A Data-Driven Evaluation of the Current Security State of Android Devices

Ernst Leierzopf
René Mayrhofer
Michael Roland
Johannes Kepler University Linz
Institute of Networks and Security
Linz, Austria
{fname.lname}@ins.jku.at

Wolfgang Studier
Lawrence Dean
Martin Seiffert
Fraunhofer Institute for Applied and Integrated Security AISEC
Berlin, Germany
{fname.lname}@aisec.fraunhofer.de

Florentin Putz
Lucas Becker
Technical University of Darmstadt
Secure Mobile Networking Lab
Darmstadt, Germany
{fputz, lbecker}@seemoo.de

Daniel R. Thomas
University of Strathclyde
Computer & Information Sciences
Glasgow, United Kingdom
d.thomas@strath.ac.uk

***Abstract*—Android's fast-paced development cycles and the large number of devices from different manufacturers do not allow for an easy comparison between different devices' security and privacy postures. Manufacturers each adapt and update their respective firmware images. Furthermore, images published on OEM websites do not necessarily match those installed in the field. Relevant software aspects do not remain static after initial device release, but need to be measured on real devices that receive these updates. There are various potential sources for collecting such attributes, including webscraping, crowdsourcing, and dedicated device farms. However, raw data alone is not helpful in making meaningful decisions on device security and privacy. We make a website available to access collected data. Our implementation focuses on reproducible requests and supports filtering by OEMs, devices, device models, and attributes. To improve usability, we further propose a security score grounded on the list of attributes. Based on input from Android experts, including a focus group and eight individuals, we have created a method that derives attribute weights from the importance of attributes for mitigating threats on the Android platform. We derive weights for general use cases and suggest possible examples for more specialized weights for groups of confidentiality/privacy-sensitive users and integrity-sensitive users. Since there is no one-size-fits-all setting for Android devices, our website provides the possibility to adapt all parameters of the calculated security score to individual needs.**

***Index Terms*—security analysis, security scoring, mobile security, data mining, webscraping, crowdsourcing, security, privacy**

This work has been carried out within the scope of ONCE (FFG grant FO999887054) in the program "IKT der Zukunft" and has partially been supported by Digidow (Christian Doppler Laboratory for Private Digital Authentication in the Physical World) and the LIT Secure and Correct Systems Lab. We gratefully acknowledge financial support by the Austrian Federal Ministry for Climate Action, Environment, Energy, Mobility, Innovation and Technology (BMK), the Austrian Federal Ministry of Labour and Economy (BMAW), the National Foundation for Research, Technology and Development, the Christian Doppler Research Association, 3 Banken IT GmbH, ekey biometric systems GmbH, Kepler Universitätsklinikum GmbH, NXP Semiconductors Austria GmbH & Co KG, Österreichische Staatsdruckerei GmbH, and the State of Upper Austria. This work has been co-funded by the LOEWE initiative (Hesse, Germany) within the emergenCITY center [LOEWE/1/12/519/03/05.001(0016)/72]. We thank Güliz Seray Tuncay, Mathias Payer, Ahmad-Reza Sadeghi, Alastair Beresford, Edgar Weippl, Billy Lau, William Enck, and Parjanya Vyas for helping us with their expertise in assigning likelihood and impact of threats, as well as the allocation of attributes to these threats.

I. Introduction

The ever-growing number of Android device models and device manufacturers increases the number of options for consumers ranging from low-budget to high-end flagship devices. It is a challenging task to evaluate and compare devices with regard to security and privacy. Psychology suggests that choice overload decreases the ability of consumers to properly compare and evaluate their options due to decreasing commitment and oversight [1].

Web services such as GSMArena[1] allow users to search for and compare devices from different manufacturers. While useful for assessing functionality, these platforms lack the depth needed to thoroughly evaluate the security and privacy posture of Android devices. This is particularly true for software-related features and support, including the timeliness of security updates, which require long-term observation on actual devices. According to Statista [2], 74.86 % of devices use an Android version where security updates could still be available[2] (Android ≥ 10). Even though the proportion of devices with a still supported Android version is quite high, the security patch levels of these devices may vary substantially.

[1] https://www.gsmarena.com/

[2] https://source.android.com/docs/security/bulletin/2024-08-01

DOI: 10.1109/CNS62487.2024.10735682
979-8-3503-7596-1/24/$31.00

Emerging developments in digital identity markets, such as the mobile driving license (mDL, ISO/IEC 18013-5:2021), often have regulatory requirements towards the security of end-user devices. For instance, the presence of "StrongBox" is considered a critical factor for digital identity applications and other scenarios involving on-device key storage. As a result of missing device information and market statistics, we can only guess that it is supported on less than 10 % of devices. While some statistics about the share of devices supporting StrongBox may exist internally at Google, the information is not publicly available. The impact of security attributes on the overall security of Android devices still remains largely unexplored.

This research aims to assist tech reviewers to include security assessments alongside their feature reviews and assists end users and businesses in making informed future purchase decisions. To facilitate comparisons between Android devices based on security attributes, we must distill the vast array of collected data into a straightforward security score. This simplification is essential because most people do not have the time or expertise to evaluate numerous attributes directly. To integrate these security attributes into a coherent score, we must first weight and normalize them, acknowledging that they vary in importance and come in different data types. The number of security relevant attributes of devices, user groups, use cases with different threat models, and the rapidly changing software make it difficult to create a one-size-fits-all security scoring formula.

