SILVER SURFERS ON THE TECH WAVE: PRIVACY ANALYSIS OF ANDROID APPS FOR THE ELDERLY

PRANAY KAPOOR

A THESIS

IN

THE DEPARTMENT OF

CONCORDIA INSTITUTE FOR INFORMATION SYSTEMS ENGINEERING

PRESENTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF MASTER OF APPLIED SCIENCE

IN INFORMATION SYSTEMS SECURITY

CONCORDIA UNIVERSITY

MONTRÉAL, QUÉBEC, CANADA

SEPTEMBER 2022

CONCORDIA UNIVERSITY

School of Graduate Studies

This is to certify that the thesis prepared

By: **Pranay Kapoor**

Entitled: **Silver Surfers on the Tech Wave: Privacy Analysis of Android Apps for the Elderly**

and submitted in partial fulfillment of the requirements for the degree of

Master of Applied Science
in Information Systems Security

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

Dr. Carol Fung _____ Chair

Dr. Mohammad Mannan _____ Supervisor

Dr. Amr Youssef _____ Supervisor

Dr. Carol Fung _____ Examiner

Dr. Ivan Pustogarov _____ Examiner

Approved by _____
Dr. Zachary Patterson, Graduate Program Director

August 24, 2022 _____

Dr. Mourad Debbabi, Dean
Gina Cody School of Engineering and Computer Science

ABSTRACT

Silver Surfers on the Tech Wave: Privacy Analysis of Android Apps for the Elderly

Pranay Kapoor

Like other segments of the population, elderly people are also rapidly adopting the use of various mobile apps, and numerous apps are also being developed exclusively focusing on their specific needs. Mobile apps help the elderly to improve their daily lives and connectivity, their caregivers and family members to monitor their loved ones' well-being and health-related activities. While very useful, these apps also deal with a lot of sensitive private data such as healthcare reports, live location, and Personally Identifiable Information (PII) of the elderly and caregivers. While the privacy and security issues in mobile applications for the general population have been widely analyzed, there is limited work that focuses on elderly apps. We shed light on the privacy and security issues in mobile apps intended for elderly users, using a combination of dynamic and static analysis on 146 popular Android apps from the Google Play Store. To better understand some of these apps, we also test their corresponding IoT devices. Our analysis uncovers numerous security and privacy issues, leading to the leakage of private information and allowing adversaries to access user data. We find that 95/146 apps fail to adequately preserve the security and privacy

of their users in one or more ways; specifically, 15 apps allow full account takeover, and 9 apps have an improper input validation check, where some of them allow an attacker to dump the database containing elderly and caregivers' sensitive information. We hope our study will raise awareness about the security and privacy risks introduced by these apps, and direct the attention of developers to strengthen their defensive measures.

Acknowledgments

I would like to thank my supervisors, Dr. Mohammad Mannan and Dr. Amr Youssef for their constant support and guidance throughout this project. Their continued support gave life to this project and made this research possible. I would also like to express my gratitude for their patience, motivation, enthusiasm, and immense knowledge. I am incredibly lucky to be able to work under the close guidance of my supervisors who inspired me with bright ideas, helpful comments, suggestions, and insights which have contributed to the improvement of this work.

I would also like to thank my peers at the Madiba Security Research Group for sharing their knowledge and experience and being there beside me on my rainy days. I learned a lot from everyone, especially, Sajjad Pourali, Rohan Pagey, and Bhaskar Tejaswi. I feel lucky and grateful to be a part of this research group. Special thanks to all professors at CIISE. They all provided me with the opportunity to learn in a positive learning environment and made me more interested in all aspects of systems security. I received substantial financial support from my supervisors and Concordia University and I am thankful for easing the financial burden during my research. Lastly, I would like to thank my family and friends. This journey would not have been possible without their encouragement and support.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Overview

The adoption of mobile devices is forcing the elderly to navigate the treacherous waters of a complex digital world [6], wherein online threats can even translate into offline harm. While over 53% of all elderly own a smartphone [2], and are keenly adopting mobile technology [8, 22], several studies have shown that older adults are more vulnerable to security and privacy threats than the general population [24]. According to US FBI and FTC, cybercrimes against older adults in the US have increased five times since 2014, costing over $650 million in yearly losses [5]. A combination of low self-efficacy, mistrust, and lack of awareness and understanding of security hazards [27] makes the elderly reluctant to adopt cyber-secure habits, hence vulnerable.[1]

Applications for the elderly offer various services such as care-giving, e-learning, and

[1]The term "vulnerable user" means a person "at-risk" due to his/her particular circumstances, and not to be confused with an app having a security "vulnerability".

improving physical and mental health (e.g., apps for exercise and fitness). While these apps might be used daily by the elderly, their inherent privacy and security implications are not fully known. Weaknesses in elderly apps may expose sensitive private data, sometimes on a large scale, and endanger users' safety (online and in the real world). Recent studies [12] have revealed several security and privacy issues in Android apps, but most large-scale research has been done on apps used by the general population (also see Sec. 2.3). A few studies have exposed privacy issues in only one particularly vulnerable group (e.g., elderly or children) on a small scale. The work on elderly groups is limited to the study of elderly behavior concerning their privacy and security.

In this thesis, we perform an in-depth analysis of 146 prominent elderly apps. We focus on the group of elderly users that may be more vulnerable to online scammers and cyber-predators, especially in the background of the COVID pandemic. Various mobile apps are available to help the elderly cope with or even improve their special situations. Mobile apps benefit the elderly with medical, health, and fitness information, self-help tools, financial planning, social and family connectivity, interactive play, and puzzles [32]. Our rationale for analyzing Android apps for the elderly is based on the fact that Google Play Store and Apple App Store have approximately 2.7 million Android apps and 1.82 million iOS apps respectively [17], and while the elderly are adopting all forms of modern technology devices, the adoption of smartphones is the highest at 77% as per a recent study by AARP [22]. We define a list of pertinent security and privacy related issues for these apps, and analyze them for such issues (e.g., security vulnerabilities, backend issues, presence of third-party trackers, and insecure data transmission). We also analyze three

IoT devices to better understand the corresponding apps and their security implications. We combine the use of several existing tools that enable dynamic and static analysis to perform a wide range of security and privacy tests.

1.2 Contributions

Our contributions can be summarized as follows:

1. We design a hybrid approach of dynamic and static analysis for evaluating security and privacy issues in elderly apps (and their corresponding IoT devices). We inspect the apps' web traffic for personally identifiable information (PII) leakage, access control issues, improper authentication management, improper input validation, dangerous third-party library permissions, and the presence of third-party trackers.

2. We apply our analysis framework to 146 Android apps (and the IoT devices corresponding to three apps). Overall, 95/146 apps fail to adequately protect the security and privacy of users due to one or more vulnerabilities.

3. 4/146 Android apps (*GoldenApp*, *POC EVV*, *Senior Discounts*, *Damava*) do not properly authenticate their server API endpoints, allowing illegitimate access to view and obtain sensitive data such as elderly users' physical address, email, health reports, and private messages on the platform.

4. 15/146 Android apps (e.g., *40 Plus Senior Dating*, *All Well Senior Care*, *Seniority*) allow an attacker to easily compromise the account of elderly users and caregivers.

5. 9/146 (*Senior Dating*, *Empowerji*, *GoldenApp*, *Caring Village*, *EZ Care*, *Generations Homecare System*, *EllieGrid*, *Seniority*, *Tricella Health*) Android apps have improper input validation with injection attack vulnerabilities such as SQL injection, allowing an adversary to dump and modify the application's database. We are assigned CVE-2022-30083 for the code injection issue we found in *EllieGrid*.

6. 16/146 Android apps transmit PII via HTTP to their client-side servers (e.g., *Empowerji*, *GoldenApp*), while 8/146 apps transmit PII (6/146 via HTTP and 2/146 via HTTPS) to various third-party domains.

Most of the work presented in this thesis has been published at the 18th EAI International Conference on Security and Privacy in Communication Networks (Securecomm 2022) [23].

1.3 Thesis Organization

The rest of the thesis is organized as follows. In Chapter 2, we present background information for consequent chapters, such as, services that can be exploited by malicious actors, threat model, ethical considerations and related work. In Chapter 3, we introduce our framework and the techniques used to collect and categorize data sets of apps, and conduct static and dynamic analysis. In Chapter 4, we present the experimental results of our analysis and discuss the impact of our findings. In Chapter 5, we discuss case studies using specific examples of apps to understand the real-world impact of our findings. Finally, in Chapter 6, we present various discussion points and end with our concluding remarks.

Chapter 2

Background

2.1 Services That Can be Exploited by Malicious Actors

2.1.1 Tracking Services

While some "tech-savvy" users may be aware of the existence of tracking services (such as cookies and geolocation) on mobile apps, the elderly population may not even be aware that such tracking techniques exist, or that these are being deployed on the apps that they use daily. Tracking services are mostly used either for analytic or advertisement purposes for companies [42]. Analytic trackers generally get users' phone information (e.g., build number, screen size, Android version, etc.) to help maintain and improve the app's performance [34]. Ad trackers on the other hand are used to track a user's behavior within the application so that it can be used to show personalized ads to the user in the future [33].

Elderly users should feel safe that there is no online threat to their privacy while using mobile apps. Any disruption or threat to the elderly persons' digital security and privacy can harm their willingness to use mobile technology — a particularly big problem considering just how much technology permeates these people's everyday lives.

For example, on July 14, 2020, a lawsuit was filed accusing Google of violating federal wiretap law and California privacy law by collecting and storing users' data, logging what the users are looking at in many types of apps [35].

2.1.2 Third-Party Libraries

Advertising networks, gaming networks, and analytics engines are an integral part of modern mobile platforms. If Android developers want to integrate functionality provided by third-party libraries, they must bundle opaque binary code into their applications. Unfortunately, developers must in essence over-privilege their Android applications by requesting dangerous permissions, such as full Internet access, solely to support third-party libraries. For example, a simple puzzle app that requires no network functionality to operate must now request multiple dangerous permissions to become advertising-supported, participate in a gaming network, or report usage statistics. Consequently, third-party libraries have access to all the systems resources as the host application does. By design, third-party libraries inherit all dangerous permissions granted to the host application and obtain the rights to application-accessible sensitive data. These permissions and data might include the following: read or write access to contacts, record audio, take pictures, read or send SMS, make phone calls, read phone state and identity, read and write to storage, obtain fine

or coarse-location, modify system settings, and perhaps create network sockets to arbitrary hosts with the ability to transmit any of this sensitive data.

In-app advertisements on Android apps are a conduit for unsafe exposure of vulnerable users [46]. Ad libraries (mobile web libraries, rich media libraries, and ad mediators) [14] commonly have user-interface code (to present ads) and network code (to request ads from servers). They are tightly bundled with host apps to prevent disabling the ad functionality, and some even discourage reverse engineering, such as embedded ad libraries of a malicious ad network can inherit risky host app permissions to set up a remote bot or root attack. Threats include collecting unnecessarily intrusive user information such as geolocation, phone contacts, browser bookmarks, and allowing unknown third-party code to execute within the hosting app. A 2019 study of 5000 Android apps found that 65% of used permissions are derived from third-party libraries, and 48-59% dangerous permissions are linked to third-party libraries [9]. As third-party libraries are widely used in Android apps, accounting for more than 60% of the code in Android apps on average, it becomes essential that we analyze the third-party libraries in each app [44].

2.1.3 App Backend Platforms

Backend services such as Firebase have recently come into focus for their vulnerabilities. Firebase is Google's platform that helps developers build apps. Firebase offers backend services such as authentication, hosting, real-time database, cloud storage services, delivering notifications and ads, crashlytics, tracking glitches and clicks, cloud messaging, remote config, and test lab for app developers. An analysis of 15,735 Android apps, found 4.8%

of mobile apps using Google Firebase to store user data were not properly secured, allowing anyone to access databases containing users' personal information, access tokens, and other data without a password or any other authentication. The exposed database of 4,282 apps included 7,000,000+ email addresses, 4,400,000+ usernames, 1,000,000+ passwords and 5,300,000+ phone numbers. Firebase databases of 9,014 apps had write permissions, thus potentially allowing an attacker to inject malicious data and corrupt the database, and even spread malware [41]. Another brute-force attack exploited Firebase Auth. that allowed hackers from Nigeria to log in to several user accounts and drain funds to a hacker-controlled PayPal [25]. Firebase messaging vulnerability allowed attackers to send push notifications to app users. Issues with FCM relating to Legacy Server Keys were abused to send requests via legacy HTTP, thereby circumventing security measures in HTTP v1 protocol [31].

2.1.4 Privacy Policies

App developers often include a privacy policy as an afterthought as they are obligated to notify users of their privacy practices to comply with legal requirements. However, prior research has suggested that many developers are not accurately disclosing their apps' privacy practices. Evaluating discrepancies between apps' code and privacy policies enables the identification of potential compliance issues. A privacy evaluation for over one million Android apps reveals broad evidence of potential non-compliance. Many apps do not have a privacy policy, to begin with. Policies that do exist are often silent on the practices

performed by apps [47]. Users are vaguely aware that apps collect data such as name, location, gender, and email id when they register. However, elderly users may be at some risk if there exist inadequate security mechanisms for data storage, transmission, and access. Users also need to be aware of their rights with respect to the data collected (e.g. right to opt-out, edit or delete), how long the data will be preserved by the company, and how they will be notified if there are policy changes.

2.2 Potential Security and Privacy Issues

We primarily consider two types of data that can be leaked over the network: (1) personally identifiable information (PII) and (2) smartphone device information and usage. A PII leak is any data leak that can be used to identify an individual (e.g., email ID, location/address, password, date of birth, health data, unique device serial number). Device information and usage is the combination of the device data (e.g., manufacturer, model, OS, API level, IP address, screen, battery, cellular carrier, free memory/disk, language, time zone, orientation), and user interaction (e.g., session time, button clicks, visited web pages). Device information and usage leaks can be used to identify an individual or a group of individuals. We define the following list of potential security and privacy issues to evaluate elderly apps.

1. Improper authentication management: The ability of an attacker to gain access to a user's account (unauthorized login).

2. Improper access control: To be able to gain or observe other users' data on a given platform without their authorization.

3. Improper input validation: Possible injection attacks (e.g., SQL injection and code injection) resulting from missing/inadequate input validation, which may compromise sensitive user data.

4. Vulnerable backend: The use of remotely exploitable outdated server software, and misconfigured or unauthenticated backend service (e.g., Firebase).

5. Plaintext transmission of authentication secrets (e.g., passwords and session IDs), which can be easily captured by a network attacker to gain unauthorized access to user accounts.

6. Insecure PII, device information and usage transmission: PII and device information and usage from the client-end is sent without encryption (i.e., plain HTTP).

7. Data transmission to third-party: Any PII and device/usage information and usage data transmitted from the client side to any third-party domains/ trackers, or library providers.

8. Inadequate security configurations: Android apps with misconfigured backend HTTP web servers (e.g., lack of Cross-Origin Resource Sharing or improper flash cross-domain policy), which may lead to large-scale attacks.

9. Dangerous permissions (e.g., Write External Storage, Access Fine Location) automatically acquired by a third-party library when requested by the elderly app, or by a malicious app using the same signed certificate third-party library as the elderly app developer.

