Android Mobile Terminal Security Assessment Based on Analytical Hierarchy Process (AHP)

Zhiyuan Hu, Linghang Shi, Huijun Chen, and Jinghui Lu(✉)

vivo Mobile Communication Co., Ltd., Dongguan, China
{huzhiyuan,shilinghang,hj.chen,john.lu}@vivo.com

Abstract. Mobile terminals especially smartphones are changing people's work and life style. For example, mobile payments are experiencing rapid growth as consumers use mobile terminals as part of modern dynamic lifestyles. However, mobile terminal security is a big challenge for mobile application services. In order to mitigate security risks, the mobile terminal security assessment should be performed before accessing the services with high security requirements. A comprehensive approach for security assessment is proposed in this paper by defining security metrics with scores, determining the relative weights of these security metrics based on the analytical hierarchy process (AHP), designing a general system architecture and implementing security assessment system. Until 30 September 2022, around 6 million smartphones support the security assessment solution proposed in this paper. In September 2022, these smartphones had made 77 million online payment transactions, of which 60,000 fraudulent transactions were detected, with the payment fraud rate of about 0.08%.

Keywords: Security Assessment · Security Metric · Relative Weight · Analytical Hierarchy Process (AHP)

1 Introduction

Today, mobile terminals especially smartphones are changing people's work and life style. For example, mobile payments are experiencing rapid growth as consumers use mobile terminals as part of modern dynamic lifestyles; more and more enterprises support employees accessing enterprise services with personal mobile terminals; etc. However, mobile terminal security is a big challenge for these mobile application services. For example, users may unknowingly expose their terminals to malware and put the sensitive data at risk of data breaches. In order to ensure mobile applications and sensitive data's security, it is very important for mobile application services to check if the mobile terminals are secure enough to access the services.

Some solutions were designed for the server to perform mobile terminal security assessment. Wang [1] proposed that the edge server conducts security assessment for mobile intelligent terminals before they access the IoT network. Xi [2] introduced machine learning to evaluate the security of power mobile terminal system. Wu [3]

© ICST Institute for Computer Sciences, Social Informatics and Telecommunications Engineering 2023
Published by Springer Nature Switzerland AG 2023. All Rights Reserved
S. Goel et al. (Eds.): ICDF2C 2022, LNICST 508, pp. 453–471, 2023.
https://doi.org/10.1007/978-3-031-36574-4_27

defined specific evaluation indexes and provided a security evaluation system of smart grid terminals. Ratchford [4] gave a model to conduct BYOD (Bring Your Own Device) security posture assessment with using Euclidian's algorithm to compare a posture among two organizations. Visoottiviseth [5] provided a solution to conduct security assessment for IoT devices by executing penetration testing on targeted IoT devices. Othman [6] discussed the feasibility of information system audit in assessing mobile device security by exploring the risks and vulnerabilities of mobile devices. As for above server-based solutions, the server collects security data from the mobile terminals and conducts security assessment. But the collected security data may include user personal data, which may generate risks to individuals' privacy.

Some solutions were designed for the mobile terminal to conduct security assessment by itself. Cavalcanti [7] and Vecchiato [8] proposed the method to identify and report potentially unsafe settings of Android terminals. Irwan [9] analyzed the latent security vulnerabilities in the applications of the terminals. Khokhlov [10], Raza [11] and Hayran [12] described security assessment for operating system. Khokhlov [13] defined security metrics and provided data security evaluation for mobile devices. Zendehdel [14] proposed a semi-automated framework that can be used to discover both known and unknown vulnerabilities in wearable health monitoring devices. All above terminal-based solutions provided security assessment for the application, operating system and data of mobile terminals, without considering hardware security or communication security of mobile terminals, which is important for some application services. For example, secure element to store sensitive data and confidentiality protection for data in transiting over the networks are important for mobile enterprise services.

So, it is necessary to design a terminal-based comprehensive solution for conducting mobile terminal security assessment, with taking application security, operating system security, hardware security and communication security into account. In this way, mobile application services can determine whether the mobile terminal is secure enough to access the services. Moreover, raw security data for security assessment will neither leave the mobile terminals nor be sent to the mobile application services. Therefore, the risks to individuals' privacy could be reduced and even avoided.

The main contributions of this paper are to define security metrics with scores, to determine relative weights of security metrics based on AHP, to design a general system architecture and to implement security assessment system for Android mobile terminal.

The rest of the paper is organized as follows. In Sect. 2, security metrics with scores are defined. In Sect. 3, Analytical Hierarchy Process (AHP) is introduced to determine the relative weights of security metrics. A general system architecture for Android mobile terminal security assessment is designed in Sect. 4. In Sect. 5, security assessment system is implemented and security assessment is conducted for mobile payment services. Section 6 concludes the paper.

2 Security Metrics and Scoring for Android Mobile Terminals

To reduce security risks, the application service will perform security risk assessment with considering the security assessment of the terminal, network and server before providing the services. This paper only studies the mobile terminal security assessment, which is an important part of security risk assessment for the application services.

To support mobile terminal security assessment, security metrics to be measured should be defined. Based on the technologies GlobalPlatform TEE (Trusted Execution Environment) [15] and ARM TrustZone [16], most Android terminals (e.g., smartphones and tablets) maintain two worlds for all trusted and untrusted applications in REE (Rich Execution Environment) and TEE separately, which is shown in Fig. 1. So, the security metrics related to the mobile terminal could be classified into four categories, i.e., REE security metrics, TEE security metrics, hardware security metrics and communication security metrics. Each security metric category includes a set of sub-metrics, which will be described in the next sections.

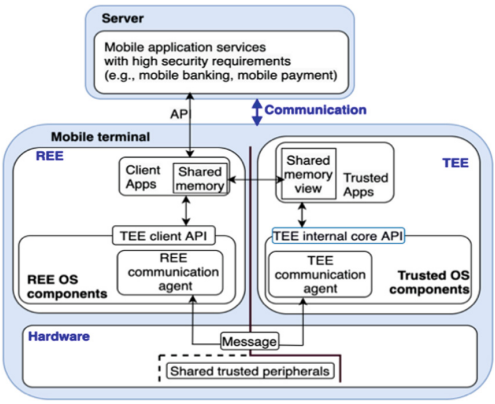

Fig. 1. Framework of Android-based mobile terminal.

