Vol. 3 No. 1 (2025)

Spectrum of Engineering Sciences

SPECTRUM OF ENGINEERING SCIENCES

Online ISSN
3007-3138
Print ISSN
3007-312X

Exploring Innovations and Security Enhancements in Android Operating System

Muhammad Hassam Shakil[1]
Faculty of Engineering Science and Technology Iqra University, Karachi, 75500, Pakistan. mhd.hasm@gmail.com
Muhammad Daud Abbasi[2]
Faculty of Engineering Science and Technology Iqra University, Karachi, 75500, Pakistan. daud_abbas@hotmail.com
Muhammad Kashif Majeed[3]
Faculty of Engineering Science and Technology Iqra University, Karachi, 75500, Pakistan. School of Electronic Science and Technology, Xi'an Jiaotong University, Xi'an 710049, China. mkashif@iqra.edu.pk
Dr. Waleed Bin Yousuf[4]
Faculty of Engineering Science and Technology Iqra University, Karachi, 75500, Pakistan. waleed@iqra.edu.pk
Syed Muhammad Daniyal[5*]
Faculty of Engineering Science and Technology Iqra University, Karachi, 75500, Pakistan. Corresponding Auther Email: syed.daniyal@iqra.edu.pk
Usama Amjad[6]
Faculty of Engineering Science and Technology Iqra University, Karachi, 75500, Pakistan. usama.amjad@iqra.edu.pk

Abstract

The rapid growth of mobile phone usage over the past decade has led to increased demand for enhanced security, reliability, and performance, making mobile application security a critical concern. This study presents a freely available method for developing secure

Vol. 3 No. 1 (2025)

Spectrum of Engineering Sciences

SPECTRUM OF ENGINEERING SCIENCES

Online ISSN
3007-3138
Print ISSN
3007-312X

Android applications, addressing key security challenges that have not been comprehensively explored before. The research analyzes security vulnerabilities in Android applications and implements security-enhanced development strategies using various Android frameworks and techniques. Findings indicate that Android security remains a significant concern due to evolving cyber threats and increasing user dependency on mobile applications. Key security measures such as encryption, permission management, and secure coding practices are highlighted to mitigate security risks. The study also explores the real-world implementation of security-enhanced Android applications, demonstrating improved protection against unauthorized access and data breaches. Analyzing modern Android security frameworks reveals that integrating robust security measures during the development phase significantly enhances app reliability and user trust. Additionally, it discusses past security vulnerabilities in Android applications and best practices developers can adopt to prevent similar risks. By addressing these security challenges, this study provides a practical approach to developing secure applications, ensuring safer user experiences while meeting modern security demands.

Keywords: Android operating system, Security System, GAOS, Smartphones, Open Source Systems.

Introduction

Over the past decade, the utilization of technology and communication has transformed significantly, marked by the widespread shift from basic cell phones to advanced smartphones. This change can be ascribed to the increasing need for gadgets that can manage a variety of task-based, real-time apps that

Spectrum of Engineering Sciences
SPECTRUM OF ENGINEERING SCIENCES
Online ISSN
3007-3138
Print ISSN
3007-312X

outdated mobile phones were unable to deliver. Often called "feature phones," general mobile phones had limited capabilities and mostly provided text messaging and voice calling. Smartphones, on the other hand, have developed into multipurpose gadgets that are useful for social networking, productivity, entertainment, navigation, and more in addition to facilitating communication [1]. Smartphones are becoming essential due to their enormous range of features. In addition to their technological capabilities, smartphones are attractive due to the wide variety of applications they support. Often, these applications are categorized based on the operating systems (OS) of the devices.

Operating systems for smartphones serve as the device's cornerstone, determining its functionality and user interface. Of the several smartphone operating systems, Android, iOS, Windows, and RIM (BlackBerry) OS have been the main contenders. However, over time, Android has emerged as the most widely used and well-liked operating system, capturing a significant portion of the global smartphone market. Android's open-source status, versatility, and the large number of apps available through the Google Play Store are just a few of the factors that make it so appealing. Android currently holds a sizable lead in the smartphone market, accounting for about 58% of sales in the last quarter of 2011 alone [2]. Due to Android's open-source nature, developers from all over the world can participate in its ecosystem and create a vast array of applications that cater to nearly any need. Unlike proprietary systems like Apple's iOS, Android offers an open platform for innovation, making it an appealing choice for both developers and customers. Another major factor in Android's

Vol. 3 No. 1 (2025)
Spectrum of Engineering Sciences
Online ISSN
3007-3138
Print ISSN
3007-312X

success is its Linux kernel core, which provides a reliable, secure, and efficient environment for doing out sophisticated apps. The Linux kernel is well known for its stability and performance, two qualities that are essential for smartphones that must multitask. It also provides enhanced security features that ensure user information is protected from weaknesses and unauthorized access [3]. Based on this solid base, the Android OS leverages Linux's performance advantages and offers an intuitive interface that appeals to a diverse user base around the globe. The Android ecosystem has grown as a result of both the consumer market and the thriving developer community, which creates apps for Android devices. Compared to previous operating systems, the number of open-source applications being developed for Android has increased dramatically. These apps, which include everything from games and entertainment to productivity and health features, are usually free or reasonably priced, making them very accessible to users everywhere. As a result, Android users can choose from a nearly limitless selection of apps that are tailored to their individual needs [4].

