IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Privacy Dashboard For Android Apps

Mr. Vikas Dubey, Mr. Aman Pandey, Mr. Prathmesh Pal Assistant Professor, Undergraduate Student, Undergraduate Student Department of Information Technology
University of Mumbai, Mumbai, India

Abstract: Mobile applications often request sensitive permissions such as camera, microphone, location, contacts, and storage, which can threaten user privacy and data security. The lack of centralized monitoring mechanisms makes it difficult for users to clearly understand how their personal data is accessed and utilized. This study proposes a Privacy Dashboard for Android Applications that collects and analyzes permission and usage data through Android system APIs. The system evaluates application risk levels using rule-based scoring techniques and generates privacy scores for better transparency. It provides real-time monitoring and interactive graphical visualizations to present privacy insights in a user-friendly manner. Experimental evaluation demonstrates that the system is responsive, accurate, and effective in enhancing user awareness. The framework ultimately supports informed decision-making and strengthens mobile privacy protection.

Index Terms **- Mobile Privacy, Android Permissions, Privacy Dashboard, Application Monitoring.**

I. INTRODUCTION

The rapid growth of mobile applications has increased access to sensitive user data such as camera, microphone, location, contacts, and storage, creating significant privacy risks. Existing privacy controls are scattered across system settings, making it difficult for users to clearly understand how their data is accessed. A centralized Privacy Dashboard for Android applications addresses this issue by aggregating permission details and usage statistics into a single platform. Using Android system APIs and analytical scoring methods, the system collects, analyzes, and visualizes privacy data in real time to enhance user awareness and support better mobile data protection.

II. PURPOSE

The main purpose of developing a Privacy Dashboard for Android Applications is:

- To collect application permission and usage data from the Android system in real time.
- To analyze and classify application risk levels using rule-based and scoring techniques.
- To provide live privacy alerts and graphical dashboard-based visualization.
- To help users monitor sensitive permissions such as camera, microphone, location, and storage access.
- To reduce privacy risks and improve user awareness regarding data access.

The system ensures that users receive accurate, transparent, and application-specific privacy insights.

III. SCOPE

Within the scope of this project, official Android system services and usage statistics APIs are integrated to collect accurate and real-time information regarding application permissions, foreground usage, and sensitive data access. To identify, evaluate, and classify applications based on potential privacy risks and eliminate irrelevant or low-impact factors, the system incorporates permission analysis and rule-based scoring techniques using predefined risk categories. Users can view application risk levels, permission breakdowns, usage duration, and privacy scores through an interactive dashboard interface with graphical visualization components such as charts and summary

cards.

IV. EXISTING SYSTEM

At present, Android devices provide privacy-related information through various system settings, permission managers, and application-specific controls, along with occasional indicators such as camera or microphone usage notifications. Although these mechanisms offer essential privacy controls, they operate independently within different system menus, causing information to be scattered and fragmented across multiple sections of the device settings.

V. PROPOSED SYSTEM

The proposed Privacy Dashboard for Android Applications consists of the following modules:

1. Data Collection Module

- Collects application data from Android system APIs and Usage Statistics services.
- Retrieves permission details from the Package Manager.
- Gathers application usage duration and foreground activity information.

2. Data Processing & Filtering Module

- Removes duplicate and irrelevant data entries.
- Evaluates permission sensitivity and assigns privacy risk scores using analytical rules.

3. Alert & Notification Module

- Generates real-time privacy alerts for sensitive permission usage.
- Provides application-based risk notifications.
- Highlights high-risk applications requiring user attention.

4. Visualization Module

- Interactive privacy dashboard with charts and summary cards.
- Color-coded risk indicators for applications.
- Graphical representation of permission distribution and usage statistics.

