

# SECURITY AND PRIVACY ASSESSMENT FOR MEDICAL TECHNICAL DEVICES

A Playbook for Evaluating Cybersecurity and Privacy

MARI MARTINI

SUPERVISOR  
Christian Auby

**University of Agder, 2023**  
Faculty of Engineering and Science  
Department of Information and Communication Technology

Master

## Obligatorisk egenerklæring

Den enkelte student er selv ansvarlig for å sette seg inn i hva som er lovlige hjelpemidler, retningslinjer for bruk av disse og regler om kildebruk. Erklæringen skal bevisstgjøre studentene på deres ansvar og hvilke konsekvenser fusk kan medføre. Manglende erklæring fritar ikke studentene fra sitt ansvar.

1.	Jeg erklærer herved at min besvarelse er mitt eget arbeid, og at jeg ikke har brukt andre kilder eller har mottatt annen hjelp enn det som er nevnt i besvarelsen.	Ja
2.	<b>Jeg erklærer videre at denne besvarelsen:</b> <ul style="list-style-type: none"><li>• Ikke har vært brukt til annen eksamen ved annen avdeling/universitet/høgskole innenlands eller utenlands.</li><li>• Ikke refererer til andres arbeid uten at det er oppgitt.</li><li>• Ikke refererer til eget tidligere arbeid uten at det er oppgitt.</li><li>• Har alle referansene oppgitt i litteraturlisten.</li><li>• Ikke er en kopi, duplikat eller avskrift av andres arbeid eller besvarelse.</li></ul>	Ja
3.	Jeg er kjent med at brudd på ovennevnte er å betrakte som fusk og kan medføre annullering av eksamen og utestengelse fra universiteter og høgskoler i Norge, jf. Universitets- og høgskoleloven §§4-7 og 4-8 og Forskrift om eksamen §§ 31.	Ja
4.	Jeg er kjent med at alle innleverte oppgaver kan bli plagiatkontrollert.	Ja
5.	Vi er kjent med at Universitetet i Agder vil behandle alle saker hvor det forligger mistanke om fusk etter høgskolens retningslinjer for behandling av saker om fusk.	Ja
6.	Jeg har satt oss inn i regler og retningslinjer i bruk av kilder og referanser på biblioteket sine nettsider.	Ja

## Publiseringsavtale

Fullmakt til elektronisk publisering av oppgaven Forfatter(ne) har opphavsrett til oppgaven. Det betyr blant annet enerett til å gjøre verket tilgjengelig for allmennheten (Åndsverkloven. §2).

Oppgaver som er unntatt offentlighet eller taushetsbelagt/konfidensiell vil ikke bli publisert.

Vi gir herved Universitetet i Agder en vederlagsfri rett til å gjøre oppgaven tilgjengelig for elektronisk publisering:	Ja
Er oppgaven båndlagt (konfidensiell)?	Nei
Er oppgaven unntatt offentlighet?	Nei

# Acknowledgements

I would like to thank Christian Auby, my supervisor from the University of Agder, for his invaluable guidance, expertise, and support throughout the entire research process. His insightful feedback and dedication to this project have been invaluable in shaping the direction and quality of this thesis.

I would also like to express my appreciation to Sindre Gjelsten, my supervisor and contact person from Sykehuspartner. His extensive knowledge and experience in the field of medical technical equipment procurement have provided valuable insights and practical guidance throughout this research. His willingness to share expertise and offer support has greatly contributed to the successful completion of this thesis.

# Abstract

This thesis presents a detailed assessment methodology for medical devices that use Bluetooth connectivity, incorporating both technical and privacy considerations. The framework, referred to as the playbook, provides a practical guide for Sykehuspartner to better evaluate and mitigate cybersecurity risks before procuring new medical technical equipment connected to applications with Bluetooth.

The evaluation of privacy and Application Programming Interface (API) security in the procurement process of medical technical equipment is addressed in the research. The study introduces a playbook divided into four sections: network traffic, Bluetooth security, terms/conditions of use, and token security. The playbook consists of questions for each section and incorporates a scoring system. The playbook also provides guidance for answering the questions. Through the use of a Man-in-the-Middle proxy and relevant documentation, suppliers can be effectively compared. The research aims to enhance privacy and security evaluations, ensuring the protection of sensitive data and promoting secure interactions within healthcare information systems.

The playbook should be improved before being used by Sykehuspartner. The playbook is not completely tested and should be improved before it can be an effective asset to Sykehuspartner in the procurement process.

# Contents

<b>Acknowledgements</b>	<b>ii</b>
<b>Abstract</b>	<b>iii</b>
<b>List of Figures</b>	<b>vii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Sykehuspartner . . . . .	1
1.2 Research Question . . . . .	1
1.3 Content decryption . . . . .	2
<b>2 Research method</b>	<b>3</b>
2.1 Background . . . . .	3
2.2 Research methods . . . . .	3
2.3 Action Design research . . . . .	3
2.4 Use of Action Design Research in this project . . . . .	6
<b>3 Literature review</b>	<b>8</b>
3.1 API Security . . . . .	8
3.1.1 API Security Audit Based on Data Flow Tracing . . . . .	8
3.1.2 Research Towards Key Issues of API Security . . . . .	8
3.1.3 Importance of API Security in Healthcare . . . . .	8
3.1.4 Bluetooth security . . . . .	9
3.1.5 GDPR regulations . . . . .	10
3.2 Information Exposure . . . . .	10
3.2.1 OWASP API Security Top 10 . . . . .	11
3.2.2 API3:2019 Excessive Data Exposure . . . . .	11
3.3 Android Architecture . . . . .	12
3.3.1 Android Security Architecture . . . . .	13
3.4 Rooting an Android device . . . . .	16
3.4.1 Rooting process . . . . .	16
3.5 Security in medical technical equipment . . . . .	16
3.6 Man-in-the-middle Proxy . . . . .	17
3.7 Tokens . . . . .	18
<b>4 Procurement Process</b>	<b>19</b>
4.1 Procurement process in Sykehuspartner . . . . .	19
4.2 Commonly used supplier procurement . . . . .	20
4.2.1 CAIQ . . . . .	21

<b>5</b>	<b>Experiment Setup</b>	<b>24</b>
5.1	Playbook . . . . .	24
5.2	Testing . . . . .	24
5.3	Mobile phone setup . . . . .	24
5.4	Man-in-the-middle Proxy . . . . .	25
<b>6</b>	<b>Results</b>	<b>26</b>
6.1	Tests . . . . .	26
6.1.1	Incomplete Testing Due to Application Block . . . . .	26
6.2	Testing the playbook . . . . .	26
6.2.1	Network traffic . . . . .	27
6.2.2	Storage . . . . .	30
6.2.3	Terms/Conditions of Use . . . . .	30
6.2.4	Bluetooth security . . . . .	33
6.2.5	Token Validation in API Security . . . . .	34
<b>7</b>	<b>Discussion</b>	<b>37</b>
7.1	Challenges . . . . .	37
7.1.1	Creating the playbook . . . . .	37
7.1.2	Apps blocking use from rooted phone . . . . .	37
7.1.3	Setting up the mitmProxy . . . . .	37
7.2	Ethics . . . . .	38
7.3	Evaluating each category in questionnaire . . . . .	38
7.3.1	Network traffic . . . . .	38
7.3.2	Bluetooth security . . . . .	38
7.3.3	Terms of Use / Conditions . . . . .	39
7.3.4	Token Security for APIs . . . . .	39
7.4	Evaluating the results . . . . .	39
7.5	Future work . . . . .	40
<b>8</b>	<b>Conclusion</b>	<b>41</b>
	<b>Bibliography</b>	<b>42</b>
<b>A</b>	<b>Playbook</b>	<b>44</b>
A.1	Introduction . . . . .	44
A.2	Setup . . . . .	44
A.3	Questionnaire . . . . .	45
A.3.1	Network traffic . . . . .	45
A.3.2	Terms/Conditions of Use . . . . .	45
A.3.3	Bluetooth security . . . . .	46
A.3.4	Token Validation in API Security . . . . .	46
A.4	How to find answers? . . . . .	47
A.4.1	Network traffic . . . . .	47
A.4.2	Terms/Conditions of use . . . . .	49
A.4.3	Bluetooth Security . . . . .	50
A.4.4	Token Validation in API security . . . . .	51



# List of Figures

2.1	ADR Stages and Principles . . . . .	5
2.2	Process . . . . .	7
3.1	Owasp API:3 Excessive Data Exposure . . . . .	12
3.2	Andorid Architecture . . . . .	13
3.3	Andorid Security Architecture . . . . .	15
3.4	MitmProxy [10] . . . . .	18
4.1	Public procurement . . . . .	20
4.2	Price vs. Quality . . . . .	21
4.3	Example Procurment . . . . .	22
4.4	Explanation Requirements for ICT-services . . . . .	23
6.1	TLS Handshake failed . . . . .	27
6.2	Example Encryption . . . . .	28
6.3	Mitmweb . . . . .	29
6.4	Error running mitmproxy in transparent mode . . . . .	30
6.5	Terms of Use: Third parties . . . . .	32
6.6	Terms of use: Updates . . . . .	33
6.7	Token Example . . . . .	35
6.8	Get Request Example . . . . .	35
7.1	TLS Handshake failed . . . . .	37







# Chapter 1

## Introduction

### 1.1 Sykehuspartner

Sykehuspartner delivers services within the areas of ICT, project, logistics, and HR to all hospitals in the region Helse Sør-Øst. Sykehuspartner operates and manages Information and Communication Technologies (ICT) systems for the hospitals, clinical and administrative applications, ICT infrastructure, and network and work surfaces for 81 000 users.

Sykehuspartner is responsible for the procurement of medical technical equipment for the hospitals in the Helse Sør-Øst region. In the procurement process, one of the aspects that are taken into consideration is the security level of the equipment, among other factors such as price and functionality.

Sykehuspartner wants to be able to verify the security of different products before purchasing them. Previous solutions for evaluating security in the procurement process consist of different questionnaires from Sykehuspartner (Consensus Assessment Initiative Questionnaire (CAIQ) for cloud), where the supplier fills out the functionality and specifics for both the supplying company and the product.

To evaluate privacy and Application Programming Interface (API) security in applications connected to medical technical equipment, a playbook for assessing privacy and API security is made. Using this Playbook, Sykehuspartner can evaluate the privacy and API security themselves before procuring new products. The playbook is sectioned into four parts, with each part containing questions about the following themes: network traffic, Bluetooth security, terms/conditions of use, and token security. The Playbook is made up of questions with a score for different results and a guide on how to answer them. This will give Sykehuspartner a score to compare different suppliers to each other easily.

To answer the questions in the playbook, a combination of a Man-in-the-Middle proxy to capture network traffic from the application and relevant documentation is needed.

### 1.2 Research Question

As healthcare organizations procure new medical technical equipment, ensuring privacy and API (Application Programming Interface) security becomes crucial. This research addresses two key questions:

RQ: Will a playbook assessment approach improve the security evaluation process when procuring new medical technical equipment?

Investigating this question gives insights into evaluating privacy and security in the procurement process, and assesses the effectiveness of the Playbook assessment approach in enhancing privacy and API security. Such research is important in safeguarding patient data and maintaining secure interactions within healthcare information systems.

### 1.3 Content description

Chapter 2 will describe the research method used in this thesis. Following is a literature review, presenting research and theory that has been relevant to form the questions presented in the playbook.

Chapter 4 gives an overview of the procurement process in Sykehuspartner and what Sykehuspartner as a public-owned enterprise must take into consideration when procuring new products.

Chapter 5 describes the setup needed to complete the technical tests, to answer some of the questions in the playbook.

Following in the thesis are the results, discussion, and conclusion.

The playbook can be found in the appendices.

## Chapter 2

# Research method

### 2.1 Background

Sykehuspartner faces limitations when verifying privacy and API security in the procurement process and is dependent on answers from suppliers. They want a method for testing and verifying privacy and API security, without being reliant on the suppliers, and also having a way to easily compare them to each other.

Discussions with Sykehuspartner lead to four themes for the playbook, that Sykehuspartner found interesting to look into when evaluating suppliers. The content of the questions of the playbook is a result of looking into concerning results of previous research on information exposure, frameworks such as OWASP API top 10, and regulatory requirements like GDPR.

### 2.2 Research methods

To create the playbook, qualitative research has been used, with the use of secondary data from already existing information about procurement in the public sector, and from previous research about medical technical equipment.

The challenge is both making a descriptive thesis with the making of the playbook, and an experimental part with testing the playbook in practice. The playbook has been tested to evaluate its effectiveness.

### 2.3 Action Design research

Action Design Research differs from commonly used design research by using inputs from both literature and an organization.

A methodology for performing action design research (ADR) is presented in the article "An extended action design research process model" and may be used by practitioners and researchers to address challenging organizational issues. The model is divided into six stages, including problem identification, diagnosis, solution design, construction, implementation, and assessment. [18]

Each step includes a number of iterative, interactive, and collaborative techniques that bring together researchers and stakeholders. To continuously enhance and refine the generated solutions, it is highlighted how crucial it is to engage in reflective practice throughout the ADR process.

ADR process models offer a framework based on both theory and real-world experience, which helps to close the gap between research and practice. The approach can be used to

provide insights that can be used to improve organizational decision-making and outcomes in a variety of contexts and settings.[11]

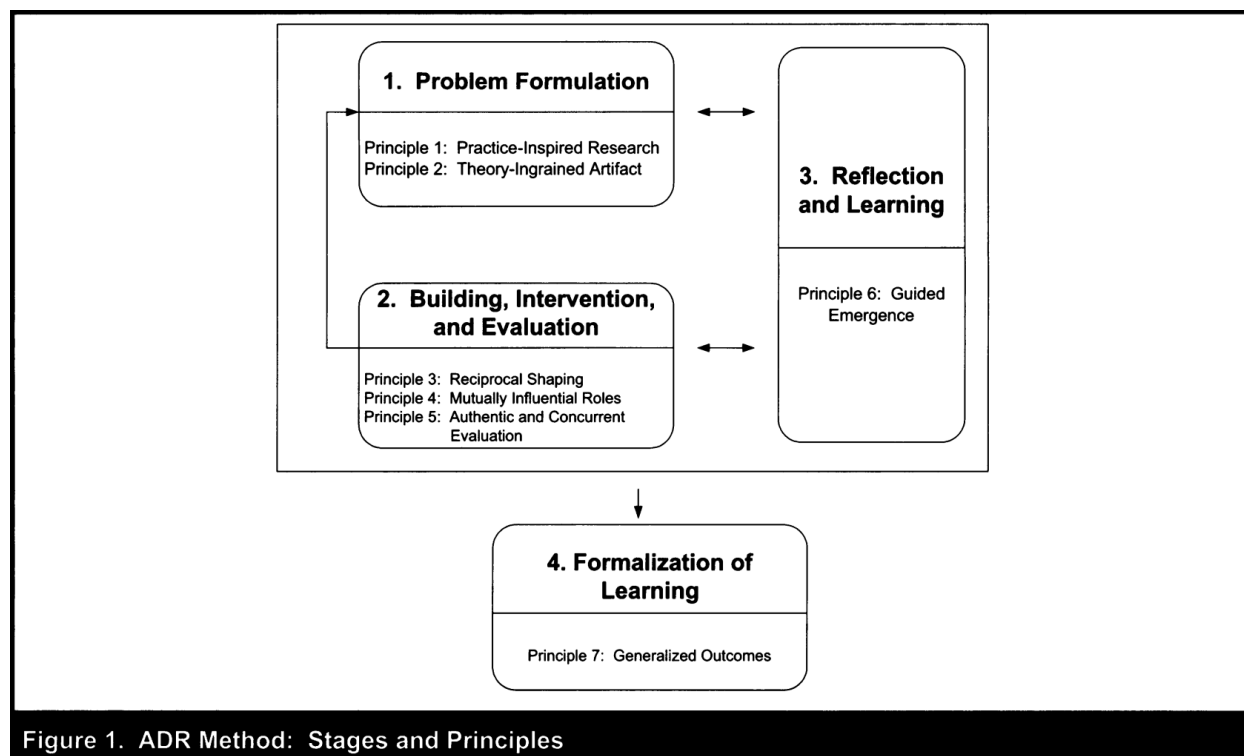


Figure 1. ADR Method: Stages and Principles

Figure 2.1: ADR Stages and Principles

Action design research (ADR) has become a popular method for solving complex problems. ADR is applied to fields such as information systems, management, engineering, and health-care.