Developers are always coming up with new and creative methods to utilize Android's many advantages because of this open ecosystem. However, developing apps for Android is not without its challenges. It requires a solid understanding of programming, particularly Java, which is the primary language used to create Android applications. Java's versatility and large array of APIs make it an excellent tool for developing feature-rich, high-performing Android apps. As the Android ecosystem expands, developers must stay up to date on the latest Java and Android SDK (Software Development Kit) advancements, as well as new

Spectrum of Engineering Sciences
SPECTRUM OF ENGINEERING SCIENCES
Online ISSN
3007-3138
Print ISSN
3007-312X

APIs and tools that are made available with each OS version. Because Android development is always evolving, there will always be new opportunities for creativity and a need for skilled developers who can turn creative ideas into reality [5].

In conclusion, Android has become the market leader in smartphones thanks to its open-source nature, versatility, and abundance of applications. The Android development environment continues to attract developers from around the globe because of its Linux kernel foundation, which ensures performance and security. As Android advances and expands, its impact on the global smartphone market and how people utilize technology is only expected to increase. For developers, the Android platform offers an exciting and rewarding environment with many opportunities to create innovative apps that cater to a broad spectrum of users.

The remainder of the document is structured as follows: The core idea of Android OS is covered in Section 2. A development environment, such as the Eclipse IDE, Android Development Tools (ADT), which includes the ADT Plugin, and the Android Software Development Kit (SDK), is required for creating any Android applications as discussed in Section 3. Section 4 further discusses the general security that Android adheres to in its architecture, while Section 5 concludes this article by discussing several security-related topics.

Objectives and Novelty Statement

This study aims to analyze security vulnerabilities in Android applications and their impact on user data protection while developing a structured approach for building secure mobile applications using best security practices. It evaluates encryption

Spectrum of Engineering Sciences
SPECTRUM OF ENGINEERING SCIENCES
Online ISSN
3007-3138
Print ISSN
3007-312X

techniques, permission management, and secure coding practices to provide a practical framework for mitigating security risks in Android applications. By integrating robust security measures, the study seeks to enhance user trust and ensure safer mobile experiences. The novelty of this research lies in presenting a structured and freely available approach to Android application development with a strong emphasis on security, addressing critical security challenges that have not been comprehensively explored before. By incorporating advanced security mechanisms, this study provides a practical solution to safeguard Android applications from unauthorized access and data breaches, contributing to the development of more secure mobile ecosystems.

Background Study

Android Operating System

Based on the Linux environment, Android is an operating system made mostly for mobile devices. The middleware, libraries, and Application-Programming Interface (API) of Android are all written in C, and the project is open-source under the Apache license [6].

Table 1: Different Variants of Android

Name	Version Number(s)	Initial Stable Release Date	API Level
No official codename	1.0	1.0	1
No official codename	1.1	1.1	2
Cupcake	1.5	1.5	3
Donut	1.6	1.6	4
Eclair	2.0 – 2.1	2.0 – 2.1	5 – 7

Spectrum of Engineering Sciences
SPECTRUM OF ENGINEERING SCIENCES
Online ISSN
3007-3138
Print ISSN
3007-312X

Froyo	2.2 – 2.2.3	2.2 – 2.2.3	8
Gingerbread	2.3 – 2.3.7	2.3 – 2.3.7	9 – 10
Honeycomb	3.0 – 3.2.6	3.0 – 3.2.6	11 – 13
Ice Cream Sandwich	4.0 – 4.0.4	4.0 – 4.0.4	14 – 15
Jelly Bean	4.1 – 4.3.1	4.1 – 4.3.1	16 – 18
KitKat	4.4 – 4.4.4	4.4 – 4.4.4	19 – 20
Lollipop	5.0 – 5.1.1	5.0 – 5.1.1	21 – 22
Marshmallow	6.0 – 6.0.1	6.0 – 6.0.1	23
Nougat	7.0 – 7.1.2	7.0 – 7.1.2	24 – 25
Oreo	8.0 – 8.1	8.0 – 8.1	26 – 27
Pie	9	9	28
Android 10	10	10	29
Android 11	11	11	30
Android 12	12	12	31
Android 13	13	13	32
Android 14	14	14	33
Android 15	15	15	34

