be able to answer this criterion.
- C.2.5 Nothing indicates that the user data is not erased.
  On this criterion the testers will only look at their device and application. Look if the device and application still remembers information given earlier.
- C.2.6 Privacy policy or user agreement covers how to erase, or delete personal data.
  In the privacy policy and user agreement there should be described how to delete or erase the personal data off the users. This information should, as well as the reset of the documents, be written with clear language and be easy to understand for a person with limited technical knowledge. If the tester are unable to find this information, this will be a very bad sign for this device and company.

4.6.7 C3: Personal data

- C.3.1 The user is able to edit and complete wrong and incomplete data without undue delay.

If the user has any inputted any wrong data about themselves, they should be able to rectify it. Try to change this data, either through the application or mail.

- C.3.2 All decisions about the user's personal data are based on consent.
  This criterion is similar to "A.1: The process is based on consent. (GDPR 7)", in regard to this criterion being a general point and encompasses all processes where personal data is being used. Here it is important that all decision is understandable for the user, and it is clear what they are consenting to.
- C.3.3 Users are able to withdraw their consent to use personal data and the processing of personal data at any time. And the process to withdraw is easy to find and understand.
  User shall be able to withdraw consent from processes they have previously given consent to. It should be easy to accomplish and preferably through the application.
- C.3.4 The user has access to an easy to understand description about how personal data is being used, by whom, and for what purposes, as well as all processes their personal data may be used in.
  This information should be located in the privacy policy, but it may also be in the user agreement. This explanation should be easy to understand or direct the user to more reading material that will help them understand.
- C.3.5 The user is informed about how to express consent (opt-in choice) to the different processes their personal data may be a part of.
  This information shall be in the privacy or user policy, normally users will express consent by agreeing to these policy's and opt-out by email/deleting their user. Some companies will make it easier for users to opt-in and out and give the options in the application, often with yes or no sliders for each process.
- C.3.6 The user shall be provided with all the data the controller/company has of the user within one month from the receipt.
  The user should be able to request all data the controller have off the user, normally through mail. The "receipt" reference the time the controller receives the request. Users check this by request all their data and then looking through the data received. If no data is received the device will fail this criteria.
- C.3.7 The users shall be informed if data is transferred to a third country or an international organization.
  This information will be in the privacy policy, and if it's not the device will fail this criteria.
- C.3.8 Data collected is kept within GDPR-compliant countries, and preferably within Europe.
  The information about which countries is collecting data will be find by using C.3.7, and testers can check if these country's is GDPR-compliant by googling them. Most countries within Europe is GDPR-compliant and there-

fore is this preference specified in the criterion.

- C.3.9 Communication with the company is adequate for the request that is made. A confirmation of receival of a request is given.
  Good communication with the users is a good sign for a company, that the user gets a confirmation that the email has been received by the company and the request is being worked on is preferred. After sending a request to the company, see if they get a confirmation, are the confirmation from an employee or is it automatic. If the received confirmation is automatic will this be a poor indicator for this criterion, but if the answer is satisfactory may an automatic respond be good enough.
- C.3.10 Privacy policy, user agreement and other relevant documentation is easy to find and contains all relevant information to the user.
  This a general criterion that encompass a lot of the earlier criteria of C.3. The main point of C.3 is that the privacy policy and user agreement is used for its intended purpose and is easy to use, find, and understand. It should also not have any missing or poor parts.

4.6.8 D1: Updating and support

- D.1.1 Support period of the device is public available and the device is updates in a auspicious time after the sale starts.
  The tester will normally find the support period on the company's website, the support period should be “auspicious time after the sale starts”. “Auspicious time” is a time period that is sufficient to fix the major problems and vulnerabilities that may be detected in the early stage of the product's lifespan. This time will then naturally be longer for more complicated and expensive items. The time may range from a few months to a few years. This criterion will therefore be very subjective and differ from product to product.
- D.1.2 The device looks for updates and initiate updates when first enabled if there is a new update.
  This will normally happen when the users first connect user device to the server and users may get a notification that an update is available. This will not always be the case, with a new device where there has not been any update yet. This can be tested by spoofing the device acting as the server initializing the update, but this require that the spoofing is successful.
- D.1.3 Software updates are automatically and periodically checked and initiated. It is recommended to give the user the ability to manually check and install updates.
  It can be difficult to perfectly check this criterion, because if no updates have been made for the device at the time of testing this function will be unused. Users will normally be able to find information about such updates on the company web page. If an update is found the device should be able to notice this and update itself. It can also ask the user for permission or

a time it should update itself. There is also given a recommendation that the user should have the function to manually check and update the device, one of the reasons this is done is because of the security the user will feel knowing with full security that they have the latest version of their device software.

- D.1.4 The user is notified about critical security-updates.
  If a critical security-update is published the users of affected devices should be informed, so they can update the device as quickly as possible. Users can find information about update like this on the company web page or on the internet.
- D.1.5 The user is able to check the software version for the device.
  The software version of the device is information that is useful for the users to know, especially to see if their device have been updated if a update has been made or a vulnerability has been discovered. This is however not something every user will have a use for or understand, and therefore not every company will implement it.
- D.1.6 Everything enabled by default is necessary for the device.
  It will normally be many default decision and values already decides for the application and device, users can go through the application and see if any of default values is not necessary for the security use of the device. The tester should also check for open ports and their functionalities, this can be done by using "Nmap" from tools.
- D.1.7 Every debug interface enabled is required, and can be disabled.
  This criterion is checked with "Nmap" and if some debug interfaces is not disable, you can try to disable them in the application or directly on the device.

