POLITECNICO DI MILANO

Facoltà di Ingegneria

Scuola di Ingegneria Industriale e dell'Informazione

Dipartimento di Elettronica, Informazione e Bioingegneria

Master of Science in

Computer Science and Engineering

POLITECNICO

MILANO 1863

Cybersecurity of Digital Therapeutics: Analysis and Modelling of Mobile Applications

Supervisor: PROF. STEFANO ZANERO

Co-advisor: MICHELE CARMINATI, PHD

Master Graduation Thesis by:

ARIANNA RICCI

Student Id n. 921275

Academic Year 2020-2021

POLITECNICO DI MILANO

Facoltà di Ingegneria

Scuola di Ingegneria Industriale e dell'Informazione

Dipartimento di Elettronica, Informazione e Bioingegneria

Corso di Laurea Magistrale in

Computer Science and Engineering

POLITECNICO
MILANO 1863

Cybersecurity of Digital Therapeutics: Analysis and Modelling of Mobile Applications

Relatore: PROF. STEFANO ZANERO

Correlatore: MICHELE CARMINATI, PHD

Tesi di Laurea Magistrale di:

ARIANNA RICCI

Matricola n. 921275

Anno Accademico 2020-2021

Alla mia famiglia

SOMMARIO

Ad oggi, sono molti i vantaggi e le innovazioni che le nuove tecnologie hanno portato in diversi settori della vita quotidiana. Tra questi, l'ambito della medicina e della cura non fanno eccezione, anche se in questo campo la fiducia e la familiarità degli utenti sono fondamentali per la diffusione di nuovi strumenti terapeutici.

Una proposta interessante ed innovativa nel panorama della medicina digitale è sicuramente costituita dalle Terapie Digitali: una nuova categoria di prodotti che "forniscono interventi terapeutici direttamente ai pazienti usando un software basato sull'evidenza e valutato clinicamente per trattare, gestire e prevenire un ampio spettro di malattie e disturbi comportamentali, mentali e fisici"[1]. Sono prodotti terapeutici, paragonabili ai comuni farmaci, il cui principio attivo è però un algoritmo [2]. La diffusione di questi prodotti è iniziata qualche anno fa, ma ha subito recentemente una forte accelerazione a causa della pandemia da Covid-19, che ne ha reso evidenti le grandi potenzialità.

Fin da subito le Terapie Digitali sono state considerate utili per la cura di malattie croniche come il diabete o l'asma, le quali si stanno diffondendo sempre di più, gravando sui sistemi sanitari, a causa dell'invecchiamento della popolazione [3]. Un altro interessante campo di applicazione delle Terapie Digitali è quello della cura dei disturbi mentali e comportamentali, disordini spesso sottovalutati o stigmatizzati per i quali è più complicato trovare un approccio terapeutico efficace [4]. Nel 2017 è nata la Digital Therapeutics Alliance (DTA), un'associazione di categoria senza scopo di lucro per la promozione delle Terapie Digitali, la quale si propone di raggruppare diverse aziende produttrici per standardizzare i requisiti e i processi di sviluppo.

A causa del Covid-19, inoltre, la considerazione delle Terapie Digitali è decisamente cambiata. Durante un momento di crisi sanitaria è sicuramente importante poter monitorare e gestire le terapie a distanza, evitando che le persone fragili si espongano al contagio, così come ridurre i costi di alcune terapie per garantire maggiori risorse al sistema sanitario. Inoltre, le condizioni di reclusione e isolamento affrontate hanno portato a una preoccupante diffusione dei disturbi mentali e comportamentali ed ad una conseguente maggiore attenzione al loro trattamento [5, 6]. La crisi sanitaria ha portato un profondo cambiamento nella percezione da parte degli utenti di prodotti innovativi come le Terapie Digitali ed anche ad un concreto aumento della loro richiesta. Per questo si è investito molto per velocizzare i processi normativi ed ampliare le opzioni di rimborso. Anche istituzioni mediche come la Food and Drug Administration (FDA) e la National Institute for Health and Care Excellence (NICE) hanno proposto nuovi regolamenti e piani di rimborso, in accordo con le aziende farmaceutiche ed assicurative [5–7]. Tutto questo ha accelerato notevolmente la diffusione delle Terapie Digitali e si prevede un aumento esponenziale del mercato nei prossimi anni [3].

Tuttavia, passare dalla promozione di un prodotto alla sua ampia diffusione sul mercato nel giro di pochi anni è un passaggio decisivo che richiede grande attenzione. Soprattutto bisogna considerare quanto la fiducia degli utenti (medici e pazienti) sia fondamentale per l'effettiva adozione di un nuovo strumento terapeutico [8]. I prodotti devono essere

percepiti come sicuri per essere credibili e non solo dal punto di vista medico ma anche da quello tecnologico, in termini di cybersecurity e privacy.

Una violazione dei dati o della privacy potrebbe avere conseguenze molto gravi, non solo per la salute del paziente, ma anche per l'adozione di questi prodotti nel lungo periodo e per gli investimenti futuri in tecnologie simili. Accelerando il processo regolatorio delle Terapie Digitali si rischia di diffondere prodotti terapeuticamente efficaci ma non sufficientemente sicuri dal punto di vista tecnologico. Infatti, diversi studi hanno dimostrato la vulnerabilità delle applicazioni mediche o di benessere già diffuse negli ultimi anni, che possono essere considerate antenate delle Terapie Digitali [9–15]. Sebbene le Terapie Digitali superino diversi limiti di queste app, potrebbero comunque aver ereditato alcuni loro difetti. Infatti l'efficacia clinica delle Terapie Digitali deve essere dimostrata e regolamentata come se si trattasse di veri e propri farmaci, ma riguardo gli standard di sicurezza informatica non sono ancora richiesti requisiti stringenti. Inoltre, le Terapie Digitali potrebbero presentare nuovi tipi di vulnerabilità e minacce, specifiche per questi prodotti e ad oggi non ancora approfondite. Purtroppo, non esistono ancora certificazioni tecniche complete ed adatte a garantire un adeguato standard di cybersecurity.

In questa tesi, ci proponiamo di compiere un'analisi sistematica e completa riguardo l'attuale livello di sicurezza delle Terapie Digitali, tenendo conto delle loro caratteristiche per individuare i possibili problemi di privacy e cybersecurity e, soprattutto, per proporre delle linee guida di sviluppo adeguate. Prima effettuiamo un'indagine teorica per delineare le caratteristiche e le proprietà comuni di questi prodotti al fine di definire un modello di minaccia specifico per essi. Successivamente, utilizziamo questo modello per testare un campione significativo di applicazioni. In particolare, grazie a diversi tipi di analisi (statica, dinamica ed automatica), abbiamo un quadro completo delle vulnerabilità presenti nelle applicazioni. Infine, grazie al modello creato e ai risultati ottenuti, possiamo valutare l'attuale livello di cybersecurity delle Terapie Digitali e definire alcune linee guida che dovrebbero essere prese in considerazione per il loro sviluppo.

Dai risultati sono emersi diversi problemi, alcuni più generali e altri più specifici, ma quasi tutti inaspettati considerando il tipo di prodotto. In particolare, crediamo sia necessario ed urgente aumentare il livello di sicurezza delle Terapie Digitali. Gli stessi sforzi che sono stati fatti a livello normativo, commerciale e medico per rendere questi prodotti accessibili, rimborsabili e soprattutto efficaci, devono essere fatti anche a livello tecnologico per renderli sicuri e affidabili.

Se si vogliono sfruttare a pieno le potenzialità di questi prodotti, che potrebbero davvero rivoluzionare l'approccio terapeutico a diverse malattie, è fondamentale garantire uno standard di cybersecurity più elevato. Riteniamo sia essenziale da un lato garantire il rispetto dei requisiti di sicurezza base delle applicazioni mobile e dall'altro definire nuove linee guida tecniche e specifiche per questi prodotti. Per questo motivo, presentiamo un modello di minaccia specifico e lo utilizziamo per esaminare l'attuale condizione delle Terapie Digitali. In conclusione, delineiamo una prima bozza di buone pratiche che possa essere considerata un punto di partenza per uno standard più ampio e ufficiale.

ABSTRACT

The Covid-19 pandemic had devastating effects on our lives over the past year, but not all changes turned out to be negative. In particular, it led to a major technology push and accelerated the digitalization process in several sectors, including healthcare.

In the medical field, Digital Therapeutics (DTx) are an innovative proposal for treating, managing and preventing a broad spectrum of behavioural, mental, and physical diseases and disorders. They provide therapeutic interventions directly to patients using evidence-based and clinically-evaluated software. These products received a significant prompt during the pandemic as many stakeholders realized their great usefulness and potential. DTx products are expected to become widespread in the coming years and could change the therapeutic approach to different diseases.

A great effort has been put into demonstrating the efficacy of DTx, ensuring their therapeutic value, and making them accessible through insurance reimbursement plans. However, there are still some challenges to overcome for their development. The technological profile and security have not yet been fully investigated. To date, there are few specific guidelines and technical requirements to ensure a good cybersecurity standard for these apps.

In this Thesis, we present a modelling and analysis of the DTx cybersecurity to broaden the debate regarding their adoption. In particular, we investigate the specific characteristics of these products to define a suitable threat model for them, and we test this model on a significant sample of applications. Through analysis, we evaluate the current security status of these applications, identifying the most vulnerable components. Finally, we outline more specific cybersecurity guidelines for the Digital Therapeutics (DTx) development based on the identified characteristics and results.

ACRONYMS

AI Artificial Intelligence.
API Application Programming Interface.
APK Android Application Package.
AWS Amazon Web Services.

CBT Cognitive Behavioral Therapy.

DB Database.
DTA Digital Therapeutics Alliance.
DTx Digital Therapeutics.

EHR Electronic Health Record.

FDA Food and Drug Administration.

GDPR General Data Protection Regulation.

HIMSS Healthcare Information and Management Systems Society.
HIPAA Health Insurance Portability and Accountability Act.
HITRUST Health Information Trust Alliance.
HTTP HyperText Transfer Protocol.
HTTPS HyperText Transfer Protocol over Secure Socket Layer.

MD5 Message Digest Algorithm 5.
MDCG Medical Device Coordination Group.
MDRF International Medical Device Regulators Forum.
mHealth Mobile Health.
MITM Man-In-The-Middle.
ML Machine Learning.

NICE National Institute for Health and Care Excellence.

OWASP Open Web Application Security Project.

PHI Protected Health Information.
PII Personally Identifiable Information.

SOC2 Service Organization Control 2.
STRIDE Spoofing, Tampering, Repudation, Information Disclosure, Denial of Service, Elevation of Privilege.
SUD Substance Use Disorder.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 INTRODUCTION

Nowadays, few areas of everyday life are not influenced by the advent of new technologies. Even though progress in this field is slower and less likely to take root, the healthcare sector does not make an exception. In healthcare, elements such as the trust of both patients and doctors and familiarity with new tools can be significant obstacles.

An interesting proposal for innovation in this field are Digital Therapeutics (DTx): a new type of products that "deliver therapeutic interventions directly to patients using evidence-based, clinically evaluated software to treat, manage and prevent a broad spectrum of behavioural, mental, and physical diseases and disorders" [1]. They are therapeutic products comparable to standard drugs, whose active ingredient is no longer a chemical or protein molecule but an algorithm [2].

DTx products already attracted attention a few years ago, but their great potential to crucially change the perspective of Digital Health emerged relatively recently, during the Covid-19 pandemic.

In particular, the possibilities of using these products to treat chronic diseases, which are increasingly widespread and costly in an ageing population [3], and mental disorders or substance abuse, often stigmatized or underestimated [4], have been studied for some years. In fact, in 2017, the Digital Therapeutics Alliance (DTA) was born. DTA is a non-profit trade association to promote the diffusion of DTx products, thanks to the standardization of requirements and development processes.

However, it is undoubtedly due to Covid-19 that the vision of these products changed. On the one hand, during a pandemic period, monitoring people remotely is crucial to preventing fragile people from contagion. Also, reducing the price of some treatments is vital to guarantee more resources for the healthcare system. On the other hand, the conditions of confinement and isolation brought a worrying spread of mental disorders, and consequently greater attention to their treatment [5, 6]. This particular time of crisis led to a change in the perception of DTx and an actual increase in demand, positive regulatory developments, and a significant advance in reimbursement options. Medical institutions such as the Food and Drug Administration (FDA) and National Institute for Health and Care Excellence (NICE) proposed new regulations and new reimbursement plans in consultation with pharmaceutical and insurance companies [5–7]. These elements greatly accelerated the spread of these products, and the market is expected to explode exponentially in the coming years [3].

However, in a few years, we went from simply promoting the DTx to making them widely available. This decisive transition requires great care, especially since the users' (doctors and patients) trust is crucial to the effective adoption of a medical product [8]. DTx must be perceived as safe to be credible in the long term, not only from a therapeutic perspective but also from a digital perspective in terms of cybersecurity and privacy. A

data breach or privacy violation could have severe consequences, not only for the therapy and the patient's safety but also for the reputation of these products and the future investment in these technologies. The sudden regulatory acceleration risks spreading products that are not secure enough from a digital point of view.

Several studies demonstrate the vulnerability of medical or wellness applications, which have become widespread in recent years [9–15]. Although DTx overcome some limitations of these applications, they may also have inherited some of their shortcomings. Unlike generic wellness apps, clinical efficacy has to be demonstrated and regulated as it works for real drugs, but there are not yet stringent requirements for security standards. In addition, they may present new cybersecurity challenges, which have not been investigated so far. There are still no comprehensive technical standards and suitable certifications for these products to ensure proper cybersecurity.

In this Thesis, we conduct a systematic and comprehensive analysis regarding the current security status of DTx, taking into account their peculiarities in analyzing the privacy and cybersecurity issues they might have and in proposing suitable guidelines. We carry out a theoretical investigation to delineate the common characteristics and properties in order to deduce a specific threat model for DTx. Then, we use this model to test a significant sample of applications with different types of analysis (static, dynamic, and automatic) to get a comprehensive report of all possible security issues. Finally, we evaluate the current security level of DTx and define some guidelines that should be taken into account for their development, based on the theoretical model and the obtained results.

Our results reveal several weaknesses in the security of DTx, some more general and others more specific, but many unexpected considering the type of product. In particular, we believe that there is a need to improve the security level of Digital Therapeutics (DTx). This means that the same efforts made at the regulatory, commercial and medical level to make DTx accessible, reimbursable, and above all effective must also be made at the technological level to make them secure and reliable.

If we want to fully exploit the potential of DTx, which could really revolutionize Digital Health and the therapy approach for the people, we need to propose a high standard of cybersecurity. Therefore, it is essential to ensure compliance with the basic security requirements for mobile applications and to define new technical and specific guidelines for these products. For this reason, we elaborate on a threat model and test it to determine the current situation. In conclusion, we outline one first draft of good practices, a starting point for a broader and more official standard.

This document is organized as follows:

- Chapter 2: In this Chapter, we discuss the research problem in more detail, defining objectives and challenges. We also present the related work we have referred to for better set-up and structure of our work.
- Chapter 3: In this Chapter, we introduce the global characteristics of the Digital Therapeutics (DTx), such as reference architecture and the common features. Therefore, we present the identified categories for the analysis and the reason why they are interesting. Eventually, we define a concrete and specific threat model to be used to test the selected products.
- Chapter 4: In this Chapter, we make a technical explanation of the analysis method of the applications: static, dynamic and automatic. We present the technical choices made, and the tools used to conduct each type of analysis.
- Chapter 5: In this Chapter, we present the results obtained from the DTx analysis. First, we describe the selected applications and motivate their relevance. Then we report the analysis results in a tabular form: we test the threat model on each selected app and assign a severity level to the vulnerabilities. Next, we comment on the results of each application, giving examples of the most interesting found vulnerabilities. Finally, we examine the global results and present our considerations about them.
- Chapter 6: In this Chapter, we propose a set of guidelines and good practices for developing DTx, based on the model and the results. We also specify why certain precautions are essential in products such as DTx.
- Chapter 7: In this Chapter, we explain the limitations to our work, such as the number of apps selected or the components analyzed, and indicate what would be interesting areas to explore in order to expand this research.
- Chapter 8: In this Chapter, we summarize our work and make some final considerations for the future developments of DTx.

2

MOTIVATION AND BACKGROUND

In this Chapter, we detail the reasons behind this work and some relevant aspects of the problem. We summarize then the studies conducted so far concerning the cybersecurity of DTx, for which suitable standards have not been defined yet. In addition, we report on some studies concerning cybersecurity issues both for Android apps in general and more specifically for mobile health applications. Finally, we define the goals of this research and the challenges we have to overcome to achieve them.

2.1 PROBLEM STATEMENT

Due to the Covid-19 pandemic, the digitalization process in several sectors, including healthcare, accelerated. Before the pandemic, different challenges and obstacles hindered the development of Digital Health [16]. However, there was a substantial increase in digital technologies in the healthcare field during the health crisis [17]. Unfortunately, along with the increase in telematic visits or medical applications, also cyber-attacks increased. Even before the pandemic, healthcare IT systems were already the target of several attacks, and many criminals took advantage of the time of crisis to get even more. The cyberattacks in the healthcare sector increased by 150 per cent during January-February 2020 [18].

The main problem in Digital Health is the inconsistency between the sensitivity of the data processed and the security measures used to protect them. Patients' medical records can be worth much money (from $10 to $1,000 per record[19]), and altering them can have severe consequences. These data can be used, for example, to produce genuine-looking insurance claims or to manipulate the medical history and prescriptions of patients.[20] An alternative to directly stealing medical data is to make it inaccessible and demand a ransom to decrypt.

However, although sensitive and valuable data, or even the patients' safety, is at stake, it seems that cybersecurity is not yet considered a priority in Digital Health. Hospital IT systems are vulnerable to DoS [1] attacks and ransomware infections. For instance, in 2018, the MGM Hospital in Vashi fell victim to a cyberattack that locked the hospital data and demanded a ransom in bitcoins. Similar incidents of WannaCry, Petya, and NotPetya ransomware attacking the healthcare sector were reported from many countries [18]. Also, some standard medical devices intended to perform a few specific tasks are not yet designed with the right level of security in mind. Although they do not store patient data directly, they interact with hospital IT systems and can facilitate an attack on a server or network [18]. The 2020 HIMSS Cybersecurity Survey reveal that "70% of hospitals surveyed had experienced a significant security incident within the past twelve months, including phishing and ransomware attacks that resulted in the disruption of

1 Definition and explanation of Denial of Service attack in Appendix A

IT operations (28%) and business functions (25%), as well as data breaches (21%) and financial losses (20%)" [19].

Overall, while Covid-19 significantly accelerated the digitalization of the healthcare sector, it also highlighted the security limitations of Digital Health. During the pandemic, weaknesses in IT systems and medical devices became apparent and were exploited for an increasing number of attacks. Many studies point out a serious lack of attention and investment in cybersecurity in Digital Health, which instead should be the key element to ensure a correct and secure digitalization process in the long term [16, 18, 19].

Within this context, our topic of interest is Digital Therapeutics (DTx), a new category of products that "deliver therapeutic interventions directly to patients using evidence-based, clinically-evaluated software to treat, manage, and prevent a broad spectrum of behavioural, mental, and physical diseases and disorders"[1]. These products are part of Digital Health [4] and are emblematic of the described situation. While their potential emerged during the pandemic, leading to their rapid spread on the market, there are no yet stringent security requirements. As a result, they could be a lucrative new target for cybercriminals.

As mention in Chapter 1, DTx have undergone significant development in the last two years, mainly due to the Covid-19, and it seems that their uptake is set to increase further. There have been many efforts to speed up the regulatory process, facilitate reimbursement and guarantee therapeutic efficacy. However, there are still many challenges to be solved, especially regarding privacy and cybersecurity [21]. On the one hand, medical data are often collected and analyzed by DTx to provide and adapt therapy to the patient, so privacy issues are very relevant. The general personal data of users (indicated as Personally Identifiable Information (PII) [2]) are already considered sensitive and have to follow a precise regulation. In particular, medical data (indicated Protected Health Information (PHI) [2]) are considered more sensitive and valuable and therefore require even more attention to their processing. On the other hand, since the therapeutic process relies partially or entirely on software, damage or failure can have severe consequences on the effectiveness of the therapy and also on patient safety. Moreover, some DTx products offer the possibility of exporting data in an electronic format suitable for integration with hospital information systems, such as the Electronic Health Record (EHR). This extends DTx features but also increases the attack surfaces. On the one hand, altering or tampering with data could damage the DTx itself and also the IT infrastructure of a hospital, with severe consequences. On the other hand, given the vulnerability of IT hospital systems, the DTx could lose a feature or be damaged. Finally, for DTx to become part of a new image of medicine, both patients and doctors must perceive the products as safe and reliable. Unfortunately, some studies show that the desired level of trust is not yet there [5].

Of course, for users to perceive these products as reliable, developers must guarantee not only the therapeutic efficacy but also the safety and security.