Android's native API is available on C/C++ platforms, allowing developers to create apps that outperform Java apps. Android provides this feature through the NDK, which is a collection of all the libraries needed to construct Android applications. Despite this, Java is preferred by over 85% of Android developers over C/C++ due to its superior modularity and a host of additional characteristics that C/C++ cannot match [7]. In 2005, Google Inc. published the initial version of Android. Since then, the platform has grown, with multiple versions shown in Table I. This timeline describes the development of Google's popular mobile platform, the Android operating system. Each version, which reflects new

Spectrum of Engineering Sciences
SPECTRUM OF ENGINEERING SCIENCES
Online ISSN
3007-3138
Print ISSN
3007-312X

features and technological developments, is identified by its codename, version number, first stable release date, and API level.

Versions 1.0 and 1.1 of Android were the first to be released without official codenames [8]. Android 1.0, which was released on September 23, 2008, brought basic smartphone functionality. Version 1.1, which was released in February 2009, added a few small improvements. Android began naming versions after desserts in alphabetical order in April 2009, with Cupcake (1.5) being the first release. Cupcake enhanced user interaction by introducing on-screen keyboards. Later iterations, such as Doughnut (1.6) and Eclair (2.0–2.1), which were released in 2009, added turn-by-turn navigation, support for a variety of screen resolutions, and live wallpapers. Released in May 2010, Froyo (2.2) added Wi-Fi tethering and improved performance. By December 2010, Gingerbread (2.3) included an enhanced user interface with NFC functionality. With its emphasis on large screens and multitasking, Honeycomb (3.0–3.2) was the first tablet-only release in 2011. Ice Cream Sandwich (4.0), released later that year, and combined the user interfaces of tablets and smartphones [9]. In 2012, Jelly Bean (4.1–4.3) was released, with a focus on Google Now, alerts, and speed enhancements. With the 2013 release of KitKat (4.4), a lighter design for entry-level devices was launched.

Following the release of Lollipop (5.0–5.1) in November 2014, which introduced Material Design for a consistent visual experience, Marshmallow (6.0) in 2015 delivered Doze mode and app permissions for improved battery management. While Oreo (8.0–8.1) in 2017 enhanced security and background process optimization, Nougat (7.0–7.1) in 2016 extended multi-window capability. 2018 saw the release of Pie (9.0), which included

Vol. 3 No. 1 (2025)
Spectrum of Engineering Sciences
SPECTRUM OF ENGINEERING SCIENCES
Online ISSN
3007-3138
Print ISSN
3007-312X

adaptive battery capabilities and gesture navigation. Google switched from names with dessert themes to numerical identifiers with Android 10 in 2019 [10]. This change made branding simpler, while Android 10 included better privacy settings and system-wide dark mode. While Android 12 in 2021 concentrated on UI customization through Material You, Android 11, which was released in 2020, improved communications and media controls. The 2022 release of Android 13 brought more precise controls over app permissions, while the 2023 release of Android 14 improved accessibility and system upgrades. Android's ongoing development is reflected in each version, as new features and enhanced developer tools are introduced at higher API levels while preserving backward compatibility.

Android Architecture

As illustrated in Figure 1, the basic design of an Android OS is separated into five main parts.