Definition of Elderly Users. The elderly population is defined by Organisation for Economic Co-operation and Development (OECD) as people aged 65 and over [29]. The United States Census Bureau commonly labels the population 65 years old and over as the "elderly" and 65 is the age that U.S. citizens are legally considered seniors [3, 16]. We, therefore, use the term "elderly" in this thesis to describe people that are 65+ years old. The elderly will comprise 22% of the world's population by 2050 [43].

2.2.1 Threat Model

We consider three attacker types with varying capabilities: (1) On-device attacker: a malicious app with limited permissions on the user's device. (2) On-path attacker: an attacker who is placed between the user's smartphone and its server. This attacker can eavesdrop, modify, and behave like a man-in-the-middle attacker between the user's device and the app's backend server. (3) Remote attacker: any attacker who can connect to an app's backend server. Our threat model does not consider attacks requiring physical access to the device.

2.2.2 Ethical Considerations and Responsible Disclosure

We test vulnerabilities only against accounts that we own and we do not interact with the data of any legitimate user. We do not use an existing vulnerability to exfiltrate data or pivot to other systems, i.e., we stop our analysis when we have enough evidence of a vulnerability and its impact. We also refrain from running any automated scanners that might bombard the servers to cause a denial of service. As part of the responsible disclosure, we contacted

the developers of our vulnerable apps to share our detailed proof-of-concept and explain to them the related security consequences. 7/35 developers contacted us back, where 2/7 were automated replies to acknowledge our email, and 5/7 developers acknowledged the issues and forwarded them to their respective security teams.

2.3 Related Work

In this section, we enumerate a few examples of real-world data privacy issues faced by elderly users of Android apps and discuss some academic studies related to privacy analyses of apps used by this demographic.

Slane et al. [39] collected seniors' perspectives on technological devices and applications to show how seniors protect their personal information, and what knowledge, tools, and support they would need in order to consider new functions or devices.

Huckvale et al. [19] assessed 79 clinically safe medical/health apps used by chronic and unwell persons, and found that 23/79 of apps sent unencrypted PII over the Internet, 63/79 apps communicated directly with third-party services and 53/79 of apps had some form of privacy policy. However, this work does not specifically study elderly users, or analyze the backend security issues.

Frik et al. [11] identified a range of complex privacy and security attitudes and needs specific to *older adults*, along with common threat models, misconceptions, and mitigation strategies. They showed how older adults' limited technical knowledge, experience, and declining abilities amplify vulnerability to certain risks. They also found that older adults

often experience usability issues or technical uncertainties in mitigating those risks, and that managing privacy and security concerns frequently consists of limiting or avoiding technology use.

Gibler et al. [12] obtained over 23,000 Android applications from several Android markets and found 9,631 potential privacy leaks in 3,258 Android applications of private data including phone information, GPS location, WiFi data, and audio recorded with the microphone. This study, although large-scale, is however not focused on any vulnerable user group in particular.

Oliveira et al. [30] showed that older women were the most vulnerable group to phishing attacks in a study of 158 Internet users. However, these studies are limited to the study of apps for elderly in terms of a vulnerable user group, and does not analyze the traffic flow through domains and trackers, third party library permissions, privacy policies of apps and backend security issues.

Razaghpanah et al. [33] identified 2,121 third-party advertising and tracking services at the traffic level, of which 233 were previously unknown to other popular advertising and tracking blacklists. Their analysis of the privacy policies of the largest advertising and tracking service providers showed rampant sharing of harvested data with subsidiaries and third-party affiliates.

Ren et al. [34] analyzed 512 apps for privacy leaks over time across three dimensions (PII leaks, HTTPS adoption, and domains contacted) independently, and found that app privacy gets worse as users upgrade apps and all apps leak at least one type of PII, such as email address, first and last name, date of birth, phone number, contact info, gender.

Rosenfeld et al. [36] analyzed 72 apps for dementia and found only 46% having a privacy policy. There was a preponderance of missing information, the majority of apps acknowledged collecting individual data for internal purposes, and most policies admitted to instances in which they would share user data with outside parties. However, this study is limited to the study of apps for *dementia* in terms of a vulnerable user group, and does not analyze the traffic flow through domains and trackers, third party library permissions and backend security issues.

In contrast to the above work, we take an in-depth look at the security and privacy threats in Android apps used by the elderly. We also analyze traffic flows, PII or device information and usage leaks, dangerous permissions used by apps and third-party libraries, and backend security issues of high severity, using various tools for both dynamic and static analysis. Our initial framework also included Lumen Privacy Monitor for dynamic analysis of test apps, but we removed it from the framework as we found that Lumen did not uncover several security flaws as compared to Burp Suite. Even though Lumen would show leaks, there was a layer of uncertainty as to how a leak was transmitted (over HTTP or HTTPS) and where it was leaked (to the client-side product itself or to some third-party domains). Also, Lumen did not work reliably on newer Android versions and only worked best below Android 7. Hence, we decided to use a more manual approach with Burp Suite.

Chapter 3

Analysis Framework

In this section, we explain how we perform our static and dynamic app analysis, and also how we select our apps.

3.1 App Selection

We begin the process of identification of existing mobile applications for the elderly by determining the issues, challenges, and characteristics related to the aging process that might affect the elderly.

We search Google Play Store for elderly apps (and also screen the best apps for older adults [18]), with relevant keywords.[2] The search was conducted on May 20, 2021, which provided us with 500 apps for further consideration.

[2]The keywords include: “elderly”, “old”, “senior”, “dementia”, “Alzheimer’s”, “retirement”, “senior dating”, “pension”, “seniority”, “caregiver”, “memory”, “maturity”, “retiree”, “Electronic Visit Verification”, “EVV”, “senior health”, “memory games”.

Inclusion Criteria. We shortlist the apps based on the following criteria: (1) apps specifically designed for elderly users, their caregivers and relatives; (2) functional to enable testing (e.g., compatible with our test mobile device). We exclude apps needing financial accounts or verified identities (e.g., bank accounts, social security numbers), and apps not intended for the elderly. We manually screen each app to check if it satisfies our key requirements. Our final dataset contains 146 apps. We found that 24/146 apps have a companion IoT device, where 5/24 apps are pill managing apps and 19/24 are elderly tracking apps. We purchased 3/24 IoT devices (available without any subscription and deliverable to our location) to better understand their functionality. Altogether, these apps have been downloaded 20.8M+ times, with a range between 10M+ (*NeuroNation*) to 1000+ downloads (*CareGo* IoT companion app). Note that each caregiver/EVV app may indirectly serve (and have access to) hundreds or thousands of elderly people.

3.1.1 App Categorization Based on "Help-type"

The categorization of mobile apps for older adults according to Cunha et al. [7] is used for the analysis to enable a classification independent of the app stores. This classification was developed using a methodological search in Google Play Store for mobile apps designed to help older adults. Table 1 lists the 10 categories - Diagnostic, History, Improve, Informative, Interface, Measurement, Protection, Simulation, Trainer, and Tutorial - with examples of content topics.

Using the above "help-type" categorization, we classify all the 146 apps into "help-how" categories as shown in Table 2 (complete list in Table 9).

Table 1: "Help-type" app categories with examples

Categories	Exemplary topics
Diagnostic	Cognitive impairments, physical and mental illnesses
History	Monitoring of vital parameters such as blood pressure, and organization of daily activities
Improve	Relaxation, speech-to-text, text-to-speech, risk assessment, magnifying glass, medication recognition, pictogram-to-speech, communication portals, and social networks
Informative	Healthy living, education, and psychoeducation about mental and physical illnesses
Interface	Mobile apps for conversion to a user-friendly interface
Measurement	Physical activity, pedometer, and GPS tracking
Protection	Drug reminder, help requests, and localization
Simulation	Simulation of diseases, impairments, or appearance
Trainer	Memory, relaxation, logical thinking, fitness, and cognitive speed
Tutorial	Accident rehabilitation, sign language, improvement of self-esteem, and improvement of communication

Table 2: Categorization of all 146 apps by "help-type" (including the number of apps in each category)

Diagnostic	**3**	**History**	**20**	**Improve**	**15**	**Informative**	**12**	**Interface**	**11**
Hearing	2	EVV	20	Dating	13	Arthritis	6	Launchers	11
Brain test	1			Social	2	Shopping	3		
						Pension	2		
						Community	1		
Measure	**9**	**Protection**	**31**	**Simulation**	**3**	**Trainer**	**34**	**Tutorial**	8
Exercise	6	Tracking	17	Memory	3	Caregiver	34	Alzheimer's	6
Care	3	Pill alerts	7					Retirement	2
		Safety	7						

3.2 Dynamic Analysis of Traffic Flow

We perform dynamic testing of the apps to simulate the real world usage of the apps so that we can observe the apps as they were intended. We set up test environments for each app (creating user accounts, setting up the IoT device, etc), emulate user actions for 20 to 60 minutes depending on the feature-set of the app, collect traffic from the elderly apps and the IoT devices (up to 24 hours), and then perform our analysis (explained further in this section). Fig. 1 illustrates our methodology. We use Burp Suite[3] for manual dynamic analysis. Burp Suite is an integrated platform for security testing of web and mobile applications, using its various extensions. We also notice that some of our apps use GraphQL [21]; note that the use of graph analytics is driving many important business applications from social network analysis to machine learning. To analyze GraphQL APIs, we use the official GraphQL IDE called GraphiQL [13] to test the network traffic on the apps using GraphQL. In-depth dynamic analysis with Burp Suite and GraphiQL[4] helps us find relevant security and privacy issues in our test apps.

The four main components for our dynamic analysis include the following: (1) *Proxy*, an intercepting proxy that lets us see and modify the contents of requests and responses while they are in transit. We use this component to analyze the complete network traffic of the app to check for insecure session management, insecure PII transmission to the app as well as to any third-parties, and look out for any suspicious activity from the app. (2)

[3] https://portswigger.net/burp/releases/professional-community-2021-12-1
[4] https://github.com/graphql/graphiql

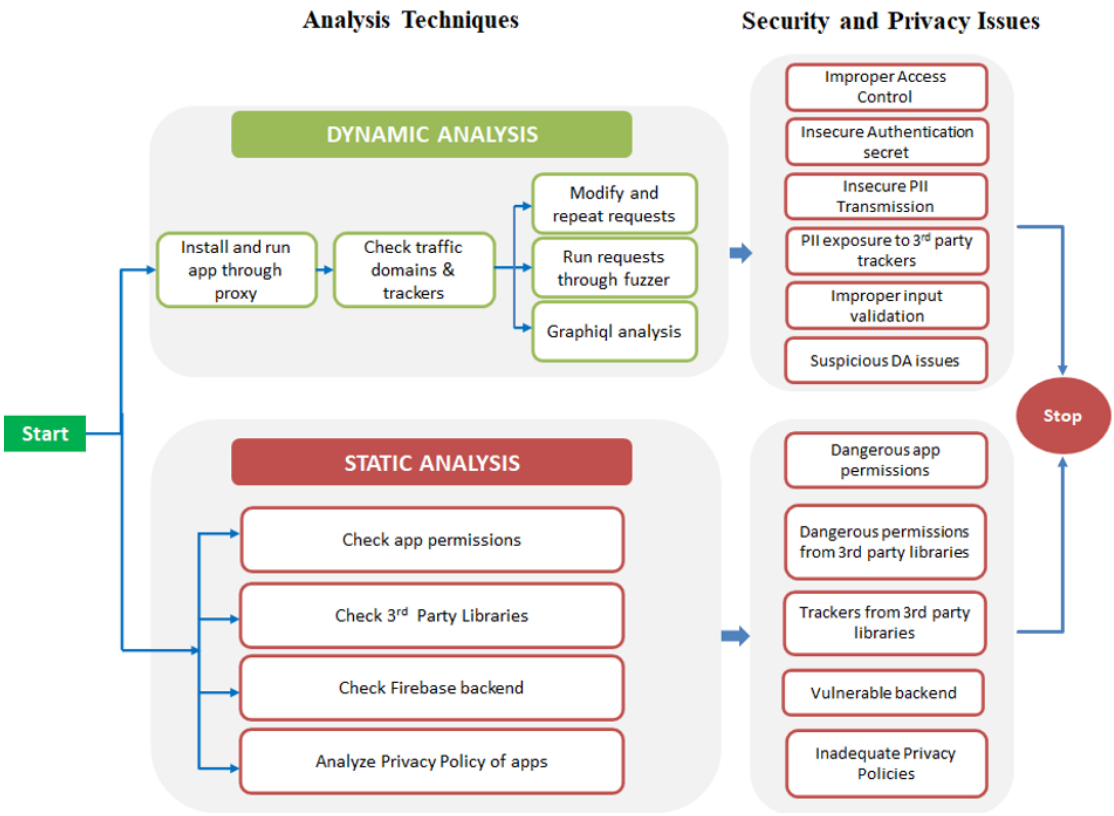

Figure 1: Overview of our methodology

Intruder, a fuzzer used to run a set of values through an input point and perform brute-force attacks and testing rate limiting on apps. We use this component to enumerate user IDs (integer values), list of passwords, and API endpoint parameters. (3) *Repeater* lets us send requests repeatedly with manual modifications to check for injection attacks and servers' response to unexpected values or requests. (4) *Decoder* lists the common encoding methods like URL, HTML, Base64, Hex, etc., when looking for chunks of data in values of parameters or headers.

We install the test app from Google Play Store and run it through Burp proxy. We analyze every request and response of the app's APIs (or any included third-party libraries) to the app server and to any third-party domain and tracker. We identify the known third-party

trackers using EasyList and EasyPrivacy [10] filtering rules. We differentiate the requests with weak authentication, like the ones which are missing authentication headers or cookies, as they are more likely to be exploitable. This differentiation is done by inspecting the HTTP request headers and searching for the presence of session headers. We also identify the requests responsible for user login/logout or any transmission of user data. We pass these requests through Burp components to check for security and privacy issues. The requests transmitted via GraphQL are analyzed using GraphiQL. In particular, we first use an introspection query to read the GraphQL documentation. Then we inspect the whole documentation to read the available API calls (queries and mutations). Vulnerabilities in GraphQL are found by probing and tampering with the queries and mutations.

We assess the collected traffic to check for PII and transmitted authentication secrets, or leakage of PII to third-party domains that can be leaked via the request URL, Referer, HTTP Cookie, and requests' payload. If encoded data is observed, we use the decoder component in Burp to check for any suspicious data that is being transmitted to the domain. We also analyze the traffic to check for API endpoints with improper access control. APIs with weak authentication are checked first. We conclude that an app has improper access control if we can retrieve any other user's data (on the given app, tested using our own accounts) by changing the existing requests sent from the app to its backend server.

To check improper input validation, we follow the OWASP manual [28] to test for injection attacks to see how the apps respond to unexpected modified requests. We check for SQL injection, code injection, and cross-site scripting (XSS) attacks. Any sensitive data observed is immediately deleted from our databases, and we only record the type of data

that the vulnerabilities exposed.

3.2.1 IoT Device Analysis

For each of the selected IoT devices, we test the companion apps, radio communications and the embedded device. We test the companion apps by following the same dynamic and static analysis process as for other apps. For the radio communications, we analyze Bluetooth and WiFi communications, and we do this by inspecting the packets sent between the IoT device and smartphone. To analyze the underlying embedded device, we pop open the IoT device and analyze the functionality of the different components. We look at the debug ports and try to exploit them to gain further access to the device.