2.1 REE Security Metrics and Scoring

REE security metrics mainly include ten sub-metrics which are defined as follows. The scores of each sub-metric are also given.

Mobile application spoofing attack detection in REE (A1) is to identify the attacks in which a malicious mobile application mimics the visual appearance and responses of the genuine one in REE of the terminal. Assigned score could be: 10 - spoofing attack is never detected, 0 - spoofing attack(s) was detected before, but not detected when accessing the services, −10 - spoofing attack(s) is detected when accessing the services.

Virus detection in REE (A2) is to identify the viruses, worms and other types of malwares in REE of the mobile terminal. Assigned score could be: 10 - virus is never detected, 0 - virus(es) was detected before, but not detected when accessing the services, −10 - virus(es) is detected when accessing the services.

Spam protection (A3) is to protect the mobile terminal from spam calls and messages by automatically intercepting suspicious calls and messages. Assigned score could be: 10 - if spam protection is supported, 0 - otherwise.

Malicious activity detection (A4) is to identify the malicious behaviors of the applications, such as inducing internet gambling, bypassing security checks, creating a fraudulent online payment, etc. Assigned score could be: 10 - malicious activity is never

detected, 0 - malicious activity was detected before, but not detected when accessing the services, −10 - malicious activity is detected when accessing the services.

Application signature verification (A5) is to assure that an application with a valid signature comes from the expected developers to make sure that: a) the APK (Android Application Package) has not been tampered in the time since it was signed, and b) the application's certificate matches that of the currently installed version. Assigned score could be: 10 - if application signature verification is supported, 0 - otherwise.

Application layer encryption (A6) is to apply the cryptographic protection for the data on application layer to prevent unauthorized disclosure. Assigned score could be: 10 - confidentiality and integrity are supported for all data on the application layer, 5 - confidentiality and integrity are supported for part of the data on the application layer, 0 - neither confidentiality nor integrity is supported for the data on the application layer.

Application integrity protection (A7) is to assure the integrity of the application, which applies IMA (Integrity Measurement Architecture) and extends the chain of trust to the application layer file. Assigned score could be: 10 - if application integrity protection is supported, 0 - otherwise.

REE OS vulnerabilities (A8) include the flaws and weaknesses, such as missing permission check, uninitialized data, buffer overflow, integer overflow, memory corruption, missing pointer check, etc. Assigned score could be: 0 - no well-known vulnerability, −5 - one or two well-known vulnerabilities are discovered, -10 - more than three well-known vulnerabilities are discovered.

Software-based defense mechanisms against memory vulnerabilities (A9) are the methods to prevent the memory corruption attacks (e.g., code injection and code reuse) based on software such as stack canaries, DEP (Data Execution Prevention), ASLR (Address Space Layout Randomization) and CFI (Control Flow Integrity). Assigned score could be: 10 - at least three defense mechanisms are supported, 5 - one or two defense mechanisms are supported, 0 - no defense mechanism is supported.

Verified boot (A10) is to assure the integrity of the software running on mobile terminal. It typically starts with a read-only portion of the terminal hardware which loads code and executes it only after cryptographically verifying that the code is authentic. Assigned score could be: 10 - if verified boot is supported, 0 - otherwise.

2.2 TEE Security Metrics and Scoring

TEE security metrics mainly include nine sub-metrics which are defined as follows. The scores of each sub-metric are also given.

Mobile application spoofing attack detection in TEE (B1) is to identify the attacks in which a malicious mobile application mimics the visual appearance and responses of the genuine one in TEE of the terminal. Assigned score could be: 10 - spoofing attack is never detected, 0 - spoofing attack(s) was detected before, but not detected when accessing the services, -10 - spoofing attack(s) is detected when accessing the services.

Virus detection in TEE (B2) is to identify the viruses, worms and other types of malwares in TEE of the mobile terminal. Assigned score could be: 10 - virus is never detected, 0 - virus(es) was detected before, but not detected when accessing the services, −10 - virus(es) is detected when accessing the services.

Trusted boot (B3) is to perform a measured and verified launch of Linux kernel. In the boot process, a log is maintained with the components that have been loaded and will be audited later. Trusted boot is implemented based on TEE and secure element. Assigned score could be: 10 - if trusted boot is supported, 0 - otherwise.

Trusted user interface (TUI) (B4) is to allow an application to interact directly with the user through the screen, which is controlled by the TEE and isolated from the REE. TUI defends the application against attacks, such as key logger and screen capturing. Assigned score could be: 10 - if TUI is supported, 0 - otherwise.

Biometric authentication (B5) is to verify and identify individuals based on their unique physical traits such as their iris, voice, face or fingerprints. In order to prevent the attacks such as fingerprint-jacking, printed photo and replayed video, biometric authentication is required to be implemented based on TUI. Assigned score could be: 10 - if biometric authentication is implemented based on TUI, 0 - otherwise.

TEE-based secure storage (B6) is to store sensitive data (e.g., user's biometric data, financial accounts, keys, etc.) securely in SE (Secure Element) or RPMB (Replay Protected Memory Block). The secure storage is implemented based on TEE. Assigned score could be: 10 - if data is stored in SE, 8 - if data is stored in RPMB, 0 - otherwise.

TEE-based real-time kernel protection (B7) is to use a security monitor located within TEE in order to provide the required protection, such as preventing kernel data from being directly accessed by user processes and preventing running unauthorized privileged code (i.e., code that has the kernel privilege) on the system. Assigned score could be: 10 - if TEE-based real-time kernel protection is supported, 0 - otherwise.

TEE OS integrity protection (B8) is to assure the integrity of the TEE OS files with establishing the chain of trust based on IMA. Assigned score could be: 10 - if TEE OS integrity protection is supported, 0 - otherwise.