ADR was introduced as a response to the limitations of traditional research methods, that would focus on theory and knowledge creation, without a direct connection to practice. In contrast, ADR emphasizes the collaboration between researchers and practitioners to co-create knowledge that is relevant and applicable to real-world situations. ADR builds on several theoretical foundations, including design science, which provides a framework for developing and evaluating artifacts, action research, which emphasizes the iterative process of problem-solving, and user-centered design, which prioritizes the needs and experiences of users.

ADR consists of several components that work together to guide the research process. These include problem identification, artifact design, field testing, and evaluation. In the problem identification phase, researchers work together with practitioners to identify a problem or opportunity for improvement. Next, researchers use design science principles to develop an artifact, which can be a product, service, or process, that addresses the identified problem. The artifact is then tested in the field, typically through a pilot study or a randomized controlled trial. Finally, the artifact is evaluated to determine its effectiveness and potential for further improvement.

ADR has been utilized in the healthcare industry to create interventions that optimize patient outcomes, including lowering pharmaceutical mistakes and enhancing care transitions, and in information systems to create software programs that cater to user needs and preferences.

ADR is a useful tool for dealing with complicated issues in several sectors. The ideas of design science, action research, and user-centered design are combined, and ADR offers a

useful and efficient method for creating creative solutions that are applicable and useful in real-world situations. As a result, by bridging the gap between theory and practice, ADR can potentially promote both research and practice.

## 2.4 Use of Action Design Research in this project

The project's success depends on selecting the right research approach. In order to provide useful and efficient answers to challenging issues, ADR blends the ideas of design science, action research, and user-centered design.

The practicality of ADR is one of its benefits. ADR places a strong emphasis on creating answers to practical issues. As a result, it is a good approach for research topics that are meant to solve real-world problems, such as creating a Playbook for evaluating new suppliers. ADR makes sure the project is applicable and relevant in the actual world.

ADR also has the benefit of being collaborative. ADR ensures close collaboration with experts in the area in order to together produce knowledge that is relevant to and usable in practical contexts. For this project, this was important to pick up important knowledge and expertise from Sykehuspartner, to improve the study.

ADR is an iterative methodology. This means the researcher can refine and improve the project as it progresses. ADR allows for the incorporation of feedback and adjustments to be made based on findings. This ensures that the project is continually evolving and improving, which can lead to better outcomes.

In conclusion, ADR is a suitable methodology for this thesis as it provides a practical approach to research questions, involves collaboration with practitioners, and allows for an iterative process. ADR can be used to develop innovative solutions that are both practical and effective, making it an excellent choice for this thesis. The playbook should be continuously reviewed, to improve the outcome of evaluation. This can affect both the effectiveness and the correctness of the answers.

Figure 2.2 shows how ADR is carried out in this project. For now, only one iteration of the cycle is completed. Based on the result of the test, the playbook must be improved, using both previous research and consulting with Sykehuspartner.



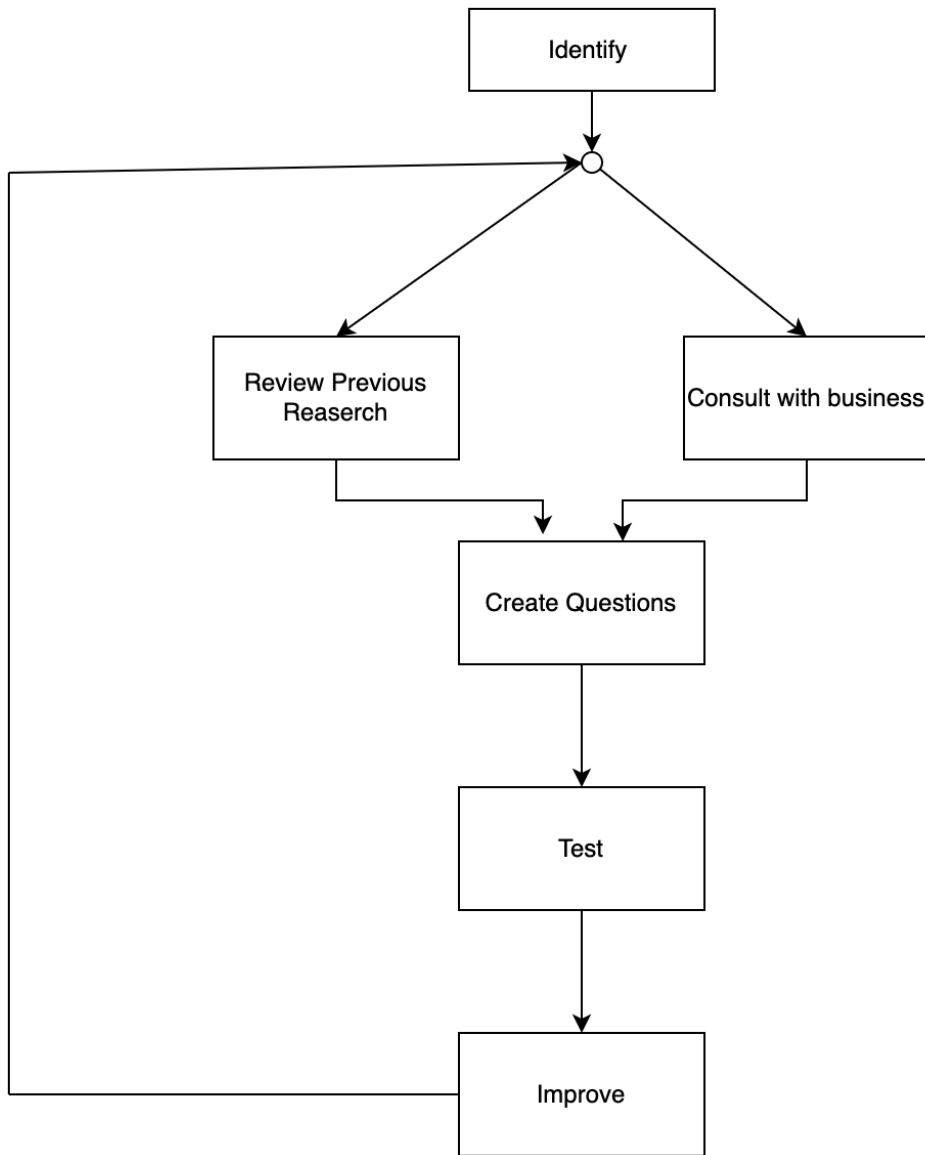


Figure 2.2: Process

# Chapter 3

## Literature review

Previous research made on the fields of API security issues and data flow, Information exposure from different devices, and general data protection regulation (GDPR), combined with information from Sykehuspartner lays the foundation for the Playbook.

### 3.1 API Security

#### 3.1.1 API Security Audit Based on Data Flow Tracing

API is a collection of commands, protocols, functions, and objects that enable interactions between different software systems. It functions as a bridge between developers, modules, and software, and enables common operations to be performed efficiently. APIs have a flexible area of use and are therefore an important part of modern applications like mobile apps, Software as a Service (SaaS), and web applications. They are used in many different industries including banking, retail, autonomous vehicles, and smart homes.[20]

#### 3.1.2 Research Towards Key Issues of API Security

A combination of many old APIs still being run and new APIs going online contributes to the high-security requirements. "There are many security risks in API design, such as various attacks caused by out-of-date API, unauthorized users abusing the API, sensitive API calls, and version confusion." [APIKeyIss] Therefore, it is desirable to conduct a straightforward technical test to verify the security and assess different vendors against each other.

#### 3.1.3 Importance of API Security in Healthcare

The importance of API security in healthcare is significant because of the amount of healthcare data and personal data processed. The increased use of APIs in healthcare contributes to the importance of the focus on API security when dealing with healthcare applications and APIs. With healthcare organizations relying on secure and robust APIs for data exchange, integration, and communication between different systems and applications, this is a critical component. APIs can also be a vulnerable attack surface, where the data could be exposed if not protected properly. This can make them a popular target for cybercriminals. With a robust and secure API, unauthorized access, data breaches, and malicious attacks can be prevented.

[9]

The healthcare sector is also experiencing a rapid digital transformation, with the possibility of remote patient monitoring and other digital health solutions. As a result, the number of APIs in use is growing, increasing the need for comprehensive API security measures to safeguard patient data and secure healthcare systems.

Regulatory compliance is another crucial aspect, as healthcare organizations must adjust to regulations concerning patient information privacy and security, such as GDPR and Pasientjournalloven. Verifying API security can help organizations maintain compliance, avoid potential fines and penalties, and protect their reputation.

Data integrity and patient safety are also critical, as API security plays an important role in preserving patient data integrity and ensuring medical devices and systems' safe operation. Breaches or attacks on APIs can result in incorrect data, leading to misdiagnoses, inappropriate treatment decisions, or compromised patient safety.

Protection against evolving threats is necessary, as cyber threats targeting healthcare APIs continually evolve, with attackers employing sophisticated techniques to bypass security measures. [9] Strong API security implementation can help healthcare organizations stay ahead of these threats and safeguard sensitive data and systems.

Reputation and trust are essential factors, as security breaches involving APIs can lead to a loss of patient trust, reputational damage, and potential business loss. As Sykehuspartner procures new medical technical equipment, investing in API security is crucial for maintaining trust and confidence in healthcare services.

With these factors in mind, this thesis aims to develop a playbook for testing API security in applications connected to medical devices via Bluetooth, focusing on information exposure. By addressing API security, the playbook will contribute to the broader goal of enhancing the security evaluation process for medical devices and improving patient data protection in healthcare.[9]

#### 3.1.4 Bluetooth security

The medical technical equipment the playbook is meant to focus on is equipment connected to an application with Bluetooth. Bluetooth low energy (BLE) is one of the most used types of Bluetooth. BLE was introduced in Bluetooth 4.0, and has several built-in security features, like encryption and data signing.

As Bluetooth has developed, more security has been implemented. In Bluetooth 1.0, there was no significant security introduced. Secure Simple Pairing (SSP) protocol for authentication and key generation during pairing was added in version 2.0. Version 4.0 introduced Bluetooth Low Energy Mode. This version supports the AES-CCM encryption algorithm and Elliptic Curve Diffie-Hellman key exchange for authentication and key generation. Randomized device addresses were also added as a security measurement, to prevent tracking and increase privacy. Bluetooth v5.0 improved the range and the transfer speed. LE secure connections for BLE devices were also implemented, to provide stronger encryption and key exchange.

The security features in each Bluetooth version have evolved to provide stronger encryption, mutual authentication, and privacy protection. It's important for devices to use the latest Bluetooth version and for users to follow best practices such as not sharing sensitive information over unsecured Bluetooth connections. [19]

"A survey on Bluetooth low Energy (BLE) security and privacy" [3] gives an overview of the security and privacy issues related to BLE. BLE is a wireless communication technology and is widely used in Internet of Things (IoT) devices, such as wearable, smart homes, and medical devices.

Bluetooth and BLE are vulnerable to different threats, including denial-of-service attacks, man-in-the-middle attacks, and eavesdropping. Implementing security measures can lower

the likelihood of a successful attack. This includes tools and procedures including key management, encryption, and authentication.

BLE also has privacy repercussions, like tracking user location and activity and gathering private information. The article offers best practices for safeguarding user privacy and lists several privacy laws and regulations that pertain to BLE. BLE presents issues with security and privacy, however, these risks may be reduced by using the right security measures and user education. [3]

### 3.1.5 GDPR regulations

#### **A Novel Privacy Framework for Secure M-Health Applications: The Case of the GDPR**

The article "A Novel Privacy Framework for Secure M-Health Applications: The Case of the GDPR" [12] provides an overview of the privacy needs of mobile health (M-Health) applications in light of the General Data Protection Regulation (GDPR) in Europe. It uses a case study of the "WELCOME" healthcare research project in the EU to analyze the project's privacy features in relation to legal privacy requirements. The challenges faced by developers of m-health applications in ensuring compliance with the GDPR while maintaining user privacy are addressed.

A four-step framework for m-health application developers to ensure GDPR compliance is proposed. This process starts with identifying the personal data processed by the application and then assessing the risks associated with the processing of personal data. Next, the technical and organizational measures necessary to mitigate the identified risks must be defined before the application must be monitored and the effectiveness of the implemented measures can be evaluated.

The importance of privacy in m-health applications is significant. M-health applications can process sensitive personal data, which only enhances the importance of complying with GDPR. A privacy-by-Design approach in the development stage of an m-health application would therefore be recommended. [12]

## 3.2 Information Exposure

The article "Information Exposure From Consumer IoT Devices: A Multidimensional, Network-Informed Measurement Approach" [17] presents a study on information exposure from consumer IoT devices. The authors propose a multidimensional, network-informed measurement approach to quantify the information exposure of IoT devices.

The study involves measuring the information exposure of popular IoT devices such as smart home hubs, security cameras, smart thermostats, and smart TVs. The authors use a network analysis technique to capture the flow of information between IoT devices and the Internet. They also develop a set of metrics to measure the amount of information exposure based on factors such as the type of data transmitted, the frequency of transmission, and the destination of the data.

The study reveals that IoT devices expose a significant amount of information to third-party services, with some devices transmitting sensitive data such as audio and video. The information exposure of the tested IoT devices varies widely depending on the device type and the manufacturer.

The need for improved privacy and security measures for consumer IoT devices is highlighted in the study. The researchers imply that manufacturers should provide users with

more transparency and control over the data collected by their devices. They also propose the development of standardized metrics to measure and compare the information exposure of different IoT devices. [17]

The study presents six research questions, on which the experiment is based. They are as followed:

RQ1: What is the destination of network traffic?

RQ2: To what extent is the traffic encrypted?

RQ3: What data is sent in plaintext?