3.3 Static Analysis: Library, App Code, and Firebase

Our static analysis aims to complement the dynamic analysis to understand the apps' intended flow so that we can correlate that with our dynamic analysis to look for any suspicious behavior or weak security measures (e.g., bad input sanitization, unprotected Firebase services, etc.) which can potentially lead to privacy or security issues. We target the following components:

3.3.1 Third-Party Libraries

Third-party libraries are widely used by Android app developers to build new functionalities and integrate external services. For an in-depth library analysis for our elderly apps,

we use LiteRadar.[5] We run the tool using our custom python script, with the APK file to be tested, so that we can automate the data (e.g., library names, type, permissions used, etc.) collection process. We analyse the libraries in terms of their permissions and purpose.

3.3.2 Firebase Analysis

We analyze the Firebase configuration for security issues by performing an automated analysis using Firebase Scanner [37]. Critical misconfigurations can allow attackers to retrieve all the unprotected data stored on the cloud server and we followed a similar approach to Appthority's work [1] on scanning apps for Firebase misconfigurations.

3.3.3 Static Code Analysis

Mobile Security Framework[6] (MobSF) is an automated, open-source, all-in-one mobile application (Android/iOS/Windows) pen-testing framework capable of performing fast static, dynamic, and malware analysis of Android, iOS, and Windows mobile applications [38]. So, we use MobSF for static analysis of 146 apps to check for vulnerabilities related to sensitive information logged or hard-coded in files, improper usage of SQLite databases, insecure implementation of SSL, and WebView implementation. We also check the Manifest file of each app to obtain their permissions.

Privacy Policy Analysis. We use an automated framework called Polisis[7] to analyze the

[5] https://github.com/pkumza/LibRadar/blob/master/docs/QuickStart.md
[6] http://opensecurity.in/mobilesecurity-framework/
[7] https://pribot.org/polisis/

privacy policies for 108 apps in Play Store. Polisis enables scalable, dynamic, and multi-dimensional queries on privacy policies. At the core of Polisis is a privacy-centric language model, built with 130K privacy policies, and a novel hierarchy of neural network classifiers that caters to the high-level aspects and the fine-grained details of privacy practices [15]. Polisis shows the output for various parameters/clauses in the policy such as Data Collection, Data shared with third parties, Choices for a user (what data users have control on and in which context - first-party collection or third-party sharing), Security practices, Data retention policy, Specific audiences (e.g., Europeans) mentioned for regulatory compliance, Data edit rights, and Policy update process.

Chapter 4

Experimental Results

Following the methodology in Sec. 3, we tested 146 Android apps for elderly people, between October 2020 and December 2021. For dynamic analysis, we ran the apps on a Samsung Galaxy M02 (SM-M022G) phone with Android 10. We report our findings in this section, with an overview of the top 30/146 apps with the most security and privacy issues in Table 3.

4.1 Improper Authentication Management

We found that 15/146 apps have authentication management vulnerabilities. Prominent examples include the following: In *Empowerji*, *40 Plus Senior Dating*, *GoldenApp*, *EZ Care*, *FlirtMatures Dating*, *POC EVV* and *Cougar Dating*, the login credentials are sent in plaintext over HTTP, so any on-path attacker sniffing the traffic can get the user login credentials (e.g., *Empowerji* leaks name, email ID, password and phone number; *POC EVV*

leaks the 6-digit user ID, a 4-digit PIN for login and the private messages sent between the caregiver and his/her supervisor). For *All Well Senior Care*, *Seniority* and *Tricella Health*, we successfully performed an OTP brute-force attack (on our test account). This is possible as these apps do not implement any rate-limiting and the OTPs consist of 4 or less numerical digits, which can easily be enumerated (even for the worst-case scenario, where we could easily try all 10000 requests for a 4-digit number); we also verified that full account takeover by a remote attacker takes only trivial efforts. In *All Well Senior Care*, the attacker can obtain the user's health data (e.g., heart rate, blood pressure, etc.), wellness data (wake up time, steps taken, etc.), see all the hourly updates the user is providing to her caregiver, the location of the user, all the health charts which are saved on the user's account, and even the private messages of the user with their caregiver or their care group (containing multiple users in one group). Wherein user information (e.g., address, phone numbers, credit card details) can be obtained in the *Seniority* app due to improper authentication management. During our retesting, we also noticed that *Senior Safety App* fixed its issues in a software update.

4.2 Insecure Session Management

We found 10/146 apps that had their session IDs sent in plaintext over HTTP. For example, *POC EVV* exposes its session ID in plaintext over HTTP, so an on-path attacker can replay a request from this app and perform an account takeover. Also, 8/146 apps did not use any authentication secret. For example, *GoldenApp* does not make use of any authentication

Table 3: Overall results for 30/146 elderly apps with maximum security flaws
Legend: ○: On-device Attacker, ◐: On-path Attacker, ●: Remote Attacker

App Name / Security Flaw	Improper Authentication Mgt.	Insecure Session Management	Insecure PII Transmission	PII Exposure to Third-party (3P)	Device Info. & Usage Exposure to 3P	Improper Input Validation	Improper Access control	Security Misconfigurations	File Path Manipulation
40 Plus Senior Dating (v9.8)	◐	◐	◐	◐	◐			◐	
Empowerji (v5.7)	◐	◐	◐		◐	●		◐	
GoldenApp (v3.2)	◐	◐	◐		◐	◐	●		
Senior Safety App (v9.7)	◐	◐	◐	◐				◐	
POC EVV (v3.2)	◐	◐	◐		◐		◐		
EZ Care (v0.0.7)	◐	◐	◐			●			
BrickHouse TrackView (v1.5.8)	◐	◐	◐		◐				
Family1st (v1.0.1)	◐	◐	◐		◐				
X-GPS Monitor (v2.10.4)	◐	◐	◐		◐				
DAGPS (v21100901)	◐	◐	◐		◐				
Tricella Health (v2.15.6)	●				◐	◐			
Oscar Senior/Enterprise (v6.8.2)			◐	◐				◐	
FlirtMatures Dating (v1.0)	◐	◐			◐				
Caring Village (v0.16.5)			◐			◐		◐	
Senior Discounts (v2.2)					◐		○	◐	
Seniority E-commerce app (v.1.0.2)	●			◐		◐			
Cougar Dating (v1.1.3)	◐		◐					◐	
Over 40 Dating Mature (v1.0)					◐				◐
Generations Homecare System (v3.3.3)						◐		◐	
Alzheimer's Daily Companion (v1.0.7)		◐	◐						
Big Launcher (v1.4)		◐		◐					
HelpAge SOS (v1.0.27)		◐	◐						
Carelinx (v3.0.1)				◐				◐	
Damava (v1.2.4)			●				●		
Doulikesenior (v1.5.1)					◐			◐	
Homage (v5.0.8)				◐				◐	
EllieGrid (v3.4.1)						●			
All Well Senior Care (v2.15.0)	●								
Mobile Caregiver+ (v2.0.35)					◐				
401(K) - Retirement Planning (v2.5)					◐				

secret for accessing any resource (which also leads to improper access control issues which is explained further in Sec. 4.4). The app's authorization mechanism is purely based on supplying a mobile number, where there is no verification from the server's end regarding which mobile number is tied to which user. An adversary can change the mobile number from the request and log into the replaced number's account. Although the victim's number is not leaked anywhere, an on-path attacker can still see the mobile number as the communications are over HTTP. For our testing, we used only our own test phone numbers. After changing the number, the attacker can impersonate the victim, e.g., to request home services on the user's behalf. Apps like *FlirtMatures Dating* send their session IDs in plaintext over HTTP; any on-path attacker can sniff these secrets, and potentially takeover a user's account, also allowing the attacker to access user's sensitive information.

4.3 PII Exposure, Data Sharing with Third-parties and Trackers

We found that 16/146 apps send plaintext PII to their servers. Examples include: *POC EVV* (login code, login pin, session ID during login), *40 Plus Senior Dating* (email ID and password during login), *Empowerji* (full name, email ID, password, mobile number and city), *GoldenApp* (username, mobile number, user address), and *EZ Care* (username and password during login and the private messages sent and received between a doctor and the user).

We found that 115/146 apps communicate with 341 third-party (non-tracker) domains:[8] 66 apps communicate with Googleapis.com domains, 43 apps with Firebase sub-domains and 29 apps with Facebook domains. 72/146 apps had traffic through at least one Google domain. The number of apps that communicate with the domains are shown in Table 4. We found 39 unique tracker domains with 137 occurrences across 76/146 apps (see Table 5). The top 3 prevalent trackers are Crashlytics (35/146), DoubleClick (22/146) and Google Syndication (12/146). Crashlytics is a crash reporting software that helps identify bugs in the apps and report the user's activity to the app developers so they can take appropriate measures to ensure that users do not stop using their app. DoubleClick is a Google ad service. In 9 apps, we detect 10 or more third-party domains and trackers (*Senior Discounts*, *Big Keyboard & Notifications*, *Free Chat & Senior Dating*, *Senior Dating by Lauber*, *Over 40 - Find People 50*, *40 Plus Senior Dating*, *NeuroNation*, *Ianacare*, *Oscar Senior*). These apps could expose elderly users to potential voluminous in-app advertisements, and extensive tracking.

Moreover, out of the 16 apps that send plaintext PII to their own servers, 6 of them also send PII in plaintext over HTTP to third-party domains/trackers. Examples include: *Oscar Senior* (email ID, user name and profile picture sent to googleapis, and geolocation to onesignal's API endpoint), *Big Launcher* (exact geolocation to openweathermap.org), *Carelinx* (email ID to intercom.com), *40 Plus Senior Dating* (email ID, user name and profile picture sent to googleapis), *Senior Dating* (user name and password sent to googleapis).

18/146 apps send device information and usage data in plaintext (6/146 over HTTP and

[8] A domain is considered to be a third-party domain if an app from a developer connects to it to enable third-party functions. Thus, the domain certificate owner is not the same as the developer of the app.

Table 4: 115/146 apps with traffic flow through 341 third-party domains (With number of apps in each app category)

Domains	Diagnostic	History	Improve	Informative	Interface	Measurement	Protection	Simulation	Trainer	Tutorial	Total
firebaseinstallations.googleapis.com	2	3	7	1	3		10	1	9	2	38
facebook.com	1	1	5	2	2			1	6	3	21
gstatic.com		1	7	6					2	1	17
google.com		4	5		2				3	1	15
fonts.googleapis.com		1	5	5	1				1	1	14
appcenter.ms		5	1					1	2		9
firebaseremoteconfig.googleapis.com	1				2		1	1	3		8
graph.facebook.com			2			1	3		1	1	8
googleapis.com				1			1	1	4		7
play.googleapis.com		1					4		2		7
googletagservices.com			3	2						1	6
e-droid.net			4								4
googleusercontent.com				3						1	4
cloudflare.com		1							1	1	3
datingfactory.com			3								3
exp.host		1							1	1	3
maps.googleapis.com		1					2				3
4tellus.com		2									2
d1wp6m56sqw74a.cloudfront.net		1							1		2
fbcdn.net				1					1		2
firebaselogging-pa.googleapis.com									2		2
firebasestorage.googleapis.com			2								2
googlevideo.com			1							1	2
hlthstar.com		2									2
myphonenumbers-pa.googleapis.com		2									2
sentry.io		2									2
Others	5	20	25	21	13	4	14	7	32	12	153
Total	9	48	70	42	23	5	35	12	71	26	341

Table 5: Top 10 trackers that receive traffic from 146 elderly apps

Tracker	# Apps
crashlytics.com	35
doubleclick.net	22
googlesyndication.com	12
google-analytics.com	8
googletagmanager.com	6
appsflyer.com	6
flurry.com	4
googleadservices.com	4
onesignal.com	3
branch.io	3

12/146 over HTTPS) to third-party domains. The most common parameters include phone model and OS build version. *Empowerji* sends CPU build, Android version and firmware version to AppsFlyer (third-party domain). *Homage*, *EZ Care* and *All Well Senior Care* send WiFi, cellular information, signal strength, and a flag to check if the device is rooted or not. *Seniority* sends email ID, device information (phone model and OS build), and the product details (that the user adds to the shopping cart or buys on the app) to a third-party analytics tracker (wzrkt.com) over HTTPS.

4.4 Improper Access Control

We found 4/146 apps with improper access control. *GoldenApp's* access control issues are due to insecure session management. As there are no authentication tokens or cookies in the requests, an attacker can replay the requests (even modify them) to create accounts in other users' names which can lead to misrepresentation or identity theft for the user. *POC EVV* contains a 5-digit "dcsId" parameter as the user ID in the requests which can be changed (by

a remote attacker) to get other users' data (e.g., phone number, home and office address, zip code). *Senior Discounts plus Coupons* has a 6-digit parameter for the user ID that can be modified to get any other user's email ID. *Damava* also has a similar issue where an attacker can fetch the user details using a GraphQL query and then modify the user ID to get other users' data (e.g., email ID, address, criminal record). The information disclosed in *Damava* could result in a full account takeover for both the patient as well as a caregiver. We also found that the appointment details query and mutation do not implement any access control in *Damava*; an adversary can view, modify and cancel any elderly patient's appointment. Moreover, given the appointment and caregiver details, the attacker can also impersonate a caregiver to harm the patient.

4.5 Improper Input Validation

9/146 apps are vulnerable to various injection attacks such as SQL/code injection, cross-site scripting. For example, *Senior Dating by Lauber*, *GoldenApp*, *Caring Village* and *Generations Homecare System* are vulnerable to reflected cross-site scripting attacks. An attacker can execute malicious JavaScript code to fetch elderly users' detail or to phish them. We note that for this attack to work, a victim would first need to click on a malicious link crafted by the attacker. *Empowerji*, *EZ Care* and *Tricella Health* are vulnerable to SQL injection attacks, allowing an adversary to view, modify and delete any elderly user's data. To verify the SQL injection vulnerability, we add an SQL query, in the intercepted request, which contacts a domain that is under our control. If we see the domain being contacted,

we confirm that the SQL query is being executed, and hence vulnerability to SQL injection attacks. *EllieGrid* and *Seniority* are vulnerable to code injection. For this attack, we added a JavaScript sleep function in the request body and then observed the response time. When there was a delay of 10 seconds for the response after the sleep command of 10 seconds, we confirmed the code injection vulnerability. This is a very serious issue that can lead to complete compromise of the application's data and functionality, and the server that's hosting the application [4]. Due to ethical reasons, we limit our attack in detecting this vulnerability. As there is no authentication secret on *EllieGrid* requests, the attacker can perform this attack remotely by constructing and sending the modified requests to the app's server.

4.6 Server-side Security Misconfigurations

We found 16/146 apps with various security misconfigurations. Apps such as *Doulike-senior*, *Carelinx*, *Pension Status Search Old Age Widow Handicap* and *Homage* transmit HTTP requests to modify an object via unprotected GET requests, and thus are vulnerable to Cross-Site Request Forgery (CSRF) attacks, mostly executed via sharing/clicking a malicious link. We found that *Over 40 Dating Mature* has a file path manipulation vulnerability where we placed user-controllable data (the file path on the app's server) into the URL path of the app's request that might be used on the server to access local resources (which may be within or outside the web root). With this vulnerability, an attacker can modify the file path to access different resources, which may contain sensitive information. For legal and

ethical reasons we did not test/validate this attack.

