NTNU

Norwegian University of
Science and Technology

Analysis of Mobile Application's Compliance with the General Data Protection Regulation (GDPR)

Peder Lind Mangset

Master of Science in Communication Technology
Submission date: June 2018
Supervisor: Stig Frode Mjølsnes, IIK
Co-supervisor: Ruxandra-Florentina Olimid, IIK

Norwegian University of Science and Technology
Department of Information Security and Communication Technology

Title:	Analysis of Mobile Application's Compliance with the General Data Protection Regulation (GDPR)
Student:	Peder Lind Mangset

Problem description:

The General Data Protection Regulation (GDRP) is a regulation by which the European Parliament and the European Council aim to strengthen and unify the data protection requirements within Europe. GDRP also applies to Norway as a member of the European Economic Area (EEA). The enforcement date of the GDRP is 25 May 2018, time when the organizations must ensure their policies comply to the regulation. More precise, all aspects of data transmission, processing and storage must adhere to the new legislation. With mobile applications being a fundamental part of the digitalized systems nowadays, it is important to know which measures must be taken to ensure the confidentiality of the data processed in the context of the GDPR and if they are correctly implemented.

The master thesis will investigate and analyze the compliance of some mobile network applications with the GDRP. The aim is to identify flaws in mobile applications that result in the application not being compliant with the new legislation, or the opposite, highlight good practices with respect to the implementation of the regulations. To fulfill his goal, the student will test the functionality of the applications, analyze their software code, investigate their structure, extract information (encryption keys, passwords, etc.) that can be used to gain access to private data, etc. The student will focus on the possible disclosure of personal data at rest and in transit.

Responsible professor:	Stig Frode Mjølsnes, IIK
Supervisor:	Ruxandra-Florentina Olimid, IIK

Abstract

Users increasingly rely on their mobile applications to fulfill everyday activities. Processing of personal data through such tools poses a significant risk to the user's privacy and security. This stems mainly from the various sensors on the device, but also from the nature of it, because they are physically difficult to secure. As a result of this, implementing the General Data Protection Regulation (GDPR) into mobile applications may pose serious challenges.

This study focused on how pharmaceutical and dating applications process user's personal data and if they do so in compliance with the GDPR. We followed a design science methodology and evaluated each application using predefined test cases. Our study revealed instances of personal data stored unencrypted on the device. This included user's social security number and sensitive personal data, such as political opinion and religious belief. This type of data warrants special consent under the new regulation. It further revealed that multiple application does not allow users to opt-out of automatic individual decision-making for direct marketing purposes. Lastly, the study revealed applications that have been updated specifically for the GDPR.

The majority of the work for this study was conducted before the implementation date. It is therefore difficult to predict how Norway's supervisory authority will impose sanctions on infringements of the regulation. However, our study revealed infringements of provisions that are eligible for the administrative fines outlined by the GDPR.

Sammendrag

Brukere blir mer og mer avhengig av mobile applikasjoner for å utføre hverdagslige aktiviteter. Prosesseringen av personlig data gjennom slike verktøy kan utgjøre en betydelig risiko med tanke på brukerens personvern. Dette stammer hovedsakelig fra de ulike sensorene på enheten, men også fordi enheten er fysisk vanskelig å sikre. Som et resultat av dette kan det være utfordrende å implementere GDPR i mobilapplikasjoner.

Denne oppgaven fokuserer på hvordan farmasøytiske og dating applikasjoner prosesserer brukeres personlige data og om prosesseringen er i samsvar med GDPR. Ved å følge en Design Science Methodology evaluerer vi applikasjoner ved bruk av predefinerete tester. Testene våre viser forekomster hvor personlig data lagres ukryptert på enheten. Dette inkluderer brukerens personnummer og sensitive personopplysninger. Sensitive personopplysninger er opplysninger om blant annet brukerens trosoppfatning og seksuell orientering. Denne type opplysninger setter strengere krav for samtykke til prosessering. I tillegg, lar ikke applikasjoner brukere avstå fra automatisk beslutningstaking for direkte individuell markedsføring. Oppgaven viser også applikasjoner som har blitt oppdatert spesielt for GDPR.

Majoriteten av arbeidet med denne oppgaven ble utført før implementeringsdatoen for GDPR. Det kan derfor være vanskelig å predikere hvordan Datatilsynet pålegger sanksjoner for overtredelser av reguleringen. Allikevel berettiger våre funn noen av de økonomiske sanksjonene definert i GDPR.

Preface

This thesis is written as the final part of a Master's degree at the Norwegian University of Science and Technology (NTNU) in the faculty of Information Technology and Electrical Engineering. The work was carried out from January to the beginning of June 2018. Guidance and supervision were performed by Prof. Stig Frode Mjølsnes and Postdoc. Ruxandra-Florentina Olimid.

The topic under discussion is of high relevance for any organization or other body that determines the purposes or means of processing of personal data. Especially data collected through a mobile application.

Contents

List of Figures

List of Tables

List of Algorithms

List of Acronyms

API Application Programming Interface.

EEA European Economic Area.

EFTA European Free Trade Association.

EULA End User License Agreement.

GDPR General Data Protection Regulation.

GUI Graphical User Interface.

GUID Global Unique Identifier.

HTTP Hypertext Transport Protocol.

HTTPS Hypertext Transport Protocol over Transport Layer Security.

IMEI International Mobile Equipment Identity.

IPC Inter Process Communication.

NTNU Norwegian University of Science and Technology.

OS Operating System.

SDK Software Development Kit.

SSL Secure Socket Layer.

TLS Transport Layer Security.

UID Unique Identifier.

Chapter 1
Introduction

In this chapter we explain the motivation for this study and state our goal and research question. Further, we proceed to explain how we plan to answer our research question by introducing our research methods.

1.1 Motivation

Mobile devices collect and process increasingly more information about their users. They have become the preeminent devices in which we store our personal data. With the introduction of various sensors like the GPS, camera and fingerprint scanning, the easiness of collecting large quantities of data have skyrocketed. Subsequently, this introduces significant data protection risks and a potential lack of safeguards. Especially concerning mobile applications, because they are often untrusted software running on a trusted platform. Mobile applications have the power to access, transfer and store data, often without the user knowing. Although certain mitigations have been introduced, it is hard to know what an application actually does with the permissions it is granted. In recent years, multiple applications have been criticized for being overprivileged. Among the most famous was the "Brightest Flashlight Free." The application quietly sent geolocation, along with persistent device identifiers to third parties, including advertising networks [23].

It is the processing of this type of data that the GDPR addresses [19]. In fact, not only does it cover data processed through a mobile device, it covers all information systems that process personal data. All data controllers, ie. public authorities, agencies or other body that processes personal data of a European citizen, must comply to this new regulation. The GDPR is set to replace The Data Protection Directive (95/46/EC) [17]. A lot has happened in 23 years. Internet usage and technology advances have led to the processing of data on a previously unprecedented scale. There is a need for an updated unified directive to protect individuals' rights online. The philosophy behind the GDPR and its main goal is to give Europeans

back control of their personal data. To give individuals a more transparent insight into how their personal data is collected, stored, transmitted and processed.

Our main motivation for this study is two-fold. First, we want to determine how pharmaceutical and dating applications process user's personal data. The reason to focus on pharmaceutical and dating apps will be further explained in Sect. 3.5, but in short, it stems from article 9 of the GDPR on the processing of special categories of data. We chose Android as our preferred platform because of the open-source nature of the Operating System (OS). In fact, Androids whole software stack is available for download [1]. Secondly, we want to investigate how the GDPR impacts the processing of personal data in mobile applications. Since the GDPR cuts across all technologies and platforms, how does it impact the computer programs that process the most personal data about an individual.

1.2 Research Goal

Based on the motivation, we define the following goal:

G: Identify how pharmaceutical and dating applications process personal data with respect to the GDPR.

We set our goal to also include good implementations concerning the requirements stipulated by the GDPR. Highlighting behavior that show lawful, transparent and fair processing of personal data. Achieving our goal will require knowledge of the Android platform and how to perform privacy and security related tests on mobile applications. In addition, it will require knowledge of the GDPR and how its legal requirements can be translated into technical solutions.

To reach **G** we address the following research question:

RQ: How does pharmaceutical and dating Android applications implemented the regulatory requirements of the GDPR, and are they in their current state compliant?

1.3 Research Methods

To reach our goal and answer the research question, we divide our research into two parts. First, we conduct a literary review on Android applications and its security mechanisms. This part also includes the study and understanding of the GDPR as a legal document. The second part will follow a design science research

[1] https://android-review.googlesource.com/q/status:open

methodology where we evaluate the mobile applications towards predefined test cases [26]. Methodology will be further discussed in Chapt. 3.

1.4 Legality

During the work of the study there will be a need to reverse engineer Android application to obtain a reconstruction of the source code. We dive further into this topic in Sect. 3.8.1. There are however, some important legal consideration that must be taken into account. Many of the applications in the Google Play store have an End User License Agreement (EULA). The EULA lays restrictions on the legal aspect of tampering with the software. For example, in the EULA of the popular application Snapchat, it is stated:

> *"You may not copy, modify, distribute, sell, or lease any part of our Services, nor may you reverse engineer or attempt to extract the source code of that software, unless laws prohibit these restrictions or you have our written permission to do so."* [32]

The part *"unless laws prohibit these restrictions"* is the part of interest. The European Union directive 2009/24 defines the legal protection of computer programs within the EU [18]. In short, it states that a person having the right to use a copy of a computer program shall be entitled, without the authorization of the rightsholder to observe, study or test the functioning of a program in order to determine the ideas and principles which underline any element of the program . . . , article 5, par. 3 [18]. Few applications examined in this study had an EULA and none mentioned reverse engineering or decompiling.

The reason to obtain an approximation of the original source code is to better understand, study and observe the behavior of the application. At no point will there be any modification, reproduction or distribution of the application. If by any chance this study reveals major security vulnerabilities in the server-side leading to the exposure of people's personal data, the data controller will be notified. However, we foresee that scenario to be unlikely because our focus is on the client-side. Lastly, this study is for educational purposes and we hope that our findings will help Android developers better protect user's personal data.

1.5 Outline

The remainder of the thesis is structured as follows:

Chapter 2 introduces the GDPR and how it relates to mobile applications. Further, it introduces the Android platform and its security mechanisms. Lastly, it covers basic principles behind mobile application security testing.

Chapter 3 explains the testing methodology used. It describes processes and techniques to analyze the security of mobile applications. These processes are represented as test cases.

Chapter 4 introduces tools needed to perform vulnerability analysis of Android applications.

Chapter 5 contains the findings of the applications tested using the processes and techniques presented in Chapt. 3.

Chapter 6 discusses the most important findings of Chapt. 5 and revisits the studies limitations.

Chapter 7 presents concluding remarks and proposals for future work.

Chapter 2
Background

This chapter introduces background theory relevant to this study. It is divided into three parts. First, we cover the GDPR and its legal requirements. Then, we study the Android platform. Lastly, we introduce key terminology behind mobile application security testing.

2.1 GDPR

The GDPR will be enforced on the 25th of May. Norway as a member of the European Free Trade Association (EFTA) has yet to implement the regulation because it has not been incorporated into the European Economic Area (EEA) Agreement [16]. However, Norwegian data controllers must follow the GDPR if they offer goods or services to European citizens, see Sect. 2.1.2. The regulation replaces the Data Protection Directive (95/46/EC) and was designed to harmonize data privacy laws across Europe while strengthening the privacy rights for the individual. The 88-page document consists of 173 recitals and 99 articles that range from the underlying principles of data processing, to the roles and responsibility of the supervisory authority.

