Final Report for OPC Contributions Program 2021-2022

“Privacy Report Card for Online Solutions Targeting Seniors”

Mohammad Mannan, and Amr Youssef

Student team members:
Pranay Kapoor and Rohan Pagey

Concordia Institute for Information Systems Engineering (CIISE)
Concordia University, Montreal

Abstract. 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. Android apps help the elderly to improve their daily lives and connectivity, their caregivers and family members to monitor the 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 108 popular Android apps from 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 90/108 apps fail to adequately preserve the security and privacy of their users in one or more ways; specifically, 10 apps allow full account takeover, and 8 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.

Keywords: Elderly Privacy · Android Security · Android Apps Privacy · Mobile Security

1 Introduction

The adoption of mobile devices is forcing the elderly to navigate the treacherous waters of a complex digital world [4], 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 [17, 5], several studies have shown that older adults are more vulnerable to security and privacy threats than the general population [18]. 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 [3]. A combination of low self-efficacy, mistrust and lack of awareness and understanding of security hazards [20] 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 improving physical and mental health (e.g., apps for exercise and fitness). While these solutions 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 [9] 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. 7). A few

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

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 report, we perform an in-depth analysis of 108 prominent elderly apps. 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 [12], 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 [16]. We analyze data collection privacy issues, security vulnerabilities, presence of third-party trackers, and insecure data transmission. Furthermore, we also analyze three IoT devices to better understand the corresponding apps and their security implications. One of the limitations of our dynamic analysis is that we are unable to create accounts in many apps, since real medical records or company access codes are needed to login into these apps. In such events, we dynamically test the app till the login page to check the app's authentication management and the behavior of the app when it loads.

Contributions and notable findings.

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 108 Android apps (and the IoT devices corresponding to three apps). Overall, 90/108 apps fail to adequately protect the security and privacy of users due to one or more vulnerabilities.
3. 5/108 Android apps *(Empowerji, 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. 10/108 Android apps *(40 Plus Senior Dating, FlirtMatures Dating, Empowerji, Senior Safety App, GoldenApp, EZ Care, POC EVV, All Well Senior Care, Seniority, Cougar Dating)* allow an attacker to easily compromise the elderly and caregivers account.
5. 8/108 *(Senior dating, Empowerji, GoldenApp, Caring Village, EZ Care, Generations Homecare System, EllieGrid, Seniority)* 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.
6. 12/108 Android apps transmit PII via HTTP to their client-side solutions (e.g., *Empowerji, GoldenApp*), while 8/108 apps transmit PII (6/108 via HTTP and 2/108 via HTTPS) to various third-party domains.
7. 14/108 apps do not provide any privacy policies. Data retention clauses are mentioned by 48/108 apps. 28/108 policies do not mention regulatory compliance for any audience. Generic security measures are declared in 57/108 policies, with 40/108 app policies having no mention of security practices and only 9/108 apps have a Privacy Audit in place as per policy.

2 Potential Privacy and Security Issues and Threat Model

In this section, we list the security and privacy issues, and the threat model that we consider.

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 which 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, timezone, 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., Google 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 Storage, Fine Location) automatically acquired by a third-party library or malicious app on the client device, when requested by the elderly app.
10. No mention of privacy regulations: we check the privacy policies (if available) for any mention of adhering to data privacy regulations (e.g., EU GDPR, California CCPA).

Threat Model. We consider three attacker types with varying capabilities: (1) On-device attacker: a malicious app with limited permissions on the elderly 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 a apps's backend server. Our threat model does not consider attacks requiring physical device access.

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 don't 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.