We take inspiration from prior mobile device research by utilizing a combination of dedicated device farms and webscraping to collect a set of security attributes from different device models [3]. However, we argue that merely gathering raw data is insufficient. Users, depending on their specific use cases and knowledge levels, might find more value in explainable data that is derived from actual devices, and in concise security "scores" facilitating swift comparisons. The ability to modify such metrics based on the needs of users improves on the granularity of the security scores. Therefore, our main contributions in this work focus on:

Threat model based pre-defined attribute selections: The Android platform security model [4] already defines the assumed threat model in the context of this work, and our pre-defined "Default Security" attribute selection uses this threat model to create the best generic combination of important attributes and to weight them according to the impact and severity of the threats. To strengthen our claims, we include the variance measure from inputs of eight experts (namely mentioned in acknowledgments) to display the agreement and confidence upon the assigned values. Our basic formula for security score calculation weights every attribute and calculates the percentage of the total weight. This allows users easier comparison of devices without the need to define complicated attribute selections and weightings themselves. Depending on a user's role or identity, the impact and severity of the threats may vary and so we also define separate scores for confidentiality/privacy or integrity sensitive users, as well as supporting custom scores.

Frontend to access the collected data: A new frontend for reviewers, researchers, and end users allows to access, filter, and export collected measurements. A set of device-specific filters, including the OEM, device, and device model as well as value-based filters for selected attributes, allow users to select and compare devices in a granular way. Attribute types include booleans, ranges of numeric values, ranges of dates, and text-based values.

We have created a domain-specific query language (DSQL) to allow users to a) reproduce and share results from their selection, b) allow text-based requests to select data from the database, and c) export the generated results into common formats like CSV, XLS, and PDF. The API used in the middleware between the website frontend and the database can also be accessed publicly, which allows seamless integration into other programs for internal evaluation of devices.

Security scores and security labels are displayed to the user with lists of all used, missing, and faulty attributes, together with separate percentages of achievement for each category. The frontend provides a temporal (historical) visualization of existing measurements including security patch levels, firmware kernel versions, and security updates for every device. Figures 1 and 2 show an excerpt of our website which is publicly available at https://www.android-device-security.org/database.

Collection and verification of security update promises: Many manufacturers publish relevant security update promises, security bulletins, and other information about devices on their websites; often lacking a machine-readable format. Due to the vast quantity of existing device models, it is challenging to collect significant amounts of first-hand information about Android devices. Android apps also cannot access most of the hardware-specific information like the system-on-chip (SoC) component. Manual or automated webscraping is usually the only way to obtain such information or to complement existing information about devices.

To the best of our knowledge, we are the first to systematically collect and verify security update promises of smartphone manufacturers. At the time of writing we are collecting security update promises from Fairphone [5], Google [6], Huawei [7], Motorola [8], Nokia [9], OnePlus [10], [11], Oppo [12], Samsung [13]–[15], Vivo [16], and Xiaomi [17]. Considering these update promises pledge time periods of up to ten years, validation of update promises will only be possible in the future, but we can detect any current deviation from the manufacturers' promises. As the Android version and security patch level are the most important security attributes of Android

devices in the calculation of our security scores, the verification of vendor's update promises helps in the ranking.

II. Related Work

Within the Android ecosystem, there is a sophisticated and ever-changing structure of various security mechanisms. Mayrhofer et al. [18] discuss the development history of Android up to version 11, focusing on the mechanisms and concepts present in the Android Open Source Project (AOSP) (a later revision updated it to cover up to version 14 [4]). In their research, they systematized numerous threats to privacy and security, such as the potential danger of various memory or privilege escalation attacks, which can be mitigated using tamper resistant hardware (particularly the StrongBox Keymaster). Our attribute selection and weighting algorithm is based on this defined threat model, which includes the categories physical or proximal access, network-level threats, application code, and data and metadata processing.

A comprehensive data collection of low-level data in 33 different event categories using their Device Analyzer app was conducted by Wagner et al. [19] over a span of almost eight years (2011–2019). This study involved over 30 000 Android users worldwide, who participated in crowdsourcing. In order to evaluate the performance of device manufacturers and network operators in terms of updates and vulnerability exposure, in 2015, Thomas et al. [20] introduced the security metric called Free-Update-Mean (FUM), using the Device Analyzer data. They concluded that manufacturers are the main bottleneck in the latency of the security update processes and left the vast majority of users critically vulnerable.

The Uraniborg risk computation framework introduced by Lau et al. [21] offers a method for determining the security risk of an Android device in terms of pre-installed apps. It utilizes factors such as the quantity of apps with signature permissions, pre-granted permissions, cleartext communication of apps, and a unique risk score for particular permissions.

In their study, Ozbay and Bicakci [22] suggest a comprehensive security score for devices that is determined by evaluating pre-installed applications using various metrics. Each metric is assigned a security score based on the number of affected pre-installed applications, the level of difficulty in exploiting them, and the potential impact of exploitation. The overall score is then calculated by summing up and normalizing the individual metric scores.

The Overall Security Evaluation Score (OSES) was proposed by Khokhlov and Reznik [23] as a means to assess device security and verify the authenticity of sensor data. The metric operates under the assumption that greater user permissions correspond with an increased likelihood of sensor data tampering.

AndScanner+ is a self-contained Android firmware image crawling, extraction, and analysis tool designed by Hou et al. [24] and used on 8 325 firmware images, with Android versions ranging between 2.x and 12, from 153 vendors and 813 Android-related CVEs. In addition to the valuable historical analysis of firmware images, future work might limit the selection of firmware images to the last few years to reflect the current development process of firmware image providers (i.e. vendors and upstream suppliers). The authors observed a patch delay of 2.5 months on average in 31.4 % of all firmware images and 5.6 % of all firmware images contained unfixed vulnerabilities, even though the security patch level was updated. Up to 61.9 % of the firmware images contained potentially vulnerable pre-installed apps of which 73.9 % were sourced to the vendor, 9.7 % to AOSP, and 16.4 % to third party partners.

Jones et al. [25] investigated the process of deploying Android security updates and OS upgrades. They utilized a pseudonymized dataset sourced from HTTP access logs of a social network app. The dataset included information such as the date of the request, hashed user account identifier, user-agent string (containing OS and build version, phone model, etc.), country code (derived from IP address). Additional data scraped from Android security bulletins, carrier and manufacturer security update announcements, and device release dates from GSMArena and PhoneArena was also incorporated. An interesting aspect of this study was the examination of metadata from Android security bulletins and CVEs, as it revealed the necessary security updates that manufacturers should implement to ensure their security update commitments [26], [27].