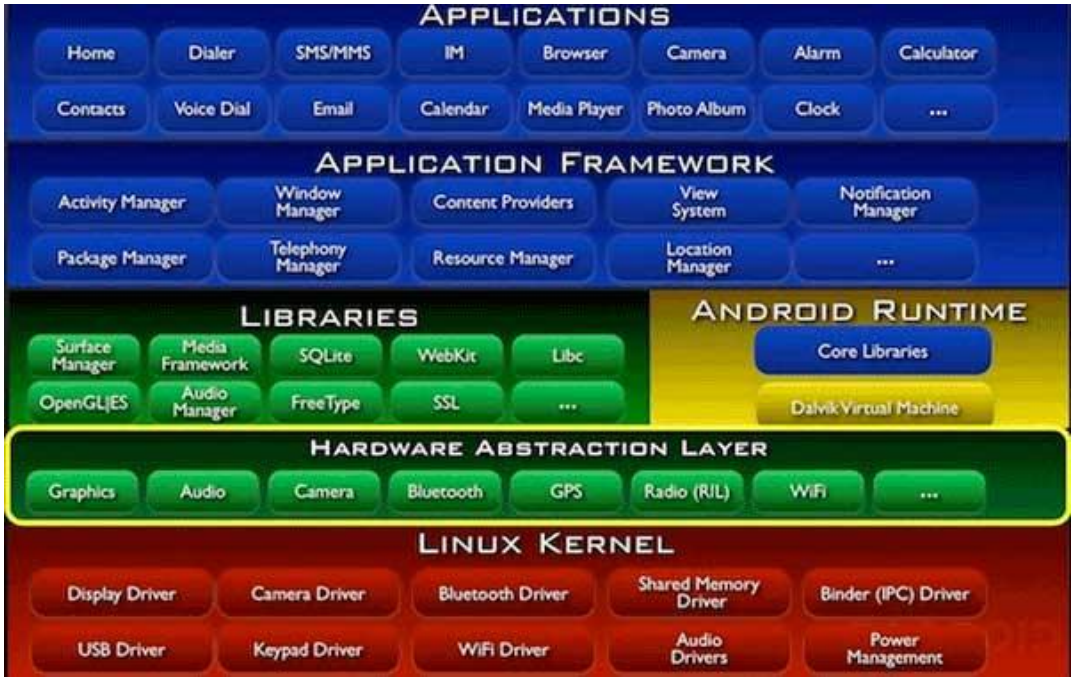

Figure 1. Components of the Android Architecture [11]

Spectrum of Engineering Sciences
SPECTRUM OF ENGINEERING SCIENCES
Online ISSN
3007-3138
Print ISSN
3007-312X

Applications, application frameworks, libraries, the Android runtime, and the Linux kernel make up the Android architecture:

Applications

An individual ".apk" file that can be installed, which can be installed on any Android smartphone and contains the entire software, contains all of the development framework's components [12].

Application Framework

Development components and an open platform that enables the creation of any application utilizing an additional API that the main application uses will be included. This framework includes numerous managers, such as Notification Area, Activity M, Window M, Resource M, Telephony M, Package M, Content M, etc. Consequently, an underlying layer functions following the completion of any program's actual development [13].

Libraries

Android's vast library, which enables programmers to construct any kind of Android application, is another well-known feature. This layer contains a core library, media library, web kit, SQLite library, and other resource packages together with their library files [14].

Android Runtime

Since each Android process needs memory to execute, the Android OS has assigned a distinct runtime for every task. Dalvik Virtual Machine, or DVM for short, is at the heart of all such operations [15].

Linux Kernel

Linux's activity management system and security are its main characteristics. Better memory management, process management, and many other advantages are inherent to Linux. The "kernel," the

Spectrum of Engineering Sciences
SPECTRUM OF ENGINEERING SCIENCES
Online ISSN
3007-3138
Print ISSN
3007-312X

abstract layer of any system, serves as a conduit for communication between software and hardware resources [16].

Related Work

Development Environment

The Development Environment is the initial phase in the process of developing an application. To create an Android application, programmers most frequently use the Eclipse Classic version. Eclipse Classic is an Android IDE or integrated development environment. The components listed below must be purchased from the open market to configure the IDE [17].

Open-Source vs. Closed-Source Development Environments

Development environments can broadly be classified into open-source and closed-source ecosystems, each with distinct advantages and limitations. An open-source development environment provides transparency, flexibility, and community-driven innovation, while a closed-source environment ensures stability, security, and official support. Understanding these differences is crucial for developers in selecting the right environment based on project requirements. In an open-source development environment, the source code is freely available, allowing developers to modify, distribute, and contribute to its improvement. Examples include Android Studio**,** Eclipse, and Visual Studio Code. These platforms benefit from active developer communities, frequent updates, and extensive plugin support. Open-source environments enable customization, making them ideal for developers seeking control over tools and frameworks. However, they may require additional effort for configuration and troubleshooting, as official support is often limited to community-driven forums and documentation.

Spectrum of Engineering Sciences
SPECTRUM OF ENGINEERING SCIENCES
Online ISSN
3007-3138
Print ISSN
3007-312X

Conversely, a closed-source development environment is proprietary software where the source code is restricted and maintained by a specific company. Examples include Apple's Xcode**,** Microsoft Visual Studio (Enterprise Edition)**,** and JetBrains IntelliJ IDEA (Ultimate Edition)**.** These environments offer robust security, professional support, and seamless integration with proprietary technologies. However, they come with licensing costs and limited customization options, making them less flexible for developers who prefer modifying or extending functionalities. While open-source environments promote innovation and collaboration, closed-source environments ensure reliability and structured development. The choice between the two depends on factors such as project scope, budget, required security measures, and the need for customization. A hybrid approach, where developers use open-source tools alongside closed-source solutions, is often an effective strategy for balancing flexibility and stability in software development.

Android SDK

Eclipse IDE is exclusively for Java-oriented development, but the combination of tools makes it possible for Android, even though Android is a Java platform language. For this reason, the Android Software Developer Kit SDK is necessary [18].