2.1.1 Replacing the Data Protection Directive

So why is a new regulation needed, and why now? Even after the Data Protection Directive (95/46/EC) of 1995, which the GDPR replaces, the levels of data protection varied among EU member states [27]. Directives differ from regulations in that member states are not legally obligated to implement them. Regulations have binding legal force throughout all member states. Member states had to transpose the directive into internal law. England was the first with their Data Protection Act of 1998, Germany followed with their own equivalent, then France. This lead to a variety of different regulations and the laws of one jurisdiction was never equivalent to another.

Since then, internet usage has become more widespread, and technological advances like cloud storage have changed the way we process data. A cloud service provider can move data between multiple facilities and countries. The scale of data collection and the sharing of personal data has increased significantly. Both privately owned companies and state authorities make use of personal data on an previously unprecedented scale in order pursue their activities. These developments require a strong unified data protection framework. A framework that all member states are legally bound too. It must be backed by strong enforcement. This will create a trust that will allow the digital economy to develop across the internal markets.

2.1.2 Coverage

Article 3 specifies the terrestrial scope of the regulation. To summarize, it applies to anyone who in the context of activities of an establishment of a controller or processor in the Union, regardless of whether the processing takes place in the Union, article 3 par. 1. It also applies to controllers not established in the Union, if the processing of data is related to the offering of goods or monitoring of their behavior, article 3 par. 2. There are some exceptions. The regulations do not apply to people processing personal data in the course of exclusively personal or household activities. In other words, you are not subject to the regulation by keeping peoples contact information on your phone, or by installing surveillance cameras around your house. The difference lies in economic activity. The processing must be part of an enterprise, article 4 par. 18. In addition, there is an exclusion from some parts of the GDPR for organizations and enterprises with fewer than 250 employees, article 30 par. 5. These companies are not required to maintain a record of processing activities, unless the processing is likely to result in a risk to the rights and freedoms of the data subjects. Or the processing includes special categories of data.

2.1.3 Sanctions and Penalties

Each member state must appoint a supervisory authority. Its primary job is to monitor and enforce the regulation. It is within the supervisory authority's power, to issue warnings and reprimands to any controller who fail to comply with the regulation. In addition, depending on the circumstance of each individual case, to impose administrative fine pursuant to article 83.

Article 83 specifies the general conditions on which a supervisory authority can impose administrative fines. These fines can be in addition to, or instead of the warnings and reprimands of article 58. The severity of each fine depends on the circumstance of each individual case. How many data subjects were affected? What level of damage did they suffer? Is there any intentional or negligent character related to the infringement? There are two degrees of administrative fines. The

first covers violations regarding the obligations that the controller and processor have. For example, the processing of data for individuals under the age of 16, privacy by design and default, security of processing, etc. Such infringements shall be subject to administrative fines up to 10 000 000 EUR, or up to 2% of the total annual turnover of the preceding financial year, whichever is higher. The second degree of administrative fines cover violations regarding the fundamental principles of processing. Administrative fines of this severity are for breaking the fundamental principles of processing of personal data, principles like lawfulness, fairness and transparency. Infringements of these provisions shall be subject to administrative fines of up to 20 000 000 EUR, or up to 4% of the total annual turnover of the preceding financial year, whichever is higher.

2.1.4 Principles

Article 5 of the GDPR states fundamental principles regarding the processing of personal data. They apply to every processing activity and is the cornerstone of the regulation. In the following section we briefly introduce these principles. Many of them are directed towards data controllers, however they are key terminology that will be used later in the study.

Lawfulness, fairness and transparency: Only process data when there is an appropriate legal basis or legislative measure. Strictly speaking, the only legitimate grounds to processed personal data are when explicit consent is given. The controller must ensure that any information or communication to the data subject is concise and easy to understand. Especially when the information is addressed to a child.

Purpose: Data controllers must have a legitimate purpose for the processing of personal data in the first place. It shall only be collected for a specific, explicit and legitimate purpose and not further processed in a manner that is incompatible with that purpose. If the data controller wish to process the data further, *the other purpose* have to be compatible with *the original purpose* for which the data was collected. This must be clearly communicated to the data subject, highlighting the possible consequences of the further processing.

Minimization: Limit data storage to what is necessary in relation to the purpose for which personal data is collected. A time limit for deletion or a periodic review is necessary to uphold this principle.

Accuracy: Data controllers should take any reasonable step to ensure that personal data is accurate and where necessary kept up to date. Inaccurate data should be erased or rectified without undo delay. To comply with this, data controlers must have policies in place to audit how they will maintain the personal data they process.

Storage Limit: This refers to the obligation to keep the data in a form which permits identification for no longer than is necessary for the purposes of which the personal data were collected. This principle encourages limits on how data is moved and stored.

Integrity and Confidentiality: Personal data should be processed in a manner that ensure appropriate security. This includes protection against unauthorized or unlawful processing, accidental loss or destruction. Negligence is no longer an excuse under the GDPR, so data controllers must spend adequate resources to ensure data confidentiality and integrity.

Accountability: The controller shall be responsible for and be able to demonstrate compliance with the regulation. For data controllers, this is easier said than done. There is a range of processes that organizations must put in place to demonstrate compliance. This may include creating a personal data inventory, a data breach reporting mechanism, etc.

2.1.5 Rights of the Data Subject

Chapter III of the GDPR presents the rights of the data subject. Whenever data is collected from a subject, the controller shall, at the time when personal data is obtained, provide the subject with the following information. The following list is adapted from article 13 [19]:

1. Where applicable, the identity and contact details of the data controller and the data protection officer.
2. The purpose of processing and the recipients of the personal data.
3. If the controller wish to transfer data to a third party or international organization, the data subject must be informed of the appropriate safeguards taken and the means in which to obtain a copy of them.
4. Information about the period of which the data will be stored, if that's not possible, the criteria used to determine that period.
5. The existence of the right to rectification, the right to erasure, the right to data portability
6. The right to withdraw consent at any time.
7. The right to lodge a complaint with the supervisory authority.
8. The existence of automatic individual decision making and meaningful information about the logic involved, as well as the envisaged consequences of such profiling for the data subject.

Article 12 defines how the information above should be presented to the data subject. It states that the information should be provided in a concise, transparent,

intelligible and easily accessible form, using clear and plain language, in particular for any information provided specifically to a child, article 12, par. 1.

In the Android application ecosystem there are a couple of alternatives to how developers can communicate the required information [20]. First is the applications privacy policy, which should be easily accessible in the app and uphold all the requirements of article 12 and 13. Second, is a link to the privacy policy of the application through the Google Play Store. Third, is through the permission based model that Android enforces, see Sect. 2.2.2. And fourth, directly through the application regarding user input and explanations of the apps functionality when interacting with the app.

2.1.6 Managing Consent

To process personal data the data controller must have legal basis. For applications in the private sector, this means obtaining the data subjects consent. Recital 32 lays out some guidelines on how controller can obtain consent. It states:

> *"Consent should be freely given by a clear affirmative act establishing a freely given, specific, informed and unambiguous indication of the data subject's agreement to the processing of personal data relating to him or her, such by a written statement, including by electronic means, or an oral statement."* [19]

The recital further goes on to specify what does not hold the legal ground for consent. This includes silence, pre-ticked boxes and inactivity. Furthermore, consent for the processing of personal data, should not be mixed with consent relating to other statements of agreement with the data controller. In the Android ecosystem consent is strongly linked to user-based permissions.

2.1.7 Article 32, Security of Processing

Article 32 addresses the nature and scope of security related to the processing of personal data. The article does not specify any technical requirements, because the GDPR must be a timeless document. It must be vague enough to be transferable to new technologies. Paragraph 1 letter (b), states that the controller and processor must ensure the ongoing confidentiality, integrity, availability and resilience of processing systems and services. This implies that data controllers and processors must constantly update their systems to provide ongoing security. In addition, par. 2 states that when assessing the appropriate level of security, special considerations must be taken into account regarding the risk that are presented from accidental destruction, loss, alteration, etc. This means is that data controller must always

asses the risk related to the types of personal data they process and the consequences for the data subject, should the personal data be compromised.

2.2 The Android Ecosystem

In this section we will introduce Android platform architecture, its security features and the structure of Android applications. Moreover, we introduce key stakeholders in the Android ecosystem and what role they play in regards to the GDPR.

2.2.1 Platform Architecture

Figure 2.1 shows the Android software stack. It is open-source and made up of five different layers, each layer provides services to the layer above.

Linux Kernel:
The foundation the Android platform is a modified version of the Linux kernel [7]. The kernel provides low level functionality like threading, memory management, scheduling and Inter Process Communication (IPC). With the adaptation of the Linux kernel comes key security features, see Sect. 2.2.2.

Hardware Abstraction Level:
The level above the kernel is the HAL. It defines standard libraries for communicating with device hardware. It allows for third party applications to use the built-in hardware functions on the device, like the microphone, camera, etc.

Android Runtime:
Programs for Android are commonly written in Java. Java code is compiled into Dalvik bytecode and stored in a `.dex` file, which is somewhat different from normal Java bytecode. This is necessary because mobile devices have restrictions regarding memory size and processor speed. With the release of Android 5.5, the previous Dalvik Virtual Machine was replaced with Android Runtime. ART uses the same Dalvik bytecode to obtain backwards compatibility.

Natice C/C++ Libraries:
Many of the system components, such as the HAL and ART uses native code that requires native libraries written in C or C++.

Android Framework:
This layer contains all the API's that are available to applications. Java libraries that supports logging, xml, sql, text, regex, etc.

System Applications:
Android comes with a core set of pre-installed applications. They offer excellent

Figure 2.1: Overview of the Android software stack. The foundation is a modified version of the Linux kernel. The HAL defines standard libraries for communicating with hardware. The ART and Native C++ libraries make up the application runtime environment and native libraries, respectively. The Java API Framework contains API's that form the building blocks of Android applications. At the top, is the set of core system apps that come pre-installed. Taken from Android Developers [7].

value to their users, but also key capabilities to developers, which can access them from their own applications. For example, the camera application can be accessed by a third-party messaging app.

2.2.2 Android Security Features

Since the operating system is based on Linux it inherits a lot of security features from the OS. These features will be explained below. This study is not concerned with security related to Googles vetting of application in the Google Play Store, nor any security features introduced by third party antivirus software. Only the mechanisms offered by the OS alone.

Android Users and Groups:
Android inherits Linux's users and groups, but not in the same way. In Android, the multi-user support of Linux is implemented to sandbox applications. Each application is assigned a unique Unique Identifier (UID) on installation. It keeps this Graphical User Interface (GUI) for the duration of its installation. Figure 2.2 shows that an example were Wordfeud get assigned the UID `u0_a109`.

Application Sandbox:
Applications executes isolated from each other, this is enforced through the UID. This sets up a kernel-level application sandbox. The default is that applications cannot interact with one another, nor access system resources. Like all security features, the Application Sandbox is not unbreakable. However, to break the Application Sandbox in a properly configured device, one must compromise the security of the Linux kernel [9].

User Based Permissions:
If an application wants to access system resources or personal data, the application must request the corresponding API's. There are four different protection levels that permissions are divided into. Normal permissions can be granted automatically, signature permission is granted within the same sandbox, dangerous permissions are inferred to those granted by the user and system permissions are granted to pre-installed apps. To make use of these API's, an application must declare these permissions in its `AndroidManifest.xml` file. The user-based permission system changed in Android Marshmallow [6]. In previous releases the user was prompted with a list of all the permissions the application requested at installation time, it was a take-it-or-leave-it scenario. If the user did not accept all the permissions, the application would not install. This changed with the release of Marshmallow, now an opt-in system is used instead. Users are prompted to grant individual permissions whenever the resource is needed by the application for the first time. This allows for a more granular permission based system. However, applications developed for the previous SDK's will continue to use an all-or-nothing approach.