A. Gaps Addressed by Our Research

Previous work provides a solid basis for assessing Android devices from different manufacturers by analyzing security updates and vulnerabilities [19], [20], [24]–[27], and permissions in pre-installed applications [21]–[23]. Although this research yields valuable insights, it concentrates on narrow software aspects like Android versions and app permissions, while overlooking hardware security features, which is taken into account in our data collection and evaluation framework. Since app permissions and firmware vulnerabilities frequently change with each update, and access to the research datasets is limited, these approaches do not offer an actionable way to evaluate the holistic security of specific devices.

Our work adopts a more comprehensive approach to security by creating and providing access to a longitudinal database of security-relevant attributes for specific devices, including historical data for each device. With our domain-specific-query-language we achieve repeatable and shareable results. The publicly available database includes previously unpublished security attribute measurements from real devices, such as the support of StrongBox, Android Virtualization Framework biometric authentication methods, seamless updates, as well as the current and historical Android, kernel and Keymaster/KeyMint versions of a device. Our novel security scoring approach

uses weighted attributes based on their importance and a non-numerical security label to quickly find suitable devices fulfilling all requirements. Combined with our web-interface, devices can be ranked based on the calculated security scores using custom selections of attributes and their weights.

III. Data Collection and Processing

In prior work [3] we introduced an open data collection and processing framework for Android security attributes that we build on in this paper. We gather data through measurements on real devices using device farms and crowdsourcing, as well as webscrapers for factual information like security update commitments. To improve on reproducibility and error management, every component (scanner modules, database schema, raw data evaluation, and real device measurements via firmware fingerprints) includes software versioning. Rebuilding the database from the raw data archive allows for fixes in structural errors of the data processing. Besides the existing framework, we developed a profile-owner app which programmatically tests if system or security updates are available for the device. This app helps us keep track of the available updates on all devices in the device farm by creating a cron job which sends an e-mail notification when updates are available.

A comprehensive list of collected security attributes used in this research is available online.[3]

IV. Analysis and Security Scoring

This section describes a method for selecting and weighting security attributes based on the threat model defined by Mayrhofer et al. [4], which we then use to calculate security scores for specific device models. The goal of our research is to provide a functional and easy to use framework for the security scoring of Android devices. For this purpose we defined the following requirements and desired functionality of potential users:

- Create a web-frontend for security scoring of Android devices without the need for time-consuming configuration. The web-frontend is not the main contribution of our research, but it is aimed to help the community to evaluate devices based on their security attributes. Future work might study the usability of our web-frontend for security scoring.
- Allow users to define fine-grained rules for the calculation of the security score.
- Define security labels for devices, in addition to the calculated numerical score. This allows for easier filtering of unsuitable devices based on the user's requirements.
- Users should be able to filter shown attributes and attributes used for the security score calculation based on their data types.
- Device and attribute selections, threshold definitions, and measurement sources, which can either be lab devices, real devices (lab devices and crowdsourcing devices), or all measurements including webscraping data, should be easily reproducible and shareable between users.

A. Security Score Calculation

The "Default Security" score is designed as a well-rounded score for average users, by selecting and weighting security attributes based on the Android platform security threat model [4], including *physical* and *proximal* (T.P), *network-level* (T.N), *application* (T.A) *code*, and *data* (T.D) *processing* threats. In our work we use the original threat identifier set $T = \{T.P1, \ldots, T.P4, T.N1, \ldots, T.N3, T.A1, \ldots, T.A8, T.D1, T.D2\}$. Android device models $D \subseteq A$ are each containing a subset of security attributes a_i out of all attributes in our database $A = \{a_1, a_2, \ldots, a_n\}$. Whether a device contains a security attribute depends on the defined requirements, such as being within a threshold or range of values. Non-boolean attributes are transformed into boolean values for the weight calculations, by testing against the required threshold.

Security attributes preventing/impeding exploitation or limiting the impact after successful exploitation are denoted by the function $M : A \rightarrow 2^T$. Table I shows the list of affected threats for each attribute. Due to the complexity of evaluating all possibilities, we chose to exclude exploit chaining from our threat model.

Since not all threats are equally dangerous, a risk value $R(t_j)$ is assigned to each threat $t_j \in T$, using the function $R : T \rightarrow \mathbb{R}$ defined in Equation 1:

$$R(t_j) = L(t_j) \cdot \frac{\sum_{c \in \{C,I,A,PT\}} (f_c \cdot I_c(t_j))}{4 - \#I_{noRating}} \qquad (1)$$

For the calculation of the risk associated with a threat t_j, the likelihood of occurrence $L(t_j)$, defined in percent (low = 33 %, moderate = 66 %, high = 100 %), is based on the difficulty to deploy and scale an attack, considering time, tools, knowledge, and capabilities. This likelihood value is multiplied by the averaged impact $I(t_j)$ (none = 0, low = 1, moderate = 2, high = 3) of the categories confidentiality (I_C), integrity (I_I), availability (I_A), and privacy and tracking (I_{PT}) to calculate the risk $R(t_j)$ of a specific threat t_j. Based on the specific threat model, impact categories may be valued in a different way than others using factor f_c. Factor f_c is 1 for all impact categories of the "Default Security" score. Table II shows the likelihood, impact, and the resulting risk of all threats T from the Android platform security threat model [4]. As shown in Table II, the impact on availability for T.P2–T.P4 has no rating, as it is evident that the attacker has physical access to the device, and therefore it cannot be available to the owner. Values without rating are excluded from the

[3] https://www.android-device-security.org/attributes

calculation of the averaged impact as taken into account by $\#I_{noRating}$.