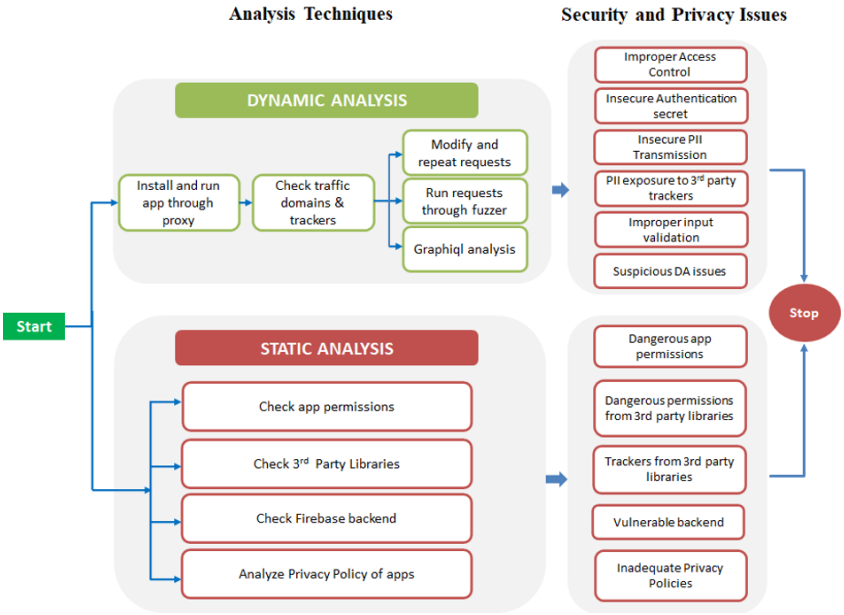

Fig. 1. Overview of our methodology

We also refrain from running any automated scanners that might bombard the servers to cause denial of service attacks. As part of the responsible disclosure, we are in the process of contacting the developer and security team of our tested apps, while sharing our detailed proof-of-concept and explaining them the related security consequences.

3 Analysis Methodology

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

3.1 App Selection

We search Google Play store for elderly apps (and also screen the best apps for older adults [13]), with relevant keywords.[2] The search was conducted on May 20, 2021, which provided us with 500 apps for further consideration. We then 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 (1) apps needing financial accounts or verified identities (such as bank and credit card accounts, social security numbers) and (2) not intended for the elderly. We manually screen each app to check if it satisfies our key requirements. Our final dataset contains 108 apps. We also found that 3/108 apps have a companion IoT device, and thus we bought these 3 IoT products to better understand their functionality.

[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”.

3.2 Dynamic Analysis of Traffic Flow

We perform dynamic testing of the apps to simulate the real world usage for 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 1 to 24 hours depending on the feature-set of the app, collect traffic from the elderly apps and the IoT devices, and then perform our analysis (explained further in this section). Fig. 1 shows an overview of 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 our apps use graphql [15]; 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 [10] to test the network traffic on the apps using graphql. Dynamic analysis with Burp Suite and Graphiql[4] tools gives us an in-depth understating of how each app works so that we can find relevant security and privacy issues.

The four main components for our dynamic analysis include the following: (1) *Proxy* is 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) *Intruder* is 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 and to any third-party domains and trackers. We identify the known third-party trackers using EasyList and EasyPrivacy [6] 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, 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

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

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 [21] 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.

IoT Device Analysis. For each of the selected IoT devices, we first categorize them into three components: (1) radio communications, (2) embedded device and hardware, and (3) companion apps. We test the companion apps by following the same dynamic and static analysis process as described in Sec. 3.2 and Sec. 3.3, respectively. The testing methodology for the other two components is described below.

Radio communications. First, we determine the type of communication between the IoT device and its companion app. We do this by going through the working manual of the device and by reading the product's description. During the Android app analysis, we observed that 3 apps offer an IoT device; 2/3 use Bluetooth as a communication protocol between the IoT device and the Android app, whereas one device uses WiFi for connectivity. Therefore we focus our approach on only analyzing these two types of protocols. To analyze the Bluetooth communication between the IoT device and the app, we retrieve the Bluetooth HCI snoop log file from the phone's internal storage (which can be extracted using adb shell) and check all the packets that have been sent between the IoT device and the smartphone. For the analysis of WiFi communication, we setup the Wireshark proxy and observe the traffic flow to/from the device.