TEE-based kernel integrity protection (B9) is to validate the integrity of the kernel files before they are loaded (and perhaps executed) based on IMA with the support from TEE. It is to help prevent modifications of kernel and driver code. Assigned score could be: 10 - if TEE-based kernel integrity protection is supported, 0 - otherwise.

2.3 Hardware Security Metrics and Scoring

Hardware security metrics mainly include nine sub-metrics which are defined as follows. The scores of each sub-metric are also given.

Hardware root of trust (C1) is to root the identity (e.g., hardware identifier) and cryptographic keys (e.g., root key, hardware unique key) in the hardware of a device in order to ensure the identity and authenticity of silicon devices at an atomic level. Assigned score could be: 10 - if hardware root of trust is supported, 0 - otherwise.

Secure element (SE) (C2) is to store sensitive data (e.g., financial accounts, keys, etc.) and run secure applications (e.g., mobile payment, mobile banking, etc.). Assigned score could be: 10 - if SE is supported, 0 - otherwise.

Hardware-based encryption (C3) is to use dedicated processor physically located on the encrypted drive to perform the task of authentication and encryption. Hardware-based encryption is more secure than software-based encryption because the encryption process is kept separate from the rest of the machine. Assigned score could be: 10 - if hardware-based encryption is supported, 0 - otherwise.

Memory vulnerabilities (C4) include the flaws and weaknesses in the memory such as memory leak, stack overflow, heap overflow, and allocation of memory without limitations. Assigned score could be: 0 - no well-known memory vulnerability, −5 - one or two well-known memory vulnerabilities are discovered, −10 - more than three well-known memory vulnerabilities are discovered.

Hardware-based defense mechanisms against memory vulnerabilities (C5) are the methods to prevent the memory corruption attacks (e.g., code injection and code reuse) based on hardware mechanisms, such as MTE (Memory Tagging Extension), PAC (Pointer Authentication Code) and BTI (Branch Target Identification). Assigned score could be: 10 - at least three defense mechanisms are supported, 5 - one or two defense mechanisms are supported, 0 - no defense mechanism is supported.

Memory encryption (C6) is to implement on-the-fly RAM encryption in order to protect data and code against attackers with physical access to a memory. Assigned score could be: 10 - if memory encryption is supported, 0 - otherwise.

Firmware version (C7) is to check which version of the firmware the terminal has. It is likely that the latest versions of firmware have less vulnerabilities than older ones. Assigned score could be: 10 - the latest firmware version, 0 - previous firmware version, −10 - outdated firmware version.

Protection against side-channel attack (SCA) (C8) is to prevent the attacks that extract secrets from a chip or a system. The countermeasures against side-channel attacks include designing cryptographic code to resist cache attacks, jamming the emitted channel with noise to deter timing attacks, power line conditioning and filtering to deter power-monitoring attacks, etc. Assigned score could be: 10 - at least three countermeasures against SCA are supported, 5 - one or two countermeasures against SCA are supported, 0 - no countermeasure against SCA is supported.

Protection against physical fault injection attack (FIA) (C9) is to prevent the attacks that insert the faults into the executing instructions and critical data. The countermeasures include concurrent error detection procedures as a hardware countermeasure against fault-injection-based cryptanalysis, an information leakage sensor against the laser fault injection attack, etc. Assigned score could be: 10 - at least three countermeasures against FIA are supported, 5 - one or two countermeasures against FIA are supported, 0 - no countermeasure against FIA is supported.

2.4 Communication Security Metrics and Scoring

Communication security metrics mainly include five sub-metrics which are defined as follows. The scores of each sub-metric are also given.

False base station detection (D1) is to check if the cellular base station, to which the terminal is connecting, is a legitimate one or false one. A false base station is employed for malicious and illegal purposes, such as sending scam messages, capturing signaling messages and collecting the terminal information. Assigned score could be: 10 - the terminal is connecting to a legitimate base station, −10 - the terminal is connecting to a false base station.

Security state of Wi-Fi access point (AP) (D2) is to check if the Wi-Fi AP the terminal connecting to is a secure one or rogue one, and also check if the Wi-Fi AP supports WPA2 (Wi-Fi Protected Access II) or WPA3. Assigned score could be: 10 - the

terminal is connecting to a secure AP with supporting at least WPA2, −10 - the terminal is connecting to a rogue AP, or the AP does not support WPA2 or WPA3.

Protected communication (D3) is to provide strong cryptographic protection of data transmitted over the networks (e.g., TLS, IPsec) to mitigate unauthorized disclosure or modification. Assigned score could be: 10 - if protected communication is supported, 0 - otherwise.

SMS verification code encryption (D4) is to encrypt the SMS verification code sent to the user from the server, as a kind of two-factor authentication, in order to make mobile application services (e.g., mobile banking, mobile payment) more secure and to reduce fraud. Assigned score could be: 10 - if SMS verification code encryption is supported, 0 - otherwise.

DNS (Domain Name System) encryption (D5) is to encrypt DNS messages and to make snoopers much harder to look into the DNS messages, or to corrupt them in transit. Two mechanisms are available for encrypting DNS, i.e., DoT (DNS over TLS) and DoH (DNS over HTTPS). It will further enhance user privacy. Assigned score could be: 10 - if DNS encryption is supported, 0 - otherwise.

2.5 Summary of Security Metrics and Scoring

33 security metrics with scores are defined for Android mobile terminals in this paper. The score of each security metric is obtained through the security management functional components of the terminal according to the security status of the terminal when accessing services. The metric is based on a scale of −10 to 10, with 10 being a perfect score. Moreover, the security metrics will increase with emerging security threats.

3 Determining Relative Weights of Security Metrics Based on AHP

Security metrics are defined for Android mobile terminal security assessment in Sect. 2. However, it's very difficult to determine the relative weights of these security metrics when conducting security assessment. Decision-makers may assign different weights to the same security metric. In such a problem of identifying relative weights of multiple metrics for security assessment, the most widely used method is analytical hierarchy process (AHP) [17, 18]. For example, He et al. [19] applied AHP to conduct the information security risk assessment. Petrova [20] performed cybersecurity risk assessment based on AHP to conduct cost-benefit analysis, design and optimize cybersecurity in the systems. Attaallah [21], Ahmad [22] and Ma [23] applied AHP to perform security assessment for medical devices which collect, store and process health data.