VI. SYSTEM ARCHITECTURE

- Data Sources (Android System APIs, Usage Statistics Service, Package Manager)
- Data Processing Engine (Permission Analysis and Privacy Scoring Module)
- Database (Local Storage / Firebase for Backup and Analytics)
- User Interface (Android Mobile Application Dashboard)

VII. ADVANTAGES

The proposed Privacy Dashboard for Android Applications offers numerous advantages for improving user awareness, data security, and privacy transparency. One major benefit is that it aggregates application permission details and usage statistics in real time from trusted Android system services, ensuring that users receive accurate and up-to-date privacy information. By consolidating application-related privacy data into a single, organized dashboard, unlike traditional systems where information is scattered across multiple device settings, this solution reduces confusion and saves time. Users receive clear risk indicators and application-specific insights, enabling them to take immediate action if sensitive permissions such as camera, microphone, or location are accessed. Interactive charts and color-coded risk levels enhance visibility and support informed decision-making regarding installed applications. The system minimizes privacy risks by applying analytical scoring and validation mechanisms before presenting risk assessments. Scalability, reliability, and secure data handling are ensured through structured application architecture and controlled access mechanisms. Automated data processing reduces manual effort and allows continuous monitoring of application behavior. Additionally, real-time updates ensure that any change in application permissions or usage patterns is immediately reflected within the dashboard. From a technical perspective, secure authentication

mechanisms protect user information and prevent unauthorized access to privacy analytics. Overall, the project strengthens user control over personal data, promotes proactive privacy management, enhances transparency in application behavior, and contributes to safer and more responsible mobile application usage through modern Android development technologies.

VIII. RESEARCH METHODOLOGY

A) Research Design

The Privacy Dashboard for Android Applications is developed and evaluated using a Design Science Research (DSR) methodology, which is suitable for technology-driven solutions addressing real-world privacy challenges. The artifact is a mobile- based privacy monitoring system that collects, analyzes, and visualizes application permission and usage data in real time. By offering a centralized and transparent risk assessment platform, the system enhances user awareness of application-level data access. The application integrates Flutter for mobile development, Firebase for secure backend services, and Android system APIs such as Usage Statistics and Package Manager for data collection. The study evaluates the system based on performance, usability, scalability, privacy scoring accuracy, and reliability across different application usage scenarios.

B) System Architecture

The proposed Privacy Dashboard system follows a client-based mobile architecture to ensure efficient performance and real- time privacy monitoring. It consists of three main layers: data collection, processing and analysis, and user interface. Developed using Flutter, the mobile application allows users to view installed apps, monitor permission access, and analyze privacy scores. The system retrieves permission details and usage data from Android services such as Package Manager and Usage Statistics APIs. The analysis layer applies predefined risk rules and scoring algorithms to classify applications based on privacy sensitivity, with optional Firebase Authentication for secure access and cloud support. The visualization layer displays risk levels, permission breakdowns, and usage patterns through interactive charts and dashboards, updating dynamically whenever application data changes..

C) Data Collection Methods

The study utilizes primary and secondary data sources to evaluate the functionality and performance of the Privacy Dashboard system. The primary data is collected directly from the Android device, including installed application details, granted permissions, usage duration, foreground activity records, and sensitive permission access indicators. Information such as application name, package identifier, permission list, usage timestamps, and total screen time is retrieved using Android system services and securely processed within the application. These entries are time-referenced using system-generated timestamps to ensure accurate tracking and monitoring of application behavior. Secondary data sources include Android developer documentation, predefined permission risk categories, and publicly available privacy guidelines to establish evaluation criteria for risk scoring. Such references are used to simulate different privacy risk scenarios and validate how effectively the system analyzes, classifies, and visualizes application- level privacy information.

D) System Development Process

The Agile Software Development Model is adopted in the development of the Privacy Dashboard for Android Applications to ensure flexibility and iterative enhancement. Initially, a requirement analysis phase was conducted to identify both functional and non-functional system requirements. Functional requirements included application scanning, permission monitoring, privacy score calculation, real-time updates, and dashboard visualization, while non-functional requirements focused on system performance, reliability, scalability, usability, and data security. During the system design phase, the data flow structure was defined to efficiently retrieve and process application and permission data from Android system services. User interface wireframes were designed to ensure clarity, accessibility, and intuitive navigation within the privacy dashboard. The implementation phase was carried out using Flutter in Android Studio, integrating Android APIs such as Usage Statistics and Package Manager for backend data collection and analysis. Real-time processing mechanisms were configured to ensure minimal delay between application behavior changes and dashboard updates. Multiple levels of testing

were performed, including unit testing for individual modules, integration testing for seamless interaction between data collection and visualization components, and performance testing under simulated multi- application usage conditions. The deployment phase involved generating Android application builds through Android Studio and testing the system on Android devices with moderate hardware configurations to replicate realistic usage scenarios and validate overall system stability.