Embedded device and hardware analysis. In this step, we pop open the IoT device to look at the different components and analyze their functionality. The device's internals usually involve many components including the printed circuit board (PCB), connectors, antenna, peripherals, and so on. Since opening a device and tampering with the internals is not safe and might lead to rendering it non-functional, we perform this step at the end of our analysis. Primarily, 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, Firebase, and Privacy Policy

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: any included third-party libraries with the app,

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 dangerous permissions and trackers.

Firebase Analysis. We analyze the Firebase configuration for security issues by performing an automated analysis using Firebase Scanner [25]. Critical misconfigurations can allow

[5] https://github.com/pkumza/LibRadar/blob/master/docs/QuickStart.md

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.

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 [26]. So, we use MobSF for static analysis of 108 apps to check for vulnerabilities related to sensitive information logged or hard-coded in files, improper usage of SQLite databases, insecure implementation of SSL, IP address disclosure, 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 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 [11]. 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.

4 Results

Following the methodology in Sec. 3, we tested 108 Android apps for elderly people, between October 2020 and December 2021. We report our findings in this section, with an overview of the top 30/108 apps with the most security and privacy issues. Table 1 highlights 30 apps with the most concerning issues.

4.1 Improper Authentication Management

We found that 10/108 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. For *All Well Senior Care* and *Seniority*, 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 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. Effects of these vulnerabilities are further discussed in the case studies in Section 5. During our retesting, we also noticed that *Senior Safety app* fixed its issues in a software update.

4.2 Insecure Session Management

We found 10/108 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

[6] http://opensecurity.in/mobilesecurity-framework/
[7] https://pribot.org/polisis/

Table 1. Overall results for 30/108 elderly apps with the most 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
Oscar Senior/Enterprise (v6.8.2)			◐	◐				◐	
Senior Dating by Lauber (v9.1)						◐			
40 Plus Senior Dating (v9.8)	◐	◐	◐	◐	◐			◐	
Over 40 Dating Mature (v1.0)					◐				◐
FlirtMatures Dating (v1.0)	◐	◐			◐				
Empowerji (v5.7)	◐	◐	◐		◐	●		◐	
Senior Safety App (v9.7)	◐	◐	◐	◐				◐	
GoldenApp (v3.2)	◐	◐	◐		◐	◐	●		
Amerigroup EVV Tennessee (v1.36.1)					◐				
EZ Care (v0.0.7)	◐	◐	◐			●			
Caring Village (v0.16.5)			◐			◐		◐	
Generations Homecare System (v3.3.3)						◐		◐	
EllieGrid (v3.4.1)						●			
POC EVV (v3.2)	◐	◐	◐		◐		◐		
Alzheimer's Daily Companion (v1.0.7)		◐	◐						
Big Launcher (v1.4)		◐		◐					
HelpAgeSOS (v1.0.27)		◐	◐						
Senior discounts (v2.2)					◐		○	◐	
Carelinx (v3.0.1)				◐				◐	
All Well Senior Care (v2.15.0)	●								
Seniority E-commerce app (v.1.0.2)	●			◐		◐			
Cougar Dating (v1.1.3)	◐		◐					◐	
Damava (v1.2.4)			●				●		
CrescendoConnect (v4.24)					◐				
401(K) - Retirement Planning (v2.5)					◐				
Doulikesenior (v1.5.1)					◐			◐	
Senior Dating: Date Mature Singles (v1.7)				◐					
Homage (v5.0.8)				◐				◐	
Mobile Caregiver+ (v2.0.35)					◐				
Rosemark Caregiver Mobile (v1.4.93)					◐				

a request from this app and perform an account takeover. Also, 4/108 apps did not use any authentication secret. For example, *GoldenApp* does not make use of any authentication secrets 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 Traffic Analysis

We found that 12/108 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), *EZ Care* (username and password during login and the private messages sent and received between a doctor and the user).