Android Development Tools Plugin

The software development kit aids in the creation of the Android coding environment, but other tools are needed to modify and optimize the code that has been developed on the Android device. These tools are referred to as Android Development Tools (ADT) Eclipse plugins. They aid in optimizing the framework, essential resources, and Android code [19].

Vol. 3 No. 1 (2025)
Spectrum of Engineering Sciences
SPECTRUM OF ENGINEERING SCIENCES
Online ISSN
3007-3138
Print ISSN
3007-312X

Android Emulator

To check the code's output, performance, KPIs, errors, bugs, latency issues, etc., an emulator is required. The configuration of this Android virtual phone is identical to that of any actual phone. Any Android application may be tested and debugged with this emulator [20]. The emulator is a built-in component of the AVD manager tool. These AVDs also have certain limitations, such as no camera, Bluetooth, and the inability to receive calls or messages. This plugin enhances our code before we test it on a real emulator by operating on the back end of Eclipse. The virtual device cannot be used for these kinds of tasks. An Eclipse IDE's built-in Check style plugin is a very helpful open-source tool that highlights code errors for programmers [21].

Eclipse IDE

As mentioned in earlier sections, Android is a Java-based programming language. Therefore, the Java development toolkit is required for the system to run the developed application. Because this JDK is open-source and integrated with the Eclipse IDE, anyone can use it for free [23-24].

Security of Android

There are two types of development environments: closed-source and open-source. Since Android is an open-source operating system, anyone can use its code for anything. This makes it very challenging to maintain security on the development side. Any background-running application that can retrieve user data from a mobile device and move it to a server-side database can be made by developers. Some of these apps impact the mobile environment and cause problems for users' apps [25]. The following features make it simple for developers to steal user data. Therefore, users

Spectrum of Engineering Sciences
SPECTRUM OF ENGINEERING SCIENCES
Online ISSN
3007-3138
Print ISSN
3007-312X

should be aware of these requirements while installing any new application.

Rpackage Application

Because the malicious software is hidden inside any application and seems like the normal application interface, users can install it without noticing. One well-known example is the Trojan horse [26].

Receiving Unknown Commands

Any little information can be sent from the server to the Android device thanks to the pushback service that Android provides. On the other hand, malicious commands can be sent via the internet using the malware server. Therefore, a user should not authorize such orders or alerts to have an impact on the system if they are not aware of them [27].

Steal Information

Users' data will be stolen by numerous malicious programs and uploaded to the malware server. GSM phone number, GPS position, Android device model, and SDK version, Examples of such data include International Mobile Subscriber Identity (IMSI) number, International Mobile Equipment Identity (IMEI) number, and other resource information [28]. For their security, users must therefore be aware of these requirements. Never accept any notice or agreement from unidentified applications. With millions of apps available, the Android market is well-known for allowing developers to release their apps without authentication. An application that is not authorized or licensed for use is known as a third-party application. A user may steal the aforementioned data if they install this kind of application. A lot of people link their Android devices to the internet constantly. Users may experience a virus problem if they are unaware of the guarded security layer.

Spectrum of Engineering Sciences
SPECTRUM OF ENGINEERING SCIENCES
Online ISSN
3007-3138
Print ISSN
3007-312X

Viruses can occasionally accompany our actual data when we browse the internet or download files [29]. To download such content, always choose a secure website that offers superior security, as seen in Figure. 2.

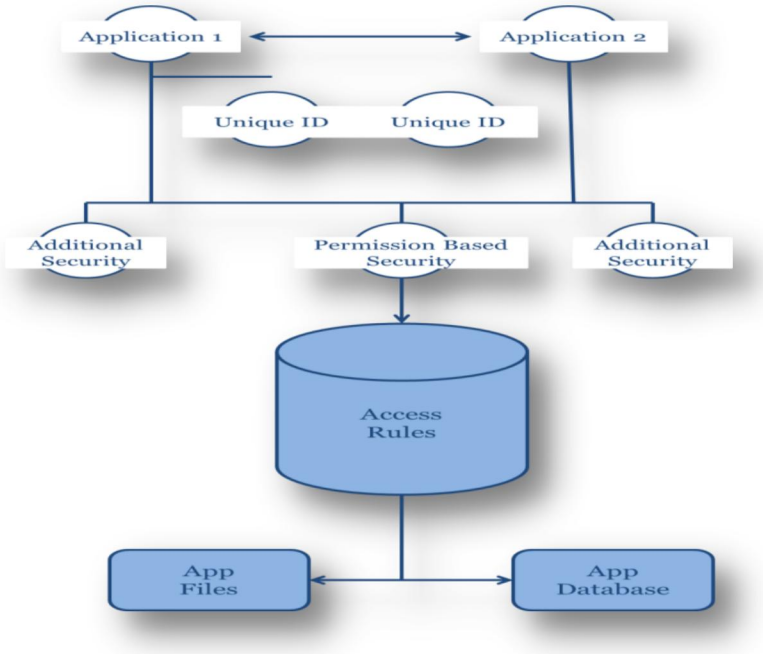

Figure 2. The Security Architecture of Android

In an Android application development environment, certain components are essential for compiling, running, and debugging applications, while others are optional tools that enhance development efficiency and code quality. Understanding the distinction between mandatory components and optional tools helps developers set up a functional workspace while integrating additional tools based on their specific needs. Mandatory components are the fundamental requirements for Android development. The Java Development Kit (JDK) is essential for compiling Java-based Android applications and running necessary development tools. The Android Software Development Kit (SDK)