In order to support overall and comprehensive Android mobile terminal security assessment, the procedure of determining relative weights of security metrics based on AHP is summarized as follows.

1) *Creating hierarchy structure of security metrics.* A hierarchy structure of security metrics is created.
2) *Forming pairwise comparison matrixes.* The pairwise comparison matrixes are constructed by comparing the security metrics of each level in pairs. A Saaty's 9-point scale [17] in Table 1 is used to determine the importance and preference in pairwise

comparisons. Preferences at forming the pairwise comparison matrixes must satisfy the reciprocal and homogeneity conditions.

Table 1. A Saaty's 9-point scale.

Scale (a_{ij})	Meaning (the comparison of two security metrics)
1	the two metrics i and j are of equal importance
3	the metric i is slightly more important than the metric j
5	the metric i is obviously more important than the metric j
7	the metric i is strongly more important than the metric j
9	the metric i is extremely more important than the metric j
2, 4, 6, 8	median of the two adjacent judgements above
1, 1/2, …, 1/9	the importance ratio of the metric i and the metric j is $a_{ij,}$ so the importance ratio of the metric j and metric i is $a_{ji} = 1/a_{ij}$

3) *Calculating local weight of security metric through the matrix*. The local weight of security metric is calculated through each pairwise comparison matrix. The weight vector of the pairwise comparison matrix can be obtained by the method of solving the characteristic root. Furtherly the consistency index (CI) is also obtained. To verify the acceptability of consistency, it is necessary to calculate the consistency ratio (CR), which is performed by dividing the CI of the matrix by the corresponding value of the random index (RI) in Table 2 suggested by Saaty [17, 18]. If the CR is less than 0.1, the pairwise comparison matrix passes the consistency test. So, the eigenvector is the weight vector.

Table 2. RI for different number of the matrix.

Number of Matrix(n)	1	2	3	4	5	6	7	8	9	10	11
RI	0	0	0.52	0.89	1.11	1.25	1.35	1.40	1.45	1.49	1.51

4) *Calculating total weight of security metric.* The total weight of each security metric is equal to the sum of the product of the local weight of the security metric relative to the weight of the corresponding group that the security metric belongs to. The higher the total weight of a security metric, the better its ranking position, the greater influence on mobile application services.

Therefore, the overall security assessment score is obtained based on the equation:

$$Score = \sum_{i=1}^{n} Security\_metric\_score_i * Total\_weight_i \quad (1)$$

where:

- *Security_metric_score*$_i$: the score of the security metric i which is obtained based on the current security status and the corresponding score defined in Sect. 2.
- *Total_weight*$_i$: the total weight of the security metric i which is determined based on AHP.

4 An Architecture for Android Mobile Terminal Security Assessment System

Figure 2 shows a functional architecture for Android mobile terminal security assessment system with four main components in blue, which will be described in Sect. 4.1.

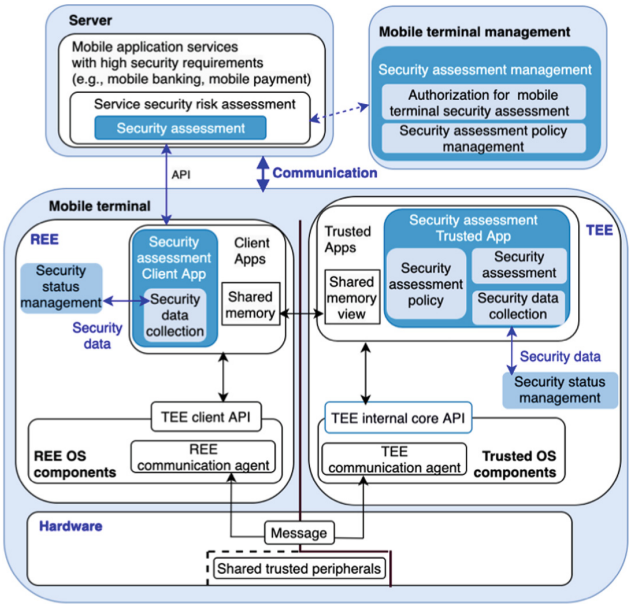

Fig. 2. Functional architecture for Android mobile terminal security assessment system.

4.1 Main Components of Functional Architecture

In Fig. 2, the *Security assessment Client App*, as a part of the mobile terminal software in REE, has the capabilities as follows.

- The *Security assessment Client App* is to receive the security assessment request together with a token from the *Security assessment* of the mobile application services and then send it to *Security assessment Trusted App* in TEE.
- The *Security assessment Client App* is to receive the security assessment reply together with the security level from *Security assessment Trusted App* in TEE and then send it to the *Security assessment* of the mobile application services.

- The *Security assessment Client App* is to collect security data from the Security status management in REE and send it to *Security assessment Trusted App* in TEE.

The *Security assessment Trusted App*, as a part of the mobile terminal software in TEE, has the capabilities as follows.

- The *Security assessment Trusted App* is to receive the security assessment request together with a token from the *Security assessment Client App* in REE.
- The *Security assessment Trusted App* is to validate the security assessment token provided by the *Security assessment* of the mobile application services.
- The *Security assessment Trusted App* is to calculate the overall security assessment score.
- The *Security assessment Trusted App* is to identify the security level based on the obtained overall security assessment score.
- The *Security assessment Trusted App* is to send the security assessment reply together with the security level to the *Security assessment Client App* in REE.
- The *Security assessment Trusted App* is to define security metrics and determine the relative weights of them.
- The *Security assessment Trusted App* is to get security data (e.g., hardware security status, etc.) from the *Security status management* in TEE.

The *Security assessment* of the mobile application services has the capabilities as follows.

- The *Security assessment* is to get the security assessment token from the *Security assessment management* of the mobile terminal management.
- The *Security assessment* is to send the security assessment request together with a token to the *Security assessment Client App* in REE and receive the security assessment reply together with security level from the *Security assessment Client App* in REE.

The *Security assessment management* of the mobile terminal management has the capabilities as follows.