4.7 Dangerous App Permissions

App permissions are one of the most fundamental variables when testing the security of the device or the app. If the apps get important permissions such as read/write storage or read contacts, the app becomes very "powerful" as it can control many components without the user being aware. We classified permissions into 3 tiers - Dangerous permissions, Medium risk permissions, and Low-risk permissions. ACCESS LOCATION, READ CONTACTS, READ STORAGE, SEND SMS are examples of dangerous permissions. Dangerous permissions grant an app access to personal user data (e.g., user's location), or control over the user's device. They are only granted after explicit user consent. We found a total of 598 dangerous permissions in 118/146 apps, i.e., an average of 5 dangerous permissions per app. See Table 6. *Ianacare* (caregiver app) and *Life Assure* (companion app for a tracking device) had the maximum of 11 dangerous permissions (Call Phone, Camera, Write External Storage, Read External Storage, Read Calendar, Write Calendar, Read Contacts, Write Contacts, Read Phone State, Access Coarse Location, Access Fine Location, Get Accounts, Record Audio). Access Fine Location permission is needed if an app wants to know detailed information about the user's location, and respond accordingly. This is often used with advertising and location-based and social-network services like Facebook. Read Calendar allows an application to read the user's calendar data. Calendar events can, and often do contain contact information. The top 2 dangerous permissions found were Write

Table 6: 598 dangerous permissions asked by 118/146 elderly apps

Dangerous Permission	# Apps
Write External Storage	92
Read External Storage	91
Access Fine Location	84
Access Coarse Location	75
Camera	61
Record Audio	44
Read Phone State	39
Read Contacts	29
Call Phone	27
Get Accounts	22
Write Settings	9
Read Calendar	8
Write Calendar	8
Write Contacts	5
Get Tasks	1
Read Call Log	1
Receive SMS	1
Write Call Log	1

External Storage (92 apps) and Read External Storage (91 apps). Rarely used permissions found were Read Call Log (*BIG Phone for Seniors*), Receive SMS (*Homedoctor Protección Senior*), Get Tasks (*DAGPS*) and Write Call Log (*BIG Phone for Seniors*).

84/146 apps required Access Fine Location and 75/146 apps required Access Coarse Location permission. 61/146 apps asked for Camera permission, such as *Petralex*, *Walk to End Alzheimer's*, *My house of Memories*, *GoutDietRecipes*, *Oscar Senior*, *Senior Discounts*, *Seniority*, *Aveanna EVV*, *401(K) - Retirement Planning*. Apps with a significantly high number of risky permissions include *Ianacare*, *401(K) - Retirement Planning*, *Aveanna EVV*, *Oscar Senior*, *Senior Safety App*, *CrescendoConnect*, *Trusted Senior Care* and *ClearCareGo*. Fig. 2 shows the number of permissions asked by 146 apps as a heatmap.

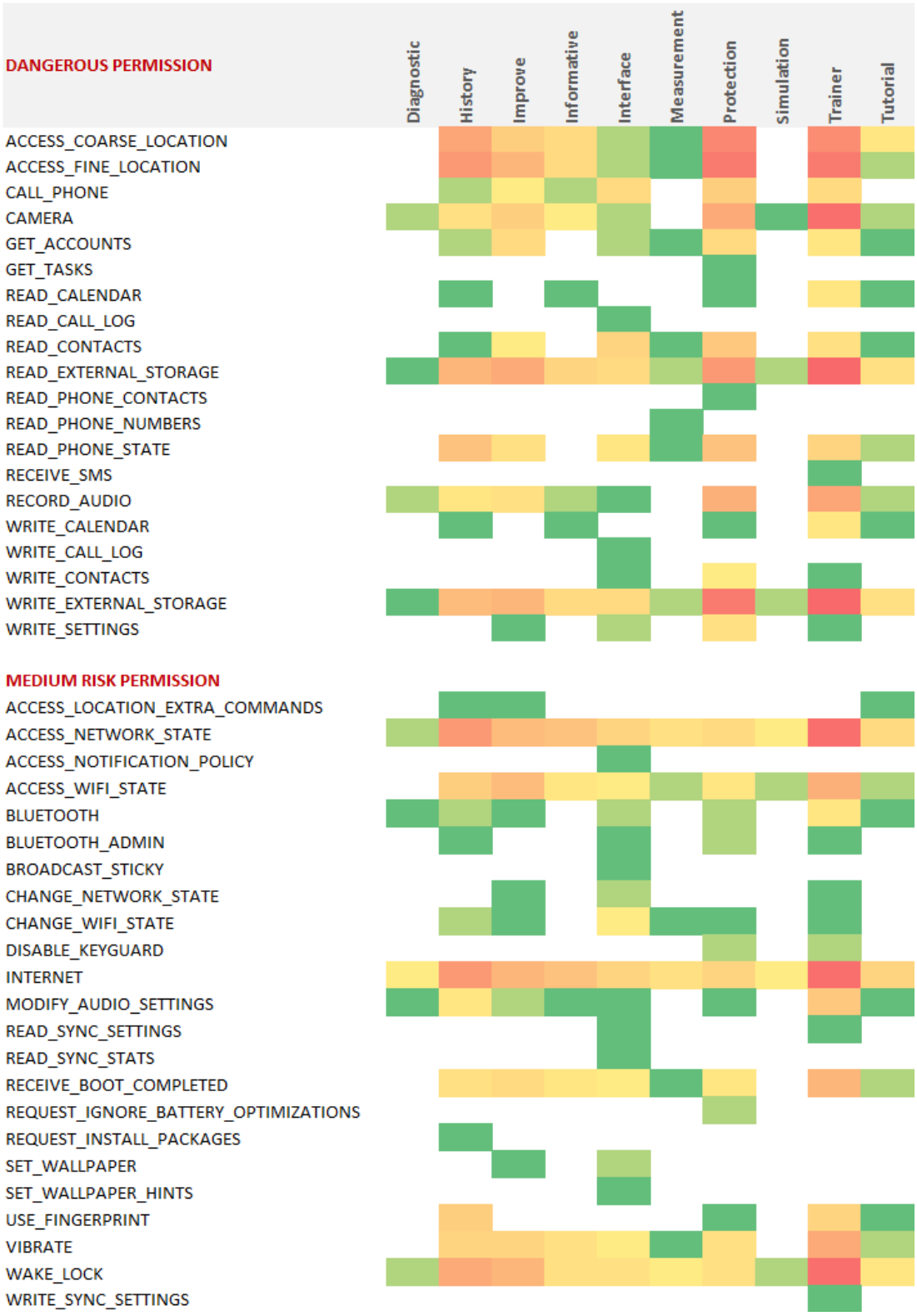

Figure 2: Heatmap of dangerous and medium risk permissions asked by 146 apps by app category

4.8 Third-Party Libraries and Permissions

Types of Libraries. We found 122 unique third-party libraries and a total of 1008 libraries in 127/146 apps, for various purposes: app development (93/122), analytics (6/122), advertisements (6/122), and social networking (2/122). See Table 10 in Appendix for complete list of 122 unique third party libraries. We found 34/146 apps with *Facebook* social media library and 14/146 apps with advertisement libraries, mainly *Google Ads* (9 apps) and *Unity3d Ads* (3 apps).

Libraries by App Category. A high number of total third-party libraries were found in 26 caregiver apps (218/1008 libraries), 20 EVV apps (162/1008), 16 location tracking apps (150/1008), 12 dating apps (107/1008) and 6 apps for Alzheimer's (55/1008). This shows that the elderly needing care, unwell and socially active may be more prone to privacy and data security issues arising via these third-party libraries. 48/146 apps have 10 or more unique third-party libraries. Examples of apps with a high number ($>$14) of unique libraries are *Knee Arthritis Exercises* (24), *Theora Link* (22), *Walk to End Alzheimer's* (19), *My SOS Family Emergency Alerts* (18), *Doulikesenior* (17), *Pension Status Search Old Age Widow Handicap* (16), *Tracki GPS* (15), *Empowerji* (15), *Alzheimer's Daily Companion* (15), *Huddle Health* (15), *Caring Village* (15). See Table 11 for the list of all apps with number of unique third-party libraries found in each one.

Kinds of Permissions Asked. We found 3 unique signature[9] permissions asked by the

[9]Enables communication between multiple apps of the same developer. Only granted if the requesting app is signed with the same certificate.

libraries (Dump, Write APN Settings, Write Secure Settings). We found the Dump permission for example used by *Firebase* (64/146 apps), *Glide* (41/146 apps) and *Facebook* (32/146 apps) third-party libraries. Write Secure Settings permission was asked predominantly by *Google Mobile Services* (62/146 apps) and *Firebase* (62/146 apps). This Development Aid library permission allows an application to read or write the secure system settings. This permission should only be seen on Android system apps (and possibly wireless carriers or hardware manufacturer pre-installed apps) [45]. Write APN Settings permission was asked by *Google Play* library (4/146 apps). See Table 7 for the number of unique libraries and permissions asked by the libraries categorized by "App Usage" and the app category.

4.9 Static Code Analysis

Static code analysis with MobSF shows that 93/146 apps can read/write to External Storage; 84/146 apps execute raw SQL queries which may expose them to SQL Injection attacks; 71/146 apps use weak hash functions; 36/146 apps have insecure WebView implementation; and 12/146 apps have insecure SSL implementation, a critical security issue. Apps with all these six concerns are *Alzheimer's Disease Pocketcard*, *Big Keyboard & Notifications*, *Over 40 - Find People 50*, *Pill Reminder & Medicine App*, *Doulikesenior*, *Senior Safety App* and *Silver 50 Dating*.

Table 7: Total permissions asked by app libraries

App Category	App Usage	# Apps	# Unique Libraries	# Permissions
Diagnostic	Brain test	1	11	22
	Hearing	2	3	13
History	EVV	20	48	476
Improve	Dating	13	43	326
	Social	2	9	31
Informative	Arthritis	6	34	167
	Pension	2	16	43
	Shopping	4	15	59
Interface	Launcher	11	21	115
Measurement	Care	3	21	100
	Exercise	6	23	158
Protection	Safety	7	49	109
	Pill	7	20	74
	Tracking	17	50	476
Simulation	Memory	3	16	57
Trainer	Caregiver	34	53	650
Tutorial	Alzheimer's	6	38	159
	Retirement	2	15	50
Grand Total		146	122	3085

4.10 Apps with an IoT Device

We acquired IoT devices that operate with 3/24 IoT companion apps: *EllieGrid*, *Carego Alphahom*, *Tuya SOS*, and tested them to understand the relationship between the device and app. We analyzed the app behavior for the remaining 21/24 IoT companion apps, to the extent possible without the IoT device, and found issues in 7/24 apps. *X-GPS Monitor*, *Family1st*, *BrickHouse TrackView* and *DAGPS* are IoT companion apps which help family members track their elderly loved ones via IoT devices. All these 4 apps had 3 main

issues: (1) improper authentication management, allowing an on-path attacker to sniff the username and password for full account takeover, (2) insecure session management, i.e., there is no use of authentication secrets in their requests, allowing an attacker to replay the requests, and (3) insecure PII transmission, leaking username and password in plaintext over HTTP. An on-path attacker can exploit these issues to track the exact location of users wherever they go with their IoT device due to full account takeover.

Tricella Health and *EllieGrid* are smart pill organizers which make medication management easier and are specifically designed to help the elderly by reminding them to take their pills on time (they consist of a pillbox and an app). *Tricella Health* has improper authentication management, where it is vulnerable to a remote OTP (3-digit number) brute-force attack during login and registration, leading to full account takeover. It also has improper input validation where its login requests are vulnerable to SQL injection attacks. These issues can lead to a remote attacker changing the user's medications causing the user to take the wrong medications, skip doses or overdose.

EllieGrid's physical pillbox is designed to store pills and receive reminders as ring notifications. The reminders and medications can be set up in its companion app. We found two major vulnerabilities in *EllieGrid*. Firstly, it offers a functionality to alert the elderly user's caregiver via email and phone, when the pillbox is not opened on time. We note that there is no access control present in this functionality, and a remote adversary can completely tamper with the associated caregiver's detail by using the caregiver profile setup option, which is present on the app UI. An adversary can enumerate a caregiver's ID by brute-forcing, and then supply it to modify the caregiver's email and send the alerts to an

attacker under his control, which would allow him to track the elderly users' activities and collect their pill taking habits; additionally, the legitimate caregiver will not receive any further notifications from the pillbox. Secondly, the *EllieGrid* solution offers a paid plan with additional functionalities for the elderly, such as viewing weekly adherence reports and adding a caregiver. Specifically, we found a parameter *subscriptionTypeId* from the user profile API, which sets the value of the current plan. An adversary can set this parameter's value to *premium* and upgrade their *EllieGrid* account for free.

We also found some vulnerabilities by following the static analysis approach in *Carego Alphahom*, which provides a personal alarm system for the elderly. In particular, the app is vulnerable to the Janus vulnerability [26], in which an adversary can prepend a malicious DEX file to an APK file while keeping its signature unaffected. Android versions 5.0 - 8.1 accept the file as a valid APK. During the radio analysis of this product, we noticed that the device is using WIFI to establish a connection with its' companion app. Upon setting up the Wireshark proxy and intercepting the communications, we found that the all of the traffic is encrypted and we did not find any sensitive information or malicious domains inside the encrypted traffic. The other two IoT devices: *EllieGrid* and *Tuya SoS*, use Bluetooth as radio communication. We let these two devices connect via Bluetooth, after which we extracted the snoop logfile using adb and checked the log packets for any sensitive information. We note that the sensitive functionalities are present only inside the traffic generated between the Android app and the server. The traffic between IoT device and the app mostly contain counter and signal values. In the dynamic analysis, we note that the app is using encrypted request parameters which cannot be tampered with easily.

4.11 Firebase Analysis

84/146 Android apps use Google Firebase as a backend service and we found 4/84 apps whose Firebase DB was exposed publicly. For ethical reasons and to protect other customers' privacy, we created elderly accounts on the four apps. Then, we updated the Firebase scanner to automatically search for our test data in its response and record the leaked information from our own account. 2/4 apps (*CogniFit* and *Carely*) fixed this issue during the time of our testing. For *UnitedHealthcare EVV Tennessee* and *Amerigroup EVV Tennessee*, at the time of testing, we could not see any sensitive data being stored on their databases.

4.12 Inadequate Privacy Policies

Privacy policies may be difficult to understand - even yielding different interpretations — by average users [20]. We select a group of 108/146 apps to perform our privacy policy analysis on.

We retrieved 87% of privacy policies i.e., for 94/108 apps and 14 apps did not have a policy (as of January 2022). We summarise the results of our privacy policy analysis along parameters defined by Polisis. See Table 8.