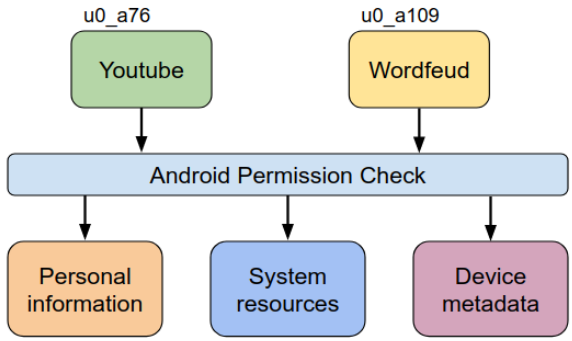

Figure 2.2: The Android permission check. Every time an application wants to access sensitive user data or certain system features, the OS performs a check to see if the user has granted the application that specific permission. Adapted from the Android Open Source Project [8].

2.2.3 Application Architecture

All Android application are distributed with the `.apk` file format. It is basically a `zip` file. The `.apk` file of installed apps are located in `/data/app/<package-name>/base.apk`. For example the `.apk` file of the Vitusapotek app examined in Sect. 5.1, is located in `/data/com.vitusapotek.eresept/base.apk`. The APK format is an archive that contains all resources required for the application to function, see Source Code 2.1.

Source code 2.1 The internal file structure when unzipping `.apk` file.

```
$ unzip example_app.apk -d example_unzip
$ cd example_unzip
$ ls -oh
total 6,5M
-rw-rw-rw-  1 root  11K des.  31  1979 AndroidManifest.xml
drwxrwxr-x  3 root 4,0K feb.  28 13:22 assets
-rw-rw-r--  1 root 5,9M des.  31  1979 classes.dex
drwxrwxr-x  8 root 4,0K feb.  28 13:22 lib
drwxrwxr-x  2 root 4,0K feb.  28 13:22 META-INF
drwxrwxr-x 31 root 4,0K feb.  28 13:22 res
-rw-rw-rw-  1 root 591K des.  31  1979 resources.arsc
```

– **AndroidManifest.xml:** This file describes essential information about the application, the Android operating system and the Google Play Store [3]. Every application comes with a manifest file that informs the OS about the

application's components. It includes services, activities, broadcast receivers and content providers. To access protected components, the application must state its permissions in the `AndroidManifest.xml` file.

- **assets:** This directory contains all the application assets (XML files, JavaScript files, pictures, etc.)
- **classes.dex:** The `.dex` file after compiling the bytecode of all the java source code in the application.
- **lib:** This directory holds third party libraries that are not part of the Android Software Development Kit (SDK).
- **META-INF:** This directory contains files that hold the applications metadata. This includes hashes if app resources and app certificates.
- **res:** Directory containing resources that are not part of the resources.arsc file.
- **resources.arsc:** File containing precompiled resources.

When using `unzip` to extract the files of the `.apk` file some metadata is lost in the process and you end up with a non-human-readable `AndroidManifest.xml` file. By using the tool jadx, see Sect. 4.1, we create a decoded XML file that can be opened with a text editor.

2.2.4 Local Data Storage

Protecting user credentials, authentication tokens and other sensitive information is essential for mobile security. The guidelines for saving data can be summarized quite easily. Public data should be available to all processes while personal data should be protected, or not reside on the phone at all. So conventional wisdom suggest that as little sensitive data should be stored in permanent local storage. However, an application often needs to store some type of user data. For example, having users enter a long complex password for every time the app starts is unpractical. Therefore, some applications cache an authentication token to avoid this.

The Android platform provides multiple alternatives for persistent data storage, each with its own advantages and drawbacks [5]. Below is a list of the most common places where app data resides on the phone.

Internal Storage:
An application can save files to the device's internal storage. By default, these files are only accessible to the application, this is enforced through the Linux UID. When the application in uninstalled, these files are removed. The directory in which the file can be stored is `/data/data/<application-name>`.

External Storage:
All Android devices support shared external storage. The word external is often

misleading because it implies that the storage could be removed, but that is not always the case. External storage is often just a larger internal storage. However, it is not guaranteed to be accessible because it can sometimes be physically removed from the device. This storage should only be used for files that should be accessible by all apps.

Shared Preferences:
This is most commonly used to store small collections of key-value pairs. For example, authentication tokens or Global Unique Identifier (GUID)'s, see Sect. 2.2.6. The key value pairs are stored in a XML format.

SQLite Databases:
Android provides support for SQLite databases. In contrast to traditional database systems, SQLite is not a client-server database engine, it is embedded in the end program. SQLite Databases support password encryption.

Realm Databases:
An increasingly popular choice is the real database for Java. Realm is an open-source database management system [36].

2.2.5 Network Communications

Almost every network connected mobile application uses Hypertext Transport Protocol (HTTP) or Hypertext Transport Protocol over Transport Layer Security (HTTPS) to communicate with back-end services. For an application to access the internet, the `AndoridManifest.xml` file must contain the INTERNET and ACCESS_NETWORK_STATE permission. Otherwise the API call gets rejected, see Fig. 2.2. If the back-end server support HTTPS it is the preferred protocol because mobile devices frequently connect to network that are not secure, such as public WiFi hotspots.

2.2.6 Unique Identifiers

The Android platform provides unique identifiers in order to identify a device. However, the ultimate goal for developer should be to identify a particular installation of the application, not the device itself [4]. Developers should avoid using hardware identifiers like International Mobile Equipment Identity (IMEI) or the MAC address. These kinds of identifiers are classified as personal data under the GDPR and can be linked to individuals. A common best practice is that applications generate their own random GUI at installation time. By generating a random GUI the identifier cannot be linked to the device. The GUI can be used for advertising purposes as well as tracking how users interact with the application.

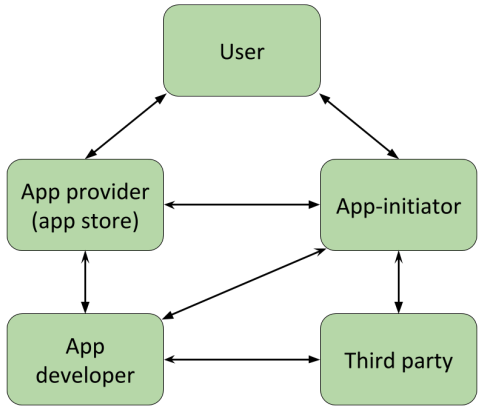

Figure 2.3: This figure shows the different stakeholders involved in an mobile application. Third parties often take the role of the processor in the context of the GDPR, processing personal data on behalf of the controller, namely the app initiator. Adapted from Datatilsynet's report on data protection challenges in mobile applications [39].

2.2.7 Application Stakeholders

Figure 2.3 shows different stakeholders in the Android ecosystem and how they relate to one another. There are two roles that imply special data protection responsibility, the data controller and data processor. Normally, the app initiator will take the role of the data controller. Often, the app initiator is the same as the app developer. Furthermore, it is likely that many third-party services qualify as data processor and will have the same regulations to abide by as the controller. These data processors process data on behalf of the controller. Processors include location tracking, add services, application tracking, etc.

2.3 Mobile App Security Testing

This part will introduce the principles behind mobile application security testing and introduce key terminology. It covers topic like black-box testing and static and dynamic analysis.

2.3.1 White-box, Black-box and Grey-box

Nidhra and Dondeti (2012) describes the following concepts [29]:

– **Black box testing:** It is testing based on the requirements specifications and there is no need to examine the code in black box testing. This is purely

done based on customers view point only tester knows the set of inputs and predictable outputs [29].

- **White box testing:** Tester designs test cases that test the internal functioning of the software from the developer's perspective, white box testing mainly focus on internal logic and structure of the code [29].
- **Grey box testing:** Everything in between the two categories mentioned above.

The type of testing that will be conducted during this thesis is similar to grey-box testing. We start with zero knowledge of the system, like black-box testing. But since we decompile the `.apk` file, we obtain knowledge of the applications internal structure, similar to white-box testing. However, we are not given any information about the application other than the decompiled source code. Section 3.8.1 describes how to decompile a `.apk` file. For Android, black-box testing is almost equivalent to white-box testing because most apps can be decompiled quite quickly [28].

2.3.2 Static Analysis

Static analysis is the analysis of computer programs without executing said program [14]. During a vulnerability analysis, a static analysis is used to ensure appropriated implementations of security controls. The goal of a static analysis must not strictly be security related, it could be used as a way to improve the overall quality of increasingly sophisticated computer programs. In this study we use two forms of static analysis techniques. First we used a manual code review, and second we used automated static analysis tools.

Manual Code Review:
Manual code review is the process of manually reading programming code. Manual code review can be slow, tedious and time-consuming, especially if there is a large code base and many dependencies.

Automatic Code Analysis:
These tools check the source code for predefined set of rules and industry standards. All the tools used in this study will be open-source. One of the limitations of automatic code analysis is that they lack app context. For example, it might struggle in finding hard coded cryptographic keys because it does not know how to identify them. The tool might also flag a potential issue that is irrelevant, because it lacks precision.

2.3.3 Dynamic Analysis

Dynamic analysis operates by executing a program and observing its behavior [14]. It is commonly used to check for common attacks, such as disclosure of data in transit

and server configuration issues. This study uses dynamic analysis to test and execute all features of an application. We observe what data is sent to back-end services and what data is stored locally on the device.

2.4 Related Work

Many tools and techniques have been designed to identify privacy leaks in Android applications. Enck et al. (2014) developed a dynamic taint tracker and analysis system and used it on 30 applications. The study showed that 20 applications misused users personal data [12]. Arzt et al. (2014) introduced another taint analysis tools, which found several privacy leaks in the top 500 applications in the Google Play Store [11]. Felt et al. (2011) developed an automatic tool that tested overprivileged Android applications [22]. They applied it to 940 applications and found that about one-third are overprivileged. Phone identifier misuse is a common potential privacy violation in mobile applications. Enck et al. (2011) performed a horizontal study of 1100 popular free Android applications. The study revealed a pervasive misuse of personal/phone identifiers, and a deep penetration of advertising and analytics networks [13]. Achara et al. (2016) analyzed 140 popular free applications for Android and iOS. The study concluded with both platform leaking personal data and misuse of identifiers [1]. Transmitting sensitive data to third-party services pose a substantial privacy violation to the user. Pultier et al. (2016) published a report on behalf of SINTEF that showed that 21 mobile applications, many of which are Norwegian, communicated with approximately 600 different primary and third-party domains [31]. In addition, central European organizations have published reports on privacy in mobile applications. These publications have played a central role in this study. The European Agency for Network and Information Security (ENISA) published a report on privacy and protection in mobile applications. The study covered the application ecosystem and the implementation of the GDPR [20]. The European Data Protection Supervisor (EDPS) have published guidelines on the protection of personal data processed by mobile applications [15].

Chapter 3
Methodology

In this chapter we explain the research methods used and rationalize their choice. Further, we introduce any limitations surrounding our study and present the test cases we evaluate each application against.

3.1 Literary Review

The main goal of the literary review was to gain a deeper understanding on how mobile applications process user's personal data. Furthermore, to learn how to perform security related tests on Android applications. In Section 1.2 we saw that the goal for the study was to identify how mobile applications process personal data with respect to the GDPR. We found that using the Android developer webpages gave us the most up-to-date information on privacy and security in applications. In addition, OWASP published a Mobile Security Testing guide, covering both Android and iOS applications [28]. Many of the test cases described later in this chapter are from this guide. We also needed an understanding of the legal requirements stipulated by the GDPR. It is a complex document written by lawmakers and full of terms and phrases unfamiliar to anyone outside the legal profession.

