USENIX

THE ADVANCED COMPUTING SYSTEMS ASSOCIATION

Health Hazard, Handle with Care: Investigating the Privacy Risks of Android's Health Connect

Konstantinos Spyridakis, Ioannis Arkalakis, Michalis Diamantaris, and Sotiris Ioannidis, *Technical University of Crete;* Jason Polakis, *University of Illinois Chicago;* Panagiotis Ilia, *Cyprus University of Technology*

https://www.usenix.org/conference/usenixsecurity26/presentation/spyridakis

This paper is included in the Proceedings of the 35th USENIX Security Symposium.

August 12–14, 2026 • Baltimore, MD, USA

ISBN 978-1-939133-58-8

Open access to the Proceedings of the 35th USENIX Security Symposium is sponsored by

جامعة الملك عبدالله للعلوم والتقنية

King Abdullah University of Science and Technology

Health Hazard, Handle with Care: Investigating the Privacy Risks of Android's Health Connect

Konstantinos Spyridakis[♥] Ioannis Arkalakis[♥] Michalis Diamantaris[♥]
Sotiris Ioannidis[♥] Jason Polakis[♣] Panagiotis Ilia[♠]

[♥]*Technical University of Crete* [♣]*University of Illinois Chicago* [♠]*Cyprus University of Technology*

{kspyridakis1, iarkalakis, mdiamantaris, sioannidis}@tuc.gr

polakis@uic.edu

panagiotis.ilia@cut.ac.cy

Abstract

Android's Health Connect framework was recently introduced to unify health and fitness data management across apps, promising transparency, user control, and privacy through on-device storage and fine-grained permissions. In this paper, we present the first empirical investigation of Health Connect and show that its design suffers from inherent flaws that introduce serious privacy risks, including the creation of a covert communication channel that can be used to bypass existing access control mechanisms. Accordingly, we use dynamic analysis to systematically examine Health Connect apps, and uncover widespread privacy-invasive behaviors. We find that 60.9% of apps fail to comply with Health Connect's UI data transparency guidelines, and 25.6% do not report the collection or sharing of health records. Additionally, 19.6% of the apps exfiltrate Health Connect data (including records generated by *other* apps) over the network, combining health information (e.g., blood pressure) with personally identifiable data such as names, emails, Advertising IDs, and phone numbers. These practices violate Google's policies and undermine user expectations of control. We disclosed our findings to Android's Health Connect team, who verified our findings and informed us that our research allowed them to strengthen their review and policy enforcement processes, and take action against bad actors. While this is a step in the right direction, our research demonstrates that Health Connect's design falls short of its stated goals and, ultimately, undermines the privacy protections Android has introduced in recent years.

1 Introduction

The widespread adoption of wearable devices and embedded sensors has led to a rapid increase in the volume, granularity, and sensitivity of personal health data collected on a continuous basis. Modern devices capture a broad range of physiological and behavioral signals (including cardiovascular, metabolic, activity, and location data), enabling detailed inferences about an individual's health and daily habits. In parallel, healthcare systems are undergoing large-scale digitization, with app-mediated services, remote monitoring, and electronic health records becoming integral to care delivery and increasingly promoted by public authorities [33, 79, 105].

Health and fitness apps now constitute a significant and growing segment of the Android ecosystem. While early apps focused on low-risk activity tracking, contemporary health apps increasingly process highly sensitive data related to mental health, women's health, symptom monitoring, and chronic disease management [31, 34, 93]. The COVID-19 pandemic further accelerated adoption of mobile health technologies [63, 77, 109], with widespread use among U.S. adults [57, 65], amplifying the privacy risks associated with the large-scale collection and sharing of health data.

This necessitated dedicated OS support for managing and sharing sensitive health data, leading Android to introduce Health Connect as a centralized platform component. However, despite the major evolution that Android's permission system has undergone [50, 67, 98], prior work shows that users frequently misunderstand or ignore permission prompts, leading to broader access than intended [48, 101], while developers often over-request permissions [81]. Moreover, privacy disclosures remain largely ineffective, as they are often absent or fail to accurately describe data practices [9, 38]. Newer transparency mechanisms (i.e., Android's Data Safety Section, Apple's Privacy Nutrition Labels) exhibit systematic mismatches between disclosed and actual data collection and sharing behaviors [27, 30, 64, 107, 108]. These shortcomings are particularly concerning for apps handling health-related data [31, 93]. However, while Android's permission misuse and data sharing risks have been widely studied, no prior work has systematically examined the risks introduced by Health Connect, despite the sensitivity of the data it manages.

To address this gap, we conduct an empirical analysis of Health Connect and the capabilities it introduces. Even though its centralized design aims to improve user control by enabling interoperable health data access across applications, it also introduces substantial privacy risks. Once permissions are

granted, applications may access data contributed by other apps, exposing sensitive health information. The architecture further enables covert communication channels that bypass established security mechanisms (e.g., scoped storage [22]) and exposes metadata (e.g., package names and user or device identifiers) that can facilitate fingerprinting and persistent re-identification, particularly when combined with distinctive health attributes. Overall, our analysis indicates that, despite its stated goals of transparency and user empowerment, Health Connect reintroduces privacy risks that existing Android security mechanisms were designed to prevent.

To better understand how Health Connect is being used in practice, we conduct the first systematic investigation of Android apps that use the Health Connect platform. To that end, we employ a semi-automated dynamic analysis approach to monitor Health Connect data flows, by instrumenting Health Connect and `binder`-level health-related APIs. We initialize and test 1,068 Android apps from our dataset that declare at least one Health Connect permission in their `Manifest` files; among them, we find 351 apps that actually invoke Health Connect APIs at run time, which we then exercise and analyze systematically. Our investigation shows that 60.96% of apps fail to comply with Health Connect's UI data transparency guidelines [19] and that 25.64% of the apps do not report the collection or sharing of health records. We found that 29 apps access and leak Health Connect records created by other apps to 42 domains, with 55.17% of these apps misleading users by failing to disclose that they are transmitting health and fitness data in their Data Safety section. Cross-app data leaks include vital signs and location information, while 13 apps leak the list of installed apps through Health Connect's metadata. We have also identified cases that violate the user's privilege for anonymity and enable precise user profiling by transmitting individual-specific health attributes combined with device characteristics (e.g., Advertising ID, user name). To make matters worse, we have uncovered an app that uses shady techniques for transmitting health records to third-parties through the use of screenshots. We responsibly disclosed our findings to Google, who acknowledged the value of our research towards improving the security and privacy of the Android ecosystem. In particular, our findings helped strengthen the Play Store's review process and policy enforcement mechanisms, contributed to security improvements in Health Connect's platform, and led to enforcement actions against non-compliant actors.

As health and medicine become increasingly digitized, health-related apps will surely become more prevalent and deeply embedded in modern everyday care. As such, it is essential that the research community proactively conducts rigorous investigations of the privacy risks introduced by the proposed frameworks, before they become the foundation of a digital health ecosystem that lacks the technical safeguards to truly protect users' sensitive data from privacy-invasive practices.

Our research makes the following contributions:

- We present the first empirical analysis of Android's Health Connect and uncover fundamental design flaws that expose users to significant privacy risks.
- We present a measurement study of Health Connect usage, leveraging dynamic analysis to monitor Health Connect data flows at the system-level and capture cross-app data interactions and health data exfiltrations.
- We present an in-depth analysis of apps' run-time behavior and Data Safety section and privacy policies, and uncover illegal practices and severe data transparency violations. We responsibly disclosed our findings to Google, contributing to ongoing remediation efforts and improvements to the Play Store review process. To facilitate additional research we publicly share our code and data [7].
- We propose a context-aware and purpose-bound access control mechanism for strengthening the privacy guarantees of Health Connect.

2 Background

In this section we discuss pertinent aspects of Health Connect's design and functionality, Google's policies and requirements for apps integrating Health Connect, and the architectural design flaws uncovered through our empirical analysis.

Health Connect. Google presented Health Connect as a platform for managing health data on Android devices. While initially released as a standalone app, it was fully integrated into Android 14 as a built-in component of the operating system [32, 103]. It was introduced as a unified health platform intended to reduce fragmentation and ease developers' burden of integrating multiple APIs, which required separate connections and costly maintenance [51–53]. Beyond that, Google emphasized the benefits for users, presenting Health Connect as a way to deliver "*better, more holistic health insights*" [53] while assuring users that "*data is stored on-device, ensuring the user is in full control of their data*" [52].

The Health Connect platform introduces a centralized on-device database that provides a standardized data schema streamlining data storage and sharing across apps. It supports health and fitness records of more than fifty data types [40], while recently Google added experimental support for medical records as well [20]. Currently, Health Connect contains various data types implementing the `Record` interface, such as `HeartRateRecord`, `BloodGlucoseRecord` and `OvulationTestRecord`. Figure 1 shows an example of `ExerciseSessionRecord`, including its properties, a nested record (i.e., `ExerciseRoute`) and a metadata object that provides additional information about the record.

Health Connect apps[1] interact with the database through a set of APIs, enabling them to read, write, or update records,

[1]We refer to apps that utilize the Health Connect SDK as health-related apps, and data obtained through Health Connect (i.e., health values and record metadata) as health-related data or Health Connect data interchangeably.

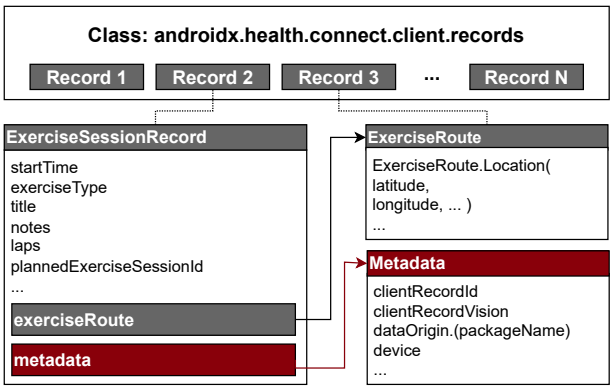

Figure 1: Health Connect records.

while users manage access through fine-grained permissions that must be explicitly granted for each data type or record. Permissions can be distinguished by the data type they protect, and between read and write operations. Apps have to declare the appropriate permissions in their Manifest file and provide a rationale describing how the user's data is used and handled. This onboarding flow is presented to users when Health Connect permissions are requested; developers are required to clearly articulate the benefits of health data access to users, and display the privacy policy dialog through the `PermissionsRationaleActivity` [23].

Data Safety labels. Data Safety is a feature in Google Play that requires app developers to disclose how their apps collect, share, and protect user data. It essentially provides users with an overview of how their data will be treated before deciding whether to install the app. For health-related data, developers are required to add the "Health and/or Fitness" label and describe, via keywords, the intended purpose.

Policies and requirements. Considering the sensitive nature of Health Connect's data, Google's policy requirements establish clear expectations for apps utilizing the Health Connect platform [88]. Data obtained through Health Connect permissions must adhere to the *Health apps policy* [59], and apps are also subject to the *User Data Policy* and *Prominent disclosure and consent requirements* [58,60]. Developers must limit permissions to approved use cases, obtain informed consent, disclose data practices transparently, maintain strong security safeguards, and comply with global health data regulations.

Importantly, apps must provide a comprehensive and accurate privacy policy that clearly explains what health and fitness data they collect/access, how the data is being used, and provide a clear description of the benefits for the user. Furthermore, Health Connect data access and use must adhere to specific limitations: i) *"provide or improve the appropriate use case or features visible in the application's user interface"*, ii) *"User data may only be transferred to third parties with explicit user consent, for security purposes, to comply with applicable laws or regulations or as part of mergers/acquisitions"*, (iii) *"Human access to user data is restricted unless explicit user consent is obtained, for security purposes, to comply with laws, or when aggregated for internal operations as per legal requirements"*. All other transfers, uses, or selling of Health Connect data are strictly forbidden. This includes transferring or selling health data to advertising platforms, data brokers and information resellers, or using it for serving ads, including personalized or interest-based advertising [87].