RQ4: What content is sent using encryption?

RQ5: Does a device expose information unexpectedly?

RQ6: Does the device's location (jurisdiction, location of network egress) impact information exposure?

Some of the tests in the playbook created in this project are based on where the researchers found the most concerning results, combined with discussions with Sykehuspartner on what they find concerning using medical technical equipment connected to an application with Bluetooth.

### 3.2.1 OWASP API Security Top 10

Open Web Application Security Project OWASP provides information about API security, including the common vulnerabilities and best practices for securing APIs. APIs are becoming an essential part of modern web applications and they can be a prime target for attackers.

OWASP lists several common API security vulnerabilities. Some of the highlighted ones are insufficient authentication and authorization mechanisms, insecure data storage, and lack of encryption. It also highlights the importance of protecting APIs against attacks such as SQL injection, Cross-Site Scripting (XSS), and Cross-Site Request Forgery (CSRF).

To mitigate these vulnerabilities, several recommendations for best practices for securing APIs are proposed, including implementing proper authentication and authorization mechanisms, using encryption to protect sensitive data, and validating input and output data. OWASP also recommends implementing rate-limiting mechanisms to prevent Denial of Service (DoS) attacks and monitoring API usage for suspicious activity.

[14]

### 3.2.2 API3:2019 Excessive Data Exposure

One of the top ten vulnerabilities identified by OWASP API Security Top 10 is API3:2019 Excessive Data Exposure. This is a vulnerability when an API exposes more data than what is necessary, putting sensitive information at risk. The sensitive data being exposed could be personally identifiable information (PII), financial data, health data, or other confidential information.

There are several reasons that excessive data exposure could occur. Some of them are:

- misconfigured endpoints
- poor access control
- lack of encryption
- inadequate data validation

- sanitation

Threat agents/Attack vectors	Security Weakness	Impacts
API Specific : Exploitability 3	Prevalence 2 : Detectability 2	Technical 2 : Business Specific
Exploitation of Excessive Data Exposure is simple, and is usually performed by sniffing the traffic to analyze the API responses, looking for sensitive data exposure that should not be returned to the user.	APIs rely on clients to perform the data filtering. Since APIs are used as data sources, sometimes developers try to implement them in a generic way without thinking about the sensitivity of the exposed data. Automatic tools usually can't detect this type of vulnerability because it's hard to differentiate between legitimate data returned from the API, and sensitive data that should not be returned without a deep understanding of the application.	Excessive Data Exposure commonly leads to exposure of sensitive data.

Figure 3.1: Owasp API:3 Excessive Data Exposure [13]

Attackers can exploit this vulnerability to gain unauthorized access to sensitive data and execute malicious commands. To prevent excessive data exposure, proper access control, validation, and sanitation of user input, secure communication protocols, and encrypted data, when it is transmitted or stored, should be implemented.

Data leakage prevention mechanisms should be put in place to monitor data access and detect any unauthorized access. Regular security testing and vulnerability assessments can help identify and mitigate this vulnerability. By addressing this vulnerability, organizations can reduce the risk of data exposure and protect their sensitive information.[13]

### 3.3 Android Architecture

Android is an open platform that has the possibility to add multiple layers of security to a device. The platform's security architecture consists of various layers, each working together to ensure that the system is secure.

Figure 3.2 shows the framework used by the Android operating system, where each layer is sandboxed within a parent layer. The security component of Linux serves as the basis for this sandboxing, ensuring that all layers are isolated and can work together seamlessly. [15]

Android as an operating system is the best choice for performing security research. Since it's open source, source code is available and makes it possible to customize and modify the operating system as needed. Other operating systems, such as iOS, are more closed-sourced, making adjustments to the OS challenging. iOS contains stricter security measures, making getting deeper access, such as root privilege a challenge.



Figure 3.2: Andorid Architecture  
[15]

### 3.3.1 Android Security Architecture

Android's security architecture is built on a combination of hardware and software with different security measures, to protect against various threats. The security architecture has several key components, one being Android OS Security. This is the Android operating system with a variety of security features like application sandboxing, permission-based access control, SELinux mandatory access control, and hardware-backed encryption to secure the data on the device.

Another key security feature on Android is the secure boot. This makes the Android device use a secure boot process to verify the integrity of the operating system and the firmware before it is loaded into memory. This process ensures that only trusted software is booted.

Android also offers hardware security solutions. TrustZone can isolate sensitive functions from the rest of the system and is a hardware-based security solution. Built-in network security solutions are also offered by Android. Android supports VPNs, TLS/SSL encryption,

and secure WiFi protocols, to protect against network attacks.

When rooting a device, several of the security features on android may be bypassed, depending on the approach. The most common of the being Secure Boot, Kernel Security, SELinux, the Android Application Sandbox, system partition and file system protections, and Android Debug Bridge (ADB). Bypassing these features will make it possible to execute unauthorized actions and modify critical system files, among other possibilities.



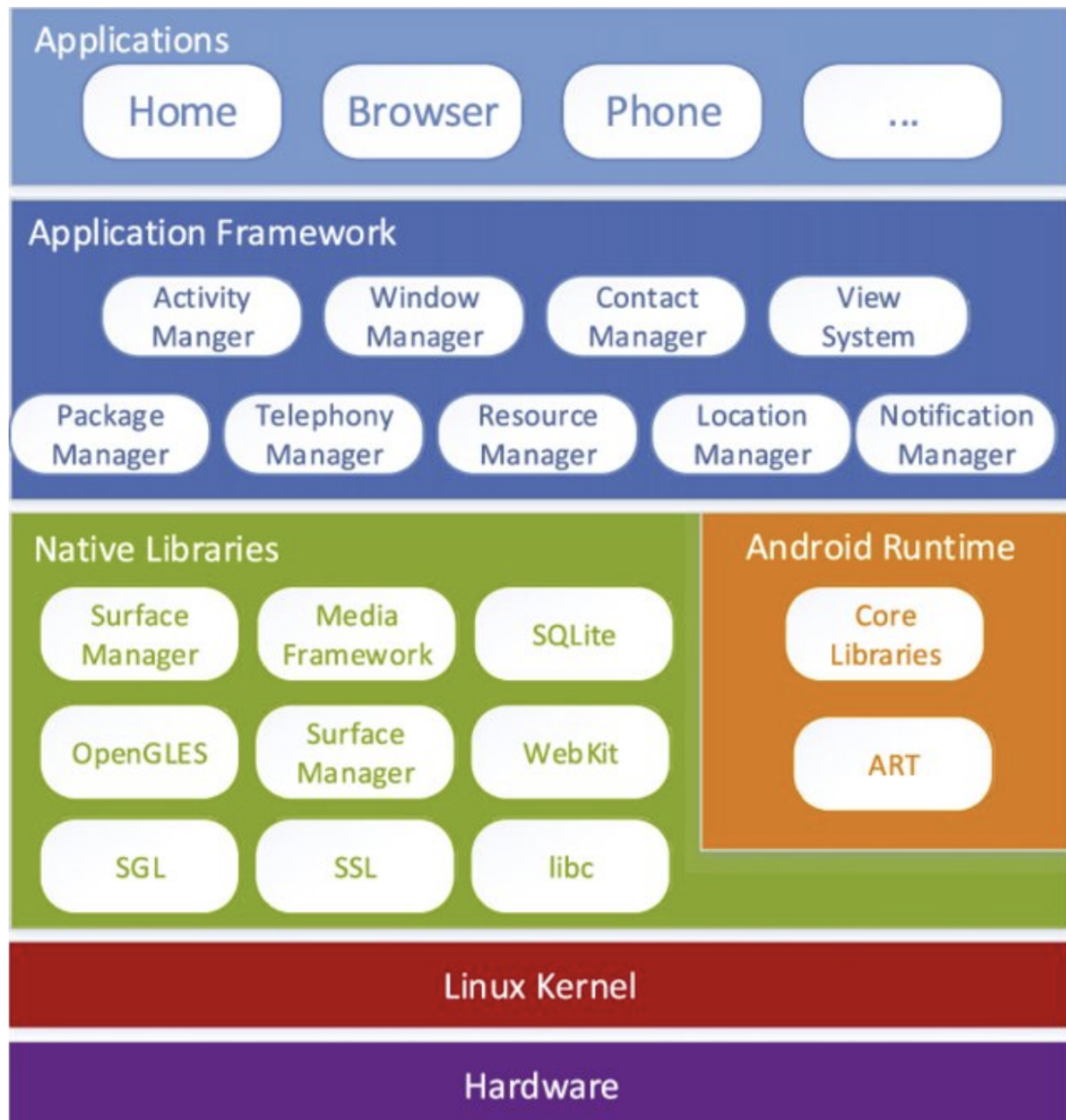


Figure 3.3: Android Security Architecture [4]

## 3.4 Rooting an Android device

To answer the questions in the Playbook, network traffic from a device must be captured. This can be done by rooting the device and setting up a man-in-the-middle proxy (MitM-Proxy).

The security aspect of rooting an Android device is described in the article "Android Security and Its Rooting - A Possible Improvement of Its Security Architecture". Rooting a phone includes gaining privileged access to a device's operating system.

When a user wants to gain administrative privileges to their device, the user can root their device, which is also known as "root access". To root an Android device, a security vulnerability in the operating system must be found. This can be done through various methods, with some common ones being flashing custom firmware, using rooting software, and exploiting system vulnerabilities.

Once root access is gained, the user can install custom Read-Only-Memorys (ROMs), remove pre-installed bloatware, access system files, and run apps that require root permissions. In addition to gaining elevated privileges and access, a rooted device can also expose it to security risks. The device could be more exposed to malware and hacking attempts. And rooting a device could also potentially void the device's warranty. [16]

### 3.4.1 Rooting process

Before starting the rooting process of a device, the data on the device must be backed up, in order to avoid the loss of any important files or information. Once this is completed, USB debugging must be enabled from the device. This allows the device to communicate with a computer during a rooting process. This step will allow the computer to recognize the device and use the tools necessary to complete the rooting process.

Before the device can be rooted, the bootloader must be unlocked. The bootloader is a security feature on Android devices that prevents unauthorized modifications to the software of the device. When the bootloader is unlocked, the software can be modified, but the device will also be more vulnerable to an attack. Next, a custom recovery must be installed. This makes it possible to flash custom firmware and ROMs on the device.

In order to install the necessary files to get root access, a rooting package must be flashed. Magisk is one of several rooting packages that contain the binaries and scripts needed to get root access and manage root permission on the device. The final step to get a rooted device is to reboot the device. This step completes the rooting process and ensures that the device has gained root access. [8]

## 3.5 Security in medical technical equipment

Medical technical equipment is often made in small quantum and with a focus on functionality. A common vulnerability scan of medical technical equipment would therefore not be as accurate as it would otherwise be, because there are few known vulnerabilities.

Research on the field of cybersecurity for healthcare medical devices supports the importance of frameworks, guidelines, and best practices tailored explicitly to the healthcare environment. Conducting comprehensive risk assessments, implementing secure design principles, employing effective access control mechanisms, ensuring robust encryption protocols, and

maintaining vigilant monitoring and maintenance of medical devices is some of the measures mentioned in the report "Cybersecurity for healthcare medical devices".

The research conducted in this field underscores the critical imperative of implementing robust cybersecurity measures within the realm of healthcare medical devices. It provides insights into the challenges, best practices, and emerging technologies, and highlights the importance of collaborative research in order to continuously improve the security in the field. [7]

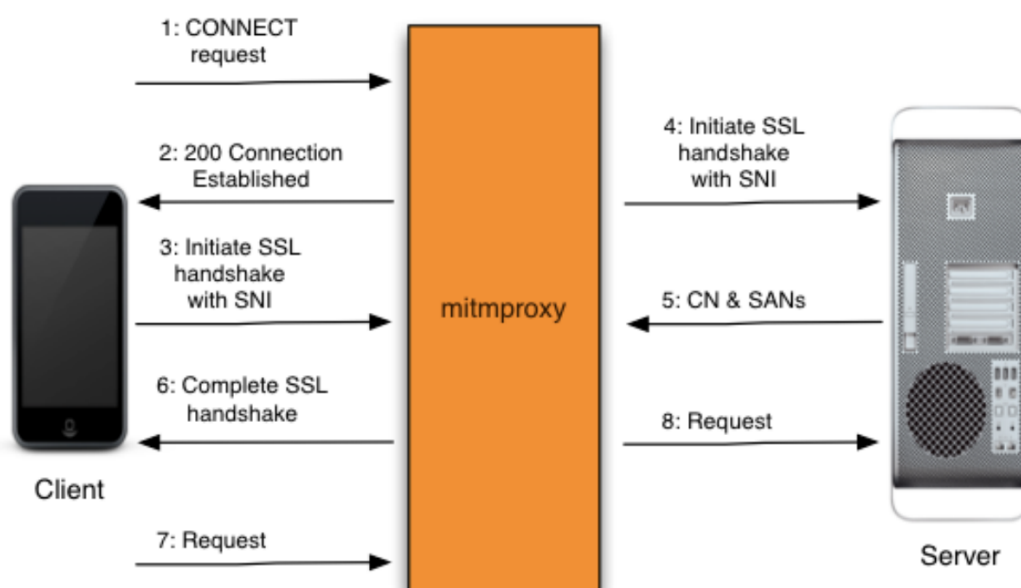
### 3.6 Man-in-the-middle Proxy

To intercept the network traffic between the rooted device and Wireshark, we need to use a man-in-the-middle proxy. Depending on the desired functionality, several MitMProxy solutions might be used. While various tools may provide varied features, they all have the ability to inspect API information such as headers, request and response parameters, cookies, and status codes for the app in use.

A MitMProxy may also be used to validate an app's behavior by enabling some API replies to be alternated. The tool may be used to test the app's timeout behavior or to assess error handling. [6]

A MitMProxy functions as a proxy server between a client and a server. When a client submits a request to the server, the request is intercepted by the MitMProxy, which transmits the request to the server on the client's behalf. Because of this, MitMProxy can record and change network traffic content in real-time.

MitMProxy is developed using Python and can be used on many different platforms, such as Windows, macOS, and Linux. It has a command-line interface and a Python API. MitMProxy is a network analysis and security research tool that allows for real-time network traffic manipulation and inspection, giving insights into network behavior and potential security vulnerabilities. [6]

**mitmproxy docs, Release 1.0.1**


1. The client makes a connection to mitmproxy, and issues an HTTP CONNECT request.
2. Mitmproxy responds with a 200 Connection Established, as if it has set up the CONNECT pipe.
3. The client believes it's talking to the remote server, and initiates the TLS connection. It uses SNI to indicate the hostname it is connecting to.
4. Mitmproxy connects to the server, and establishes an TLS connection using the SNI hostname indicated by the client.
5. The server responds with the matching certificate, which contains the CN and SAN values needed to generate the interception certificate.
6. Mitmproxy generates the interception cert, and continues the client TLS handshake paused in step 3.
7. The client sends the request over the established TLS connection.
8. Mitmproxy passes the request on to the server over the TLS connection initiated in step 4.