- The *Security assessment management* is to upgrade *Security assessment Client App* in REE and S*ecurity assessment Trusted App* in TEE, including security assessment policy and the rule of security data collection.
- The *Security assessment management* is to generate the security assessment token used by the mobile application services to get the security assessment result.

4.2 Procedure of Mobile Terminal Security Assessment

According to Fig. 2, the procedure of a mobile application service obtaining an overall security assessment result from the mobile terminal is described as follows.

1) The *Security assessment* of a mobile application service sends a security assessment request to the *Security assessment Client App* in REE. The security assessment request includes a token, which is obtained from the *Security assessment management* in mobile terminal management.

2) The *Security assessment Client App* in REE receives the security assessment request and sends it to the *Security assessment Trusted App* in TEE through the communication mechanisms defined in [15]. The *Security assessment Client App* in REE collects security data (e.g., REE security status, communication security status) from time to time and forwards it to the *Security assessment Trusted App.*
3) The *Security assessment Trusted App* in TEE receives the security assessment request and verifies the token, and then calculates the overall security assessment score according to the Eq. (1).
4) The *Security assessment Trusted App* in TEE identifies the security level based on the obtained security assessment score.
5) The *Security assessment Trusted App* in TEE sends the security assessment reply including the security level to the *Security assessment Client App* in REE.
6) The *Security assessment Client App* in REE receives the security assessment reply including the security level and then sends it to the *Security assessment* of the mobile application services.

5 Implementation and Evaluation

According to the system architecture for Android mobile terminal security assessment in Sect. 4, the security assessment system is implemented. Android mobile terminal security assessment is carried out for mobile payment services.

5.1 Conducting Mobile Terminal Security Assessment for Mobile Payment

When implementing security assessment system, the security assessment policies should also be created. Of course, the related security metrics with the corresponding weights, as part of security assessment policies, should be defined.

5.1.1 Defining Security Metrics and Relative Weights

As for mobile payment services, relevant security metrics with the corresponding weights are defined as follows according to the method in Sect. 3.

1) A two-level hierarchy structure of security metrics is created and shown in Fig. 3. The level-2 security metric *C1 (Hardware root of trust)* also contributes to level-1 REE security assessment (A) as level-2 *A11*. In the same way, the level-2 *C2 (Secure element)* and *C3 (Hardware-based encryption)* also contribute to TEE security assessment (B) as level-2 *B10* and *B11* respectively.
2) Based on the two-level hierarchy structure of security metrics in Fig. 3, five pairwise comparison matrixes are constructed by comparing the security metrics of each level in pairs according to the Saaty's 9-point scale in Table 1. These five pairwise comparison matrixes are M_S, M_A, M_B, M_C and M_D, which are constructed for Android mobile terminal security assessment (S), REE security assessment (A), TEE security assessment (B), hardware security assessment (C) and communication security assessment (D) respectively.

In order to construct these five pairwise comparison matrixes, some experts for academic and industry background are consulted. After introducing this project briefly to

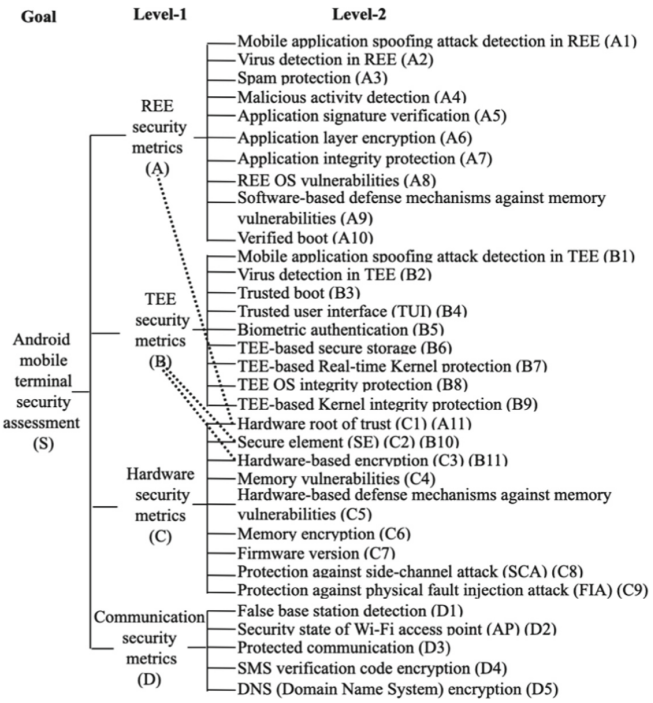

Fig. 3. Two-level hierarchy structure of security metrics.

them, the values of security metrics are provided by them based on their own experience in relevant field. According to their values, five pairwise comparison matrixes are constructed. For example, the pairwise comparison matrix M_S is constructed as the Eq. (2) for mobile payment services.

$$Ms = \begin{bmatrix} & A & B & C & D \\ A & 1 & 5 & 3 & 7 \\ B & 1/5 & 1 & 1/2 & 2 \\ C & 1/3 & 2 & 1 & 4 \\ D & 1/7 & 1/2 & 1/4 & 1 \end{bmatrix} \tag{2}$$

3) The weight vector W_S of the pairwise comparison matrix M_S is obtained as follows.

- *According* to the matrix M_S, the vector $\Phi = (2.3164, 0.4851, 0.9304, 0.2681)^{\mathrm{T}}$ is calculated by normalizing columns and then calculating the sum of the row. By normalizing the vector Φ, the eigenvector $W_S = (0.5791, 0.1213, 0.2325, 0.0670)^{\mathrm{T}}$ is obtained. The largest eigenvalue $\lambda_{\max}$ associated with the matrix M_S is obtained as 4.1185. Furtherly the consistency index (CI) is also obtained as 0.0095.
- To verify the acceptability of consistency, it is necessary to calculate the consistency ratio (CR), which is performed by dividing the CI of matrix M_S by the corresponding value of the random index (RI) in Table 2 suggested by Saaty [17, 18]. As for matrix S, CR = CI/RI = 0.0095/0.89 = 0.011, which is less than 0.1. So, the pairwise

comparison matrix M_S passes the consistency test. What's more, the eigenvector W_S is the weight vector.

