Research on android user privacy permission analysis and protection mechanism under big data environment

Mei Liu*, and *Qun* Wang

Shandong Xiehe University, 250100, Jinan, Shandong, China

Keywords: Android, Big data, Privacy permissions, Protection mechanisms.

Abstract. With the rapid development of big data technology, the issue of user privacy security on the Android platform is becoming increasingly prominent. This paper aims to conduct an in-depth analysis of the privacy permissions of Android users under the big data environment and explore effective protection mechanisms. Through research on permission management, application behavior, and user privacy leakage pathways in the Android system, this paper proposes a comprehensive privacy protection strategy to enhance the privacy security level of Android users in the big data environment.

1 Introduction

The widespread application of big data technology provides rich data resources for Android applications, but it also brings risks of user privacy leakage. The openness and fragmentation of the Android system make privacy protection an urgent issue to be addressed. Therefore, researching the privacy permission analysis and protection mechanism of Android users under the big data environment holds important practical significance and application value.

2 Analysis of the Current Status of Android User Privacy Permissions

2.1 Overview of android permission management mechanism

The Android permission management mechanism is a crucial part of the Android operating system, designed to ensure that apps follow user intentions when accessing system resources or performing specific operations, and safeguard user privacy and data security.

* Corresponding author: 710060230@qq.com

First of all, permissions in the Android system are mainly divided into two categories: Normal Permissions and Dangerous Permissions. Normal permissions typically involve low-risk operations, such as network access or device vibration, and these permissions are automatically granted when the app is installed. On the other hand, Dangerous Permissions cover operations that may involve user privacy or system security, such as accessing contacts, camera, or recording, and these permissions need explicit authorization from the user at runtime.

Next, the declaration and request of permissions are key steps in the Android permission management mechanism. When developers are developing applications, they must declare all the permissions required by the application in the AndroidManifest.xml file. For normal permissions, the system will automatically grant them; for dangerous permissions, the system will display a permission request dialog to the user at runtime, and the user can choose to accept or reject it.

Finally, permission management is an important means for the Android system to ensure that application behavior is compliant. The system monitors the usage of permissions by applications to ensure that they only perform operations within the scope of the permissions granted to them. If a user revokes a certain permission, the application will no longer be able to perform operations related to that permission. Additionally, users can also view and manage the permissions of installed applications in the device's settings menu to further control the use of permissions by applications.

2.2 Analysis of existing privacy permission issues

In the current Android system, privacy permission issues have always been a concern. Although Android has established a relatively complete permission management mechanism, there are still some significant issues in practical applications.

Firstly, the issue of permission abuse cannot be ignored. Some applications may excessively request and use permissions during the development process for the purpose of collecting user data, pushing advertisements, and other commercial purposes. This not only violates the user's right to information and choice but may also lead to the disclosure and misuse of user privacy.

Secondly, excessive permission granting is also a common issue. When installing applications, users are often required to grant a series of permissions, some of which may not be directly related to the core functions of the application. This "one-size-fits-all" permission granting approach undoubtedly increases the risk of user privacy leakage.

Furthermore, permission leakage is also a major challenge facing the current Android system. Due to the openness and complexity of the Android system, some applications may have security vulnerabilities, allowing malicious software or hackers to exploit these vulnerabilities to obtain sensitive permissions and data from users. Once this data is leaked, users will face serious privacy threats.

In summary, the current Android system still faces numerous issues in terms of privacy permissions, including permission misuse, excessive permission granting, and permission leaks. These problems not only infringe upon users' privacy rights but also impact the overall security and trustworthiness of the Android system. Therefore, it is crucial to further strengthen the privacy permission management in the Android system, enhance users' awareness of privacy protection, and ensure the security and privacy of user data.

2.3 Analysis of user privacy breach cases

A popular social networking app requested a large number of permissions when users installed it, including access to contacts, photos, and video recording. However, during

actual usage, users found that the app did not adequately explain the purposes of these permissions. Moreover, it frequently collected users' sensitive information in the background, such as call logs and message contents. The app developer used this information for advertising and data analysis, seriously infringing on users' privacy rights.

The real impact of privacy permission issues on users cannot be ignored. To protect users' privacy rights, we need to strengthen the security protection and vulnerability fixes of the Android system, while also raising awareness and capabilities regarding privacy protection among users. Additionally, app developers should comply with relevant laws and regulations, respect users' right to information and choice, and avoid excessive collection and use of user data.

3 Analysis of privacy permissions for android users

3.1 Analysis of permission management mechanisms

The Android system controls applications' access to system resources through a permission management mechanism. However, existing permission management mechanisms have some issues, such as overly broad permission granularity and permission abuse. These problems make it difficult for users to determine whether an application truly needs certain sensitive permissions when installing and using the app, thereby increasing the risk of privacy leaks.

3.2 Analysis of application behavior

By analyzing the behavior of Android applications, we can understand how these apps collect, process, and utilize user's private data. Some applications may excessively collect user's private data or upload data without user consent, which can lead to privacy breaches.

3.3 Analysis of user privacy leakage channels

In the big data environment, there are various ways in which user privacy can be leaked, including network transmission, data sharing, third-party service integration, and more. These channels can be exploited by malicious applications or attackers to steal users' private data.

4 Research on android user privacy protection mechanisms

In response to the shortcomings of existing privacy protection mechanisms, this paper proposes a comprehensive Android user privacy protection mechanism. The mechanism mainly includes the following aspects:

4.1 Fine-grained permission management

The implementation of fine-grained permission management relies on a detailed analysis and redesign of Android system permissions. Traditional permission management typically exists in a coarse-grained form, where applications need to request a set of permissions at once. However, fine-grained permission management breaks down each permission into smaller units, allowing applications to request specific permissions based on their specific functional requirements.