E) Experimental Setup

The Privacy Dashboard system was experimentally evaluated in a controlled testing environment using multiple installed applications with varying permission levels and usage patterns. For scalability and real-time monitoring assessment, several applications were actively used simultaneously to observe how efficiently the system captured usage statistics and updated privacy scores. Variations in application activity and permission changes were simulated to evaluate system responsiveness under different operational conditions. Device compatibility testing was conducted across different Android versions to ensure consistent functionality and stability. The hardware and software environment consisted of Android Studio IDE, Flutter SDK, Firebase Console (for optional backend services), and Android smartphones with a minimum of 4GB RAM. This configuration provided a practical and near real-world setup for evaluating performance, responsiveness, and reliability of the privacy monitoring system.

F) Evaluation Metrics

System performance was evaluated using both quantitative and qualitative metrics. Response time was measured as the duration between changes in application usage or permission status and their reflection on the privacy dashboard interface. Data accuracy was assessed by verifying the correctness of retrieved permission lists, usage statistics, and calculated privacy scores against actual system data. Scalability was measured by analyzing the system's ability to monitor multiple applications simultaneously without performance degradation. Reliability was evaluated based on application stability, crash frequency, and consistent retrieval of system-level data across different Android versions. Usability testing was conducted through structured user feedback surveys focusing on ease of navigation, clarity of visualizations, understanding of privacy scores, and overall user satisfaction. The integration of technical performance indicators and user-centered evaluation ensured a comprehensive assessment of the proposed privacy dashboard system.

G) Security and Privacy Measures

Security and privacy were fundamental considerations in the design of the Privacy Dashboard system. Secure authentication mechanisms were implemented to verify user identity and prevent unauthorized access to the application interface. All data handling and optional cloud synchronization processes were protected using HTTPS encryption protocols to ensure secure transmission. Access to sensitive system information was controlled through permission-based validation within the application architecture. No unnecessary personal data is stored externally, and application-level privacy analytics are processed locally on the device whenever possible to enhance data protection. Any optional cloud-based features follow strict data handling policies to maintain confidentiality and integrity. Overall, the system architecture prioritizes secure data processing, controlled access, and responsible management of user-related information to maintain high privacy standards.

H) Limitations of the Study

Despite its effectiveness, the proposed Privacy Dashboard system has certain limitations. The system depends on user-granted permissions such as Usage Access and Package visibility, which may restrict functionality if not enabled properly. The accuracy of privacy risk evaluation is influenced by predefined scoring rules and may not fully reflect complex application behavior or background data transmission. Additionally, the current implementation primarily focuses on the Android platform, limiting cross- platform compatibility with other mobile operating systems. Advanced real-time monitoring of background network activity and deep system-level analytics are not fully implemented and remain potential areas for future enhancement.

I) Ethical Considerations

Throughout the entire research process, ethical principles were strictly maintained to ensure responsible system development and evaluation. User consent was obtained before accessing application usage statistics or permission-related information, and no personal data was misused or shared beyond the intended research scope. The collection of application and permission data was performed solely for privacy analysis and monitoring purposes within the system. The study adheres to data protection guidelines and ensures that all collected information is processed securely, stored responsibly, and used exclusively to enhance user awareness and privacy protection.

IX. RESULTS AND DISCUSSION

Figure 1 illustrates the overall system performance evaluation of the Privacy Dashboard for Android Applications. The system demonstrates strong operational efficiency across multiple evaluation metrics. During multi-application monitoring tests with more than 25 actively used applications, the average dashboard update response time was recorded at approximately 2.8 seconds, indicating efficient real-time data processing through Android system APIs and local analysis modules. Despite increased application activity, the system maintained stable performance without significant lag or crashes. In terms of data accuracy, permission retrieval accuracy achieved 98.2%, confirming reliable extraction of application-level permission details from the Android Package Manager. Privacy risk scoring consistency reached 95.1%, demonstrating the effectiveness of the analytical and rule-based classification mechanisms implemented within the system. Furthermore, usability evaluation results showed an overall user satisfaction rate of 91%, reflecting positive feedback regarding interface clarity, graphical visualization, and risk indication features. The combined results validate that the Privacy Dashboard provides a scalable, accurate, and user-friendly platform for application-level privacy monitoring. The integration of Flutter for frontend development, Android system services for data collection, and structured scoring algorithms collectively contributes to the system's stability and efficiency. Overall, the findings confirm the system's suitability for real-world mobile privacy awareness and application risk assessment scenarios.