3.2 Design Science Research

The second part of the research will follow a design science methodology with the goal of answering the research question. "Design science creates and evaluates IT artifact intended to solve identified organizational problems" [26, p. 77]. It involves a rigorous process to design artifacts to solve observed problems, to make research contributions, to evaluate the designs, and to communicate the results to appropriate audiences [26]. Hevner et al. (2004) describes practical rules for conducting design science research in the form of seven guidelines that describes the characteristics of well conducted research [26]. The seven guidelines are:

1. Design an artifact.

2. Problem relevance.
3. Design evaluation.
4. Research contributions.
5. Research rigor.
6. Design as a search process.
7. Communication of research.

Many of these guidelines are not directly relatable to our study. For example, guideline one says that design science research must produce a viable artifact in the form of a construct, a model, a method, or an installation [26]. Mobile applications fall well within the definition of artifacts, a purposeful IT artifact that addresses an important organizational problem [26]. Our study did not create or design any new artifacts, but evaluate existing ones.

The second guideline is about the relevance of the technology-based solution to solve important business problems. The GDPR is very much a business problem. It is a problem that affects a high number of European businesses. The financial repercussions of not being compliant could be devastating, should a supervisory authority choose to impose them. In addition to the financial reproductions, the data controllers must be able to show compliance with the GDPR.

Guideline three is the one most relevant to this study. It revolves around evaluating artifacts via well executed evaluation methods. The underlying metricizes for all testing conducted in this study was to be GDPR compliant. In order to evaluate this, we gathered necessary data from the applications. This will be explained later in Sect. 3.8.1. Our methodology was analytical, we used static and dynamic analysis to examine the complexity of the mobile applications. The evaluation process was not static, but an iterative and incremental process. Since no two apps examined in this study were the same, it was difficult to predefine test cases that could be applied to all applications. Since the applications are so different from one another, some deviations from the test cases must be expected.

3.3 Evaluation of Research Method

The project description for this study states that the student shall analyze applications software code, investigate their structure, extract information, etc. This is a highly technical approach to answer our research question. Another approach could have been to follow a qualitative research methodology using *in-depth* interviews. By interviewing Android developers, we could collect data on individuals personal experience with the implementation of the GDPR in mobile applications. In fact, in the preliminary work for this study our research question contained parts that

would be most conveniently answered by interviewing developers. However, we choose not to pursue this. Mainly because we wanted a more technical approach, answering the question of *how* rather than *what*. How specifically have the GDPR been implemented, and not what measures developers said to have done. There is also the process of choosing interview subjects. Should they be general app developers or developers of specific apps? If the latter, what incentive do they have to admit what is noncompliance with the GDPR. Another more practical barrier to overcome would have been to come in contact with developers willing to participate. Since this thesis is written independently and not in cooperation with a data controller. Nonetheless, interview subject bias contributed to the decision of favouring a design science methodology over a qualitative research one. Although it was not the deciding factor.

3.4 Limitations

As mentioned in Sect. 2.3.2, the manual code review played a big part in this study. Many apps used multiple third-party libraries resulting in a large code base. As a result, it became difficult to pinpoint what functionality the application actually used from these libraries. In many cases there was hundreds upon hundreds of lines of dead-code. This study was further limited by the technical proficiency and understanding of the researchers. We had limited knowledge on the topic of privacy and security in Android platform prior to this study, not to mention how to interpret the GDPR as a legal document.

Another limitation is the number of apps included in this study. This have been a rather small study compared to previously conducted studies that run automated tools upon hundreds of applications, see Sect. 2.4. Therefore, it can then be difficult to generalize our finding and draw conclusions that apply to other applications. Our study is specific, it examined popular pharmaceutical and dating applications in Norway. Therefor our findings are limited to this. Nonetheless, we hope that our findings can help other app developers to better understand the legal requirements of the GDPR and how it impacts personal data processing in Android applications.

The majority of work conducted in this study was before the enforcement date of the GDPR. Therefore, there are some functionalities of the regulation that is difficult to test. Because they rely on the data controller abiding to the regulation before the implementation date. These functionalitites include the right to erasure ("right to be forgotten"), right to data portability and the right to rectification.

3.5 Choosing Applications

During the first stages of the thesis, we had not yet decided which applications to examine, or moreover, which category of applications to choose from. There are several factors that led to the decision to focus on medical and pharmaceutical applications. In the preliminary work of this study, a variety of applications were briefly examined to figure out what types of data the different applications processed. Our first assumption was to target applications that offer some sort of members club to its users. The thought being that these applications would collect a great deal of personal data and some form of authentication mechanism was used. However, it turned out that many of these app have most of their logic server side, and the application only functions as a WebView. The result is an application that stores little to no information on the device.

One of the deciding factors in the decision to focus on pharmaceutical and dating applications is Article 9 of the GDPR, on the processing of special categories of data. To summarize, it states that the processing of certain categories of data like personal data reveling an individual's race or ethnic origin, political opinion, religious or philosophical beliefs, data concerning health or a natural person's sex life or sexual orientation shall be prohibited. The first paragraph of the article strictly forbids the handling of such data while the second paragraph states the circumstances in which processing of this data is acceptable. Letter (a) states that such data can be processed if the data subject has given explicit consent to the processing of that special category of personal data for one or more specific purposes. What this means in practice is that data controllers must provide a lawful purpose for such processing and must be able to demonstrate that the data subject has provided explicit consent for that intended purpose. Letter (e) states that processing is legal if the personal data have been made manifestly public by the data subject. However, it does not specify the rules in which such processing is allowed. It does not exempt it from the underlying principles described in Sect. 2.1.4.

Many of the most popular applications in the Google Play Store, regardless of category have a bounty program. This is a reward program for discovering and reporting security vulnerabilities in software. An example being Facebook's whitehat program [21]. In addition, applications that have an enormous user base frequently update their applications and have many developers working on these updates. Because of this, this study will focus on the much smaller Norwegian app market. However, this does not imply that applications on the Norwegian app market have more security vulnerabilities, just that we want to increase our probability of finding non-compliant behavior.

3.6 Device Setup

For the dynamic analysis a real Android device is needed to run applications on. In theory, it is possible to run any application on an emulator. However, applications execute quite slowly which can be tedious when security testing. Therefore, testing on a real device gives a much more streamlined process. The device we used is a Nexus 5X running Android Marshmallow (6.0.0) with build number is MDA89E.

Rooting is the act of modifying the OS to obtain superuser privileges. By default, the root user has access to all commands and files on Unix like operating system. On Android, this allows you to bypass restrictions such as android sandboxing. Virtually any Android device can be rooted, but the most common devices are the ones running stock Android. Phones designed by Google and manufactured by companies like Samsung, LG and Motorola. The Google Nexus and Pixel series are among the most popular, particularly because many developers use them. Google also offers binary image files that allows you to flash your device back to the original factory firmware [1]. This is useful if the device contains flashed custom builds or if something went wrong during the rooting process.

To root the nexus 5X we first needed a working adb/fastboot environment. The first step was to unlock the bootloader. The next step was to install a custom recovery. This allowed us to install custom software on the device. We used twrp version 2.8.7.2. The next step was to download SuperSu and store in internal storage. SuperSu is a popular superuser access management tool available for download from the Google Play Store, see Sect. 4.3. The last step was to boot into recovery mode, navigate to SuperSu and swipe to install. This added a binary file named *su* to the `/su/bin/` directory.

3.7 Network Setup

Nearly every network-based mobile application uses the HTTP or HTTPS to communicate with remote endpoints. This makes applications vulnerable to common network-based attacks, like packet sniffing and man-in-the-middle-attacks. In order to intercept traffic generated by the application, we set up an intercepting proxy on the host machine. Figure 3.1 shows how we set up our environment to observe network traffic being exchanged between the mobile device and remote endpoints. In addition, the mobile device must be configured to route HTTP(S) traffic through the proxy, see Fig. 3.2. This is because the proxy server breaks the Secure Socket Layer (SSL) certificate validation. As a result, the application will usually fail to establish a secure Transport Layer Security (TLS) connection. To work around this, the proxy

[1] https://developers.google.com/android/images

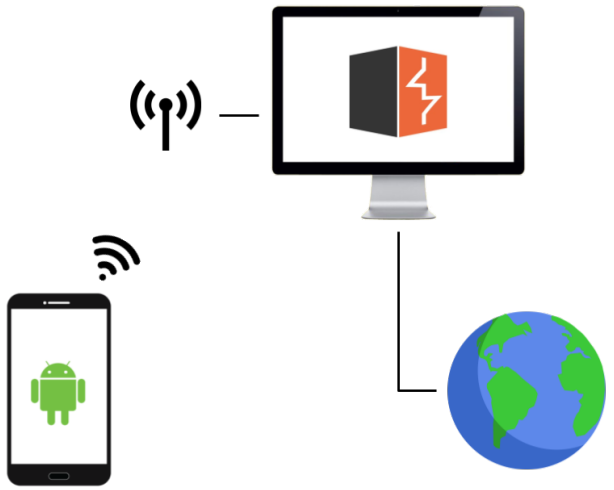

Figure 3.1: Figure shows our network setup in order to intercept traffic generated by applications. We use a wireless USB adapter to create a Wii hotspot on the host machine. The host machine is connected to the Internet via an Ethernet cable. We then connect our Android device to the WiFi hotspot. The host machine run the intercepting proxy that traffic is routed through.

servers CA certificate must be installed as a trusted certificate on the device. We used BurpSuite Community Edition to set up our proxy, see Sect. 4.4.

3.8 Testing Process

The following section describes the steps undergone for each application. We start by downloading the application from the Google Play Store using the Nexus 5X.

3.8.1 Obtaining Source Code

The title of this section is misleading, since we do not obtain a bit-by-bit copy of the original source code, but an approximation. The first step was to obtain a copy of the `.apk` file. There are multiple ways to achieve this, but since we had obtained root access on the phone, we copied the `base.apk` file from `/data/app/<name_of_app>/` directory to external storage. From there we simply pulled the `base.apk` file using `adb`.

```
$ adb pull /sdcard/files/base.apk
```

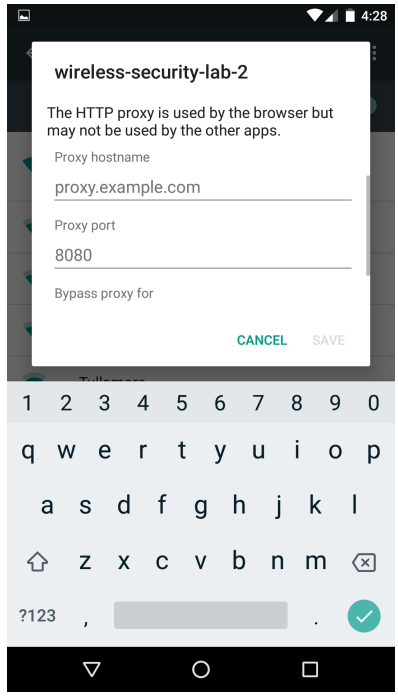

Figure 3.2: Screenshot showing how to connect to the intercepting proxy under WiFi settings on the device. Proxy hostname is the IP address of the machine running the proxy.

The next step was to run the application through a decompiler. We used `jadx` which is a dex to java decompiler. We then opened the source code in a GUI to examine it.

```
$ jadx <name-of-application.apk>
```

In addition, we used `dex2jar` to obtain a `.jar` file from the `.apk` file. A `.jar` file is an Zip archive containing one or multiple java class files. We did this because we wanted to use another GUI, JD-Gui because we found it to have a more reliable search function than IntelliJ.

```
$ sh d2j-dex2jar.sh -f <name-of-application.apk>
```

3.8.2 Using Automated Tools

Once we have extracted the `.apk` file from the device we run it through automatic static analysis tools. These are tools that take a `.apk` file as input, decompiles it and scans it for common security vulnerabilities. They are not 100% reliable and can produce a high number of false positives. Nonetheless, they are useful for providing

an initial starting point for further analysis. In this study we used SUPER and Qark [34, 37].