Empirical analysis. We performed preliminary experiments with Health Connect apps between May and December 2024, in conjunction with an empirical analysis of Health Connect's design, implementation, and documentation. This resulted in us identifying concerning flaws in the Health Connect platform that enable behaviors that do not align with Google's policies and requirements, thereby limiting user control and transparency, and undermining their privacy.

The first fundamental flaw in Health Connect's design is that granting an app permission to access a specific record type enables access to *all* such record instances independent of the app that created them. For example, `android.permission.health.READ_SEXUAL_ACTIVITY` allows an app to access all record instances that belong to the `SexualActivityRecord`. In practice, this design choice can lead to over-privileged cross-app data access, as apps are known to request more permissions than necessary. Additionally, cross-app data interactions can lead to scenarios where health data from one app is accessed by another and then transmitted over the network without the user's knowledge. As an illustrative scenario, consider an app that has been granted access to `SexualActivityRecord`. Based on Health Connect's current design, an app can read *all* sexual activity records generated by other apps, including highly sensitive attributes, such as whether encounters are marked as protected or unprotected. Additionally, it can infer sexual activity frequency based on the `Time` fields of these records. By observing longitudinal patterns and changes in these records, this app can infer intimate personal and medical information (e.g., changes in reproductive health or sexual function).

The metadata included in Health Connect records is also accessible across apps, and contains the record's origin (i.e., package name), unique record IDs and device characteristics (e.g., type, model, manufacturer). Such data can be used as additional device fingerprinting identifiers, combined with Android identifiers (e.g., Advertising ID) and unique health values to create persistent identifiers. Moreover, `Metadata.DataOrigin.getPackageName()` can be used to identify other health-related apps. This is a *known* privacy risk [85,96], since identifying as few as four apps installed on a user's device can enable precise identification [8]. In the case of Health Connect apps, the risk is significantly amplified due to the very nature of these apps [85]. Crucially, while Android has taken explicit measures to prevent apps from obtaining information about other installed apps since v11, this is now possible again through the Health Connect platform.

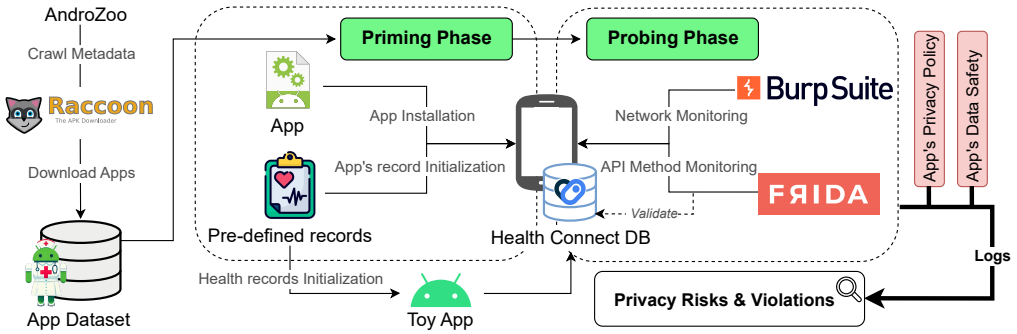

Figure 2: Overview of our dynamic cross-app data analysis.

We also identified a covert-channel for cross-app communication and data exchange. Specifically, certain record types expose free-form string fields (e.g., `Title` or `Notes`). One app can, therefore, store arbitrary information in these fields, which is readable by any other app that has access to the same type of record, effectively enabling cross-app data exchange outside of the intended data-sharing model. To make matters worse, we found that apart from simple string values, the system indirectly allows the exchange of more advanced data types by not enforcing strict character limits. Specifically, we were able to use the record's `Title` field to exchange a base64-encoded `PNG` image (with a size of 5.24KB) between two mock apps, therefore granting image access to an app that did not have the appropriate Android file permission. This flaw essentially allows apps to bypass Android's current access control mechanisms, which explicitly prevent such data exchanges (e.g., scoped storage [22]).

3 Methodology

In this section, we present our analysis framework and workflow for studying the use of Health Connect within Android apps. First, we present our dynamic cross-application data analysis pipeline (§3.1), then we discuss the rationale behind our design choices (§3.2), and conclude with details of how we conduct system and network monitoring (§3.3).

As our analysis targets privacy risks and violations of health-related apps and potential cross-app interactions and leakage though the Health Connect SDK, we installed and initiated all apps in our device prior to our analysis, so as to populate the device's database with records from all of the apps. Our pipeline, as can be seen in Figure 2, is divided into two distinct phases: *Priming* and *Probing*. Overall, our workflow consists of automatically identifying Health Connect apps, populating the device's Health Connect database with data generated by the apps in our dataset, manually interacting with each app while intercepting cross-app data flows and, finally, correlating system and network logs to identify privacy risks and violations.

3.1 Dynamic Analysis Pipeline

Here we describe our dynamic analysis pipeline and experimental setup for analyzing health-related apps.

Priming. We interacted with Android apps to trigger their functionality and granted access to all Android permissions that each app requests. We also completed any initial setup steps requiring additional user input (e.g., inputting age, sex, date of birth, weight, height, blood pressure) by populating them based on a pre-defined list. This list contains unique values with decimal numbers for every Health Connect record, facilitating identification within network traces. To ensure that every record in the device's database has at least one value, we created a toy app that sets all the available records (39 during the time of our experiments) in the Health Connect database [40]. This step populates the device's database with every available record and not only those that are added by the apps in our dataset. Consequently, it enables us to systematically test whether apps access or handle data beyond what is necessary or disclosed, providing a more comprehensive coverage of app's data collection and sharing practices.

Probing. During this phase we interact with each app for 5-15 minutes (depending on the available functionality) and monitor the app's use of Health Connect APIs and network traffic. We refrain from using automated tools (e.g., monkey testing or other UI-exercising tools) to dynamically analyze apps, as these tools are not optimized for the contextual input generation needed in our analysis (see §3.2). Our interaction consists of actions such as signing in, interacting with all health-related elements, adding input/records and completing various app challenges or tasks (e.g., performing a specific exercise, playing mini games and enrolling in fitness courses). Also, we re-populate the database on a daily basis with pre-defined records using our toy app. We included this step, as our analysis spans several days and apps may request access solely to the most recent records.

3.2 Design Choices

Health Connect apps handle sensitive data, which is often protected through legislation, that can reveal intimate details

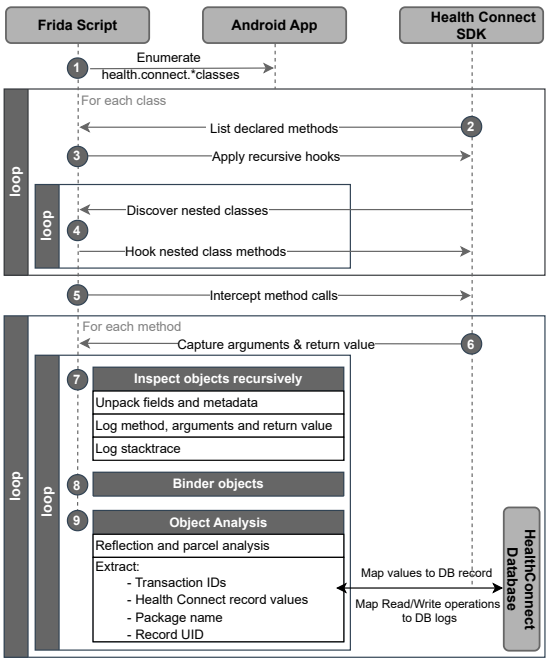

Figure 3: System monitoring process.

about a user's physical condition and daily habits. To accurately assess how apps handle this data, we adopted a carefully designed analysis setup and a semi-manual process, intentionally prioritizing precision over scalability. First, our analysis setup grants all permissions requested by each app and populates the device's database with all available records. While this setup may not reflect all real-world user scenarios, as some users may grant fewer permissions or have less data available, it enables consistent and comprehensive evaluation of apps' behavior, allowing us to better uncover the underlying risks that exist. Next, our investigation requires manual exercising of apps, since maximizing coverage of Health Connect-related code paths depends on contextual input, and only human-guided interaction can reliably trigger complex user interface flows. Many app activities are exposed only through nuanced, sequential actions that are difficult, and in some cases infeasible, to reproduce through automated means across varied UI interfaces. Moreover, the highly sensitive nature of the data necessitates manual processing of network logs to eliminate false positives that could be introduced by automated approaches.

3.3 System and Network Monitoring

Here we provide technical details about our system and network monitoring process, which allows us to bridge cross-app data flows from the operating system to Health Connect data transmitted over the network.

System. We leverage the Frida framework [5] to monitor and examine the behavior of calls to *all* Health Connect APIs. Figure 3 presents our system monitoring process. At a high-level, our system intercepts all Health Connect classes and methods, including those within nested classes (❶ - ❺) and systematically instruments method calls to capture and inspect both argument objects and return values (❻ - ❾).

Method interception. Apart from offering functionality for reading or writing health records, the Health Connect SDK also provides APIs for accessing the records' metadata. These metadata APIs can retrieve device information (e.g., manufacturer, model, type), and also record operations in the database (i.e., read, write, update) along with the associated metadata for each record (e.g., uuid, timestamps and package names). Initially, we hooked all `health.connect.*` classes and methods involved in reading and inserting records (❶, ❷), applying recursive hooks (❸) to capture nested classes (❹) and systematically instrumenting all declared methods within them. Then, we inspected method calls, arguments, and return values, to identify access to record values (❺). Since the arguments and the return values (❻) in several cases consisted of Java objects containing multiple fields and metadata, we recursively hooked their internal structures (❼) to extract additional relevant information. To ensure comprehensive monitoring of Health Connect data flows, we instrumented `binder`-level health-related APIs, since Health Connect values are read from and written to the database through the Android Binder. Additionally, we implemented hooks targeting the package names of apps utilizing the Health Connect platform. This process allowed us to map records and data values to their corresponding application, since without these hooks we would not be able to determine which app generated them.

Stacktrace analysis. During method interception, we captured the stacktrace of each API, and analyzed whether the first- or third-party is responsible for access to Health Connect information using a list of third-party library package names based on prior research [29, 66, 95]. While third-party libraries were present in our dataset, we did not observe any third-party library accessing Health Connect APIs.

Object inspection. Our approach enabled us to capture Android parcels [21] (❽), which required separate handling due to the complexity and variations in variable names, object types, and classes, that prevented tracing via the public Health Connect API alone. This necessitated combining reflection, object inspection, and parcel analysis, to extract relevant data (❾). We verified that all method variants were traced, objects were unpacked successfully, and raw serialized values exchanged via Binder IPC were revealed to extract pertinent information (e.g., uuid, package names).

Database Validation. We extract logs from the device's database and cross-correlate them with the Health Connect data flows that we monitor. These logs provide us with additional insights (timestamps, uuids, etc.) alongside the values' format in the database. This step is crucial since apps may parse record values into various formats before the data is inserted into the database, which would result in breaking our

mapping of pre-defined records (see *Priming phase*), that is required to identify these values in the network logs.

Network. We setup a transparent BurpSuite [4] proxy and monitor each app's network traffic for Health Connect values. Specifically, we check all the values in our pre-defined list using string-matching, accounting for common transformations (e.g., GZIP, Base64, MD5, SHA256), as well as multiple transformations (e.g., URL encoding followed by Base64, MD5 and SHA256), based on techniques commonly used in detecting network leaks [35, 43, 80]. We perform this same step for Health Connect metadata that exists in the device's database and manually verify all network leaks.

4 Dataset and Experimental Setup

In this section we describe how we collected our application dataset and provide details about our experimental setup.