Data Collection Practices. 76% of privacy policies (83/108) mentioned some kind of data being collected from users. IP Address and Device Ids (68/108), Cookies and Tracking Elements (63/108) and User Online Activities (61/108) were the three most common types of data collection declared by apps. 31/108 of app policies mention the collection of Financial

Table 8: Data of privacy policies for 30/108 apps with the maximum security flaws

Legend: C - Californians, E - Europeans, I - International, 1P - First party, 3P - Third party, X - No mention in policy, Unsp. - Unspecified in policy, 1P mktg. - First party marketing communication / newsletters, Deactivate - User can deactivate account, Part deletion - User can delete some data in account

App Name	# of Type of Data Collected	# Data shared 3P	User Choices	Data Security	# of Type of Data Retention	Audience Data	Data Edit Rights
40+ Senior Dating	9	1	1P mktg.	Medium	Unsp.	C,E,I	Deactivate
401(K) retirement	X	X	X	X	X	X	X
All Well Senior Care	11	6	1P cookies, 3P ads	Medium	1 listed	C,E	Yes
Alzheimer's Daily	8	3	1P collection	X	X	C	Yes
Amerigroup EVV	5	2	X	Low	1 listed	E	Yes
Big Launcher - Old Age	8	6	X	Low	X	X	Part deletion
CareLinx	X	X	X	X	X	X	X
Caring Village	13	7	X	High	X	X	Yes
Cougar Dating	11	7	X	X	X	X	X
Crescendo	5	3	1P cookies	X	X	X	X
Damava	2	1	1P cookies	Low	X	X	X
Doulikesenior	4	6	1P cookies, emails	Medium	1 listed	X	Deactivate
EllieGrid	7	3	1P mktg.	Medium	1 listed	C	Part deletion
EZ Care	6	4	1P cookies	X	X	C	yes
FlirtMatures	X	X	X	X	X	X	X
Generations Homecare	11	7	1P mktg., 3P ads	Medium	3 listed	I	Yes
GoldenApp	X	X	X	X	X	X	X
HelpAge SOS	2	1	X	X	X	X	X
Homage	8	3	1P cookies	Medium	2 listed	I	Full deletion
Empowerji	5	1	X	X	Unsp.	X	Yes
Mobile Caregiver	12	5	X	Medium	X	C, I	Part deletion
Oscar Senior	4	3	1P analytics	Medium	1 listed	I	Yes
Over 40 Dating	8	3	1P search	Low	2 listed	X	Full deletion
POC Evv	6	3	1P cookies	Low	Unsp.	C	X
Rosemark	5	3	X	Medium	X	X	Yes
Senior Dating2	2	4	X	X	X	X	Yes
Senior Dating	5	1	1P cookies	Low	1 listed	X	X
Senior discounts	X	X	X	X	X	X	X
Senior Safety App	2	1	1P emails	Medium	2 listed	X	X
Seniority	3	1	1P cookies	Low	X	X	Part deletion

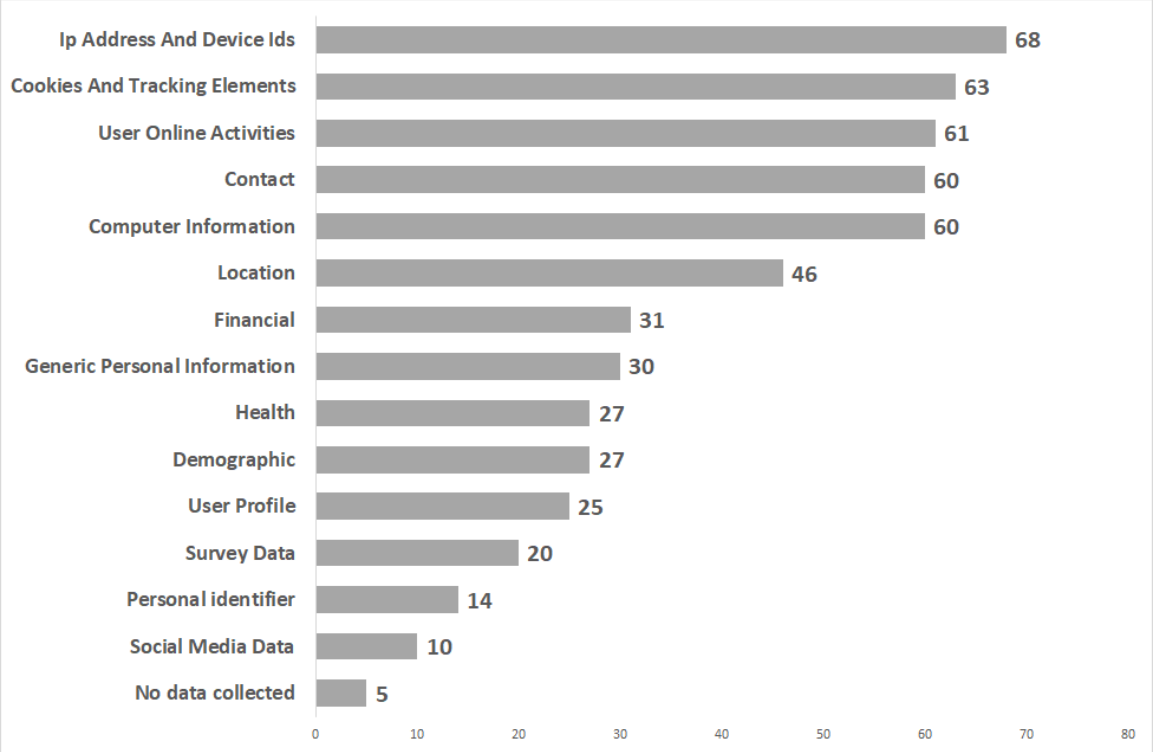

Figure 3: Types of data collected as mentioned by 83/108 app policies

data. See Fig. 3 for types of data collected as mentioned in the privacy policies of 83/108 apps. The common reasons cited for this data collection include: "Service improvement", "Analytics research" and "Advertising & Marketing".

Third-Party Data Sharing Practices. Many mobile apps depend on third-party libraries that provide crash reporting, analytics, development, or advertising services. When any embedded third-party service collects or processes personal data from users, both GDPR and CCPA require developers to list them on the privacy policy of the app. We found that 79/108 of privacy policies mentioned that data is shared with third-party service providers, and common reasons for data sharing are "Service operation and security", "Legal requirement", "Analytics research", "Advertising", "Marketing" and "Service feature".

User Choices. 64/108 app policies mentioned choices given to the user for first-party use of data, or third-party sharing. Examples of User Choices are “First Party - accept / refuse cookies” (30/108 apps), “First Party - opt-out of marketing communication / promotional emails” (19/108 apps), “First Party - accept / refuse location data” (3/108), “Third party - accept / refuse cookies” (5/108), “Third Party - opt-out of personalised ads” (4/108), “Third Party - opt-out of analytics” (2/108).

Security Practices. Polisis detects 6 types of security practices (along with generic security statements), which are; data access limitation, privacy security program, privacy review audit, secure data storage, secure data transfer and secure user authentication. We consider the data security to be low when the policy only mentions any 1 or 2 types, we consider the data security to be medium when the policy mentions 3 or 4 types, and we consider the data security to be high when the policy mentions 5 or 6 types. 57/108 of policies make generic security statements, “we protect your data” or “we use technology/encryption to protect your data”. Only 25/108 policies mention that data is accessible to employees/third parties on a need-to-know basis. 29/108 of policies mention secure user authentication, e.g., login to a user account, is encrypted/secured. Only 22/108 apps have Secure Data storage and 9/108 have a Privacy Review Audit in place. 40/108 policies did not have any mention of security practices. See Fig. 4.

Data Retention. 48/108 policies mention retention of various data. Data that is retained as per the app policies are “Other data” (41/108), “Generic Personal Information” (17/108), “Contact” (14/108), “User Profile” (7/108), “Location” (4/108) and “Financial” (4/108). Reasons cited for data retention are *“Legal requirements”* and *“Service operation and*

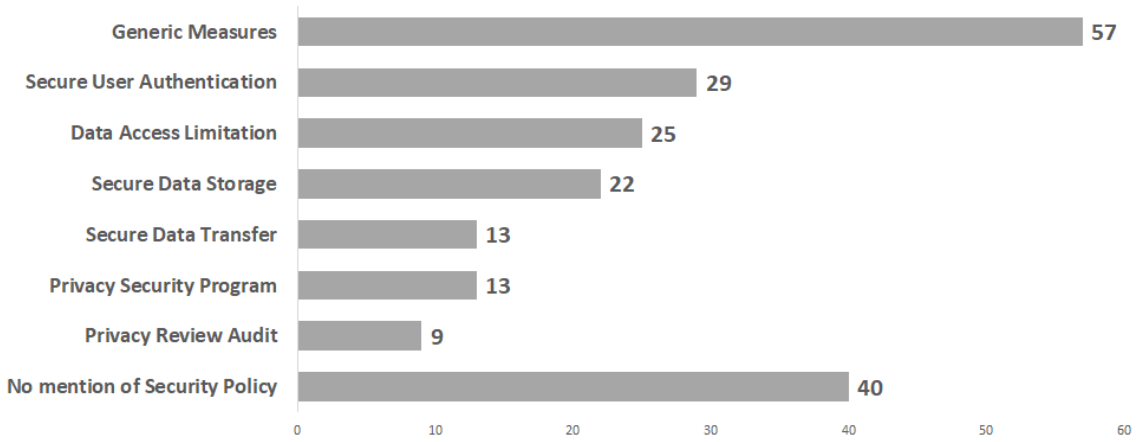

Figure 4: Security practices adopted as mentioned by 83/108 app policies

security".

Specific Audiences. Some audiences such as Californians are entitled to be disclosed a list of third parties with whom their data has been shared, e.g., for direct marketing purposes; the users can then exercise their rights under the California Consumer Privacy Act (CCPA). EU citizens can demand how their data is treated when transferred across borders as per EU General Data Protection Regulation (GDPR). 17/108 policies mentioned how data from EU users are treated and 19/108 mentioned about CCPA.

Right to Edit. 51/108 privacy policies mentioned that users had rights such as opt-out of data collection, access to data, edit, rectification, and erasure. Policies offered the right to edit user account data (33/108), partially delete an account (25/108), and deactivate an account (7/108).

Policy Change. 55/108 policies mentioned that there could be (mostly unspecified) changes to the policy. Policies mentioned that changes will be notified on-site (26/108), communicated by email (13/108), or updated on site (24/108).

Chapter 5

Case Studies and Practical Attacks

In this chapter, we discuss a few selected apps and highlight the practical attacks that can potentially be executed by exploiting the vulnerabilities we found. We again need to consider that since this app would be used by the elderly who are not familiar with navigating through Android apps, there is a high likelihood that the users can be fooled into clicking fraudulent links in messages, spam emails, downloading potentially dangerous files or apps and unknowingly disclose passwords or credit card credentials, while malicious processes running in the background.

5.1 Empowerji

This technology learning app[10] (10,000+ installs on the Google Play store) helps the elderly learn how to use everyday apps. Login uses email/mobile number. One can learn from a library of multilingual learning videos and seek assistance at every stage. Firstly, we noticed

[10] https://play.google.com/store/apps/details?id=com.app.empowerj

that all the traffic for this app is transmitted in plaintext over HTTP. This can lead to an on-path sniffing attack that obtains PII details (name, email id, password, phone number) which can also lead to a full account takeover. Secondly, we found 10 requests which are vulnerable to SQL injection. We used sqlmap [40] to confirm that we can take over the database server due to the SQL injection vulnerabilities. We did not perform the exploitation of this attack due to ethical and legal reasons. The app also sent device information and usage data to AppsFlyer (third-party domain). This included device information like CPU build, Android version, and firmware version which can help attackers profile the user.

5.2 Seniority: E-Store For Senior Citizens

This app[11] (10,000+ installs on Google Play store) offers a large online shopping portal for senior citizens. *Seniority* app has an elder-friendly interface along with multiple easy-to-use features, which makes online shopping easier for senior citizens. We found 2 fundamental issues with this app during our dynamic analysis. Firstly, during login, if the user logs in using their mobile number, they will consequently receive a 4 digit one-time password (OTP) on their mobile number. As the app does not implement any form of rate limiting[12], one can easily perform a brute-force attack to obtain the correct OTP. This will lead to a full account takeover (wherein user information e.g., address, phone numbers, credit card details can be obtained), and the attacker can perform this remotely. Secondly, this app is

[11] https://play.google.com/store/apps/details?id=com.org.seniority.application

[12] Rate limiting is a strategy for limiting network traffic. It puts a cap on how often someone can repeat an action within a certain timeframe – for instance, trying to log in to an account. (https://www.cloudflare.com/en-ca/learning/bots/what-is-rate-limiting/)

also vulnerable to code injection (at the login page). As mentioned in Section 4.5, we add a JavaScript sleep function in the request body and then observe the response time. This is a very serious issue that can lead to complete compromise of the application's data and functionality, and maybe of the server that is hosting the application, but again due to legal reasons, we limit our attack in detecting this vulnerability. We also see the app sharing the user's data like email ID, device information (phone model and OS build), and the product details that the user adds to the shopping cart or buys on the app to wzrkt.com (a third-party analytics tracker). This information can be used to provide personalized ads to the elderly user or launch phishing attacks.

5.3 Damava

Damava is a platform offering solutions to hire professional caregivers. In particular, it offers two Android apps, one for the patient who can register, search for caregivers and make an appointment with them.[13] The second app is for the users who want to become the caregiver, who can sign-up, upload documents to verify, and start to give their caregiving services through this app. We found three major issues: (1) The app for patients uses GraphQL and exposes the documentationthrough an introspection query. This documentation has all the API queries and mutations that a user can make. The GraphQL query to fetch user details has a vulnerable access control implementation, which can be exploited by a remote attacker just by knowing the victim's user ID. The user detail query leaks sensitive information such as authentication token, email, address, and criminal record (a

[13] https://play.google.com/store/apps/details?id=com.damava.damava_helper, 5,000+ active users.

Boolean value). The information disclosed results in a full account takeover for both the patient as well as a caregiver; (2) We found that the appointment details query and mutation do not implement any access control. An adversary can view, modify and cancel any elderly patient's appointment. Moreover, given the appointment and caregiver details, the attacker can also impersonate a caregiver to harm the patient; (3) We found that the caregiver app is using AWS S3 to store private documents of the caregivers when they register on the app. Although the S3 bucket is not publicly exposed, we found a vulnerable GraphQL query exposing both the accessKey and secretKey. A malicious attacker can use these keys to access the S3 bucket and download sensitive documents of the registered caregivers.

5.4 POC EVV

This app[14] (10,000+ installs on the Google Play store) is developed for caregivers to help them view their schedules and report the completion of their visits. It requires the user to enter 2 separate numerical values; (1) a 6-digit user ID and (2) a 4 digit PIN to login into the app. As the requests are transmitted in plaintext over HTTP, any on-path attacker can sniff the traffic and obtain these 2 codes to log in and perform a full account takeover. We also noticed that the private messages sent between the caregiver and his/her supervisor are in plain HTML. A more worrying issue we find is that we could get any user's information on the platform by changing the requests. The "dcsId" parameter in the request header is a 5 digit number that is assigned to individual users on the platform. We could intercept the request, and modify this parameter (for example, increment the parameter by 1) and obtain

[14] https://play.google.com/store/apps/details?id=com.aquila.poc

other users' private information in the response. This information includes the zip code, address, city, street, name, location type, office address, state, and phone number. This attack can also be performed remotely by the attacker with an active platform subscription. Also, as the requests are sent over HTTP, the authentication secret (session ID) is left exposed. The session IDs expire in approximately 3 minutes, but an on-path attacker can easily get the new IDs from the network.