Security attributes $a_i \in A$ are mapped to a weight $w(a_j)$ based on the number of addressed threats and their respective risks divided by the number of total threats, i.e., the cardinality of set T.

$$w(a_j) = \frac{\sum_{t_i \in M(a_j)} R(t_i)}{|T|} \tag{2}$$

The attribute weights are normalized to the range $[0, 100]$ and used to calculate the final security score $s(D)$ for device D, by adding up all weights of this device's security attributes and dividing them by the sum of all attribute weights, according to Equation 3:

$$s(D) = \frac{\sum_{a_j \in D} w(a_j)}{\sum_{a_j \in A} w(a_j)} \cdot 100 \tag{3}$$

In addition, we defined usability points for attributes that do not affect specific threats, but improve the total security of the device by incentivizing secure user behavior. This is currently only applied to the "A/B (seamless) system updates" attribute as this does not improve security in general but lowers the time for which the device is not available during updates[4] and thus users might install updates faster. The option of using usability points might be useful in the future for new attributes that improve usability.

B. Data Verification with Control Group

To strengthen the claimed values in Table I and Table II, we decided to include input from eight expert raters. All of the raters, as well as the members of the focus group, have similar, senior-level knowledge within the research area. Variance is used to show the agreement within one value, whereas Krippendorff's alpha α [28] is a statistical measure of inter-coder agreement or inter-rater reliability. It allows for a flexible number of raters, any number of variables with different levels of measurement [29] and incomplete (missing) data. In essence it works by weighting the rating based on the expected agreement by chance, which means that Krippendorff's alpha will stay low, if the expected agreement by chance is high. Table II includes ordinal data and an agreement within the control group of α is 30.06%. Comparing the median of the control group to the data from the expert group the agreement α is 54.46% and the percentage of agreement in which both values match is 56.10%. Table I contains nominal dichotomous data and an agreement within the control group of α is 18.08%. Comparing the median of the control group to the data from the expert group the agreement α is 43.94% and the percentage of agreement in which both values match is 72.79%. The low agreement between the experts and between the experts and the focus group might be derived from the the general definitions of the threat model. The weighting for the "Default Security" score should only propose a possible weighting, but a more specific threat model might help to improve agreement upon experts.

[4]A/B (seamless) system updates: https://source.android.com/docs/core/ota/ab.

C. Customizing the Score for Sensitive Groups of Users

The selection and weighting of attributes is fully configurable by the end user. To demonstrate and explain possible custom security score attribute selections and weights based on the threat models, we chose two different user groups of confidentiality/privacy-sensitive users (e.g., journalists and politically persecuted activists) and integrity-sensitive users (e.g., CEOs and government officials). Threats impacting the privacy and confidentiality are likely more dangerous for journalists than for other people and threats impacting the integrity of a device and its data are probably more dangerous for people with significant decision-making power like CEOs. For accurate definitions of impacts for specific groups such as journalists and CEOs, it is necessary to design interviews and capture relevant insights to the actual threats of these groups. We use these user groups only to illustrate possible use cases of custom security scores.

In comparison to our "Default Security" pre-defined score, the custom security scores focus on very specific threats in contrast to the broad threat model evaluation. These threats are mapped into the security score by multiplying the affected impact categories with a group-specific factor. As the impact of specific categories depends on the user group, we demonstratively defined the factors for the impact of privacy and tracking and the impact of confidentiality on our example group of journalists as 1.5 and 1.3 respectively, and a factor of 1.5 for the integrity impact on the CEOs example. These multiplication factors for impact categories are not based on any qualitative data analysis and they serve only to illustrate the concept of security scoring of different user groups. Future work might collect and evaluate requirements of various user groups through well-designed surveys and interviews.

Journalists and politically persecuted activists are often targets of attacks in the digital space, especially in politically unstable countries. Therefore, these people are often at the risk of *physical access* and *network-level* threats. In some cases, journalists are at risk with *data processing* when receiving and unintentionally running exploit code disguised as confidential documents.

CEOs and government officials often have access to either classified or sensitive data and services. They specifically may become target of criminal groups aiming to steal money/intellectual property or demand blackmail/ransom rather than nation state actors. Therefore, *physical access* threats of CEOs are not as impactful compared to journalists. Despite that, CEOs are still subject to unlocked device control of an authorized user. As potential mitigation, work profiles are one example that may prevent this kind of abuse. Malicious authorized users may also

TABLE I

ALLOCATION OF THE COLLECTED ATTRIBUTES TO THREATS. ATTRIBUTES ARE RELEVANT FOR A THREAT, IF THEY EITHER PREVENT/IMPEDE EXPLOITATION OR LIMIT THE IMPACT AFTER SUCCESSFUL EXPLOITATION. THE CALCULATED AND NORMALIZED WEIGHTS ARE ALSO INCLUDED. DUE TO SPACE CONSTRAINTS, WE OMITTED THE 'T.' FROM THE ORIGINAL THREAT IDENTIFIERS DEFINED BY MAYRHOFER ET AL. [4]. THE BACKGROUND COLOR SIGNALS THE CONFIDENCE AND AGREEMENT IN THE INPUT DATA $\sigma^2 = [0, 1]$ FROM THE FOCUS GROUP AND INDIVIDUAL EXPERTS, WITH WHITE MEANING HIGH CONFIDENCE ($\sigma^2 \leq 0.7$), YELLOW MEANING MODERATE CONFIDENCE ($0.7 \leq \sigma^2 \leq 0.85$), AND GREY MEANING LOW CONFIDENCE ($0.85 < \sigma^2 \leq 1$). VALUES OF THE FOCUS GROUP MATCHING WITH THE MEDIAN OF THE CONTROL GROUP ARE PRINTED BOLD.