3.8.3 Testing Data Storage

We performed static analysis on the obtained source code, starting with the `Andndroid.Manifest.xml` file. Here we looked for permissions that allow access to external storage, in particular the two permissions WRITE_EXTERNAL_STORAGE and READ_EXTERNAL_STORAGE. We then performed a manual code review. We searched for class names that could indicate that the class handles personal data, classes named `ProfileActivity.java`, `SignUpActivity.java`, `OrderActivity.java`, etc. These classes are of interest because they provide information on how the application handles user input. In addition, we also performed a search for keywords and common Application Programming Interface (API)'s, classes like the SharedPreferences class, FileOutPutStream (which is used by both internal and external storage), getReadableDatabase and getWritableDatabase. After the static analysis we ran through all the functionalities of the application. For example, we created a user account, sent requests for services (subsequently dropped the request in our BurpProxy), we used location services, logged out, logged back in, etc. This was to stimulate normal use of the application. Afterwards, we extracted the `data/data/<name-of-application>` directory and examined the files. This was performed multiple times to make sure the application clears sensitive data upon session termination and logout.

3.8.4 Testing Secure Communications

As mentioned in Sect. 3.7, we set up a intercepting proxy using Burp Suit Community Edition 4.4. Since we are not concern with any vulnerabilities or leaking of personal data on the server side, we will focus on what information is exchanged between the two endpoints and how the application performs certificate validation. In regard to certificate validation, there are three cases we want to test for:

1. Accepting self-signed certificates.
2. Accepting invalid certificates.
3. Accepting wrong host name.

Figure 3.3 shows the certificate options for the intercepting proxy in Burp Suite. By changing this option and checking if we observe any traffic picked up by our proxy we can determine if the application accepts certain certificates. By choosing "Generate a CA signed certificate with a specific hostname," and typing in example.org we can check if the application accepts an invalid hostname.

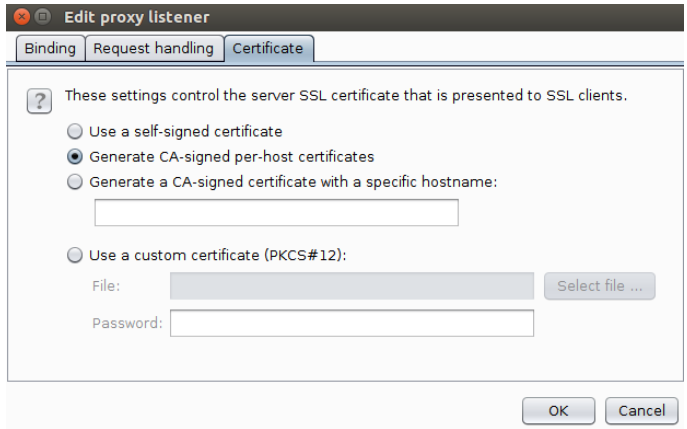

Figure 3.3: Screenshot from Burp Suite Community Edition. Different settings that control the server SSL certificate that is presented to SSL clients.

In addition to performing certificate validation, we observe what information is exchanged over the network. Any request to a domain that does not directly host the main functionality of the application could be a potential breach of the privacy policy. We looked at all the data being exfiltrated from the phone. We searched for personal data, geolocation, device identifiers, etc. Any information relating to an individual or identifiable natural person.

3.8.5 Testing for Lawfulness, Transparency and Consent

This test case evolves around how the applications manages its communication and consent with the data subject. There are basically four channels in which the applications can provide information to its users.

1. **Privacy policy:** Does the application have a privacy policy which should include all the information mentioned in Sect. 2.1.5. Is it easily available and written in a clear and transparent language, such that a child would be able to understand it?
2. **Link to privacy Policy:** Does the Google Play Store links to the applications privacy policy. Which is usually found under their web application domain.
3. **Permissions:** Information through the permissions the app requests will allow users to make informed decisions. Requesting permission through the standard user-based-permission system is not enough to fully comply with the requirements of the GDPR. It helps the application developers with *how*, and not the *why*.
4. **Directly in the app:** Information provided when direct user input is given

or when the user interacts with the app. This can be provided in the form of pop-ups or information icons.

Chapter 4 Tools

This chapter lists the different tools used in this study. All of them are freely available. A brief explanation is given for each tool as well as a justification for its use. Common Linux command-line tools are omitted from this list, only specialized tools important to this study are included.

4.1 Jadx

Jadx is a dex to java decompiler [25]. It is a command line and GUI tool for reverse engineering Java source code from Android `.dex` and `.apk` files. We found jadx to be a easy-to-use decompiler because it generates Java source code in the form of `.java` classes and a human readable `AndroidManifest.xml` file. This tools was used after we extracted the applications `.apk` file from the device.

4.2 Dex2jar

Dex2jar is another decompiler [24]. As the name suggests, it converts .jar files from a `.apk` file. We used it to generate a .jar file that we opened in another GUI.

4.3 SuperSu

This is an Android application that resides on the device. The application is available through the Google Play Store. Its purpose is to manage super user access on the device, thus it requires the device to be rooted [35].

4.4 Burp Suite Community Edition

Burp suite is a graphical tool for testing web application security [30]. This study uses the freely available version, namely community edition. This version still offer a wide variety of tools. The tool most relevant for this study is the HTTP Proxy

that works as a man-in-the-middle. This allows for the interception and inspection of raw traffic going in both directions. Burp also has a self signed certificate that can be installed as a trusted certificate on a mobile device. Thus allowing for the interception of HTTPS traffic.

4.5 Wireshark

Wireshark is a free and open-source network protocol analyzer [38]. It is used for network monitoring. In this study we used it to view how the application sets upp a secure communicatin channel with back-end services.

4.6 Android Debug Bridge

Android Debug Bridge is a versatile command-line tool that lets you communicate with a Android device [2]. It facilitates a variety device actions, like the ability to extract files, install apps, debugging programs and it provides a UNIX shell that can be used to run commands on the device. To communicate with the device it works as a client-server program that includes the following components. The client, which runs on the development machine. The daemon (adbd), which runs command on the device, It functions as a background process. And finally, the server, which manages the communication between a client and the daemon. All of the communication between the host machine and the device has been done through adb.

4.7 SUPER

SUPER stands for Secure, Unified, Powerful and Extensible Rust Android Analyzer [34]. It's a command line tool that works on multiple platform and can be used to analyze .apk files in search of vulnerabilities. It differs from other Android analyzers in that it is written in Rust, rather than Python or Java. A HTML report is generated after execution that separates vulnerabilities into critical, high, medium, low and warnings. We used this tool to get an overview of the vulnerabilities of the application.

4.8 QARK

This is another automated security vulnerability scanner for Android written in python [37]. QARK stands for Quick Android Review Kit. QARK automates the use of multiple decompilers, leveraging their combined outputs, to produce superior results, when decompiling APKs. We used the tool for the same reason we used SUPER, to get a starting point for the manual code review..

Chapter 5
Findings

The following chapter will present our findings, starting with pharmaceutical applications. Each application is structured with a brief introduction, followed by general info. Then follows our findings from the test cases. Some findings are similar to one another. Nonetheless, they are included because the findings are still relevant to that specific application.

5.1 Vitusapotek

Vitusapotek is a Norwegian pharmacy company with 250 pharmacies across Norway. The company is owned by Norsk Medisinaldepot AS. In August of 2018, the application had over 50,000 downloads in the Google Play Store. The app is intended to facilitate services that Vitusapotek offers consumers. Services include registering for their VI+ customer club, locate your nearest pharmacy, access the newsletter and prepare electronic prescriptions. The application operates under two levels of data processing, each with its own requirements for consent. The first level are individuals who download the application, but choose not to register for the VI+ member club. This mean that the application processes less information about the user. However, the application still allows for preparing e-prescriptions in which they process a person's full name, phone number and social security number. The second level is when users register for the VI+ membership club. This opens the door for automatic individual decision-making and transferring data to third parties outside the European Union.

5.1.1 General Info

- **Industry:** Pharmaceutical
- **Stakeholders:** Norsk Medisinaldepot AS (NMD) and Vitusapotek.

- **Goal:** Order e-prescriptions, locate nearest pharmacy, manage their VI+ account, read the newsletter and visit the vitusapotek.no website.
- **Privacy Policy:** The Google Play Store links to the companies privacy policy, it is the same as one would access through a desktop environment.
- **Personal Data:** Name, phone number, year of birth, email, geolocation, social security number, address, medicine type.
- **Local Storage:**SQLite databases, Realm Databases, Shared Preferences.
- **Network:** The app communicates with back end services over HTTPS.
- **Third Party Services:** McKesson Corporation, Google analytics and AdForm and

5.1.2 Findings

Finding 1: *Storing users' credentials in an unencrypted Realm database.* When registering for the VI+ membership club the application requests the user's full name, phone number and address. In addition, when the user agrees to the terms and conditions, there is an option to register your gender, year of birth and if you are responsible for children. Naturally, this information is transferred to Vitusapoteks server. However, it is also stored in clear in a realm database. The file `files/vitus_apotek.realm` holds static information used by the application. For example, it holds the addresses of all Vitus pharmacies in Norway, their geopoints, opening hours etc. In addition, it holds a class called user. Table 5.1 shows the class object that gets stored locally. The attribute pharmacies store the pharmacies that the users have marked as their favorite. In addition to the realm file, the applications logs changes to the database in a log file, that contains the same user information as the realm file, see Source Code 5.1.

Finding 2: *Storing e-prescriptions in an unencrypted Realm database.* The user has the ability to register an e-prescription. It is possible to do so without being a member of the VI+ club. The app requests the user's full name, phone number and social security number. When the user prepares the e-prescription, there is a small question mark in the top right corner of the screen. By pressing it, the user is prompted with information on how this data is processed, see Fig. 5.1. The information is contradictory, the first paragraph states that in order to speed up the processing of e-prescription, the information will be stored locally on the phone. However, the second paragraph states that Vitusapotek does not store any information that the user provides to them through the application, and that they only use the data to find your prescription at the pharmacy. The order gets stored in the realm database, see Tab. 5.2. The pharmacy value/attribute represents the same is in Table 5.1. As you can see, the app stores the users social-security number in clear, here with the value 12345678910.

Table 5.1: A modified version of the realm database object stored in local storage. Some values are abbreviated to increase readability.

Attribute	Type	Value
uid	string?	ttm4905.2018@gmail.com
type	string?	userWsDTO
firstName	string?	Wireless
lastName	string?	Lab
mobileNumber	string?	99999999
interests	interests[]	[list of interests]
consentForNewsletter	bool	false
membership	bool	true
pharmacies	Pharmacy[]	[list of Pharmacy: 153, 1381]
gender	Gender?	Gender code = MALE
memberClubs	MemberClub[]	[list of MemberClub: family]
responsibleForKids	bool	false
birthYear	string?	1995
notifications	Notifications[]	[list of Notifications: fPEI(...)]
address	Address?	Address = id = 8835300687895

Table 5.2: The LastOrder object stored in the realm database. The value 12345678910 represents the user's social security number.

Attribute	Type	Value
personId	string?	12345678910
firstName	string?	Wireless Security
lastName	string?	Lab
mobileNumber	string?	99999999
pharmacy	string?	1553

Finding 3: *Collecting responsible for kids.* We found that the application has the option to register if the user is responsible for children. Freely given information is still subject to principles of processing described in Sect. 2.1.4. For the app to collect this type of data the user must already be a member of the VI+ Club. The user is not required to give the information, but the application makes it rather easy. Nowhere in the application's privacy policy does it state that it collects this type of information, or its reason for doing so.

Finding 4: *Information provided upon data collection.* The application's privacy policy is only accessible in the application if the user is logged into their VI+ profile.