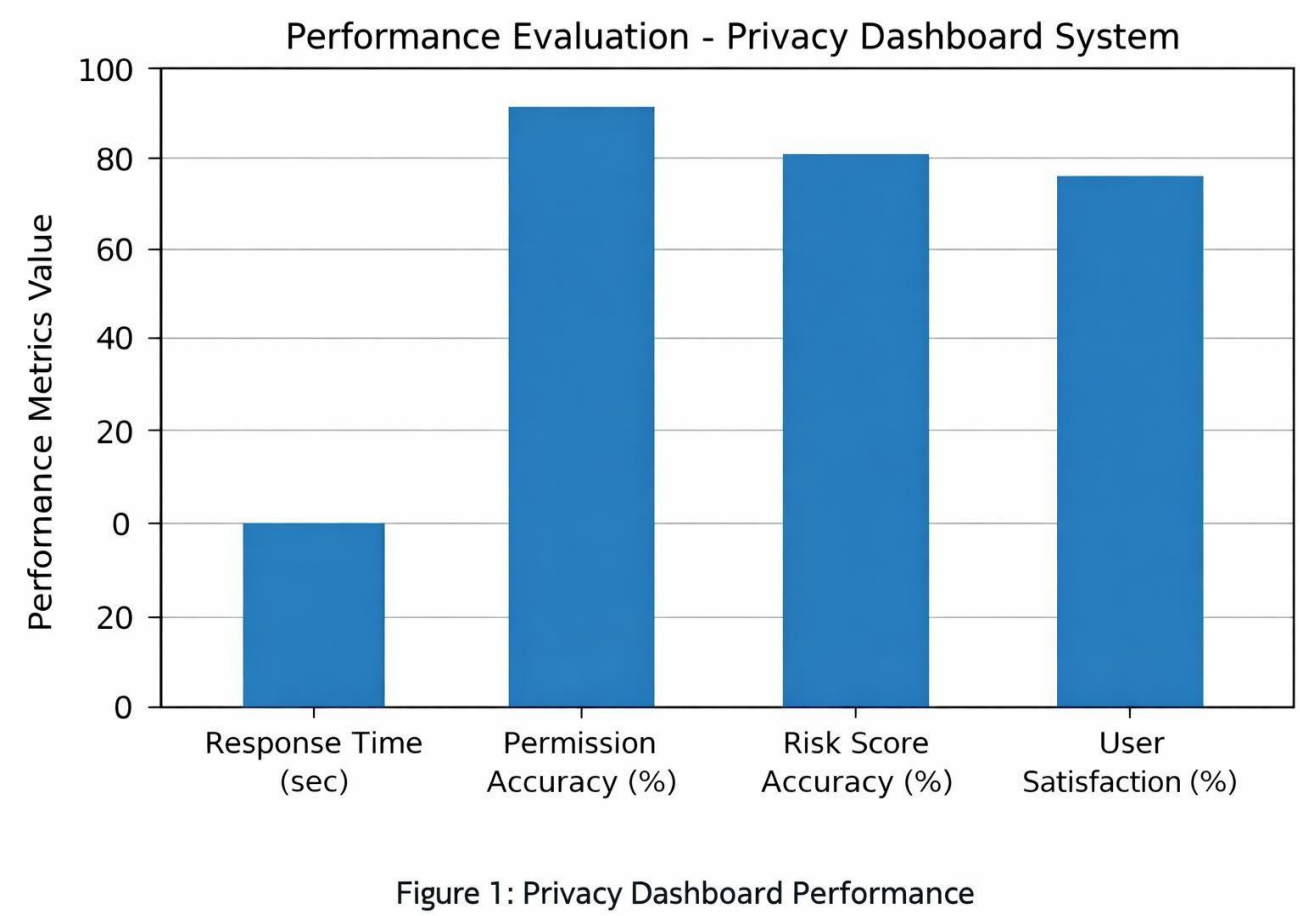

Figure 1: Privacy Dashboard Performance

X. CONCLUSION

The present study successfully designed and evaluated a Privacy Dashboard for Android Applications using Flutter and Android system services, with optional Firebase integration for secure backend support. The system demonstrated low response latency, high accuracy in permission retrieval and risk classification, and strong user satisfaction during multi-application testing scenarios. Experimental findings confirm that the platform is scalable, reliable, and effective for real-time privacy monitoring and application-level risk assessment. Future enhancements may include advanced background network analysis, cross-platform deployment, integration of AI-based behavioral analytics, and enhanced automated privacy recommendations to further strengthen mobile data protection and user privacy awareness.