Dataset. We collected daily AndroZoo [13] snapshots of Android apps and their metadata, between February 2024 and March 2025. We identified apps that make use of the Health Connect SDK by checking the permissions in their Manifest file. After Google's stringent policies on the use of health-related data were announced in March 2025 [88], we downloaded the latest version of each app in our list from Google Play, using Raccoon [1]. We discarded apps that did not include Android health permissions (i.e., `android.permission.health.*`), since access to Health Connect APIs requires developers to explicitly add the appropriate health permission in the app's Manifest. Furthermore, during our manual dynamic analysis we excluded from our dataset apps that required a paid subscription (24) or membership (361), compatible external device such as a smartwatch (39), or if they didn't actually issue any calls to the Health Connect APIs (283). Additionally, we were not able to analyze 10 apps because they failed to run after installation; this behavior persisted even on a clean, non-rooted device for five out of these apps. The remaining five apps executed normally without a network proxy and in a non-rooted device, indicating the use of advanced anti-analysis techniques [111]. In total we preprocessed 2,348,217 unique apps, identified and manually exercised 1,068 apps with Health Connect permissions, with our final dataset consisting of 351 health-related apps.

Device setup. Our testbed consists of a Google Pixel 6 device running Google's AOSP version 14 and a proxy server using BurpSuite. We set up our device with a Gmail account used for logging into apps whenever this is supported, either through manual login or using Google's SSO. We added a SIM card to make the environment more realistic and support OTP messages as well. We use Magisk [106] to root the device and Frida server to monitor Android APIs. Furthermore, we intercept network traffic by configuring the device with the BurpSuite's certificate and using Frida scripts for bypassing SSL pinning and anti-root or anti-tampering detection. Finally, our experiments were conducted within Europe.

5 Measurements and Investigation

Here we present the results of our study of Health Connect apps' data practices and cross-app interactions. We also examine whether apps disclose collecting and sharing of health-related data in their Data Safety and whether they adhere to Google's privacy policy and prominent disclosure requirements [59]. We note that a complete analysis of apps' Data Safety sections and privacy policies (i.e., for non-health-related data) is out of the scope of our work, since prior studies have extensively explored such issues. We focus our analysis on identifying cross-app interactions and health-related leaks.

5.1 Data Safety and Policy Compliance

Applications that use the Health Connect SDK must comply with Google's regulations designed to enhance data transparency and enable users to make informed decisions about their sensitive data. Accordingly, developers are required to clearly disclose any access to, or sharing of, Health Connect data in the app's Data Safety section, and provide a comprehensive and accurate privacy policy.

Data Safety disclosure. We followed the methodology of [27] in order to assess whether there are inconsistencies between the Data Safety section of health-related apps and their actual run-time behavior. As we focus only on the case of health-related inconsistencies in apps' Data Safety sections, we are looking specifically for the existence of the "Health and/or Fitness" label, and whether developers describe via keywords their intended purpose, as required. Specifically, to assess Data *Collected* disclosures, we dynamically examined all triggered Health Connect APIs (§3.3) whose arguments or return values contained a Health Connect record or subfield. To assess Data *Shared* disclosures, we checked whether the pre-defined record values of the Health Connect database exist in the app's network logs. This process was independently carried out by two researchers to ensure no cases were overlooked. When such data accesses or transmissions were observed, we checked whether the "Health and/or Fitness" label was disclosed. Our analysis shows that apps do not universally follow requirements, as 54 (15.38%) do not disclose collecting Health Connect records and 47 (13.39%) do not disclose sharing them. Overall, 90 (25.64%) apps do not report the collection or sharing of health records, thus misleading users and affecting their ability to make informed decisions.

Privacy policy requirement. Two researchers independently examined the privacy policies of the 351 apps in our dataset for the policy requirements described in §2, using a two-step process. First, they checked whether the app provides a privacy policy at all. For apps that do, they checked whether the policy describes access or handling of Health Connect or health-related data. The initial raters agreed in 97.43% of the cases; from the disputed nine cases, only three were found to have appropriate privacy policies. Overall, we found six

Table 1: Device's Health Connect records.

Records created by applications in our dataset	
R1: ActiveCaloriesBurnedRecord	R2: BasalMetabolicRateRecord
R3: BloodGlucoseRecord	R4: BloodPressureRecord
R5: BodyFatRecord	R6: BodyTemperatureRecord
R7: BodyWaterMassRecord	R8: BoneMassRecord
R9: CervicalMucusRecord	R10: CyclingPedalingCadenceRecord
R11: DistanceRecord	R12: ElevationGainedRecord
R13: ExerciseRoute	R14: ExerciseSessionRecord
R15: HeartRateRecord	R16: HeartRateVariabilityRmssdRecord
R17: HeightRecord	R18: HydrationRecord
R19: LeanBodyMassRecord	R20: NutritionRecord
R21: OxygenSaturationRecord	R22: OvulationTestRecord
R23: PowerRecord	R24: RespiratoryRateRecord
R25: RestingHeartRateRecord	R26: SexualActivityRecord
R27: SleepSessionRecord	R28: SpeedRecord
R29: StepsRecord	R30: TotalCaloriesBurnedRecord
R31: Vo2MaxRecord	R32: WeightRecord
Additional records created only by our toy application	
R33: BasalBodyTemperatureRecord	R34: FloorsClimbedRecord
R35: MenstruationFlowRecord	R36: MenstruationPeriodRecord
R37: StepsCadenceRecord	R38: WheelchairPushesRecord
R39: IntermenstrualBleedingRecord	

apps that do not provide a privacy policy, and 17 apps that fail to describe in their policies that they access health data. Surprisingly, we found three apps that share or sell this data to advertisers in an aggregated anonymized form, a practice that is strictly forbidden according to Google's policies, even in aggregated and anonymized form. Moreover, we examined whether the permission rationale presented to the user when apps request access to health permissions provided a valid privacy policy link that clearly articulates the benefits of health data access to users. Specifically, we checked whether the link exists, resolves, and leads to a valid policy. We consider policies valid if they are labeled as "Privacy Policy" or "Terms of Use", or if they describe data access or handling practices. We identified that 60.96% of the apps fail to adhere to Health Connect's UI data transparency guidelines [19, 23], as the `PermissionsRationaleActivity`'s link fails to direct to a valid privacy policy.

5.2 Health Connect Data Access

Here, we present the results of our analysis of Health Connect data access across health-related apps.

Health Connect records. The Health Connect SDK provides a diverse number of classes for various health records [40]. Upon completion of our experiments, the device's Health Connect database had 11,504 records created by 91 apps. Table 1 lists the records that have been created by the apps in our dataset, and additional records that were created by our toy app. Each record has its own properties, classes, methods and fields. For example, the `BloodPressureRecord` contains the systolic and diastolic blood pressure values, includes attributes about the body's position when the measurement was taken, and the record's generation timestamp. Several of these records contain sensitive information, such as `SexualActivityRecord`, `OvulationTestRecord`, `MenstruationFlowRecord` and `MenstruationPeriodRecord`. We captured 100 (28.49%) apps accessing 2,334 records created by other apps; 25 did so without disclosing the collection of this data in their Data Safety's *Collected* section. Figure 4 (left) shows that 15.67% of the apps access at least five records that do not belong to them, while 10% access at least 20 records. In Figure 4 (right) we observe that 17.37% gather data from at least four apps. While apps may access records individually, we also found 58 using `HealthConnectClient.aggregate()` to access multiple records (as an avg/min/max value) across a specified time period.

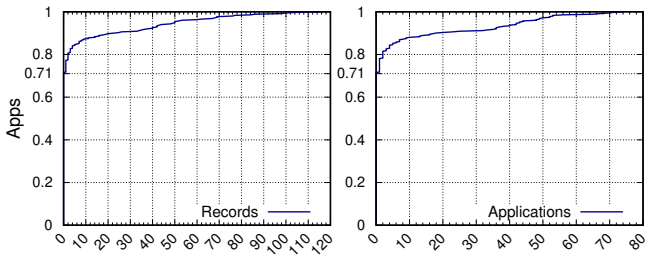

Figure 4: CDF of apps reading other apps' records (left) and the distribution of the apps they collect data from (right).

Furthermore, we found that `ExerciseSessionRecord` has the nested `ExerciseRoute.Location` class which may contain one or multiple location points (i.e., latitude, longitude and altitude) and a timestamp of when the record was generated. Four apps created records that included the user's location, which constitutes a significant privacy risk [68, 76], as location data enables the inference of diverse sensitive data [45]. Three of these records are being accessed by five different apps and can be used to identify the user's past locations with high precision, or specific routes recorded during activities like running or walking. Records such as the `ExerciseSessionRecord` and the `SleepSessionRecord` also contain a `Title` or `Notes` string field which can be used by apps for adding richer contextual information. Unfortunately, they can also serve as a covert-channel (see §2).

Health Connect metadata. Health Connect data types that implement the `Record` interface have a `Metadata` object [41, 42] that provides relevant information about the record. It includes, among other elements, a Health Connect unique identifier (uuid) assigned upon creation, a description of the device that created the record (e.g., Pixel 6, Xiaomi smartscale, HEALBE smartwatch, Galaxy ring), the app's package name, and the timestamp of the record's last modification. Our investigation reveals that 120 apps access the `Metadata.DataOrigin.getPackageName()` of various records in our database, which reveals a subset of the user's installed apps (i.e., other apps with Health Connect records). Identifying installed apps is a *known* privacy risk [85, 96], which Android has explicitly prevented since v11, that is now

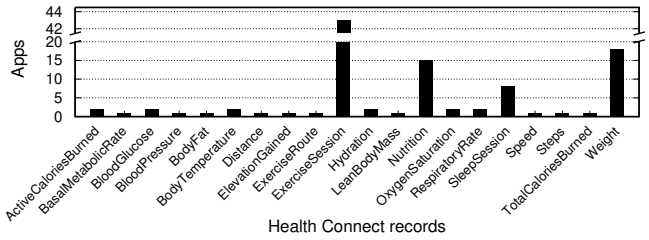

Figure 5: Health Connect leaks by records.

being reintroduced for health-related apps through the Health Connect platform. Additionally, we found 120 apps using `Metadata.getDevice()` to obtain information about the device that added the record. The `Metadata.getDevice()` object [18] provides information about the device's type (phone, watch, scale, chest strap, etc.), model and manufacturer. In our case, these correspond to `DEVICE_TYPE_PHONE`, `Pixel 6` and `Google` respectively. This data can be combined with other device characteristics obtained from OS (Advertising ID, Build ID, Android version, etc.), and individual-specific health attributes obtained from the Health Connect platform (e.g., height, bone mass), to create persistent identifiers that allow consistent re-identification of the user.

5.3 Off-Device Data Exposure

After interacting with all the apps in our dataset, we manually analyzed their network logs and identified 69 (19.60%) apps that transmit Health Connect data over the network. We note that Health Connect data may contain any type of information that can be retrieved using the Health Connect SDK, including record values and metadata. Since we are able to identify whether the same value (e.g., weight) has been leaked from records of different apps, we count such cases individually.

Health Connect leaks. While prior work defined leaks as any data not disclosed in the Data Shared section [27], transmitting health data over the network fundamentally contradicts the intended Health Connect design, and creates avenues for privacy loss and invasive practices. As such, we adopt a stricter definition, treating any off-device health data transmission as a leak. We observe that 5.98% of apps have at least five data leaks, while a single app (`com.fitbod.fitbod`) is responsible for 481 leaked Health Connect values, without disclosing this information in its Data Safety *Shared* section. We also identified six applications leaking data in aggregate format from the `ActiveCaloriesBurnedRecord`, `TotalCaloriesBurnedRecord`, `DistanceRecord`, `BasalMetabolicRateRecord` and `NutritionRecord` records.