5.5 Elderly Dating Apps

40 Plus Senior Dating[15] and *Cougar Mature Women Dating App*[16] (both with 50,000+ installs on the Google Play store) are dating apps designed for elderly people. The major issue with these apps is that all the network traffic goes in plaintext HTTP. This means that sensitive user information like email ID, password, date profiles the user is looking at or liking, and even private messages between users can be sniffed by an on-path attacker. This cannot only lead to a full account takeover but the attacker could also obtain private messages and user preferences which can be used as leverage in any form of blackmailing or threat. *40 Plus Senior Dating* also leaks private information like the user's exact location to adrta.com which is a third-party adware program.

Cougar Mature Women Dating App also leaks other users' privacy preferences on the platform (in the traffic responses) which is again a privacy breach.

We also found 4 apps (*Senior Dating: Date Mature Singles, Dating for 50 plus Mature*

[15] https://play.google.com/store/apps/details?id=app40.plusdating
[16] https://play.google.com/store/apps/details?id=com.cougardating.cougard

Singles, SeniorsHug - Chat & Meet with Seniors Dating, DateMyAge: Chat Meet Date Mature singles online), where an attacker can check if any user has an account on these apps or not, by using the "forgot password" button in each app (needs only the email ID). As dating is a sensitive topic for many users in any age group, just knowing the fact that a user is active on these platforms can be problematic.

5.6 All Well Senior Care

All Well Senior Care[17] (50,000+ installs on the Google Play store) provides 24x7 health and well-being reports for seniors, elderly parents, or loved ones. The monitoring user can get hourly updates, instant reports, set up activity alerts, and communicate better. Similar to *Seniority* app, there is an OTP brute force attack possible. During login, the users need to input their registered email ID or mobile number and they will consequently receive a 3 digit OTP on their mobile number or their email ID. Again, as the app does not implement any form of rate-limiting we can perform a brute force attack to obtain the correct OTP which leads to a full account takeover. After this, the attacker can obtain the user's health data (e.g., heart rate, blood pressure, etc.), wellness data (wake up time, steps taken, etc.), see all the hourly updates the user is providing to her caregiver, the location of the user, all the health charts which are saved on the user's account, and even the private messages of the user with their caregiver or their care group (containing multiple users in one group).

[17] https://play.google.com/store/apps/details?id=com.atman.allwell

5.7 Big Launcher - Launcher For Old Age People

This app[18] (50,000+ installs on the Google Play store) provides an interface with big icons and big letters to help older people to see easily. This app leaks the user's exact coordinates in plaintext over HTTP to a third-party domain openweathermap.org. This domain's API is used to show the weather information to the user in the app, but unfortunately, any on-path attacker can sniff the exact location of the user, which may compromise the physical security of the elderly person. So, even an app that is as simple as a basic launcher, can lead to privacy and security risks for the user. *Big Launcher* app leaks the user's exact coordinates in plaintext over HTTP to a third-party domain openweathermap.org

5.8 Big Keyboard & Notifications - Senior Home Screen

Specifically designed for Elderly users, Big Keyboard & Notifications - Senior Home [19] is the easiest way to view all the important apps on your phone. It enlarges everything to make the screen easier to read and use. Large buttons make it quick to select the right icon and app every time and the Big & Easy Keyboard makes typing and texting convenient. We found 6 PII leaks over 50.10% HTTP traffic: Android ID, Installed Apps (7 domains), Board info, Android Serial, Private IP (2 domains), Build Fingerprint (4 domains) to 15 domains including 45tu1c0.com, appsflyer.com (marketing analytics), ncvcontent-api.co, branch.io, facebook.com, crashlytics.com, evnttrck.io, doubleclick.net (marketing

[18] https://play.google.com/store/apps/details?id=com.phongphan.launcher.older
[19] https://play.google.com/store/apps/details?id=com.myhomescreen.access&hl=en

platform), flurry.com (marketing analytics), nexage.com (ad platform), oath.com (marketing campaigns). The app has Read External Storage, Write External Storage, Read Contacts, Read Phone State dangerous permissions and Internet, Set Wallpaper, Receive Boot Completed, Set Wallpaper Hints, Bluetooth, Bluetooth Admin, Read Sync Settings, Read Sync Stats, Access Wifi State, Access Network State, Change Wifi State, Change Network State medium risk permissions. Firebase Messaging Service, Firebase Authentication activity and Recaptcha activity are not protected. The PII leaks can be captured by a MITM attack, malicious actors can try to access credentials through Firebase or other Installed apps with same credentials, using private IP address and device profile. Malware can access/modify storage card information, and /or install bot malware by manipulating Bluetooth and Wifi access points and using Trojaned applications. The many ad/marketing domains can lead to personalized ads, email spamming and phishing attacks since mobile ads can be served by many unknown intermediaries. Vulnerable elderly users of this app are likely to have vision problems, which may be exploited by serving ads with the fine print to fool them into clicking paid services.

Chapter 6

Discussion and Conclusion

6.1 Discussion

Our comprehensive analysis of apps for the elderly reveals a large number of undesirable risks, that arise from privacy and security issues, backend vulnerabilities and dangerous permissions. These applications request a large number of dangerous permissions (especially read and write external storage, Camera and user location) and reveal vulnerabilities due to numerous high risk data leaks of Android ID, Android serial, Private IP and Account information. Apps with user interaction between apps have significantly more high risk leaks. We find that third-party libraries have high-privilege signature permissions granted (Dump, Interact Across Users and Backup) that are to be used only by app developers and they can also gain dangerous permissions that can allow the 3PL to exploit device components and access stored data, or lead to malicious apps/adversaries gaining control of device components and access sensitive user information. Suppose a 3PL (e.g. Facebook,

Inmobi, Smaato, Unity3D) is embedded in a few apps on the smartphone, it will gain the dangerous permissions allowed for each of these apps, ending up with a large number of dangerous permissions. This 3PL can then access more information than intended, the user being unaware. A large number of marketing analytics, social network and advertisement third-party libraries collect personal data without explicitly being mentioned in the privacy policies. These 3PL provide a fine profile of the vulnerable user who can be then targeted with more ads that can lead to unwanted purchases or behavior, spamming or phishing attacks.

We find HTTP traffic without encryption in many apps, with PII leaks that can be sniffed by MITM attackers and used to extract meaningful financial information or sensitive messages (e.g. poor health condition, financial plans, legal issues).

Our analysis of the apps' privacy policies shows that their data collection practices are usually vague and almost none mention the specific duration that the data would be stored on their servers. The regulatory compliance of many of these apps is suspect as there is often no mention in the policies of specific audiences such as children. Security measures to safeguard data transfer and storage are not consistent nor adequate.

Android users are not clearly informed about third-party tracking software tracking and advertising services embedded in apps, the types of data they collect from them, the capabilities and the amount of control they have on their devices, and the partnerships that allow information to be shared and control to be given to various other companies through custom permissions, backdoors, and side-channels. This necessitates a new form of privacy policy to be defined and enforced to ensure that private information is at least

communicated to the user in a clear and accessible way, accompanied by mechanisms to enable users to make informed decisions about how or whether to use such devices.

Given the severe privacy risks revealed by our analysis, we suggest that privacy implications should be fundamentally considered in any app designed for the elderly users. App stores should take extra measures for verifying that applications directed at them comply with current legislation and treat their data with extreme care using security measures, such as data encryption.

In this thesis, we show that applications for the elderly have a very high density of trackers on average and almost all apps are vulnerable to one or more threats, and we highlight in detail the security flaws in 30/146 apps. Security and privacy misbehavior is a common trend in Android apps as they often request more permissions than needed and include a large number of third-party libraries. Trust would be broken when these services treat their data carelessly, share it with third-parties, or send it across the Internet unencrypted. Considering the financial importance and growing adoption of technology by the elderly, developers have to sensibly shoulder the immense responsibility of providing apps that offer zero-risk and peace of mind.

6.2 Limitations and Future Work

As Google Play Store does not have a defined “Elderly” or “Senior” app category, our app search is limited to the keywords used. A major limitation we faced during our dynamic analysis was the inability to create accounts for 67/115 of our apps because the companies

either make accounts for the users beforehand (and provide access information) or the apps will validate the user's information (e.g., medical insurance numbers, and organization email IDs, which we cannot provide in our test accounts) before creating the account. This was most applicable for the EVV and caregiver apps. For those apps, we conduct a limited dynamic analysis of pre-login application behaviors. We also did not test any paid apps

We also believe that our analysis should not stop here and we can explore deeper in this topic. As we only tested Android apps (and some IoT devices) for elderly people, future work could include testing of iOS apps and websites for elderly people. We also think that to perform a large scale analysis, it would be easier to develop an automated framework which perform UI automation in the app and also collects traffic information and leaks. We noticed that there are lots of privacy and data laws for other demographics or users (e.g., Children's Online Privacy Protection Act (COPPA) [20], Health Insurance Portability and Accountability Act (HIPAA) [21], General Data Protection Regulation (GDPR) [22]), but there isn't one for elderly users. As seen from previous research, we believe that elderly users are also one of the top demographics to be vulnerable online, hence there should be the type of laws specifically for elderly apps to protect elderly users more.

6.3 Recommendations for Developers

Companies and developers may use our test framework to check their elderly apps for security and privacy issues. This will help avoid several known issues. Such tests must be

[20] https://www.ftc.gov/legal-library/browse/rules/childrens-online-privacy-protection-rule-coppa
[21] https://www.hhs.gov/hipaa/index.html
[22] https://gdpr-info.eu/

performed on every new app versions, and after any server-side modifications as well. Developers may also go through publicly available guidelines and best practices, such as from Android Developers[23] and OWASP mobile security guidelines.[24] As these documentations are often updated by security professionals, following these practices can ensure that the applications can avoid the most common security and privacy management issues. More specific to the apps that we analyzed, and the security and privacy problems we observed, we would also suggest the following recommendations for developers to follow.

1. All the communication between the application and its server should be encrypted. For example, all network traffic should be transmitted over HTTPS, with the proper configuration of HTTPS on the app and the server. This simple step will prevent attackers from eavesdropping the network traffic (which usually contains sensitive user information), and will help preserve the confidentiality of the user's data. The same measure must be adopted on all server end-points that the app may communicate with (including any third-party servers, if there is a need for such communication).

2. When using one time passwords (OTPs) for authentication during login or registration, which is used as part of two-factor identification (2FA) or multi-factor authentication (MFA), the developers need to use OTPs of at least 6 characters long, and implement rate limiting[25] on the network requests. This will prevent OTP brute force and DoS attacks on the application.

3. To prevent insecure session management issues, developers should follow the basic

[23] https://developer.android.com/topic/security/best-practices
[24] https://owasp.org/www-project-mobile-security/
[25] See e.g., https://www.cloudflare.com/en-ca/learning/bots/what-is-rate-limiting.

practices including: using HTTPS for communication, making sure that new session tokens are being assigned at every new login session, and expiring the session tokens after a period of user inactivity (e.g., after 5 minutes). Also, the session tokens must be unique, long, and random to prevent any guessing attacks.

4. To prevent access control issues, the developers must force all sensitive requests to go through access control checks and allow only requests with proper access control tokens—e.g., the provided session ID is valid at the time when the request is received and the session ID also belongs to the requesting user.

5. Developers should limit the collection, storage, and transmission of user data to what is strictly necessary. For instance, the apps should not store PII which is not required for the its functionality. The apps should also allow the user to selectively opt-out of the data collection for certain features (perhaps less important for specific user groups). Developers should limit the usage of trackers and tracking SDKs in apps intended for the elderly. Some SDKs also have limited modes of operation where they do not collect as much data. These limited features should be used where possible. The best practice is the complete avoidance of using any tracking services, which also remain beyond the reach of app developers.

6. Developers should make sure to only use Android resources/permissions which are necessary for the functionality of the app, and explain the necessity to users. Over-privileged apps, if compromised or exploited, will be more harmful to user privacy (and affect the company reputation as well).

7. Developers should not save user data on the phone's internal service, so that an attacker with physical access to the device cannot get the user data stored on the device. Similarly, the use of any hard-coded or fixed credentials must be avoided, e.g., to process/transmit/store sensitive user or service data. Such values can be easily reverse-engineered and exploited. Also, if the app uses Firebase backend, developers need to ensure that they do not expose their sensitive Firebase API keys (e.g., hard-coding in the app), and protect the contents of that database (i.e., not accessible publicly), as an attacker can obtain the Firebase keys and server URLs from the APK file.

8. As we observed, privacy policies for elderly apps are also very similar to other apps – i.e., long and difficult to understand. For elderly people, some of who might have limited visual/comprehension capabilities, these policies should be tailored to this specific user group so that they can understand the policies more clearly.

6.4 Conclusion

We presented a comprehensive analysis of 146 Android apps that are intended to assist elderly people. Our methodology included dynamic analysis of traffic domain flows, trackers, leaks, and permissions, static analysis of third-party libraries for risky permissions and vulnerable backend issues using various automated as well as manual tools. We reveal individually many red flags in 30/146 apps and how they are most likely to be a security risk. But also, in a wider sense, we have noticed trends in apps' permissions and domain

flows which show us how some companies, third-party libraries, or permissions dominate the segments. This is why, as mentioned before, we think the analysis should not stop here, as we can delve even deeper to find more flaws and vulnerabilities. This will create a safe environment for the elderly to have the peace of mind that their new smartphones are safe and they have one less thing to worry about.

Bibliography

[1] Ionut Arghire. Thousands of mobile apps leak data from Firebase databases, 2018. Online article. https://www.securityweek.com/thousands-mobile-apps-leak-data-firebase-databases.

[2] Jacquelyn Bengfort. Senior care and mobility: Why smartphones and tablets make sense, 2019. Online article. https://healthtechmagazine.net/article/2019/11/senior-care-and-mobility-why-smartphones-and-tablets-make-sense.

[3] Census.gov. We, the American elderly, October 2021. Online Article. https://www.census.gov/library/publications/1993/dec/we-09.html.

[4] Hyunwoo Choi and Yongdae Kim. Large-scale analysis of remote code injection attacks in Android apps. *Security and Communication Networks*, 2018:1–17, 04 2018.

[5] CNBC.com. Here's how online scammers prey on older Americans, and what they should know to fight back, Nov 2019. Online article. https://www.cnbc.com/2019/11/23/new-research-pinpoints-how-elderly-people-are-targeted-in-online-scams.html.

[6] Louis Columbus. Roundup of internet of things forecasts, 2017. Online article. https://www.forbes.com/sites/louiscolumbus/2017/12/10/2017-roundup-of-internet-of-things-forecasts/?sh=4f00f1d11480.

[7] António Cunha, Evandro Cunha, Emanuel Peres, and Paula Trigueiros. Helping older people: is there an app for that? *Procedia Computer Science*, 100:118–127, 2016.

[8] Jordan Davidson and Christoph Schimmele. Evolving internet use among Canadian seniors, 2019. Online article. https://www150.statcan.gc.ca/n1/pub/11f0019m/11f0019m2019015-eng.htm.