We found that 92/108 apps communicate with 310 third-party (non-tracker) domains[8]. We found 36/108 apps communicate with Googleapis.com domains, 35/108 apps with Firebase, and 21/108 apps with Facebook domains. 55/108 apps had traffic flowing through at least one Google domain. We found a total of 35 unique tracker domains with 112 occurrences across 57/108 Android apps. Table 2 shows the top 10 trackers found. The top 3 prevalent trackers are DoubleClick (21/108), Crashlytics (21/108), and Google Syndication (11/108). Crashlytics is a "crash reporting" software designed to help identify bugs in the apps and also report the user's activity to the app developers so they can take appropriate measures to ensure that users don't stop using their product. DoubleClick is a Google ad service. *Senior Discounts, Big Keyboard & Notifications, Free chat & senior dating, Senior Dating by Lauber, Ianacare, Oscar Senior* are some popular apps in which we detected over 10 domains and trackers. These apps could expose their vulnerable users to potential data leaks, or voluminous in-app advertisements.

Moreover, out of the 12 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* (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).

We also observe that 13/108 apps send device information and usage data in plaintext (6/108 over HTTP and 7/108 over HTTPS) to third-party domains. The most common parameters include, phone model, OS build version etc. Apps such as *Homage*, *EZ Care*,

[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 2. Top 10 Trackers that receive traffic from elderly apps

Tracker	# Apps
crashlytics.com	21
doubleclick.net	21
googlesyndication.com	11
google-analytics.com	8
googletagmanager.com	5
appsflyer.com	5
flurry.com	4
onesignal.com	3
googleadservices.com	3
branch.io	3

All Well Senior Care send WiFi, cellular information, signal strength, and a flag to check if the device is rooted or not.

4.4 Improper Access Control

We found 4/108 apps with improper access control. *GoldenApp*'s access control issues are due to insecure session management. As there is no authentication token 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 identity theft for the user. *POC EVV* contains a 5 digit parameter as the user ID in the requests which can be changed to get other users' data (e.g., phone number, address, zip code) on the platform. *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 on the platform. *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) on the platform. The detailed impact of these apps is discussed in the case studies (Sec. 5).

4.5 Improper Input Validation

We found that 8/108 apps are vulnerable to various injection attacks such as SQL/code injection, cross-site scripting. Examples include: *Senior dating by Lauber, GoldenAPP, Caring Village* and *Generations homecare systems* 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* and *EZ Care* are vulnerable to SQL injection attacks. With SQL injection one can view, modify and delete any elderly user's data, the impact of which is described in Sec. 4.11. *Elliegrid* is vulnerable to code injection. For this particular 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. As there was no authentication secret on the request, 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 16/108 apps having various types of security misconfigurations. Apps such as *DoULikeSenior, CareLinx, Pension Status Search OldAge Widow Handicap, Homage* are vulnerable to Cross-Site Request Forgery (CSRF) attacks as they transmitted the requests to modify an object via unprotected GET requests. CSRF attacks are 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 into a file or URL path 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 execute this attack.

4.7 Dangerous App Permissions

Dangerous permissions grants the application access to personal user data (e. g., user's location) or control over the user's device. They are only granted after explicit consent from the user. Various dangerous permissions as used in our tested apps are summarized in Table 3. We found a total of 412 dangerous permissions in 86/108 apps, i.e., an average of almost 5 dangerous permissions per app (maximum of 11 in *Ianacare* and minimum of 1 in *Hearing Test, FlirtMatures Dating, Simple Launcher, Pill Reminder & Medicine App, PointClickCare Companion*). The top 2 dangerous permissions found were Read External Storage (69 apps) and Write External Storage (66 apps). Rarely used permissions found were Write Contacts (*Senior Safety Phone and Ianacare*), Read Phone Contacts (*HelpAge SOS*) and Read Phone Numbers (*Trusted Senior Care*).

Table 3. Dangerous permissions asked by elderly apps

Dangerous Permission	# Apps
Read Storage	69
Write Storage	66
Fine Location	62
Coarse Location	54
Camera	42
Record Audio	28
Read Phone State	27
Call Phone	17
Get Accounts	15
Read Contacts	15
Read Calendar	7
Write Calendar	7
Write Contacts	2
Read Phone Numbers	1

42/108 apps asked for Camera permission, such as *Petralex, Walk to End Alzheimer's, My house of memories, Gout Diet recipes, Oscar Senior, Senior Discounts, Seniority, Aveanna EVV, 401(K) Retirement planning.* 62/108 apps required Access Fine Location permission. Apps with a significantly high number of risky permissions include *Ianacare, 401(K)*

- Retirement Planning, Aveanna EVV, Oscar Senior, Senior Safety App, Cougar Dating, CrescendoConnect, Generations Homecare, Trusted Senior Care, ClearCareGo.