- So, the weight of level-1 REE security assessment (A), TEE security assessment (B), Hardware security assessment (C) and Communication security assessment (D) for Android mobile terminal security assessment for mobile payment are 0.5791, 0.1213, 0.2325, 0.0670 respectively.

In the same way, the weight vectors of other four matrixes (i.e., M_A, M_B, M_C and M_D) for mobile payment services are obtained as follows.

$$W_{A1-A11} = (0.0780,\ 0.0551,\ 0.0377,\ 0.2769,\ 0.2029,\ 0.1081,\ 0.1484,\ 0.0204,\ 0.0170,\ 0.0249,\ 0.0307)^T \quad (3)$$

$$W_{B1-B11} = (0.0197,\ 0.0197,\ 0.1432,\ 0.0258,\ 0.0258,\ 0.2039,\ 0.0564,\ 0.0789,\ 0.1088,\ 0.2784,\ 0.0393)^T \quad (4)$$

$$W_{C1-C9} = (0.3086,\ 0.2197,\ 0.1557,\ 0.0264,\ 0.1074,\ 0.0748,\ 0.0350,\ 0.0515,\ 0.0209)^T \quad (5)$$

$$W_{D1-D5} = (0.0987,\ 0.0642,\ 0.2810,\ 0.5121,\ 0.0440)^T \quad (6)$$

So, the local weights of the level-2 security metrics are obtained based on the Eqs. (3)–(6), i.e., the weight vectors of the four pairwise comparison matrixes.

4) Table 3 shows the total weight of each security metric for mobile payment services, which is equal to the sum of the product of the local weight of the level-2 security metric relative to each weight of the level-1 security metric. Table 3 also shows the ranking position of each security metric, which is ranked based on its total weight. The higher the total weight of a security metric, the better its ranking position, the greater influence on mobile payment services.

In Table 3, security metrics *malicious activity detection (A4)*, *application signature verification (A5)* and *hardware root of trust (C1)* have a comparatively great proportion of the weight, which shows that they have great influence on the Android mobile terminal security assessment for mobile payment services.

5.1.2 Conducting Security Assessment of Mobile Terminal for Mobile Payment

Based on the system architecture for Android mobile terminal security assessment in Sect. 4, a security assessment system is implemented with Android mobile phone settings as follows.

- OS: Android 12.
- Linux kernel: 5.10.
- CPU: Snapdragon 8 Gen 1.
- Internal memory: RAM: 8G/12G, ROM: 256G/512G.
- TEE: QTEE 5.0.

According to the procedure of security assessment in Sect. 4.2, the mobile payment service receives an overall mobile terminal security assessment result including security level from the Android mobile terminals.

Table 3. Total weight of each security metric for mobile payment.

Factors of Level 1 (metric & weight)	Factors of level 2		Total Weight	Rank
	Security metric	Local weight		
REE security metrics (0.5791)	A1	0.0780	0.0451	7
	A2	0.0551	0.0319	10
	A3	0.0377	0.0219	13
	A4	0.2769	0.1603	1
	A5	0.2029	0.1175	2
	A6	0.1081	0.0626	6
	A7	0.1484	0.0859	4
	A8	0.0204	0.0118	19
	A9	0.0170	0.0099	20
	A10	0.0249	0.0144	17
TEE security metrics (0.1213)	B1	0.0197	0.0024	32
	B2	0.0197	0.0024	32
	B3	0.1432	0.0174	15
	B4	0.0258	0.0031	29
	B5	0.0258	0.0031	29
	B6	0.2039	0.0247	12
	B7	0.0564	0.0068	23
	B8	0.0789	0.0096	21
	B9	0.1088	0.0132	17
Hardware security metrics (0.2325)	C1	0.3086	0.0896*a	3
	C2	0.2197	0.0849*b	5
	C3	0.1557	0.0410*c	8
	C4	0.0264	0.0061	26
	C5	0.1074	0.0250	11
	C6	0.0748	0.0174	15
	C7	0.0350	0.0081	22
	C8	0.0515	0.0120	18
	C9	0.0209	0.0049	27
Communication security metrics (0.0670)	D1	0.0987	0.0066	25

(*continued*)

Table 3. (*continued*)

Factors of Level 1 (metric & weight)	Factors of level 2		Total Weight	Rank
	Security metric	Local weight		
	D2	0.0642	0.0043	28
	D3	0.2810	0.0188	14
	D4	0.5121	0.0343	9
	D5	0.0440	0.0030	31

[a]sum of C1 (0.0718) and A11 (0.0178)
[b]sum of C2 (0.0511) and B10 (0.0338)
[c]sum of C3 (0.0362) and B11 (0.0048)

The security level is defined based on the obtained security assessment score according to the Eq. (1). Security level is a measure of the security strength that the mobile terminal achieves. It represents the level corresponding to the required effectiveness of countermeasures and inherent security properties of the mobile terminals. For example, security level of mobile terminals can be defined in Table 4. High security level means the security of the mobile terminal is strong enough to access all mobile application services.

Table 4. Security rating scale.

Rating	Security assessment score
High	8.0–10.0
Medium	6.0–7.9
Low	Less than 5.9

It's suggested that only the mobile terminal with high security level (i.e., the security assessment score is higher than 8.0) is allowed to perform mobile payments.

5.2 Evaluation

According to the system architecture in Sect. 4, the security assessment system has already been implemented since June 2022. Figure 4 shows around 6 million smartphones which support the security assessment solution proposed in this paper in Sep. 2022.