In Figure 5 we present the leaked data based on the Health Connect records they belong to. We observe values from 20 records being exfiltrated, with the most common records being `ExerciseSession`, `Nutrition`, `SleepSession` and `Weight`. Additionally, in Figure 6 we present the number of apps leaking Health Connect data over the network based on the corresponding record value that is being leaked. The user's location (latitude, longitude), stored in the `ExerciseRoute` record of three apps (toy app, `com.fitbit.FitbitMobile` and `fi.polar.polarflow`), was accessed and leaked by `com.habito.edf` to two domains (*api.heiaheia.com* and *trackgpx-heiaheia-com.s3.eu-west-1.amazonaws.com*). We note that `com.habito.edf` does not disclose the collection and sharing of health-related data, or the sharing of location data, in its Data Safety section.

Furthermore, the most leaked values are the `ExerciseSessionRecord` fields of `Title` and `Notes`, which are exposed by 32 and 26 apps respectively. These values can be inserted manually by users or automatically by apps. Examples of app generated values include indoor or outdoor exercise types ("Cycling", "Office workout", "Wednesday Evening Run"), muscle groups targets ("Full body", "Sit-ups", "Push-ups") or gymnastics types ("Pilates", "Muay Thai") and can be used to construct a distinct exercise and fitness profile of the user. The same applies for the `SleepSessionRecord` with two apps leaking the `Title` and `Notes` fields. Concerning the `NutritionRecord`, we found 15 apps leaking information such as `Nutrition.name`, nutrient values (protein, magnesium, sodium) as well as fat and cholesterol. These values reveal the user's eating habits and dietary preferences. Lastly, three apps leak the vital signs, blood pressure and body temperature, with the `com.squats.fittr` app failing to disclose the latter in its Data *Shared* section.

Health Connect metadata leaks. Records' `Metadata` objects contain sensitive information that can be used to accurately fingerprint the user and create persistent identifiers. Table 2 shows 13 apps accessing and leaking other records' package names, thereby bypassing Google's enforcements concerning access to the user's installed apps [16]. We observe that while seven apps leak between one and six package names of other health-related installed apps, six apps leak between 14 and 69 app package names. Apart from accurate user identification [8], this data can be used to infer additional personal sensitive information. For example, the `health.mydiabetes` package name reveals that the user probably suffers from diabetes, while the `org.iggymedia.periodtracker` suggests that the user is a female within a specific age range. We also observe apps leaking the record's uuid through `getClientRecordId()`, a unique, auto-generated identifier created by the Health Connect platform for each new data object. While the uuid alone does not reveal the record's values, these identifiers can also be misused for user re-identification, as is the case with the list of installed apps.

Cross-app data leaks. We observe that while several apps access and send health-related data to their own domains, 29 apps access on average data from 23 apps and leak these values across one to four domains. In Figure 7 we visualize all cross-app interactions and the transmission of Health Connect data to various domains. Blue nodes (●) indicate apps with

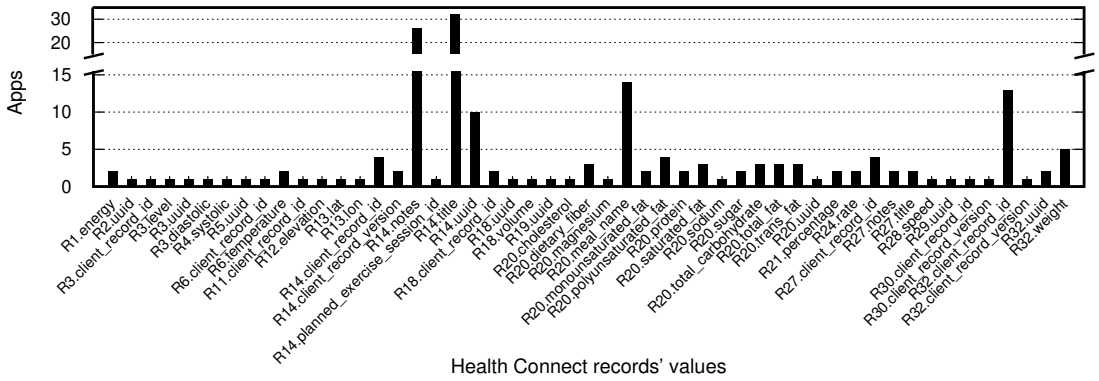

Figure 6: Health Connect leaks by records' values. *R#* refers to the record id of Table 1.

Table 2: Apps leaking other apps records' package names.

App	Package names	App	Package names
health.mydiabetes	69	com.myfitnesspal.android	5
com.ultimate531	43	com.motionapp.motionapp	4
com.catapult.fitup	41	com.sillens.shapeupclub	3
com.hasz.gymrats.app	41	es.as.deportesapp	1
coach.zing.fitness	35	com.meditation.live	1
com.guavahealth.app	14	com.runnii.walkiiapp	1
com.whoop.android	6		

Health Connect records, while black nodes (●) indicate apps that may have their own health records but are also accessing and leaking records of one or more apps. Red nodes (●) indicate domains receiving data from Health Connect apps, while domains with red text are classified as analytics and advertising. We label only red nodes with the corresponding network domain name to preserve visual clarity and highlight cross-app interactions and network transmissions.

Overall, 29 apps that access records that do not belong to them, leak data to 42 domains, with 10.34% and 55.17% of these apps misleading users by failing to disclose in their Data Safety section that they are collecting and transmitting health and fitness data respectively. Regarding the domains receiving this data, apart from *online.fitdays.cn* that receives the user's weight from two apps belonging to the same developer, we did not identify any other domains receiving data from multiple apps. Moreover, cross-app leaks include sensitive health values (blood glucose/pressure), location information (exercise routes/location) and record's metadata such as package names of other apps. Our investigation shows that apps access multiple types of records from other apps based on the health permissions they are granted. 12 apps (e.g., `com.catapult.fitup`, `health.mydiabetes`) leak a plethora of other apps' records using JSON objects that contain the package name of the records' origin and the records' values. In certain cases the leaked values were encoded using custom transformation techniques (e.g., base64) or inside GZIP archives.

Furthermore, we identified 56 apps transmitting the user's name/surname, email, Advertising ID, phone number, Build version and MCC+MNC (Mobile Country Code, Mobile Network Code), together with health-related data, such as the blood glucose/pressure, body fat/temperature/lean-mass, exercise routes, hydration, nutrition, respiratory rate, weight, steps, calories burned and sleep records. For example, the app `com.sillens.shapeupclub` accesses `NutritionRecord`'s values (e.g., sodium, cholesterol, sugar, total fat), the `ExerciseSessionRecord`'s `Title` and uuid from three different apps and sends this data combined with the user's email and full name to the domain *api.lifesum.com*. While we do not know how the data is being used after leaving the user's device, the app's Data Safety *Shared* section states that "Fitness info" is shared for the purpose of "Analytics, Advertising or marketing and Personalization". We identified five more apps (`com.freeletics.lite`, `au.com.medibank.phs`, `com.fatsecret.android`, `com.sillens.shapeupclub` and `com.withings.wiscale2`) that use the label "Health Info" or "Fitness info" for the same purpose under their Data Safety section. We note that under Google's guidelines such practices are forbidden [87], and exfiltration of health data combined with detailed personal information violates the user's anonymity. Additionally, the Data Safety and privacy policies for three of these apps provide conflicting statements, as the latter explicitly states that they never share health information for marketing or advertising.

We analyzed domains receiving Health Connect data using the McAfee classifier [73], DuckDuckGo [46] and VirusTotal [6], and found that the majority seem to function as apps' backend APIs. Surprisingly, as shown by the domains marked with red text in Figure 7, we found three domains that are flagged by DuckDuckGo as analytics (*amplitude.com*) and advertising (*pubads.g.doubleclick.net*). The `com.lyfta` and `com.ellisapps.itrackbitesplus` apps send the `ExerciseSessionRecord`'s `Title` and `NutritionRecord`'s meal name to *amplitude.com*. In the case of `com.lyfta` this information is sent along with the Advertising ID, Build ID, full name and email. We examined their Data Safety *Shared* section and both apps do *not* disclose this. Next we reviewed their privacy policies, since it is mandatory for apps accessing or handling health-related data to have a privacy policy [59],

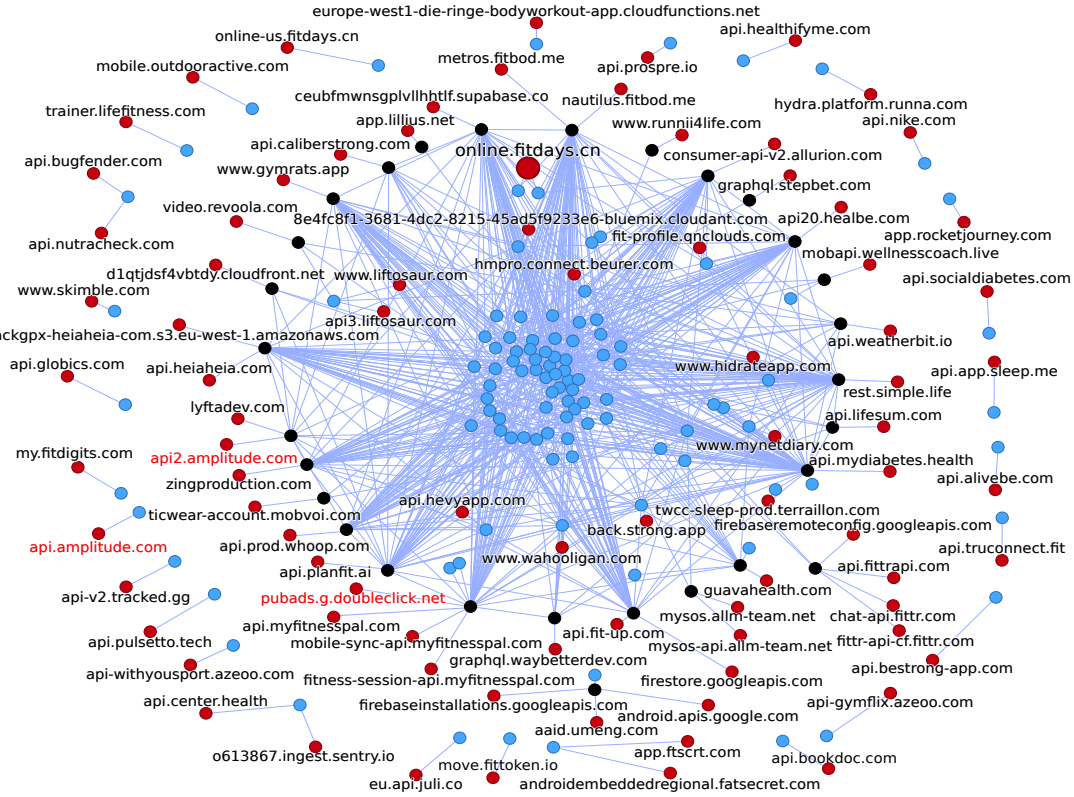

Figure 7: Cross-app data interactions and leaks. Blue nodes indicate apps with Health Connect records, while black nodes indicate apps that may have their own health records but are also accessing and leaking records of one or more apps. Red nodes indicate domains receiving data from Health Connect apps, while domains with red text are classified as analytics and advertising.

explaining the types of data being accessed or collected, and how the data will be used and/or shared [60]. The `com.lyfta` app does not have a privacy policy which violates Google's practices concerning Health Connect apps. The privacy policy of `com.ellisapps.itrackbitesplus` explicitly states that it shares user data to improve user experience and serve more relevant and tailored information, but does not mention other purposes like transferring to advertising platforms, data brokers, or information resellers. While we cannot label this as a direct violation since Amplitude is an analytics company, that is not the case for `com.myfitnesspal.android`. This app sends the `ExerciseSessionRecord`'s `Title` along with Advertising ID, Build ID and MCC+MNC to Google's advertising domain *pubads.g.doubleclick.net*, in direct violation of Google's policies for health-related data [87]. Additionally, the app does not disclose sharing "Health and fitness" data and, to make matters worse, features an inaccurate and deceptive privacy policy.[2]

Background access and past records. Android offers functionality for continuous monitoring of the Health Connect database using `android.permission.health.READ_HEALTH_DATA_IN_BACKGROUND`. Apps with this permission are able to access the database without being actively used. Furthermore, `android.permission.health.READ_HEALTH_DATA_HISTORY` allows apps to read the entire database history, contrary to the default policy that allows accessing records of other apps only during a 30-day period.