4.6.9 D2: Physical

- D.2.1 Device is made by adequate physical material for its use case.
  Here will “adequate physical material” mean that the material will not be easily broken into, either by mistake or with intent. If the device is to be used outdoors or public the requirement for "adequate material" will therefore rise. Hard plastic will be adequate for most use cases.
- D.2.2 All exposed interfaces (Physical) are covered, protected or enabled as required, and accessible physical debug interfaces are disabled.
  Any open USB or other “entrances” to the device is closed and covered if it is not meant to be used by the user.

Chapter 5

Conclusion

For this chapter of the thesis we will conclude and discuss our finding and achievements. It will be discussed to what extent we have obtained our goals for the project and how the final product solves the problem that has been presented. There will also be suggested further work and improvements to our product and the field of IoT security.

5.1 Results

As a result to the project, we have created a product and improved our understanding of IoT and cyber security as a whole. It is in this section discussed what we have achieved in regards to learning objectives and impact goals, in comparison to what is described in the bachelor-thesis subject page[1].

5.1.1 Learning objectives

During our work on the bachelor thesis, we improved our knowledge within IoT and risk management. Understanding how the IoT-devices worked, communicated utilizing tests, tools and searching for weaknesses gave us an understanding of how penetration-testing could be used to improve security. Analyzing and utilizing previous regulations the group constructed a framework of how to approach an evaluation of IoT-devices. Testing our own framework by using it on different devices, we improved our knowledge of the process of risk management and what considerations and criteria is important to archive a through evaluation.

5.1.2 Impact goals

The popularity of IoT-devices, and the practicality of an IoT-devices, makes the security of these devices, crucial for the security of its users. This framework is a step in the direction for more secure IoT-devices. There are many possibilities

[1] https://www.ntnu.no/studier/emner/DCSG2900/2021#tab=omEmnet

when it comes to security improvements for IoT-devices. However, security threats in IoT-devices are plentiful, for both the individual consumer and organizations.

A lot of IoT-devices includes, cameras, microphones. They ask for location, email, name, and more. The likes of email, name, and location goes under personal data. Some augmentation is possible to achieve, to keep privacy, but cameras, microphones and sensors that may be active are more difficult for a user to control, and may lead to consequences on a personal level.

On a social level there are a lot of bad consequences when it comes to security and IoT. An example of this is a botnet of IoT-devices, where owners may not know that their devices is part of an attack on important systems, or websites. An example of a botnet is the Mirai botnet, it is a known botnet that exploits security gaps in IoT-devices [51].

We have for this project taken the issues into consideration during development of the framework. We thing that our product could impact the field of IoT positively, by helping distributors of such devices, evaluate their level of security before distributing them to the consumers.

5.1.3 Product overview

The result of our work is a framework to test the security in IoT-devices, especially in regards to the processing of personal data. The framework includes criteria for authentication, personal data and communication. The framework contains a total of 57 different criteria dived into ten different tables.

- Registration
  - Set up user and device
- Security
  - Authentication
  - Communication
  - Disclosure of vulnerabilities
- Data
  - Default password
  - Erasure
  - Personal data
- Updates and interfaces
  - Updating and support
  - Physical
- Further testing

The last category includes criteria that does need special equipment, or software that we didn't have access to, but felt was important enough criteria to evaluate if able to do so.

Supporting the framework are documents that will help give a deeper understanding of process of evaluation. The "Tools.md" file introduces the tools and programs that has been used to perform the evaluation and gather data from the device that is being tested upon. The "Method.md" references specific criteria that is found in the framework and gives a further understanding of how to test and what to look when performing the test. Supporting the framework, there are also found example of completed evaluation of commonly found IoT-devices. This is made for the future evaluators to see what the result of a final evaluation look like and give insight to how the evaluation process is performed.

5.2 Discussion of processeseth and mods

In this section the final product as well as the different implications and alternative methods are discussed.

5.2.1 Final product

We are happy with the product that we have created. Based on feedback from employer the framework would make a beneficial tool for evaluating IoT-devices and make the process efficient. Another part of what we are happy about is that the product will be publicly available on "GitHub". The group started with the first stage of the first round of research in late January, and worked through numerous rounds of improvements.

5.2.2 Implications

During the development of the framework, we had many ideas, and solutions on different stages of the development. Below are the possible alternatives that could have made it to the final version, if we chose another alternative in the development.

Alternative scoring system

From the beginning of the development to the final versions of the framework, there was presented several different suggestions of a scoring system, where some of them was suggestions thought out by the group it self. This is also one of the reasons that they was not used in the final product as there was little to no background or previous tests of the formulas. The thought of creating formulas as a method of scoring the vulnerability of the device, came from how CVE evaluates vulnerabilities based on Common Vulnerability Scoring System (CVSS) [52]. These formulas are:

- $consequence * implementation/10$: 2,6

This value is calculated with putting every criteria through the formula, and than adding them together. The initial thought was that every score over 1 should fail the test, but the margin for failure was found to be too small for the devices in today's market.