Figure 3.4: MitmProxy [10]

Figure 3.4 illustrates and describes how MitmProxy establishes a TLS connection. [10]

### 3.7 Tokens

"A token is a piece of data that has no meaning or use on its own, but combined with the correct tokenization system, becomes a vital player in securing your application. Token-based authentication works by ensuring that each request to a server is accompanied by a signed token which the server verifies for authenticity and only then responds to the request." [2]

## Chapter 4

# Procurement Process

### 4.1 Procurement process in Sykehuspartner

As a public-owned enterprise, Sykehuspartner has to follow certain rules when procuring new equipment. In figure 4.1, the five important aspects that need to be taken into consideration when procuring an new product on behalf of a public establishment are listed.

- Competition
- Equal treatment
- Verifiability
- Proportionality
- Predictability



Figure 4.1: Public procurement

In order to decide on a vendor, Sykehuspartner takes the price and quality of the product into consideration. How much each of these two factors is weighted differs on the product being procured. If the price is weighted more than quality, there is likely that the requirements are commodities so deviation between consortia solutions is likely to be low, or the budget could be highly constrained. A price-focused solution would likely lead to the consortia driving a low-cost solution, and will also dissuade high-cost, quality-driven outsourcers.

A quality-focused solution would focus on getting the best possible solution, regardless of the price. The budget would normally be unconstrained and the solution quality and design are more likely to differ among consortia, compared to a price-focused model. When focusing more on quality instead of price, more innovative solutions are likely to be proposed, which ensures maximum quality marks are achieved. However, focusing only on quality will dissuade price-driven commodity outsourcers from competing.

The key question for Sykehuspartner and Helse Sør-Øst will be how important price is, compared to quality. Figure 4.2 illustrates the price vs. quality aspects.

In figure 4.3, there is an example of how a product can be weighted in the procurement process. In this example, the quality is weighted 70%, and price 30%. The quality section is divided into four sections again, where the functionality is weighted 45%, the technical part 20%, maintenance and support 30%, and project understanding and introduction plans 5%. The technical section covers information security and privacy.

## 4.2 Commonly used supplier procurement

Sykehuspartner has several processes to evaluate different suppliers when procuring new medical technical equipment. One of these documents is called "Specification of requirements: ICT services and information security for Medical Devices". The objective of this document is:

"This document will be used for the evaluation/assessment of the Vendor's offered solution

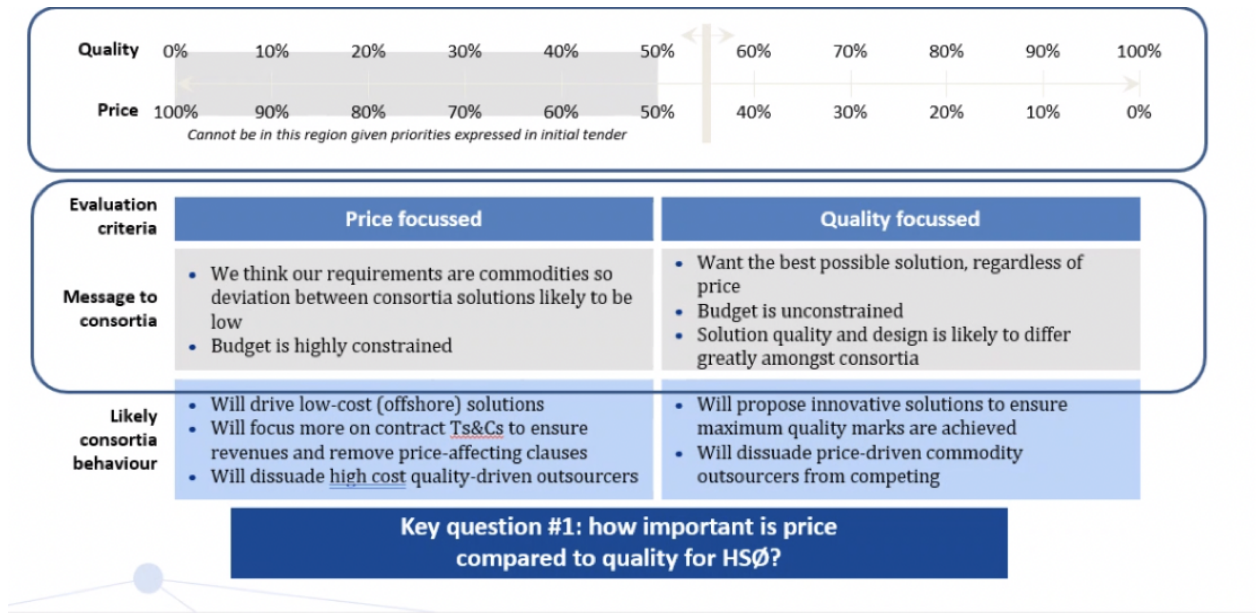


Figure 4.2: Price vs. Quality

in the Information and communications technology (ICT) and information security areas. Additionally, it shall to the greatest possible extent map the solution's basic functionality and suitability of the Client's ICT infrastructure prior to a final customer design. This minimizes the risk of unintended implementation costs, increased implementation time or that desired and offered functionality must be reduced in order to meet the Client's mandatory requirements to information security and privacy. This document also serves to help the Client comply with the statutory requirements of the GDPR. "

#### 4.2.1 CAIQ

CAIQ is an abbreviation that stands for Consensus Assessments Initiative Questionnaire. It is a questionnaire intended to assist enterprises in evaluating the security capabilities of cloud service providers (CSPs). The Cloud Security Alliance (CSA), a nonprofit organization that advocates best practices for safe cloud computing, created the CAIQ.

The questionnaire addresses a wide variety of security and privacy issues, such as data governance, data center security, compliance, and incident management. Organizations may acquire a better knowledge of a CSP's security controls and capabilities by completing the CAIQ and evaluating its ability to fulfill its unique security requirements.

The CAIQ is meant to be used together with the CSA's Cloud Controls Matrix (CCM), which has a more detailed set of security controls and requirements for cloud service providers. Together, the CAIQ and CCM provide a detailed framework for evaluating the security and compliance capabilities of CSPs. Sykehuspartners uses this type of assessment for cloud services.

[1]

## Eksempel – på hvordan det kan se ut

Tildelingskriterie	Vekting nivå 1	Vekting nivå 2	Områder	Vekting nivå 3
<b>Kvalitet</b>	<b>70 %</b>			
- funksjonelt		45 %	Oversikt akutsenter Oversikt sengeposter Kapasitet og ressursplanlegging etc.	X % X % X %
- teknisk		20 %	Informasjonssikkerhet og personvern etc.	X % X % X %
- vedlikehold og support		30 %	Support Feilretting etc.	X % X % X %
- oppdragsforståelse og innføringsplaner		5 %		
<b>Samlede priser</b>	<b>30 %</b>		Site-lisens Årlig vedlikehold Timepriser	x % x % x %
<b>Totalt</b>	<b>100 %</b>			

Figure 4.3: Example Procurment

Sykehuspartners wants to be able to verify the API security and privacy, not needing to rely on answers from the supplier. A Playbook assessment approach to look into API security was discussed, and could possibly fill a gap in the procurement process.

Figure 4.4 explains the different requirements Sykehuspartner has for ICT services and Information security for medical devices. Together with Sykehuspartner, it was discussed, that the questions in the Playbook created, should be a B requirement. This means this requirement should be fulfilled, but does not exclude a vendor if not fulfilled.



### **Explanation of form for Specification of requirements for ICT-services and Information security for medical devices**

<b>Requirements: (A/B/C/D)</b>		
<b>A</b>	Mandatory	Mandatory requirement that must be met. Inability to meet the requirement will entail that the offered solution will be rejected.
<b>B</b>	“Should” requirement	The Vendor’s fulfilment of the requirement is either given a suitability assessment at evaluation or a score in the event of an actual tender evaluation.
<b>C</b>	Documentation	May be combined with A/B/D designation of requirement type. Emphasizes thus that the Client expects a more comprehensive answer (>100 words) that is elaborated/documented in appendices.  If used alone, C is merely an information item that does not require a response or evaluation.
<b>D</b>	High	Combined with B to indicate that the requirement is very important, but not mandatory. The Vendor’s ability to meet the requirement is awarded a score with an associated <b>high weighting</b> upon assessment of the offer

Figure 4.4: Explanation Requirements for ICT-services

## Chapter 5

# Experiment Setup

### 5.1 Playbook

This section explains how the experiment is set up in order to answer some of the questions given in the playbook.

The content in the playbook is based on Sykehuspartners' requirements and previous literature and research. Research shows excessive information exposure from applications, and is also one of the OWASP API top 10 vulnerabilities. Compliance with GDPR and privacy-related issues is also highly relevant issues when talking about healthcare applications and medical technical equipment. Gaps in Sykehuspartner's evaluation of different vendors were discovered, and Sykehuspartner had no way of performing a technical test for API security and information exposure.

### 5.2 Testing

In order to look into the network traffic, a Google Pixel 6a provided by Sykehuspartner is being used. The phone needs to be rooted in order to access the necessary information, we need in the playbook, to evaluate different vendors. A look into Bluetooth security were made to create questions for testing on the area. The terms of use/conditions section were based on Sykehuspartner's already existing guidelines, GDPR requirements, and privacy regulations. The token section of the playbook was based on recommendations and best practices.

### 5.3 Mobile phone setup

To analyze the network traffic from a specific app, we need to root the phone to elevate our access.

This is done following these steps: <https://www.xda-developers.com/how-to-unlock-bootloader-root-magisk-google-pixel-6a/>

This is done by first unlocking the bootloader. This step allows for custom modifications on the device. Unlocking the bootloader will wipe all the data on the phone. Since the phone being used in this project was new, there was no need for backing up data. Then OEM Unlocking and USB Debugging are enabled from options, Developer Options in the device's settings.

The Android SDK Platform-tools package must then be downloaded and installed on the computer. This package contains the necessary tools, such as ADB and Fastboot, which are required for unlocking the bootloader and flashing custom files.

The next step was to unlock the bootloader using Fastboot. The phone was connected to the computer via a USB cable. A command prompt or terminal window on the computer was opened and navigated to the folder containing the platform-tools. The phone was then rebooted into Fastboot mode and use the Fastboot command to unlock the bootloader. After the bootloader was successfully unlocked using the bootloader, the device was rebooted.

The correct recovery image for your device then had to be downloaded to the device. The device was then rebooted into Fastboot mode again and the recovery image flashed using the Fastboot command. After successfully flashing the recovery image, the device was rebooted into recovery mode.

The latest Magisk ZIP file was downloaded from the official Github repository, and the file was transferred to the device's internal storage. The device was booted into recovery mode and the "Install" option was used to flash the Magisk Zip file. The device was rebooted, once the installation was complete.

## 5.4 Man-in-the-middle Proxy

To intercept the traffic, a Man-in-the-Middle Proxy, was set up. The one used for this project was <https://docs.mitmproxy.org/stable/>. This proxy was set up to listen to the current IP address the device and computer were connected to by using the command "mitmdump -w output\_file.pcap". This writes the captured traffic to the file "output\_file.pcap", which then can be opened in Wireshark.

The device must also be set up to the correct proxy settings. Under settings, internet, and advanced settings, the proxy must be set to manual. Here, the IP address being used and the correct port number (8080 as standard), must be inserted. Now the proxy was set up and ready to capture traffic.

In order to capture HTTPS traffic, a valid certificate was needed. When the proxy was listening and the device was correctly set up, a valid certificate could be downloaded from [mitm.it](https://mitm.it).

# Chapter 6

## Results

### 6.1 Tests

#### 6.1.1 Incomplete Testing Due to Application Block

During the evaluation process of the application using the proposed playbook, it was discovered that certain tests could not be completed, as the application was blocked. This limitation prevented a thorough and complete assessment of the application according to the defined test procedures in the playbook. The following sections outline the findings from the tests that were successfully executed, as well as the challenges faced during the testing process.

### 6.2 Testing the playbook

Since the application with the CGM system Sykehuspartner recommended, where testing equipment was also available, was not able to use on a rooted device, it was not possible to test the playbook with the planned equipment. A backup CGM device that was from another vendor was available, but this device was already taken into use on a patient. After some investigation, it turned out that connecting to equipment already in use was not possible, and out-ruled this possibility.

Connecting to a personal smartwatch was a third option. In this case, the questions about Network Traffic, some about network traffic and tokens could be answered with data from an application not approved by Sykehuspartner, and only to test the playbook on a health-care application. However, the current application did not accept the certificate downloaded from the MITM proxy, and no traffic was sent or received. This is a commonly known problem when using mitmproxy to intercept network traffic. A possible solution is to patch the application using a tool called apk-mitm. The .apk file for the current application has to be downloaded and patched, and this can possibly accept the certificate. In the current situation, the application did not patch properly and could not intercept TLS/SSL traffic.

The test of the Playbook is therefore only halfway finished. Given the limitations of not having access to technical documentation and errors that occurred, the Playbook is tested and suggested solutions to fix the errors are given. Within the given timeframe, it was not possible to come up with a solution to all the errors that occurred.

```

[18:13:02.362][192.168.1.149:33448] client disconnect
[18:13:02.363][192.168.1.149:33448] server disconnect achievement-dre.things.dbankcloud.cn:443 (80.158.41.130:443)
[18:13:02.364][192.168.1.149:33450] client disconnect
[18:13:02.365][192.168.1.149:33450] server disconnect sportdata-dre.things.dbankcloud.cn:443 (80.158.37.10:443)
[18:13:02.403][192.168.1.149:33452] server connect connect-drcn.dbankcloud.cn:443 ([2407:c080:1400:2e:5cb:6021:98fc:d437]:443)
[18:13:02.695][192.168.1.149:33452] Client TLS handshake failed. The client does not trust the proxy's certificate for connect-rcn.dbankcloud.cn (OpenSSL Error([('SSL routines', 'ssl3_read_bytes', 'ssl3 alert certificate unknown'])))
[18:13:02.696][192.168.1.149:33452] client disconnect
[18:13:02.697][192.168.1.149:33452] server disconnect connect-drcn.dbankcloud.cn:443 ([2407:c080:1400:2e:5cb:6021:98fc:d437]:443)
)
[18:13:03.132][192.168.1.149:33456] client connect
[18:13:03.188][192.168.1.149:33458] client connect
[18:13:03.191][192.168.1.149:33456] server connect healthdeviceagent-dre.things.dbankcloud.cn:443 (80.158.37.10:443)
[18:13:03.206][192.168.1.149:33460] client connect
[18:13:03.215][192.168.1.149:33462] client connect
[18:13:03.235][192.168.1.149:33464] client connect
[18:13:03.246][192.168.1.149:33458] server connect healthtrade-dre.things.dbankcloud.cn:443 (80.158.33.11:443)
[18:13:03.275][192.168.1.149:33460] server connect healthtrade-dre.things.dbankcloud.cn:443 (80.158.33.11:443)
[18:13:03.282][192.168.1.149:33462] server connect healthtrade-dre.things.dbankcloud.cn:443 (80.158.33.11:443)
[18:13:03.346][192.168.1.149:33456] Client TLS handshake failed. The client does not trust the proxy's certificate for healthdeviceagent-dre.things.dbankcloud.cn (OpenSSL Error([('SSL routines', 'ssl3_read_bytes', 'ssl3 alert certificate unknown'])))
[18:13:03.346][192.168.1.149:33456] client disconnect
[18:13:03.347][192.168.1.149:33456] server disconnect healthdeviceagent-dre.things.dbankcloud.cn:443 (80.158.37.10:443)
[18:13:03.374][192.168.1.149:33466] client connect
[18:13:03.398][192.168.1.149:33458] Client TLS handshake failed. The client does not trust the proxy's certificate for healthtrade-dre.things.dbankcloud.cn (OpenSSL Error([('SSL routines', 'ssl3_read_bytes', 'ssl3 alert certificate unknown'])))
[18:13:03.398][192.168.1.149:33458] client disconnect
[18:13:03.399][192.168.1.149:33458] server disconnect healthtrade-dre.things.dbankcloud.cn:443 (80.158.33.11:443)
[18:13:03.402][192.168.1.149:33464] server connect healthactivity-dre.things.dbankcloud.cn:443 (80.158.41.130:443)
[18:13:03.425][192.168.1.149:33468] client connect
[18:13:03.435][192.168.1.149:33466] server connect healthdeviceagent-dre.things.dbankcloud.cn:443 (80.158.37.10:443)
[18:13:03.447][192.168.1.149:33462] Client TLS handshake failed. The client does not trust the proxy's certificate for healthtrade-dre.things.dbankcloud.cn (OpenSSL Error([('SSL routines', 'ssl3_read_bytes', 'ssl3 alert certificate unknown'])))
[18:13:03.448][192.168.1.149:33460] Client TLS handshake failed. The client does not trust the proxy's certificate for healthtrade-dre.things.dbankcloud.cn (OpenSSL Error([('SSL routines', 'ssl3_read_bytes', 'ssl3 alert certificate unknown'])))
[18:13:03.449][192.168.1.149:33462] client disconnect
[18:13:03.449][192.168.1.149:33460] client disconnect