2 The list of PII and PHI data can be found in Appendix A

To conclude, a high level of cybersecurity and data management is fundamental to the long-term adoption of these products, but to date, these two aspects have not been explored in depth. Our contribution is to identify, model, and test the vulnerabilities of Digital Therapeutics (DTx) to understand their security level and draw up guidelines to ensure the secure development of these products.

2.2 RELATED WORK

In this Section, we present work related to our research. In particular, in Subsection 2.2.2, we consider the few but important studies concerning the cybersecurity of Digital Therapeutics (DTx). In Subsection 2.2.1, we report some relevant studies concerning the Android apps and mHealth apps cybersecurity, which help analyze DTx.

2.2.1 *DTx Security Guidelines*

This Section presents the current guidelines defined for DTx and why they are not considered suitable. Several studies point out a lack of attention to cybersecurity and the inadequacy of the currently available standards.

Patel and Butte in [21] underline how very few of the FDA's product summaries (about 5%) include cybersecurity content. They also state the need for DTx products to be subjected to an independent cybersecurity risk assessment documentation, updated annually to keep up with new threats.

Casalicchio et al. in [22] claim the need to define a specific appropriate standard for DTx, as the standards currently available do not cover certain specific characteristics of these products. They argue that documents like Medical Device Coordination Group (MDCG) [23] and International Medical Device Regulators Forum (MDRF) [24], which provide an overview of cybersecurity risks and indications for generic medical systems or applications, are very general and indicative because they are designed for a vast category of products. Therefore, Casalicchio et al. suggest reducing the level of abstraction and providing a more detailed view because DTx have specific peculiarities which need to be treated in depth and which, in a way, make the use case atypical. They finally stress that companies and users must be ready to adopt these products by receiving digital training.

Even the few documents written and explicitly designed for DTx are rather abstract and generic.

The DTA manifesto mentions "Incorporate patient privacy and security protections" as one of the fundamental principles DTx products must adhere to [25]. However, to guarantee both patient privacy and product security is recommended to adhere to exist certifications, such as Service Organization Control 2 (SOC2) or Health Information Trust Alliance (HITRUST), and regulations, like General Data Protection Regulation (GDPR) or Health Insurance Portability and Accountability Act (HIPAA) [26]. Again, these certifications are insufficient because they are generally designed for digital products

and focus only on certain aspects of security or privacy. Furthermore, obtaining one of these certifications is recommended but not mandatory for the DTx release. Further suggestions present in the DTA "Best Practices" [26] are also generic and abstract, such as alerting users about data breaches or having robust data monitoring system, without specifying in technical terms how these indications are to be implemented.

Torous et al. in [27] propose several other guidelines for the promotion of mental health apps, such as having transparent policies for data sharing and storage or frequently reviewing apps to identify new threats immediately. Again, although they are fundamental concepts, they remain at a high level.

It is not easy to embed technical details within guidelines, which is why we should work more on creating ad-hoc standards and certifications for DTx products, similar to those that already exist for other types of products [22].

2.2.2 *Andorid and mHealth Applications Security*

In this Section, we report some studies conducted in the last decade regarding the cybersecurity of generic Android applications and more specific medical applications known as Mobile Health (mHealth). These studies are interesting to understand the general issues of the Android platform and, more specifically, of mHealth apps that can be considered ancestors or relatives of DTx. However, the mHealth category is quite extensive, comprehending wellness apps or devices as well as more disease-specific ones.

Android is an open-source operating system for mobile devices. The fact that it is open source makes it compatible with so many different devices that it currently dominates the mobile operating system market [28]. However, both the fact that Android is open source and its wide diffusion make the system an attractive target for cybercriminals. Therefore, the widespread use of Android applications does not guarantee its security.

A recent study shows numerous vulnerabilities of different types in at least one-third of the approximately 3,000 applications analyzed. These vulnerabilities mainly relate to the use of insecure components or libraries, leaky code, and unsafe permissions [29].

University of Zakho et al. in [30] also identify different categories of vulnerability in the Android platform, such as Information Leakage, Privilege Escalation, Repackaging Apps, Denial of Service Attacks, Colluding. Information leakage problems are linked to the permission control mechanism of the Android operating system that ineffectively protects user resources and privacy from malware. Permission issues occur when an unauthorized activity gains extra permissions to obtain resources or sensitive information to which it should not have access. The possibility of repackaging is critical because it allows to insert malicious code into the main code of an app and then distribute as if it were the original app, thanks to repackaging techniques. Because smartphones are not fortified or have fewer protections than PCs, the developers of malicious applications find it an appropriate platform for DoS attacks. The primary objectives of DoS attacks are targeting overusing limited CPU, battery power, memory, and network bandwidth. Finally, colluding consists of deploying a group of applications with the same certificate

and granting various permissions. These applications can get access to all their resources and permissions by taking advantage of a shared UID.

Another overview also shows similar categories for grouping Android app vulnerabilities. In particular, it mentions Unprotected Activities and Private File Access. However, it also gives examples of vulnerabilities related to network configurations or encryption algorithms [31].

Finally, the tool developed by Wang et al. in [32] is relevant because it uses different types of analysis, static and dynamic, to detect Privacy Leakage in Android applications and, above all, to understand if they are justifiable or not by the app's functionality. The interesting aspects are the analysis and methods used by this tool to detect and find possible privacy leaks. They provide an idea of which vulnerabilities or components may expose sensitive information.

The reported categories and studies are helpful to understand the general cybersecurity issues of the Android platform and get an idea of the vulnerabilities that may exist in the popular Android mobile apps. However, to have an even more precise reference for our work, we focus on the current and widespread threats and vulnerabilities in applications similar to DTx. In particular, the mHealth apps.

For instance Knorr, Aspinall, and Wolters in [9] and in [10] detect serious security problems on a sample of more than 150 applications for the treatment of diabetes and hypertension. In particular, they denounce a serious lack of encryption and input validation, such as health reports saved in clear-text, the use of the HTTP protocol for some communications, which makes it particularly easy to intercept the app's traffic, and a serious absence of privacy policies, often non-existent or incomplete.

Also, Vermeulen et al. in [11] analyze three glucose monitoring apps, among the most popular, and still found many security problems. In particular, they find several shortcomings and vulnerabilities in data storage (personal and medical data stored in clear-text and the phone's public memory), communication protocols (use of HTTP) and authentication methods (weak password requirements).

Aliasgari, Black, and Yadav in [12] also analyze similar applications, focusing on communication protocols and compliance with HIPAA (US data management regulation). They discover that about 20 out of 25 applications have HTTPS protocol configuration problems, and only one app is completely transparent in how the data are managed.

Finally, Papageorgiou et al. comprehensive research conducted on more than 150 applications report several common and alarming problems and vulnerabilities, again in terms of encryption, communication and exposure of sensitive data [13].

Many of these apps are ancestors or similar to DTx, so it is helpful to keep their vulnerabilities in mind for our work. Other interesting works concern the vulnerabilities of apps or devices that interact with medical apps, many of which interact with DTx products.

Braghin, Cimato, and Della Libera in [14] point out some issues related to communication with tracking devices, such as Bluetooth communication and data encryption, and

test an example of a fitness tracker, finding that Bluetooth communication had problems with both authentication and encryption.

Similarly, Zhang, Shahriar, and Riad in [15] analyze three important trackers such as FitBit, Jawbone and Google Glass and find several vulnerabilities related to the Bluetooth protocol both in terms of incorrect configuration and weak authentication.

In general, we keep in mind the results of these studies both in the modelling phase, considering integration with insecure applications as a possible issue, and in the testing phase, taking inspiration for selecting tools and analyses to be carried out. We discuss both of these aspects in more detail in Chapters 3 and 4, respectively.

2.3 GOALS AND CHALLENGES

This work aims to model and test the possible vulnerabilities of DTx products to understand their security level and draw up more fitting guidelines. Our goal is also to clear up doubts concerning the limits of DTx, with a study focusing on the cybersecurity aspect. On the one hand, this objective is essential to overcome the problems presented in Section 2.2.1, but on the other hand, it has some challenging aspects.

Firstly, the specific characteristics of the DTx products are often presented generically and abstractly, without a concrete and in-depth reference to the aspect of cybersecurity, as they are included in a global analysis of the product. This requires an initial theoretical evaluation to identify the truly original and relevant aspects to be analyzed.

Secondly, even once they have been identified, it is not always possible to fully test these points of interest for technical reasons. As DTx are already present on the market, we must respect the manufacturing companies' limits and permissions. We cannot access some components without the consent of the manufacturer or owner. For example, while it is possible to do different analyses on an app downloaded on a personal device, attacking or hacking into a company's server is not legal without the consent of the manufacturer or owner. Therefore, we cannot test some fundamental components of the architecture of these products, such as the Cloud Platform, the doctors' Web Dashboards and the integration with the EHR.

Also, some functionalities of mobile applications are only available on prescription. We contacted the companies to request permission to access their products in order to expand our research. Still, unfortunately, only one company was cooperative and provided us with an access code for their app. However, we consider these components in the modelling phase, focusing on the accessible parts for testing.

Despite these remarks, we think that the guidelines proposed at the end of this work can be used as a starting point for defining an official standard dedicated to these products. A standard and official security regulation would undoubtedly help the spread of DTx, which could be perceived as more secure and reliable.

3

THREAT MODEL

In this Chapter, we explain the threat model designed and used for the analysis of DTx. In Section 3.1, we illustrate the typical characteristics of DTx and the key elements for a cybersecurity analysis: assets, threat agents and attack surfaces. In Section 3.2, we expose and motivate the different categories of this model. On the one hand, 3 macro-categories defined ad-hoc for this type of application. On the other hand, the STRIDE categories, already widespread and used for threat modelling, can help formulate a more formal and universal standard for the DTx development. Finally, in Section 3.3 the complete model is presented, with the specific list of vulnerabilities of each category and the threatened assets.

3.1 Digital Therapeutics (DTx) characteristics

In order to define a suitable threat model for DTx products, it is important to understand their characteristics and key elements, especially from a cybersecurity perspective.

3.1.1 *Reference Architecture*

We define a DTx reference architecture, illustrated in Figure 3.1 and, for each component, we describe the different functionalities implemented. Not all products have the same features or components, but this architecture reflects the typical DTx product and is useful to understand particular cases.

1. **Sensor & Smart Devices**: One of the main characteristics of DTx is to be data-driven and based on real-world outcomes. Many products, especially those for the treatment of chronic diseases such as diabetes and asthma, use sensors or medical devices to collect patient data, such as health values or medication intake, and transmit it to the application, usually via Bluetooth.

2. **Mobile or Web Application for the Patient**: The main characteristic of DTx is that they are digital therapies, i.e. they have software as an active ingredient [2]. The mobile application is the main interface of the patient to the therapy and is also a fundamental tool to make him participate and be involved in his treatment. Depending on the disease, different functionalities are offered: daily monitoring of the patient (through data collected by the sensors or entered manually), medications reminders and suggestions, data sharing and exporting in different formats, charts and insights about therapy progress, Cognitive Behavioral Therapy (CBT) programs with interactive therapeutic sessions such as video, audio, questionnaire, lessons and rewards, community and blog to communicate with the other users, chat and email to communicate with the clinician or the company.

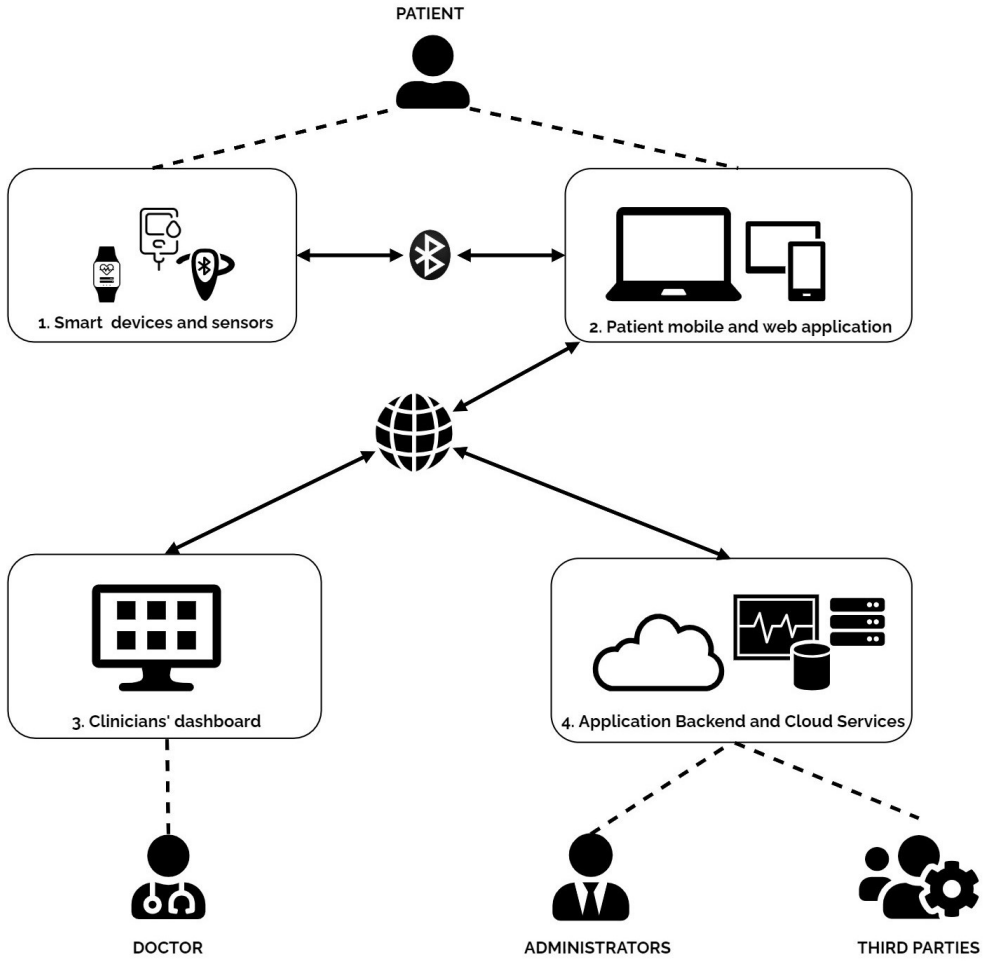

Figure 3.1: Reference Architecture Digital Therapeutics (DTx)

3. **Web Dashboard for Clinicians**: Another important functionality of DTx products, whose relevance became evident during Covid-19, is the possibility for doctors to monitor patients remotely. Especially applications for chronic diseases treatment, such as diabetes or asthma, offer a web dashboard with which the doctor, or sometimes even a relative, can access the patient's data and medical values saved on the application and synchronized on the dashboard. Some products designed for people with addiction problems offer this dashboard, too. In particular, through the dashboard, clinicians can register patients to the treatment program, set up therapy, view patients' medical data and progress, access charts and insights about one or more patients, update and modify treatment values and parameters.

4. **Application Backend & Cloud Services**: The last component, also fundamental to the functioning of DTx products, is the application backend, often hosted on a private, public or hybrid Cloud Platform. Patient data is saved and stored on the Cloud to be synchronized with the doctors' dashboard and used for further analysis. In fact, a major innovation in DTx is the use of Machine Learning (ML) and Artificial Intel-

ligence (AI) algorithms to improve and personalize therapy. Thus, on the one hand, by analyzing the anonymized data of many patients, it is possible to gain a better understanding of the disease and its trends. On the other hand, by examining each patient history, it is possible to more accurately predict the amount of medication recommended or to adapt the CBT program undertaken.

The Figure 3.1 also shows the users that interface with the different parties: the patient mainly uses the medical devices and the application, the doctor the web dashboard and the company administrators the application backend services. Finally, there is a third type of user, third parties from outside the company who use and access the data stored in the Cloud Platform to offer intelligent analysis and monitoring services, such as ML and AI algorithms, which are often outsourced. Moreover, external services are often used for app performance monitoring or advertisements and may threaten user privacy, as considered in the model in Section 3.3.

Finally, a somewhat unique third party may be the hospital IT infrastructure. For example, some DTx products offer the possibility to send and share data collected by the application with the IT infrastructure of certain hospitals. In particular, they use digital formats compatible with those implemented by hospitals IT systems, such as EHR.

3.1.2 *Threat modeling elements*

Thanks to the reference architecture and the common features of DTx, it is possible to identify the central elements for cybersecurity analysis.

First, it is important to define key cybersecurity concepts, such as vulnerability, threat and risk. A vulnerability is a flaw or weakness in a computer system that can be exploited by an attacker to breach or attack the system. A threat is a potential malicious event that can exploit vulnerabilities and results in an unwanted impact on a computer system or application. Finally, the risk is a statistical and economic evaluation of the exposure to damage because of vulnerabilities and threats. These elements are at the cores of the threat model in Section 3.3.

Then it is important to define elements that are more specific to the systems and linked to their characteristics, such as assets, threat agents and attack surfaces. Considering what DTx products do, what components they include and who uses each part, we define these elements.

3.1.2.1 *Assets*

From a cybersecurity perspective, an asset is anything valuable in the analyzed system. For instance, it could be hardware, software, information or even reputation. In the case of DTx, there are some undeniable assets, others less evident. However, being therapeutic products, as their name implies, the two main assets are the patients' privacy and, above all, the patients' health and safety.

Nevertheless, an equally fundamental, and perhaps less clear asset, is also the reputation of the companies. Since these products are receiving significant attention and

could become an integral part of future medicine, a cyber attack, even without severe consequences for patients' health, could seriously harm the future development of these products and their long-term success.

Lastly, we must also consider the various components with which DTx interface and interact, which are often expensive medical devices.

In particular, we determinate a list of 10 assets described in Table 3.1.

ID	ASSET	DESCRIPTION
A1	Personal Identifiable Information	PII includes all sensitive data that can identify a person [1]
A2	Protected Health Information	PHI includes all medical data that can identify a person[1]
A3	Therapy	An attack on the implemented therapy could reduce its effectiveness, modify its effects or prevent the patient to access it.
A4	Health Safety	Severe consequences for the patient's health could also occur if the therapy is heavily altered or denied.
A5	Device Information or Localization	This category includes data about the device or its location. They are often used for app features, but if exposed or shared with unauthorized third parties, they can breach privacy.
A6	Physical Devices (wearable, phone, pc)	Some attacks can compromise the functioning and availability of a physical device
A7	Company Reputation	The company's reputation could be severely compromised by any kind of attack, especially if not properly managed
A8	AI and ML Algorithms	Altering input data or patients trends could alter the AI/ML algorithms are used to calculate and predict disease and treatment progression.
A9	Hospital IT System	If the input data is altered, it may not be compatible with the hospital's IT system or may even damage it.

Table 3.1: Digital Therapeutics (DTx) Assets

1 Full list of which data fall into the category of Personally Identifiable Information (PII) and which fall into the category of Protected Health Information (PHI) in A

3.1.2.2 *Threat Agents*

Threat agents are the people or the things that could cause an attack. In the case of Digital Therapeutics (DTx) this category certainly includes all the people interested in obtaining such sensitive data, either to sell them or to use them, but also people interested in demonstrating the weakness of these products or their inadequacy for large-scale distribution. However, there is a further category to consider: the users themselves who are not yet ready or able to use these products.

Thimbleby in [34] report the 2018 case of legal proceedings against nurses in England who tampered with patient data. In reality, this tampering was mainly due to misuse and misunderstanding of the medical devices used.

Both in analyzing the vulnerabilities of DTx and even more in reasoning about the guidelines for their development, in Chapter 6, we take into account the fact that a user, whether a patient or a doctor, not trained and prepared to use the product could cause even more severe damage than that caused intentionally by a cybercriminal.

In Table 3.2 we present the main threat agents identified.

ID	THREAT AGENTS	MOTIVATION
TA1	Cyber-criminal	Resell data, demand a ransom, block one or more parts of the system.
TA2	Insurance Companies	Classify patients, modify insurance plans, exploit statistics or forecasts
TA3	Competitor	Damage reputation or slow down product development
TA4	Third Parties	Access to, use or resell more data than required by privacy policy
TA5	Internal of the Company	Voluntary or involuntary access to sensitive data or alteration of the product and software
TA6	Illegitimate Users	User who would not have the right to access the product who finds a way to access it or accesses it to carry out an attack
TA7	Legitimate Users	User with right of access to the product who misuses or damages it
TA8	Malware or Virus	Malware or viruses already present on the patient's phone or doctor's PC that may attack the app to obtain information or control

Table 3.2: Digital Therapeutics (DTx) Threat Agents

3.1.2.3 *Attack Surfaces*

Attack surfaces are the parts of a system that can be exposed to attacks and vulnerabilities. For our analysis, we can consider the 4 components of the reference architecture and their communication protocols (Bluetooth, network) as generic attack surfaces. Furthermore, each component is composed of many specific attack surfaces.

However, for our modelling and analysis we focus mainly on the patient application (Component 2) due to the access issues explained in Section 2.3. Regarding the various attack surfaces of this component, it is more useful to refer directly to Chapter 4, where the method of analysis of each surface is also explained (communication protocol, memory, code, memory).