Source code 5.1 Showing the personal data stored in the log_a file.

```
$ strings /files/vitus_apotek.realm.management/log_a
userWsDTO
Wireless
Lab
99999999
Mann
Velv
1995"
android
%O.S. Bragstads Plass, TRONDHEIM, 7034
O.S. Bragstads Plass
7034
Norge
         TRONDHEIM
```

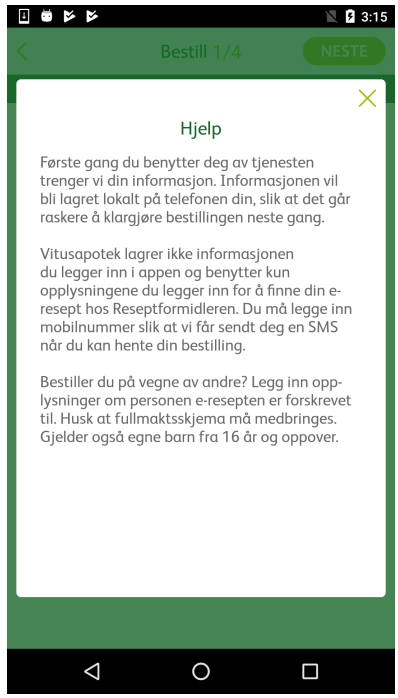

Figure 5.1: Screenshot from Vitusapotek mobile application. When the user prepares an e-prescription, the application displays information about how the data is processed.

The privacy policy fails to inform the user of the right to lodge a complaint with the supervisory authority.

Finding 5: *Right to object to automatic individual decision-making.* We found that the application does not provide data subjects with the appropriate option regarding the right to object to automated individual decision-making for direct marketing purposes. The application only provides two option in regards to consent, see Fig. 5.2. First, if the data subject agrees to the terms and conditions. Second, if the user wants to receive relevant information by SMS and email. The membership agreement states the data subject's personal information is used for profiling, with the goal of providing a more complete experience. By agreeing to the membership agreement, the user agrees to automatic individual decision-making for direct marketing. The app fails in providing an option that agrees to the term and conditions, but reject automatic decision-making and still receive general offers from Vitusapotek.

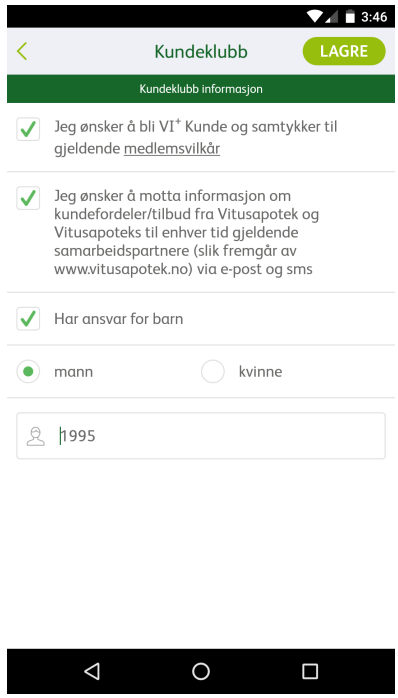

Figure 5.2: Screenshot from Vitusapotek mobile application. The user is not presented with a separate option to opt-out of individual decision-making. That right should be explicitly brought to the attention of the user.

5.1.3 Summary

Vitusapoteks Android application fails to comply with many of the requirements of the GDPR. The application was not designed with focus on the confidentiality of personal data. This is most easily visible in the writing of a user's social security number in clear to a realm database, especially when encrypted realm databases are available. Their privacy policy seemingly only applies to VI+ members, and is not available in the app if the user is not logged in. Lastly, it does not provide users with the option to opt-out of automatic individual decision-making and still receive general marketing info from Vitsuapotek.

5.2 Ditt Apotek

Ditt Apotek is owned by the same parent company as Vitusapotek, namely Norsk Medisinaldepot AS. It therefore shares a lot of the code base with Vitusapotek. The application have less functionality than Vitusapotek. As mentioned in the introduction to Vitusapotek, the Vitusapotek application operates under two levels of data processing. Ditt Apotek operates under the first level described earlier. The application has no option to create a user, or to log in as an existing user. The functionalities of the app includes locating the nearest pharmacy, prepare e-prescriptions, visit the online store and accessing the customer newspaper.

5.2.1 General Info

- **Industry:** Pharmaceutical
- **Stakeholders:** Ditt Apotek, Norsk medisinaldepot AS.
- **Goal:** Locate pharmacies, prepare e-prescriptions, read newsletter and visit online store.
- **Privacy Policy:** Google Play Store links to dittapotek.no that does not have a privacy policy. When preparing an e-prescription the data subject must agree to the terms and conditions.
- **Personal Data:** Location data, name, phone number, social security number, medicine.
- **Local Storage:** Realm database, SQLite database.
- **Network:** The application communicates with back-end services over HTTPS.
- **Third Party Services:** BoostCom AS, Google Analytics, AdForm.

5.2.2 Findings

Finding 6: *Storing e-prescriptions in an unencrypted Realm database.* This is in very much related to finding 2 in Vitusapotek. Ditt Apotek stores the user's last order in clear, exposing the data subject's first name, last name, phone number and social security number. It is stored as a class named LastOrder in a Realm Database. In addition, the information gets written in a log file in the `com.dittapotek.eresept/files/ditt_apotek.realm.management` directory. Source Code 1 shows how it is possible to use command line tools to find a person social security number in the realm file.

Source code 5.2 Using the command-line utility strings and egrep to find strings of length 11 in the ditt_apotek.realm file. Norwegian social security number are 8 digits long.

```
$strings ditt_apotek.realm | egrep ^[0-9]{11}$
12345678910
12345678910
```

Finding 7: *Absence of privacy policy.* Ditt Apotek does not have a clear privacy policy, neither in the application or on their website. There is a terms and conditions when using the e-prescription function. Before sending the request to Ditt Apotek's back-end-service, the user is prompted with a pre-ticked box agreeing to the terms and conditions. The terms and conditions does mention duration of storage and the purpose of processing, which is sufficient for the handling of e-perceptions. However, the application provides no information about how cookies are handled while interacting with the online store. Further, it provides no information on how the application uses location services. After a Google search for "ditt apotek cookies" we are redirected to a web page that explains how cookies are handled and used in an anonymous way. The web page further goes on to explain how to disable cookies in the most popular desktop web browser, there is no instructions on how to prevent cookies in the mobile application.

Finding 8: *Lack of affirmative consent.* When preparing the e-prescription, there is a pre-ticked box presented to the user. The box is meant to collect consent from the user that they agrees to the terms and services. This is not a freely given affirmative act, see Fig. 5.3.

5.2.3 Summary

As mentioned in the introduction, Ditt Apotek shares almost all of its code base with Vistuapotek, although with less functionality. It seems that the application

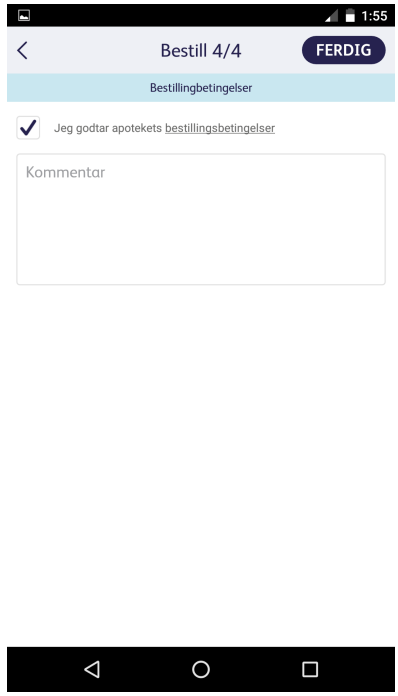

Figure 5.3: Screenshot from the Ditt Apotek mobile application. Picture is from the last action in ordering an e-prescription. The user is prompted with a pre-ticked box when agreeing to the terms and conditions.

was not designed with a clear privacy policy. The use of pre-ticked boxes does not establish a clear affirmative act.

5.3 Apotek 1

Apotek 1 is a pharmacy chain owned by Apotek 1 Gruppen AS. It roughly has a 40% market share in Norway. In April of 2018 the application had 100,000+ downloads in the Google Play Store. The description of the application in the Play Store is scarce, but the application lets users locate pharmacies, read the newsletter, visit their website and receive special membership deals. Ordering electronic prescriptions are performed through a WebView and users are authenticated using BankID.

5.3.1 General Info

- **Industry:** Pharmacutical
- **Stakeholders:** Apotek 1 Gruppen AS

- **Goal:** Ordering electronic prescriptions, locating nearest pharmacy, accessing the newsletter.
- **Privacy Policy:** Available in the app, no reference from Google Play Store.
- **Personal Data:** Users name, date of birth, gender, email, postal code and location. In addition to general interests from a precreated list. Options include, babies, skin care, etc.
- **Local Storage:** Shared Preferences, files, SQLite databases.
- **Network:** The application communicates with back-end services over HTTPS.
- **Third Party Services:** BoostCom AS, Nordic family group, Mailchimp, Solteq, Google Analytics, Facebook and AdForm.

5.3.2 Findings

Finding 9: *Obfuscated source code.* Apotek 1 has obfuscated source code, see Fig. 5.4. This complicates the static analysis. Searching for method names and class names become difficult because classes are assigned one letter names. For example, one class can import another class with the line `import d.a.a.a.a.c.m`. Most likely the application has been "minified" with ProGuard [28].

Figure 5.4: This figure shows the file hierarchy of the Apotek 1 application. Regular class and folder names has been replaced with single letters.

Finding 10: *Allow backup enabled.* The Application has set the `android:allowBackup="true"`. This is not strictly a vulnerability but it can lead to personal data being extracted if the application stores any personal data in internal storage. If `allowBackup` is set to false, in order to extract a copy of the internal data, the user will have to obtain superuser privileges on the device. However, in its current state, anyone can make a copy of the `data/data/com.ompalabs.stores.apotek1` directory.

Finding 11: *Right to object to automatic individual decision-making.* The application makes it possible to opt-out of any automated individual decision-making, see Figure 5.5. It does so by making sure the user declines or accepts two questions. The third question in Fig. 5.5 ask if the data subject allows Apotek 1 to provide the user with customized discounts and offers. By pressing the little *i* icon next to the question, a pop-up appears. It states that Apotek 1 can use the personal data they process on the user, in addition to the user's order-history and campaign-response as a basis for providing discounts. The fourth question in Fig. 5.5 ask the data subject accepts that Apotek 1 provides offers to the data subject through other media.

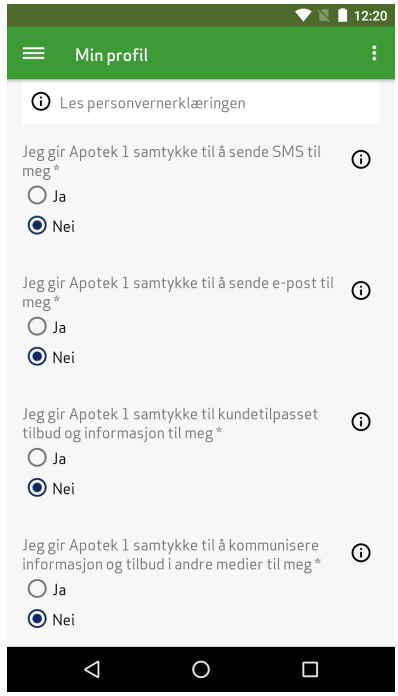

Figure 5.5: Screenshot from the Apotek 1 mobile application. Picture taken from the My Profile Activity. A clearly visible opportunity to opt-out of automatic individual decision-making for direct marketing purposes.