We found the `so.obvio.lina` weight-tracking app accessing the database's history. Additionally, this app runs in the background and is capable of retrieving information such as when and which apps inserted records in the database, their records, and details regarding when the user installed other health-related apps. Through further experiments with this app, we found that it accesses records from 50 other apps and transmits them to a backend API along with the user's email, age, mobile carrier, weight, height, body fat percentage and the records' timestamps. Surprisingly, we found that it also transmits an image to *ingest.sentry.io*. This image is a screenshot that was taken without our knowledge or consent during our interaction with the app, and contained health-related information (weight records). This practice raises significant privacy concerns, as it enables a third-party entity to access sensitive personal health information, potentially including insights about other installed apps and their recorded data. While the privacy policy of

[2] "The unique information you choose to send from Google Health Connect is not transferred by MyFitnessPal to third parties for marketing and advertising" [78].

this app states that they use third-party providers for analytics, crashlytics, and event tracking, taking screenshots should be *explicitly* disclosed in the relevant privacy policy. Moreover, screenshots should *never* leak health-related data. We found that the app does not stop the background data collection when the user performs a UI swipe up action (i.e., attempts to terminate the app), and the only way to successfully stop the background data collection is to force-stop the app through the Android system settings. We also identified 11 more apps with background access to the database. Seven of these applications accessed records (`ExerciseSessionRecord`, `ActiveCaloriesBurnedRecord`, `StepsRecord`, and `TotalCaloriesBurnedRecord`) from other apps, while two leaked health records alongside the user's email and full name.

Health insurance apps. Our dataset contains six apps developed by health insurance companies (two of those promoted as wellness apps) that request Health Connect permissions and use relevant APIs; while none of them accessed health-related data during our experiments, they did access and exfiltrate PII (e.g., name, email, MCC/MNC). The lack of Health Connect API access is due to them requiring specific information during account creation (e.g., insurance policy number, government ID number) that we cannot spoof. However, even though we were unable to fully exercise the apps for this reason, our static analysis indicates that they include code for accessing Health Connect records. Notably, we found an app that does not disclose any sharing of health data in its Data *Shared* section, yet lists "Advertising or marketing" as a justification for collecting health data in the *Collected* section. Even more strikingly, while the privacy policy of `ch.swica.benevita2` explicitly advises users "not to enter any sensitive data (such as health data, including diagnoses and symptoms, or financial problems)", this statement stands in direct contrast to the app's Data Safety declaration, which indicates that it collects fitness data.[3] Considering that some insurers leverage users' social media activity to adjust premiums or deny claims [2,3], granting similar entities unrestricted access to health data could enable even more privacy-invasive practices. This underscores a fundamental risk inherent in Health Connect's centralized data management architecture.

5.4 Analysis Summary

Our study highlights fundamental privacy risks in Android's Health Connect platform, and presents app violations with severe implications. First and foremost, it shows that apps accessing health records created by other apps is common among Health Connect apps; in many cases, this sensitive data, and records' metadata, are transmitted off device. To make matters worse, these apps often fail to disclose the collection or sharing of health data in their privacy policies and/or Data Safety sections, thereby misleading users about their data practices. In addition, our study shows that the majority of apps fail to comply with Health Connect's UI guidelines, as the onboarding flow presented fails to direct to a valid privacy policy, while certain apps have no privacy policy at all.

Our investigation also reveals behaviors that directly violate Health Connect's guidelines. We discovered a health app taking a screenshot that contains health data, which was captured without our knowledge or consent, and transmitted to a third-party domain. Furthermore, we identified apps sharing health data for marketing and advertising purposes, in direct violation of Google's guidelines. Crucially, one app was found to transmit health-related data along with unique device identifiers to a Google advertising domain, constituting a clear breach of Google's policies regarding health-related data, and highlighting the absence of adequate oversight and control mechanisms.

6 Mitigations

In this section, we propose design changes and enforcement mechanisms to address the privacy risks uncovered by our research. These measures inevitably introduce additional complexity, as more granular controls, purpose-binding, and systematic enforcement may add steps to consent flows and compliance checks. While such changes could slightly reduce the seamless interoperability that Health Connect was designed to provide, they significantly enhance transparency and user control over their data.

Context-aware access control. A systemic flaw of the Health Connect design is that records are accessible to any app with the corresponding permission type, regardless of which app created the records. Even though records include provenance information (i.e., the originating app's package name), this information is not taken into account by the access control mechanism when making authorization decisions. We propose incorporating data provenance into existing access control decisions. Access authorization should be granted based on the tuple $\langle$*requesting_app*, *record_type*, *originating_app*, *purpose*, *timeframe*$\rangle$. By default, apps would access only their own records, while accessing records created by other apps would require explicit user authorization for a specific purpose within a given timeframe.

Purpose-bound data access. While provenance limits cross-app data access, it does not allow users to make informed decisions on how their data can be used by the requesting apps. Building on prior work on purpose-binding [91], we propose requiring each Health Connect data access request to specify an explicit purpose of use. This purpose of use should align with the app's Data Safety disclosures and privacy policies, similarly to how Android handles the location permission by displaying a permission rationale dialog [17]. As such, Android can then enforce purpose-specific policies based on

[3] According to Android [86], fitness data includes sensitive information, e.g., physical activity and heart rate. Recent studies have shown that heart rate data from wearable devices can be used to detect cardiac conditions [70].

the user's preferences and settings, such as restricting certain records for specific purposes (e.g., analytics). This approach builds on top of Google's current mechanisms, enabling a more fine-grained access control, and making existing policy requirements [58, 59] enforceable at runtime.

Stringent metadata access. We recommend placing metadata access under the same provenance and purpose-bound controls as record data. Specifically, record's origin package names or device identifiers would be returned only when strictly necessary for the declared functionality. In addition, record's package names could be randomized and revealed only when a data request is accompanied by a declared purpose and explicitly authorized by the user.

Automated enforcement and vetting. On the policy side, instead of relying solely on self-disclosure, automated checks could be incorporated into Google Play's vetting process to ensure compliance. Static analysis of app's code would enable detection of Health Connect data uses that are inconsistent with the declared purposes, while dynamic analysis of runtime behavior and network flows could reveal undeclared transmissions. Although these techniques may impose scalability challenges, they would significantly strengthen user control. In addition, consistent auditing of apps' privacy policies and ensuring through the aforementioned technical means that app behavior complies with their stated policies, would further align practice with intention and increase transparency.

7 Discussion

Our findings demonstrate that despite Google's assurances of on-device data storage, transparency, and user control, users are not in full control of their health data. Here we further discuss the structural flaws in Health Connect's design, as well as issues related to policy oversight and enforcement.

Architectural design. Health Connect was designed to simplify interoperability by centralizing health data, while allowing cross-app access. This reduces fragmentation and provides convenience for both developers and users, but also weakens user control. Centralization concentrates sensitive information in one place, magnifying the impact of any app being granted access. Metadata such as package names and device identifiers reintroduce fingerprinting vectors that Android had previously restricted (e.g., [16]). Together, these design choices highlight the tension between usability and privacy, showing how the same mechanisms that make Health Connect seamless also create avenues for misuse.

Policy oversight and enforcement. Google's policies around Health Connect and health permissions are explicit: apps must treat health records as "*personal and sensitive user data*" [58], request access only for approved uses, obtain informed consent through prominent in-app disclosures [59], and maintain a comprehensive and accurate privacy policy [59, 60]. Moreover, developers are prohibited from transferring or selling Health Connect data to advertising platforms, data brokers, or for marketing purposes [87]. Our analysis, however, reveals that these requirements are not consistently met or effectively enforced. Privacy policies are also problematic, as several apps failed to describe how they use Health Connect data, and a small number even reported sharing health information with advertisers, in direct violation of Google's rules [87]. These findings align with prior research showing that privacy policies, including health and fitness apps, are frequently absent, incomplete, or misleading [38, 69, 99].

Real-world impact. Google emphasizes that Health Connect data remains "on device" [52]. Yet we find that this assurance does not hold in practice. We observed numerous cases where Health Connect records were transmitted over the network; in the majority of instances data was sent to the first party (i.e., the app's own backend servers). While this might appear benign, the critical issue is that once data leaves the device, users have no meaningful visibility or control over how it is processed, monetized, or further shared. One illustrative example is the Flo Health app, a popular period-tracking application included in our dataset. Our analysis revealed that this app read the `OvulationTest.result` and `CervicalMucusRecord.appearance` records and their respective timestamps (created and inserted into the Health Connect database by our toy app), and subsequently sent them to their backend. We also found that Flo Health app's Data Safety section mentions the collection of *Health and fitness* data, but does not state that it is shared. Interestingly, these findings align with a very recent enforcement action [92]; in September 2025, Google and Flo Health agreed to a $56 million settlement with the FTC over deceptive data-sharing practices involving reproductive health information, demonstrating the real-world impact of Health Connect's limitations and the risks we present.

Medical Records. As of January 2026, Medical Records are included in Health Connect as an experimental feature, reflecting the platform's strategic direction towards moving beyond wellness and fitness data, and becoming a broader repository of health and medical data. Health Connect's medical data records already encompass record types such as *Allergies*, *Conditions*, *Lab results*, *Medications*, *Pregnancy*, and *Vaccines* [20], which are inherently more sensitive than health and fitness data, and are typically protected under strict legislative regulations. Notably, Google currently designates the Medical Records APIs as experimental, stating that the API and corresponding Play Store policy for Medical Records access are still under development and could change, and that apps may need to meet additional requirements before they can be released on the Play Store.

The integration of medical data into Health Connect further amplifies our concerns. Under the current design, access decisions are made at the level of record types and, once granted, allow apps to read data originating from other apps. Combined with the exposure of record metadata that can be repurposed for fingerprinting, these design choices extend

the risk of inappropriate access or unintended sharing of data of much higher sensitivity. Misuse or leakage of medication histories, allergy information, or laboratory results could have tangible adverse consequences for users, including discrimination, erroneous automated inferences, or inappropriate medical decisions made outside of clinical contexts. This shift underscores the urgency of strengthening Health Connect's access control mechanisms. In particular, access models that are both context-aware and purpose-bound, that bind clinical records to their origin and declared legitimate uses, like the one that we proposed, are essential for safeguarding against misuse of such highly sensitive data.

8 Limitations

Scalability. Our work required manual effort for dynamically exercising apps. We refrained from using automated tools (e.g., monkey [24], UIHarvester [44]) since they are not optimized for targeted content-oriented exploration, and our analysis required a human-driven direction and input of diverse health-related values. Manual testing provided us with insightful results regarding apps' usage of the Health Connect platform, which remains beyond the coverage capabilities of current automation tools. Accordingly, we devoted considerable time to manually analyze all network logs, allowing us to identify major privacy violations, which would otherwise remain unidentified. While our reliance on semi-manual dynamic analysis limits coverage and scalability, the total number of apps we manually processed is substantial, and our findings highlight serious privacy risks and violations.

Furthermore, we used various transformation techniques to decode network data, which is commonly used in detecting network leaks [35, 43, 80]. Nonetheless, data can also be encrypted and our results should be interpreted as a lower bound of privacy violations. Differential analysis [36] could potentially be employed for revealing obfuscated or encrypted network data, however, the dynamic nature of Android apps introduces considerable complexity and significant challenges to this approach.