3.2 CATEGORIES DEFINITION AND EXPLANATION

In this Section, we present and motivate the chosen categories for our threat model. Starting from the reference architecture, functionalities and cybersecurity elements exposed in the previous Section 3.1, we define a specific threat model for Digital Therapeutics (DTx). In particular, we identify 3 macro-categories to investigate, common to all products and relevant from a cybersecurity point of view. To each of these macro-categories, we apply the STRIDE categories, described in Section 3.2.2, to explain with more detail the potential vulnerabilities. Moreover, the STRIDE method has existed for many years. It is also popular in the corporate world, so it is helpful to make our model more standard and convenient for companies.

3.2.1 *Macro-Categories*

I. Mobile Environment

A key element, being also the main component of DTx products, is the mobile environment. All products offer a mobile application for patients to interface with the therapy. From a cybersecurity perspective, this represents a wide field of investigation because the mobile environment has several intrinsic issues that must be taken into account during the development of a DTx product.

Mobile devices are certainly more vulnerable and exposed than others since they are under the user's control and not of the company. In addition to the vulnerabilities common to mobile applications, such as those mentioned in Section 2.2, the inherent problems to the environment must be considered. Even if an application is developed in the best possible way, and we say in Section 2.2 that this is not true on average, the phone could be compromised by viruses or malicious applications already present before the app installation. In addition, the smartphone is also physically vulnerable, and it can easily be stolen, damaged or even just temporarily used by someone other than the owner without them noticing. An unaware user may have no idea of the security status of his or her own device and could nullify the app's security measures.

Similar reasoning applies to the Web Dashboard of doctors. Although less vulnerable than the mobile environment, the web dashboard is also outside the company's boundaries. Even doctors' computers could be compromised or easily hacked, even by an internal employee. Moreover, in this case, the health data of all patients and their treatment settings would be accessed so that an attack could have very severe consequences. However, as explained in Section 2.3, it is not possible to test this component without the company's permission, and therefore we do not include it in our threat model.

II. Therapeutic Purpose

Another key and particular aspect in analyzing the cybersecurity of DTx is that they have a therapeutic purpose.

A therapeutic purpose means that the products deal with very sensitive data, such as the Protected Health Information (PHI) among the assets 3.1, that must be appropriately managed. Therefore, it is essential for the correct course of therapy that the data are consistent and controlled. Medical data must always be protected by encryption, saved and stored appropriately, only accessed by authorized users, and, above all, they should never be altered or manipulated.

In addition, the issue of authentication and access also becomes very relevant in the therapeutic field: accessing a therapy that is not authorized and intended for that specific user can be as harmful as altering someone else's therapy. Even just knowing that a user is registered to a specific application and therefore suffers from a particular pathology or disorder is valuable information, especially for insurance companies or, worse, employers.

Finally, it is also very important that the users can manage and control their own health data and be actually aware of how they are processed.

III. Machine Learning (ML) & Artificial Intelligence (AI)

The last peculiar feature of DTx is the use of ML or AI algorithms to enhance the therapy's effectiveness and adapt it to the individual user according to his progress. This indeed constitutes a great potential in medical terms, but it implies many vulnerabilities to be considered.

A possible attack against this type of algorithm is data poisoning, i.e., manipulating and altering the data that train the algorithms to alter results. An alteration of the algorithms could not only invalidate the therapeutic benefits but also lead to undesirable collateral effects.

Moreover, these tools are often outsourced, and this increases the attack surface of DTx. The Cloud platform utilized to store and manipulate health data is often used by many companies simultaneously, which could voluntarily or involuntarily tamper with the algorithms or access sensitive data. In addition, since many companies use the same Cloud service (AWS), an issue or attack against AWS would damage almost all DTx products, resulting in a loss of credibility for the whole category.

However, again due to the arguments expressed in Section 2.3, the analysis of this category focused heavily on data poisoning caused by user input, leaving out the algo-

rithms details and the problems related to the Cloud Platform to which we have no access.

The same applies to the integration with the electronic files of the hospital information systems (EHR). Apart from analyzing the degree of manipulation of the inputs, we can not analyze anything else for the legal reasons expressed in Section 2.3.

These macro-categories are helpful to understand which characteristic aspects of Digital Therapeutics (DTx) to investigate from a cybersecurity point of view and which issues might affect them. However, to have a more precise and attractive model for companies, we apply the STRIDE method to define subcategories and organize the threats more formally, adapting it to DTx products.

3.2.2 *STRIDE Categories*

STRIDE is a threat model developed by Praerit Garg and Loren Kohnfelder in 1999 to identify threats to computer security. It is a mnemonic for security threats in 6 categories and is still used by companies today to identify things that could go wrong with security [35]. In Table 3.3, there are the 6 categories, their description and the properties violated by an attack in that category.

On the one hand, the STRIDE categories are handy to make our threat model clearer and more precise. However, on the other hand, we decide to apply them to the already defined macro-categories and adapt to the characteristic and specific elements of the Digital Therapeutics (DTx). For this reason, not all categories are applied to each macro-category, and, in some cases, we define new ad-hoc categories. In addition, some threats may belong to more than one category. In this case, we include a reference without reporting the same threat multiple times.

THREAT	THREAT DEFINITIION	PROPERTY VIOLATED
Spoofing	Impersonating something or someone else	Auhtentication
Tampering	Modifying data or code	Integrity
Repudation	Claiming to have not performed an action	Non-repudation
Information Disclosure	Exposing information to someone not authorized to see it	Confidentiality
Denial of Service	Denying or degrading service to users	Availability
Elevation of Privilege	Gaining capabilities without proper authorization	Authorization

Table 3.3: STRIDE Categories

3.3 THREAT MODEL

This Section presents the complete threat model, including the list of specific vulnerabilities for each category and the threatened assets. Then, we use this model to analyze and test a sample of applications, and the results obtained are presented in Chapter 5.

In particular, we define possible threats: potential malicious events or actions that may threaten the analyzed system, i. e. the DTx products. However, we also specify the types of attacks or vulnerabilities involved by these threats to make the model more concrete and facilitate both the testing activity and the drafting of technical and specific guidelines. Also, indicating the resources involved is helpful to understand the risk posed by each threat.

Mobile Environment		
STRIDE Category	**Threat**	**Assets**
Spoofing	**1.** It is possible to impersonate the legitimate app **1.1** It is possible to modify and resign the code (use of (only) SHA1 or v1 Scheme) **1.2** It is possible to perform a MITM attack [2] (see **3.**)	**A1, A2, A3 A4, A5, A7 A8, A9**
Tampering	**2.** It is possible to manipulate or modify the app code **2.1** There are present vulnerable classes (Activities, Events, Remote WebView) **2.2** Deprecated/outdated versions of libraries are used **2.3** Weak/outdated encryption algorithms are used (md5/SHA1) **2.4** Weak pseudo-random number generators are used **3.** It is possible to manipulate or modify the app requests **3.1** HTTPS is never used **3.2** HTTPS protocol is not used for every connection **3.3** Total absence of Certificate Pinning **3.4** It is possible to bypass Certificate Pinning [3]	**A1, A2, A3 A4, A5, A6 A7, A8, A9**

2 Definition of MITM Attack in Appendix A

3 See Section about *Dynamic Analysis* (4.1.2)

Mobile Environment		
STRIDE Category	**Threat**	**Assets**
Repudation	**4.** It is possible to assert that a certain operation has or has not been carried out **4.1** It is possible to modify the app's log functions (the logs function are present in the code, and it is possible to resign the code, see **1.1**) **4.2** The log files are saved in the phone's public/private memory and can be modified	**A3**, **A7**
Information Disclosure	**5.** Sensitive information is exposed in the app code **5.1** It is possible to decompile the code easily (automatic tools) **5.2** There is no obfuscation of the code **5.3** There is present relevant information in the code (secrets, URLs, APIs, firebase address) **5.4** Dangerous permissions are present/granted that seem unrelated to the functionality of the application and that could be used to obtain illegitimate information (READ/WRITE ON EXTERNAL STORAGE, GET ACCOUNTS, READ/WRITE CONTACTS, FINE/COARSE LOCATION) **6.** Sensitive information is exposed in the phone memory **6.1** PII and/or PHI data are stored in clear-text in the phone's public memory **6.2** The log files are stored in clear-text in the phone's memory (public or private) and contains sensitive info **6.3** Credentials, hash or tokens are stored in clear-text in the app's private memory **6.4** Medical DBs are stored in clear-text in the app's private memory	**A1**, **A2**, **A5**, **A8**

Mobile Environment		
STRIDE Category	**Threat**	**Assets**
Elevation of Privilege	**7.** It is possible to carry out privileged operations **7.1** It is possible to overwrite code functions to unlocked hidden/premium features **7.2** Dangerous permissions are granted that seem unrelated to the functionality of the application, and that could be used to obtain control/access to illegitimate functionalities (READ/WRITE ON EXTERNAL STORAGE, GET ACCOUNTS, READ/WRITE CONTACTS, AUTHENTICATE ACCOUNTS, FINE/COARSE LOCATION)	**A1**, **A2**, **A3**, **A4**, **A6**, **A7**

Table 3.4: Threat model- Mobile Environment

Therapeutic Purpose		
STRIDE Category	**Threat**	**Assets**
Spoofing	**1.** It is possible to impersonate another patient or legitimate patient **1.1** There is no access code **1.2** It is possible to know the format of the access code (when present) **1.3** There is no authentication factor, or it is not effective, so it is easy to register other users or fake users **1.4** There are not or weak requirements on the password **1.5** There is not a maximum number of attempts for the login **1.6** The authentication token does not expire **1.7** Automatic logout is not forced **1.8** The link to reset the password does not expire (never or only if a new one is generated) **2.** It is possible to impersonate the company **2.1** It is possible to send emails from the company domain	**A1, A2, A3 A4,A7, A8**

Therapeutic Purpose		
STRIDE Category	**Threat**	**Assets**
Tampering	**3.** It is possible to modify own data **3.1** It is possible to modify own personal data (e-mail/number) without being checked **3.2** It is possible to edit or delete own activity (post/comment/challenge) even if not allowed by the app **4.** It is possible to modify data of another patient **4.1** It is possible to edit or delete an activity (post/comment/challenge) of another user **4.2** It is possible to modify other users' messages	**A1, A2, A3 A4,A7,A8**
Information Disclosure	**5.** It is possible to know if a patient is registered to the application through **5.1.** Registration **5.2.** Login **5.3.** Reset Password **5.4.** API **6.** It is possible to gather information about other patients **6.1.** APIs are partially or completely public **6.2.** It is possible to carry out a SQL injection or an XSS attack **6.3.** Information from other users is saved in the phone's memory (see **6.**) **7.** It is possible to connect devices unsafe or with a lower level of privacy	**A1, A2,A3 A4 A5**

Therapeutic Purpose		
STRIDE Category	**Threat**	**Assets**
Denial of Service	**8.** It is possible to prevent a patient from the therapy (registration to the app) **8.1** There is no authentication factor, or it is not effective, so It is possible to register other people (see **1.3**)	**A3, A4,A7**
Elevation of Privilege	**9.** It is possible to access unauthorized therapies or functions **9.1** It is possible to register an authorized user (a user belonging to a category or living in a country for which the product is not authorised)	**A3, A4 A7, A8**
Patient Awareness	**10.** The patient is not aware of how data are processed **10.1** There is not a privacy policy, or it is not very clear **10.2** There are one or more intrusive tracking services (profiling/advertising)	**A1, A2,A7**

Table 3.5: Threat model- Therapeutic Purpose

Machine Learning & Artificial Intelligence		
STRIDE Category	**Threat**	**Assets**
Spoofing	**1.** It is possible to easily register a large number of fake patients (impersonate a real patient) **1.1** There is no authentication factor, or it is not effective, so It is possible to register other users or fake users (see **1.3**)	**A7, A8 A9**
Tampering	**2.** It is possible to send entries that do not respect the pre-set or threshold values	**A7, A8 A9**
Information Disclosure	**3.** It is possible to collect information and details about the the data used by the algorithms (thresholds and range)	**A8**

Table 3.6: Threat model- ML & AI

4

APPROACH AND IMPLEMENTATION

This Section presents the approach used to analyze and test the threat model on the applications' sample. In particular, we explain the assumptions considered and the 3 different types of analysis performed: static, dynamic, and automatic.

The practical part focuses on testing the threat model on a set of 10 relevant and meaningful applications. We describe the selection criteria and the description of each app in Chapter 5.1. As explained in Chapter 3, the application tests focus mainly on the mobile components of Digital Therapeutics (DTx), i.e. the mobile applications available for smartphones. However, there are different ways of testing a mobile application, and we refer to the guide proposed by OWASP for setting up the testing activity [36]. In particular, below there are our technical choices.

Android Applications
We test only Android applications. For years, the market was dominated mainly by Android and iOS applications, but the Android system has significant advantages in simplicity and relevance. The Android operating system is open-source and is used by a wide range of mobile devices, unlike iOS, which is closed and only runs on Apple's proprietary hardware. As almost all DTx products offer both Android and iOS versions, we consider it adequate to analyze only Android apps, for which there are more powerful testing tools available.

Physical Device
We prefer to test the applications on a rooted physical device rather than on an emulator. This is because, as also explained in the OWASP Mobile Security Testing Guide [36], "testing on a real device makes for a smoother process and a more realistic environment". It is possible to have greater access to the entire mobile environment, including memory and interfaces, and more realistic and faster application execution.

Gray-box Testing
The executed tests can be considered grey-box as the information available for each application is partial. Since the applications are already on the market, the only information available is the one published by companies. Some apps are completely accessible by the user, while others require an access code that some companies have provided while others have not. Also, concerning the code and documentation, it is only possible to access the source codes through decompilation and the few public reports or those provided by the companies.

To sum up, we download the APK from the Google Play store and test the apps on a rooted Samsung S7 device running Android 9, API level 28.

4.1 ANALYSIS METHOD

We refer to OWASP Mobile Security Testing Guide [36] and also to the existing literature on Android applications (Section 2.2), to decide to carry out 3 types of analysis on each app: Static, Dynamic, and Automatic. Figure 4.1 shows the followed steps for the analysis of each application schematically. In addition, the implementation details and the tools used for each analysis are explained in the dedicated sections.

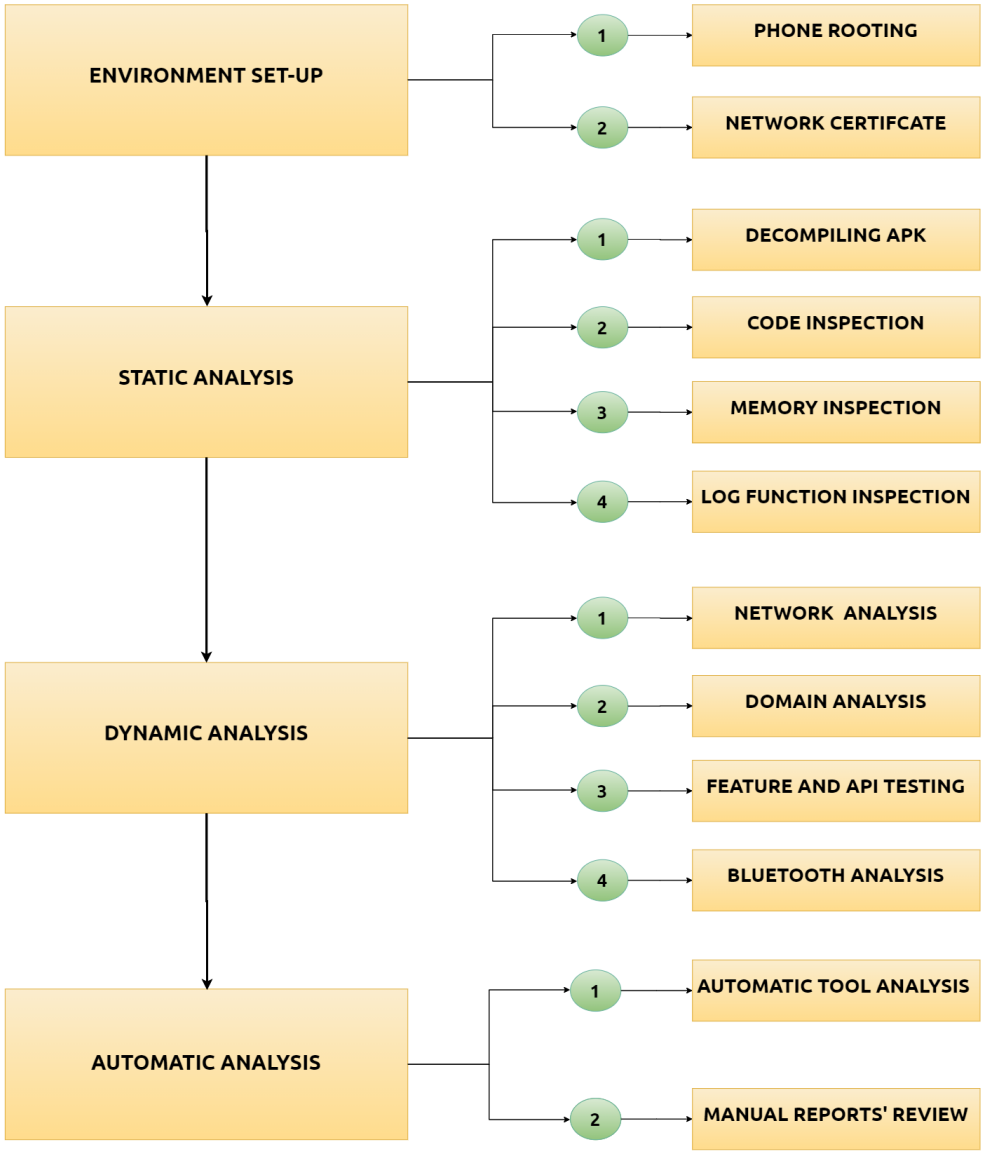

Figure 4.1: Implementation Steps Diagram

4.1.1 *Static Analysis*

The static analysis includes examining the application code, the encryption algorithms, the libraries and the data stored in memory. Below, we explain which tools we used for the analysis of these elements and which procedure we followed.

Code
After downloading the app from the Google Play store, it is possible to decompile the Android Application Package (APK) using tools such as apktool [38], and jadx [37]. These tools provide the application's source code and configuration files. If the application is written in *React-Native* (2 of the analyzed applications), it is not even necessary to use these tools. Simply convert the APK into .zip format and open the *index.android.bundle* file with an editor. Sometimes, the decompilation is not perfect, and some classes or methods are not very clear. However, none of the apps uses obfuscation methods, and it is quite easy to understand how the app works and the code structure.

Moreover, we inspect the more relevant elements of the source code. The *AndroidManifest.xlm* describes the app structure, its components (activities, services, content providers, and intent receivers), and requested permissions. In particular, it is helpful to see what permissions are requested and whether or not they are linked to the app's functionality. The most interesting elements are the libraries, which may contain known vulnerabilities, especially outdated ones. Also, the files in the *res* folder, which sometimes have configuration files containing details or credentials, and above all, the primary classes, which in some cases may be manipulated or overwritten to perform malicious operations.

Finally, some relevant hard-coded information or secrets can be found through keyword searches (*username, password, secret, firebase, API*).

Data Storage
After using and dynamically testing the app to generate fictitious data, it is possible to discover what kind of information is saved by the app, how it is saved and in which memory's space.

The adb tool [39] is helpful to download the desired folders of the phone memory. In particular, the *adb pull* command allow downloading both the phone's public memory and the application's private memory.

The data saved in the public memory are accessible by any application with permission to read the memory (READ EXTERNAL STORAGE) and by anyone who takes possession of the device. Therefore, this information is highly exposed. The data stored in the application's private memory are only accessible by the application itself. However, if the device is not secure, i.e. if it has been previously rooted or has been infected by a virus, this storage space may be no longer secure. Furthermore, if the phone is stolen or lost, rooting is sufficient to access these memory folders. Therefore, data stored in the app's private memory is safer than data stored in the public memory but should not be saved in clear-text if sensitive. In particular, the most interesting folders to explore are *databases* and *shared_prefs* which often contain sensitive information. To open the databases, it is

sufficient to use a DB reader such as *SQLite* [45].

Log
There are several aspects to investigate to understand what kind of information is logged by the application. On the one hand, searching for log functions within the code is helpful to see how often and for what purpose they are invoked.

Secondly, it is interesting to analyze the application memory to look for the log files present (often in the *files* folder) and check what information they contain.

Lastly, a further aggravation occurs when the app's logs are accessible via the *logcat* function, which can be executed even without rooting the phone. In this way, even just a few minutes of access to the device makes it possible to download all the logs and access the sensitive information they contain.

4.1.2 *Dynamic Analysis*

The dynamic analysis concerns mainly the communication protocols and the app's functionalities. We manually generate fictitious users' profiles to analyze the application's requests and responses between the client and the server. In concrete terms, we perform the analysis explained below.