As the number of smartphones supporting security assessment increases, so does the volume of payment transactions. Figure 5 shows that the smartphones supporting security assessment made the total online payment transactions per month, of which the fraudulent transactions were detected. Furtherly, it's obtained that the payment fraud rate per month is around 0.08%, which is shown in Fig. 6. It's observed that the payment fraud rate in June 2022 is around 0.05%. One of reasons is that it was the first month to

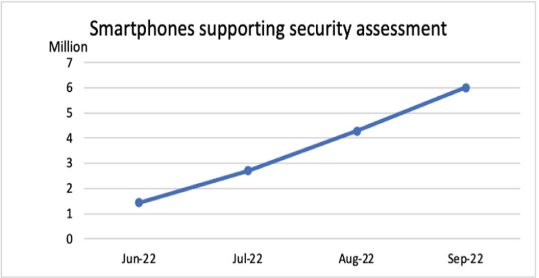

Fig. 4. Smartphones supporting security assessment since June 2022.

implement this security assessment system and the collected payment transaction data may be incomplete. Note: in this paper, security assessment of the mobile terminal is conducted for the mobile payment, which is only triggered by the third-party mobile Apps.

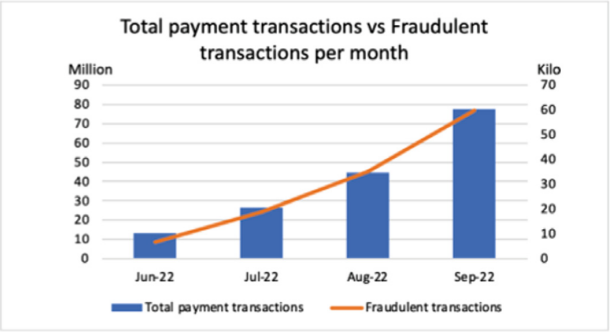

Fig. 5. Smartphones supporting security assessment performed the total payment transactions per month, of which fraudulent transactions were detected.

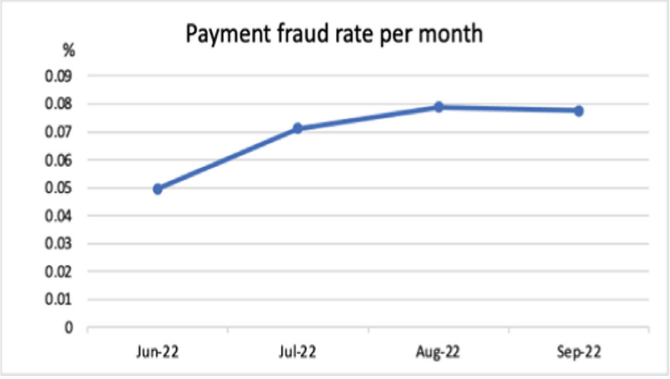

Fig. 6. Payment fraud rate per month was obtained according to Fig. 5.

It's found that top 5 security metrics with great influence on security assessment to support payment security risk assessment are *malicious activity detection (A4)*, *application signature verification (A5)*, *hardware root of trust (C1)*, *application integrity protection (A7)*, and *secure element (SE) (C2)*, shown in Table 3.

Currently, security assessment of Android mobile terminals is also being tested for mobile enterprise services. Consulting with some experts from academia and industry, the relative weights of security metrics are determined based on AHP. It's found that a security metric has different total weight values for mobile payment and mobile enterprise service, which is shown in Fig. 7.

It's also found that top 5 security metrics with great influence on security assessment to support enterprise service security risk assessment are *secure element (SE) (C2)*, *hardware root of trust (C1)*, *hardware-based encryption (C3)*, *protected communication (D3)*, and *TEE-based kernel integrity protection (B9)*.

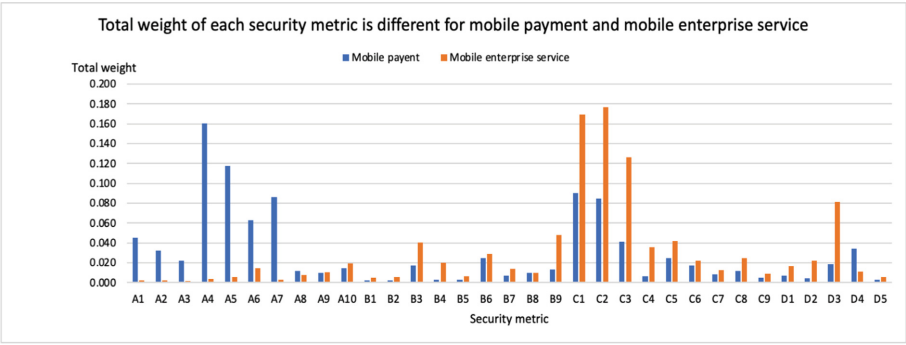

Fig. 7. Total weight of each security metric is different for mobile payment and mobile enterprise service.

5.3 Limitation

In this paper, the proposed approach for security assessment only works for the mobile terminal with Android OS. Moreover, mobile terminal security assessment is only conducted for mobile payment services, which are only triggered by the third-party mobile Apps. Some important security metrics for other mobile application services may not be included in this paper.

With emerging security threats, the security metrics will increase and the relative weights will change accordingly. It's not appropriate to consult experts from time to time to determine the relative weights. So, it is useful for optimizing the current solutions or design new solutions to determine the relative weight of each security metric without involving experts.

In this paper, all possible security metrics are taken into account for security assessment, which may cause computational burden and performance degradation if a lot of mobile application services need to conduct mobile terminal security assessment. In order to simplify the calculation and improve the system performance, it's suggested to

take the top security metrics with great influence on security assessment into account for service security risk assessment.

6 Conclusion and Future Work

In order to reduce security risks, mobile terminal security assessment should be performed before accessing the services with high security requirements. To support conducting mobile terminal security assessment, we defined security metrics with scores, determined the relative weights of these security metrics based on AHP and designed a system architecture. Moreover, we implemented a security assessment system and carried out security assessment for mobile payment services. Until 30 September 2022, around 6 million smartphones support the security assessment solution proposed in this paper. In September 2022, these smartphones made 77 million online payment transactions, of which 60,000 fraudulent transactions were detected, with the payment fraud rate of about 0.08%.

We also tried to determine the relative weights of security metrics for mobile enterprise services and found that a security metric has different weight values for different mobile application services.

In the future, we plan to optimize the current solution and consider the top security metrics with great influence on security assessment for service risk assessment in order to improve the system performance. We plan to design new solutions to determine the relative weight of each security metric without involving experts. We also plan to define new security metrics from time to time because new security threats emerge all the time.