Finding 12: *Information provided upon data collection.* We found a difference between the privacy policy presented to the user in the Android application and the one listed on apotek1.no. The one in the application fails to inform the user of the right to lodge a complaint with the supervisory authority. In addition, there is no information about the right to data portability, only to receive a copy of the data they process.

5.3.3 Summary

Apotek 1 has been developed with the rights of the data subject in mind. It does not store any user data locally and it protects its code base by obfuscating it. Further, the application makes it easy to opt-out of automatic decision in direct-marketing purposes. It even distinguishes between if the user allows for marketing through other media.

5.4 Møteplassen

Møteplassen is an online dating platform owned by Schibsted Media Group. The application had 10,000+ downloads in the Google Play Store in April 2018. The application is free, however some activities are restricted for paying users. It processes the user's name, email, age, etc. In addition, it processes sensitive categories of data like political opinion, religion, sexual preference, etc.

5.4.1 General Info

- **Industry:** Online dating
- **Stakeholders:** Möteplanssen i Norden AB, Schibsted Media Group.
- **Goal:** Register and update users profiles. Search for other users of the app.
- **Privacy Policy:** The Google Play Store links to the companies terms and condition, which again link to their privacy policy.
- **Personal Data:** Name, age, gender, location, likes, appearance, profession, religion, income, numebr of children.
- **Local Storage:** Shared preferences, local files, SQLite databases.
- **Network:** The application communicates with back-end services over HTTPS.
- **Third Party Services:** Other companies in the Schibsted Media Group.

Finding 13: *No granularity in automatic individual decision-making.* When registering an account the user is prompted for consent regarding the terms of service and the privacy policy, see Fig. 5.6. Automatic individual decision-making is an important part of any dating service. The application fails to distinguish between agreeing to let Møteplassen use the personal data for the purpose of finding matches, and for the purpose of direct marketing. Everything is agreed upon with the same consent. There is no option to let Moteplassen use personal data for matchmaking, but opt-out of for direct marketing.

Finding 14: *Storing personal data in shared preferences.* The applications stored the user's age, username, gender and email locally on the device. In Source Code 5.3 it is possible to read the data subjects, age, email, gender (assuming 1=male) and

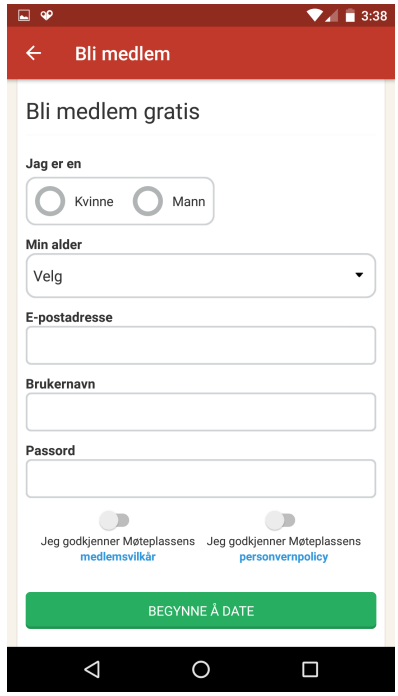

Figure 5.6: Screenshot from the Møteplassen application. The user is not presented with a clear option to opt-out of automatic individual decision-making for direct-marketing purposes.

username. It is stored in the file `data/data/com.moteplassen.app/shared_prefs`. The privacy policy states that the application stores tokens in internal storage, not personal data.

Finding 15: *Possible transfers to third countries.* The privacy policy first states that by agreeing to it, personal data will be transferred to a third party outside the European Union. It does not specify which countries, only that appropriate safeguards will be in place. Later in the policy it is written that the company *may* transfer data to a country outside the EU. This could lead to confusion for the data subject, thus not complying with article 12.

5.4.2 Summary.

As mentioned in finding 14, automatic individual decision-making is a big part of an online dating service. But the affirmative consent the application collects, cover both the dating and marketing aspect of the application. Further, the privacy policy states that it will not use sensitive personal data for marketing purposes.

Source code 5.3 The XML file stored locally. The file holds the user's age, email, gender and username. Original file have been abbreviated for readability.

```
  <?xml version='1.0' encoding='utf-8' standalone='yes' ?>
<map>
    <string name="userloginname">Wireless-Security</string>
    <int name="age" value="25" />
    <string name="token">eyJ0eXAiOiJKV1QiLC(...)</string>
    <int name="userId" value="4802964" />
    <string name="authorize">7a6d4f77af9e4fc(...)</string>
    <string name="email">ttm4905.2018@gmail.com</string>
    <int name="vipstatus" value="0" />
    <string name="profilepicturefilename"></string>
    <string name="userName">Wireless-Securityn</string>
    <string name="registrationId">fp0tuq5WH(...)</string>
    <int name="sex" value="1" />
</map>
```

5.5 Sukker.no

Sukker.no advertises itself as Norway's biggest online dating site. It has 50,000 + downloads in the Google Play Store and it has 830 000 registered members as of 2018 [33]. The application allows users to create profiles and communicate with each other free of charge. The application is built using Apache Cordova, which is platform that allows applications to be written with web application technology like HTML, CSS and JavaScript. In addition, the application processes sensitive categories of data.

5.5.1 General Info

- **Industry:** Online dating
- **Stakeholders:** Warm Systems AS
- **Goal:** Create profiles and interact with other users.
- **Privacy Policy:** Available in the application and the Google Play Store.
- **Personal Data:** Email, username, age, gender, appearance, location, interests.
- **Local Storage:** Shared Preferences, SQLite databases, files.
- **Network:** The application communicates with back-end services over HTTPS.
- **Third Party Services:** -

Finding 16: *Lack of affirmative consent.* When registering the user's credentials, the application does not collect consent through a clear affirmative act. The application

presents the user with a pre-ticked box, see Fig. 5.7. Sukker.no does not use the personal data for marketing purposes. When the user is agreeing to the terms and conditions, it is only for the apps intended purpose, namely matchmaking.

Figure 5.7: Screenshot from the sukker.no application. When creating an account, the user is presented with a pre-ticked box to accept the terms and conditions.

Finding 17: *Logging changes to user's account in clear.* The application collects personal data as well as sensitive personal data, data like religion and political view. The privacy policy states that sensitive personal data is not used in the matchmaking process of the application, but can be viewed by other users visiting your profile. Further, it states that the information used for the matching process is only the data provided upon registration. Figure 5.7 shows the minimum amount of information needed to create a profile. Both the personal data used for the matchmaking process and the data provided voluntarily are stored locally on the device. Neither the terms and conditions or the privacy policy informs the user of this. Source Code 5.4 shows what sensitive personal data is stored.

5.5.2 Summary

The application fails to inform users that their profile is stored locally and unencrypted on the device. In addition, users do not provide a clear affirmative act when agreeing

Source code 5.4 This shows a simplified version of the JSON object stored locally. The two JSON objects "politic" and "belief" are nested JSON objects in the object with "id":837588. The object with "id":837588 holds additional information about the user, but have been abbreviated for complexity reasons.

```
{
"id":837588,
"nick":"Wireless",
"member":2,
"publicCode":"1HtVfx",
"age":23,
"gender":"male",
"height":180,
(...)
}

{                               {
"id":"politic",                 "id":"belief",
"label":"Politikk",             "label":"Religion",
"value":"Liberal",              "value":"Agnostisk",
"valueId":25,                   "valueId":120,
"importance":"Veldig",          "importance":"Veldig",
"importanceId":40               "importanceId":40
}                               }
```

to the terms and conditions.

Chapter 6
Discussion

Our study has shown writing of personal data to local storage, incomplete information about the rights of the data subject and insufficient granularity when opting out of automatic individual decision-making. In addition, our study revealed that multiple applications have updated their privacy policy to cohere with the GDPR, although with some shortcomings. In this section we will explain the most important findings, and how they relate to similar studies. Further, we will revisit the studies limitations.

6.1 Summary of Findings

The following section contains the most important findings of our study.

Security of processing:
The study revealed four applications were personal data was written in clear to local storage. Both Vitusapotek and Ditt Apotek wrote e-prescriptions to local storage, including the data subject social security number. In addition, Vitusapotek and Møteplassen stored the user's credentials when registering for the VI+ membership program. Sukker.no logged any changes the user did to their profile in clear, including personal data and sensitive personal data.

One of the reasons that merit storing personal data locally on the device is user functionality when the device is offline. But this should be static information and data not related to the data subject. For example, Vitusapotek and Ditt Apotek stores the address and opening hours of all their pharmacies on the device. This is justifiable because the user might have limited bandwidth and transferring all that data every time the user opens the application is unnecessary. But any information directly related to the data subject should be omitted, or if the functionality requires it, encrypted.

Out of all the personal data disclosures found in this study, the social security number can unmistakably identify a natural person. Legal names can be shared among

individuals, and usernames and emails can be made to not identify an individual. In Norway, the social security number is used as an identifier and authenticator, which makes it highly sensitive information. Finding 18 of Sukker.no also shows sensitive personal data being processed and stored locally. Although letter (e) of article 9, par. 2, states that sensitive categories of data may be processed if it is manifestly made public by the data subject. It does not identify the rules by which this is possible, and it certainly does not mean that the data can be processed indiscriminately. Further, it does not exempt it from the general principles defined in the GDPR, like purpose limitations, secure processing etc.

All our findings related to local storage was found in internal storage. None of the applications wrote any personal data to external storage. Internal storage is automatically protected by the application sandbox. In order to extract this information, a user must either compromise the security of the Linux kernel or obtain root access on the device. Rooting in itself would not erase anything, except temporary files created during the process. On the Nexus 5X used in this study, it was necessary to unlock the bootloader in order to obtain root access. Unlocking the bootloader performs a factory-reset which erases everything. However, it is possible to make a backup of the data before the rooting process, but this depends on the Android version.

Automatic individual decision making:
The right to object to automatic individual decision-making is a important part of the GDPR. Only one of the applications in this study had an option to specifically opt-out for direct marketing purposes. Apotek 1 provided the information clearly and separately from all other information. For dating applications, automatic individual decision-making is a big part of their service. However, these applications do not provide an option to restrict the processing for direct marketing purposes. Agreeing to the privacy policy and/or terms and conditions acts as consent for agreeing to automatic individual decision-making for direct marketing purposes and matchmaking.

Affirmative consent:
An ongoing trend was to present users with pre-ticked boxes when collecting consent. Recital 32 states that this does not establish a freely given affirmative unambiguous indication of the data subject's agreement to the processing of personal data. This is something that can be easily patched when updating an application. App developers need to separate between consenting to the processing of personal data and be the subject of direct marketing. Apotek 1 showed good practices in separating the affirmative consent.

Information provided upon data collection:

Only one of the applications did not have a clear privacy policy, namely Ditt Apotek. Further, only a handful had the privacy policy available in the application when the device was offline. Many of the applications had updated their policy because they contained information specific to the GDPR. However, many failed to inform the user about the right to lodge a complaint with the supervisory authority. This is a clear violation of article 13, par. 2, letter (e). All applications should have the privacy policy available in the application at all times, regardless of the device is offline.

6.2 Relation to Previous Studies

The study did not reveal any missuses of phone identifiers like previous studies on Android application security [13, 1]. A couple of mobile applications contained code that had the ability to retrieve the IMEIHowever, the functions could be dead-code, or the `AndroidManifest.xml` file did not contain the READ_PHONE_STATE permission, meaning any function call would raise an error. We were not able to see any phone identifiers being exfiltrated over the network. This might be the result of Android publishing best practices on how to identify devices, with the number #1 practice being how to avoid using the IMEI and Android ID. However, the goal is not to identify the device, but rather the installation of the application on the device. We observed that many of the applications generated their own GUID to identify the installation on the device. These are randomly generated and not considered personal data.

