International Journal of Engineering Research & Management Technology

ISSN: 2348-4039

March-April-2023 Volume 10, Issue-2

Email: editor@ijermt.org www.ijermt.org

ADVANCED TECHNIQUES FOR SECURITY ENHANCEMENT IN MOBILE APPLICATIONS

Ngangom James Singh, Research Scholar, Department Computer Applications,

Maharaja Agrasen Himalayan Garhwal University

Dr. Sayed Mohd. Saqib, Assistant Professor, Department Computer Applications, Maharaja Agrasen Himalayan Garhwal University

Abstract

As mobile applications become increasingly prevalent, enhancing their security is paramount. This paper investigates advanced techniques for security enhancement in mobile applications, focusing on secure coding practices, encryption methods, multi-factor authentication, and blockchain technology. The objective is to provide a comprehensive overview of how these techniques can be implemented to safeguard mobile applications against emerging threats.

Introduction

The rapid growth of mobile applications has brought about significant security challenges. Enhancing the security of these applications is crucial to protect sensitive user data and maintain trust. This paper explores various advanced techniques for security enhancement, examining their effectiveness, implementation challenges, and future directions.

Privacy and security are also paramount concerns for users in the digital age. With increasing incidents of data breaches and cyber-attacks, users are becoming more cautious about the security of their personal information. iOS is often lauded for its robust security measures, including end-to-end encryption and stringent app permissions. Android, while offering extensive customization options, has also made significant improvements in its security features, such as regular security updates and enhanced privacy controls.

The debate over which mobile operating system is the best continues, with each platform having its ardent supporters. For end-users, particularly those who are new to smartphones, making an informed decision can be challenging. Manufacturers often highlight the strengths of their OS while downplaying any drawbacks, making it essential for users to conduct their own research and consider their specific needs before choosing a platform.

In the Android platform has achieved remarkable success by combining a flexible and powerful architecture with a commitment to security, user control, and backward compatibility. The major API levels discussed in this dissertation illustrate the platform's ongoing evolution and its ability to adapt to the changing needs of users and developers. By studying these API levels, we gain insight into the key innovations and design principles that have made Android a dominant force in the mobile industry. Mobile applications are become an essential aspect of everyday life in the current digital era, offering consumers easy access to a variety of services including social networking, entertainment, and banking. The need to ensure the security of mobile applications is growing along with the dependence on them. Because mobile applications manage sensitive personal and financial information, they are typically the target of cybercriminals. Therefore, in order to safeguard users and their data from possible attacks, it is essential to adopt modern approaches for security evaluation and enhancement in mobile apps.

Developers and security experts must negotiate a massive ecosystem as a result of the growth of mobile apps brought about by the widespread use of mobile devices.

Every program, whether it is made for iOS, Android, or another platform, has different security issues. Because mobile platforms are open and there is a wide range of hardware and software combinations, security must be approached from several angles. Even while they are still important, traditional security measures are unable to counter the sophisticated tactics that fraudsters use nowadays. Thus, the creation of sophisticated security methods has become essential for protecting mobile apps.

Android is a versatile operating environment based on the Linux kernel, structured as a layered system. Its architecture encompasses several integral components, each contributing to its robustness and flexibility. The topmost layer, the application layer, forms the user interface (UI) of all Android applications, including Email, SMS, GPS, Web Browser, and others. These applications are developed using Java programming language and Java APIs, which is a testament to Android's compatibility and flexibility. Central to the Android ecosystem is the application framework, which provides essential services and capabilities to application developers.

The Android application framework is replete with components designed to streamline app development and enhance user experience. A rich set of Views is one such component, enabling developers to create vibrant and interactive UIs. This set includes lists, grid views, input boxes, buttons, and even an integrated browser, which can be seamlessly incorporated into applications. Another pivotal component is the Content Providers, which allow apps to access and share data with other applications. This capability fosters a highly interconnected app ecosystem, where data can be leveraged across multiple applications, enhancing functionality and user experience.