```

Figure 6.1: TLS Handshake failed

In figure 6.6, an extract from the terminal when the certificate is not accepted is displayed. For the technical testing, a randomly selected weight-tracking application from Google play store, that accepted the mitmproxy's certificate was selected.

### 6.2.1 Network traffic

This section focuses on questions related to the traffic flow from the application:

1. **Encryption:** Is sensitive personal information encrypted? (If yes: 3, no: 0)

No.	Time	Source	Destination	Protocol	Length	Info
45	0.423049	192.168.1.149	192.168.1.109	TLSv1...	967	Application Data, Application Data
47	0.423486	192.168.1.149	192.168.1.109	TLSv1...	430	Application Data
57	0.457902	192.168.1.149	192.168.1.109	TLSv1...	1076	Application Data
60	0.577264	192.168.1.109	192.168.1.149	TLSv1...	826	Application Data
62	0.833010	192.168.1.109	192.168.1.149	TLSv1...	636	Application Data
63	0.835069	192.168.1.109	192.168.1.149	TLSv1...	569	Application Data
64	0.870062	192.168.1.109	192.168.1.149	TLSv1...	728	Application Data
67	0.895581	192.168.1.109	192.168.1.149	TLSv1...	1401	Application Data, Application Data
69	0.895684	192.168.1.109	192.168.1.149	TLSv1...	111	Application Data

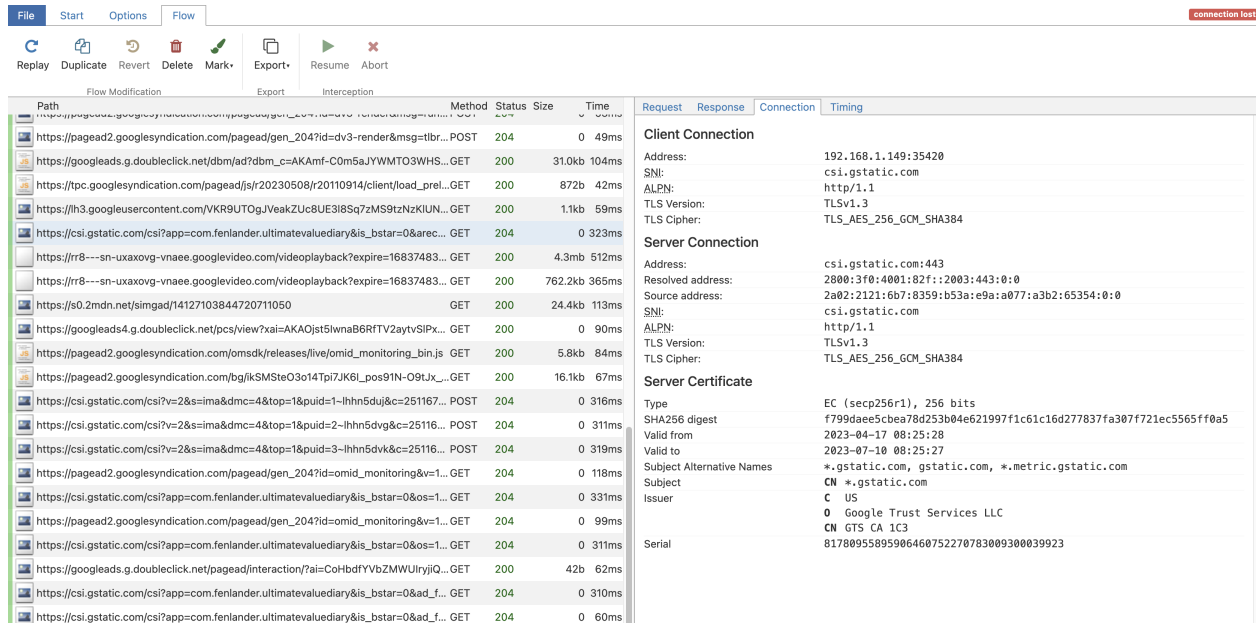
```

> Frame 47: 430 bytes on wire (3440 bits), 430 bytes captured (3440 bits)
> Ethernet II, Src: Google_a8:f2:1c (c8:2a:dd:a8:f2:1c), Dst: Apple_55:3b:c8 (8c:85:90:55:3b:c8)
> Internet Protocol Version 4, Src: 192.168.1.149, Dst: 192.168.1.109
> Transmission Control Protocol, Src Port: 33886, Dst Port: 8080, Seq: 798, Ack: 1437, Len: 364
> Hypertext Transfer Protocol
> Transport Layer Security
  < TLSv1.2 Record Layer: Application Data Protocol: Hypertext Transfer Protocol
    Content Type: Application Data (23)
    Version: TLS 1.2 (0x0303)
    Length: 359
    Encrypted Application Data: 000000000000001f883a4fb849d17b2c9a9b0730746ddc0a700832ab2f2757808662590...
    [Application Data Protocol: Hypertext Transfer Protocol]

```

Figure 6.2: Example Encryption

- Encryption:** Is non-personal information encrypted? (If yes: 1, no: 0)  
Since there was no traffic sent from the captured traffic that was not encrypted, no non-personal data was sent without encryption.
- Locations:** How many locations receive the collected data? (Manual evaluation)
- Location:** Where is the collected data sent? (If EU/EØS: 2, USA (o.l.): 1, "Liste over høysikkerhetsland": 0, other: manual evaluation)



Path	Method	Status	Size	Time
https://pagead2.googlesyndication.com/pagead/gen_204?id=dv3-render&msg=tlbr...	POST	204	0	49ms
https://googleads.g.doubleclick.net/dbm/ad?dbm_c=AKAmf-C0m5ajYWMTO3WHS...	GET	200	31.0kb	104ms
https://tpc.googlesyndication.com/pagead/js/rt20230508/r20110914/client/load_pre...	GET	200	872b	42ms
https://lh3.googleusercontent.com/VKR9UT0gJVeakZUc8UE3i8Sq7zMS9tzNzKIUN...	GET	200	1.1kb	59ms
https://csi.gstatic.com/csi?app=com.fenlander.ultimatevaluediary&is_bstar=0&arec...	GET	204	0	323ms
https://rr8---sn-uxaxovg-vnaee.googlevideo.com/videoplayback?expire=16837483...	GET	200	4.3mb	512ms
https://rr8---sn-uxaxovg-vnaee.googlevideo.com/videoplayback?expire=16837483...	GET	200	762.2kb	365ms
https://s0.2mdn.net/simgad/1412710384420711050	GET	200	24.4kb	113ms
https://googleads4.g.doubleclick.net/pcs/view?xai=AKA0jst5lwnaB6RFTV2aytvSlP...	GET	200	0	90ms
https://pagead2.googlesyndication.com/omsdk/releases/live/omid_monitoring_bin.js	GET	200	5.8kb	84ms
https://pagead2.googlesyndication.com/bg/fkSMSteO3o14Tp7JK6L_pos91N-09tJK...	GET	200	16.1kb	67ms
https://csi.gstatic.com/csi?v=2&s=ima&dmc=4&top=1&puid=1-lhln5dudj&c=251167...	POST	204	0	316ms
https://csi.gstatic.com/csi?v=2&s=ima&dmc=4&top=1&puid=1-lhln5dudj&c=25116...	POST	204	0	311ms
https://csi.gstatic.com/csi?v=2&s=ima&dmc=4&top=1&puid=1-lhln5dudj&c=25116...	POST	204	0	319ms
https://pagead2.googlesyndication.com/pagead/gen_204?id=omid_monitoring&v=1...	GET	204	0	118ms
https://csi.gstatic.com/csi?app=com.fenlander.ultimatevaluediary&is_bstar=0&os=1...	GET	204	0	331ms
https://pagead2.googlesyndication.com/pagead/gen_204?id=omid_monitoring&v=1...	GET	204	0	99ms
https://csi.gstatic.com/csi?app=com.fenlander.ultimatevaluediary&is_bstar=1...	GET	204	0	311ms
https://googleads.g.doubleclick.net/pagead/interaction?ai=CohBdfYVbZMMUlrjQ...	GET	200	42b	62ms
https://csi.gstatic.com/csi?app=com.fenlander.ultimatevaluediary&is_bstar=0&ad_f...	GET	204	0	310ms
https://csi.gstatic.com/csi?app=com.fenlander.ultimatevaluediary&is_bstar=0&ad_f...	GET	204	0	60ms

Request	Response	Connection	Timing
<b>Client Connection</b>			
Address:	192.168.1.149:35420		
SN:	csi.gstatic.com		
ALPN:	http/1.1		
TLS Version:	TLSv1.3		
TLS Cipher:	TLS_AES_256_GCM_SHA384		
<b>Server Connection</b>			
Address:	csi.gstatic.com:443		
Resolved address:	2800:3f0:4001:82f::2003:443:0:0		
Source address:	2a02:2121:6b7:8359:b53a:e9a:a077:a3b2:65354:0:0		
SN:	csi.gstatic.com		
ALPN:	http/1.1		
TLS Version:	TLSv1.3		
TLS Cipher:	TLS_AES_256_GCM_SHA384		
<b>Server Certificate</b>			
Type	EC (secp256r1), 256 bits		
SHA256 digest	f799daee5cbea78d253b04e621997f1c61c16d277837fa307f721ec5565ff0a5		
Valid from	2023-04-17 08:25:28		
Valid to	2023-07-10 08:25:27		
Subject Alternative Names	*,gstatic.com, gstatic.com, *.metric.gstatic.com		
Subject	CN *.gstatic.com		
Issuer	C US O Google Trust Services LLC CN GTS CA 1C3		
Serial	81780955895906460752270783009300039923		

Figure 6.3: Mitmweb

In the man-in-the-middle web interface, the IP addresses, ports, and certificates are easily displayed.

In order to receive the destination IP-address in mitmproxy/wireshark, and not only a private IP-address, the mitmproxy must be running in transparent mode with IP forwarding. Due to an issue with running mitmproxy in transparent mode in this test of the playbook, this step is not confirmed. See figure 6.4

```
marimartini@Maris-MBP ~ % mitmdump --listen-host 192.168.1.109 --mode transparent
[12:51:51.610] transparent proxy listening at 192.168.1.109:8080.
[12:51:54.076] Transparent mode failure: RuntimeError('Could not resolve original destination.')
[12:51:54.102] Transparent mode failure: RuntimeError('Could not resolve original destination.')
[12:51:54.129] Transparent mode failure: RuntimeError('Could not resolve original destination.')
[12:51:55.944] Transparent mode failure: RuntimeError('Could not resolve original destination.')
```

Figure 6.4: Error running mitmproxy in transparent mode

4. **Necessary locations:**How many of the total locations are necessary to process the data? (If All: 2, some: 1, non: 0)

Total possible set points: 10

The network traffic section has some flaws. Mitmproxy and wireshark are only showing private IP addresses, and not the IP-address for the service provider. The location section under Network Traffic can be difficult for Sykehuspartner to answer, as they don't necessarily give a clear picture of what each location does for the data.

The Network Traffic section would also need manual handling for most of the questions, where the logs must be examined for each question.

### 6.2.2 Storage

This section focuses on how what and where collected data is stored:

1. **Location stored:**Where is the data collected stored? (If EU/EØS: 3, USA (o.l.): 2, "Liste over høysikkerhetsland": 0)
2. **Duration of storage:**For how long is the data collected stored? (If Until purpose fulfilled: 1, Undecided: 0)
3. **Anonymization**Is sensitive personal data stored anonymized? (If yes: 2, no: 0)
4. **Anonymization:**Is personal data stored anonymized? (If yes: 1, no: 0)

### 6.2.3 Terms/Conditions of Use

This section focuses on the conditions of use for the application.

When starting up the application, the terms and conditions must usually be accepted in order to take the application and the medical device into use. The information needed to answer the questions below may not be provided in the terms of use. In those cases, Sykehuspartner must either discuss the current question with the vendor or skip the question and subtract the possible score from the total. Even though the vendor might get a good total score, it should be handled manually if the vendor is unable to answer the questions, or if the questions below can not be found in the terms of use agreement.

1. **Data Sharing Consent:** Does the user have to accept that their data is being shared with third parties?

In the application Freestyle Libre 3, which is the top prioritized CGM system from Sykehuspartner/Sykehusinnkjøp, information about sharing with third parties is found under section 13, "Lisens for overførte data". The user of this application has to accept that Abbott, which is the distributor of the CGM system Freestyle Libre 3, receives a royalty-free and non-exclusive license for the use, distribution, reproduction, modification, adaptation, publication, and translation of the data shared with the application. 0 points.