4.8 Third-Party Libraries and Permissions

Types of Libraries. We found 109 unique third-party libraries in 108 Android apps and a total of 2524 libraries. These libraries serve various purposes: app development (84/109), analytics (6), advertisements (6), and social networking (2), and more.

We found 26/108 apps with social media libraries and 13/108 apps with advertisement libraries. The predominant social network library we found was *Facebook*, requiring a total of 79 permissions (Bluetooth Admin, Wake Lock, Backup, Internet, Dump) in 26 apps. Advertisement libraries found mainly were *Google Ads* with 8 permissions (Dump) in 8 apps and *Unity3d Ads* with 16 permissions (Bluetooth Admin, Wake Lock, Backup, Internet, Dump) in 3 apps.

Libraries by App Category. A high number of total third-party libraries were found in Caregiver apps (700/2524), EVV apps (554/2524), Dating apps (325/2524) and apps for Arthritis (182/2524). 69% of all 1264 permissions asked by total 2524 libraries are from these 4 categories alone. This shows that vulnerable user groups (comprising the elderly needing help, chronically unwell, single elderly) may be more prone to privacy and data security issues due to these third-party libraries. Examples of apps with a high number (>20) of library permissions (from multiple libraries) are *DoULikeSenior (38), Walk to End Alzheimer's (29), Caring Village (23), Knee Arthritis Exercises (24), Empowerji (27), Aveanna EVV (24). DoULikeSenior* had the maximum number of 38 library permissions (Bluetooth Admin, Dump, Internet, Wake Lock, Backup, Write Secure Settings). The least number of library permissions, 3 (Dump), were in *RecoverBrain Therapy.*

Table 4. Permissions asked by 109 unique third-party libraries

Permission	# Unique Libraries	# Total Permissions
Dump	60	484
Internet	32	292
Backup	19	126
Bluetooth Admin	9	106
Wake Lock	9	106
Write Secure Settings	5	131
Bluetooth	4	15
Write APN Settings	1	4

Libraries with High Number of Permissions. 2524 libraries asked for a total of 1264 permissions and 60/109 unique libraries asked for one or more permissions. The libraries with the highest number of permissions asked are *Android support* (75/108 apps, 349 total & 6 unique permissions), *Google Mobile services* (77/108 apps, 381 total & 7 unique permissions), *Firebase* (47/108 apps, 98 total & 7 unique permissions), *Facebook* (26/108 apps, 79 total & 5 unique permissions), and *Glide* (27/108 apps, 27 total & 1 unique permission).

Kinds of Permissions Asked. We found 8 unique permissions asked by the libraries; see Table 4. We found Dump permission for example used by *Firebase* (47/108 apps), *Glide*

(27/108 apps) and *Facebook* (24/108 apps) third-party libraries. Backup permission was used by *Facebook* (11/108 apps) and *Unity3d Ads* (4/108 apps) libraries.

Write Secure Settings permission was asked by 5 libraries, predominantly by *Google Mobile Services* (42/108 apps) and *Firebase* (45/108 apps). Bluetooth Admin permission was used by 9 libraries, mainly by *Facebook* (10/108 apps), *Unity3d Ads* (2/108 apps), and *Smaato* (2/108 apps).

4.9 Static Code Analysis

Our analysis of Static Code with MobSF shows that 75/108 apps can read/write to External Storage, 68/108 apps execute raw SQL queries which may expose them to SQL Injection attacks, 59/108 apps use weak hash known to have hash collisions, 47/108 apps disclose IP address, 30/108 apps have insecure WebView implementation, 13/108 apps have insecure SSL implementation, a critical security issue.