Vol. 3 No. 1 (2025)
Spectrum of Engineering Sciences
SPECTRUM OF ENGINEERING SCIENCES
Online ISSN
3007-3138
Print ISSN
3007-312X

provides essential APIs, libraries, and tools required for building, testing, and debugging Android applications. Android Studio, the official Integrated Development Environment (IDE), offers a complete suite of development tools, including code editors, debugging tools, and design interfaces. Additionally, the Gradle Build System is crucial for automating project builds, managing dependencies, and compiling applications efficiently. Lastly, an Android Emulator or a physical device is required to test applications and ensure they function correctly before deployment. On the other hand, optional tools enhance productivity and code maintainability but are not strictly required for basic development. The CheckStyle Plugin helps maintain coding standards by analyzing Java code for style violations. Similarly, the Lint Tool detects potential performance, usability, and security issues, improving overall code quality. Version Control Systems like Git, GitHub, GitLab, and Bitbucket aid in source code management, collaboration, and version tracking.

Additionally, Firebase provides backend services such as authentication, real-time database management, and cloud messaging, offering extended functionalities for app development. Developers also use third-party libraries and frameworks, such as Retrofit for networking, Glide for image loading, and Dagger for dependency injection, to enhance application capabilities. By distinguishing between mandatory components and optional tools, developers can prioritize setting up essential elements for development while selectively incorporating additional tools to optimize productivity, security, and maintainability.

Spectrum of Engineering Sciences
SPECTRUM OF ENGINEERING SCIENCES
Online ISSN
3007-3138
Print ISSN
3007-312X

Manifest .xml File

When creating an Android application, this is the fundamental file that is included. The user is asked to approve this file, which includes all the rights required for Android smartphones. The structure of the permissions and how they are used in the application are defined by the tags included in this extensible markup language file. In general, the following permissions are available to developers [30]:

- Android.permission.READ_CONTACTS: Provides the ability to view the contact list on the device.
- Android.permission.WRITE_CONTACTS: Permits changes to the list of contacts.
- Android. permission.SEND_SMS: Enables SMS message transmission.
- Android.permission.WRITE_SMS: Permits the device to edit or save SMS messages.
- Android.permission.Receiving incoming SMS messages is made possible by RECEIVE_SMS.
- Android.permission.Access to read stored SMS messages is provided via READ_SMS.
- Android.permission.INTERNET: Makes the device's internet connection accessible.
- Android.permission.BLUETOOTH: Enables Bluetooth functionality.
- Android.permission.READ_PHONE_STATE: Enables you to see the phone's current status, such as idle or in-call.
- Android.permission.Permission to write data to external storage is granted by WRITE_EXTERNAL_STORAGE.
- Android.permission.Access to read files from external storage

Spectrum of Engineering Sciences
SPECTRUM OF ENGINEERING SCIENCES
Online ISSN
3007-3138
Print ISSN
3007-312X

is made possible by READ_EXTERNAL_STORAGE.

Discussion

Android Security Updates and Malware Prevention

A major concern in open-source development, particularly in Android, is security. Since Android is an open-source operating system, it is more susceptible to security vulnerabilities compared to closed-source ecosystems like Apple's iOS. To address these risks, Google has implemented several security measures, including regular Android security updates**,** Google Play Protect**,** and various built-in security mechanisms to prevent malware and unauthorized access.

Google releases monthly security updates to fix vulnerabilities and improve device security. These updates are essential for protecting Android devices from emerging threats such as malware, phishing attacks, and data breaches. However, the fragmentation of the Android ecosystem means that manufacturers may delay or skip these updates, leaving devices vulnerable.

To further enhance security, Google Play Protect plays a crucial role in safeguarding Android users from malicious applications. It is an integrated security service that continuously scans apps available on the Google Play Store and installed on users' devices for potential malware. By leveraging machine learning and behavioral analysis, Google Play Protect identifies suspicious activities, automatically removes harmful apps, and alerts users about security risks. This proactive security feature helps mitigate the impact of malicious applications that might bypass initial screening during app submission.