The key to implementing this mechanism lies in:

Permission splitting and redefinition: Developers need to conduct a detailed review and splitting of the existing Android permissions in collaboration with partners or system maintainers. Each newly defined permission should correspond to specific operations or resource access.

Permission declaration and request interface: The Android SDK provides new API interfaces that allow applications to request fine-grained permissions as needed during runtime. Applications need to explicitly declare the required fine-grained permissions in their code and request them through the API when actually used.

User interface optimization: The system UI layer needs to optimize permission requests to display clearer permission descriptions and purposes, helping users understand and make decisions.

4.2 Dynamic permission request and authorization

Dynamic permission request and authorization mechanism ensures that applications request permissions from users at runtime based on actual needs. The implementation of this mechanism relies on Android's runtime permission checking system.

The implementation steps are as follows:

Runtime permission check: When an application attempts to perform an operation that requires a specific permission, the system first checks if the application has been granted that permission.

Permission request trigger: If the application has not been granted the required permission, the system triggers a permission request dialog, displaying the reason and description of the permission request to the user.

User authorization decision: Based on the information in the dialog, the user can choose to accept or deny the permission request. The user's decision is recorded by the system and used for subsequent permission checks.

In this way, applications can only request permissions when they truly need them, and users can make decisions based on the actual behavior and usage of the application.

4.3 De-identification of privacy data

De-identification of privacy data is a technical means of protecting sensitive data by replacing, deleting, or transforming data in a way that reduces the risk of misuse without losing its original value.

Key steps to implement this mechanism include:

Sensitive data identification: The system first needs to be able to identify which data belongs to sensitive data, such as ID numbers, bank account numbers, passwords, etc. This can be achieved through methods like regular expression matching, machine learning algorithms, etc.

De-identification algorithm design: Design corresponding de-identification algorithms for different types of sensitive data. For example, for ID numbers, only partial numbers may be retained or replaced with asterisks; for passwords, hashing or encryption may be applied for storage.

Data de-identification processing: Apply these de-identification algorithms to sensitive data during data collection, transmission, and storage processes. Ensure that even in the event of data leakage, sensitive information cannot be easily accessed.

4.4 Privacy leakage detection and response

Privacy leakage detection and response mechanisms involve monitoring the behavior of applications and network traffic to promptly identify and address potential privacy risks.

Implementation mechanisms include:

Behavior monitoring and network traffic analysis: The system monitors and analyzes the behavior of applications and network traffic in real-time using built-in security components or third-party security tools.

Risk identification and warnings: Based on monitoring and analysis results, the system can identify potential privacy leakage risks, such as abnormal data transmission, unauthorized use of permissions, etc. Once risks are detected, the system immediately sends warning notifications to users.

Risk response and handling: Depending on the nature and severity of the risks, the system can take different response measures. For example, for minor risks, it may restrict certain functions of the application or alert users; for serious risks, it may immediately terminate the application or report to relevant departments for handling.

Through this mechanism, the system can promptly detect and respond to privacy leakage risks, ensuring the security and privacy of user data.

5 Conclusion

This article proposes a series of effective privacy protection strategies through the analysis of Android user privacy permissions and protection mechanisms in the context of big data. These strategies aim to enhance the privacy security level of Android users in a big data environment and reduce the risk of privacy breaches. In the future, we will continue to delve into research on Android user privacy protection technology and explore more comprehensive and efficient privacy protection solutions.

References

1. Kouliaridis V ,Karopoulos G ,Kambourakis G .Assessing the Security and Privacy of Android Official ID Wallet Apps[J].Information,2023,14(8):
2. Gianni D, Eslam F, Massimo F, et al. Privacy-preserving malware detection in Android-based IoT devices through federated Markov chains[J].Future Generation Computer Systems,2023,14893-105.
3. Hutchinson S ,Stanković M ,Ho S , et al.Investigating the Privacy and Security of the SimpliSafe Security System on Android and iOS[J].Journal of Cybersecurity and Privacy,2023,3(2):145-165.
4. Shinelle H ,Meraj M M ,Nicholas W , et al.Investigating Wearable Fitness Applications: Data Privacy and Digital Forensics Analysis on Android[J].Applied Sciences,2022,12(19):9747-9747.
5. Horn L I ,Ali M B ,Awais R , et al.On the privacy of mental health apps[J].Empirical Software Engineering,2022,28(1):2-2.
6. E. K O, O. C Y .A Multilateral Privacy Impact Analysis Method for Android Applications [J].Annals of Science and Technology, 2022, 7(2):1-20.
7. Caputo D, Pagano F, Bottino G, et al.You Can't Always Get What You Want: Towards User-Controlled Privacy on Android [J].IEEE Transactions on Dependable and Secure Computing, 2023, 20(2):975-987.

8. Haoyu L ,Paul P ,J D L .On the data privacy practices of Android OEMs.[J].PloS one,2023,18(1):e0279942-e0279942.
9. Suchul L .Distributed Detection of Malicious Android Apps While Preserving Privacy Using Federated Learning [J].Sensors, 2023, 23(4):2198-2198.
10. Hasnat A ,Komal B ,Muhammad Y , et al.Security Hardened and Privacy Preserved Android Malware Detection Using Fuzzy Hash of Reverse Engineered Source Code[J].Security and Communication Networks,2022,2022.