Attribute	**P1**	**P2**	**P3**	**P4**	**N1**	**N2**	**N3**	**A1**	**A2**	**A3**	**A4**	**A5**	**A6**	**A7**	**A8**	**D1**	**D2**	**Default Weight**	**Journalist Weight**	**CEO Weight**	**Usability Points**
Android Version	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓	100	100	100	
Security Patch Level	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓	100	100	100	
Fingerprint Authentication	✗	✗	✓	✓	✗	✗	✗	✗	✗	✗	✓	✓	✗	✗	✗	✗	✓	41	42	42	
Face Authentication	✗	✗	✓	✓	✗	✗	✗	✗	✗	✗	✓	✓	✗	✗	✗	✗	✓	41	42	42	
Iris Authentication	✗	✗	✓	✓	✗	✗	✗	✗	✗	✗	✓	✓	✗	✗	✗	✗	✓	41	42	42	
Keystore	✗	✗	✗	✓	✗	✗	✗	✓	✓	✓	✗	✗	✗	✓	✗	✗	✓	54	53	54	
Embedded Secure Element	✗	✓	✓	✓	✗	✗	✗	✓	✓	✓	✗	✗	✗	✓	✗	✓	✓	65	66	66	
Embedded SIM	✗	✓	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗	✗	✓	✗	✓	✓	60	60	60	
StrongBox	✗	✓	✓	✓	✗	✗	✗	✓	✓	✓	✗	✗	✗	✓	✓	✓	✓	70	70	71	
Trusted Exec. Environment	✗	✓	✓	✓	✗	✗	✗	✓	✓	✓	✗	✗	✗	✓	✗	✓	✓	65	66	66	
A/B System Updates	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	0	0	0	20
Android Virt. Framework	✗	✗	✗	✗	✗	✗	✗	✓	✓	✓	✓	✗	✗	✓	✗	✗	✓	49	48	49	
Identity Credentials	✗	✓	✗	✓	✗	✗	✗	✓	✓	✓	✓	✗	✓	✓	✗	✓	✓	74	73	74	
Multiple User Support	✗	✗	✗	✓	✗	✗	✗	✓	✓	✓	✓	✓	✗	✓	✗	✗	✓	69	69	69	
Protected Confirmation	✗	✓	✗	✓	✗	✗	✗	✓	✓	✓	✓	✓	✓	✓	✗	✗	✓	75	75	76	
Enc. Shared Preferences	✗	✗	✗	✓	✗	✗	✗	✓	✓	✓	✓	✗	✗	✓	✗	✗	✓	64	63	65	

TABLE II

ASSIGNMENT OF LIKELIHOOD OF OCCURRENCE, THE IMPACT DIVIDED INTO THE CATEGORIES CONFIDENTIALITY, INTEGRITY, AVAILABILITY, AND PRIVACY AND TRACKING, AND THE RESULTING RISK FOR THE PRE-DEFINED "DEFAULT", "JOURNALIST", AND "CEO" SCORES. THE ORIGINAL THREAT IDENTIFIERS DEFINED BY MAYRHOFER ET AL. [4] ARE USED. THE VALUES IN THE BRACKETS SHOW THE VARIANCE σ^2 IN THE INPUT DATA FROM THE FOCUS GROUP AND INDIVIDUAL EXPERTS. LOW VALUES SIGNAL CONFIDENCE AND AGREEMENT IN THE DATA. THE BACKGROUND COLOR SIGNALS THE CONFIDENCE AND AGREEMENT IN THE INPUT DATA $\sigma^2 = [0, 1]$ FROM THE FOCUS GROUP AND INDIVIDUAL EXPERTS, WITH WHITE MEANING HIGH CONFIDENCE ($\sigma^2 \leq 0.7$), YELLOW MEANING MODERATE CONFIDENCE ($0.7 \leq \sigma^2 \leq 0.85$), AND GREY MEANING LOW CONFIDENCE ($0.85 < \sigma^2$). VALUES OF THE FOCUS GROUP MATCHING WITH THE MEDIAN OF THE CONTROL GROUP ARE PRINTED BOLD.

Threat (t_j)	**Likelihood (L)**	**Impact (I_C) Confidentiality**	**Impact (I_I) Integrity**	**Impact (I_A) Availability**	**Impact (I_{PT}) Privacy/Tracking**	**Risk (R) Default**	**Risk (R) Journalist**	**Risk (R) CEO**
T.P1	**Mod. (0.44)**	Low (0.62)	**Low (0.67)**	Low (0.99)	Low (0.47)	66	79	74
T.P2	**Low (0.44)**	Low (0.99)	Low (0.99)	**No Rating**	**High (0.69)**	55	75	61
T.P3	**Mod. (0.4)**	**Low (0.47)**	**Low (0.67)**	**No Rating**	High (0.69)	110	150	121
T.P4	**High (0.47)**	**High (0)**	**High (0.44)**	**No Rating**	**High (0)**	300	380	350
T.N1	**High (0.1)**	**Low (0.47)**	None (0.32)	None (0.22)	Low (0.54)	50	70	50
T.N2	**Mod. (0.25)**	**High (0.84)**	High (0.77)	High (0.77)	**High (0.47)**	198	238	223
T.N3	Mod. (0.22)	**High (0.84)**	High (0.77)	**High (0.44)**	**High (0)**	198	238	223
T.A1	**High (0.47)**	Low (0.99)	**Low (0.91)**	**Low (0.77)**	**High (1.06)**	150	195	163
T.A2	**Mod. (0.36)**	Low (0.73)	**Low (0.44)**	**Low (0.36)**	**Moderate (0.69)**	83	104	91
T.A3	**High (0.22)**	Low (0.67)	**Low (0.62)**	**Low (0.32)**	**Moderate (0.84)**	125	158	138
T.A4	High (0.75)	Moderate (0.23)	Low (0.69)	Low (0.69)	**High (0.5)**	200	253	225
T.A5	High (0.73)	Moderate (0.48)	None (0.36)	None (0.36)	High (1.11)	99	134	107
T.A6	Mod. (0.23)	High (0.48)	High (0.69)	High (0.61)	**Low (0.36)**	74	86	83
T.A7	High (0.84)	**High (0.1)**	**High (0.1)**	**High (0.69)**	**High (0.17)**	300	360	338
T.A8	**Low (0.1)**	**High (0.69)**	**High (0.44)**	High (0.89)	**High (1.11)**	99	119	111
T.D1	Low (0.86)	High (1.19)	Mod. (0.36)	**Low (0.36)**	**High (0)**	66	86	70
T.D2	Mod. (0.23)	**Mod. (0.75)**	**High (0.48)**	**Mod. (0.5)**	Low (0.69)	116	134	132