The Resource Manager is another critical element of the Android framework. It provides access to various resources such as strings and layouts, which are essential for creating consistent and adaptable UIs. The Notification Manager plays a crucial role in user engagement, enabling applications to display custom alarms and notifications in the status bar, ensuring users are kept informed of important events and updates. Lastly, the Activity Manager is indispensable for managing the lifecycle of applications. It provides a standardized navigation model, ensuring a coherent and intuitive user experience across different apps.

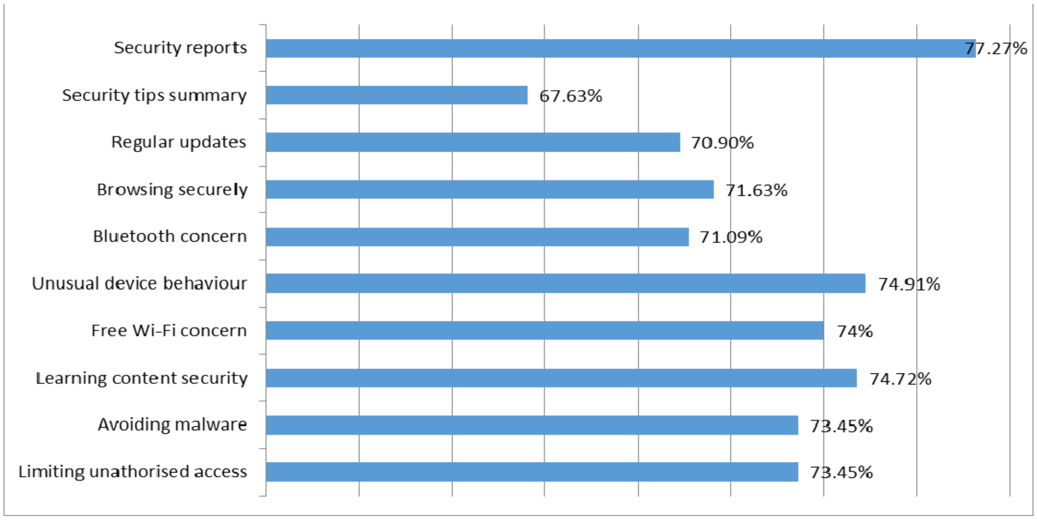

Fig 1: Different security features.

"Practical Mobile Forensics" by Heather Mahalik (2017)

Practical Mobile Forensics" by Heather Mahalik, published in 2017, is a comprehensive guide designed to equip readers with the skills and knowledge required to effectively perform forensic investigations on mobile devices. The book is structured to cater to both beginners and experienced forensic practitioners, providing a step-by-step approach to mobile forensics while covering a wide range of topics pertinent to the field. The text delves into the technical intricacies of mobile devices, the tools necessary for forensic analysis, and the methodologies for extracting and analyzing data, all while emphasizing practical application.

The initial chapters of the book lay the foundation by introducing readers to the basics of mobile forensics. Mahalik starts with an overview of mobile device architecture, explaining the hardware and software components that constitute modern smartphones and tablets. This foundational knowledge is crucial as it helps readers understand the complexities involved in mobile forensics and the various points of data extraction. The book also covers the different operating systems used in mobile devices, such as iOS, Android, and Windows Phone, highlighting the unique challenges and opportunities each platform presents for forensic investigators.

One of the core strengths of "Practical Mobile Forensics" is its focus on the forensic process, which Mahalik breaks down into manageable stages. She emphasizes the importance of proper evidence handling to ensure the integrity of data. The book discusses the chain of custody, acquisition methods, and the significance of maintaining a documented procedure throughout the forensic investigation. This meticulous approach is vital for ensuring that the findings are admissible in court and that the investigation withstands legal scrutiny.

Data acquisition is a critical aspect of mobile forensics, and Mahalik dedicates significant attention to this topic. She explains various acquisition techniques, including physical, logical, and file system extractions. Physical extraction involves obtaining a bit-by-bit copy of the device's storage, allowing investigators to access deleted and hidden data. Logical extraction focuses on retrieving active data such as contacts, messages, and application data. File system extraction lies between physical and logical, providing a comprehensive view of the file system without requiring a complete physical image. Mahalik details the tools and methods for each type of extraction, providing practical examples and case studies to illustrate the concepts.