Dataset. We made our environment as realistic as possible by employing a physical device equipped with a SIM card, creating app accounts, and by processing OTP messages as needed. Moreover, we used Frida scripts to bypass anti-root or tampering techniques. However, certain apps could still not be fully analyzed due to inherent limitations in triggering their restricted functionality. Such apps are those that require a paid subscription or special identifiers (e.g., gym memberships or insurance contract identifiers), which we were unable to obtain, and also apps requiring specific external hardware/devices (e.g., smartscale). For such apps we followed standard practice, and evaluated them without creating a subscription or providing any such identifiers. As a result, our analysis is limited to the functionality accessible without such identifiers or devices. While excluding apps requiring subscriptions, payments, or external hardware may affect generalizability, such filtering is common [37, 80, 82]. Moreover, since free apps dominate the Play Store, our dataset reflects the apps most commonly used by users. Importantly, the pervasive non-compliance and violations we observe strongly suggest broader ecosystem trends rather than isolated incidents. Finally, we were unable to analyze five apps that employ advanced anti-analysis techniques.

9 Related Work

Our work presents the first in-depth investigation of the Health Connect platform, which constitutes a significant development for the Android OS, and its use in the app ecosystem. Here, we review prior relevant work on mobile health (mHealth) apps that predates Health Connect and discuss studies that are more directly relevant to ours. This prior work primarily analyzes individual mHealth applications and does not consider the Health Connect platform or its cross-app data-sharing mechanisms, which are the focus of our study.

Privacy in mHealth apps. Prior studies on mHealth apps [26, 62, 100, 110] relied on static and dynamic analysis, GUI-based user input, and Java code inspection to identify leaks and exposure to malicious domains. Hassan et al. [55] investigated female health apps and found that they request excessive permissions and collect identifiers and sensitive health data without user consent. Cory et al. [37] analyzed the network logs of 152 mHealth apps for the collection and sharing of PII and protected health information. They found that 94.7% of apps embed at least one third-party tracker, 80.4% transmit personal identifiers, 19.6% transmit fitness-related information and 9.3% transmit health information.

Minen et al. [75] and Huckvale et al. [61] found that mHealth apps share user data with third parties, including major analytics and advertising platforms such as Google and Facebook. Forsberg and Iwaya [49] analyzed ten health and fitness apps and uncovered permission over-privilege and significant interaction with third parties. Sentana et al. [97] observed that symptom-checking and condition-tracking apps requested more sensitive permissions and embedded more third-party trackers for advertising purposes. Ardalani et al. [25] observed that medical apps leak 20% more personal information than fitness apps, due to handling additional sensitive data. Aljedaani et al. [12] and Prasad et al. [89] found that apps suffer from security and privacy weaknesses, including inconsistent privacy claims, missing encryption, and third-party trackers, often linked to developers' limited security awareness, and risky third-party integrations.

Tangari et al. [102] found that 1.8% of mHealth apps contained suspicious code, 45% relied on non-encrypted communication and 23% transmitted personal data (e.g., passwords and location) over non-secure channels. Priambodo et al. [90] analyzed two mHealth apps and showed that they have multiple OWASP mobile vulnerabilities, including improper plat-

form usage, insecure data storage, and weak cryptography. Martínez-González et al. [72] introduced App-PI, a set of static analysis tools to analyze and visualize apps' privacy impact and performed a case study for the Samsung Health app. Hassan et al. [54] employed static analysis and user reviews to assess the privacy of ten mHealth apps. Salman et al. [94] discovered that mHealth apps embed third-party libraries flagged as malicious by antivirus tools, which collect and transmit user information to external servers. Díez et al. [39] identified risk of code injection attacks in the HeartKeeper app which could enable unauthorized access to health data.

Regulatory compliance. Several studies analyzed apps' data transparency and privacy policies [14, 15, 31, 38, 99] and showed that mHealth apps often have incomplete, inconsistent, or difficult-to-understand privacy policies, with notable discrepancies between stated policies and actual data practices [28, 71, 82]. Papageorgiou et al. [83] found mHealth apps failing to comply with security standards and legal data protection regulations. Fan et. al [47] presented HPDROID that detects compliance violations in mHealth apps and found that most of them are not GDPR compliant. Alfawzan et al. [10] demonstrated apps collecting data prior to user consent, while 30% did not provide privacy policies. In [104] the authors report insufficient privacy policies concerning the collection and use of health data, while Parker et al. [84] showed that apps continue to request permissions to access user data that are unrelated to their stated purpose. Mia et al. [74] evaluated HIPAA safeguards and found many apps lacked essential protections. Similar studies by Dongjing et al. [56] and Aliasgari et al. [11] also examined mHealth apps and identified problematic practices under HIPAA.

10 Conclusion

This paper presented the first empirical study of Android's Health Connect, showing that its core design choices introduce serious privacy risks, despite intending to promote interoperability and user control. Our comprehensive dynamic analysis of real-world apps uncovered a range of concerning practices, including unfettered cross-app access and exfiltration of sensitive health records. In many cases, these practices directly violated both stated privacy policies and Google's own guidelines. Taken together, our findings demonstrate that Health Connect not only fails to meet its stated goals but also erodes Android's broader privacy progress. To address these shortcomings, we proposed design refinements and stronger policy enforcement mechanisms aimed at mitigating the identified risks and ensuring that Health Connect can evolve into a trustworthy and transparent platform for managing health data. To that end, we responsibly disclosed our findings to the Health Connect team, leading to improved review processes and policy enforcement, and sanctions against bad actors.

Acknowledgments

We thank the anonymous reviewers and shepherd for their valuable feedback. This work was supported by the National Science Foundation under grants CNS-2211574 and CNS-2143363, and by the European Union under Horizon Europe grants 101168465 and 101168438, and the Digital Europe Programme grant 101158509. Any opinions, findings, conclusions, or recommendations expressed herein are those of the authors, and do not necessarily reflect those of the NSF, the European Union, or any granting authority.

Ethical Considerations

This work examines privacy risks arising from architectural design choices in Android's Health Connect through an empirical investigation of Health Connect apps' behavior. Ethical considerations were assessed using a stakeholder-based ethics analysis guided by the principles of the Menlo Report.

Methodology. This study did not involve human subjects and no real user data was accessed, collected, or processed in any way. All experiments were conducted by installing and exercising apps on our own clean device, which was dedicated exclusively to the experiments. When apps required login, we used a new Gmail account created specifically for this purpose. To populate the device's Health Connect database, we developed a toy application that provided synthetic but plausible health data. Similarly, synthetic data was also provided during app exercising whenever health-related input was required.

Stakeholders and Ethical Principles. We considered the impact on multiple stakeholders: end users of Android devices using Health Connect apps, app developers, the platform provider (Google/Android), the broader security and privacy research community, and the authors of this study. Our analysis was guided by the principles of the Menlo Report: *Beneficence*, *Respect for Persons*, *Justice*, and *Respect for Law and Public Interest*.

Potential Harms and Risks. The potential harms discussed in this section arise from the disclosure and publication of our findings, rather than from our experimental methodology, which did not involve any real users or real user data, and introduced no direct risk to end users or external systems.

The publication of this work may affect some app developers and the platform provider. While the behaviors identified are enabled by the current design of Health Connect, disclosure of our findings may result in reputational impact or regulatory scrutiny for some developers, particularly in cases where observed behavior diverges from stated disclosures or platform policies. The platform provider may also face reputational and operational impact, as our findings highlight architectural and enforcement limitations. Addressing such issues may require non-trivial design changes and additional enforcement efforts. These harms are, however, a foreseeable consequence of public-interest security and privacy research.

Furthermore, the publication of findings such as the exposure of metadata through Health Connect, could in principle be misused by adversarial or opportunistic parties to design privacy-invasive apps or tracking mechanisms. We consider this risk to be limited, as the identified issues stem from Health Connect's architecture rather than from exploitable vulnerabilities. Moreover, the potential for misuse is outweighed by the benefits of making these risks visible to users and all relevant stakeholders, and initiating discussions about mitigations.

Mitigations and Responsible Disclosure. To mitigate potential harms arising from the publication of this study, we followed a coordinated responsible disclosure process with the Android Security Team prior to submission. We disclosed our findings and recommendations to the Android Security Team on October 6, 2025, in accordance with established ethical research and responsible disclosure practices, and provided additional clarification and concrete examples on December 4, 2025. On January 29, 2026, the Health Connect team responded to our previous communication with the Android Security Team, confirming our findings and reporting that our follow-up enabled a targeted investigation into the applications and behaviors identified in our study.

Following our disclosure, Google conducted a review of the reported applications, as well as the broader claims raised in our report. While they maintained that many cross-application data flows align with Health Connect's intended design for user-authorized sharing, they confirmed multiple policy violations consistent with our findings. These included the transmission of health-related data or derived context to advertising domains, the non-consented capture of screen content by analytics SDKs, and inconsistencies between declared and actual data collection and sharing practices.

As a result, enforcement actions were initiated against a subset of non-compliant apps, several of which have already submitted corrective updates. In addition, our disclosure informed improvements to internal review and detection mechanisms targeting undisclosed data collection and SDK-driven behaviors. The Health Connect team also acknowledged the risk of covert cross-app communication via text fields and noted forthcoming API changes to mitigate this issue.

We view these outcomes as a positive and concrete impact of our work. However, we maintain that the underlying architecture of Health Connect remains fundamentally problematic. The current design relies primarily on post-hoc policy enforcement and developer compliance, while providing users with limited visibility into, or control over, how their health data is processed, shared, or monetized once it leaves the device. Given the sensitivity of the data involved, we argue that addressing these risks requires reconsideration of Health Connect's architectural assumptions, rather than reliance on enforcement mechanisms alone.

Decision to Publish. We determined that publication of this work is ethically justified and in the public interest. Our findings demonstrate that Health Connect's architectural design choices result in a systematic loss of user control and introduce significant privacy risks. These risks are amplified by the sensitivity of the data managed by Health Connect and by inconsistencies between observed behavior and stated data practices, resulting in limited user awareness and misaligned expectations. We concluded that publishing our study will engage and inform the research community, promote transparency, and encourage the adoption of more privacy-preserving mechanisms.

Open Science

In accordance with the USENIX Open Science policy, we provide the materials necessary to reproduce our findings in [7]. The repository includes the source code for our framework, our application dataset, our pre-defined list of records, the toy app used to populate the device's database with every available record, the device's Health Connect database, the raw Frida and network logs, the analysis code, a `README` file and the results of our analysis.

References

[1] Raccoon - APK downloader, 2018. `http://www.onyxbits.de/raccoon`.

[2] Wall Street Journal - Can a Facebook Post Make Your Insurance Cost More?, 2019. `https://www.wsj.com/articles/can-a-facebook-post-make-your-insurance-cost-more-11552915222`.

[3] Conroy Simber - Social Media Investigations – A Winning Strategy for Insurers, 2023. `https://www.conroysimberg.com/blog/social-media-investigations-a-winning-strategy-for-insurers/`.

[4] Burp Suite, 2025. `https://portswigger.net/burp`.

[5] Frida - A world-class dynamic instrumentation toolkit, 2025. URL: `https://frida.re/docs/home/`.

[6] VirusTotal: Analyse suspicious files, domains, IPs and URLs to detect malware and other breaches, automatically share them with the security community., 2025. `https://www.virustotal.com`.

[7] Artifacts, 2026. `doi:10.5281/zenodo.18409978`.

[8] Jagdish Prasad Achara, Gergely Acs, and Claude Castelluccia. On the Unicity of Smartphone Applications. In *Proceedings of the 14th ACM Workshop on Privacy in the Electronic Society (WPES)*, 2015.

[9] Hamad Alamri, Carsten Maple, Saad Mohamad, and Gregory Epiphaniou. Do the Right Thing: A Privacy Policy Adherence Analysis of over Two Million Apps in Apple iOS App Store. *Sensors*, 2022.