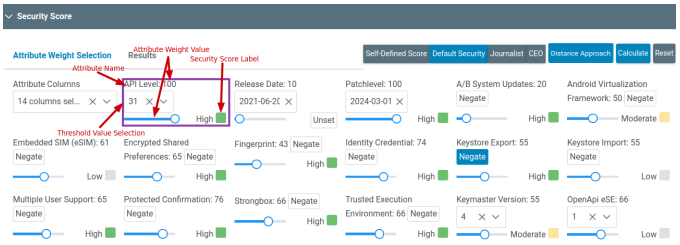
install spyware, encrypting trojans, and other unwanted software on devices and the smartphone owner is therefore also vulnerable to *application code* threats—particularly with the elevated access available to CEO and comparable roles. CEOs of companies are assumed to be in a controlled secure network environment, e.g., through an always-on VPN, which mitigates most of the *network-level* threats. Targeted unique identifier abuses, which are defined in T.D1, are also a common threat to CEOs. Impact of threats defined in Table II affect the weights of attributes for different groups in Table I, because of the factors used for the various user groups and impact categories.

D. Security Label and Default Security Score

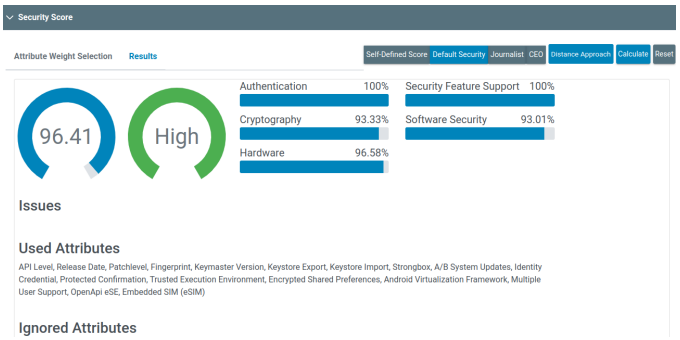
In addition to numerical security scoring we propose a non-numeric scoring label that helps users quickly make decisions about the security of a device. These labels are based on the categorization of selected security attributes as high, moderate, low, or unset, and result in displayed security labels for the device model of high, moderate, low, or insecure.Thresholds are used to determine whether a security attribute requirement is fulfilled. Afterwards, the maximum value of the unfulfilled attribute requirements is subtracted from the maximum security label (High (3)) to give the resulting security label. For the resulting security label to be High (3), all selected security attribute thresholds must be met or have unset importance (0). If the maximum importance of unmet security attributes is Low (1), then this results in a label of Moderate (2), a maximum importance of Moderate (2) results in a label of Low (1), and a maximum of High (3) leads to a label of Insecure (0). Figure 1 illustrates the "Default Security" attribute selection and the resulting security score diagram of the Google Pixel 7 with the devices in the table ranked by the security score respectively. The "Attribute Weight Selection" tab allows users to select the desired attributes used for the security score calculation. Next to the pre-defined security score attribute selections "Default Security", "Journalist" (confidentiality/privacy-sensitive), and "CEO" (integrity sensitive), there is also an option for users to create their own attribute selections and weights. For every attribute the desired weight, security score label, and possible operations defined in Table III can be applied.

E. DSQL and Data Type Definitions

We developed a domain-specific query language (DSQL) to access a defined view on the collected data. The frontend website encodes its state and selections within the URL which allows for reproducible queries and pre-defined settings. Direct access via the documented API requests introduce the possibility to integrate the data in other applications to be used for security checks and decision-making. Based on the data type of attributes, different operations can be applied with the filtering and attribute selection for the security score calculation.



(a) Interface for defining the "Default Security" score in the frontend. It shows the name, threshold value, security score level, and weighting for each attribute.



(b) Interface displaying the resulting security score (left) and security label (mid) along with the security scores across the five different categories (right). It also lists the attributes used in the calculation of the score, and those ignored.

Fig. 1. Proposed security score frontend interface

TABLE III
ATTRIBUTE DATA TYPES AND EVALUATIONS.

Data Type	Data Filters	Security Score Operations
Boolean	True, False, N/A	Negate
Numeric	Range	Threshold, Distance
Date	Range	Threshold, Distance
Text	Set Selection	-

Table III shows the existing data types and possible operations for data selection and security scoring. By using these security score operations, we convert all data into boolean values, which enables us to use a simple security scoring formula to summarize the weights of all true (or false if negated) values and divide them by the total weight number.

Boolean values can be negated to indicate that an attribute or experiment must not be found, like for example the ability to export keys from the keystore. Numeric and date values allow the definition of thresholds (threshold approach). The second, more sophisticated, approach (distance approach) also uses thresholds, but the calculation includes the distance to the best possible value in the scoring. This approach enables better scoring and ranking within a list of suitable devices. In some cases, like with the security patch level, the risk might increase non-linearly with larger distances, as devices with a security patch level older than one year are much more vulnerable than devices with a three-month-old security patch level.

For example, the URL *https://www.android-device-security.org/database/?realMeasurementsOnly=true&securityScoreLabel-API%20Level=High&show=Face&minThreshold-api_level=32&selectedDeviceModel=7* filters for only real device measurements with the face authentication attribute. The very simple security scoring is only based on the check if devices have an API level greater than or equal to 32. The security score label is set to high, which means that all devices with an API level smaller than 32 are labeled as insecure (section IV-D).