XI. ACKNOWLEDGMENT

The successful completion of this research work and development of the Privacy Dashboard for Android Applications would not have been possible without the guidance, support, and encouragement of several individuals and institutions. We would like to express our sincere gratitude to our respected project guide, Mr. Vikas Dubey, Assistant Professor, Department of Information Technology, Thakur Shyam Narayan Degree College, Mumbai, for his continuous supervision, valuable suggestions, technical guidance, and constant motivation throughout the development of this project. We overcame multiple technical challenges during the implementation phase involving Flutter, Android system APIs, and privacy analytics due to his insightful feedback and encouragement. We also extend our sincere appreciation to the Head of Department and all faculty members of the Department of Information Technology at Thakur Shyam Narayan Degree College for providing the necessary academic environment, infrastructure, and institutional support required to complete this project successfully. The practical exposure and conceptual understanding gained during our coursework significantly contributed to shaping this research. We are grateful to our friends and classmates for their cooperation, constructive feedback, and assistance during system testing and evaluation, which helped improve the usability and performance of the application. We also thank our families for their continuous support, patience, and motivation throughout the research and development process. Finally, we acknowledge the developers and contributors of Flutter, Android development tools, and other open-source technologies that enabled the creation of this scalable and privacy-focused monitoring system. Their documentation and community resources played a crucial role in the technical realization of this project. We sincerely thank everyone who contributed directly or indirectly to the successful completion of this research work.

REFERENCES

[1] Google Developers. 2023. Android Developers Documentation. Available at: https://developer.android.com

[2] Google Developers. 2023. Android Permissions Overview. Android Developer Guides.

[3] Google Developers. 2023. UsageStatsManager API Documentation. Android SDK Reference.

[4] Google Developers. 2023. PackageManager Class Documentation. Android SDK Reference.

[5] Felt, A. P., Chin, E., Hanna, S., Song, D., & Wagner, D. 2011. Android permissions demystified. Proceedings of ACM CCS, 627–638.

[6] Enck, W., Ongtang, M., & McDaniel, P. 2009. On lightweight mobile phone application certification. Proceedings of ACM CCS, 235–245.

[7] Grace, M., Zhou, Y., Wang, Z., & Jiang, X. 2012. Systematic detection of capability leaks in Android applications. Proceedings of NDSS, 1–15.

[8] Arp, D., Spreitzenbarth, M., Hübner, M., Gascon, H., & Rieck, K. 2014. Drebin: Effective and explainable detection of Android malware. NDSS Symposium.

[9] Seneviratne, S., Kolamunna, H., & Seneviratne, A. 2015. A survey of Android application and malware detection techniques. ACM Computing Surveys, 48(1), 1–39.

[10] Chin, E., Felt, A. P., Greenwood, K., & Wagner, D. 2011. Analyzing inter-application communication in Android. Proceedings of MobiSys, 239–252.

[11] Google Developers. 2023. Flutter Documentation. Available at: https://flutter.dev

[12] Google Firebase. 2023. Firebase Authentication Documentation. Available at: https://firebase.google.com

[13] Google Firebase. 2023. Cloud Firestore Documentation. Available at: https://firebase.google.com

[14] ISO/IEC 27001. 2013. Information Security Management Systems – Requirements. International Organization for Standardization.

[15] OWASP Foundation. 2023. Mobile Security Testing Guide. Available at: https://owasp.org

[16] Zhou, Y., & Jiang, X. 2012. Dissecting Android malware: Characterization and evolution. IEEE Symposium on Security and Privacy, 95–109.

[17] Rashidi, B., & Fung, B. C. M. 2016. Privacy-preserving data publishing: A survey on recent developments. ACM Computing Surveys, 51(4), 1–41.

[18] Android Open Source Project (AOSP). 2023. Android Security Overview. Available at: https://source.android.com

[19] Gartner Research. 2022. Mobile Application Security Trends Report.

[20] Stallings, W. 2018. Cryptography and Network Security: Principles and Practice. Pearson Education.