(a) If yes:

i. Is the data anonymized? (If yes: 2, no: 0)

It is stated in the terms of use and condition in section 13, that the data that are shared with external researchers are anonymized, de-identified (or pseudonymized). 2 points

ii. Are third parties clearly identified, and is the purpose of data sharing specified? (If yes: 2, no: 0)

The purpose of the sharing with third parties is clearly stated in the terms of use. Sharing of data is done with the purpose of evaluation of the system and its components, or to evaluate the clinical effect on personnel or between clinics. 2 points

(b) If no: 4

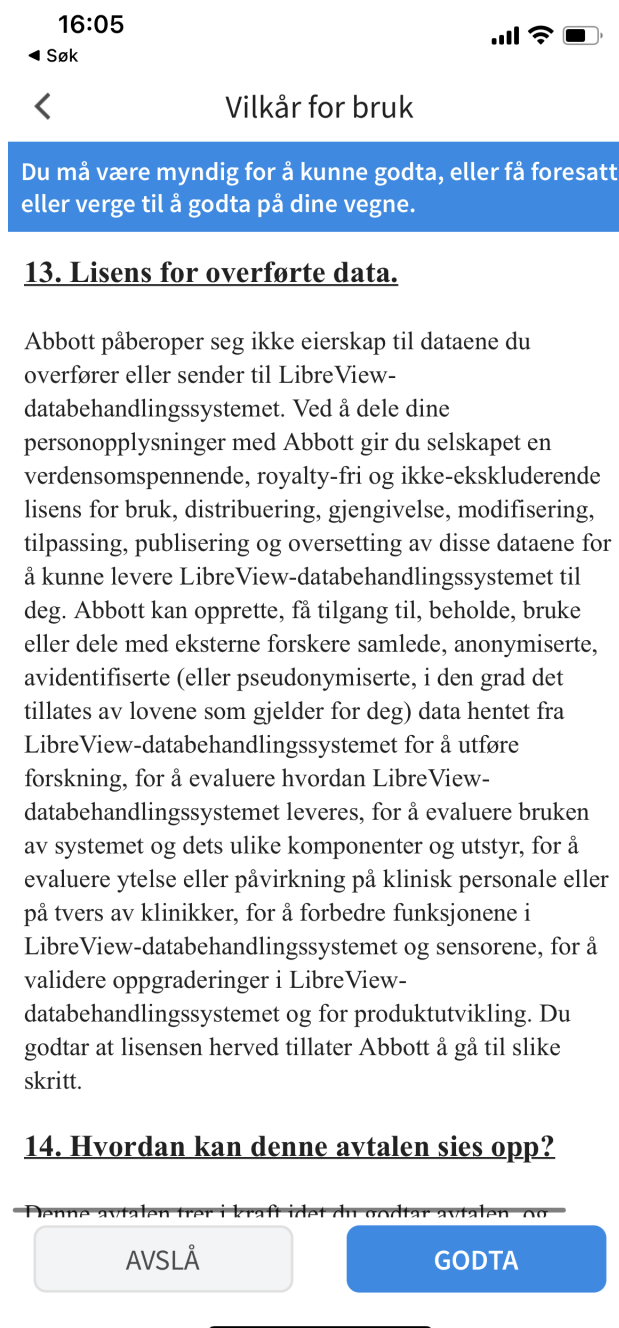


Figure 6.5: Terms of Use: Third parties

2. **Data Retention:** Does the application clearly state the duration for which user data will be stored? (If yes: 2, no: 0)  
It is not stated anywhere in the terms of use the duration which user data will be stored. 0 points
3. **Data Deletion:** Is there a clear procedure for users to request the deletion of their data? (If yes: 2, no: 0)  
A clear procedure on how to delete the user's data was not written in the terms of use. There is, however, a guide on how to terminate the agreement between the user and Abbott. It is not stated if the already collected data will be deleted when the contract is terminated. 1 point
4. **User Rights:** Are users' rights concerning their data, such as the right to access, correct, or object to processing, clearly stated and easy to exercise? (If yes: 2, no: 0)

The right to access and correction of the data is not stated. By not accepting the terms of use, the user can object to the processing. This will lead to the user being unable to use the CGM's systems functionality. 1 point

5. **Usage Restrictions:** Does the application impose reasonable restrictions on its use, such as prohibiting the use of the application for illegal activities or harming others? (If yes: 1, no: 0) Yes, section 11, "Hvilke regler gjelder for bruken av kontoen til LibreView-databehandlingssystemet?" states nine rules when using the Freestyle Libre 3 application. Some of them are, not allowing the user to modify or access the source code and interrupt the LibreView-system. 1 point

6. **Modification of Terms:** Is there a clear procedure for notifying users of changes to the terms and conditions? Are users given the opportunity to review and accept new terms before continuing to use the application? (If yes: 1, no: 0)

It is stated in the terms of use that changes in the agreement will be notified in a recent amount of time. This will include the updated agreement, with a requirement that the user accepts the new conditions for using the application. 1 point.

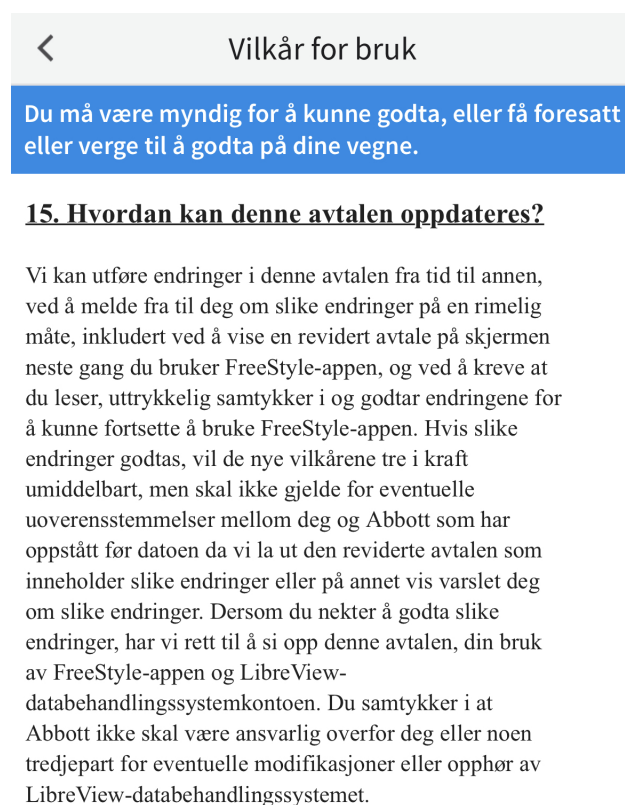


Figure 6.6: Terms of use: Updates

Received points: 8

Total possible points: 12

Score: 66,7%

#### 6.2.4 Bluetooth security

To answer this section, Sykehuspartner has to look up the technical specifications for the device they want to assess. Without access to the device or technical documentation about this product, this part will not be tested in the playbook. A description of how to answer and perform tests on the device is described in the playbook section.

1. **Bluetooth Version:** Which version of Bluetooth is being used? (If v5.0 or v4.0: 1, lower: 0)  
"The FreeStyle Libre 3 sensor utilizes Bluetooth Low Energy (BLE) technology to stream glucose readings and enable real-time glucose alarms\*†." <https://www.freestyle.abbott/us-en/support/faq.html?page=device/freestyle-libre-3-system/faq/topic/sensor>. This information was found on the website for the product. This information may also be available in the user manual and/or additional technical specifications about the product.
2. **Pairing Method:** What method is used for pairing the devices? (If Secure Simple Pairing (SSP) or LE Secure Connections (LESC): 2, legacy pairing: 0)
3. **Encryption:** Is the Bluetooth connection encrypted? (If AES-CCM with a key length of 128 bits or higher: 2, lower encryption strength or no encryption: 0)
4. **Authentication:** Is there a strong authentication mechanism in place to prevent unauthorized devices from connecting? (If yes: 2, no: 0)
5. **Signal Range:** Does the application limit the Bluetooth signal range to minimize the risk of unauthorized connections or eavesdropping? (If yes: 1, no: 0)
6. **Bluetooth Low Energy (BLE) Privacy:** If using BLE, is the privacy feature enabled to protect against tracking by randomizing device addresses? (If yes: 1, no: 0)
7. **Firmware Updates:** Does the device support secure firmware updates to fix potential Bluetooth security vulnerabilities? (If yes: 1, no: 0)

Total possible points: 8.5

Without access to technical documentation, this section could not be answered completely.

### 6.2.5 Token Validation in API Security

1. **Token Type:** What type of token is used? (If OAuth 2.0 access tokens or JWT: 3, other standard tokens: 2, custom tokens: 1, no token: 0)  
Look for token information in the request field in mitmproxy or use Wireshark. See example: Use of custom token in request, which will give 1 point. See figure 6.7.

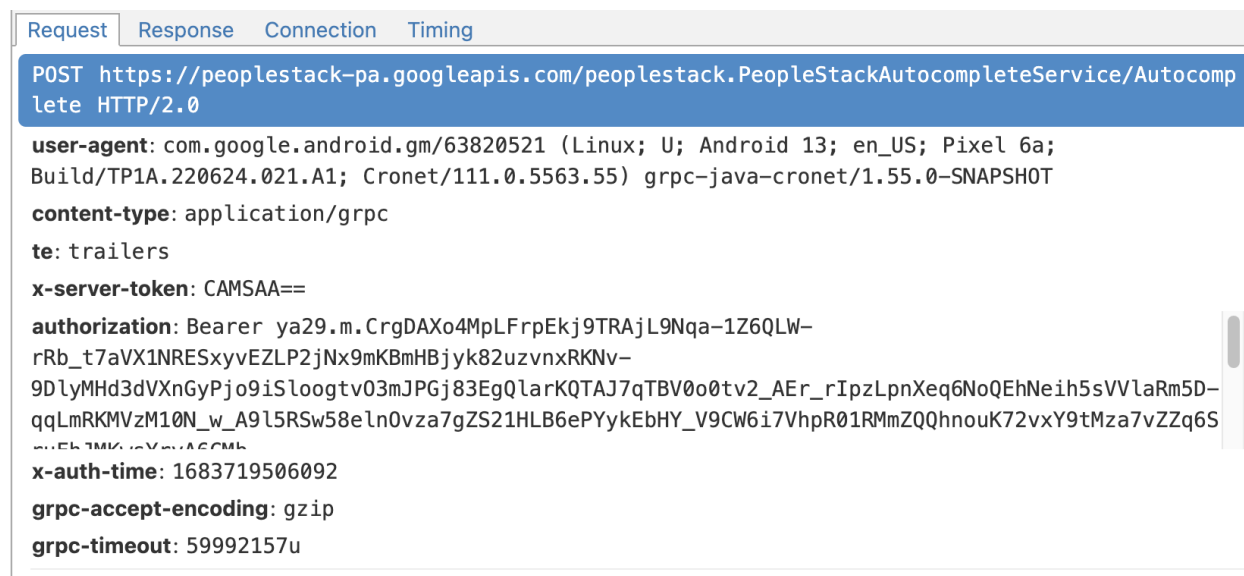


Figure 6.7: Token Example

```

92.168.1.148:38004: GET https://119.29.29.99/d?dn=192.168.1.109&token=934149168&ttl=1
<< 200 OK 5b
10:23:21.253][192.168.1.148:38016] client connect

```

Figure 6.8: Get Request Example

- Token Expiration:** Are tokens set with an appropriate expiration time? (If yes: 2, no: 0)  
See fig 6.8. token=934149168: The query parameter token is present in the request, and its value is set to 934149168. This suggests that the token being used for authentication or authorization is 934149168. ttl=1: The query parameter ttl is included, and its value is set to 1. "TTL" often stands for "Time To Live" and indicates the duration or lifetime of the token. In this case, a value of 1 suggests a short-lived or temporary token with a TTL of 1 unit (the specific unit of time may depend on the system or application).

The rest of the "Token Validation in API Security" might require detailed documentation about the API in order to give a score. This can be provided from the application owner.

- Token Revocation:** Is there a mechanism for revoking tokens when necessary? (If yes: 2, no: 0)
- Token Storage:** Is the token securely stored on the client-side? (If stored in HttpOnly cookies or secure client-side storage: 2, insecure storage: 0)
- Token Transmission:** Are tokens transmitted securely, e.g., over HTTPS? (If yes: 2, no: 0)
- Token Validation:** Is the token validation process robust and follows best practices, such as checking for token signature, issuer, audience, and other claims? (If yes: 2, no: 0)
- Scope Management:** Are token scopes managed correctly, allowing access only to necessary resources and actions? (If yes: 2, no: 0)
- Error Handling:** Are token validation errors appropriately handled, with clear and informative error messages returned to the client without revealing sensitive information? (If yes: 1, no: 0)

Total possible points: 14

# Chapter 7

## Discussion

The playbook was made using action design research, combining previous research on relevant subjects, and discussions with Sykehuspartner. The meetings with Sykehuspartner have mainly been with Sindre Gjelsten who works there as a security advisor. Trond Ødegård from Sykehusinnkjøp also contributed with important knowledge about the procurement process in Sykehuspartner and Sykehusinnkjøp.

### 7.1 Challenges

#### 7.1.1 Creating the playbook

When creating the playbook, several considerations had to be made. First, Sykehuspartner's wishes and requirements for the playbook must be met. The test had to be easy to execute for an employee with limited knowledge of cyber security. Together with Sykehuspartner, the different parameters that should be included, were discussed.

#### 7.1.2 Apps blocking use from rooted phone

One of the challenges was that the main priority CGM system Sykehusinnkjøp recommends was blocks for use through a rooted phone. So was the second and third recommended system as well. Because of the dependency on having the physical sensor available to complete the testing of the playbook, the Dexcom CGM system was used for the test. This CGM system is not one of the recommended systems from Sykehusinnkjøp, but it was the only one available to complete the testing.

#### 7.1.3 Setting up the mitmProxy

When setting up the Man in the Middle Proxy, the TLS handshake failed. See details in figure 7.1.