[9] Michalis Diamantaris, Elias P Papadopoulos, Evangelos P Markatos, Sotiris Ioannidis, and Jason Polakis. Reaper: real-time app analysis for augmenting the Android permission system. In *Proceedings of the Ninth ACM Conference on Data and Application Security and Privacy*, pages 37–48, 2019.

[10] Easylist.to. Easylist, 2022. https://easylist.to/.

[11] Alisa Frik, Leysan Nurgalieva, Julia Bernd, Joyce S. Lee, Florian Schaub, and Serge Egelman. Privacy and security threat models and mitigation strategies of older adults. In *Proceedings of the Fifteenth USENIX Conference on Usable Privacy and Security*, SOUPS'19, page 21–40, USA, 2019.

[12] Clint Gibler, Jonathan Crussell, Jeremy Erickson, and Hao Chen. Androidleaks: Automatically detecting potential privacy leaks in Android applications on a large scale.

In *International Conference on Trust and Trustworthy Computing*, pages 291–307. Springer, 2012.

[13] Github.com. Graphiql, January 2022. Online Article. https://github.com/graphql/graphiql.

[14] Michael C Grace, Wu Zhou, Xuxian Jiang, and Ahmad-Reza Sadeghi. Unsafe exposure analysis of mobile in-app advertisements. In *Proceedings of the fifth ACM conference on Security and Privacy in Wireless and Mobile Networks*, pages 101–112, 2012.

[15] Hamza Harkous, Kassem Fawaz, Rémi Lebret, Florian Schaub, Kang G Shin, and Karl Aberer. Polisis: Automated analysis and presentation of privacy policies using deep learning. In *27th USENIX Security Symposium (USENIX Security 18)*, pages 531–548, 2018.

[16] Helpadvisor.com. What age is considered elderly?, January 2022. Online Article. https://www.helpadvisor.com/retirement/what-age-is-considered-elderly.

[17] Simon Hill and Mark Jansen. Android vs. iOS: Which smartphone platform is the best?, April 2021. Online article. https://www.digitaltrends.com/mobile/android-vs-ios/.

[18] Jeff Hoyt. Senior citizen apps., 2020. Online article. https://www.seniorliving.org/cell-phone/apps/.

[19] Kit Huckvale, José Tomás Prieto, Myra Tilney, Pierre-Jean Benghozi, and Josip Car.

Unaddressed privacy risks in accredited health and wellness apps: a cross-sectional systematic assessment. *BMC medicine*, 13(1):1–13, 2015.

[20] Carlos Jensen and Colin Potts. Privacy policies as decision-making tools: an evaluation of online privacy notices. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, pages 471–478. Springer, 2004.

[21] Alekh Jindal and Samuel Madden. Graphiql: A graph intuitive query language for relational databases. In *2014 IEEE International Conference on Big Data (Big Data)*, pages 441–450. IEEE, 2014.

[22] Brittne Nelson Kakulla. Older adults keep pace on tech usage. *AARP Research*, 2020. https://www.aarp.org/research/topics/technology/info-2019/2020-technology-trends-older-americans.html.

[23] P. Kapoor, R. Pagey, M. Mannan, and A. Youssef. Silver surfers on the tech wave: Privacy analysis of Android apps for the elderly. In *18th EAI International Conference on Security and Privacy in Communication Networks (Securecomm 2022)*, Kansas City, United States, October 2022.

[24] Wiebke Maaß. The elderly and the internet: How senior citizens deal with online privacy. In *Privacy online*, pages 235–249. Springer, 2011.

[25] Medium.com. Google auth vulnerability, October 2020. Online Article. https://medium.com/swlh/google-firebase-authentication-vulnerability-245050cb7ceb.

[26] Medium.com. Exploiting apps vulnerable to janus (cve-2017–13156), 2021. Online

article (26 March 2021). https://medium.com/mobis3c/exploiting-apps-vulnerable-to-janus-cve-2017-13156-8d52c983b4e0.

[27] Benjamin Morrison, Lynne Coventry, and Pam Briggs. How do older adults feel about engaging with cyber-security? *Human Behavior and Emerging Technologies*, 3(5): 1033–1049, 2021.

[28] Ian Muscat. What are injection attacks, April 2019. Online article. https://www.acunetix.com/blog/articles/injection-attacks.

[29] OECD.org. Elderly population, January 2022. Online Article. https://data.oecd.org/pop/elderly-population.htm.

[30] Daniela Oliveira, Harold Rocha, Huizi Yang, Donovan Ellis, Sandeep Dommaraju, Melis Muradoglu, Devon Weir, Adam Soliman, Tian Lin, and Natalie Ebner. Dissecting spear phishing emails for older vs young adults: On the interplay of weapons of influence and life domains in predicting susceptibility to phishing. In *Proceedings of the 2017 chi conference on human factors in computing systems*, pages 6412–6424, 2017.

[31] Charlie Osborne. Google firebase messaging vulnerability allowed attackers to send push notifications to app users, August 2020. URL https://portswigger.net/daily-swig/google-firebase-messaging-vulnerability-allowed-attackers-to-send-push-notifications-to-app-users.

[32] Sylvia E Peacock and Harald Künemund. Senior citizens and internet technology. *European journal of ageing*, 4(4):191–200, 2007.

[33] Abbas Razaghpanah, Rishab Nithyanand, Narseo Vallina-Rodriguez, Srikanth Sundaresan, Mark Allman, Christian Kreibich, Phillipa Gill, et al. Apps, trackers, privacy, and regulators: A global study of the mobile tracking ecosystem. In *The 25th Annual Network and Distributed System Security Symposium (NDSS 2018)*, 2018.

[34] Jingjing Ren, Martina Lindorfer, Daniel J Dubois, Ashwin Rao, David Choffnes, Narseo Vallina-Rodriguez, et al. Bug fixes, improvements, ... and privacy leaks - a longitudinal study of PII leaks across Android app versions. In *The 25th Annual Network and Distributed System Security Symposium (NDSS 2018)*, 2018.

[35] Reuters.com. Google faces lawsuit over tracking in apps even when users opted out, July 2020. URL https://www.reuters.com/article/us-alphabet-google-privacy-lawsuit-idUSKCN24F2N4.

[36] Lisa Rosenfeld, John Torous, and Ipsit Vahia. Data security and privacy in apps for dementia: An analysis of existing privacy policies. *The American Journal of Geriatric Psychiatry*, 25, 06 2017.

[37] S. Sahni. Firebase scanner., 2018. Online article (28 February 2018). https://github.com/shivsahni/FireBaseScanner.

[38] Kshitija Shirke. Mobile security framework (MobSF) static analysis, Jan 2019. Online

article. https://medium.com/@kshitishirke/mobile-security-framework-mobsf-static-analysis-df22fcdae46e.

[39] Andrea Slane, Isabel Pedersen, and Patrick C. K. Hung. Involving seniors in developing privacy best practices: Towards the development of social support technologies for seniors. in: Office of the Privacy Commissioner of Canada, 2020. Online article (2020). https://www.priv.gc.ca/en/opc-actions-and-decisions/research/funding-for-privacy-research-and-knowledge-translation/completed-contributions-program-projects/2019-2020/p_2019-20_03/.

[40] Sqlmap.org. Sqlmap automatic SQL injection and database takeover tool, January 2022. Online article. https://sqlmap.org/.

[41] TheHackerNews.com. Over 4000 Android apps expose users' data via misconfigured firebase databases, May 2020. URL https://thehackernews.com/2020/05/android-firebase-database-security.html.

[42] USAToday.com. How to stop your smartphone from tracking your every move, sharing data and sending ads, March 2019. URL https://www.usatoday.com/story/tech/columnist/komando/2019/02/14/your-smartphone-tracking-you-how-stop-sharing-data-ads/2839642002/.

[43] Kaiyu Wan, Vangalur Alagar, and Peter Oyikanmi. Elderly health care-security and privacy issue. In *International Conference of Pioneering Computer Scientists, Engineers and Educators*, pages 276–291. Springer, 2017.

[44] Haoyu Wang and Yao Guo. Understanding third-party libraries in mobile app analysis. In *2017 IEEE/ACM 39th International Conference on Software Engineering Companion (ICSE-C)*, pages 515–516. IEEE, 2017.

[45] XDA-developers.com. Android permissions & security explained, 2020. Online article. https://forum.xda-developers.com/t/android-permissions-security-explained.2312066/.

[46] Ma Zi'ang. How to use LibRadar, 2018. Online article (2018). https://github.com/pkumza/LibRadar/blob/master/docs/QuickStart.md.

[47] Sebastian Zimmeck, Peter Story, Daniel Smullen, Abhilasha Ravichander, Ziqi Wang, Joel Reidenberg, N Cameron Russell, and Norman Sadeh. Maps: Scaling privacy compliance analysis to a million apps. *Proceedings on Privacy Enhancing Technologies*, 2019(3):66–86, 2019.

Appendix A

Dataset of Analyzed Apps

Table 9: Dataset of 146 Android apps

#	App Category	App Usage	App ID
1	Tutorial	Retirement	com.plootus.android
2	Improve	Dating	app40.plusdating
3	Trainer	Caregiver	com.keema.users
4	Improve	Dating	com.agegapdating.agematch
5	Measurement	Care	com.atman.allwell
6	History	EVV	alora.aloraplus
7	Tutorial	Alzheimer	air.alzheimer
8	Tutorial	Alzheimer	com.homeinstead.alzheimersassistantandroid2
9	Tutorial	Alzheimer	com.bbi.alzheimer
10	Protection	Tracking	com.obd
11	Protection	Tracking	com.mm.android.direct.AmcrestViewPro
12	History	EVV	com.amerigroup
13	History	EVV	com.visitverify.agp
14	Protection	Tracking	com.angelsense.mobile
15	Informative	Arthritis	com.minuli.arthritis
16	Informative	Arthritis	com.csdg.uab.arpower
17	History	EVV	com.bluesummit.ascend
18	History	EVV	com.carewhen.cwmobile2
19	History	EVV	com.firstdata.fdgs.authenticare2
20	History	EVV	com.aveanna.carekeeper
21	Trainer	Caregiver	com.axiscare
22	Interface	Launcher	com.bald.uriah.baldphone.gp
23	Informative	Arthritis	com.andromo.dev421413.app468176
24	Interface	Launcher	com.myhomescreen.access
25	Interface	Launcher	com.phongphan.launcher.older
26	Interface	Launcher	name.kunes.android.launcher.bigphone
27	Protection	Tracking	com.brickhousesecurity.brickhouseLocateGPS

Continued on next page

Table 9 – continued from previous page

#	App Category	App Usage	App Name
28	Protection	Tracking	com.brickhouse.trackview
29	Trainer	Caregiver	com.carebridge.byod
30	Trainer	Caregiver	com.careconnectmobile.android
31	Trainer	Caregiver	com.comforcare.hm.caregiverapp
32	Trainer	Caregiver	com.gtindependence.Caregiver
33	Protection	Safety	com.alphahom.carego.international
34	Trainer	Caregiver	com.carelinx
35	Trainer	Caregiver	com.caremonster.appcelerator
36	Trainer	Caregiver	com.caresmartz360.pro
37	Trainer	Caregiver	caresnap.provider
38	Trainer	Caregiver	com.dgwell.caresquare.patient
39	History	EVV	com.caretime.CareTime
40	Trainer	Caregiver	com.caringvillage.app
41	Improve	Social	com.caringbridge.app
42	Trainer	Caregiver	com.clearcare.clearcareconnect
43	History	EVV	com.stpl.CICO
44	Diagnostic	Brain test	com.cognifit.app
45	Improve	Dating	com.cougar.sugarmomma.dating.hookupapps
46	Trainer	Caregiver	com.healthyroster.virtualathletictrainer.delta
47	Protection	Tracking	com.blueskyhomesales.cube
48	Protection	Tracking	com.desn.dagps
49	Measurement	Exercise	com.ebmacs.dailyseniorfitnessexcercise
50	Trainer	Caregiver	com.damava.damava_helper
51	History	EVV	com.dcisoftware.dcimobileevv
52	Protection	Tracking	yc.bluetooth.blealarm
53	Trainer	Caregiver	eka.care
54	Interface	Launcher	xyz.arjunsinh.elderlauncher
55	Protection	Pill	com.elliegrid.elliegridapp
56	Trainer	Caregiver	com.emoha
57	Tutorial	Retirement	com.app.empowerji
58	Trainer	Caregiver	net.ersp.mobileConnect
59	Trainer	Caregiver	org.ezcare.app
60	Protection	Tracking	com.family1stGPSNew
61	Protection	Tracking	com.family1stGPS
62	Improve	Dating	meet.flirtymatures.dating.com
63	Improve	Dating	com.plusde40karima.dating
64	Improve	Dating	com.seniorleserieux.seniorleserieuxcom
65	Trainer	Caregiver	com.idbsys.mobile
66	Measurement	Care	com.usbmis.reader.gayf14
67	Protection	Safety	com.mpebbles.goldenapp
68	Informative	Arthritis	com.eduven.cc.goutDiet
69	Improve	Social	net.grandpad.puma
70	Trainer	Caregiver	com.uhg.mobile.health4me
71	Diagnostic	Hearing	mobile.eaudiologia
72	Interface	Launcher	com.primuxtech.launcher
73	Protection	Safety	com.folder.HelpAgeSOS
74	Trainer	Caregiver	co.homage.careowner
75	Trainer	Caregiver	au.com.homecarenet.caregiver
76	Trainer	Caregiver	com.gadaca.dime

Continued on next page

Table 9 – continued from previous page

#	App Category	App Usage	App Name
77	Trainer	Caregiver	com.huddlehealth
78	Trainer	Caregiver	com.ianacare.ianacare
79	Trainer	Caregiver	com.kanrad.kantime
80	Interface	Launcher	com.alphabetickeyboard
81	Informative	Arthritis	com.kneepainrelief.kneearthritisexercises
82	Protection	Tracking	com.lifeAssure.lifeAssure
83	Protection	Tracking	com.brickhouse.lightningGps
84	Tutorial	Alzheimer	com.tahsin.memoryexercise
85	History	EVV	com.tellus.evv.v2
86	History	EVV	com.hopeinhomecare.evv
87	Simulation	Memory	com.nml.myhouseofmemories
88	Protection	Safety	com.mysosfamily
89	Protection	Pill	eu.smartpatient.mytherapy
90	Simulation	Memory	air.nn.mobile.app.main
91	Informative	Shopping	com.lakeba.seniorscard
92	Trainer	Caregiver	com.nursinghomecarebd.www.nursinghomecare
93	Measurement	Exercise	io.getsetup.getsetup
94	Interface	Launcher	com.oscarsenior.oscar
95	Informative	Arthritis	com.OneLife2Care.OsteoarthritisFeverHelp
96	Improve	Dating	com.topsmaour.datingmature
97	Informative	Pension	com.vikas2chandra.oldagepensionup
98	Diagnostic	Hearing	com.it4you.petralex
99	Protection	Pill	app.medcontrol.alarm.pillreminder
100	Protection	Pill	com.bestfuncoolapps.TakeYourPills
101	Protection	Pill	com.aidareminder.pillreminder
102	History	EVV	com.aquila.poc
103	History	EVV	com.pointclickcare.ceandroid
104	Simulation	Memory	com.imagiration.aphasia
105	Trainer	Caregiver	com.it.restup
106	Trainer	Caregiver	com.shoshana.caregivermobile
107	History	EVV	com.sandata.mvvhybrid.prod
108	Informative	Pension	com.swd.sccs
109	Improve	Dating	senior.dating2
110	Improve	Dating	seniordating.app
111	Improve	Dating	com.ciliara.doulikesenior
112	Informative	Shopping	com.goodbarber.seniorfree
113	Interface	Launcher	com.SeniorEasyPhone
114	Measurement	Exercise	fitness.com.senior
115	Interface	Launcher	nl.endran.seniorlauncher
116	Informative	Community	com.apfm.seniorlivingfinder.android
117	Improve	Dating	com.onlineconnections.seniornext
118	Protection	Safety	app.seniorsafety
119	Protection	Safety	com.senior_safety_phone
120	Improve	Dating	com.over.seniors
121	Informative	Shopping	com.org.seniority.application
122	Measurement	Exercise	com.andromo.dev516135.app1015140
123	Improve	Dating	com.spark.com.silversingles.app
124	Protection	Tracking	com.landairsea.silvercloud
125	Interface	Launcher	com.tct.simplelauncher