Network
We use Burp-Suite [40] to intercept and manipulate network packets. This tool not only allows us to see the requests exchanged but also to intercept and modify them.

The set-up of this tool requires more or less effort depending on the application's network configuration. To intercept HTTP traffic, it is sufficient to set up Burp's Proxy to see the requests exchanged. Fortunately, almost no applications use HTTP. To intercept HTTPS traffic, it is necessary to install Burp's certificate on the mobile phone. This makes communication more secure, but the certificate may already be present on the phone or installed by a malicious application without the user being aware of it.

The best security measure to prevent the interception of network traffic is the implementation of Certificate Pinning. In this case, it is necessary to use Brida [41] (an extension of Frida [42] for Burp) to be able to intercept the packets exchanged. However, using Brida to bypass Certificate Pinning is quite complicated and requires specific technical skills. In particular, it requires access to the mobile device to intercept its network traffic. For us, it helps analyze applications, but we do not consider it a vulnerability. The vulnerability "it is possible to bypass pinning" (see 3.4), refers to the opportunity to intercept traffic relatively straightforwardly, without having to access the device or simply by installing a customized certificate. Otherwise, if it is necessary to use Brida, it means that the network configuration is correct.

Domain
By looking at the intercepted requests, it is possible to see which domains the calls are made to and get a clearer idea of the application architecture. In addition, we use Subfinder [46] to explore any other public domains of the company and check if any relevant information is exposed (domains used for testing, public APIs etc.).

Furthermore, the automatic analysis checks where the corresponding IP addresses are located and which company they belong to. This is also useful to understand whether the information entered in the Privacy Policy is correct and complete.

Features and API
The most challenging analysis is the analysis of application features and calls to the APIs. Thanks to the interception of network packets, we can reconstruct the paths and processes underlying each feature to understand their logic and correctness. We use Burp to manipulate requests and responses to test for inconsistencies or logical errors.

It is the most extended and most delicate analysis. We first use the app to get an idea of how it works. Then, we modify the parameters and the information sent by the app to alter the app features. In particular, it is necessary to analyze the exchange of many requests before detecting a vulnerability or a logical error, which often turn out to be very severe. Only by following all the steps of a specific operation is it possible to understand whether it is correct or not and whether it can be manipulated in some way.

Other Communications
As some applications offer pairing with medical or wellness devices, we also test the protocols for these communications, such as Bluetooth. In particular, we investigate the pairing process, which sometimes can be violated. If the pairing process is weak or vulnerable, it could be possible to associate any mobile phone to the device and intercept all the data exchanged.

Unfortunately, since there are not many applications with this option and not all devices are available, this analysis is somewhat limited, as explained in Chapter 7.

4.1.3 *Automatic Analysis*

For the sake of completeness, we also subject the applications to automated analysis. In particular, we select two interesting tools for their completeness and clarity.

MobSF [43] provides an extensive report based on the analysis of the APK file. In particular, the sections are:

- Certificate Information
- Permissions Information
- Manifest Analysis
- Code Analysis
- Domain Malware Check
- URLs
- Firebase Databases
- Emails
- Trackers
- Hardcoded Secrets

The most interesting information provided by this tool is the information about the APK signature in the first section (Certificate Information) and the information about

domains and server locations. The other information about code or permissions is undoubtedly interesting but very extensive and requires careful selection.

Ostorlab [44] is another online tool that analyses the APK. Its strength is that it provides a report divided into three very compact and clear sections:

- Vulnerabilities
- Information
- Libraries and Dependencies

In particular, the Vulnerabilities section also provides the severity of the detected issue. The Libraries section indicates the detected versions of the most known libraries, which are not always easy to find manually.

Both tools are helpful to verify and validate the manual analysis and get a general idea of the security level of the applications. However, they perform comprehensive analyses looking for known and widespread vulnerabilities, providing generic results that must be selected manually to find relevant information. Unfortunately, it is impossible to give an example of the reports obtained because it would be easy to get the name of applications that we present anonymously in this work. However, it is sufficient to use any APK to get an idea of the generated report.

5 EXPERIMENTAL RESULTS

In this Chapter, we present and comment on the obtained results. In Section 5.1, we describe the 10 applications selected for testing and their characteristics. In Section 5.2, we present our results. We use 3 Tables to summarize the results of testing each application against each vulnerability described in Chapter 3.3. Based on the analysis conducted, we also evaluate the severity level of each threat, indicated in the Table with a different shade of grey. In Section 5.3, we comment on the performed analysis and the most relevant results per each app. Finally, in Section 5.4, we discuss the general results and observations that can be deduced.

5.1 APPLICATION SELECTION AND DESCRIPTION

As explained in Chapter 4, all the tested DTx applications are Android and have been downloaded from the Google Play store. In this work, we present them anonymously as they are already available on the market. We contacted all the producer companies, but not all of them responded or consented to the analysis, so to be fair, we indicate the applications with fictitious names (alphabet letters).

We select a significant sample of 10 Applications based on two key elements. On the one hand, the fulfilment of the requirements to officially fall into the Digital Therapeutics (DTx) category. For this reason, we choose only applications that are present in the Digital Therapeutics Alliance (DTA) Library [47] or in the literature concerning DTx [3, 4, 21, 22]. On the other hand, the coverage of the two main fields of DTx intervention, namely chronic diseases and mental and behavioural disorders [33]. Unfortunately, some applications require an access code or a doctor's prescription. Therefore, we contacted the companies to request access, but only one company gave us a fictitious access code (app H). Other apps require interaction with medical devices that are not available in Italy. Again, we tried to contact the companies, but only one sent us the required device (app C). In Chapter 7 more details are given about the limitations encountered.

However, some analyses (the static and automatic ones above all) can be conducted even without complete access to the apps' functionalities. In any case, the selected sample includes at least two full access apps for both the main areas of intervention.

Below is a summary Table 5.1 with the selected applications and a brief description of them.

App	Disease	Access	Features & Notes
A	Diabetes 1 & 2	No	• Reporting manually or automatically (pairing with smartwatch and glucose meter) the glucose level • Synchronizing with wellness app to provide customized messages and notifications about lifestyle and meal plans • Exporting of SMART Visit (suitable for integration with hospitals) • Limited analysis
B	Diabetes 1 & 2	Yes	• Reporting glucose values manually or by plug-in a smart device directly in the smartphone. • Tracking carbohydrate count, physical activity and medication intake by synchronizing wellness app. • Sharing diabetes diary and charts by mail or web portal • Setting an emergence number • Perfect example of Tier 1 app. The innovative device for glucose is not currently available in Italy and it is not possible to test it.

App	Disease	Access	Features & Notes
C	Diabetes 1 & 2	Yes	• Reporting blood glucose values either manually or by pairing the glucometer • Calculating insulin and bolus with precise insulin dose recommendations and HbA1c estimation • Exporting report in PDF, Excel and CSV. • Perfect example of Tier 1 app, possible to test the pairing with a real glucometer (provided by the manufacturer)
D	Asthma	Partial	• Attaching custom-built sensors to the inhaler and connecting to the app via Bluetooth. Automatic medication monitoring • Insights, reminders and rating about the therapy such as Daily Asthma Forecast • Web portal for the doctor • Exporting asthma and COPD report in PDF • Most significant example about respiratory diseases. Proprietary sensors are not available in Italy. Partial test of the application
E	Depression	No	• Personalized CBT-based program to help improve depression symptoms in 12-weeks • Web application and not mobile • The access code has not been provided, limited analysis.

App	Disease	Access	Features & Notes
F	Anxiety	Yes	• CBT-based therapy to improve anxiety and stress management with different programs consisting of activities, games and tasks • Support community to communicate with other users • Perfect example of Tier 2 app, full access and opportunity to test all functionalities
G	Alcohol Addiction	Yes	• Changing and improving users' relationship with alcohol. More than 100 activities and challenges to achieve the user's goal • Peer community to exchange information and support with other users • Perfect example of a Tier 2 app, full access and analysis. Very important privacy issue as it treats addiction disorder
H	Chronic Back Pain	Yes	• Managing and improving of chronic back pain with a personalized training plan, the exercises adapt continuously. • High-quality videos ensuring the exercises are performed correctly. • Tracking of training progress and pain and sleep perception over time. • Only not free application for which we have been provided with the access code

App	Disease	Access	Features & Notes
I	Cancer Symptoms	No	• Managing and remotely monitoring of cancer symptoms through customized recommendations. • Information and resources to understand symptoms and learn about cancer care in general • Web dashboard for clinicians • Only application on the market oncology area, it is only available on prescription and therefore the analysis is limited.
L	Substance Use Disorder (SUD)	No	• Increasing of abstinence and retention with a 12-weeks CBT program • Evidence-based rewards system designed to support recovery and treatment • Recording substance use, cravings, and triggers and setting of reminders for medication • Web dashboard for clinicians • Prescription required, limited analysis

Table 5.1: Application Summary

5.2 TABLES OF RESULTS

In this Section, we present the results. For each vulnerability of the threat model in Chapter 3, we test each app. Some vulnerabilities are more severe than others, and therefore we use greyscale to express the severity level.

In particular, the following colours and meanings are associated as below.

The vulnerability is present and can be easily exploited.

There are problems with the vulnerability but not so easy to exploit (warning).

The vulnerability is not present or could not be tested.

Finally, it must be taken into account, as also explained in Section 5.1, that we had full access to all features of only half of the selected applications. We could perform an exhaustive dynamic analysis for these apps, while it was not possible for the others. This may create a misleading effect looking at the Table, but in Section 5.3 the issues and observations for each app are explained in detail. Indeed in some cases, even without a full analysis, we found some relevant problems. In other cases, however, we have considered correctly implemented apps as a starting point for defining good development practices to follow. This aspect is also explored in the following Sections (5.3, 5.4).

The Tables are divided on the three macro-categories defined in Chapter 3 and the description of the vulnerability is reported in legenda.

Table 5.2: Table of Results - Mobile Environment

APP	Spoofing		Tampering								Repudation	
	1.1	1.2	2.1	2.2	2.3	2.4	3.1	3.2	3.3	3.4	4.1	4.2
A												
B												
C												
D												
E												
F												
G												
H												
I												
L												

1.1. It is possible to modify and resign the code
1.2. It is possible to perform a MITM attack
2.1. There are present vulnerable classes
2.2. Deprecated/outdated versions of libraries are used
2.3. Weak/outdated encryption algorithms are used
2.4 Weak pseudo-random number generators are used
3.1. HTTPS is never used
3.2. HTTPS protocol is not used for every connection
3.3. Total absence of Certificate Pinning
3.4. It is possible to bypass Certificate Pinning
4.1. It is possible to modify the app's log functions
4.2. The log files are saved in the phone's public/private memory

Table 5.3: Table of Results - Mobile Environment

APP	Information Disclosure								Elevation of Privilege	
	5.1	5.2	5.3	5.4	6.1	6.2	6.3	6.4	7.1	7.2
A										
B										
C										
D										
E										
F										
G										
H										
I										
L										

5.1. It is possible to decompile the code easily

5.2. There is no obfuscation of the code

5.3. There is present relevant information in the code

5.4. Dangerous permissions could be used to obtain illegitimate info

6.1. PII and/or PHI data are stored in clear-text in the phone's public memory

6.2. The log files are stored in clear-text and contain sensitive info

6.3. Credentials, hash or tokens are stored in clear-text in the app's private memory

6.4. Medical DBs are stored in clear-text in the app's private memory

7.1. It is possible to overwrite functions to unlocked hidden/premium features

7.2. Dangerous permissions could be used to obtain access to illicit features

Table 5.4: Table of Results - Therapeutic Purpose

APP	Spoofing								
	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	2.1
A									
B									
C									
D									
E									
F									
G									
H									
I									
L									

1.1. There is no access code

1.2. It is possible to know the format of the access code

1.3. There is no authentication factor, or it is not effective

1.4. There are not or weak requirements on the password

1.5. There is not a maximum number of attempts for the login

1.6. The authentication token does not expire

1.7. Automatic logout is not forced

1.8. The link to reset the password does not expire

2.1. It is possible to send e-mails from the company domain

Table 5.5: Table of Results - Therapeutic Purpose

APP	Tampering				Information Disclosure							
	3.1	3.2	4.1	4.2	5.1.	5.2.	5.3.	5.4.	6.1.	6.2.	6.3.	7.
A												
B												
C												
D												
E												
F												
G												
H												
I												
L												

3.1. It is possible to modify personal data (e-mail/number) without being checked
3.2. It is possible to edit or delete an own activity even if not allowed by the app
4.1. It is possible to edit or delete an activity of another user
4.2. It is possible to modify other users' messages
5.1.. It is possible to know if a patient is registered through registration
5.2.. It is possible to know if a patient is registered through login
5.2.. It is possible to know if a patient is registered through reset password
5.4.. It is possible to know if a patient is registered through APIs
6.1.. APIs are partially or completely public
6.2.. It is possible to carry out a SQL injection or an XSS attack
6.3.. Information of other users is saved in the phone's memory
7.. It is possible to connect devices unsafe or with a lower level of privacy

Table 5.6: Table of Results - Therapeutic Purpose

APP	Denial of Service	Elevation of of Privilege	User Awereness	
	8.1	9.1	10.1	10.2
A				
B				
C				
D				
E				
F				
G				
H				
I				
L				

8.1. There is no authentication factor or it is not effective so it is possible to register other people

9.1. It is possible to register an authorized use

10.1. There is not a privacy policy or it is not very clear

10.2. There are one or more intrusive tracking services (profiling/advertising)

Table 5.7: Table of Results - ML & AI

APP	Spoofing	Tampering	Information Disclosure
	1.1	2.	3.
A			
B			
C			
D			
E			
F			
G			
H			
I			
L			

1.1. There is no authentication factor, or it is not effective, so it is possible to register other users or fake users

2.. It is possible to send entries that do not respect the pre-set or threshold values

3.. It is possible to collect information and details about the data used by the algorithms(thresholds and range)

5.3 RESULTS DISCUSSION - APPLICATION

5.3.1 *Application A*

App A is one of the most popular apps for treating type 1 and type 2 diabetes and therefore is an interesting case study. Unfortunately, it requires registration by the doctor and an access code, which was not provided to us by the company. Despite this, there are some significant problems.

Mobile Environment
The code is easy to decompile and contains considerable useful information, including all the paths and parameters for network requests, the structure of the DB tables, the required formats (regex) for the user's id and password. In addition, the code is also signed with outdated mechanisms (v1 and SHA1), and old versions of the necessary libraries are used for data storage and encryption. This information could be valuable if, for example, an attacker wants to register a dummy user or manipulate the app's code. Moreover, the application does not implement Certificate Pinning, making it easier to intercept network packets.

Therapeutic Purpose
Regarding the therapy, unfortunately, it is not possible to access the functionalities. However, it is interesting to note a few things: although the login is implemented correctly, either by password reset or by request to the API it is easy to know if a specific e-mail is registered or not. This information, as also mentioned in Chapter 3, is in itself very relevant because it indicates whether or not a person suffers from a particular pathology.

In addition, it is possible to understand that the access code consists of 15 char. It is not advisable to provide any information about the code, but such a high number of characters is enough to be protected against bruteforcing attempts. Also, a significant feature is that of being able to connect many different devices to synchronize data. Obviously, the larger the number of devices, the more insecure some of them could be.

Finally, the two warnings reported in Table 5.5 concern the presence of analytics services, which are reported in the privacy policy, and the possibility of easily downloading the application even in countries where it would not yet be available. However, since we did not make a real registration, we cannot know if control on location is implemented or not.

Machine Learning (ML) & Artificial Intelligence (AI)
Unfortunately, we could not test this functionality, which is why the row is also empty in the Table 5.7.

5.3.2 *Application B*

The application B is among the most significant for our study. It is a pretty widespread application for the management and care of diabetes. It has recently been included in the Digital Therapeutics Alliance (DTA) library [47], after having been updated (July 2021). However, it presents several serious vulnerabilities that have not been resolved even with the update. It is a perfect example that in some cases, to promote a product, the appropriate inspections are not carried out and that suggesting security measures without imposing them is not sufficient to guarantee the product's security.

Mobile Environment
Despite being one of the few applications to implement both Certificate Pinning and code signing methods correctly, it has many memory and encryption vulnerabilities.
Looking at the code, it is easy to see that a hash obtained from *MD5(e-mail in lowercase+md5(password)* is used to identify the user. First of all, the MD5 algorithm is now outdated because it is not difficult to break, and since the user's username and password are used, the probability of breaking it is even greater. In addition, the hash is saved in clear-text on the log file in the application memory, and it is also accessible via *logcat*. Since the logs are accessible via *logcat*, taking possession of the phone for a few minutes, it is possible to obtain the information necessary not only to authenticate the user even without discovering the password (by simply passing the hash) but also to trace the user's password. Bypassing authentication means accessing all user data and therapy settings, as discussed in more detail later.

In addition, in the *databases* folder, there are many DB not encrypted, which contain information about other users or employees (e-mail address and role is indicated), and other configuration files again contain credentials and tokens. Finally, despite the many functionalities implemented, granting some permissions is not justifiable (READ/WRITE EXTERNAL STORAGE/RECORD AUDIO). Moreover, it is pretty easy to unlock the Pro version by overwriting some functions in the code. Among these functions, it is also possible to unlock one that seems still under development.

Unfortunately, it was not possible to test the use of the proprietary device for measuring blood glucose because it is not available in Italy.

Therapeutic Purpose
Regarding therapy, the most severe vulnerabilities found were those concerning authentication and privacy violation. Although a second authentication factor is implemented, i.e. a code sent to the phone, this is not effective, as it is only checked on the client-side. In addition, the patient's e-mail is never verified, and no welcome e-mail is sent, making it very easy to register other users or fake users.

Regarding e-mail, the request sent to share a patient's diary by e-mail can easily be manipulated to change the content. The file with the user's diary remains attached, but the recipient could easily ignore it. Since the e-mail is sent from the corporate domain,

an attacker can easily ask the user to provide his private credentials. An example of a phishing e-mail [1] is shown in Figure 5.1.

About the telephone number, on the other hand, it is possible to enter any value as it is not effectively checked, not even when the user changes the value in the profile settings. Finally, there is no maximum number of login attempts and, more importantly, no forced logout. In this way, anyone with access to the phone could alter the patient's treatment parameters without even needing to find out the patient's hash or password.

As regards the violation of privacy, there are many problems. As for application A, it is easy to find out whether a user is registered or not. The login error message is correct ("wrong mail or password"), but it is easy to detect this information through the password reset feature and network requests manipulation. More serious is the fact that there are several tracking services to which details are sent every minute about every action performed by the user. This tracking activity is reported in the privacy policy, but it is very invasive. Another problem could be synchronization with fitness applications that are less secure than DTx in general or the use of HTTP for some requests.

However, the most severe vulnerability of all concerns the violation of the privacy of all patients: it was possible to manipulate the user's data request to perform SQL injections [1] and potentially obtain all the data stored in the company's DB. There is also an XSS [1] on the app's web portal. The SQL injection was fixed after the app's last update (July 2021), but such a severe and known vulnerability was significantly present in such a sensitive product. It is also very significant that this data breach was not reported to users. Potentially, all user data could have been accessed, and users must be informed when such a severe threat arises. The XSS is still present after the update, as all the other issues listed.

Machine Learning (ML) & Artificial Intelligence (AI)

The fact that unauthorized or fake users can easily be registered constitutes a real possibility of data poisoning. Furthermore, sending values designed to alter algorithms through these false accounts could actually alter their effectiveness.

It is also interesting to note that thresholds and formats must be respected for many medical data (insulin, bolus, blood glucose). This is undoubtedly very useful, on the one hand, because it protects against malicious input. However, on the other hand, it provides important detail on the values expected by ML and AI algorithms and therefore on their possible alteration.