The error message "Client TLS handshake failed. The client does not trust the proxy's certificate for api.ipify.org (OpenSSL Error([('SSL routines', 'ssl3\_read\_bytes', 'ssl3 alert certificate unknown'])))

```
marimartin@Moris-MBP desktop % mitmdump --mode socks5 --listen-host 172.20.10.4 --w outfile --s tls_passthrough.py
[12:20:50.150] Loading script tls_passthrough.py
[12:20:50.150] SOCKS v5 proxy listening at 172.20.10.4:1080.
[12:29:16.680] client connect
[12:29:16.810] [172.20.10.11:6896] server connect api.ipify.org:443 (104.237.62.211:443)
[12:29:16.940] [172.20.10.11:6896] client TLS handshake failed. The client does not trust the proxy's certificate for api.ipify.org (OpenSSL Error([('SSL routines', 'ssl3_read_bytes', 'ssl3 alert certificate unknown'])))
[12:29:16.940] TLS handshake failed: 104.237.62.211:443
[12:29:16.940] client disconnect
[12:29:16.940] [172.20.10.11:6896] server disconnect api.ipify.org:443 (104.237.62.211:443)
[12:29:16.950] [172.20.10.11:6896] client connect
[12:29:17.340] [172.20.10.11:6896] server connect ip.seep.org:443 ([2602:fed3:2:b74f:112:9a23:af4f:2219]:443)
[12:29:17.510] [172.20.10.11:6896] server TLS handshake failed: connection closed
[12:29:17.510] [172.20.10.11:6896] Unable to establish TLS connection with server (connection closed). Trying to establish TLS with client anyway. If you plan to redirect requests away from this server, consider setting 'connection_strat
gy' to 'lazy' to suppress early connections.
[12:29:17.520] [172.20.10.11:6896] server disconnect ip.seep.org:443 ([2602:fed3:2:b74f:112:9a23:af4f:2219]:443)
[12:29:17.530] [172.20.10.11:6896] client TLS handshake failed. The client does not trust the proxy's certificate for ip.seep.org (OpenSSL Error([('SSL routines', 'ssl3_read_bytes', 'ssl3 alert certificate unknown'])))
[12:29:17.540] [172.20.10.11:6896] client disconnect
[12:29:17.540] [172.20.10.11:6896] client connect
[12:29:17.620] [172.20.10.11:6896] server connect www.trackip.net:443 ([2606:4700:3030:6815:4f85]:443)
[12:29:17.780] [172.20.10.11:6896] client TLS handshake failed. The client does not trust the proxy's certificate for www.trackip.net (OpenSSL Error([('SSL routines', 'ssl3_read_bytes', 'ssl3 alert certificate unknown'])))
[12:29:17.780] TLS handshake failed: [2606:4700:3030:6815:4f85]:443
[12:29:17.790] [172.20.10.11:6896] client disconnect
[12:29:17.790] [172.20.10.11:6896] server disconnect www.trackip.net:443 ([2606:4700:3030:6815:4f85]:443)
```

Figure 7.1: TLS Handshake failed

certificate unknown' ]))" indicated that the proxy's certificate was not trusted. The traffic was still not running through the proxy, so acquiring a valid certificate from mitm.it was not an option as the site shows the message "If you can see this, traffic is not passing through mitmproxy."

## 7.2 Ethics

Since the app to the sensor, Sykehuspartner was able to provide, was not supported on rooted devices, the second CGM system was not able to connect to the device, and no personal information was gathered in this work.

## 7.3 Evaluating each category in questionnaire

### 7.3.1 Network traffic

To evaluate network traffic, a decision to divide encryption into three parts was made. One is sensitive personal information, such as healthcare data. One being non-sensitive personal information and one non-personal information. The sensitive personal information is the most critical if sent unencrypted, and therefore gives the most points if achieved. The second most important is non-sensitive, but personal information. The third type of information one could expect to be sent is non-personal information. This information should also be encrypted, but is not as critical as the two above, and was therefore given a maximum of total points of 1.

Question number 4. in the network traffic category must be dealt with manually as it is hard for Sykehuspartner to decide what is an acceptable number of locations that can receive the data. This is dependent on the complexity of the application and the purpose of the application.

Sykehuspartner has an official list of high-risk countries. The score on where the data is stored is based on this list. The high-risk countries are Russia, China, Iran, North Korea and Pakistan. This list can be found at <https://sykehuspartner.no/Documents/Sikkerhet%20-%20regionale%20bruksvilk%C3%A5r/NO-41%20-%20Liste%20over%20h%C3%B8yrisikoland.pdf>

The preferred location for Sykehuspartner to store data is in the EU/EØS since the GDPR applies there.

The final question about how many of the total locations are necessary to process the data is not numbered. This again depends on the complexity and purpose of the application. If all the locations are necessary the fulfill the purpose and use of the application, the total score will be 2 points. If some of the locations are not necessary, one point will be given. If non of the locations that receive the data is necessary to complete the app's purpose, no points will be received.

The score given to the location of the data storage is based on the list of high-risk countries. If the collected data is stored longer than needed, the score given will be 0, otherwise 1. The section about storage also differs between sensitive personal information and non-sensitive. It is possible to gain more points storing sensitive personal information than non-sensitive.

### 7.3.2 Bluetooth security

If the device uses Bluetooth as the connection method, the score given is based on the Bluetooth version used. This was not discovered using the test and may need to be discussed



with the producer. As described earlier, v4.0 and v5.0 have improved Bluetooth security by a lot and are therefore given a higher score than the lower versions.

### 7.3.3 Terms of Use / Conditions

The questions chosen for the "Terms of Use / Conditions" in the playbook give a fundamental overview of user privacy and data protection. By evaluating data sharing consent, data retention, data deletion procedures, user rights, usage restrictions, and terms modification notifications, Sykehuspartner ensures that the end-user of the application has control of their data and can make informed decisions. This section in the Playbook ensures that users have control over their data, understand how it is shared and used, and can exercise their rights when it comes to accessing, correcting, or objecting to data processing.

### 7.3.4 Token Security for APIs

Tokens are of significant importance to API Security, and are therefore included in the assessment. Confirming token type, expiration, revocation mechanism, storage, transmission security, validation process, scope management, and error handling is part of verifying a strong token authentication system.

Identifying the token type aids in understanding the underlying security protocols, with OAuth 2.0 access tokens or JWTs signifying standardized practices. Validating token expiration and implementing appropriate revocation mechanisms prevent unauthorized access. Securely storing tokens on the client side and transmitting them over HTTPS enhance data protection. Implementing a robust validation process, including checking token signatures and claims, helps prevent tampering and unauthorized usage. Effective scope management ensures that tokens only provide access to necessary resources and actions. Proper error handling safeguards sensitive information by returning informative error messages without exposing sensitive details.

## 7.4 Evaluating the results

The results given from testing the Playbook can give information about both the effectiveness of the Playbook and possible changes that should be made.

The first section, "Network traffic" has the potential to be improved. As the playbook is at this stage, most of the work answering the questions must be handled manually. Several issues were discovered while testing this section. The main issue was the built-in security features in the applications that were to be tested. One of them is that the application is not accepting the downloaded mitmproxy certificate. This made all testing from the application in the Network section impossible at this stage.

In order to still try to test the application, a few different methods were used. One of them was to download the application .apk file and try to patch this application to accept the mitmproxy, using the apk-mitm feature in the mitmproxy. After this was unsuccessful, using Just Trust Me to disable certificate checking was also tried. [5].

To answer the questions in the Playbook, improvements must be made to the Playbook and the guide on how to answer. The section about network traffic is not efficient for Sykehuspartner to use at this current stage. The Playbook requires manual handling and could be time-consuming to fill out. To set up the needed mitmproxy, Wireshark and proxy on the device with the application, could also require technical skills, which might take up time

from valuable recourses.

The terms of use / Conditions section is efficient and could be a valuable resource to Sykehuspartner when evaluating different vendors and applications. When testing the Playbook, the questions are clear and if the answer exists, it should be easy to find, given that a Table of content is present in the Terms of Use. The section provides valuable information about privacy for the user of the application and sensor system and can reveal the lack of information on privacy.

The Bluetooth section provides an easy guide that could help Sykehuspartner in their evaluation. However, this section has not been properly tested because of the lack of technical specifications about the CGM system. Given that Sykehuspartner is provided with the correct documentation about the product from the service provider, they could have an efficient system for evaluating Bluetooth security for medical technical equipment that is connected to an application with Bluetooth. Sykehuspartner must verify if this statement is correct by performing such a test with access to technical documentation.

The "Token Validation in API Security" section could be valuable for Sykehuspartner when evaluating API security. Some of the questions could be answered using the captured traffic in mitmweb/mitmproxy/wireshark, but answers must be found manually by investigating the traffic. Other answers must be found by reviewing relevant documentation. This could result in time-consuming work.

## 7.5 Future work

The playbook must be improved in order to fulfill Sykehuspartner's requirements. The Playbook contains a lot of manual handling, that should be atomized in order to make an efficient evaluation of API security. As for now, the results vary based on the security of the application. This may make the test impossible to complete. To make the test efficient, it should either be specialized for each individual application based on the security of the application, or the test could be expanded to cover several aspects and cases.

The test could also be expanded, both with new sections about other parts relevant to API security, and expanding the questions within each section to go more into detail.

For future work, it is suggested that:

- Further attempts should be made to gain access to the applications not accepting the mitmproxy's certificate and the applications blocked from being used on rooted devices in order to complete the remaining tests within the playbook.
- Collaboration with the application developers or owners should be considered to facilitate the completion of the assessment process.
- The playbook could be refined and updated based on the findings and limitations encountered during this testing process.

## Chapter 8

# Conclusion

The Playbook is a practical idea on how to evaluate different vendors, based on how Sykehuspartner already evaluates possible new suppliers. The Playbook is created based on a combination of previous research and Sykehuspartners' requirements. Each question is given a score, based on best practices and Sykehuspartners standards.

The results of the testing of the Playbook show that it is partially functional and with some adjustments could be used by Sykehuspartner to assess new possible vendors that supply medical technical equipment that is connected to an application with Bluetooth.

A Playbook to test API Security could be an efficient way to evaluate different vendors, in order to decide which one to choose. The test in this Playbook must be improved in order to be effective and useful for Sykehuspartner in their procurement process. The Playbook is only partially tested, due to unforeseen security measurements applied in the applications, and was therefore not completely tested. Technical specifications for the CGM system and the application were also not available to complete the tests. The section of the test that was completed in full, gives an example of how the other parts may look when completed.

RQ: Will a playbook assessment approach improve the security evaluation process when procuring new medical technical equipment?

Using a playbook assessment approach, privacy and API security can be tested before procuring new medical technical equipment using a playbook assessment approach. This approach may be time-consuming for Sykehuspartner but can give valuable information about API security and privacy, which can help decide on a supplier.

It can also be an effective method to test privacy and API security. To make this approach effective, the playbook must either be broad to cover all possible aspects of the situation or made specifically for each individual situation.

An improved version of the playbook could improve the procurement process, making it easier for Sykehuspartner to make decisions when choosing suppliers.

# Bibliography

- [1] Cloud Security Alliance. *Cloud Controls Matrix (CCM) Version 3.0.1*. <https://cloudsecurityalliance.org/research/cloud-controls-matrix/>. 2016.
- [2] Auth0. *Token-Based Authentication Made Easy*. <https://auth0.com/learn/token-based-authentication-made-easy>. Website. accessed 2023.
- [3] Matthias Cäsar et al. “A survey on Bluetooth Low Energy security and privacy.” In: *Computer Networks* 205 (2022), p. 108712. ISSN: 1389-1286. DOI: <https://doi.org/10.1016/j.comnet.2021.108712>. URL: <https://www.sciencedirect.com/science/article/pii/S1389128621005697>.
- [4] Rui Chang et al. “Towards a multilayered permission-based access control for extending Android security.” In: *Concurrency and Computation: Practice and Experience* 30 (June 2017), e4180. DOI: [10.1002/cpe.4180](https://doi.org/10.1002/cpe.4180).
- [5] Fuzion24. *JustTrustMe*. <https://github.com/Fuzion24/JustTrustMe>. GitHub repository. accessed 2023.
- [6] Chetan Gaikwad. “Monitoring & Modifying Android App Network Traffic via MITM Proxy [Part 1].” In: *Medium* (May 2020). URL: <https://gaikwadchetan93.medium.com/monitoring-modifying-android-app-network-traffic-via-mitm-proxy-part-1-886f6324f705>.
- [7] Brian Greer and A Capstone. “CYBERSECURITY FOR HEALTHCARE MEDICAL DEVICES.” PhD thesis. May 2018.
- [8] SKANDA HAZARIKA. *How to unlock bootloader and root with Magisk on Google Pixel 6a*. Jan. 2022. URL: <https://www.xda-developers.com/how-to-unlock-bootloader-root-magisk-google-pixel-6a/>.
- [9] “Importance of API Security in Healthcare Grows as Cyberattacks Increase.” In: *HealthITSecurity* (). URL: <https://healthitsecurity.com/news/importance-of-api-security-in-healthcare-grows-as-cyberattacks-increase>.
- [10] mitmproxy. *Documentation*. n.d. URL: <https://docs.mitmproxy.org/stable/>.
- [11] Matthew T. Mullarkey and Alan R. Hevner. “An elaborated action design research process model.” In: *European Journal of Information Systems* 28 (2019). Ed. by Pär Ågerfalk. URL: <https://doi.org/10.1080/0960085X.2018.1451811>.
- [12] Uzma Mustafa, Eckhard Pflugel, and Nada Philip. “A Novel Privacy Framework for Secure M-Health Applications: The Case of the GDPR.” In: *2019 IEEE 12th International Conference on Global Security, Safety and Sustainability (ICGS3)*. 2019, pp. 1–9. DOI: [10.1109/ICGS3.2019.8688019](https://doi.org/10.1109/ICGS3.2019.8688019).
- [13] OWASP. *Excessive Data Exposure*. <https://github.com/OWASP/API-Security/blob/master/2019/en/src/0xa3-excessive-data-exposure.md>. [Accessed: April 22, 2023]. 2019.
- [14] OWASP. *OWASP API Security Project*. 2023. URL: <https://owasp.org/www-project-api-security/> (visited on 03/28/2023).
- [15] Nick Rahimi, John Nolen, and Bidyut Gupta. “Android Security and Its Rooting—A Possible Improvement of Its Security Architecture.” In: *Journal of Information Security* 10 (Jan. 2019), pp. 91–102. DOI: [10.4236/jis.2019.102005](https://doi.org/10.4236/jis.2019.102005).

- [16] Nick Rahimi, John Nolen, and Bidyut Gupta. “Android Security and Its Rooting—A Possible Improvement of Its Security Architecture.” In: *Journal of Information Security* 10 (Jan. 2019), pp. 91–102. DOI: [10.4236/jis.2019.102005](https://doi.org/10.4236/jis.2019.102005).
- [17] Jingjing Ren et al. “Information Exposure From Consumer IoT Devices: A Multidimensional, Network-Informed Measurement Approach.” In: (2019). URL: <https://doi.org/10.1145/3355369.3355577>.
- [18] Maung K. Sein et al. “Action Design Research.” In: *MIS Quarterly* 35.1 (2011), pp. 37–56. ISSN: 02767783. URL: <http://www.jstor.org/stable/23043488> (visited on 03/08/2023).
- [19] National Institute of Standards and Technology (NIST). *Guide to Bluetooth Security*. Tech. rep. Special Publication 800-121 Revision 2. 2012. URL: <https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-121r2.pdf>.
- [20] Sun R. and Wang Q. and Guo L. “Research Towards Key Issues of API Security.” In: (2022). URL: [https://doi.org/10.1007/978-981-16-9229-1\\_11](https://doi.org/10.1007/978-981-16-9229-1_11) (visited on 03/08/2023).

# Appendix A

## Playbook

### A.1 Introduction

This playbook is meant as a guide to help Sykehuspartner better evaluate API security in the procurement stage of medical technical equipment connected to an application with Bluetooth. The playbook can also be used later as a part of internal control or as part of quality assurance.