- $(consequence^2 * implementation^2)/100$: 1,08

  This method of calculating is done by the same method as the first way, by calculating every individual criterion, and then calculating all the scores together to represent the device. This method was not chosen because it was found to be a bit to sensitive, and not really that representative for the devices security. As an example will a criteria for a device with the "value of consequence" as three, and "value of implementation" as two get a 0.36, but a score as a "value of consequence" at four, and "value of implementation" as one be scored to 0.16. This represent too big of a difference for 1 and 2 as a value, that is set as a subjectively based on requirements. The thought behind the formula was to fail any score over 1, and therefore would every device with no implementation of a serious criteria, valued to 5 as consequence and 2 as implementation, fail.

- $Consequence * implementation^2/50$:

  In this formula was the goal to make a bit more room for failure then the first tries, but this time was the limit a bit too big, and would open up possibilities for two bit whole, valued $to5(consequence)and2(implementation)$, without a failed score.

- Two scores, one for GDPR and one for Security

  The idea of making two different scores, by using one of the formulas mentioned earlier, was an idea to check if a device fulfilled the GDPR laws, and the security criteria in the framework. This idea was abandoned when the risk matrix was implemented. The thinking behind the idea were to calculate how much GDPR and security were taken into consideration in the development of the device.

These formulas was not taken into the latest version of the framework for many reasons, firstly because they are self-made, secondly the versions did not work with today's market of devices. The CVE scoring system goes from zero to ten, and none of the formula really showed how the evaluation of the devices where thought to be either to easy to the devices, or to hard on the device. The reason the idea of two scores were dropped when risk matrix was taken inn, is because the risk matrix do show the weakest criteria, and thereby much of the same image as the GDPR and security scores show.

Alternate risk matrix

After many attempts of making a formula to represent a device's security, the scoring system ended up being a risk matrix to show the points of weaknesses with a device. When the decision were made to go for a risk matrix, there was two sizes that had potential, 4x4 and 5x5 [35]. The final decision was to go for the 4x4 size for a one big reason. The "value of implementation" would be divided into the requirements; "Implemented after best practice", "Implemented with some flaws", "Implemented with moderate flaws", "Implemented with many flaws" and "No implementation". The first reason for not using this size is because the differences between "some flaws", and "moderate flaws" were deemed to be to small, and thereby become more subjective in its evaluation then the smaller sizes. In the same way would the differences between "moderate flaws" and "many flaws" become to small to justify the middle value.

By dividing into a 4x4 risk matrix, the "value of implementation" could be divided into the requirements "implemented according to best practice", "implemented with some flaws", "implemented with many flaws", and "not implemented". In such requirements can a user of the framework check if it is implemented, if it is implemented after best practice, or how many the number of flaws in the implementation is acceptable or worrying.

The reason it was up for debate was because the framework had already implemented the 5 "value of consequence" into its devices, and thereby needed an evaluation of the size. There is an understanding of 5x5, and the practicalities of using it, as seen in the project plan, but it was deemed to be too big, and open up for too many differences into the valuation of an implementation.

Alternate methodologies

There were other options available as methods of work, including the waterfall and agile. Waterfall method was never an option because we wanted to work iterative and the waterfall method is a linear methodology. This would have meant that once we finished the framework and possible errors were found we could not just go back an fix them. Another option was agile methodology. Agile is very flexible and has little structure. This meant that because we wanted more structure, but at the same time allow for some flexibility agile would not have worked as well as unified process. The reason for wanting structure was because this was the first major project any of us had undertaken. If the group had been more experienced with such large project agile would possibly have been selected instead of unified process[53, 54].

5.2.3 Design

There were a couple of design versions up for debate for the final framework. Two of these ideas were to put it all in one table and reasoning for the values after the scoring system. The one table design would look a bit like the latest versions of the framework. With this design, there would be a need for a table of content before the actual framework, to differentiate the different parts in the framework, another solution that would be possible is to change the "Nr." column to a "theme" column. The biggest problem with this solution would be that it would quite a bit harder to link a criteria to other part, due to a lack of "identification" for each criteria.

5.2.4 Testing

When testing the devices against the framework, there were a lot of different ways to do it. There were possibilities with physical testing, but there was some reasons why this was not the chosen testing method. First of all is time, we had two months of testing, and making the framework. This meant that one method of testing had to be chosen, and testing of the communication seemed like best way to use the time we had.

5.2.5 Why did we go for the version we went for?

The scoring system we ended on using in the framework is a widely used method for evaluating risk, and risk analysis. A risk matrix is a known method of evaluation for the group and gives a easy and readable way to show the level of risk. A risk matrix also gives a clear picture on what criteria and aspects of the device that needs improvement.

The design of the framework is based on what we felt is a logical setup and is inspired by the structure found in ETSI and GDPR. The reason we designed the framework as we did was because it gives a user a clear way to progress from table to table, read the theme in the framework, and easily give a valuation and move on.

When we had to test the framework, and test the device that we had available, we focused on testing the communication from the devices, including looking up where servers were located based on IP addresses, checking for encryption and open ports.

5.2.6 Technical implications

The thesis did not end up using the Vancouver method for writing references as planned. As strongly recommended by NTNU, the group ended up using a number system that was part of a BibLaTeX package available in LaTeX. The number system utilizes numbers in the text that correspond with the number in the reference list.