References

1. Wang, F., Jiang, D., Wen, H., Song, H.: Adaboost-based security level classification of mobile intelligent terminals. J. Supercomput. **75**(11), 7460–7478 (2019). https://doi.org/10.1007/s11227-019-02954-y
2. Xi, Z., Chen, L., Chen, M., Dai, Z., Li, Y.: Power mobile terminal security assessment based on weights self-learning. In: 2018 10th International Conference on Communication Software and Networks (ICCSN) (2018). https://doi.org/10.1109/ICCSN.2018.8488313
3. Wu, S., Ma, Y., Jiang, H., Liu, T., Zuo, J., Peng, T.: Smart grid terminal security assessment method based on subjective and objective comprehensive weighting. In: 2021 IEEE Sixth International Conference on Data Science in Cyberspace (DSC) (2021). https://doi.org/10.1109/DSC53577.2021.00102
4. Ratchford, M., Wang, Y.: BYOD-insure: a security assessment model for enterprise BYOD. In: 2019 5th International Conference Mobile Secure Services MOBISECSERV 2019, no. 1, pp. 1–10 (2019). https://doi.org/10.1109/MOBISECSERV.2019.8686551
5. Visoottiviseth, V., Kotarasu, C., Cheunprapanusorn, N., Chamornmarn, T.: A Mobile application for security assessment towards the internet of thing devices. In: 2019 IEEE 6th Asian Conference on Defence Technology (ACDT), Bali, Indonesia, 13–15 November 2019 (2019). https://doi.org/10.1109/ACDT47198.2019.9072921
6. Othman, N.A.A., Norman, A.A., Kiah, M.L.M.: Information system audit for mobile device security assessment. In: 2021 3rd International Cyber Resilience Conference (CRC), Langkawi Island, Malaysia. https://doi.org/10.1109/CRC50527.2021.9392468

7. Cavalcanti, K., Viana, E., Lins, F.: An integrated solution for the improvement of the mobile devices security based on the android platform. IEEE Lat. Am. Trans. **15**(11), 2171–2176 (2017)
8. Vecchiato, D., Vieira, M., Martins, E.: Risk assessment of user-defined security configurations for android devices. In: 2016 IEEE 27th International Symposium on Software Reliability Engineering (ISSRE), Ottawa, ON, Canada (2016). https://doi.org/10.1109/ISSRE.2016.30
9. Asnar, Y., Hendradjaya, B.: Confidentiality and privacy information security risk assessment for Android-based mobile devices. In: 2015 International Conference on Data and Software Engineering, Yogyakarta, Indonesia. Proceedings of the ICoDSE 2015, pp. 1–6 (2015). https://doi.org/10.1109/ICODSE.2015.7436972
10. Khokhlov, I., Reznik, L.: Android system security evaluation. In: 2018 15th IEEE Annual Consumer Communications & Networking Conference (CCNC) (2018). https://doi.org/10.1109/CCNC.2018.8319325
11. Raza, N., Kirit, N.: Security evaluation for android OS using expert systems (2019). https://doi.org/10.13140/RG.2.2.27162.08640
12. Hayran, A., İğdeli, M., Yılmaz, A., Gemci, C.: Security evaluation of IOS and android. Int. J. Appl. Math. Electron. Comput. IJAMEC. **4**(Special Issue), 258–261 (2016). https://doi.org/10.18100/ijamec.270378
13. Khokhlov, I., Reznik, L.: Data security evaluation for mobile android devices. In: 2017 20th Conference of Open Innovations Association (FRUCT) (2017). https://doi.org/10.23919/FRUCT.2017.8071306
14. Zendehdel, G.A., Kaur, R., Chopra, I., Stakhanova, N., Scheme, E.: Automated security assessment framework for wearable BLE-enabled health monitoring devices. ACM Trans. Internet Technol. **22**(1), 1–31 (2022). https://doi.org/10.1145/3448649
15. GlobalPlatform Technology TEE System Architecture Version 1.2. https://globalplatform.org/wp-content/uploads/2017/01/GPD_TEE_SystemArch_v1.2_PublicRelease.pdf
16. ARM TrustZone for AArch64. https://developer.arm.com/documentation/102418/0101/?lang=en
17. Saaty, T.L.: The Analytic Hierarchy Process Mcgraw Hill, New York. Agricultural Economics Review, vol. 70 (1980)
18. Abu Dabous, S., Alkass, S.: Decision support method for multi-criteria selection of bridge rehabilitation strategy. Constr. Manage. Econ. **26**(8), 883–893 (2008). https://doi.org/10.1080/01446190802071190
19. He, M., An, X.: Information security risk assessment based on analytic hierarchy process. Indonesian J. Electr. Eng. Comput. Sci. **1**(3), 656–664 (2016). https://doi.org/10.11591/ijeecs.v1.i3.pp656-664
20. Petrova, V.: A cybersecurity risk assessment. Int. Sci. J. Sci. Tech. Union Mech. Eng. "Ind. 4.0" **6**(1), 37–40 (2021). https://stumejournals.com/journals/i4/2021/1/37
21. Attaallah, A., Ahmad, M., Jamal Ansari, M.T., Pandey, A.K., Kumar, R., Khan, R.A.: Device security assessment of internet of healthcare things. Intell. Autom. Soft Comput. **27**(2), 593–603 (2021). https://doi.org/10.32604/iasc.2021.015092
22. Ahmad, M., Al-Amri, J.F., Subahi, A.F., Khatri, S., Seh, A.H., Nadeem, N., Agrawal, A.: healthcare device security assessment through computational methodology. Comput. Syst. Sci. Eng. **41**(2), 811–828 (2021). https://doi.org/10.32604/csse.2022.020097
23. Ma, P., Wang, Z., Hei, X., Zou, X., Zhang, J., Liu, Q., et al.: A quantitative approach for medical imaging device security assessment. In: 2019 IEEE/IFIP International Conference on Dependable Systems and Networks Supplemental, Portland, OR, USA (2019). https://doi.org/10.1109/DSN-S.2019.00008