Spectrum of Engineering Sciences
SPECTRUM OF ENGINEERING SCIENCES
Online ISSN
3007-3138
Print ISSN
3007-312X

Additionally, Android incorporates sandboxing, permission management, and encryption to enhance security. App sandboxing isolates applications from each other and from the system, preventing unauthorized access to sensitive data. Enhanced permission controls allow users to grant or deny app access to specific features, reducing the risk of privacy breaches. Furthermore, full-disk encryption and secure boot mechanisms help protect user data against unauthorized access and device tampering. While open-source development offers flexibility and innovation, security remains a critical challenge. By implementing strict security updates, leveraging Google Play Protect, and enforcing robust security policies, Android maintains a balance between openness and protection, ensuring a safer ecosystem for both developers and users.

Conclusion

Due to improvements in smartphone security and applications, the working economics of cellular phones have increased dramatically in recent decades. In preparation for the most exciting use of extremely concrete mobile phones, modernization, and enhancement configuration are underway. Customers in this industry are expecting and demanding more from their mobile phones as the operation of mobile phones continues to intensify. As a result, there are many different types of physiognomies. Therefore, one of the leaders in the mobile phone industry considers the advancement in modernization to satisfy the needs of customers. This leader is the Google Android OS. The creation of Android applications is still in its infancy and is always evolving. Due to its numerous potential uses and necessity for satisfying the security standards of modern authenticity, it has been a mainstay

Spectrum of Engineering Sciences
SPECTRUM OF ENGINEERING SCIENCES
Online ISSN
3007-3138
Print ISSN
3007-312X

of research for the past few centuries. Regardless, the framework is quite unusual, making it difficult to create. Important details regarding the Android operating system and its application in the real-world market were covered in this article. Additionally, it illustrated and explained the development background components, as well as the various applications and accessories of the Android operating system, so that even the most inexperienced user can easily understand and grasp them. Additionally, the essential elements of Android security are discussed so that any newcomer will remember that point before installing or using such apps.

References

[1] Acharya, S., Rawat, U., & Bhatnagar, R. (2022). [Retracted] A Comprehensive Review of Android Security: Threats, Vulnerabilities, Malware Detection, and Analysis. Security and Communication Networks, 2022(1), 7775917.

[2] Garg, S., & Baliyan, N. (2021). Comparative analysis of Android and iOS from a security viewpoint. Computer Science Review, 40, 100372.

[3] Sikder, R., Khan, M. S., Hossain, M. S., & Khan, W. Z. (2020). A survey on android security: development and deployment hindrance and best practices. TELKOMNIKA (Telecommunication Computing Electronics and Control), 18(1), 485-499.

[4] Sharma, A., Singh, S. K., Kumar, S., Chhabra, A., & Gupta, S. (2021, September). Security of Android banking mobile apps: Challenges and opportunities. In International conference on cyber security, privacy and networking (pp. 406-416). Cham: Springer International Publishing.

Spectrum of Engineering Sciences
SPECTRUM OF ENGINEERING SCIENCES
Online ISSN
3007-3138
Print ISSN
3007-312X

[5] Mahor, V., Pachlasiya, K., Garg, B., Chouhan, M., Telang, S., & Rawat, R. (2021, December). Mobile operating system (Android) vulnerability analysis using machine learning. In International Conference on Network Security and Blockchain Technology (pp. 159-169). Singapore: Springer Nature Singapore.

[6] Alanda, A., Satria, D., Mooduto, H. A., & Kurniawan, B. (2020, May). Mobile application security penetration testing based on OWASP. In IOP Conference Series: Materials Science and Engineering (Vol. 846, No. 1, p. 012036). IOP Publishing.

[7] Yousuf, W. B., Talha, U., Abro, A. A., Ahmad, S., Daniyal, S. M., Ahmad, N., & Ateya, A. A. (2024). Novel Prognostic Methods for System Degradation Using LSTM. IEEE Access.

[8] Gowda, V. D., Prasad, K., Shekhar, R., Srinivas, R., Srinivas, K. N. V., & Lakineni, P. K. (2023, October). Development of a Real-time Location Monitoring App with Emergency Alert Features for Android Devices. In 2023 4th IEEE Global Conference for Advancement in Technology (GCAT) (pp. 1-8). IEEE.

[9] Abdullah, H., & Zeebaree, S. R. (2021). Android mobile applications vulnerabilities and prevention methods: A review. 2021 2nd Information Technology To Enhance e-learning and Other Application (IT-ELA), 148-153.

[10] Daniyal, S. M., Abbasi, M. M., Hussain, D., Amjad, U., Abro, A. B., & Naeem, M. (2024). A hybrid approach for simultaneous effective automobile navigation with DE and PSO. VAWKUM Transactions on Computer Sciences, 12(2), 01-15. Xie, Yu, and Attila Buchman. "A survey for Communication security of the embedded system." Carpathian Journal of Electronic & Computer Engineering 14, no. 2 (2021).