Apps which had the above 6 concerns are *Alzheimer's Disease Pocketcard, Big Keyboard & Notifications, Over 40 - Find People 50, Pill Reminder & Medicine App, DoULikeSenior, Senior Safety App, Silver 50 Dating.*

4.10 Inadequate Privacy Policies

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 5.

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 data. The common reasons cited for this data collection include: "Service improvement", "Analytics research" and "Advertising & Marketing".

Third-Party Data Sharing Practices. 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. 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.

Table 5. Data 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

App Name	Data Collected	Data shared 3P	User Choices	Data Security	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 sharing	x	x	C	Yes
Amerigroup EVV	5	2	x	Low	1 listed	E	Yes
Big Launcher - Old Age	8	6	x	Low	x	x	Partial 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, don't use	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	Partial deletion
EZ Care	6	4	1P cookies, don't use	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, don't use	Medium	2 listed	I	Full deletion
Empowerji	5	1	x	x	Unsp.	x	Yes
Mobile Caregiver	12	5	x	Medium	x	C, I	Partial 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, don't use	Medium	2 listed	x	x
Seniority	3	1	1P cookies, don't use	Low	x	x	Partial deletion

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 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 account (25/108), and deactivate account (7/108).

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

4.11 IoT Devices Analysis

Out of the 108 Android apps, we found that 3 of them offer an IoT device. We ordered these 3 IoT devices: *Elliegrid*, *Carego Alphahom*, *Tuya SoS* and we performed testing on them to better understand their companion apps. *Elliegrid* is a smart pill organizer which makes medication management easier and is specifically designed to help the elderly by reminding them to take their pills timely. This solution consists of a pillbox and an Android app. The physical pillbox is designed to store the pills and receive reminders in the form of ring notifications. The reminders and medications can be set up in its companion Android app. We found two major vulnerabilities in *Elliegrid*. Firstly, they offer functionality to alert the elderly user's caregiver via e-mail 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 UI. An adversary can select a random caregiver's ID to modify the caregiver's email and send the alerts to an attacker under his control, which would allow him to track the elderly 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 which offers additional functionalities for the elderly, such as viewing their weekly adherence report and adding a caregiver. Specifically, we found a parameter *subscriptionTypeId* from the user profile API, which is setting the value of the current plan. We note that an adversary can set the above parameter's value to *premium* and upgrade their *Elliegrid* account for free. During the hardware analysis phase for this device, we noticed that they are not using any common serial communication ports such as the JTAG or UART, as a result we could not debug and extract the firmware from the device. Also, none of the devices' vendors offer the firmware on their websites.

We also found some vulnerabilities by following the static analysis approach in the *AplhaOM CareGo*, which provides a personal alarm system for the elderly. In particular, the application is vulnerable to Janus vulnerability [19] in which an adversary can prepend a

malicious DEX file to an APK file while keeping its signature unaffected. The Android versions from 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.

4.12 Firebase Analysis

68/108 Android apps use Google Firebase as a backend service and we found 4/68 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, so we could not look at what data is being stored on their servers. For the other 2 apps (*UnitedHealthcare EVV Tennessee* and *Amerigroup EVV Tennessee*), at the time of testing, we could not see any sensitive data being stored on their database.

5 Case Studies and Practical Attacks

In this section, we discuss a few selected apps and highlight the practical attacks that can potentially be executed by exploiting the vulnerabilities we found.

5.1 POC EVV

This app[9] (10,000+ installs on Google Play store) is developed for caregivers to help them view their schedules and report 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 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.

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

5.2 Empowerji

This technology learning app[10] (10,000+ installs on Google Play store) helps the elderly learn how to use everyday apps. Firstly, we noticed 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 [28] to confirm that we can take over the database server due to the SQL injection vulnerabilities. We did not perform 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.

5.3 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, 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 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).

5.4 Elderly dating apps

40 Plus Senior Dating[12] and *Cougar Mature Women Dating App*[13] (both with 50,000+ installs on 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 can not 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*

[10] https://play.google.com/store/apps/details?id=com.app.empowerj
[11] https://play.google.com/store/apps/details?id=com.org.seniority.application
[12] https://play.google.com/store/apps/details?id=app40.plusdating
[13] https://play.google.com/store/apps/details?id=com.cougardating.cougard

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 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.