The report from SINTEF (2016) showed that sports and dating applications transmitted potentially sensitive user data to a complex myriad of third-party services [31]. We did not observe similar behavior in our study. Even though Møteplassen is owned by a large multimedia company, their privacy policy and cookie policy gives meaningful information about the recipients of the personal data, same goes for two of the pharmaceutical companies. Vitusapotek and Apotek 1 provides information about the use of map services, ad services and application monitoring.

6.3 Repercussion and Limitations

The immediate consequence of not being GDPR compliant is administrative fines. Norway's supervisory authority do have other sanctions to impose, in addition to, or instead of administrative fines. For example, they could issue warnings, reprimands or a temporary ban. The severity of each administrative fine depends on multiple factors, like negligence, previous infringements, categories of data, etc. None of our findings include server-side vulnerabilities that could lead to the exposure of personal data for multiple users. That was never in the study scope. All our findings are related to local storage, or the infringement of the rights of the data subject. At

the time of writing, it is unclear how Norway's supervisory authority would impose sanctions on these types of violations. However, violations of the rights of the data subject are eligible for administrative fines up to 20 000 000 EUR.

It is difficult to draw general conclusion from a small study like this. Our findings are specific to each application, even each released version of the application. Therefore, based on our current findings we cannot draw conclusions that apply to multiple other applications, or categories of applications. Simply because our sample size is too small. Although, our study covers applications that together make up 75% of the pharmaceutical market share in Norway [10]. It paints a picture on how these applications process user's personal data and if they do so in compliance with the GDPR. For the dating applications, it is more difficult to determine their market share because they all claim to be Norway's most popular dating application. Nonetheless, dating applications have the ability to collect a substantial amount of personal data about an individual. This makes it especially difficult for them to navigate the GDPR with all its regulatory requirements.

Chapter 7
Conclusion

In this study we set out to answer the following research question:

RQ: How does pharmaceutical and dating Android applications implemented the regulatory requirements of the GDPR, and are they in their current state compliant?

We conclude that most of the applications examined in this study does not fully comply with the requirements of the GDPR. Our study revealed instances of disclosure of personal data. This is most prevailing in the storing of user's social security number unencrypted in a realm database. Further, multiple applications did not allow the user to opt-out of automatic individual decision-making for direct marketing purposes. We also found instances of processing of special categories of data like religious belief and political opinion. These types of data warrant special consent under article 9. Lastly, in relation to our goal, we found examples of good implementations of the GDPR in regard to managing consent and providing meaningful information about the purpose of personal data processing.

We followed a design science methodology which revolves around evaluating IT artifacts to solve identified organizational problems. Each application was rigorously tested to evaluate its compliance with the GDPR. Our test cases focused on the security of processing and how the applications communicated information to the data subject. We set up a mobile application testing environment using freely available tools. Our analysis approach was both static and dynamic. We started by decompiling each application to obtain an approximation of the original source code. Subsequently, we ran two automatic analysis tools on the code. This gave us a starting point for the manual code review. After the static analysis, we executed all the functionalities of the application. An intercepting proxy was set up to observe the information being exchanged between the device and third-party services. During this stage, we looked for persistent device identifiers, like the IMEI and the Android ID. Any data sent to a third-party domain not directly used to host the main functionality

of the application, could be a violation of the data subject's rights. We also examined what personal data was written to local storage during normal use of the application, and how the application handles personal data upon termination.

Our study is limited by the amount of applications tested. Previous studies on Android application security have tested hundreds of applications using mostly automatic tools. These studies did not test for compliance with the GDPR, but rather on the disclosure of personal data and device identifiers. Many of these studies suffer from lack of application context. Since our study was rather small, we were able to fully understand how the application collects and processes personal data. Our study examined five applications in total, three pharmaceutical and two dating applications. None of the them fully complied with all the requirements of the GDPR.

To successfully implement the GDPR into a mobile application requires knowledge of the current app ecosystem and development strategies. Mobile application developers are not lawyers. They might struggle with translating the legal requirements to technical solutions. If the best solution is to develop a top-ten list, single PDF or some other solution is a topic for further discussion. They must be easy to comprehend and vague enough to be transferable to new technologies.

Ensuring compliance with the GDPR is an ongoing process. It must be embedded as an operational activity for data controllers. Mobile application developers must constantly update their application to ensure confidentiality and integrity of user's personal data. An interesting study would be to examine the applications in six months' time to see if they have been updated to comply with the GDPR.

At the time of writing the implementation date for the GDPR has yet to arrive in Norway. Currently, it is under consideration to be incorporated into the EEA agreement [16]. However, Norwegian businesses that offer goods or services to European citizens, or monitor the behavior of European citizens, must comply with the regulation. Norway's supervisory authority predicts that the GDPR will come into force in July 2018, at the earliest. In the months after its implementation date, we will better understand how Norway's supervisory authority will enforce the regulation. But if Datatilsynet is to learn from Steve Wood, the Head of International Strategy & Intelligence at the Information Commissioner's Office (England's supervisory authority), "there will be no grace period."

References

[1] Jagdish Prasad Achara, Vincent Roca, Claude Castelluccia, and Aurélien Francillon. Mobileappscrutinator: A simple yet efficient dynamic analysis approach for detecting privacy leaks across mobile OSs. *CoRR*, abs/1605.08357, 2016.

[2] Android Developers. Android Debug Bridge. Available at: https://developer.android.com/studio/command-line/adb.html. Accessed: 14.04.2018.

[3] Android Developers. App Manifest Overview. https://developer.android.com/guide/topics/manifest/manifest-intro.html. Accessed: 29.05.2018.

[4] Android Developers. Best Practices for Unique Identifiers. https://developer.android.com/training/articles/user-data-ids. Accessed: 29.05.2018.

[5] Android Developers. Data and File Storage Overview. https://developer.android.com/guide/topics/data/data-storage. Accessed: 29.05.2018.

[6] Android Developers. Permissions Overview. https://developer.android.com/guide/topics/permissions/overview. Accessed: 29.05.2018.

[7] Android Developers. Platform Architecture. https://developer.android.com/guide/platform/index.html. Accessed: 23.03.2018.

[8] Android Open Source Project. Application Security. https://source.android.com/security/overview/app-security. Accessed: 16.04.2018.

[9] Android Open Source Project. System and Kernel Security. https://source.android.com/security/overview/kernel-security. Accessed: 12.04.2018.

[10] Apotekforeningen. Apotek i Norge. http://www.apotek.no/fakta-og-ressurser/statistikk-for-2016/1--apotek/1-1-apotek-i-norge. Accessed: 20.05.2018.

[11] Steven Arzt, Siegfried Rasthofer, Christian Fritz, Eric Bodden, Alexandre Bartel, Jacques Klein, Yves Le Traon, Damien Octeau, and Patrick McDaniel. Flowdroid: Precise context, flow, field, object-sensitive and lifecycle-aware taint analysis for android apps. *SIGPLAN Not.*, 49(6):259–269, June 2014.

[12] William Enck, Peter Gilbert, Seungyeop Han, Vasant Tendulkar, Byung-Gon Chun, Landon P. Cox, Jaeyeon Jung, Patrick McDaniel, and Anmol N. Sheth. Taintdroid: An information-flow tracking system for realtime privacy monitoring on smartphones. *ACM Trans. Comput. Syst.*, 32(2):5:1–5:29, June 2014.

[13] William Enck, Damien Octeau, Patrick McDaniel, and Swarat Chaudhuri. A study of android application security. In *Proceedings of the 20th USENIX Conference on Security*, SEC'11, pages 21–21, Berkeley, CA, USA, 2011. USENIX Association.

[14] Michael D. Ernst. Static and dynamic analysis: Synergy and duality. In *WODA 2003: Workshop on Dynamic Analysis*, pages 24–27, Portland, OR, USA, May 2003.

[15] European Data Protection Supervisor. Guidelines on the protection of personal data processed by mobile applications. Technical report, EDPS, November 2016.

[16] European Free Trade Association. Document 32016R0679. http://www.efta.int/eea-lex/32016R0679. Accessed: 31.05.2018.

[17] European Parliament and the Council of the European Union. The protection of individuals with regard to the processing of personal data and on the free movement of such data. October 1995.

[18] European Parliament and the Council of the European Union. The legal protection of computer programs. April 2009.

[19] European Parliament and the Council of the European Union. Regulation on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC. April 2016.

[20] European Union Agency for Network and Information Security. Privacy and data protection in mobile applications. A study on the app development ecosystem and the technical implementation of GDPR. Technical report, ENISA, November 2017.

[21] Facebook. Whitehat Program. https://www.facebook.com/whitehat. Accessed: 25.10.2018.

[22] Adrienne Porter Felt, Erika Chin, Steve Hanna, Dawn Song, and David Wagner. Android permissions demystified. In *Proceedings of the 18th ACM Conference on Computer and Communications Security*, CCS '11, pages 627–638, New York, NY, USA, 2011. ACM.

[23] Tom Fox-Brewster. Check the permissions: Android flashlight apps criticized over privacy. *The Guardian*, 2014. https://www.theguardian.com/technology/2014/oct/03/android-flashlight-apps-permissions-privacy Accessed: 19.03.2018.

[24] Github User pxb1988. Dex2jar - Tools to work with android .dex and java .class files. Available at: https://github.com/pxb1988/dex2jar. Accessed: 14.04.2018.

[25] Github User Skylot. Jadx - Dex to Java decompiler. Available at: https://github.com/skylot/jadx. Accessed: 14.04.2018.

[26] Alan R. Hevner, Salvatore T. March, Jinsoo Park, and Sudha Ram. Design science in information systems research. *MIS Q.*, 28(1):75–105, March 2004.

[27] Douwe Korff. EC Study on Implementation of Data Protection Directive 95/46/EC. September 2002.

[28] Bernhard Mueller, Sven Scleier, Romuald Szkudlarek, and Jeroen Willemsen. OWASP Mobile Security Testing Guide. https://sushi2k.gitbooks.io/the-owasp-mobile-security-testing-guide/content/. Accessed: 20.03.2018.

[29] Srinivas Nidhra and Jagruthi Dondeti. Black box and white box testing techniques-a literature review. *International Journal of Embedded Systems and Applications (IJESA)*, 2(2):29–50, June 2012.

[30] Portswigger. Burp Suite Community Edition. Available at: https://portswigger.net/. Accessed: 14.04.2018.

[31] Antoine Pultier, Nicolas Harrand, and Bae Brandtzæg. Privacy in Mobile Apps. Technical report, SINTEF - Networked Systems and Services, February 2016.

[32] Snapchat. Snap Inc. Terms of Service. https://www.snap.com/en-US/terms/ Accessed: 12.02.2018.

[33] Sukker.no. Norges største på dating. https://sukker.no/. Accessed: 16.05.2018.

[34] SUPER Team. SUPER - Secure, Unified, Powerful and Extensible Rust Android Analyzer. Available at: https://github.com/SUPERAndroidAnalyzer/super. Accessed: 23.04.2018.

[35] SuperUser.com. SuperSu. Available at: https://play.google.com/store/apps/details?id=eu.chainfire.supersu&hl=en. Accessed: 13.05.2018.

[36] Marin Todorov. Realm is an Object Centric Modern Database for Mobile App. Available at: https://realm.io/. Accessed: 03.06.2018.

[37] Tony Trummer and Tushar Dalvi. QARK - Quick Android Review Kit. Available at: https://github.com/linkedin/qark. Accessed: 23.04.2018.

[38] Wireshark.org. Wireshark. Available at: https://www.wireshark.org/. Accessed: 14.04.2018.

[39] Atle Årnes and Catharina Nes. Datatilsynet - What does your app know about you? September 2011. https://www.datatilsynet.no/globalassets/global/english/apprapp_english.pdf Accessed: 19.03.2018.