[10] Najd Alfawzan, Markus Christen, Giovanni Spitale, and Nikola Biller-Andorno. Privacy, Data Sharing, and Data Security Policies of Women's mHealth Apps: Scoping Review and Content Analysis. *JMIR mHealth and uHealth*, 2022.

[11] Mehrdad Aliasgari, Michael Black, and Nikhil Yadav. Security Vulnerabilities in Mobile Health Applications. In *2018 IEEE Conference on Application, Information and Network Security (AINS)*, 2018.

[12] Bakheet Aljedaani, Aakash Ahmad, Mansooreh Zahedi, and M. Ali Babar. An Empirical Study on Developing Secure Mobile Health Apps: The Developers' Perspective. In *2020 27th Asia-Pacific Software Engineering Conference (APSEC)*, 2020.

[13] Kevin Allix, Tegawendé F Bissyandé, Jacques Klein, and Yves Le Traon. AndroZoo: Collecting Millions of Android Apps for the Research Community. In *Proceedings of the 13th international conference on mining software repositories (MSR)*, 2016.

[14] Benjamin Andow, Samin Yaseer Mahmud, Wenyu Wang, Justin Whitaker, William Enck, Bradley Reaves, Kapil Singh, and Tao Xie. PolicyLint: Investigating Internal Privacy Policy Contradictions on Google Play. In *28th USENIX Security symposium*, 2019.

[15] Benjamin Andow, Samin Yaseer Mahmud, Justin Whitaker, William Enck, Bradley Reaves, Kapil Singh, and Serge Egelman. Actions Speak Louder than Words: Entity-Sensitive Privacy Policy and Data Flow Analysis with PoliCheck. In *29th USENIX Security Symposium*, 2020.

[16] Android Developers. Query All Packages Permission. `https://support.google.com/googleplay/android-developer/answer/10158779#zippy=%2Cinvalid-uses%2Cpermitted-uses-of-the-query-all-packages-permission`, 2021.

[17] Android Developers. Data safety information is more visible. `https://developer.android.com/about/versions/14/changes/data-safety`, 2025.

[18] Android Developers. Device. `https://developer.android.com/reference/android/health/connect/datatypes/Device`, 2025.

[19] Android Developers. Health Connect UI guidelines. `https://developer.android.com/health-and-fitness/health-connect/ui/guidelines`, 2025.

[20] Android Developers. Medical Records. `https://developer.android.com/health-and-fitness/health-connect/medical-records`, 2025.

[21] Android Developers. Parcel. `https://developer.android.com/reference/android/os/Parcel`, 2025.

[22] Android Developers. Scoped storage. `https://source.android.com/docs/core/storage/scoped`, 2025.

[23] Android Developers. Show your app's privacy policy dialog. `https://developer.android.com/health-and-fitness/health-connect/get-started#show-privacy-policy`, 2025.

[24] Android Developers. UI/Application Exerciser Monkey. `https://developer.android.com/studio/test/other-testing-tools/monkey`, 2025.

[25] Alireza Ardalani, Joseph Antonucci, and Iulian Neamtiu. Towards Precise Detection of Personal Information Leaks in Mobile Health Apps. In *16th International Conference on e-Health (MCCSIS)*, 2024.

[26] Ardalani, Alireza and Antonucci, Joseph and Neamtiu, Iulian. A Study of Personal Information Leaks in Mobile Medical, Health, and Fitness Apps. *IADIS International Journal on WWW/Internet*, 2024.

[27] Ioannis Arkalakis, Michalis Diamantaris, Serafeim Moustakas, Jason Polakis, Sotiris Ioannidis, and Panagiotis Ilia. Abandon All Hope Ye Who Enter Here: A Dynamic, Longitudinal Investigation of Android's Data Safety Section. In *33rd USENIX Security Symposium*, 2024.

[28] Bachiri M, Idri A, Fernández-Alemán JL, Toval A. Evaluating the Privacy Policies of Mobile Personal Health Records for Pregnancy Monitoring. *J. Med. Syst.*, 2018.

[29] Michael Backes, Sven Bugiel, and Erik Derr. Reliable Third-Party Library Detection in Android and Its Security Applications. In *Proceedings of the 2016 ACM SIGSAC conference on Computer and Communications Security (CCS)*, 2016.

[30] Bishnu Bhusal, Yuanye Ma, and Rohit Chadha. Privacy Nutrition Labels: Promise, Practice, and Paradoxes in Communicating Privacy. In *International Conference on Human-Computer Interaction*, 2025.

[31] Sarah R Blenner, Melanie Köllmer, Adam J Rouse, Nadia Daneshvar, Curry Williams, and Lori B Andrews. Privacy Policies of Android Diabetes Apps and Sharing

of Health Information. *American Medical Association (Jama)*, 2016.

[32] Developers Health & Fitness Dev Center. Health Connect: simplify connectivity between apps, 2025. URL: https://developer.android.com/health-and-fitness/guides/health-connect.

[33] Pew Research Center. About one-in-five americans use a smart watch or fitness tracker. https://www.pewresearch.org/short-reads/2020/01/09/about-one-in-five-americans-use-a-smart-watch-or-fitness-tracker/, 2020.

[34] K Cheah, Z Abdul Manaf, A Fitri Mat Ludin, N Razalli, N Mohd Mokhtar, and S Md Ali. Mobile Apps for Common Noncommunicable Disease Management: Systematic Search in App Stores and Evaluation Using the Mobile App Rating Scale. *JMIR Mhealth Uhealth*, 2024.

[35] Quan Chen, Panagiotis Ilia, Michalis Polychronakis, and Alexandros Kapravelos. Cookie Swap Party: Abusing First-Party Cookies for Web Tracking. In *Proceedings of the Web Conference 2021*, 2021.

[36] Andrea Continella, Yanick Fratantonio, Martina Lindorfer, Alessandro Puccetti, Ali Zand, Christopher Kruegel, and Giovanni Vigna. Obfuscation-Resilient Privacy Leak Detection for Mobile Apps Through Differential Analysis. In *Network and Distributed System Security Symposium (NDSS)*, 2017.

[37] T. Cory, W. Rieder, and T.-M. Huynh. A Qualitative Analysis Framework for mHealth Privacy Practices. In *2024 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, 2024.

[38] Hao Cui, Rahmadi Trimananda, Athina Markopoulou, and Scott Jordan. PoliGraph: Automated Privacy Policy Analysis using Knowledge Graphs. In *32nd USENIX Security Symposium*, 2023.

[39] Isabel De la Torre Díez, Bruno Olivar, Joel Rodrigues, and Miguel Lopez-Coronado. Security Analysis of a mHealth App in Android: Problems and Solutions. In *IEEE 19th International Conference on e-Health Networking, Applications and Services*, 2017.

[40] Android Developers. API reference: androidx.health.connect.client.records, 2025. URL: https://developer.android.com/reference/androidx/health/connect/client/records/package-summary.

[41] Android Developers. API reference: Health Connect Metadata, 2025. URL: https://developer.android.com/reference/kotlin/androidx/health/connect/client/records/metadata/Metadata.

[42] Android Developers. Health Connect data types, 2025. URL: https://developer.android.com/health-and-fitness/guides/health-connect/plan/data-types.

[43] Michalis Diamantaris, Serafeim Moustakas, Lichao Sun, Sotiris Ioannidis, and Jason Polakis. This Sneaky Piggy Went to the Android Ad Market: Misusing Mobile Sensors for Stealthy Data Exfiltration. In *Proceedings of the 2021 ACM SIGSAC conference on Computer and Communications Security (CCS)*, 2021.

[44] 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 (CODASPY)*, 2019.

[45] Kostas Drakonakis, Panagiotis Ilia, Sotiris Ioannidis, and Jason Polakis. Please Forget Where I Was Last Summer: The Privacy Risks of Public Location (Meta)Data. In *25th Annual Network and Distributed System Security Symposium (NDSS)*, 2019.

[46] Duckduckgo. Duckduckgo Tracker Radar. https://github.com/duckduckgo/tracker-radar, 2021.

[47] Ming Fan, Le Yu, Sen Chen, Hao Zhou, Xiapu Luo, Shuyue Li, Yang Liu, Jun Liu, and Ting Liu. An Empirical Evaluation of GDPR Compliance Violations in Android mHealth Apps. In *IEEE 31st international symposium on software reliability engineering*, 2020.

[48] Adrienne Porter Felt, Elizabeth Ha, Serge Egelman, Ariel Haney, Erika Chin, and David Wagner. Android Permissions: User Attention, Comprehension, and Behavior. In *Proceedings of the Eighth Symposium on Usable Privacy and Security (SOUPS)*, 2012.

[49] Albin Forsberg and Leonardo Horn Iwaya. Security Analysis of Top-Ranked mHealth Fitness Apps: An Empirical Study. In *Nordic Conference on Secure IT Systems (NordSec)*, 2024.

[50] Yanick Fratantonio, Chenxiong Qian, Simon P Chung, and Wenke Lee. Cloak and Dagger: From Two Permissions to Complete Control of the UI Feedback Loop. In *IEEE Symposium on Security and Privacy (SP)*, 2017.

[51] Google. What's New in Android Health. https://www.googblogs.com/whats-new-in-android-health/, 2023.

[52] Google Developers Blog. Evolving Health on Android: Migrating from Google Fit APIs to Android Health. https://android-developers.googleblog.com/2024/05/evolving-health-on-android-migrating-from-google-fit-apis-to-android-health.html, 2024.

[53] Sara Hamilton and Google Developer Relations. Leading Health and Fitness Apps Roll Out Health Connect Integrations. https://android-developers.googleblog.com/2022/11/leading-health-and-fitness-apps-roll-out-health-connect-integrations.html, 2022.

[54] Muhammad Hassan, Masooda Bashir, and Ian S. Brooks. Listening to Users: Privacy and Security in Mobile Health Apps. In *Human-Centered Design, Operation and Evaluation of Mobile Communications (HCII)*, 2025.

[55] Muhammad Hassan, Mahnoor Jameel, Tian Wang, and Masooda Bashir. Unveiling Privacy and Security Gaps in Female Health Apps. *arXiv preprint arXiv:2502.02749*, 2025.

[56] Dongjing He, Muhammad Naveed, Carl A Gunter, and Klara Nahrstedt. Security Concerns in Android mHealth Apps. In *American Medical Informatics Association (AMIA)*, 2014.

[57] Phillip C. Hegeman, Daniel T. Vader, Kristyn Kamke, and Sherine El-Toukhy. Patterns of digital health access and use among US adults: a latent class analysis. *BMC Digital Health*, 2024.

[58] Play Console Help. Personal and Sensitive User Data, 2025. URL: https://support.google.com/googleplay/android-developer/answer/10144311#personal-sensitive.

[59] Play Console Help. Privacy Policy and Prominent Disclosure Requirements, 2025. URL: https://support.google.com/googleplay/android-developer/answer/12261419?visit_id=638943337492445418-3358576433&rd=1.

[60] Play Console Help. Transparent and Accurate Notice and Control, 2025. URL: https://support.google.com/googleplay/android-developer/answer/16324062?visit_id=638943337492445418-3358576433&rd=1#ahp.

[61] Kit Huckvale, John Torous, and Mark Larsen. Assessment of the Data Sharing and Privacy Practices of Smartphone Apps for Depression and Smoking Cessation. *JAMA Network Open*, 2019.

[62] Iwaya LH, Babar MA, Rashid A, Wijayarathna C. On the privacy of mental health apps: An empirical investigation and its implications for app development. *Empirical Software Engineering*, 2023.

[63] Pankush Kalgotra, Uzma Raja, and Ramesh Sharda. Growth in the development of health and fitness mobile apps amid COVID-19 pandemic. *DIGITAL HEALTH*, 2022.

[64] Rishabh Khandelwal, Asmit Nayak, Paul Chung, and Kassem Fawaz. Unpacking Privacy Labels: A Measurement and Developer Perspective on Google's Data Safety Section. In *33rd USENIX Security Symposium*, 2024.