5.2.7 Challenges

The subject of IoT and securing devices, was mostly a new subject to us, when starting the project. Especially learning to search for, finding and make use of vulnerabilities in devices to perform an attack would prove to be a big challenge with a lot to learn. Even with lots of research and documented proof of concepts for attacks, there are many conditions that must be right to perform such attacks. We would during research and testing, familiarize ourselves with tools and resources that could make the process of scanning the device for attack-vectors.

Succeeding performing and getting control of IoT-devices and finding specific attack-vectors would require us gaining more knowledge of how specific protocols of IoT-devices work and can be used. This would enable us documenting more in-depth methods of what weaknesses was found. Had we had the knowledge we have now, we would probably follow a methodology more similar to vulnerability-hunting and base search for vulnerabilities on resources such as OWASP TOP 10[2].

During development of the framework we had to make decisions on what criteria should be included in the framework to make an adequate evaluation for testing IoT-devices security. This meant familiarizing ourselves in the field of IoT and understanding what vectors could be vulnerable and what consequences such vulnerabilities could have. Researching, experimenting, testing and gathering of knowledge of a field with few standards ended up being a big task. Basing our framework on current regulations was therefore necessary to be able to include the most important aspects of security and not leave out criteria that is essential to the security of the user. At the same time we did not want to make a copy of, or create regulations that works in parallel to current regulations, as our framework then would be of little use. As stated in our task, our goal was to create a framework that stores and distributors of devices, could use to confirm the security of their customers and consumers to evaluate their security when using the devices. Including all criteria that GDPR or ETSI require in our framework, would not be a feasible way to evaluate the devices.

5.3 Limitations of work

In this section we critique our own project, from start to finish. This is so that we know what we did wrong, and how to change and prevent the wrongdoings for future projects.

5.3.1 Limitations of final product

When evaluating an IoT-device and assigning a “value of implementation” to each criteria the value that is assigned will be mostly subjective to the individual’s experience and knowledge within the field. As the framework is targeted for people with limited knowledge of cybersecurity, the value assigned might vary from person to person. If an evaluator does not know how a criteria should be implemented or what best practice is or the capabilities of threat actors, certain criteria might not be assigned the optimal value of implementation. Fixing this, and making

[2] `https://owasp.org/`

the process of evaluation more objective would require parameters for each criteria, describing a scenario of what each value would look like. Implementing this would however take a lot of time and work for a result that would arguably still be subjective by the end.

The “value of consequence” is a value that will vary from product to product, but that we in our framework have set based on our experience of what we think the consequence could be. This value, even though defined in the framework, is an aspect of the framework that does require further iterations, more discussions and debates as well as more data.

The current methodology for testing the devices for vulnerabilities and weaknesses is currently just a collection of tools and how one can use them to check and evaluate each criteria. However, in retrospect, it is clear that this methodology should have been based upon how penetration-testers would go about to expose these devices for weaknesses. Following this methodology, the results of the tests would be more thorough and attack-vectors on the device would be more clearly presented. Starting the evaluation-process by creating a threat-model for the device, would make the process of identifying possible threats easier. Recognizing and rating the threats that are identified will also help finding countermeasures of how to improve such device and help improving the security where it is needed [55]. If we had used the "DREAD" system as a method to evaluate we would gain a score for each treat based on; Damage potential, Reproducibility, Exploitability, Affected users and Discoverability [56].

The criteria that are found in "E: Further tests that require specialized knowledge", are further tests that we found could be useful to find certain vulnerabilities within a device, but that we did not have the tools, experience or knowledge to test. The criteria are mentioned as potential future work on the framework.

Even though we targeted this framework towards people with limited technical knowledge, we recognise that our introduction and guide to the tools utilized to testing, is not adequate for this kind of user. The explanation of the tools lack a descriptions of how the tools work, how to interpret outputs and how to use them. As it stands now, it functions as an introduction to what they are for, and points to relevant resources.

5.3.2 Limitations of methodology

Our workflow was somewhat slower compared to what we had planned in our GANTT diagram, this was however expected as this task would require some shifts in attention and focus on parts of the task where more time was needed. We believe that choosing unified process allowed us to allocate time and resources as well as iterate processes was the right choice of method. Unified process was helpful with policing the framework, something we may not have had the opportunity with other methods.

When doing research for methods on testing devices for vulnerabilities, we should have implemented a method for sharing, comparing and peer-reviewing what we had found. For example once a week going through what we had done and tested could have improved learning outcome, motivation and would more easily inspire each other. Instead what we did, was testing out different methods for performing tests and moved on without discussing why it did not work and if there were any changes we could do. We could also have streamlined the testing phase more, it could have been more efficient that one person looked for a certain sets of vulnerabilities, and not all on the devices. So the testing would look more like a line that the IoT unit goes through. The downside with that is that each group member becomes highly specialized and misses out on learning how to do the other tests and research. In addition all of the group members should have had access to a machine with Kali installed on it, because on of the bottlenecks was that we had to contact and wait for the group member that had "Kali" so that we could run "Nmap" and port scan. In hindsight, we could have perhaps utilized our task giver more, and perhaps he could have supplied us with USB-flashdrives that we could have installed "Kali" on.

5.4 Future work

The framework is still a work in progress and there are many areas of the field that are possible to integrate into it. The areas that we would have continued on with more time would probably be a continuation of the usage with "Binwalk" and "Ettercap", to review the firmware of the device and observe what information is located on the device itself. How to best perform code-review of IoT-devices, would in itself be a possibility for a bachelor thesis for future students.