5.5 All Well Senior Care

All Well Senior Care[14] (50,000+ installs on 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).

5.6 Big Launcher - Launcher For Old Age People

This app[15] (50,000+ installs on 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.

5.7 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.[16] 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

[14] https://play.google.com/store/apps/details?id=com.atman.allwell
[15] https://play.google.com/store/apps/details?id=com.phongphan.launcher.older
[16] https://play.google.com/store/apps/details?id=com.damava.damava_helper, 5,000+ active users.

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 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.

6 Limitations

As Goole Play Store does not have a defined "Elderly" or "Senior" app category (e.g., "Health"), 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 43/84 of our apps. This was limited because we cannot directly create an account ourselves as the companies either make accounts for the users beforehand (and give them their access information) or the apps will validate the user's information (like medical insurance numbers, organization email IDs, etc., which we cannot provide in our test accounts) before creating the account. This was most applicable for the EVV and caregiver apps. Nevertheless, for those specific apps, we conduct a limited dynamic analysis of pre-login application behaviors. We did not test any paid apps.

We initially used Lumen Privacy Monitor to analyze the set of test apps. Lumen is an advanced traffic analysis tool for Android that leverages Android's VPN permission to capture and analyze real-world network traffic – including encrypted flows – locally on the mobile device itself and in user-space [7]. But we found that Lumen did not uncover many security flaws as compared to Burp Suite. Even though Lumen would show leaks, there will be a layer of uncertainty as we will not know how a leak was transmitted (over HTTP or HTTPS) and where it was leaked (to the client-side product itself or some third-party domain). Lumen would also show traffic statistics but this would constantly change every time we used the app, giving us inconsistent results. Also, Lumen did not work reliably on newer Android versions and we noticed it worked best below Android 7. Hence, we decided to use a more manual approach with Burp Suite.

7 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. [27] collected seniors' perspectives on technological devices and applications. Seniors shared their experiences and views on using specific technologies for social support functions. This work showed how seniors protect their personal information, and what knowledge, tools, and support they would need in order to consider expanding their current practices to new functions or devices.

Huckvale et al. [14] 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. [8] identified a range of complex privacy and security attitudes and needs specific to *older adults*, along with common threat models, misconceptions, and mitigation strategies. This work showed how older adults' limited technical knowledge, experience, and age-related declines in ability 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. Oliveira et al. [22] showed that *older women* were the most vulnerable group to phishing attacks in a study of 158 Internet users.

Razaghpanah et al. [23] 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. [24] analyzed 512 apps (of various categories) for privacy leaks over time across three dimensions (PII leaks, HTTPS adoption, and domains contacted) independently, and found that app privacy is getting 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 (DOB), phone number, contact info, gender.

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 privacy policies, using various tools for both dynamic and static analysis.

8 Conclusion

In this report, we have shown how apps that are intended to help or assist elderly people in one way or another, can become a "suspect" rather than a "solution". Our comprehensive analysis of 108 Android apps included dynamic analysis of traffic domain flows, trackers, leaks, and permissions, static analysis of third-party libraries for risky permissions, and analysis of privacy policies, using various automated as well as manual tools. We reveal individually many red flags in 30/108 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 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.

References

1. Arghire, I.: Thousands of mobile apps leak data from firebase databases. (2018), online article. https://www.securityweek.com/thousands-mobile-apps-leak-data-firebase-databases