F. Research Methods

Tables I and II are defined by a group of academic Android experts from different universities in multiple countries. Because the threat model was (intentionally) defined in a general unspecific way, there is no objective and exact definition of the values in these tables. To verify our assignments, we let a control group of eight Android experts define these tables individually. Based on the separate inputs from Android experts, including a focus group and eight individuals, we calculate the variance σ^2 of each field in the data and the Krippendorff alpha over the whole dataset. Further details of the data verification are described in Section IV-B. Using this methodology, we are able to distinct between results that are agreed upon and uncertain ones. Finding the best-fitting values is a difficult task and using inputs from a control group does not necessarily improve the resulting assignments, but it shows which data is agreed upon and thus trustworthy.

Before adopting a basic risk formula for calculating attribute weights, we considered using the Attack Potential method defined in ISO/IEC 18045:2022 [30]. This method requires defining attack potential values for each threat across categories such as elapsed time, expertise, target knowledge, window of opportunity, and the required equipment. However, since most of these values depend more on expert opinions than on hard facts, using this method could introduce more error risks compared to the basic risk formula we ultimately chose.

Due to missing data, it is possible that not all devices have the same set of attribute measurements. Missing attributes are excluded from the security score calculation and listed in the "Result" tab (Figure 1), in addition to the list of used attributes and the list of issues (attribute does not match the expected value or threshold). Because of potentially missing attributes, the user still needs to interpret the security score and check if enough information is used.

V. Historical Data Analysis

Many aspects of software, like for example the security update time and frequency, can only be measured over time. Therefore, our data collection and evaluation framework is designed to store and present historical data of Android devices, such as the security patch level, kernel version, and patch delay at a specific time.

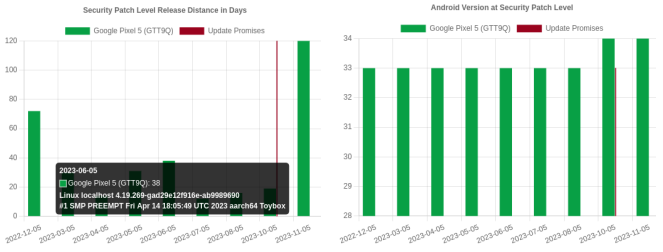
As of the release of the Pixel 8 devices, Google is promising security and Android version updates for at least seven years [6]. By using the Qualcomm QCM6490 extended life chipset, Fairphone is able to promise five Android version updates and at least eight (up to ten) years of security updates [5]. Other manufacturers either provide security update promises and update frequencies per device (Google [6], Motorola [8], Nokia [9], OnePlus [10], [11], Samsung [13]–[15]) or only provide the current security update frequencies (Huawei [7], Oppo [12], Vivo [16]). It is notable that Xiaomi is only providing a list of devices which are no longer supported [17].

The Android Device Security Database[5] also provides an interface to display time diagrams for the promised and measured security update frequency of specific devices. The update frequency is calculated by grouping the first-seen time distance of patches into the proper groups of 30, 60, 90, or 120 days, and calculating the average distance in days for each of these groups. These groups should represent the lifecycle of Android devices in which security updates are usually published more frequently in the beginning. Even though this information is useful, it must be verified by the user due to the possibility of missing measurements. Figure 2 shows the security patch level distance to the most current security patch level at the time and the API level of the Google Pixel 5. The tooltips show the build date and the kernel version of the firmware. As the first data point of the security patch level distance indicates, the distances are based on real measurements and as the device was not in an updated state at the beginning, the first distance is greater than in the following months.

VI. Conclusion & Future Work

Our work introduces the first platform to holistically compare and evaluate Android devices based on their security attributes. The website provides pre-defined security attributes and weighting for general use cases and the possibility to define a repeatable, fine-grained scoring for individual use cases. In order to demonstrate examples of specific weight priorities for user groups with different

[5]https://www.android-device-security.org/database



(a) Security patch level release distance in days (b) Android version release distance in days

Fig. 2. Historical data representation

threat models, we use journalists and CEOs as representatives of known differing threat models. The weights are calculated using the threat model for Android devices as well as expert knowledge. Using a control group of eight Android experts, we are able to show the confidence of our assignments and we are able to reduce the subjectiveness as much as possible. Due to the general definitions of the threat model, the agreement between each rater and between the focus group and group of raters is relatively low. This result is expected and the "Default Security" score is only meant to be used for quick comparisons between devices. Users are encouraged to select and weight the importance of the attributes based on their own specific threat model. Our prior belief in the importance of the Android version and security patch level was corroborated by our threat model based attribute weighting.

Future work will also include data from crowdsourcing which will improve accuracy of the security update release dates and mitigate the errors due to updates being distributed in batches for different regions.

The current implementation still leaves room for enhancing functionality. At present, it is not possible to analyze detailed views of firmware images, specifically differentiating the default packages installed. Future implementations may introduce the analysis of installed SELinux policies and raw data collected from live devices in the field via crowdsourcing.

References

[1] B. Schwartz, *The Paradox of Choice: Why More Is Less*, ser. Harper Perennial. HarperCollins, 2003.

[2] Statista GmbH, "Anteile der verschiedenen Android-Versionen an der Internetnutzung von Geräten mit Android OS weltweit im November 2023," 2023. [Online]. Available: https://de.statista.com/statistik/daten/studie/180113/umfrage/anteil-der-verschiedenen-android-versionen-auf-geraeten-mit-android-os/

[3] E. Leierzopf, M. Roland, F. Putz, and R. Mayrhofer, "A Large-Scale Data Collection and Evaluation Framework for Android Device Security Attributes," in *IDIMT-2023: New Challenges for ICT and Management*, ser. Schriftenreihe Informatik, vol. 52. Trauner Verlag, SEP 2023, pp. 63–72.