### A.2 Setup

A pre-rooted Google Pixel 6a phone should be used when using the playbook for evaluating API security. The device is locked with the code: 000. If the provided phone has been updated, or a new un-rooted phone is being used, the phone must first be rooted. This is done following these steps:

1. First, the stock boot image for the currently running Android version must be retrieved. This can be found here [To find the correct Android version](#); go to setting on a google pixel phone, then tap About device and look for the build number.
2. In the folder image-bluejay-[version].zip, find the file boot.img and store it on the phone.
3. Download Magisk on the phone from [Magisk](#)
4. In the Magisk app, choose Install, and then "Select and Patch a File". Here you choose the boot.img which was transferred in the previous step.
5. A new file called magisk\$\_patched\_[random\_strings].img\$ will now appear on the phone. This file must be transferred back to the computer.
6. Next, you need to unlock the bootloader.
  - (a) Go to settings, about phone, and find the build number.
  - (b) Tap the build number seven times to activate developer mode.
  - (c) Now, go back to system and developer options.
  - (d) Activate the OEM unlocking.
  - (e) Turn off the phone.
7. Hold down the power and volume buttons simultaneously to boot back on into boot-loader mode.

Now that the phone is rooted, the app RProxid can be used together with a mitmProxy. One mitmProxy that can be used is <https://mitmproxy.org/>. Follow the installation guide.

To set up the mitm proxy, go to terminal or similar, and input "mitmdump -w output\_file.pcap" or "mitmweb". This will write the logs to the file output\_file.pcap. If this file does not exist, it must be made. Go to the settings in the device and choose network. Tap edit and advanced settings to find the proxy settings. Set proxy to manual and enter the IP address to the device and computer are connected to. Write port 8080 as port, if not any other port is chosen manually in the mitmproxy. The logs from the device should now appear in the terminal.

## A.3 Questionare

### A.3.1 Network traffic

This section focuses on questions related to the traffic flow from the application:

1. **Encryption:**Is sensitive personal information encrypted? (If yes: 3, no: 0)
2. Is non-sensitive personal information encrypted? (If yes: 2, no: 0)
3. **Encryption:**Is non-personal information encrypted? (If yes: 1, no: 0)
4. **Locations:**How many locations receive the collected data? (Manual evaluation)
5. **Location:**Where is the collected data sent? (If EU/EØS: 2, USA (o.l.): 1, "Liste over høysikkerhetsland": 0, other: manual evaluation)
6. **Necessary locations:**How many of the locations of the total are necessary to process the data? (If All: 2, some: 1, non: 0)

Total possible set points: 10

This section focuses on how what and where collected data is stored:

1. **Location stored:**Where is the data collected stored? (If EU/EØS: 3, USA (o.l.): 2, "Liste over høysikkerhetsland": 0)
2. **Duration of storage:**For how long is the data collected stored? (If Until purpose fulfilled: 1, Undecided: 0)
3. **Anonymization**Is sensitive personal data stored anonymized? (If yes: 2, no: 0)
4. **Anonymization:**Is personal data stored anonymized? (If yes: 1, no: 0)

### A.3.2 Terms/Conditions of Use

This section focuses on the conditions of use for the application.

1. **Data Sharing Consent:** Does the user have to accept that their data is being shared with third parties?
  - (a) If yes:
    - i. Is the data anonymized? (If yes: 2, no: 0)
    - ii. Are third parties clearly identified, and is the purpose of data sharing specified? (If yes: 2, no: 0)
  - (b) If no: 4
2. **Data Retention:** Does the application clearly state the duration for which user data will be stored? (If yes: 2, no: 0)

3. **Data Deletion:** Is there a clear procedure for users to request the deletion of their data? (If yes: 2, no: 0)
4. **User Rights:** Are users' rights concerning their data, such as the right to access, correct, or object to processing, clearly stated and easy to exercise? (If yes: 2, no: 0)
5. **Usage Restrictions:** Does the application impose reasonable restrictions on its use, such as prohibiting the use of the application for illegal activities or harming others? (If yes: 1, no: 0)
6. **Modification of Terms:** Is there a clear procedure for notifying users of changes to the terms and conditions? Are users given the opportunity to review and accept new terms before continuing to use the application? (If yes: 1, no: 0)

Total possible points: 12

### A.3.3 Bluetooth security

Applies if device uses Bluetooth for connections

1. **Bluetooth Version:** Which version of Bluetooth is being used? (If v5.0: 1.5, v4.0: 1, lower: 0)
2. **Pairing Method:** What method is used for pairing the devices? (If Secure Simple Pairing (SSP) or LE Secure Connections (LESC): 2, legacy pairing: 0)
3. **Encryption:** Is the Bluetooth connection encrypted? (If AES-CCM with a key length of 128 bits or higher: 2, lower encryption strength or no encryption: 0)
4. **Authentication:** Is there a strong authentication mechanism in place to prevent unauthorized devices from connecting? (If yes: 2, no: 0)
5. **Signal Range:** Does the application limit the Bluetooth signal range to minimize the risk of unauthorized connections or eavesdropping? (If yes: 1, no: 0)
6. **Bluetooth Low Energy (BLE) Privacy:** If using BLE, is the privacy feature enabled to protect against tracking by randomizing device addresses? (If yes: 1, no: 0)
7. **Firmware Updates:** Does the device support secure firmware updates to fix potential Bluetooth security vulnerabilities? (If yes: 1, no: 0)

Total possible points: 8.5

### A.3.4 Token Validation in API Security

1. **Token Type:** What type of token is used? (If OAuth 2.0 access tokens or JWT: 3, other standard tokens: 2, custom tokens: 1, no token: 0)
2. **Token Expiration:** Are tokens set with an appropriate expiration time? (If yes: 2, no: 0)
3. **Token Revocation:** Is there a mechanism for revoking tokens when necessary? (If yes: 2, no: 0)
4. **Token Storage:** Is the token securely stored on the client-side? (If stored in HttpOnly cookies or secure client-side storage: 2, insecure storage: 0)



5. **Token Transmission:** Are tokens transmitted securely, e.g., over HTTPS? (If yes: 2, no: 0)
6. **Token Validation:** Is the token validation process robust and follows best practices, such as checking for token signature, issuer, audience, and other claims? (If yes: 2, no: 0)
7. **Scope Management:** Are token scopes managed correctly, allowing access only to necessary resources and actions? (If yes: 2, no: 0)
8. **Error Handling:** Are token validation errors appropriately handled, with clear and informative error messages returned to the client without revealing sensitive information? (If yes: 1, no: 0)

Total possible points: 14

This token validation chapter assesses various aspects of token handling and validation in API security. Evaluating these factors helps to ensure that API access is secure and only granted to authorized clients, minimizing the risk of unauthorized access or data breaches.

## A.4 How to find answers?

To find and analyze the data to answer the questions set up the mitmproxy as described in 8.2 Setup. The file `output_file.pcap` can be opened in Wireshark from File, and then Open. Select the file `output_file.pcap` to open the logs from mitm. When using "mitmweb", the mitmproxys web interface will open in the browser.

### A.4.1 Network traffic

#### 1-3. Encryption:

- In mitmproxy, inspect the captured HTTP requests and responses.
- Look for any sensitive or non-sensitive personal information being transmitted.
- Open traffic in Wireshark
- Apply filter "`tls.record.content_type == 23`" to display only packets containing TLS application data, which includes the encrypted payload.
- Inspect packets to see in the content is encrypted.

#### 4. Locations:

- In mitmweb, analyze the captured packets and identify the destination IP addresses.
- Count the number of unique destination IP addresses to determine how many locations receive the collected data.

#### 5. Location:

- Perform a geolocation lookup for each unique destination IP address. Use online tools like <https://www.iplocation.net/> for this.
- Assign points based on the location of the destination IP addresses.

#### 6. Necessary locations:

- Research each destination IP address to determine its purpose.

- Evaluate if the location is necessary for processing the data and assign points accordingly.

Note that some of the information might not be visible if the application uses strong encryption (e.g., TLS) for all its network communications. In such cases, you might need to use additional tools or techniques, such as SSL/TLS decryption in Wireshark or certificate pinning bypass for the mobile application, to analyze the encrypted traffic.

### SSL/TLS decryption guide for Wireshark:

#### 1. Obtain the private key or pre-master secret:

- For SSL/TLS decryption, you'll need either the server's private key (in the case of RSA key exchange) or the pre-master secret log file (in the case of Diffie-Hellman key exchange).
- If you have access to the server's private key, export it in PEM format.
- If the server uses Diffie-Hellman key exchange, you'll need to obtain the pre-master secret log file. For example, in Google Chrome or Mozilla Firefox, you can set an environment variable before starting the browser to generate this file:  
`SSLKEYLOGFILE=path/to/sslkeylog.log`

#### 2. Start Wireshark and open a capture file or begin a live capture:

- Open Wireshark and either open a previously saved capture file containing SSL/TLS traffic or start a new live capture session.

#### 3. Configure Wireshark for SSL/TLS decryption:

- Go to **Edit > Preferences > Protocols > SSL** (or TLS, depending on your Wireshark version).
- If you have the server's private key in PEM format, click on the **Edit** button next to RSA keys list, and then click on the **New** button. Enter the IP address and port of the server, the protocol (e.g., `http`), and the path to the private key file. Click **OK** to save the settings.
- If you have the pre-master secret log file, set the **(Pre)-Master-Secret log filename** field to the path of the log file.
- Click **OK** to save your preferences.

#### 4. Analyze decrypted SSL/TLS traffic:

- Once you've configured Wireshark for SSL/TLS decryption, you should be able to see decrypted data in the packet details pane for the SSL/TLS packets. You can then use Wireshark's display filters and analysis tools to inspect the decrypted traffic.

#### 1. **Location stored:** Where is the data collected stored? (If EU/EØS: 3, USA (o.l.): 2, "Liste over høysikkerhetsland": 0)

To find the answer, inspect the network traffic using Wireshark and mitmproxy. The server's IP address and domain name should provide information about the location of data storage. You can also discuss this question with the application owner.

#### 2. **Duration of storage:** For how long is the data collected stored? (If Until purpose fulfilled: 1, Undecided: 0)

Inspect any terms and conditions or privacy policies associated with the application to find information on data retention policies. You may also need to discuss this question with the application owner.

3. **Anonymization:** Is sensitive personal data stored anonymized? (If yes: 2, no: 0)  
Analyze the data being transmitted between the application and the server using mitm-proxy and Wireshark. Check if any sensitive personal information is being anonymized before being stored. You may also need to discuss this question with the application owner.
4. **Anonymization:** Is personal data stored anonymized? (If yes: 1, no: 0)  
Similar to the previous step, analyze the data being transmitted and check if personal data is anonymized before being stored. Discuss this question with the application owner if necessary.

#### A.4.2 Terms/Conditions of use

##### 1. Data Sharing Consent:

- (a) Check the application's terms of service, privacy policy, or any consent dialogues during the sign-up process to see if users have to accept that their data is being shared with third parties.
- (b) If yes, look for information about whether the data is anonymized and whether third parties are clearly identified, along with the purpose of data sharing.
- (c) If no, assign a score of 4.

##### 2. Data Retention:

- (a) Review the application's privacy policy or terms of service to find information about the duration for which user data will be stored.
- (b) If the duration is clearly stated, assign a score of 2. If not, assign a score of 0.

##### 3. Data Deletion:

- (a) Check the application's privacy policy, terms of service, or user settings to find information about the procedure for users to request the deletion of their data.
- (b) If there's a clear procedure, assign a score of 2. If not, assign a score of 0.

##### 4. User Rights:

- (a) Review the application's privacy policy or terms of service to find information about users' rights concerning their data, such as the right to access, correct, or object to processing.
- (b) If users' rights are clearly stated and easy to exercise, assign a score of 2. If not, assign a score of 0.

##### 5. Usage Restrictions:

- (a) Examine the application's terms of service or any usage guidelines to find information about any imposed restrictions on its use, such as prohibiting the use of the application for illegal activities or harming others.
- (b) If there are reasonable restrictions, assign a score of 1. If not, assign a score of 0.

##### 6. Modification of Terms:

- (a) Check the application's terms of service, privacy policy, or any in-app notifications to find information about the procedure for notifying users of changes to the terms and conditions.

- (b) If there's a clear procedure and users are given the opportunity to review and accept new terms before continuing to use the application, assign a score of 1. If not, assign a score of 0.

In most cases, you'll need to review the application's privacy policy, terms of service, and user interfaces to find the answers to these questions. Be prepared to discuss your findings with the application owner if necessary.

### A.4.3 Bluetooth Security

#### 1. Bluetooth Version:

- Check the device's specifications or documentation to find the Bluetooth version being used. This information may also be found in the device's settings or configuration menu.

#### 2. Pairing Method:

- Review the device's documentation or perform a hands-on test of the pairing process. Look for indications of Secure Simple Pairing (SSP) or LE Secure Connections (LESC) in the device settings or documentation.

#### 3. Encryption:

- Check the device's documentation for information about encryption. You can also use a Bluetooth sniffer like Ubertooth One or Adafruit Bluefruit LE Sniffer along with Wireshark to capture and analyze Bluetooth packets to determine the encryption used.

#### 4. Authentication:

- Investigate the device's documentation and settings for any authentication mechanisms. Perform hands-on tests to determine whether unauthorized devices can connect to the target device.

#### 5. Signal Range:

- Test the device's signal range by moving the connecting device away from the target device and noting the maximum distance at which a connection is maintained. Check the device's documentation for any features that limit Bluetooth signal range.

#### 6. Bluetooth Low Energy (BLE) Privacy:

- Review the device's documentation to determine whether the BLE privacy feature is enabled. You can also use a Bluetooth sniffer to capture and analyze packets, looking for randomized device addresses.

#### 7. Firmware Updates:

- Check the device's documentation or settings for information about firmware updates. Determine whether the device supports secure firmware updates to fix potential Bluetooth security vulnerabilities.

#### A.4.4 Token Validation in API security

1. **Token Type:** Inspect the network traffic using mitmproxy or Wireshark. Look for the “Authorization” header in the HTTP requests or any token-related information in the API responses. Check if the tokens used are OAuth 2.0 access tokens, JWT, other standard tokens, custom tokens, or if no token is used.
2. **Token Expiration:** Analyze the token content (if it’s JWT, you can use a JWT decoder like jwt.io) and look for the “exp” claim or any other field that indicates the token’s expiration time. Check if the expiration time is appropriate for the application’s security requirements.
3. **Token Revocation:** Inspect the API documentation, responses, and requests to find evidence of a token revocation mechanism (e.g., a token revocation endpoint). You may also need to discuss this question with the application owner.
4. **Token Storage:** Examine the client-side code of the application (e.g., JavaScript) to determine how the token is being stored. Check if it’s stored in HttpOnly cookies or secure client-side storage, such as localStorage or sessionStorage with proper security measures.
5. **Token Transmission:** Inspect the network traffic using mitmproxy or Wireshark and ensure that tokens are transmitted over HTTPS by checking if the request URLs have “https://” as the protocol.
6. **Token Validation:** Review the server-side code (if accessible) or API documentation to verify if the token validation process follows best practices like checking for token signature, issuer, audience, and other claims. You may also need to discuss this question with the application owner.
7. **Scope Management:** Examine the token content (e.g., using a JWT decoder) and look for the “scope” claim or any other field that indicates the allowed resources and actions. Check if the scopes are managed correctly and provide access only to necessary resources and actions.
8. **Error Handling:** Trigger token validation errors by sending invalid tokens or expired tokens in the API requests. Observe the error responses and verify if they provide clear and informative messages without revealing sensitive information.