1 Definition and explanation of these attacks in Appendix A

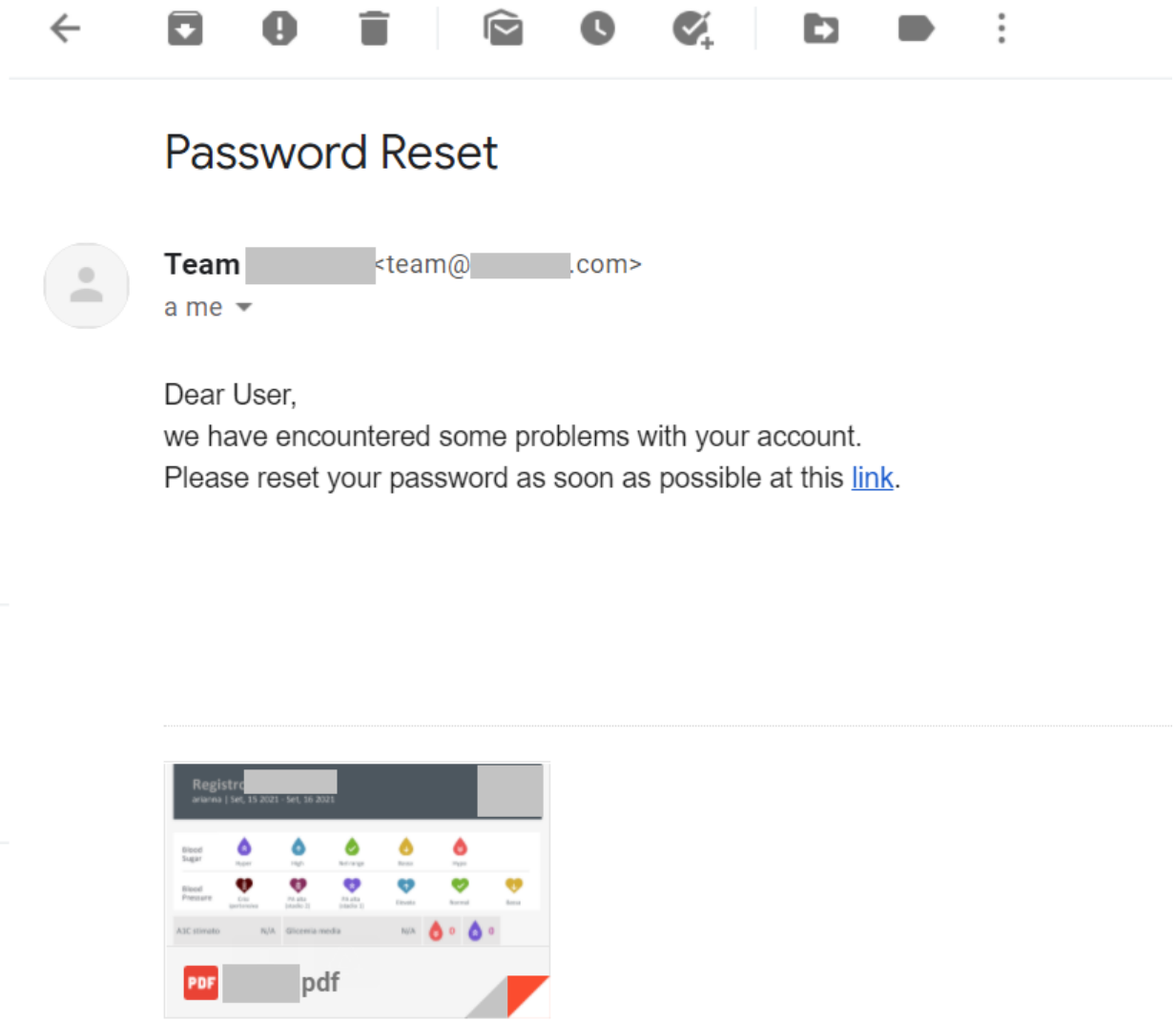

Figure 5.1: Phishing e-mail example. The user receives an e-mail from the company's domain (partially obscured because the applications are presented anonymously), asking him to reset his password. The user, thinking it is a legitimate request from the company, clicks on the malicious link and enters the new password. The page to which the user is redirected may be fake or contain a malicious script that intercepts his credentials. The logbook remains an attachment, but an inexperienced user could still proceed with the password reset without even noticing the attachment.

5.3.3 *Application C*

Application C is another representative application in the field of diabetes care and management products, as we were able to conduct profound analysis, finding some positive but also many negative aspects.

Mobile Environment
The positive aspects are the correct implementation of Pinning, which is not easy to bypass, and the absence of log files on the device. This means that logs are only saved on the server, a more secure environment than the mobile device. Another positive aspect is the connection to the Glucometer. Although the list of compatible devices is extensive, we were able to test one in particular. In addition, we checked that both pairing and data exchange appears to be secure as it is not possible to connect other devices or intercept data exchanged via Bluetooth in any way.

However, regarding data storage and code manipulation, there are several problems. The patient's medical data and all therapy settings are saved in clear-text in the databases folder, so this data would be easily retrieved if the phone is already rooted. The most severe problem, however, is that when the patient exports his diary, it is saved in the *Download* folder of the phone, as shown in Figure 5.2. This is serious because the phone's public memory is accessible by any application with reading permission and any person with access to the phone. While it may be convenient to export the report and have it in an easily accessible folder, the user may not realize the level of exposure of the data. Therefore, it would be desirable to notify the user of the potential consequences so that he/she can make a more informed choice about whether or not to expose his/her medical data.

Finally, it is possible to override certain functions to have the app's Pro version and discover features under development. Exposing unfinished features in an already widespread version of the app does not make much sense.

Therapeutic Purpose
Regarding the management of therapy and sensitive data, the problems are very similar to those of app B. It is possible to tell whether a user is registered even if only by logging in because the user is redirected to two different pages according to the e-mail entered (registered or not), but also by resetting the password and requests to API.
Also, the second authentication factor, a code sent by e-mail, is not effectively checked. As a result, it is possible to easily register another user or a fake user silently by manipulating network requests avoiding the welcome e-mail. This can have consequences on the ML algorithms and the possible registration of a user already registered by others. The example of the request to be manipulated is shown in Figure 5.3.

The password requirements are also very weak (8 characters), logout is not forced when closing the app, and there are no maximum login attempts. The password reset link does not work properly, and the authentication token is not invalidated at logout and does not have an expiry date set. All this makes user authentication very weak and access by attackers to someone else's application much easier. In particular, a weak password without a check on login attempts could make brute force an actual possibility.

Machine Learning (ML) & Artificial Intelligence (AI)
A major consequence of 'silent' registration of fake users or other people's e-mails can be to poison the data with which ML and AI algorithms are trained and alter the statistics on which treatment advice are based. Again, as in app B, several pre-set values are used, and suitable ranges are defined for the inputs. This certainly helps to control the input but also provides significant detail on the expected values. Moreover, by manipulating the calls to the API, it is possible to insert entries with random and inconsistent ids, and this could cause problems in the management of these data (data not considered or excluded from the statistics).

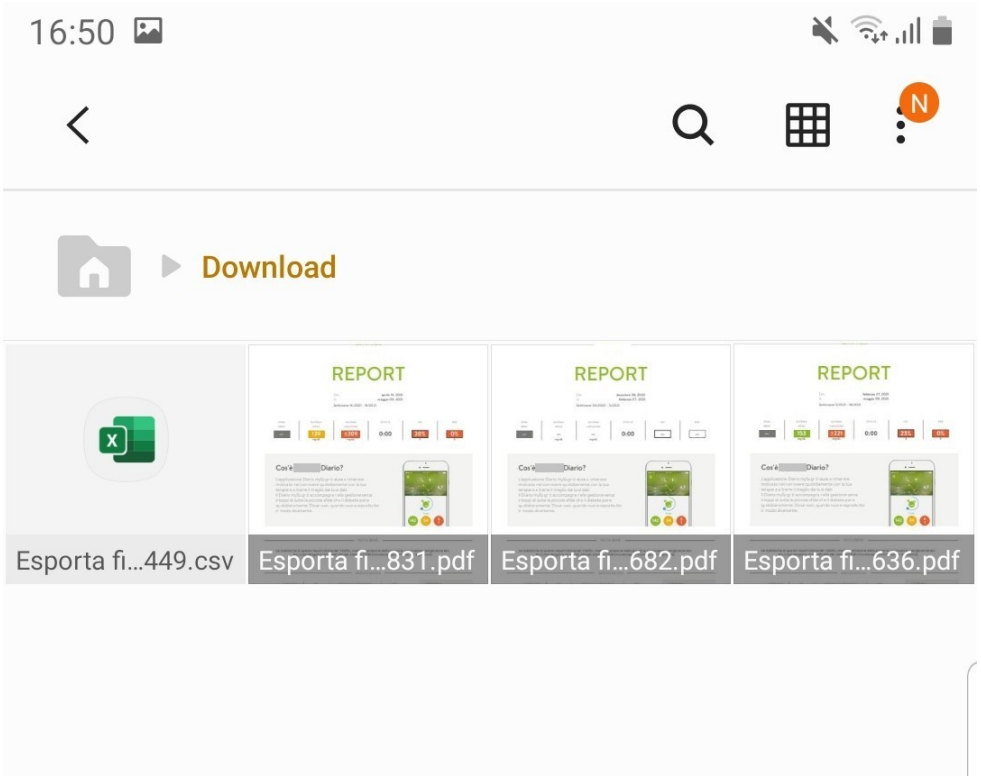

Figure 5.2: Phone "Download" folder containing reports of patient health values exported in CSV or PDF. The "Download" folder can be accessed by any application that has permission to read the phone's memory (READ EXTERNAL STORAGE), and many apps have this permission granted for various reasons (9 out of 10 of the apps analyzed have it). Therefore, any app with this permission can read these files saved in this folder and all the patient's medical values are contained in these reports.

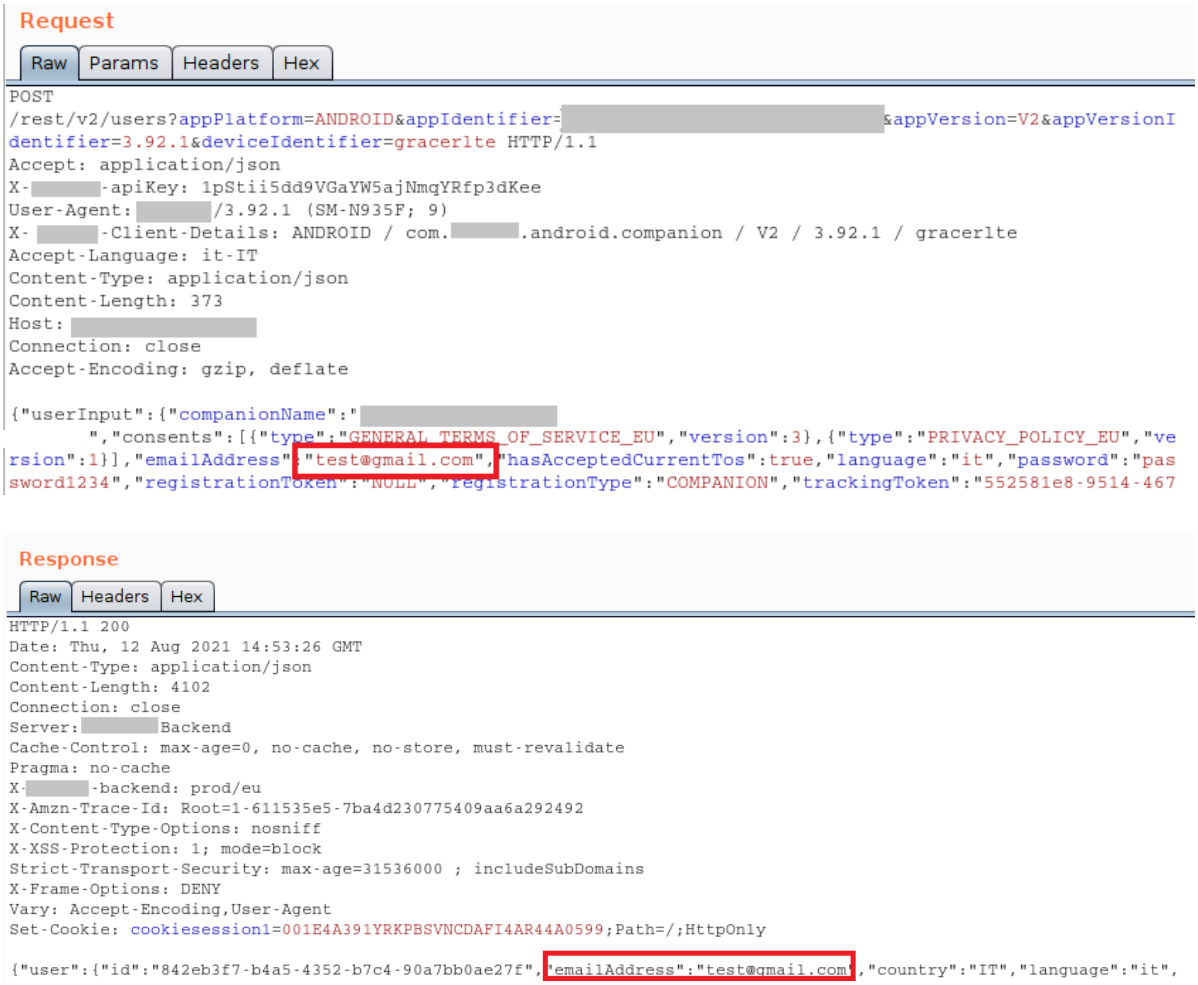

Figure 5.3: This request is sent to the server to register a new user. Theoretically, it should only be sent after the user's e-mail has been verified by code. However, in reality, modifying the "emailAddress" field in the request body makes it possible to register any e-mail, even a non-existent e-mail, without any verification code being sent or requested. In addition, no welcome mail is received if the mail exists (belongs to another person). In this way, it is possible to register other users as well as fake users. In this case, we have changed the e-mail to an example e-mail: "user@gmail.com".

5.3.4 *Application D*

Application D is the only application for treating respiratory diseases that we have analyzed, as many similar applications are still under development. The sensors used are not available in Italy, and the company did not provide them to us, so the analysis is partial. However, we did identify some problems, often similar to those of other applications.

Mobile Environment
The application has well-known problems: Pinning is not implemented, so it is relatively easy to intercept the communication, code signing is also implemented with insecure mechanisms, and in general, outdated encryption algorithms are also used.

However, the two relevant elements are the data storage and the permissions required in the *AndroidManifest.xml*. This is because all medical data, including all related data such as the patient's location or frequency of medication intake, are stored in unencrypted form in the db directory of the app's memory. In addition, access credentials and identification tokens are also saved in clear-text in both the db and the configuration files (*shared_prefs*). Finally, many permissions are granted, including WRITE/READ EXTERNAL STORAGE, CAMERA, READ/WRITE ACCOUNTS, which do not seem to be related to functionality.

Therapeutic Purpose
Unfortunately, the functionality test is limited without connecting the sensors because it is impossible to access the HomePage, but there are significant problems even in the registration phase. By manipulating network requests, it is not only possible to find out whether a user is registered, but it is also possible to register for additional functions. For example, therapy for COPD is not available in all countries, but it is possible to gain access to that therapy by altering the request sent. In the same way, it is possible to register other users or fake users, as there is no registration code or authentication method (e.g. e-mail code). The only restriction is the invitation link needed to access the app in some countries, but it can be bypassed, as we have just said. Again, there is no maximum number of attempts and logout is not forced. However, the token system is implemented correctly via JWT and could be taken as an example if all ids, hashes and even tokens were not stored in memory.

Machine Learning (ML) & Artificial Intelligence (AI)
Undoubtedly, the lack of authentication can be a problem in this respect, but apart from that, we cannot say much more as we have not been able to test the entries.

5.3.5 *Application E*

The analysis of application E is particular. It is the only Web and not Mobile application included in the analysis sample. Unfortunately, this app also requires a prescription code which was not provided to us.

However, we decided to look at it, mainly to see if it had any common vulnerabilities with the other apps. A similar element is the interception of network traffic, which is even more accessible for a web application than for a mobile one.

This app is interesting because it can be considered an excellent example of how registration, login, requests and password resets can be implemented in such a way that makes it impossible to find out whether a user is registered or not. Of course, this should be the rule for all DTx but for this one in particular, as it deals with depressive disorder, which is often heavily stigmatized. The only downside is that, on the other hand, it is possible to know the format of the access code because there is an example. Moreover, it is not very long, so it could be possible to bruteforce it and register with someone else's code.

5.3.6 *Application F*

The application F is a fairly popular app to help people suffering from anxiety and stress. Although it may seem less medical as it does not monitor or prescribe medication, it is certainly therapeutically sensitive as it interacts with very fragile people through a community and CBT pathways. Therefore, an alteration in therapy may be less evident but certainly dangerous, as it could aggravate an already unstable and less controlled situation. Unfortunately, there are many vulnerabilities in this application, including modifying the activities of other users.

Mobile Environment
The application has a mobile version and a web version, and for both versions, it is pretty easy to intercept network traffic, as Pinning is not implemented. The mobile application's code is easy to decompile and analyze and is signed with both insecure mechanisms and more up-to-date schemes.

However, the most severe vulnerability concerns the storage of data in memory: in the app's memory, not only the patient's login credentials, i.e. **username and password**, but also the request authentication tokens are stored in clear-text. The file with the credentials in clear-text is shown in Figure 5.4. Thus, if the phone is compromised or stolen, it would be effortless to access another patient's app and send requests in their name. Finally, as almost all the apps analyzed, outdated algorithms and encryption libraries are used, and some granted permissions appear to be unrelated to the app's functions (READ/WRITE EXTERNAL STORAGE, BODY SENSORS, LOCATION).

Therapeutic Purpose
The most severe vulnerabilities concern the exposure of sensitive data and the functionality of the app itself.

First, one of the app's two main features is the Community, where patients can exchange messages and progress. Unfortunately, there is no authentication factor for registering with the app, so anyone with a fake identity can register and see users' posts. By the way, the Community posts can be private (no one but the user can see them) or public (all users can see them). It is also straightforward to check whether a specific e-mail address is registered or not, either by registration, login (on the web app, not on the mobile app), password reset or network requests. It is in general confidential information, but in the case of behavioural disorder treatments, even more. A patient might be discriminated against due to his/her registration to the app.

However, there are even more severe authentication problems. Each request is authenticated with a token consisting of *id_user* | *token* which remains valid for one year. No logout is forced when the app is closed, and the password reset link is not invalidated once used. Anyone who can retrieve the link, even after months (for instance, from the patient's e-mail), can change the password and enter the patient's profile.

In addition, the user ids are sequential, and the calls to the APIs are partially public: any registered user can make a call to API by changing the request id and access additional information about other users. It is allowed that users can search for each other within the app and view the progress of others if the patient has set his profile as public. However,

with these requests, information is accessed that is not present in the profile (date of enrolment, type of subscription, account settings, ids of programs, and users followed or who follow him). Furthermore, this information can be gathered through these calls even if the patient has set his profile as private.

However, the most severe vulnerability is not the fact that the authentication of the individual patient is very weak, but the fact that the authentication token and the request parameters are not checked simultaneously. It is enough to have a valid authentication token, registering with a fake e-mail to the app since the e-mail is not checked, and it is possible to send changes to other users' posts. Each resource (users, posts, challenges, comments) is identified with sequential ids, which are easy to discover or guess. For example, by searching for a user with a public profile, it is easy to find his id and the ids of his posts. If the profile is private, it is possible to proceed by attempts as the ids are sequential and simple to guess. At this point, by sending a request with any valid authentication token but with the id of another user's post, it is possible to change both the visibility and the content (text) of the post. An example of the attack and the modified post are reported in the Figures 5.5 , 5.6, 5.7. This is severe because writing in the name of another patient can ruin the patient's and company's reputation. A patient already suffering from anxiety or panic attacks might even feel harassed or attacked if he sees his posts changed.

Finally, it is also possible to manipulate requests in such a way as to delete own comments, although this would not be a feature. Again, this is serious because a fake user could comment heavily or be discriminatory on someone else's posts and then delete the proofs.

Machine Learning (ML) & Artificial Intelligence (AI)

Rather than ml algorithms to estimate medical values (drug quantity), CBT modules designed to help the patient are interesting. These are games, small challenges or tasks that the user does every day and are often summarised in a post on his/her profile. Therefore, all the vulnerabilities pointed out in the previous Section, including those concerning patient authentication and the possibility to view and modify the activities of others easily, can also alter modules and paths designed to improve patients' anxiety and stress levels.

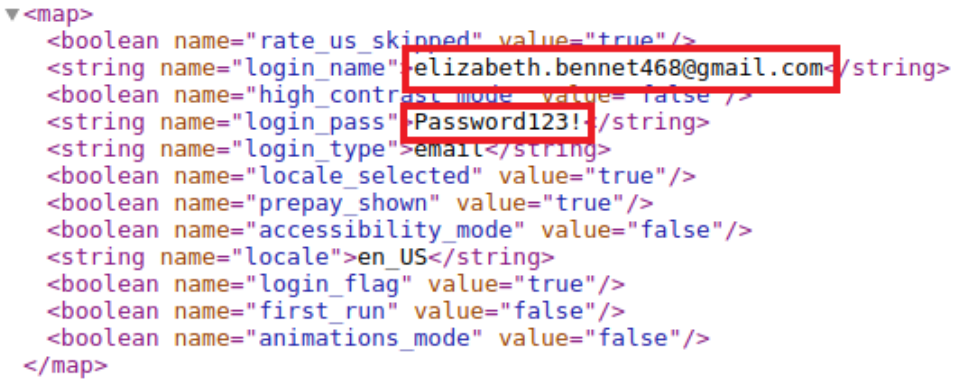

Figure 5.4: The user's access credentials are stored in clear-text in the app's private memory and so are not immediately accessible. However, if the phone has already been compromised or is stolen, the user's credentials can easily be accessed.