The book also explores the challenges associated with data acquisition, such as dealing with locked devices, encrypted data, and the rapid evolution of mobile technology. Mahalik offers solutions and best practices for overcoming these obstacles, ensuring that investigators can retrieve the necessary information despite these hurdles. She also addresses the importance of keeping forensic tools and techniques up-to-date to cope with the continuous advancements in mobile technology.

Once the data is acquired, the next step is data analysis, which Mahalik covers extensively. She explains how to analyze various types of data, including call logs, messages, emails, multimedia files, and application data. The book provides detailed instructions on how to use forensic tools to parse and interpret this data, highlighting the significance of context in understanding the evidence. Mahalik also discusses advanced analysis techniques, such as timeline analysis, which helps investigators reconstruct events by correlating data from different sources.

A significant portion of the book is dedicated to application analysis, recognizing that mobile apps are a treasure trove of information. Mahalik provides a thorough examination of popular apps, including social media, messaging, and navigation apps, showing how to extract and analyze data from these sources. She explains the artifacts generated by these apps and how they can be used to piece together user activities and behaviors. The book includes practical examples and screenshots to guide readers through the process, making it accessible even to those new to mobile forensics.

Mahalik also addresses the legal aspects of mobile forensics, underscoring the importance of adhering to legal and ethical standards. She discusses the legal framework governing digital evidence, including search and seizure laws, privacy concerns, and the admissibility of evidence in court. The book provides guidance on how to write forensic reports and present findings in a clear and concise manner, which is essential for communicating the results to legal professionals and stakeholders.

The book includes several case studies and real-world examples to illustrate the practical application of forensic techniques. These case studies provide insights into the complexities and nuances of actual forensic investigations, highlighting the importance of attention to detail and critical thinking. They also demonstrate how to adapt forensic methodologies to different scenarios, ensuring that readers can apply the knowledge gained from the book to their investigations.

In addition to the technical content, Mahalik emphasizes the importance of continuous learning and professional development in the field of mobile forensics. She encourages readers to stay informed about the latest trends, tools, and techniques by participating in professional communities, attending conferences, and pursuing certifications. The book includes a list of resources, including websites, forums, and training programs, to help readers continue their education and stay current in this rapidly evolving field.

"Practical Mobile Forensics" also features chapters dedicated to specific tools used in mobile forensics. Mahalik provides detailed tutorials on using popular forensic tools, such as Cellebrite UFED, Oxygen Forensic Suite, and Magnet AXIOM. These tutorials include step-by-step instructions, screenshots, and tips for maximizing the effectiveness of each tool. By covering a range of tools, the book ensures that readers are equipped with the knowledge to choose the right tool for their specific needs and to use it effectively.

The book concludes with a discussion on the future of mobile forensics, considering the emerging trends and technologies that will shape the field. Mahalik explores the impact of advancements in mobile security, the proliferation of Internet of Things (IoT) devices, and the increasing importance of cloud forensics. She highlights the challenges these developments pose for forensic investigators and offers strategies for staying ahead of the curve. The forward-looking perspective ensures that readers are not only prepared for current challenges but are also equipped to adapt to future changes in the field.

Overall, "Practical Mobile Forensics" by Heather Mahalik is an invaluable resource for anyone involved in mobile forensic investigations. Its comprehensive coverage, practical approach, and emphasis on real-world application make it an essential guide for both novice and experienced forensic practitioners. The book's detailed explanations, case studies, and tutorials provide readers with the knowledge and skills needed to conduct thorough and effective mobile forensics investigations, ensuring the integrity and admissibility of their findings.

Secure Coding Practices

Secure coding practices are fundamental to developing secure mobile applications. By adhering to best practices, developers can minimize vulnerabilities and improve the overall security of their applications.

1. Input Validation: Ensuring that all user inputs are properly validated helps prevent common attacks such as SQL injection and cross-site scripting (XSS).