[65] Rajiv Leventhal. Nearly two-thirds of US consumers are mobile health app users, 2023. URL: https://www.emarketer.com/content/nearly-two-thirds-of-us-consumers-mobile-health-app-users.

[66] Li Li, Tegawende F. Bissyande, Jacques Klein, and Yves Le Traon. An Investigation into the Use of Common Libraries in Android Apps . In *IEEE 23rd International Conference on Software Analysis, Evolution, and Reengineering*, 2016.

[67] Rui Li, Wenrui Diao, Zhou Li, Jianqi Du, and Shanqing Guo. Android Custom Permissions Demystified: From Privilege Escalation to Design Shortcomings. In *2021 IEEE Symposium on security and privacy (SP)*, 2021.

[68] Bo Liu, Wanlei Zhou, Tianqing Zhu, Longxiang Gao, and Yong Xiang. Location Privacy and Its Applications: A Systematic Study. *IEEE access*, 2018.

[69] Kaijun Liu, Guoai Xu, Xiaomei Zhang, Guosheng Xu, and Zhangjie Zhao. Evaluating the Privacy Policy of Android Apps: A Privacy Policy Compliance Study for Popular Apps in China and Europe. *Scientific Programming*, 2022.

[70] Steven A Lubitz, Anthony Z Faranesh, Caitlin Selvaggi, Steven J Atlas, David D McManus, Daniel E Singer, Sherry Pagoto, Michael V McConnell, Alexandros Pantelopoulos, and Andrea S Foulkes. Detection of Atrial Fibrillation in a Large Population Using Wearable Devices: The Fitbit Heart Study. *Circulation*, 2022.

[71] Lisa Mekioussa Malki, Ina Kaleva, Dilisha Patel, Mark Warner, and Ruba Abu-Salma. Exploring Privacy Practices of Female mHealth Apps in a Post-Roe World. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems (CHI)*, 2024.

[72] M. Mercedes Martínez-González, Alejandro Pérez-Fuente, Amador Aparicio, and Pablo A. Criado-Lozano. Using the Metadata-Based App-PI Ecosystem to Assess the Privacy Impact of Health Apps. In *Proceedings of the International Conference on Ubiquitous Computing and Ambient Intelligence*, 2024.

[73] McAfee. Customer URL Ticketing System. https://sitelookup.mcafee.com/, 2025.

[74] Mia Md Raihan, Hossain Shahriar, Maria Valero, Nazmus Sakib, Bilash Saha, Abdul Barek, Md Jobair Hossain Faruk, Ben Goodman, Rumi Khan, and Sheikh Ahamed. A Comparative Study on HIPAA Technical Safeguards Assessment of Android mHealth Applications. *Smart Health*, 2022.

[75] Minen MT, Stieglitz EJ, Sciortino R, Torous J. Privacy Issues in Smartphone Applications: An Analysis of Headache/Migraine Applications. *Headache: The Journal of Head and Face Pain*, 2018.

[76] Jaron Mink, Amanda Rose Yuile, Uma Pal, Adam J Aviv, and Adam Bates. Users Can Deduce Sensitive Locations Protected by Privacy Zones on Fitness Tracking Apps. In *Proceedings Of The 2022 CHI Conference On Human Factors In Computing Systems*, 2022.

[77] Hasan Erdem Mumcu. Fitness-related on health mobile applications during COVID-19: case of Turkey. *Progress in Nutrition*, 2021.

[78] MyFitnessPal. Privacy Policy, 2025. URL: https://www.myfitnesspal.com/privacy-policy.

[79] National Heart, Lung, and Blood Institute. Study reveals wearable device trends among U.S. adults. https://www.nhlbi.nih.gov/news/2023/study-reveals-wearable-device-trends-among-us-adults%7D%7D, 2023.

[80] Trung Tin Nguyen, Michael Backes, Ninja Marnau, and Ben Stock. Share First, Ask Later (or Never?) Studying Violations of GDPR's Explicit Consent in Android Apps. In *30th USENIX Security Symposium*, 2021.

[81] Mehdi Nobakht, Yulei Sui, Aruna Seneviratne, and Wen Hu. Permission Analysis of Health and Fitness Apps in IoT Programming Frameworks. In *IEEE 17th International Conference on Trust, Security and Privacy in Computing and Communications*, 2018.

[82] Kristen O'Loughlin, Martha Neary, Elizabeth Adkins, and Stephen Schueller. Reviewing the data security and privacy policies of mobile apps for depression. *Internet Interventions*, 2018.

[83] Achilleas Papageorgiou, Michael Strigkos, Eugenia Politou, Efthymios Alepis, Agusti Solanas, and Constantinos Patsakis. Security and Privacy Analysis of Mobile Health Applications: The Alarming State of Practice. *IEEE Access*, 2018.

[84] Lisa Parker, Vanessa Halter, Tanya Karliychuk, and Quinn Grundy. How private is your mental health app data? An empirical study of mental health app privacy policies and practices. *International journal of law and psychiatry*, 2019.

[85] Anh Pham, Italo Dacosta, Eleonora Losiouk, John Stephan, Kévin Huguenin, and Jean-Pierre Hubaux. HideMyApp: Hiding the Presence of Sensitive Apps on Android. In *28th USENIX Security symposium*, 2019.

[86] Play Console Help. Health apps declaration form, 2024. URL: https://support.google.com/googleplay/android-developer/answer/14738291#zippy=%2Chealth-and-fitness.

[87] Play Console Help. Android Health Permissions: Guidance and FAQs. https://support.google.com/googleplay/android-developer/answer/12991134, 2025.

[88] Play Console Help. Policy announcement: March 5, 2025, 2025. URL: https://support.google.com/googleplay/android-developer/answer/15931464?sjid=5803033873878675532-EU.

[89] Aarathi Prasad, Matthew Clark, Ha Nguyen, Ruben Ruiz, and Emily Xiao. Analyzing Privacy Practices of Existing mHealth Apps. In *13th International Conference on Health Informatics (ICHI)*, 2020.

[90] Dimas Febriyan Priambodo, Guntur Satria Ajie, Hendy Aulia Rahman, Aldi Cahya Fajar Nugraha, Aulia Rachmawati, and Marcella Risky Avianti. Mobile Health Application Security Assesment Based on OWASP Top 10 Mobile Vulnerabilities. In *2022 International Conference on Information Technology Systems and Innovation (ICITSI)*, 2022.

[91] R. Rahman Md, E. Miller, M. Hossain, and A. Ali-Gombe. Intent-aware Permission Architecture: A Model for Rethinking Informed Consent for Android Apps. In *Proceedings of the 8th International Conference on Information Systems Security and Privacy (ICISSP)*, 2022.

[92] Reuters. Google, Flo Health to pay $56 million in period-tracking app privacy case. https://www.reuters.com/sustainability/boards-policy-regulation/google-flo-health-pay-

56-million-period-tracking-app-privacy-case-2025-09-25/, 2025.

[93] Julie M Robillard, Tanya L Feng, Arlo B Sporn, Jen-Ai Lai, Cody Lo, Monica Ta, and Roland Nadler. Availability, readability, and content of privacy policies and terms of agreements of mental health apps. *Internet interventions*, 2019.

[94] Muhammad Salman, I Wayan Budi Santana, Muhammad Ikram, and Mohamed Ali Kaafar. Auditing and Attributing Behaviours of Suspicious Android Health Applications. In *International Conference on Network and System Security (NSS)*, 2024.

[95] Jordan Samhi, Marco Alecci, Tegawendé F Bissyandé, and Jacques Klein. A Dataset of Android Libraries. *arXiv preprint arXiv:2307.12609*, 2023.

[96] Gian Luca Scoccia, Ibrahim Kanj, Ivano Malavolta, and Kaveh Razavi. Leave my Apps Alone! A Study on how Android Developers Access Installed Apps on User's Device. In *Proceedings of the IEEE/ACM 7th International Conference on Mobile Software Engineering and Systems*, 2020.

[97] I Wayan Budi Sentana, Muhammad Ikram, Dali Kaafar, and Shlomo Berkovsky. Empirical Security and Privacy Analysis of Mobile Symptom Checking Applications on Google Play. In *18th International Conference on Security and Cryptography (SECRYPT)*, 2021.

[98] Jaebaek Seo, Daehyeok Kim, Donghyun Cho, Insik Shin, and Taesoo Kim. FLEXDROID: Enforcing In-App Privilege Separation in Android. In *Network and Distributed System Security Symposium (NDSS)*, 2016.

[99] Ali Sunyaev, Tobias Dehling, Patrick L Taylor, and Kenneth D Mandl. Availability and quality of mobile health app privacy policies. *The Journal of the American Medical Association (JAMA)*, 2015.

[100] Surani, Aishwarya and Bawaked, Amani and Wheeler, Matthew and Kelsey, Braden and Roberts, Nikki and Vincent, David and Das, Sanchari. Security and Privacy of Digital Mental Health: An Analysis of Web Services and Mobile Applications. In *Data and Applications Security and Privacy XXXVII (DBSec)*, 2023.

[101] Mohammad Tahaei, Ruba Abu-Salma, and Awais Rashid. Stuck in the Permissions With You: Developer & End-User Perspectives on App Permissions & Their Privacy Ramifications. In *Conference on Human Factors in Computing Systems (CHI)*, 2023.

[102] Gioacchino Tangari, Muhammad Ikram, I Wayan Budi Sentana, Kiran Ijaz, Mohamed Ali Kaafar, and Shlomo Berkovsky. Analyzing security issues of Android mobile health and medical applications. *Journal of the American Medical Informatics Association*, 2021.

[103] Techcrunch. Google's Health Connect platform is coming to Android 14 with new features, 2023. URL: https://techcrunch.com/2023/05/10/googles-health-connect-platform-is-coming-to-android-14-with-new-features/.

[104] Xiaoyin Wang, Xue Qin, Mitra Bokaei Hosseini, Rocky Slavin, Travis D. Breaux, and Jianwei Niu. GUILeak: Tracing Privacy Policy Claims on User Input Data for Android Applications. In *Proceedings of the 40th International Conference on Software Engineering*, 2018.

[105] World Health Organization. Global Strategy on Digital Health 2020–2025. https://www.who.int/publications/i/item/9789240020924, 2021.

[106] John Wu. Magisk, 2025. URL: https://github.com/topjohnwu/Magisk.

[107] Yue Xiao, Zhengyi Li, Yue Qin, Xiaolong Bai, Jiale Guan, Xiaojing Liao, and Luyi Xing. Lalaine: Measuring and Characterizing Non-Compliance of Apple Privacy Labels. In *32nd USENIX Security Symposium*, 2023.

[108] Yue Xiao, Chaoqi Zhang, Yue Qin, Fares Fahad S Alharbi, Luyi Xing, and Xiaojing Liao. Measuring Compliance Implications of Third-party Libraries' Privacy Label Disclosure Guidelines. In *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security (CSS)*, 2024.

[109] Le Yang, Jiadong Wu, Xiaoxiao Mo, Yaqin Chen, Shanshan Huang, Linlin Zhou, Jiaqi Dai, Linna Xie, Siyu Chen, Hao Shang, Beibei Rao, Bingtao Weng, Ayiguli Abulimiti, Siying Wu, and Xiaoxu Xie. Changes in Mobile Health Apps Usage Before and After the COVID-19 Outbreak in China: Semilongitudinal Survey. *JMIR Public Health and Surveillance*, 2023.

[110] Zhao, Wanrong and Shahriar, Hossain and Clincy, Victor and Bhuiyan, Zakirul Alam. Security and Privacy Analysis of Mhealth Application: A Case Study. In *IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications*, 2020.

[111] Onur Zungur, Antonio Bianchi, Gianluca Stringhini, and Manuel Egele. APPJITSU: Investigating the Resiliency of Android Applications. In *IEEE European Symposium on Security and Privacy (EuroS&P)*, 2021.