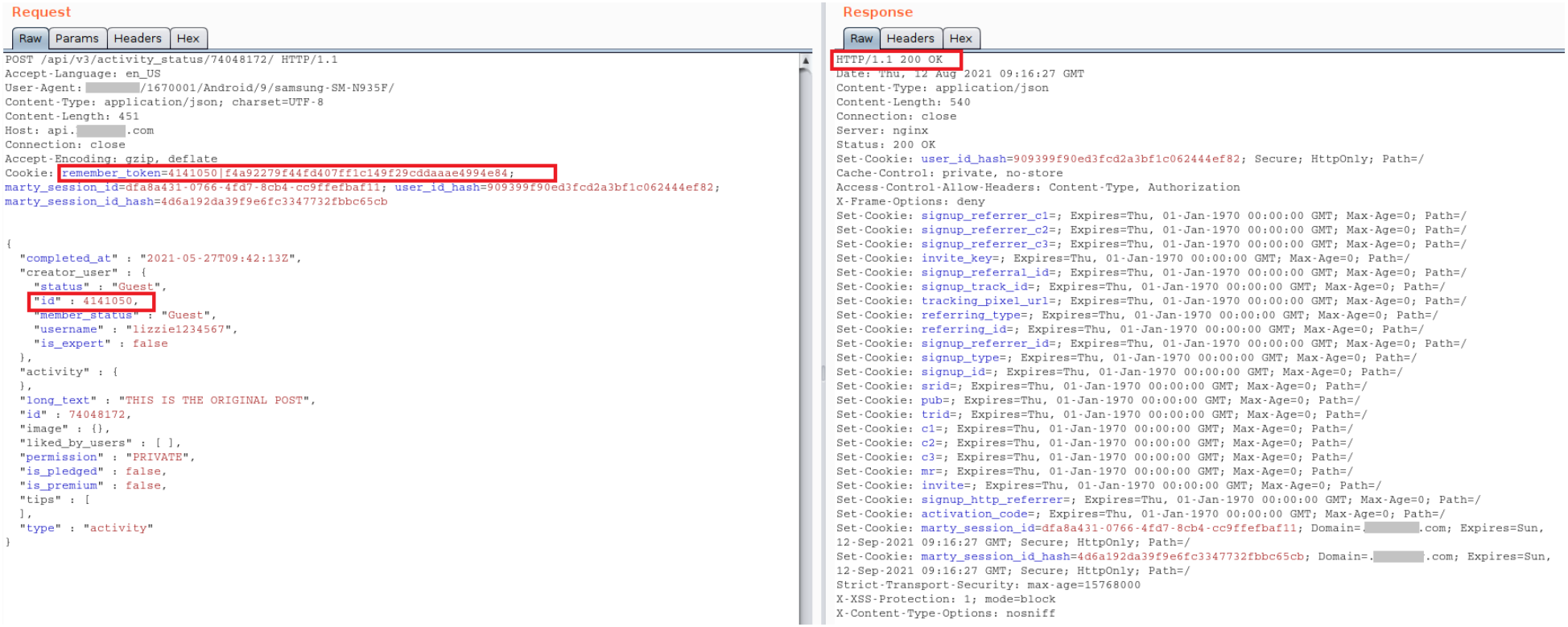

Figure 5.5: POST request for activity from user 4141050. The id to authenticate the user is used in the remember token and in the request body match.

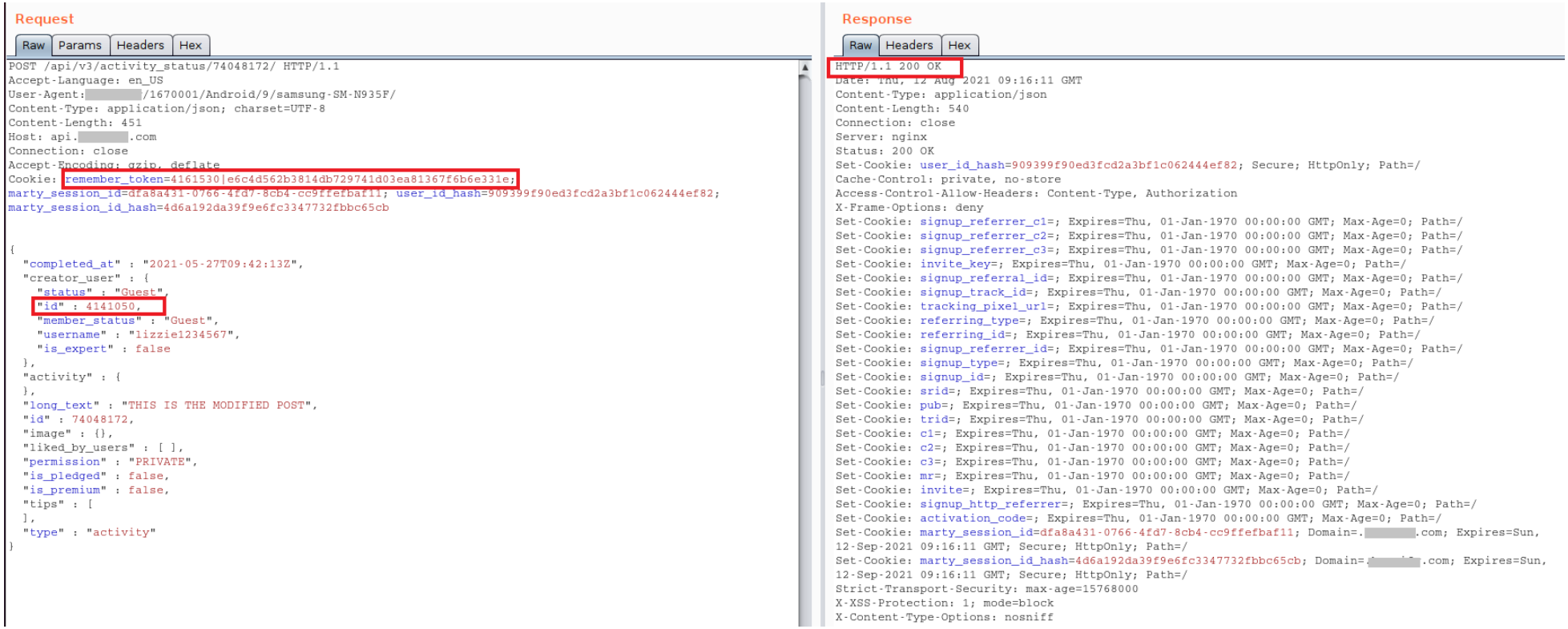

Figure 5.6: POST request for malicious activity by user 4161530. The authentication token is of user 4161530, but the user id in the request body is 414105. Thus the activity with the text written by user 4161530 will appear on user 414105's profile, as shown in the Figures below.

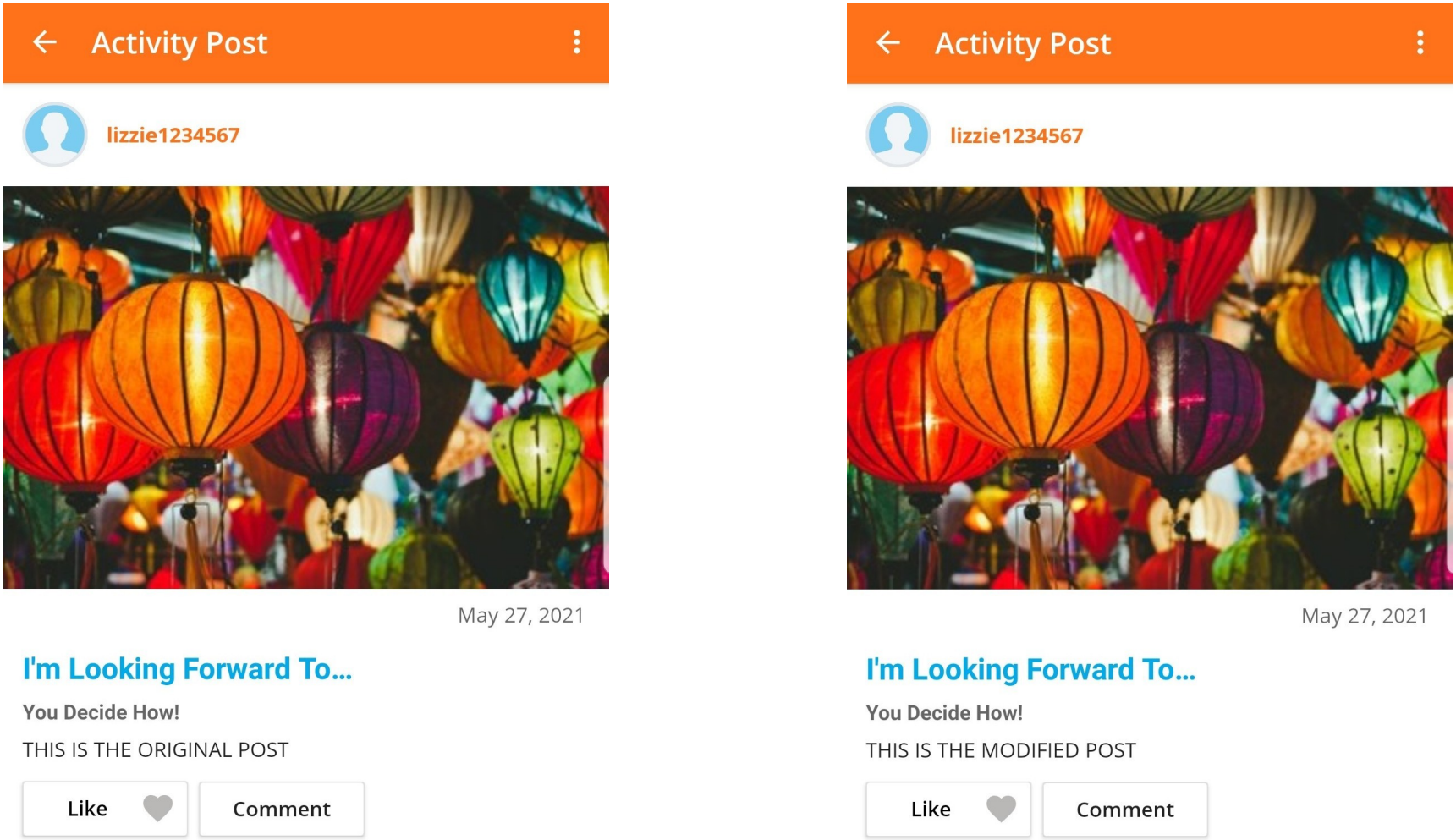

Figure 5.7: Profile of user 4141050. On the left, the activity has the text sent by the user ("THIS IS THE ORIGINAL POST"), on the right, it has the text sent by user 4161530 ("THIS IS THE MODIFIED POST"). The request is considered valid and the activity is modified even if the id in the authentication token does not match the id of the activity owner.

5.3.7 *Application G*

Application G is similar to application F because they have similar features, but it is, if possible, even more sensitive. It aims to modify and improve the patient's relationship with alcohol and also helping to manage dependence. This type of disorder is generally very stigmatized and judged, and it is therefore imperative that a patient feels that their data is kept private, but unfortunately, we have found that this is not the case.

Mobile Environment
Concerning the mobile application, the problems are similar to those presented by the other applications. For example, it is elementary to obtain the source code containing details about the app's functionality, and Certificate Pinning is not implemented. So, it is sufficient to install a trusted certificate to intercept HTTPS traffic. Moreover, some logs are saved in clear-text in the app memory, and not all code signing methods are up to date. It should also be noted that, as with many other apps, the username, id and especially the user recognition token are stored in clear-text in the phone's memory.

Therapeutic Purpose
As with application F, the vulnerabilities of this application mainly concern data and therapy management.

First, the registration e-mail is neither checked nor verified, making it very easy to register fake users. This is problematic because one of the main functionalities is the Community, where the user can share his or her moods and thoughts. This feature is inspired by Alcoholics Anonymous metho, where there is physical control of the participants. Without control over the e-mail, a user can register several times or register dummy e-mails to comment and interact in malicious ways with other users.

Furthermore, it is easy through the registration function to find out whether an e-mail is registered or not. This violates privacy as a patient uses a username to share posts with other users and may not like it to be traced back to their personal e-mail. Finally, the password requirements are very weak (6 characters). The logout is not forced. The link to reset the password only expires when a new one is requested, all of which makes it possible to access another user's account. However, as with the F app, the most severe problems concern users' activities and the possibility of manipulating them. The management of tokens is correct, as JWT tokens are used, which expire at reasonable times (after logout or after a day). However, also in this app, sequential ids are used to identify all resources: users, posts, experiments, comments and messages. This, added to the fact that the authentication token and the request parameters are not checked simultaneously, makes it possible to modify other users' posts and their progress and reminders, influencing their treatment path in a significant way.

The process is not as straightforward as for the F app because it is necessary to collect various information such as the user id and, above all, the id of the comment or experiment. To retrieve the first one is enough to open the comment, while the second one is not public. However, knowing that each category of experiments has the same id and each experiment has a sequential id, it is possible to proceed by attempts. It could also be possible to automate the task. Therefore, with relative effort, an attacker could

edit a patient's progress, feedback and reminders. Furthermore, it is also possible to write messages on behalf of another user to the application team. This requires more effort than altering posts because there is an additional parameter to guess. However, it is a severe breach of privacy because an attacker could send inconsistent information or requests for immediate help. A very immediate operation, instead, is to edit or delete both own posts and comments. Theoretically, the app does not provide this feature, but by simply changing the HTTP request method to DELETE, it is effortless to delete own content.

Finally, it is also straightforward to unlock the paid features, available only to Australian patients, by registering as an Australian citizen. No verification is done, and access to all the app features is granted for unlimited time.

Machine Learning (ML) & Artificial Intelligence (AI)
The discussion on algorithms is linked to the discourse on therapy; rather than poisoning the algorithms, it is possible to alter the therapies of others directly.

5.3.8 *Application H*

Application H is perhaps the least therapeutically sensitive of all. It is aimed at managing chronic back pain through daily exercises by a coach and training videos. However, no drugs are prescribed, and no sensitive content is shared. However, it is interesting because it is the only app with an access code whose company provided us with the code, and it can be considered both a good and a bad example on different sides.

Mobile Environment
The weak points of the mobile app are those common to other apps: Certificate Pinning is not implemented, the code is easy to decompile and analyze, and it is also signed with an outdated method. Moreover, some suspicious permissions are granted that seemed unrelated to app features (RECORD AUDIO, LOCATION, CAMERA). However, it is interesting to note that although the log files are stored in clear-text in memory and some DB files, they do not contain particularly relevant information. Especially the main DB of the app, where we imagine the most sensitive data is contained, is encrypted. All apps should adopt this security measure.

Therapeutic Purpose
Regarding the management of therapy and sensitive data, there are some vulnerabilities but also some positive aspects.

Neither from login, registration, or password reset is it possible to determine whether a user is registered or not, but by sending a modified request, it is easy. The management of authentication tokens and password reset is also correct. The formers expire within a reasonable time and are checked simultaneously as the request parameters are sent (it is impossible to use any token to modify someone else's post). The latter expire after use.

However, there are two critical problems regarding registration and data management. Manipulating a registration request allows to bypass the app's access code and register unauthorized users. It is necessary to have a sample request, but it is not complicated to modify it. This is significant because one of the major security measures is actually ineffective, and anyone can be registered. Moreover, despite being mentioned in the privacy policy, there is a really invasive tracking service that collects many data per minute. The user should be better informed. Finally, the user can change the e-mail and data in the settings without them being rechecked.

Machine Learning (ML) & Artificial Intelligence (AI)
This aspect has not been explored in-depth as there are not many inputs that the user can enter or modify.

5.3.9 *Application I*

The application I helps cancer patients better manage their treatment and symptoms. The doctor registers the patient, and an access code is sent to him. Unfortunately, the company did not provide us with this code, so the analysis is not very thorough.

However, the analysis of this application is relevant as it can be an example for defining guidelines. The access code is actual and cannot be bypassed. Therefore, there is no way to know if a specific e-mail is registered even by manipulating network requests. Furthermore, network packets cannot be easily intercepted because Certificate Pinning is correctly implemented.

Still, static analysis of the code has reveal some incorrectness: old versions of some libraries are used (SQL, OpenSSL) and some permissions seem unrelated to the functionality of the app (READ/WRITE EXTERNAL STORAGE)

5.3.10 *Application L*

App L is an app for the management of Substance Use Disorder (SUD). This app, like G, handles very sensitive data as there is a lot of prejudice and discrimination regarding these disorders. The patient is registered by his doctor and receives an access code not provided to us.

We could still detect some problems, especially on the mobile app. The code is easy to decompile and contains relevant information regarding the formats of parameters and requests. It is only signed with the v1 schema, which is vulnerable for versions of Android before 7, and old versions of certain libraries (OpenSSL and URL-parse) are used. Pinning is also not implemented, so only by installing a certificate the traffic can be intercepted. Finally, by analyzing the memory, we find some DB in clear-text that from the structure seem to contain sensitive data, but not having been able to use the app, we could not verify this issue.

However, a good practice for the privacy of registered users is that there is no way to know if a specific e-mail corresponds to a registered user. Login and registration error messages, responses to sign-up requests and password reset notifications never indicate the presence of an e-mail and should be taken as examples by other companies.

5.4 RESULTS DISCUSSION - GLOBAL

Based on the results obtained and the analysis of each application, we make more general considerations regarding the current security level of the Digital Therapeutics (DTx) products. In particular, we report our observations divided into the 3 macro-categories defined in the modelling phase (Section 3.2.1), and we use histograms to make the results more immediate to understand.

5.4.1 *Mobile Environment*

As explained by Casalicchio et al. in [22] and as also demonstrated in several previous studies (2.2), the mobile environment is extremely vulnerable. In particular, the mobile device is outside the company's domain and is managed by a user who may or may not be familiar with the technology. In general, it is difficult for the user to realize whether or not the phone is already compromised (rooted) or infected with a virus. On the contrary, the user himself could, for many reasons, install malicious applications on his phone. Each application has its own memory space, but this security measure is useless in a rooted phone. In addition, the mobile phone is physically exposed to various threats: it is easy to steal, damage, and handle. An attacker who has access to the phone, even for a short time, could obtain precious information. For this reason, the mobile environment should be considered as inherently insecure for the development of applications, even more, if they are therapeutic applications.

However, it seems that developers of DTx have not yet sufficiently embraced this assumption. The mobile environment is the category in which there is the most significant number of bugs and vulnerabilities, mainly regarding the memory storage and network settings.

Almost all the analyzed applications store sensitive information in clear-text, be it medical data, patient credentials, or both. Only one application (H) stores data in an encrypted DB. About half of the applications save logs files in the app's memory and one in a file accessible via *logcat* (a function that does not require rooting the phone).

Regarding the network protocol, although the use of HTTP is outperformed compared to the results reported by Aliasgari, Black, and Yadav in [12], only 3 out of 10 apps implement Certificate Pinning. This means that by installing a custom certificate on the phone, it is possible to intercept all HTTPS traffic and to manipulate the network packages. Rooting a phone takes some time, while installing this certificate can be done quickly and even automatically by a bad app.

Moreover, there are several minor problems, but they can become more severe when combined with the above problems. No company obfuscates the code, so it is straightforward to access and inspect the application code and possibly overwrite parts of it. However, all mobile applications use outdated algorithms or encryption methods, and at least three out of ten applications use deprecated versions of some important libraries (SQLite, OpenSSL). Also, 8 out of 10 applications have permissions that seem unrelated to the app's functionality and could be used for other purposes, such as reading or writing to the phone memory, accessing the contact list, or recording audio and video.

Finally, 8 out of 10 apps use weak encryption mechanisms, even if combined with more secure ones. However, this could make it possible to modify the code and distribute an altered version of the app. In particular, these problems are relevant not so much for their gravity but because it would be simple to avoid them. There is no need to invent new encryption algorithms or libraries. Just use the secure ones already in circulation and the latest version of the libraries.

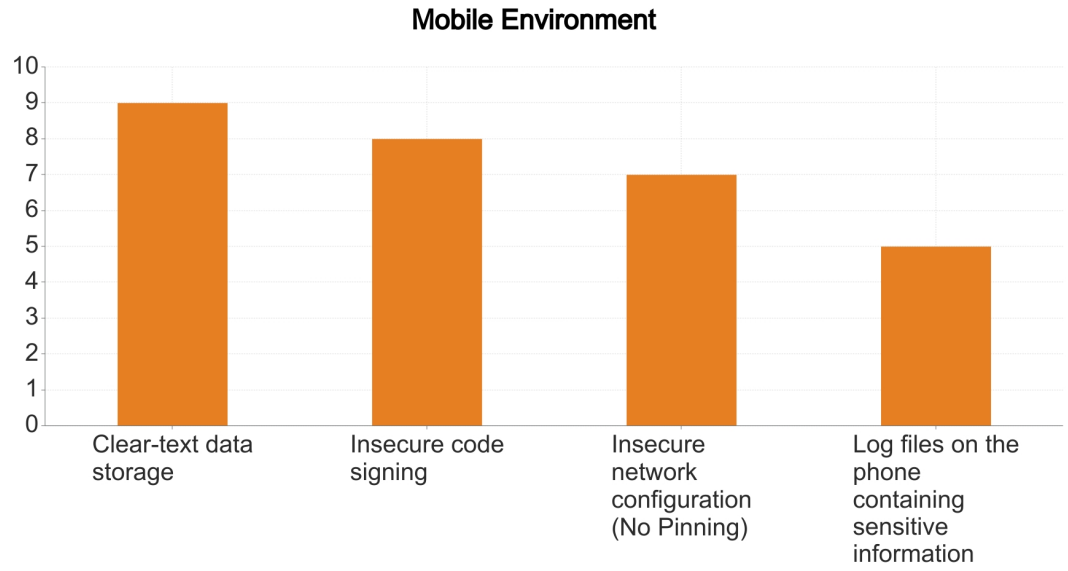

Figure 5.8: Summary histogram of relevant vulnerabilities found in the Mobile Environment category.