2. Bengfort, J.: 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. 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
4. Columbus, L.: 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
5. Davidson, J., Schimmele, C.: Evolving internet use among canadian seniors. statistics canada research paper series. (2019), online article. https://www150.statcan.gc.ca/n1/pub/11f0019m/11f0019m2019015-eng.htm
6. Easylist.to: Easylist. (2022), online article. https://easylist.to/
7. Feal, Á., Calciati, P., Vallina-Rodriguez, N., Troncoso, C., Gorla, A., et al.: Angel or devil? a privacy study of mobile parental control apps. Proceedings of Privacy Enhancing Technologies (PoPETS) **2020** (2020)
8. Frik, A., Nurgalieva, L., Bernd, J., Lee, J.S., Schaub, F., Egelman, S.: Privacy and security threat models and mitigation strategies of older adults. In: Proceedings of the Fifteenth USENIX Conference on Usable Privacy and Security. p. 21–40. SOUPS'19, USENIX Association, USA (2019)
9. Gibler, C., Crussell, J., Erickson, J., Chen, H.: Androidleaks: Automatically detecting potential privacy leaks in android applications on a large scale. In: International Conference on Trust and Trustworthy Computing. pp. 291–307. Springer (2012)
10. Github.com: graphiql (January 2022), https://github.com/graphql/graphiql
11. Harkous, H., Fawaz, K., Lebret, R., Schaub, F., Shin, K.G., Aberer, K.: Polisis: Automated analysis and presentation of privacy policies using deep learning. In: 27th USENIX Security Symposium (USENIX Security 18). pp. 531–548 (2018)
12. Hill, S., Jansen, M.: Android vs. ios: Which smartphone platform is the best? (April 2021), https://www.digitaltrends.com/mobile/android-vs-ios/
13. Hoyt, J.: Senior citizen apps. (2020), online article. https://www.seniorliving.org/cell-phone/apps/
14. Huckvale, K., Prieto, J.T., Tilney, M., Benghozi, P.J., Car, J.: Unaddressed privacy risks in accredited health and wellness apps: a cross-sectional systematic assessment. BMC medicine **13**(1), 1–13 (2015)
15. Jindal, A., Madden, S.: Graphiql: A graph intuitive query language for relational databases. In: 2014 IEEE International Conference on Big Data (Big Data). pp. 441–450. IEEE (2014)
16. Kakulla, B.N.: Tech trends of the 50+. AARP Research: Washington, DC, USA (2020)
17. Kakulla, B.N.: 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
18. Maaß, W.: The elderly and the internet: How senior citizens deal with online privacy. In: Privacy online, pp. 235–249. Springer (2011)
19. Medium.com: Exploiting apps vulnerable to janus (cve-2017–13156)., online article (26 March 2021). https://medium.com/mobis3c/exploiting-apps-vulnerable-to-janus-cve-2017-13156-8d52c983b4e0
20. Morrison, B., Coventry, L., Briggs, P.: How do older adults feel about engaging with cybersecurity? Human Behavior and Emerging Technologies **3**(5), 1033–1049 (2021)
21. Muscat, I.: What are injection attacks (April 2019), online article. https://www.acunetix.com/blog/articles/injection-attacks
22. Oliveira, D., Rocha, H., Yang, H., Ellis, D., Dommaraju, S., Muradoglu, M., Weir, D., Soliman, A., Lin, T., Ebner, N.: 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. pp. 6412–6424 (2017)

23. Razaghpanah, A., Nithyanand, R., Vallina-Rodriguez, N., Sundaresan, S., Allman, M., Kreibich, C., Gill, P., 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)
24. Ren, J., Lindorfer, M., Dubois, D.J., Rao, A., Choffnes, D., Vallina-Rodriguez, N., et al.: Bug fixes, improvements,... and privacy leaks. In: The 25th Annual Network and Distributed System Security Symposium (NDSS 2018) (2018)
25. Sahni, S.: Firebase scanner., online article (28 February 2018). https://github.com/shivsahni/FireBaseScanner
26. Shirke, K.: Mobile security framework (mobsf) static analysis (Jan 2019), online article. https://medium.com/@kshitishirke/mobile-security-framework-mobsf-static-analysis-df22fcdae46e
27. Slane, A., Pedersen, I., Hung, P.C.K.: Involving seniors in developing privacy best practices: Towards the development of social support technologies for seniors. in: Office of the privacy commissioner of canada., 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/
28. Sqlmap.org: Sqlmap automatic sql injection and database takeover tool (January 2022), online article. https://sqlmap.org/