Further scope of the framework would also include devices that use Zigbee and Z-Wave as their primary method of communication. Figuring out what differences there are in regards to communication and what attack vectors can be used to expose the devices.

We have in our framework not taken any "end-user programming" into consideration of the evaluation. If-This-Then-That (IFTTT) is one such tool to control devices and make new functionalities based on actions that are performed when a parameter is fulfilled. As discussed in [57], such functionalities comes with the cost of security. It was found that 50% of IFTTT-rules, called *recipes*, implemented was potentially unsafe as they contained a security violation, integrity violation or both. How producers of IoT-devices was implemented has not been part of this thesis, but we see this as a possible thesis to create a method of evaluating such functionality.

5.5 Evaluation of our work

In this section the evaluation of the work done is presented and discussed. The section includes how the work was organized, work was allocated and project as a work method.

5.5.1 Introduction

The group dynamic was great because of the number of participants. It was a great help to us that we were four people, because it meant that we were not that affected by illness or other reasons for not turning up. It also helped us by having another angle to look at problems from. The product we created was a framework that should help people test IoT-devices, both more effectively and easy.

5.5.2 Organization

The way we organized the group for the project was by having daily meetings, either held digitally or in person. We created a GANTT chart to keep track of deadlines, and when to start the next set of tests. In addition, we kept a log to keep track of all the hours we used. The log has information about time spent, and on what. The log tracks every day, and helped us keep the target for number of hours we should spend every week. The number of hours was to be approximately 30 hours per person per week. We averaged around 25+ a week. The time spent every week varied because of how much we had to do, the first and last weeks was the weeks we worked the most, to see more see appendix - Miscellaneous. In the first weeks we also had weekly meetings with both of our supervisor and task giver. Those meetings were held either via "Google Meets" or "Microsoft Teams". The reasoning behind why we used to different video calling software was because the firs meeting we had with our task giver in December was via Google Meets, and that was a call that he had set up, so we decided to use the same for contacting him. We used "Microsoft Teams" with our supervisor because that is the standard video calling platform that NTNU uses along with "Zoom". In addition to that we created a Slack channel so that we could easier communicate with our task giver, that was something he recommended we do.

5.5.3 Allocation of work

The thesis was large and over a long period of time, so to keep track of everything we created a GANTT chart to make it easier to reach the deadlines. In addition to a GANTT chart we also kept a time log of our working hours. This helped us also reach roughly the recommended hours to work every week. The last way to help us keep track was that every document was kept either on a shared "Google Drive" or on our "GitHub". This meant that all of our members had access to everything they needed everywhere they where.

5.5.4 Project as a work method

Project as a work method is something that we were familiar with. We have had many courses that have included large projects resulting in a report. The course “PROG1004: Programvareutvikling" had a very large project that could in some way be an introduction to a Bachelor’s thesis. It did not have the same requirements, but the length of the report and some of its content was partially equal to a Bachelor’s thesis. It also creates a more realistic scenario on how to work in the real world, as opposed to having to attend classes.

One way for the group to keep track of progress was milestones. The milestones were:

- Project plan
- First version of the framework
- Beginning testing the IoT-devices with the framework
- Final version of the framework
- Beginning the project thesis
- Final version of the project thesis

How we reached the milestones were by different methods. Each milestone were reached, but there were changes to both the timeline. For example the beginning of the project thesis was pushed back some time. The way we went about organizing the group was that everyone got to choose an IoT-devices to test. We decided that the units we got to choose from was a more basic type of IoT-device, like a smart plug or camera, and that we should tackle the more complicated devices together, a robot vacuum. We gave each other a lot of freedom to do these tasks, the only requirement was that we should be done testing by a certain date.

For the writing of the thesis we also divided the tasks between us, and we stood freely to choose what we wanted to write about. This freedom of choice was great, and it helped our morale and productivity for such a big project.

5.6 General conclusion

Securing IoT-devices is a challenge for both the producers and consumers. The field of IoT has a huge attack-surface with lots of different vectors. An efficient and thorough method of evaluating the security of such devices is in itself a challenge. Our framework is just the beginning and foundation of such a tool to help assess the security. The framework is constructed in a way and is available for others to use and append further criteria, methods or tools that would fulfill their need within a greater scope.

As we built the framework we included criteria that we felt would be necessary to the evaluation and excluded criteria that we ended up being difficult or unnecessary to test. As our knowledge grew, we expanded the framework with further methods and descriptions of how specified tools were utilized. Learning how to test the devices, we would do research on successful penetrations-tests and try to perform them. None of our tests were successful to a point, where we had control over it, but we would find responses and reactions of interest, from the device. The tests that gave us the most concrete results were the manual tests performed towards the application of the device. Attempting to manually brute-force the passwords to check for any prevention or observe what default functionalities were on, gave a clear indication of how security had been taken into consideration by the producer. We were also happy with the results the tests that utilized "Nmap". The result gave us a technical insight to the device and showed us a little bit of what services was running under the hood of the device.

As well as testing the general security of IoT-devices, we have compared how data has been collected and processed with the principles of "GDPR". We have through development of the framework and attempts at exercising our rights of knowing what information is kept, familiarized ourselves with what rights and regulations corporations must follow.