[4] R. Mayrhofer, J. Vander Stoep, C. Brubaker, D. Hackborn, B. Bonné, G. S. Tuncay, R. P. Jover, and M. A. Specter, "The Android Platform Security Model (2023)," Jan. 2024. [Online]. Available: https://arxiv.org/abs/1904.05572

[5] Fairphone, "Fairphone OS," Fairphone Support, 2024. [Online]. Available: https://support.fairphone.com/hc/en-us/articles/9979180437393-Fairphone-OS

[6] Google, "Learn when you'll get software updates on Google Pixel phones," Pixel Phone Help, 2024. [Online]. Available: https://support.google.com/pixelphone/answer/4457705

[7] Huawei Consumer Support, "Huawei EMUI/Magic UI security updates," Security Updates, 2024. [Online]. Available: https://consumer.huawei.com/en/support/bulletin

[8] Motorola, "Security Updates," Motorola Support, 2024. [Online]. Available: https://en-us.support.motorola.com/app/software-security-update

[9] HMD Global, "Nokia Smartphone Security Maintenance Release Summary," 2024. [Online]. Available: https://www.nokia.com/phones/en_int/security-updates

[10] OnePlus, "Security Advisories," AER Updates, 2024. [Online]. Available: https://security.oneplus.com/en/mend

[11] Gary C., "Announcing our new expanded software commitment," OnePlus Community, NOV 2022. [Online]. Available: https://community.oneplus.com/thread/1211291251581124608

[12] OPPO Security Response Center, "Security Advisories," 2024. [Online]. Available: https://security.oppo.com/en/mend

[13] Samsung Mobile Security, "Announcing up to five (5) years support for Samsung Security Updates on select Galaxy devices," Security Post, 2022. [Online]. Available: https://security.samsungmobile.com/securityPost.smsb

[14] ——, "Samsung Sets the New Standard With Four Generations of OS Upgrades To Ensure the Most Up-to-Date and More Secure Galaxy Experience," 2022. [Online]. Available: https://news.samsung.com/global/samsung-sets-the-new-standard-with-four-generations-of-os-upgrades-to-ensure-the-most-up-to-date-and-more-secure-galaxy-experience

[15] ——, "Introduction to Samsung Security Updates," Security Updates, 2024. [Online]. Available: https://security.samsungmobile.com/workScope.smsb

[16] Vivo, "vivo Android™ Security Updates," 2024. [Online]. Available: https://www.vivo.com/en/security

[17] Xiaomi, "Security updates for smartphones," Support Policy, 2024. [Online]. Available: https://trust.mi.com/misrc/updates/phone?tab=policy

[18] R. Mayrhofer, J. Vander Stoep, C. Brubaker, and N. Kralevich, "The Android Platform Security Model," *ACM Trans. Priv. Secur.*, vol. 24, no. 3, Aug. 2021.

[19] D. T. Wagner, A. Rice, and A. R. Beresford, "Device Analyzer: Large-Scale Mobile Data Collection," *ACM SIGMETRICS Performance Evaluation Review*, vol. 41, pp. 53–56, Mar. 2014.

[20] D. R. Thomas, A. R. Beresford, and A. Rice, "Security Metrics for the Android Ecosystem," in *Proceedings of the 5th Annual ACM CCS Workshop on Security and Privacy in Smartphones and Mobile Devices (SPSM '15)*, ser. SPSM '15. New York, NY, USA: ACM, 2015, pp. 87–98.

[21] B. Lau, J. Zhang, A. R. Beresford, D. R. Thomas, and R. Mayrhofer, "Uraniborg's Device Preloaded App Risks Scoring Metrics," Aug. 2020. [Online]. Available: https://www.android-device-security.org/publications/2020-lau-uraniborg/Lau_2020_Uraniborg_Scoring_Whitepaper_20200827.pdf

[22] A. Ozbay and K. Bicakci, "Should Users Trust Their Android Devices? A Scoring System for Assessing Security and Privacy Risks of Pre-Installed Applications," 2023.

[23] I. Khokhlov and L. Reznik, "Data Security Evaluation for Mobile Android Devices," in *2017 20th Conference of Open Innovations Association (FRUCT)*. IEEE, 2017, pp. 154–160.

[24] Q. Hou, W. Diao, Y. Wang, C. Mao, L. Ying, S. Liu, X. Liu, Y. Li, S. Guo, M. Nie, and H. Duan, "Can we trust the phone vendors? comprehensive security measurements on the android firmware ecosystem," *IEEE Transactions on Software Engineering*, vol. 49, no. 7, pp. 3901–3921, 2023.

[25] K. R. Jones, T.-F. Yen, S. C. Sundaramurthy, and A. G. Bardas, "Deploying Android Security Updates: An Extensive Study Involving Manufacturers, Carriers, and End Users," in *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security (CCS '20)*. New York, NY, USA: ACM, 2020, pp. 551–567.

[26] S. Farhang, M. B. Kirdan, A. Laszka, and J. Grossklags, "An Empirical Study of Android Security Bulletins in Different Vendors," in *Proceedings of The Web Conference 2020 (WWW '20)*. New York, NY, USA: ACM, 2020, pp. 3063–3069.

[27] D. Wu, D. Gao, E. K. T. Cheng, Y. Cao, J. Jiang, and R. H. Deng, "Towards Understanding Android System Vulnerabilities: Techniques and Insights," in *Proceedings of the 2019 ACM Asia Conference on Computer and Communications Security (Asia CCS '19)*. New York, NY, USA: ACM, 2019, pp. 295–306.

[28] K. Krippendorff, "Estimating the reliability, systematic error and random error of interval data," *Educational and Psychological Measurement*, vol. 30, no. 1, pp. 61–70, 1970. [Online]. Available: https://doi.org/10.1177/001316447003000105

[29] S. S. Stevens, "On the theory of scales of measurement," *Science*, vol. 103, pp. 677–680, 1946. [Online]. Available: http://www.jstor.org/stable/1671815

[30] ISO/IEC 18045:2022(E), "Information security, cybersecurity and privacy protection — Evaluation criteria for IT security — Methodology for IT security evaluation," International Organization for Standardization, Geneva, CH, Standard, Aug. 2022.