The minor problems regarding this category were pretty predictable because they are common in mobile applications. However, these are inaccuracies can be easily avoided, and the already widespread good practice for mobile programming, in general, suggests avoiding them.

In contrast, the problems with data storage and network configuration are pretty surprising. Avoiding these vulnerabilities do not require much effort, whereas exploiting these vulnerabilities can be severe, especially for the type of product proposed. Given the sensitivity of the data handled, we expect great attention to be paid to their storage and communication protocols, while often the minimum effort to use a library to encrypt DB 's or to implement Certificate Pinning is missing. However, both these measures can be implemented using external libraries and established procedures.

5.4.2 *Therapeutic Purpose*

The most interesting and relevant feature of DTx is undoubtedly to help the patient in the management or treatment of a disease, which can be diabetes as well as substance addiction. This means that the data managed are extremely sensitive and that their alteration could also have consequences on the health of the patient himself and not only on his privacy.

Again, we detect a general lack of attention and several severe issues. Some data exposures could be avoided with little effort, while the most severe vulnerabilities are logical or implementation errors. From the Table of results, it may seem that this category presents fewer problems than the others, but we have limited access to half of the apps. The use of access codes and prescriptions is undoubtedly a good security measure against possible attacks, as also explained in Section 6, but we cannot exclude the possibility that, once this vital barrier has been overcome, many vulnerabilities may still be found. Therefore, it would be very interesting to extend our study with total access to all applications (7).

Still, in each of the apps analyzed, we find significant malfunctions. In particular, none of the analyzed apps has an effective authentication factor, which allows both the registration of other people and fake users. For example, registering another person's e-mail to an app for diabetes treatment or addiction management can damage their reputation, while registering fake users can impact the algorithms used to create statistics and customization. It is also harmful to the user's reputation and privacy that it is possible to know whether a user is registered or not, as having a chronic illness or addiction is often grounds for discrimination. Unfortunately, this is not possible for 4 out of 10 applications, while for all the others, it is possible and often very easy.

The major problems, however, concern access to and manipulation of other users' data. Specifically, there are problems in the authentication phase: none of the apps analyzed has set a maximum number of attempts for the login, 3 apps do not impose any password requirements, and for 3 apps the password reset link does not expire. All these errors can be exploited to gain access in the name of another user.

More serious is gaining access to the data of another patient without hacking into his account but using the attacker's one. 2 similar applications have the same logical error: the authentication token and the request parameters are not checked together, allowing requests to be sent with one's valid authentication token but with someone else parameters. This makes it possible, for example, to edit another user's posts or activities and influence their reputation and treatment. In one application, we found an SQL injection. This well-known vulnerability can be exploited to access all the company's DBs , i.e. the data of all the patients.

Finally, it is important to note that all applications have a clear and comprehensive privacy policy, unlike the one found in 2015 by Knorr, Aspinall, and Wolters in [9], and this indeed ensures a higher degree of user awareness. However, the tracking and analysis services used are very intrusive, and very much information is sent per minute. Therefore, the users must be aware that their data is being shared with third parties and know precisely which third parties are involved and how much data is shared.

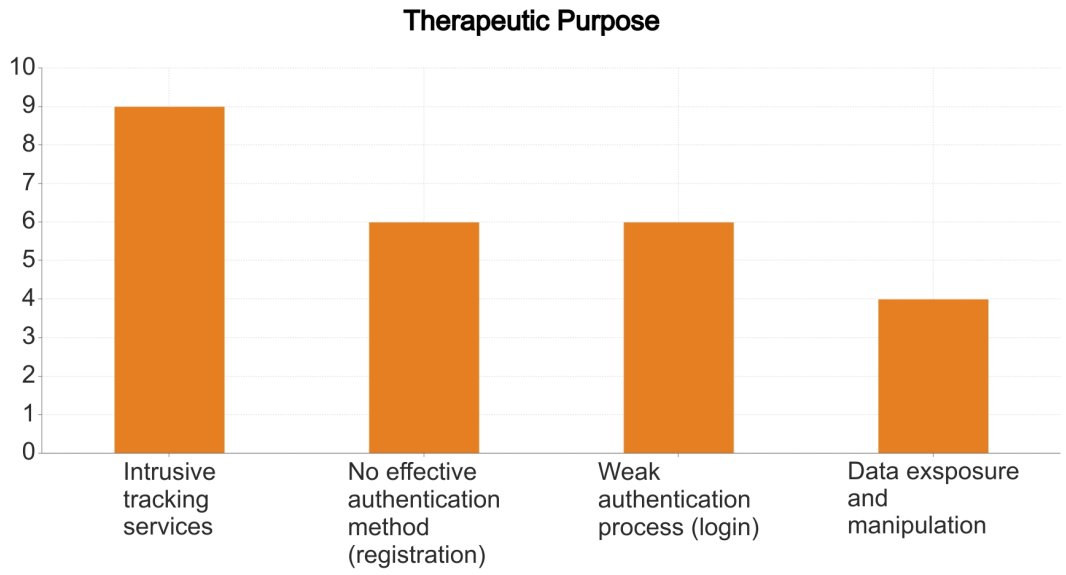

Figure 5.9: Summary histogram of relevant vulnerabilities found in the Therapeutics Purpose category. For this category, it is important to remember that for 3 applications (A, I, L) we had limited access which did not allow us to test certain functions.

The results for this category are pretty surprising for two reasons: on the one hand, some bugs require minimal effort to be corrected (e.g. the error message during login or registration) but can have severe consequences for the patient's privacy and reputation. On the other hand, even on a sample of only 10 applications, there is a significant number of information leakage and data tampering vulnerabilities, even severe ones. This is entirely unexpected for the way Digital Therapeutics (DTx) are presented and for the sensitivity of the data managed.

5.4.3 *Machine Learning (ML) & Artificial Intelligence (AI)*

This category is undoubtedly significant and emblematic for dt, as a major innovation is to provide the potential of Machine Learning (ML) and Artificial Intelligence (AI) to improve the effectiveness of proposed therapies and to better tailor them to the patient. However, it is the most sacrificed from the testing point of view as it would require a more in-depth and detailed study of the algorithms.

Still, it is relevant that it is possible to register fake users for practically all the apps we have access to. This could allow a data poisoning attack: by registering a large number of fictitious users and entering targeted data and values, the statistics and classifications operated by these algorithms could be altered.

The threshold and reference values question is also interesting: setting predefined ranges and inputs prevents inserting random values. Indeed, it provides valuable details about the values expected by the algorithms. With more information on how these algorithms work, it would be possible to discuss this further 7.2.

6 GOOD PRACTICES AND GUIDELINES

In this Chapter, we propose some guidelines and good practices for DTx development, starting from both theoretical analysis and testing results.

By analyzing the applications, we notice that there are not only some specific issues related to their features but also some general bad practices that are pretty common. Of course, these shortcomings should always be avoided to guarantee the security of a mobile application, but we do not expect to find them in these apps. They can lead to more severe vulnerabilities in the case of DTx. Also, some of these errors could have been avoided by following the good practices proposed by the DTA, however general they may be [26]. Therefore, we define guidelines and good practices, both general and specific, but concrete and significant to the cybersecurity of DTx in particular.

6.1 MOBILE ENVIRONMENT

In this category, there is the highest number of problems and especially many unexpected minor issues. In particular, as Casalicchio et al. also argue in [22], it is crucial to consider the inherent vulnerabilities of the mobile environment and to assume that it is an untrusted environment in order to take all the necessary security measures during the development of a mobile application.

This consideration applies to apps in general, as also explained in the Section 2.2, but even more for Digital Therapeutics (DTx). The mobile app is the user's first interface with the therapy. For this reason, the data and features managed are much more delicate than those of a usual app and require all the necessary cybersecurity measures.

However, the analysis of the results shows that no particular cybersecurity measures are used, and often not even the best practices of mobile programming are respected. In Table 6.1 there is a list of guidelines to follow when developing a mobile app and why they are necessary and relevant to DTx in particular.

Mobile Environment	
Guideline	**Motivation & Comments**
1. Protect the code from manipulation **1.1** Use a secure and up-to-date signature method **1.2** Use the most up-to-date and secure versions of the libraries used, especially the encryption libraries **1.3** Use robust, up-to-date encryption mechanisms **1.4** Obfuscate or encrypt the most sensitive parts of the code **1.5** Avoid hardcoding sensitive information in the code	• These good pratices are valid for mobile application development in general, but even more so for DTx. • A vulnerable signature method allows an altered version of the app to be disseminated, accessible without an access code or with some functionality unlocked/modified. • An unobfuscated code allows the app's functioning to be understood (e.g. request formats or encryption algorithms) and, combined with an incorrect network configuration, allows malicious requests to be sent.
2. Use a communication protocol secure and encrypted **2.1** Use always HTTPS **2.2** Implement Certificate Pinning correctly	• Fortunately, almost all applications always use HTTPS and this is a concrete improvement on the situation with mHealth [12]. • However, almost no apps implement Certificate Pinning, and this is significant because it makes the possibility of MITM attacks much more real. It is enough to install a trusted certificate to be able to intercept and manipulate network traffic. This could be already installed on the phone or added with a few minutes' access to the device, and it is certainly easier to access someone's phone than a PC.

Mobile Environment	
Guideline	**Motivation & Comments**
3. Store necessary data only in the app's private memory and always encrypted **3.1** Avoid saving sensitive data (passwords and identifiers) in clear-text **3.2** Avoid saving sensitive data in log files **3.3** Avoid saving any kind of data in the phone's public memory, or at least inform the user. **3.4** Encrypt databases or at least those containing medical data	• These are also pretty standard measures common to all mobile applications but are particularly important in the case of DTx. • In the case of medical databases containing very sensitive data such as PHI, encryption should be the norm, whereas it is currently the exception (1 in 10 apps reviewed, app H). • It is important to keep in mind that the mobile environment is not trusted and that the application's private memory might not be so private in case the phone is already corrupted or stolen. • An additional aggravating factor is that data encryption is guaranteed in many privacy policies even if it is not actually implemented.

Table 6.1: Guidelines - Mobile Environment

6.2 THERAPEUTIC PURPOSE

In this category, we find the most significant results. Although there are fewer vulnerabilities than in the Mobile Environment category, they are pretty severe, both from a therapeutic point of view but mainly from a reliability point of view. If the patient perceives the application as unreliable, even if only because his privacy is violated rather than his safety threatened, he will undoubtedly be reluctant to use the app, and this could halt the incredible development of DTx.

In addition, it must be borne in mind that the use of sensitive or medical data may also have more or less severe consequences for the patient's health. Moreover, some problems can be easily fixed, while others are known vulnerabilities or underlying logical problems. The fact that they are present in DTx apps is unexpected because they are relatively new or recently updated products, which offer innovative therapy approaches from a medical point of view, but not from a cybersecurity one.

Therapeutic Purpose	
Guideline	**Motivation & Comments**
4. Check that only authorised users have access to the therapy and verify their identity **4.1** Use an effective second authentication factor **4.2** Access applications only with an access or prescription code **4.3** Check effectively user requirements (nationality, age, subscription) **4.4** Check user data (email and phone number) even when they are changed in settings **4.5** Avoid inserting functionality that is still under development in the code	• It is essential that only authorised users have access to the application and its features, and that their identity is effectively controlled in some way. Accessing therapy or certain functions without authorization can be risky for one's own health but also for that of others, especially if there is interaction between users (community). • If anyone (even a person who does not exist) can access the app, this can have severe consequences for the reliability of the therapy. On the one hand, it is possible to obtain data on patients and even influence them with malicious comments. On the other hand, it is possible to alter the statistics and input data of the algorithms. • An access code provided after registration of the user by the doctor (as in the I and L app) is a good security measure and ensures that only authorised and identified patients access the DTx.

Therapeutic Purpose	
Guideline	**Motivation & Comments**
5. Control properly the user authentication process **5.1** Set sufficient password requirements **5.2** Impose a maximum number of login attempts **5.3** Force automatic logout **5.4** Correctly implement password reset (link must expire after use)	• These good practices are the basis for a proper authentication process, but again they must also be respected in the case of DTx. • In particular, forcing a logout or inserting a maximum number of login attempts may be detrimental to the app's usability. However, they are essential to prevent someone from accessing someone else's app, always considering the ease with which one can physically take possession of a mobile phone. • In fact, not only would the patient's data be exposed, but it would be possible to change the patient's treatment settings, even in a small way, with significant consequences.

Therapeutic Purpose	
Guideline	**Motivation & Comments**
6. Protect patient privacy and improve awareness of data processing. **6.1** Avoid providing sensitive information in error messages (registered mail or not, wrong password, correct value or not) **6.2** Avoid providing sensitive information in network replies **6.3** Always validate inputs and parameters **6.4** Check that only authorized users can access resources **6.5** Inform users about the management of their data and clearly indicate all the analysis and tracking services used **6.6** Inform the user if there is a data breach	• Patient privacy is one of the most important assets of these apps, mainly as medical data (phi) are managed, and all security aspects must be respected. • Even just knowing that a user is registered with the dtx and therefore suffers from a specific pathology is sensitive information, especially in the case of conditions that are often stigmatized (mental disorders or addiction). App I is the only one that preserves this information. • Input validation is fundamental to protect against known attacks (SQL injection/XSS) and avoid data poisoning. Therefore, it was surprising to find SQL injection (which allows access to all users' data) and some public resources, including users' data. Even more surprising is that the vulnerability was fixed, but the users were not informed at all. • All DTx products have pretty clear and public privacy policies. However, often the use of analytic services or sharing of data with third parties is indicated generically. It should be listed more specifically as there are often many different and very intrusive services.

Therapeutic Purpose	
Guideline	**Motivation & Comments**
7. Avoiding manipulation of other users' data **7.1** Check that all steps to perform a certain operation are consistent with each other **7.2** Check that authentication tokens and request parameters are consistent **7.3** Avoid the use of easily guessable identifiers (sequential ids or those containing e-mails or usernames)	• These types of problems are the most difficult to find and fix, but they are also among the most serious. • They require a relatively thorough manual analysis, and even from the point of view of countermeasures, it is not enough to fix a function and update a library. However, they require a logical and in-depth analysis of the system. • It is essential to try to find them, or at least to take all possible precautions (e.g. avoid guessable ids) to prevent users from seeing their data or activities modified, damaging not only their therapy but also the reputation of DTx in general.

Table 6.2: Guidelines - Therapeutic Purpose

6.3 Machine Learning (ML) & Artificial Intelligence (AI)

As explained in 7.1, we are not able to carry out an in-depth analysis of ML and AI algorithms and tools. So we can not define precise and exhaustive guidelines for this category. However, even with limited analysis, we deduce some important indications to follow.

Machine Learning & Artificial Intelligence	
Guideline	**Motivation & Comments**
8. Prevent 'silent' registration of fake users **8.1** Implement effective authentication methods	• The recording of a substantial number of invented profiles in which data can be manipulated at the attacker's will could influence statistics and analyses of patient data and evolve into a data poisoning attack.
9. Avoid providing detailed information about expected inputs and algorithm operation **9.1** Avoid detailed error messages and technical explanations regarding the operation of ML and AI services	• This guideline is challenging for developers because, on the one hand, providing much information about threshold and reference values can be harmful. However, on the other hand, using pre-set inputs and values protects against malicious input.

Table 6.3: Guidelines - ML & AI

6.4 DISCUSSION OF GUIDELINES

We have tried to define good practices as more concrete and suitable for DTx than those currently available. In particular, from the results obtained, we realized that even the guidelines currently proposed for DTx ([26]) are not really considered and that many good practices in mobile programming, in general, are undervalued. So we have taken them into account but highlighting the importance they assume in a product such as DTx.

We have also defined other guidelines that are more specific to DTx and linked to their specific characteristics, especially in the *Therapeutic Purpose* macro-category 6.2. Again, we based these good practices on the vulnerabilities found in the apps and the potential threats. In both cases, however, we have defined a detailed and concrete list of security measures at a reasonably low level to overcome the level of abstraction of the guidelines proposed so far.

This proposal could be the starting point for defining a series of official and technical requirements, which are mandatory and not just recommended for the development of DTx which we believe are necessary to make these products secure and reliable in terms of cybersecurity.

A final guideline to be established is specific training for both patients and doctors who decide to use a Digital Therapeutics (DTx). For example, no matter how well the apps are made, or if they are only available by prescription or access code, if a patient gives her code to someone else, all the security measures would be cancelled immediately. Similarly, suppose the doctor left his Dashboard exposed in his office, perhaps easily accessible by other doctors or other staff in general. In that case, there could be severe consequences for the treatment of all patients.

On the one hand, it is essential to make all possible efforts to ensure that apps meet good cybersecurity standards and that the guarantee of privacy included in privacy policies is effective and not just stated. On the other hand, it is crucial that once suitable standards have been defined, users are also informed and trained on the correct and secure use of the DTx product.

7 LIMITATIONS AND FUTURE WORK

In this Chapter, we detail some aspects which limit the current work. Furthermore, we provide some future research lines.

7.1 LIMITATIONS

Experimental results are narrowed because, in all cases, the software and the infrastructures are proprietary. The law, of course, recognizes the authors' rights to control the access to their products: some companies do not want their products to be inquired by research, and those who do, sometimes allow only for partial access. For these reasons, the sample used for the experiments is not as large as we would have liked, and the analysis of some applications is also limited.

Firstly, many DTx apps are usually accessed through a code or a medical prescription: bypassing it often means an illegal act. Nonetheless, it is of no interest to enter an application to investigate certain functionalities (app screens, for example) without a valid account because apps' features need a registered user to test them. The company's cooperation is then required to get a test code and full access to the application.

Secondly, sometimes functionalities are associated with a particular medical device that interacts with the app. Buying the associated devices presents various issues: the conspicuous cost of the products, the unavailability in Italy (where this work is being carried out), the need for a manual configuration with a doctor. When the device cannot be procured, the functionality must be ruled out from the research. For the current work, only the device for the C app is provided. However, as also explained in Chapter 4, some analyses, mainly the static and the automatic ones, can be conducted without accessing the applications. This allows extending the test sample with a few more products on which applying only these analyses.

Regarding the apps selection, there are also further problems to consider. First of all, still few applications officially possess the requirements to fall into the category of DTx. In particular, we selected only products present in the Digital Therapeutics Alliance (DTA) library [47] or the DTx's specific literature [3, 4, 21, 22]. We obtained a more extensive initial list than the one presented in Table 5.1, about twice the size of the final list. This is because different reasons make some apps unsuitable for testing. Some apps are still under development and are not available on the market. Also, some applications were launched by companies later bought by others who decided not to invest in their development. Lastly, we excluded the applications for minors that require the cooperation of parents to set up therapeutic path and also the apps that required a series of preliminary meetings with a clinician.

We overcome this problem by conducting a more in-depth analysis of the available applications, following the example of other work focused on a few applications (two or three) [11].

As explained in Chapter 5, although we could not analyze all the desired components and applications, we found several problems and vulnerabilities, both generic and specific. Nevertheless, the results are more than sufficient to conclude the current security level of Digital Therapeutics (DTx) and also to propose some guidelines for their development.

7.2 FUTURE WORKS

As explained in Chapter 2, there are still some challenges to be solved for DTx to become widespread, and our research shows that cybersecurity is an aspect that needs to be improved and standardized. This work could be a starting point for further investigation into this aspect.

On the one hand, the companies or the Digital Therapeutics Alliance (DTA), given the results obtained, could decide to collaborate and provide permission to extend this research. In this way, new applications could be analyzed and, above all, components left out of our tests. Owners of external services used by DTx could also push to extend the analysis to their components. For example, hospitals themselves might be interested in requiring that specific security requirements be met before data are integrated into their IT structure. A more complete and comprehensive analysis could raise awareness of the security level of DTx and help identify different possible solutions.