Spectrum of Engineering Sciences
SPECTRUM OF ENGINEERING SCIENCES
Online ISSN
3007-3138
Print ISSN
3007-312X

[11] Xie Y, Buchman A. A survey for Communication security of the embedded system. Carpathian Journal of Electronic & Computer Engineering. 2021 Dec 1;14(2).

[12] Garg, S., & Baliyan, N. (2021). Android security assessment: A review, taxonomy, and research gap study. Computers & Security, 100, 102087.

[13] Razgallah, A., Khoury, R., Hallé, S., & Khanmohammadi, K. (2021). A survey of malware detection in Android apps: Recommendations and perspectives for future research. Computer Science Review, 39, 100358.

[14] Musa, Hussam Saeed, et al. "Survey on blockchain-based data storage security for Android mobile applications." Sensors 23.21 (2023): 8749.

[15] Munjal, G., Chakravarti, A., & Sharma, U. (2024). Malware Analysis Techniques in Android-Based Smartphone Applications. Applying Artificial Intelligence in Cybersecurity Analytics and Cyber Threat Detection, 87-105.

[16] Kunda, D. and Chishimba, M., 2021. A survey of Android mobile phone authentication schemes. Mobile Networks and Applications, 26(6), pp.2558-2566.

[17] Garg, Shivi, and Niyati Baliyan. "Android security assessment: A review, taxonomy, and research gap study." Computers & Security 100 (2021): 102087.

[18] Garg S, Baliyan N. Android security assessment: A review, taxonomy, and research gap study. Computers & Security. 2021 Jan 1;100:102087.

[19] Y Sathwara, B., & Singh Mann, P. (2024). A Hybrid Approach Based On Boosting Algorithm For Effective Android Malware

Spectrum of Engineering Sciences
SPECTRUM OF ENGINEERING SCIENCES
Online ISSN
3007-3138
Print ISSN
3007-312X

Detection. International Journal of Computing and Digital Systems, 15(1), 1-11.

[20] Shrivastava, G., Kumar, P., Gupta, D. and Rodrigues, J.J., 2020. Privacy issues of Android application permissions: A literature review. Transactions on Emerging Telecommunications Technologies, 31(12), p.e3773.

[21] Musa, Hussam Saeed, Moez Krichen, Adem Alpaslan Altun, and Meryem Ammi. "Survey on blockchain-based data storage security for Android mobile applications." Sensors 23, no. 21 (2023): 8749.

[22] Sihag V, Vardhan M, Singh P. A survey of Android application and malware hardening. Computer Science Review. 2021 Feb 1;39:100365.

[23] Сентюрін, Є. Є., & Мельник, М. Б. (2024). Android development (Doctoral dissertation, ВНТУ).

[24] Sutter, T. Apps in Android's Security Landscape. The Salto Project: Static Analysis of OCaml Programs by Abstract Interpretation, 17.

[25] Mithun, M., & Chandran, S. (2024, April). AndroPack: A Hybrid Method To Detect Packed Android Malware With Ensemble Learning. In 2024 12th International Symposium on Digital Forensics and Security (ISDFS) (pp. 01-04). IEEE.

[26] Kumar, P., & Singh, S. (2024). An efficient security testing for android application based on behavior and activities using RFE-MLP and ensemble classifier. Multimedia Tools and Applications, 1-30.

[27] Abdul Kadir, A. F., Habibi Lashkari, A., & Daghmehchi Firoozjaei, M. (2024). Android Operating System. In Understanding

Spectrum of Engineering Sciences
SPECTRUM OF ENGINEERING SCIENCES
Online ISSN
3007-3138
Print ISSN
3007-312X

Cybersecurity on Smartphones: Challenges, Strategies, and Trends (pp. 25-42). Cham: Springer Nature Switzerland.

[28] Sahani, R. K., Anand, M., Tagore, A. B., Mehrotra, S., Tabassum, R., & Raja, S. P. (2024). Android malware analysis using multiple machine learning algorithms. International Journal of Electronic Security and Digital Forensics, 16(6), 752-774.

[29] Ozturk, M., Yilmaz, B., Arslan, Z., & Demirbas, A. (2024). An Effective Strategy for Ransomware Mitigation on Android Devices via Android OS File System API.

[30] Khan, A. A., Laghari, A. A., Kumar, A., Shaikh, Z. A., Baig, U., & Abro, A. A. (2023). Cloud forensics-enabled chain of custody: a novel and secure modular architecture using Blockchain Hyperledger Sawtooth. International Journal of Electronic Security and Digital Forensics, 15(4), 413-423.