There is potential for future work, iterations and expansions of such a framework we have made. The scope has the potential to be expanded into further specializations if there are other requirements that must be fulfilled. This could be further criteria for testing specific products such as smart-locks or wearable health trackers and go further into their functionalities.

The task performed has covered a wide area of the cybersecurity space. We have touched upon risk management, network communication, penetration testing, GDPR and privacy control as well as framework development. We have in this project applied knowledge gathered from our time of studying, as well as new knowledge from the work with the project. Exploring these new aspects and fields within cybersecurity and working with practical problems has been an informative and useful experience. Performing hands-on testing and attempting attacks towards devices has been a fun and challenging project. We are very fond with the opportunity of the task that was provided and we hope that the product we have produce will be of use.

Bibliography

[1] jerry Zeyu Gao, H.-S. J. Tsao and Y. Wu, *Testing and Quality Assurance for Component-based Software*. Artech House, 2003, ISBN: 9781580537353.

[2] T. W. Edgar and D. O. Manz, *Research Methods for Cyber Security*. Syngress Media,U.S., 2017, ISBN: 978-0-12-805349-2.

[3] Z. Zhu, G. Lu, Y. Chen, Z. J. Fu, P. Roberts and K. Han, 'Botnet research survey,' pp. 967–972, 2008. DOI: 10.1109/COMPSAC.2008.205.

[4] 'General data protection regulation.' (2022), [Online]. Available: https://gdpr-info.eu/ (visited on 22/04/2022).

[5] 'Cyberspace.' (2015), [Online]. Available: https://csrc.nist.gov/glossary/term/cyberspace (visited on 09/05/2022).

[6] 'What is devops?' (), [Online]. Available: https://aws.amazon.com/devops/what-is-devops/ (visited on 20/05/2022).

[7] 'What is gdpr, the eu's new data protection law?' (2022), [Online]. Available: https://gdpr.eu/what-is-gdpr/ (visited on 28/04/2022).

[8] 'Imrad - how to structure your text.' (), [Online]. Available: https://www.ntnu.edu/sekom/imrad (visited on 19/05/2022).

[9] L. Irwin. 'The gdpr: What is sensitive personal data?' (2020), [Online]. Available: https://www.itgovernance.eu/blog/en/the-gdpr-what-is-sensitive-personal-data (visited on 19/05/2022).

[10] 'Cyber security for consumer internet of things: Conformance assessment of baseline requirements,' 2021. [Online]. Available: https://www.etsi.org/deliver/etsi_ts/103700_103799/103701/01.01.01_60/ts_103701v010101p.pdf.

[11] 'Readme: Key details at a glance – including a template.' (2020), [Online]. Available: https://www.ionos.com/digitalguide/websites/web-development/readme-file/ (visited on 19/05/2022).

[12] 'Reverse engineer.' (), [Online]. Available: https://www.merriam-webster.com/dictionary/reverse%5C%20engineer (visited on 19/05/2022).

[13] S. Bliim. 'What is sensitive data? sensitive data definition & types.' (2020), [Online]. Available: https://www.cpprotect.com/blog/what-is-sensitive-data/ (visited on 19/05/2022).

[14] K. Jindal, S. Dalal and K. K. Sharma, 'Analyzing spoofing attacks in wireless networks,' pp. 398–402, 2014. DOI: 10.1109/ACCT.2014.46.

[15] 'Threat actor.' (), [Online]. Available: https://csrc.nist.gov/glossary/term/threat_actor (visited on 19/05/2022).

[16] 'Z-wave.' (), [Online]. Available: https://www.z-wave.com/ (visited on 19/05/2022).

[17] 'The full-stack solution for all smart devices.' (), [Online]. Available: https://csa-iot.org/all-solutions/zigbee/ (visited on 19/05/2022).

[18] S. Sinha. 'State of iot 2021: Number of connected iot devices growing 9% to 12.3 billion globally, cellular iot now surpassing 2 billion.' (2021), [Online]. Available: https://iot-analytics.com/number-connected-iot-devices/ (visited on 06/04/2022).

[19] M. J. Farooq and Q. Zhu, 'Iot supply chain security: Overview, challenges, and the road ahead,' 2019.

[20] S. Ikeda. 'Iot-based ddos attacks are growing and making use of common vulnerabilities.' (2020), [Online]. Available: https://www.cpomagazine.com/cyber-security/iot-based-ddos-attacks-are-growing-and-making-use-of-common-vulnerabilities/ (visited on 06/04/2022).

[21] 'What is cybersecurity.' (), [Online]. Available: https://www.ibm.com/topics/cybersecurity (visited on 07/04/2022).

[22] K. Cabaj, Z. Kotulski, B. Księżopolski and W. Mazurczyk, 'Cybersecurity: Trends, issues, and challenges,' *EURASIP Journal on Information Security*, no. 10, 2018.

[23] C. Vladescu, M.-A. Dinisor, O. Grigorescu, D. Corlatescu, C. Sandescu and M. Dascalu, 'What are the latest cybersecurity trends? a case study grounded in language models,' *2021 23rd International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC)*, pp. 140–146, 2021.

[24] '*privacy not included.' (2022), [Online]. Available: https://foundation.mozilla.org/en/privacynotincluded/ (visited on 25/04/2022).

[25] 'European union agency for cybersecurity; Secure supply chain for iot,' 2020. [Online]. Available: https://www.enisa.europa.eu/publications/guidelines-for-securing-the-internet-of-things.