On the other hand, both the companies and the competent authorities could build on the guidelines outlined in Chapter 6 to define a more official but also more detailed cybersecurity standard to be followed in the development of these products. A comprehensive, official standard, imposed as a mandatory rather than optional requirement and updated annually, could be of enormous benefit in the long run. Moreover, investing early in the secure and reliable development of DTx could ensure its effective deployment. A possible attack or flaw, on the other hand, could profoundly affect user confidence in these products and halt their rapid spread.

8

CONCLUSIONS

With this work, we enriched the current debate about the spread of Digital Therapeutics (DTx). In particular, we analyzed the cybersecurity of these products.

To date, aspects such as therapeutic efficacy or regulatory process have been much investigated, while cybersecurity has been neglected. The Digital Therapeutics Alliance (DTA) itself includes compliance with privacy regulations and a high cybersecurity standard as key elements for the development of DTx, but does not provide precise and fitting details in the proposed best practices [26]. Casalicchio et al. in [22] also argue the need to investigate certain characteristics of DTx and especially to define a more targeted and comprehensive regulation regarding their security. For this reasons, we focused precisely on the cybersecurity and privacy aspects of these products.

First, we looked at Digital Therapeutics (DTx) from a more theoretical point of view, analyzing their particular characteristics and the security issues that could be linked to these specifics. In particular, we defined a threat model based on 3 macro-categories subdivided into the STRIDE categories. The macro-categories (Mobile Environment, Therapeutic Purpose, Machine Learning (ML) & Artificial Intelligence (AI)) refer to the peculiarities of the DTx. At the same time, the STRIDE categories are handy to have a more concrete idea of the vulnerabilities and, above all, to formalize our model.

We then selected a significant sample of DTx products and analyzed them in-depth, using different analysis methods: static, dynamic and automatic. We tested the threat model on each application to get a complete and global idea of the security level of the DTx and to understand which categories are the most vulnerable.

Lastly, based on both the theoretical observations and the results obtained, we defined guidelines and good practices to follow for the DTx development, explaining why they are important.

The results showed that DTx products have several problems in terms of cybersecurity. Some issues are more generic and typical of mobile apps but take on a different relevance in products such as DTx. In particular, vulnerabilities in the Mobile Environment macro-category, such as insecure network configurations or the clear storage of sensitive data in memory, suggest that the mobile environment is not considered as unreliable as it should be. Therefore, not enough effective security measures are taken.

Other vulnerabilities are more specific to the functionality of DTx and especially to their therapeutic function. In fact, in the Therapeutic Purpose macro-category, we found the most relevant problems, mainly linked to the improper management of medical data and, in general, of app features. In almost all the apps analyzed, we encountered problems with the exposure of sensitive data. Through error messages, manipulation of network requests, exposure of resources and more specific attacks on the company database, it is possible in many cases to obtain information about other users or the

application algorithms. Moreover, in at least two apps, we discovered logical mismatches that allow us to perform privileged operations and also modify other users' profiles or activities.

Finally, the user identity is rarely properly checked. This inattention can have important consequences from both a therapeutic and a technical point of view. On the one hand, it means that unauthorized or even non-existent users (fake users) can access therapy and often patients' Community. On the other hand, several details regarding the features of the apps combined with the possibility of registering a large number of unauthenticated users could be the first step to carry out a data poisoning attack against the algorithms of ML and AI used to improve the proposed therapies.

In general, the level of security revealed by our analyses does not correspond to what we expected to find. The Digital Therapeutics (DTx) products undoubtedly offer important therapeutic innovations, and the Covid-19 pandemic has highlighted their immense potential. However, they do not meet the expected cybersecurity requirements yet. On the one hand, they present minor problems common to mobile apps, which we would not expect to find in a relatively new product, presented as very secure and updated by the companies [1]. However, on the other hand, they present more specific issues related to their features that can jeopardize patients' privacy and the effectiveness of therapy. Finally, it would be better to test and analyze some aspects and components more in-depth before DTx become widely available.

If the aim is to use DTx as a therapeutic tool for years to come, their cybersecurity standard must rise. A data breach or an attack on the application functionalities could have serious consequences not only for the patient's safety but also for the level of trust with which DTx and Digital Health products in general are and will be perceived in the future. More technical and specific cybersecurity requirements should be included in product approval processes to make them more reliable and ensure their correct development. In this thesis, we tried to propose draft guidelines that could be the starting point for a more official, comprehensive and detailed regulation for Digital Therapeutics (DTx).

BIBLIOGRAPHY

[1] *Home - Digital Therapeutics Alliance*. URL: https://dtxalliance.org/ (visited on 05/08/2021) (cit. on pp. vii, 1, 6, 84).

[2] Giuseppe Recchia, Daniela Maria Capuano, Neeraj Mistri, and Roberto Verna. "Digital Therapeutics-What they are, what they will be." en. In: *Act Scie Medic* 4.3 (Feb. 2020), pp. 01–09. ISSN: 25820931 (cit. on pp. vii, 1, 11).

[3] *Digital Therapeutics Market - Global Forecast to 2026 | MarketsandMarkets*. URL: https://www.marketsandmarkets.com/Market-Reports/digital-therapeutics-market-51646724.html (visited on 11/07/2021) (cit. on pp. vii, 1, 33, 81).

[4] Anita Caracciolo. "DTx and Costs." it. In: (Aug. 2020) (cit. on pp. vii, 1, 6, 33, 81).

[5] Gaby Galvin. *COVID-19 Accelerated Use of Digital Therapeutics, but Coverage Issues and Regulatory Questions Could Slow Their Momentum*. en. June 2021. URL: https://morningconsult.com/2021/06/15/digital-therapeutics-use-coverage-polling/ (visited on 06/08/2021) (cit. on pp. vii, 1, 6).

[6] Tom Guthrie. *How COVID-19 Will Accelerate A Digital Therapeutics Revolution*. URL: https://www.forbes.com/sites/columbiabusinessschool/2020/04/21/how-covid-19-will-accelerate-a-digital-therapeutics-revolution/ (visited on 07/08/2021) (cit. on pp. vii, 1).

[7] Amit Dang, Dimple Dang, and Pawan Rane. "The Expanding Role of Digital Therapeutics in the Post-COVID-19 Era." en. In: *Open COVID J.* 1.1 (May 2021), pp. 32–37. ISSN: 2666-9587 (cit. on pp. vii, 1).

[8] *How Does Trust Impact the Digital Healthcare Experience?* en-us. URL: https://medicalalley.org/2021/04/how-does-trust-impact-the-digital-healthcare-experience/ (visited on 01/08/2021) (cit. on pp. vii, 1).

[9] Konstantin Knorr, David Aspinall, and Maria Wolters. "On the Privacy, Security and Safety of Blood Pressure and Diabetes Apps." In: *ICT Systems Security and Privacy Protection* (2015), pp. 571–584 (cit. on pp. viii, 2, 9, 66).

[10] Konstantin Knorr and David Aspinall. "Security testing for Android mHealth apps." In: *2015 IEEE Eighth International Conference on Software Testing, Verification and Validation Workshops (ICSTW)* (Apr. 2015), pp. 1–8 (cit. on pp. viii, 2, 9).

[11] Roy Vermeulen, Edgar Bohte, Alex Stavroulakis, and Vincent de Jager. "Security evaluation of glucose monitoring applications for Android smartphones." en. In: *University of Amsterdam Archive* (2019), p. 21 (cit. on pp. viii, 2, 9, 82).

[12] Mehrdad Aliasgari, Michael Black, and Nikhil Yadav. "Security Vulnerabilities in Mobile Health Applications." en. In: *2018 IEEE Conference on Application, Information and Network Security (AINS)* (Nov. 2018), pp. 21–26 (cit. on pp. viii, 2, 9, 64, 70).

[13] Achilleas Papageorgiou, Michael Strigkos, Eugenia Politou, Efthimios Alepis, Agusti Solanas, and Constantinos Patsakis. "Security and Privacy Analysis of Mobile Health Applications: The Alarming State of Practice." en. In: *IEEE Access* 6 (2018), pp. 9390–9403. ISSN: 2169-3536 (cit. on pp. viii, 2, 9).

[14] Chiara Braghin, Stelvio Cimato, and Alessio Della Libera. "Are mHealth Apps Secure? A Case Study." en. In: *2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC)* (July 2018), pp. 335–340 (cit. on pp. viii, 2, 9).

[15] Chi Zhang, Hossain Shahriar, and A B M Kamrul Riad. "Security and Privacy Analysis of Wearable Health Device." en. In: *2020 IEEE 44th Annual Computers, Software, and Applications Conference (COMPSAC)* (July 2020), pp. 1767–1772 (cit. on pp. viii, 2, 10).

[16] Patty Kostkova. "Grand Challenges in Digital Health." In: *Frontiers in Public Health* 3 (2015), p. 134. ISSN: 2296-2565 (cit. on pp. 5, 6).

[17] Francesco Petracca, Oriana Ciani, Maria Cucciniello, and Rosanna Tarricone. "Harnessing Digital Health Technologies During and After the COVID-19 Pandemic: Context Matters." en. In: *J Med Internet Res* 22.12 (Dec. 2020), e21815. ISSN: 1438-8871 (cit. on p. 5).

[18] Kritika Roy. *Digital healthcare and rising cyber concerns*. en-us. Section: Research & Analysis. June 2020. URL: https://www.thedispatch.in/digital-healthcare-and-rising-cyber-concerns/ (visited on 03/08/2021) (cit. on pp. 5, 6).

[19] Emily Skahill and Darrell M. West. *Why hospitals and healthcare organizations need to take cybersecurity more seriously*. en-us. Aug. 2021. URL: https://www.brookings.edu/blog/techtank/2021/08/09/why-hospitals-and-healthcare-organizations-need-to-take-cybersecurity-more-seriously/ (visited on 01/08/2021) (cit. on pp. 5, 6).

[20] Shirley Li. *Your Medical Data Is Worth More on the Black Market Than Financial Data*. en. Section: Technology. Mar. 2015. URL: https://www.theatlantic.com/technology/archive/2015/03/the-next-cybersecurity-target-medical-data/388180/ (visited on 03/08/2021) (cit. on p. 5).

[21] Nisarg A. Patel and Atul J. Butte. "Characteristics and challenges of the clinical pipeline of digital therapeutics." en. In: *npj Digit. Med.* 3.1 (Dec. 2020), p. 159. ISSN: 2398-6352 (cit. on pp. 6, 7, 33, 81).

[22] Emiliano Casalicchio, Sebastiano Filetti, Sabrina Grigolo, Luigi V Mancini, Alessandro Mei, Giulio Pagnotta, Alice Ravizza, Angelo Spognardi, and Silvia Stefanelli. "Privacy e cybersecurity nell'ambito delle terapie digitali." it. In: *Tendenze Nuove* Numero Speciale 1 (2021), pp. 51–70 (cit. on pp. 7, 8, 33, 64, 69, 81, 83).

[23] Medical Device Coordination Group. *Guidance on Cybersecurity for medical devices*. en. European Commission, Dec. 2019, p. 45 (cit. on p. 7).

[24] Medical Device Cybersecurity Working Group. *Principles and Practices for Medical Device Cybersecurity*. en. IMDRF (Medical Device Regulators Forum), 2020, p. 46 (cit. on p. 7).

[25] *Digital Therapeutics Definition and Core Principles*. en. Nov. 2019. URL: https://dtxalliance.org/wp-content/uploads/2021/01/DTA_DTx-Definition-and-Core-Principles.pdf (visited on 08/02/2021) (cit. on p. 7).

[26] *DTx Product Best Practices*. en. Nov. 2019. URL: https://dtxalliance.org/wp-content/uploads/2021/01/DTA_DTx-Definition-and-Core-Principles.pdf (visited on 08/02/2021) (cit. on pp. 7, 8, 69, 79, 83).

[27] John Torous, Gerhard Andersson, Andrew Bertagnoli, Helen Christensen, Pim Cuijpers, Joseph Firth, Adam Haim, Honor Hsin, Chris Hollis, Shôn Lewis, David C. Mohr, Abhishek Pratap, Spencer Roux, Joel Sherrill, and Patricia A. Arean. "Towards a consensus around standards for smartphone apps and digital mental health." en. In: *World Psychiatry* 18.1 (Feb. 2019), pp. 97–98. ISSN: 1723-8617, 2051-5545 (cit. on p. 8).

[28] *Mobile Operating System Market Share Worldwide*. en. URL: https://gs.statcounter.com/os-market-share/mobile/worldwide (visited on 06/08/2021) (cit. on p. 8).

[29] Scott Ikeda. *Pandemic Pressures Reveal Significant Mobile Application Security Issues; Thousands of the Most Popular Apps Contain Common Vulnerabilities*. en-us. May 2021. URL: https://www.cpomagazine.com/cyber-security/pandemic-pressures-reveal-significant-mobile-application-security-issues-thousands-of-the-most-popular-apps-contain-common-vulnerabilities/ (visited on 08/08/2021) (cit. on p. 8).

[30] University of Zakho, Omar Ahmed, Amira Sallow, and Nawroz University. "Android Security: A Review." en. In: *ACAD J NAWROZ UNIV* 6.3 (2017), pp. 135–140. ISSN: 2520789X (cit. on p. 8).

[31] Sebastian Porst. *Overview of common Android app vulnerabilities | Bugcrowd*. en. Oct. 2019. URL: https://www.bugcrowd.com/resources/webinars/overview-of-common-android-app-vulnerabilities/ (cit. on p. 9).

[32] Xiaolei Wang, Andrea Continella, Yuexiang Yang, Yongzhong He, and Sencun Zhu. "LeakDoctor: Toward Automatically Diagnosing Privacy Leaks in Mobile Applications." en. In: *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 3.1 (Mar. 2019), pp. 1–25. ISSN: 2474-9567 (cit. on p. 9).

[33] *DTx Product Best Practices*. en. Mar. 2020. URL: https://dtxalliance.org/wp-content/uploads/2020/03/DTx-Disease-State-Targets_03.20.pdf (visited on 08/04/2021) (cit. on p. 33).

[34] Harold Thimbleby. "Misunderstanding IT: Hospital cybersecurity and IT problems reach the courts." en. In: *DEESLR* 15.0 (June 2018), pp. 11–32. ISSN: 2054-8508, 1756-4611 (cit. on p. 15).

[35] Adam Shostack. *Threat Modeling: Designing for Security*. 1st. Wiley Publishing, 2014. ISBN: 1118809998 (cit. on p. 18).

[36] Carlos Holguera, Sven Schleier, Bernhard Mueller, and Jeroen Willemsen. *OWASP Mobile Security Testing Guide*. OWASP. July 2021. URL: https://github.com/OWASP/owasp-mstg/ (cit. on pp. 27, 28).

[37] skylot. *skylot/jadx*. GitHub. Sept. 2021. URL: https://github.com/skylot/jadx (cit. on p. 29).

[38] Connor Tumbleson. *iBotPeaches/Apktool*. GitHub. Sept. 2021. URL: https://github.com/iBotPeaches/Apktool (cit. on p. 29).

[39] *Android Debug Bridge (adb)*. Android Developers. URL: https://developer.android.com/studio/command-line/adb (cit. on p. 29).

[40] *Burp Suite - Application Security Testing Software*. PortSwigger. URL: https://portswigger.net/burp (cit. on p. 30).

[41] *Brida*. GitHub. URL: https://github.com/federicodotta/Brida (cit. on p. 30).

[42] *Frida*. GitHub. URL: https://github.com/frida (cit. on p. 30).

[43] *Mobile Security Framework (MobSF)*. GitHub. URL: https://github.com/MobSF/Mobile-Security-Framework-MobSF (cit. on p. 31).

[44] Bee Team. *Ostorlab*. Ostorlab. URL: https://report.ostorlab.co/crosscom/ (cit. on p. 32).

[45] *SQLite Home Page*. URL: https://www.sqlite.org/index.html (visited on 08/16/2021) (cit. on p. 30).

[46] *projectdiscovery/subfinder*. URL: https://github.com/projectdiscovery/subfinder (cit. on p. 30).

[47] *Product Library*. en-us. Digital Therapeutics Alliance. URL: http://dtxalliance.org/understanding-dtx/product-library/ (cit. on pp. 33, 46, 81).

[48] Kapua Iao. *Personally identifiable information: HIPAA compliance key facts*. en-us. Section: Definitions. Jan. 2020. URL: https://www.paubox.com/blog/personally-identifiable-information-hipaa-and-pii-compliance/ (visited on 02/08/2021) (cit. on p. 89).

[49] *Man-in-the-Middle (MITM) Attacks: Techniques and Prevention*. en. URL: https://www.rapid7.com/fundamentals/man-in-the-middle-attacks/ (cit. on p. 90).

[50] *Understanding Denial-of-Service Attacks | CISA*. URL: https://us-cert.cisa.gov/ncas/tips/ST04-015 (visited on 09/08/2021) (cit. on p. 90).

[51] Nisarg A. Patel and Atul J. Butte. "Characteristics and challenges of the clinical pipeline of digital therapeutics." en. In: *npj Digit. Med.* 3.1 (Dec. 2020), p. 159. ISSN: 2398-6352 (cit. on p. 90).

[52] Daniel Brecht. *Phishing techniques: Expired password/account*. en-US (cit. on p. 90).

[53] Archiveddocs. *SQL Injection*. en-us. URL: https://docs.microsoft.com/en-us/previous-versions/sql/sql-server-2008-r2/ms161953(v=sql.105) (visited on 08/15/2021) (cit. on p. 91).

[54] *Cross Site Scripting (XSS) Software Attack | OWASP Foundation*. en. URL: https://owasp.org/www-community/attacks/xss/ (cit. on p. 91).

APPENDIX

A.1 Personally Identifiable Information (PII) list [48]

1. First name only
2. First initial with last name
3. Place of birth
4. Geographic indicators
5. Basic demographic information
6. Zip code
7. Date of birth/death

A.2 Protected Health Information (PHI) list [48]

1. Full names or last name and initial
2. Height or weight
3. Dates (other than year) directly related to an individual such as birthday or treatment dates
4. Phone Numbers including area code
5. Medical identification numbers
6. Health insurance
7. Beneficiary numbers
8. Health status
9. Blood test results
10. X-rays
11. Invoices
12. Payment history
13. Appointment reminders
14. Admission and discharge dates

15. Medical device identifiers and serial numbers
16. Mental health records

A.3 MITM ATTACKS

Man-in-the-middle attacks (MITM) are a common type of cybersecurity attack that allows attackers to eavesdrop on the communication between two targets. The attack takes place between two legitimately communicating hosts, allowing the attacker to "listen" to a conversation they should normally not be able to listen to, hence the name "man-in-the-middle" [49].

The category of MITM attacks includes different types of attacks that exploit the network protocol in different ways to intercept communication between two targets. In our work, we consider the possibility of using a malicious certificate to intercept the HTTPS protocol.

A.4 DOS ATTACKS

A denial-of-service (DoS) attack occurs when legitimate users are unable to access information systems, devices, or other network resources due to the actions of a malicious cyber threat actor. Services affected may include email, websites, online accounts (e.g., banking), or other services that rely on the affected computer or network. A denial-of-service condition is accomplished by flooding the targeted host or network with traffic until the target cannot respond or simply crashes, preventing access for legitimate users. DoS attacks can cost an organization both time and money while their resources and services are inaccessible [50].

A.5 PHISHING ATTACKS

Phishing is a form of social engineering in which criminal attempts to fraudulently retrieve legitimate users' confidential or sensitive credentials by mimicking electronic communications from a trustworthy or public organization. Phishing e-mails always ask victims to click a link that will guide the victim to a forged website where personal information is requested [51].

In the case of a password reset e-mail, two things can happen. The user is redirected to a page that looks like part of the legitimate site. This is actually a bogus page where both new and existing passwords are requested. The old, original password is hijacked and quickly used by the attacker to access the account and network. Otherwise, the link lands the user on the actual password renewal page. In the background, it loads a malicious script that hijacks the user's session cookie, results in a reflected XSS attack and opens the account to the attacker [52].

A.6 SQL INJECTIONS

SQL injection is an attack in which malicious code is inserted into strings that are later passed to an instance of SQL Server for parsing and execution. Any procedure that constructs SQL statements should be reviewed for injection vulnerabilities because SQL Server will execute all syntactically valid queries that it receives. Even parameterized data can be manipulated by a skilled and determined attacker [53]. In particular, it is very dangerous because the attacker could dump the entire company database, i. e. all the health data of all the patients in the case of the analysed app.

A.7 XSS ATTACKS

Cross-Site Scripting (XSS) attacks are a type of injection, in which malicious scripts are injected into otherwise benign and trusted websites. XSS attacks occur when an attacker uses a web application to send malicious code, generally in the form of a browser side script, to a different end user. Flaws that allow these attacks to succeed are quite widespread and occur anywhere a web application uses input from a user within the output it generates without validating or encoding it.

An attacker can use XSS to send a malicious script to an unsuspecting user. The end user's browser has no way to know that the script should not be trusted, and will execute the script. Because it thinks the script came from a trusted source, the malicious script can access any cookies, session tokens, or other sensitive information retained by the browser and used with that site. These scripts can even rewrite the content of the HTML page [54].