Continued on next page

Table 9 – continued from previous page

#	App Category	App Usage	App Name
126	Protection	Pill	com.iconiqStudios.pillID
127	Trainer	Caregiver	com.swyftops.caregiver
128	History	EVV	com.tellus.evv.ahca
129	Protection	Tracking	com.theoracare.theoralink
130	Trainer	Caregiver	net.therap.app
131	History	EVV	com.paragyte.publicpartnerships
132	Protection	Tracking	com.fw.gps.tkstar
133	Protection	Tracking	com.trackimo.android.tracki
134	Protection	Pill	com.tricella.pillbox
135	Measurement	Care	com.levstone.mobility.trustedelderlycare
136	Protection	Safety	com.tuya.smart
137	History	EVV	com.uhg.mobile.uhcglobal
138	History	EVV	com.visitverify.uhc
139	Trainer	Caregiver	com.velaapp
140	History	EVV	com.datalogicsoftware.vestamobile
141	Trainer	Caregiver	com.vitaltech.vitalcare
142	Tutorial	Alzheimer	com.alzheimer.activities
143	Measurement	Exercise	eu.fitric.seniorexercise
144	Protection	Tracking	com.navixy.xgps.client.app
145	Measurement	Exercise	com.andromo.dev598202.app634359
146	Tutorial	Alzheimer	es.lapisoft.yotecuido

Table 10: List of 122 unique third party libraries in 146 apps

#	Library Name	Library Type
1	Smaato	Advertisement
2	Unity3d Ads	Advertisement
3	Inmobi	Advertisement
4	Google Ads	Advertisement
5	Applovin	Advertisement
6	StartApp	Advertisement
7	Google Play	App Market
8	Android Support v4	Development Aid
9	Fabric	Development Aid
10	Volley HTTP library	Development Aid
11	Google Mobile Services	Development Aid
12	ZXing ('Zebra Crossing')	Development Aid
13	Firebase	Development Aid
14	Google Gson	Development Aid
15	PhoneGap	Development Aid

Continued on next page

Table 10 – continued from previous page

#	Library Name	Library Type
16	OKHttp3.0	Development Aid
17	Bolts Base Library	Development Aid
18	Apache Cordova	Development Aid
19	SLF4J	Development Aid
20	Logback	Development Aid
21	Adobe Air FRE	Development Aid
22	Mono for Android	Development Aid
23	Glide	Development Aid
24	Github	Development Aid
25	Zip4j	Development Aid
26	Android Support v7	Development Aid
27	ActionBarSherlock	Development Aid
28	picasso	Development Aid
29	Multidex	Development Aid
30	newrelic	Development Aid
31	hamcrest	Development Aid
32	jUnit Java Unit Test	Development Aid
33	Apache Http	Development Aid
34	JDOM2	Development Aid
35	MaterialProgressBar	Development Aid
36	OkHttp	Development Aid
37	Nine Old Androids	Development Aid
38	Google Core Libraries for Java 6+	Development Aid
39	Google Core Libraries (3rd Party)	Development Aid
40	Esoteric Software 2D	Development Aid
41	Roboguice	Development Aid
42	Google Protocol Buffers	Development Aid
43	BugSense	Development Aid
44	GNU KAWA	Development Aid
45	GNU Mapping	Development Aid
46	GNU Common Lisp	Development Aid
47	GNU XML	Development Aid
48	ECMAScript	Development Aid
49	Google Appinventor	Development Aid
50	OkHttp okio Framework	Development Aid
51	Apache Common	Development Aid
52	Kawa for Android	Development Aid
53	Nostra13 Image Loading	Development Aid
54	Android Support Design	Development Aid
55	YouTube Android Player API	Development Aid

Continued on next page

Table 10 – continued from previous page

#	Library Name	Library Type
56	Adobe FlashPlayer	Development Aid
57	Adobe Air	Development Aid
58	Titanium-Modules	Development Aid
59	Jaxen	Development Aid
60	Appcelerator	Development Aid
61	AndroidAsync	Development Aid
62	HttpClient Android repackage buildscript	Development Aid
63	XML Pull	Development Aid
64	Smack Extensions	Development Aid
65	getui	Development Aid
66	Disk LRU Cache	Development Aid
67	FlexJson Library	Development Aid
68	Dagger	Development Aid
69	Joda Time	Development Aid
70	simple framework	Development Aid
71	Amazon AWS	Development Aid
72	Scribe Java Lib	Development Aid
73	KObjects	Development Aid
74	Kxml2	Development Aid
75	Jsoup	Development Aid
76	Android Support v13	Development Aid
77	Bouncy Castle	Development Aid
78	Spongy Castle	Development Aid
79	retrofit RESTful Library	Development Aid
80	DATE4J	Development Aid
81	otto	Development Aid
82	Google GData	Development Aid
83	ksoap2	Development Aid
84	json smart	Development Aid
85	Apache Harmony	Development Aid
86	Dnsjava	Development Aid
87	SourceForge ZBar	Development Aid
88	Junit	Development Aid
89	Google GCM	Development Aid
90	ISO Parser	Development Aid
91	EventBus	Development Aid
92	AspectJ	Development Aid
93	ACRA (App Crash Reports for Android)	Development Aid
94	RxJava Retrofit	Development Aid
95	J256	Development Aid

Continued on next page

Table 10 – continued from previous page

#	Library Name	Library Type
96	Google Internationalization	Development Aid
97	Fastjson	Development Aid
98	Baidu APP SDK	Development Aid
99	Google API Client Libraries	Development Framework
100	goodbarber	Development Framework
101	Fmod	Game Engine
102	butterknife UI Framework	GUI Component
103	Kankan Wheel Android scroller	GUI Component
104	android widget	GUI Component
105	Keyboard Surfer	GUI Component
106	Google Maps Utils	Map/LBS
107	Baidu Location	Map/LBS
108	Baidu Map	Map/LBS
109	urbanairship	Mobile Analytics
110	Crashlytics	Mobile Analytics
111	AppsFlyer	Mobile Analytics
112	Flurry	Mobile Analytics
113	Google Analytics	Mobile Analytics
114	HockeyApp	Mobile Analytics
115	Smack	Payment
116	Amazon In-App Purchasing	Payment
117	PayPal	Payment
118	Facebook	Social Network
119	Tencent Wechat	Social Network
120	Fasterxml	Utility
121	JavaX Annotation API	Utility
122	TagSoup	Utility

Table 11: Number of unique third party libraries in 146 Android apps

App Name	App Category	App Usage	Unique Libraries
com.kneepainrelief.kneearthritisexercises	Informative	Arthritis	24
com.theoracare.theoralink	Protection	Tracking	22
com.alzheimer.activities	Tutorial	Alzheimer's	19
com.mysosfamily	Protection	Safety	18
com.ciliara.doulikesenior	Improve	Dating	17
com.vikas2chandra.oldagepensionup	Informative	Pension	16
com.app.empowerji	Tutorial	Retirement	15
com.homeinstead.alzheimersassistant-android2	Tutorial	Alzheimer's	15
com.huddlehealth	Trainer	Caregiver	15
com.caringvillage.app	Trainer	Caregiver	15
com.trackimo.android.tracki	Protection	Tracking	15
app.seniorsafety	Protection	Safety	14
com.aveanna.carekeeper	History	EVV	14
com.datalogicsoftware.vestamobile	History	EVV	14
com.it.restup	Trainer	Caregiver	14
com.emoha	Trainer	Care	14
com.cougar.sugarmomma.dating.hookup-apps	Improve	Dating	13
com.tellus.evv.ahca	History	EVV	13
com.stpl.CICO	History	EVV	13
au.com.homecarenet.caregiver	Trainer	Caregiver	13
com.brickhousesecurity.brickhouse-LocateGPS	Protection	Tracking	13
com.brickhouse.lightningGps	Protection	Tracking	13
meet.flirtymatures.dating.com	Improve	Dating	12
com.visitverify.uhc	History	EVV	12
com.carelinx	Trainer	Caregiver	12
com.vitaltech.vitalcare	Trainer	Caregiver	12
com.over.seniors	Improve	Dating	12
com.bbi.alzheimer	Tutorial	Alzheimer's	11
com.andromo.dev598202.app634359	Measurement	Exercise	11
com.goodbarber.seniorfree	Informative	Shopping	11
com.cognifit.app	Diagnostic	Brain test	11
com.visitverify.agp	History	EVV	11
com.uhg.mobile.health4me	Trainer	Caregiver	11
com.mm.android.direct.AmcrestViewPro	Protection	Tracking	11
com.seniorleserieux.seniorleserieuxcom	Improve	Dating	10

Continued on next page

Table 11 – continued from previous page

App Name	App Category	App Usage	Unique Libraries
com.phongphan.launcher.older	Interface	Launcher	10
com.onlineconnections.seniornext	Improve	Dating	10
com.firstdata.fdgs.authenticare2	History	EVV	10
com.amerigroup	History	EVV	10
com.uhg.mobile.uhcglobal	History	EVV	10
com.carebridge.byod	Trainer	Caregiver	10
com.healthyroster.virtualathletic-trainer.delta	Trainer	Caregiver	10
com.clearcare.clearcareconnect	Trainer	Caregiver	10
com.idbsys.mobile	Trainer	Caregiver	10
yc.bluetooth.blealarm	Protection	Tracking	10
com.landairsea.silvercloud	Protection	Tracking	10
com.fw.gps.tkstar	Protection	Tracking	10
com.obd	Protection	Tracking	10
com.nml.myhouseofmemories	Simulation	Memory	9
com.usbmis.reader.gayf14	Measurement	Care	9
com.tellus.evv.v2	History	EVV	9
com.caremonster.appcelerator	Trainer	Caregiver	9
eu.fitric.seniorexercise	Measurement	Exercise	9
com.csdg.uab.arpower	Informative	Arthritis	8
com.ebmacs.dailyseniorfitnessexcercise	Measurement	Exercise	8
com.hopeinhomecare.evv	History	EVV	8
com.sandata.mvvhybrid.prod	History	EVV	8
com.comforcare.hm.caregiverapp	Trainer	Caregiver	8
com.shoshana.caregivermobile	Trainer	Caregiver	8
com.gadaca.dime	Trainer	Caregiver	8
com.family1stGPSNew	Protection	Tracking	8
air.nn.mobile.app.main	Simulation	Memory	7
app40.plusdating	Improve	Dating	7
fitness.com.senior	Measurement	Exercise	7
com.spark.com.silversingles.app	Improve	Dating	7
com.topsmaour.datingmature	Improve	Dating	7
com.dcisoftware.dcimobileevv	History	EVV	7
com.aquila.poc	History	EVV	7
com.swyftops.caregiver	Trainer	Caregiver	7
net.ersp.mobileConnect	Trainer	Caregiver	7
com.caringbridge.app	Improve	Social	7
com.andromo.dev421413.app468176	Informative	Arthritis	6
com.plusde40karima.dating	Improve	Dating	6

Continued on next page

Table 11 – continued from previous page

App Name	App Category	App Usage	Unique Libraries
com.tct.simplelauncher	Interface	Launcher	6
net.therap.app	Trainer	Caregiver	6
com.velaapp	Trainer	Caregiver	6
com.tricella.pillbox	Protection	Pill	6
com.lifeAssure.lifeAssure	Protection	Tracking	6
com.family1stGPS	Protection	Tracking	6
com.minuli.arthritis	Informative	Arthritis	5
com.OneLife2Care.OsteoarthritisFever-Help	Informative	Arthritis	5
com.eduven.cc.goutDiet	Informative	Arthritis	5
eu.smartpatient.mytherapy	Protection	Pill	5
nl.endran.seniorlauncher	Interface	Launcher	5
com.axiscare	Trainer	Caregiver	5
com.gtindependence.Caregiver	Trainer	Caregiver	5
com.kanrad.kantime	Trainer	Caregiver	5
com.nursinghomecarebd.www.nursing-homecare	Trainer	Care	5
com.apfm.seniorlivingfinder.android	informative	Care	5
com.bestfuncoolapps.TakeYourPills	Protection	Pill	5
com.angelsense.mobile	Protection	Tracking	5
com.navixy.xgps.client.app	Protection	Tracking	5
com.tahsin.memoryexercise	Tutorial	Alzheimer's	4
co.homage.careowner	Trainer	Caregiver	4
com.primuxtech.launcher	Interface	Launcher	4
com.paragyte.publicpartnerships	History	EVV	4
org.ezcare.app	Trainer	Caregiver	4
net.grandpad.puma	Improve	Social	4
com.andromo.dev516135.app1015140	Measurement	Exercise	4
io.getsetup.getsetup	Measurement	Exercise	4
es.lapisoft.yotecuido	Tutorial	Alzheimer's	4
com.blueskyhomesales.cube	Protection	Tracking	4
seniordating.app	Improve	Dating	3
com.senior_safety_phone	Protection	Safety	3
com.swd.sccs	Informative	Pension	3
com.agegapdating.agematch	Improve	Dating	3
com.imagiration.aphasia	Simulation	Memory	3
com.SeniorEasyPhone	Interface	Launcher	3
app.medcontrol.alarm.pillreminder	Protection	Pill	3
com.bluesummit.ascend	History	EVV	3

Continued on next page

Table 11 – continued from previous page

App Name	App Category	App Usage	Unique Libraries
com.pointclickcare.ceandroid	History	EVV	3
alora.aloraplus	History	EVV	3
com.careconnectmobile.android	Trainer	Caregiver	3
com.bald.uriah.baldphone.gp	Interface	Launcher	3
com.iconiqStudios.pillID	Protection	Pill	3
com.aidareminder.pillreminder	Protection	Pill	3
mobile.eaudiologia	Diagnostic	Hearing	2
com.oscarsenior.oscar	Interface	Launcher	2
air.alzheimer	Tutorial	Alzheimer's	2
com.folder.HelpAgeSOS	Protection	Safety	2
com.caretime.CareTime	History	EVV	2
name.kunes.android.launcher.bigphone	Interface	Launcher	2
com.brickhouse.trackview	Protection	Tracking	2
com.it4you.petralex	Diagnostic	Hearing	1
com.levstone.mobility.trustedelderlycare	Measurement	Care	1
com.carewhen.cwmobile2	History	EVV	1
eka.care	Trainer	Caregiver	1