[26] R. Achkoudir and Z. Alsaadi, 'Ethical hacking of a smart plug,' 2021.

[27] C. Robberts and J. Toft, 'Finding vulnerabilities in iot devices; ethical hacking of electronic locks,' 2021.

[28] W. Chai. 'Confidentiality, integrity and availability (cia triad).' (2021), [Online]. Available: https://www.techtarget.com/whatis/definition/Confidentiality-integrity-and-availability-CIA (visited on 29/04/2022).

[29] 'What is iot architecture?' (), [Online]. Available: https://www.mongodb.com/cloud-explained/iot-architecture (visited on 06/04/2022).

[30] M. binti Mohamad Noor and W. H. Hassan, 'Current research on internet of things (iot) security: A survey,' *Computer Networks*, vol. 148, pp. 283–294, 2018.

[31] S. Naveen and S. G. Hegde, 'Study of iot: Understanding iot architecture, applications, issues and challenges,' *International Journal of Advanced Networking & Applications*, pp. 477–482, 2016.

[32] 'Architecture of internet of things (iot).' (2020), [Online]. Available: https://www.geeksforgeeks.org/architecture-of-internet-of-things-iot/ (visited on 06/04/2022).

[33] 'Risk management framework for information systems and organizations: A system life cycle approach for security and privacy,' 2018. DOI: https://doi.org/10.6028/NIST.SP.800-37r2.

[34] 'Veileder i sikkerhetsstyring.' (2020), [Online]. Available: https://nsm.no/regelverk-og-hjelp/veiledere-og-handboker-til-sikkerhetsloven/veileder-i-sikkerhetsstyring/om-denne-veilederen/ (visited on 27/04/2022).

[35] D. Boers. 'Beyond the risk matrix.' (2017), [Online]. Available: https://www.armsreliability.com/page/resources/blog/beyond-the-risk-matrix (visited on 09/05/2022).

[36] 'Security architecture design phase: The concept of a threat intelligence driven defendable architecture.' (2020), [Online]. Available: https://www.telenor.com/security-architecture-design-phase-the-concept-of-a-threat-intelligence-driven-defendable-architecture/ (visited on 28/04/2022).

[37] S. Elhadi, A. Marzak, N. Sael and S. Merzouk, 'Comparative study of iot protocols,' 2018. [Online]. Available: http://dx.doi.org/10.2139/ssrn.3186315 (visited on 06/05/2022).

[38] G. M. Honti and J. Abonyi, 'A review of semantic sensor technologies in internet of things architectures,' vol. 2019, 2019. [Online]. Available: https://doi.org/10.1155/2019/6473160 (visited on 06/05/2022).

[39] 'The role of wifi in iot.' (2020), [Online]. Available: https://www.iotforall.com/wifi-role-iot (visited on 06/05/2022).

[40] C. M. Ramya, M. Shanmugaraj and R. Prabakaran, 'Study on zigbee technology,' vol. 6, pp. 297–301, 2011. (visited on 06/05/2022).

[41] D. Soni and A. Makwana, 'A survey on mqtt: A protocol of internet of things (iot),' 2017. (visited on 06/05/2022).

[42] I. Unwala, Z. Taqvi and J. Lu, 'Thread: An iot protocol,' pp. 161–167, 2018. (visited on 06/05/2022).

[43] S. Arvind and V. A. Narayanan, 'An overview of security in coap: Attack and analysis,' pp. 655–660, 2019. (visited on 28/04/2022).

[44] R. A. Rahman and B. Shah, 'Security analysis of iot protocols: A focus in coap,' 2016. (visited on 06/05/2022).

[45] S. Bendel, T. Springer, D. Schuster, A. Schill, R. Ackermann and M. Ameling, 'A service infrastructure for the internet of things based on xmpp,' pp. 385–388, 2013.

[46] 'An overview of xmpp.' (2021), [Online]. Available: https://xmpp.org/about/technology-overview/ (visited on 02/05/2022).

[47] 'Mqtt: The standard for iot messaging.' (2022), [Online]. Available: https://mqtt.org/ (visited on 06/04/2022).

[48] 'Consumer iot security.' (2022), [Online]. Available: https://www.etsi.org/technologies/consumer-iot-security (visited on 10/05/2022).

[49] S. Ravoof. 'Gitlab vs github: Explore their major differences and similarities.' (2022), [Online]. Available: https://kinsta.com/blog/gitlab-vs-github/ (visited on 26/04/2022).

[50] S. Vaughan-Nichols. 'Github vs gitlab: The key differences.' (2022), [Online]. Available: https://www.zdnet.com/article/github-vs-gitlab-the-key-differences/ (visited on 26/04/2022).

[51] G. Kambourakis, C. Kolias and A. Stavrou, 'The mirai botnet and the iot zombie armies,' pp. 267–272, 2017. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8170867 (visited on 22/04/2022).

[52] 'Cve vulnerability.' (), [Online]. Available: https://www.imperva.com/learn/application-security/cve-cvss-vulnerability/ (visited on 18/05/2022).

[53] E. Kavlakoglu. 'Agile vs. waterfall.' (2020), [Online]. Available: https://www.ibm.com/cloud/blog/agile-vs-waterfall (visited on 20/05/2022).

[54] T. Barenscheer. 'What is the difference between agile and unified process methodology?' (), [Online]. Available: https://www.teamly.com/blog/difference-between-agile-and-unified-process-methodology/ (visited on 20/05/2022).

[55] A. Seeam, O. S. Ogbeh, S. Guness and X. Bellekens, 'Threat modeling and security issues for the internet of things,' *2019 Conference on Next Generation Computing Applications (NextComp)*, pp. 1–8, 2019.

[56] V. Jagannathan. 'Threat modeling; architecting & designing with security in mind.' (), [Online]. Available: https://owasp.org/www-pdf-archive/AdvancedThreatModeling.pdf (visited on 19/04/2022).

[57] M. Surbatovich, J. Aljuraidan, L. Bauer, A. Das and L. Jia, 'Some recipes can do more than spoil your appetite: Analyzing the security and privacy risks of ifttt recipes,' pp. 1501–1510, 2017.

[58] ‘Usb til seriell.’ (), [Online]. Available: https://www.elkim.no/produkt/usb-til-seriell-6pin-usb-til-ttl-uart-serial-converter-cp2102-stc-replace-ft232/?utm_source=Google2020Shopping&utm_campaign=Elkim20Norge&utm_medium=cpc&utm_term=6343&gclid=CjwKCAiAl-6PBhBCEiwAc2GOVKr6oqcubMKySnYLVO48WmnK0m5_rTvQLUsFNGLZR3u0NM1aUVsFMRoC6CgQAvD_BwE (visited on 22/04/2022).

Appendix

Appendix A

Appendix - Framework

A.1 Dlink camera evaluation

papers/Framework2/Dlink_camera.pdf

papers/Framework2/Dlink_camera.pdf

papers/Framework2/Dlink_camera.pdf

papers/Framework2/Dlink_camera.pdf

papers/Framework2/Dlink_camera.pdf

papers/Framework2/Dlink_camera.pdf

papers/Framework2/Dlink_camera.pdf

papers/Framework2/Dlink_camera.pdf

A.2 Ledvance plug evaluation

papers/Framework2/Ledvance.pdf

papers/Framework2/Ledvance.pdf

papers/Framework2/Ledvance.pdf

papers/Framework2/Ledvance.pdf

papers/Framework2/Ledvance.pdf

papers/Framework2/Ledvance.pdf

papers/Framework2/Ledvance.pdf

A.3 Ecovacs evaluation

papers/Framework2/Ecovacs.pdf

papers/Framework2/Ecovacs.pdf

papers/Framework2/Ecovacs.pdf

papers/Framework2/Ecovacs.pdf

papers/Framework2/Ecovacs.pdf

papers/Framework2/Ecovacs.pdf

A.4 Framework 0.5

figures/Framework_developement/Whole_frameworks/Rammeverk_0.7.pdf

figures/Framework_developement/Whole_frameworks/Rammeverk_0.7.pdf

figures/Framework_developement/Whole_frameworks/Rammeverk_0.7.pdf

figures/Framework_developement/Whole_frameworks/Rammeverk_0.7.pdf

figures/Framework_developement/Whole_frameworks/Rammeverk_0.7.pdf

figures/Framework_developement/Whole_frameworks/Rammeverk_0.7.pdf

figures/Framework_developement/Whole_frameworks/Rammeverk_0.7.pdf

figures/Framework_developement/Whole_frameworks/Rammeverk_0.7.pdf

figures/Framework_developement/Whole_frameworks/Rammeverk_0.7.pdf

figures/Framework_developement/Whole_frameworks/Rammeverk_0.7.pdf

figures/Framework_developement/Whole_frameworks/Rammeverk_0.7.pdf

figures/Framework_developement/Whole_frameworks/Rammeverk_0.7.pdf

Appendix B

Appendix - Pre project

B.1 Project Agreement

appendices/Prosjektavtale gruppe 120.pdf

appendices/Prosjektavtale gruppe 120.pdf

appendices/Prosjektavtale gruppe 120.pdf

appendices/Prosjektavtale gruppe 120.pdf

appendices/Prosjektavtale gruppe 120.pdf

appendices/Prosjektavtale gruppe 120.pdf

B.2 Project Plan

appendices/Prosjektplan.pdf

appendices/Prosjektplan.pdf

appendices/Prosjektplan.pdf

appendices/Prosjektplan.pdf

appendices/Prosjektplan.pdf

appendices/Prosjektplan.pdf

appendices/Prosjektplan.pdf

appendices/Prosjektplan.pdf

appendices/Prosjektplan.pdf

appendices/Prosjektplan.pdf

appendices/Prosjektplan.pdf

appendices/Prosjektplan.pdf

Appendix C

Appendix - Miscellaneous

C.1 GANTT

figures/Gantt.pdf

figures/Gantt.pdf

C.2 Meeting minutes

papers/Meetings.pdf

papers/Meetings.pdf

papers/Meetings.pdf

papers/Meetings.pdf

papers/Meetings.pdf

papers/Meetings.pdf

papers/Meetings.pdf

papers/Meetings.pdf

papers/Meetings.pdf

papers/Meetings.pdf

papers/Meetings.pdf

papers/Meetings.pdf

papers/Meetings.pdf

papers/Meetings.pdf

papers/Meetings.pdf

C.3 Time log

papers/time table.pdf

papers/time table.pdf

papers/time table.pdf

papers/time table.pdf

papers/time table.pdf

papers/time table.pdf

papers/time table.pdf

papers/time table.pdf