2. Secure APIs: Using secure APIs for data transmission and communication between application components reduces the risk of data breaches and unauthorized access.

3. Code Obfuscation: Obfuscating the code makes it more difficult for attackers to reverse-engineer the application, protecting intellectual property and sensitive information.

Encryption Methods

Encryption is a critical technique for protecting data in transit and at rest in mobile applications.

1. End-to-End Encryption: Implementing end-to-end encryption ensures that data is encrypted from the sender to the receiver, preventing unauthorized access during transmission.

2. Data Encryption: Encrypting sensitive data stored on the device protects it from being accessed if the device is compromised.

3. Key Management: Securely managing encryption keys is essential to maintaining the integrity and confidentiality of encrypted data.

Multi-Factor Authentication (MFA)

Multi-factor authentication enhances the security of mobile applications by requiring multiple forms of verification before granting access.

1. Biometric Authentication: Utilizing biometric methods such as fingerprint or facial recognition provides a higher level of security compared to traditional passwords.

2. OTP (One-Time Password): Implementing OTPs for login and transaction verification adds an additional layer of security, reducing the risk of unauthorized access.

3. Push Notifications: Sending push notifications for authentication requests allows users to verify their identity securely and conveniently.

Blockchain Technology

Blockchain technology offers innovative solutions for enhancing the security of mobile applications, particularly in areas such as data integrity and secure transactions.

1. Decentralized Data Storage: Storing data on a blockchain ensures that it is tamper-proof and secure from unauthorized modifications.

2. Smart Contracts: Implementing smart contracts can automate security processes, such as access control and transaction verification, reducing the risk of human error and malicious activities.

3. Secure Transactions: Blockchain-based transactions provide a high level of security and transparency, making them ideal for financial applications and other use cases requiring secure data exchanges.

Security Frameworks and Standards

Adopting established security frameworks and standards is essential for enhancing the security of mobile applications.

1. OWASP Mobile Security Project: The OWASP Mobile Security Project provides guidelines and tools for developing secure mobile applications, covering various aspects of security assessment and enhancement.

2. NIST Guidelines: The National Institute of Standards and Technology (NIST) offers comprehensive guidelines for securing mobile applications, including best practices for encryption, authentication, and data protection.

3. GDPR and CCPA Compliance: Ensuring compliance with regulations such as the General Data Protection Regulation (GDPR) and the California Consumer Privacy Act (CCPA) is crucial for protecting user data and maintaining legal compliance.

Methodologies:

In reverse engineering plays a critical role in the implementation of Need-Based Security for Android applications. By decompiling and analyzing APK files, researchers and developers can identify and remove unnecessary permissions, enhance the security of applications, and protect users from potential threats. The process involves a detailed examination of the AndroidManifest.xml file, the application code, and resources, as well as the development of automated tools and user education initiatives. Through collaboration, ongoing research, and a focus on best practices, it is possible to create a more secure and trustworthy mobile application ecosystem. The work of M. A. Dar and Parvez provides a solid foundation for these efforts, and the proof of concept implementation outlined in their work offers a practical roadmap for enhancing mobile application security. By adopting the principles of Need-Based Security and leveraging reverse engineering techniques, developers can ensure that their applications are both safe and effective for users.

The next phase is assessing the potential harm and feasibility of removing each permission. This involves a risk-benefit analysis where the benefits of removing a permission are weighed against the possible negative impact on the app’s functionality. For instance, permissions that allow access to sensitive information like contacts or SMS can pose significant privacy risks. Even if an app justifies the need for such permissions, users may prefer to disallow them to protect their personal information. In such cases, developers must evaluate whether the app can offer alternative functionalities that do not compromise user privacy.

Some permissions, however, are deeply integrated into the app's operations. For example, a messaging app requires access to the user's contacts to facilitate communication. Removing such a permission would render the app unusable. Therefore, developers need to assess whether it is feasible to remove a permission without breaking the app. This assessment might involve restructuring the app's code to decouple it from the unwanted permissions or providing optional functionalities that users can enable or disable based on their preferences.

A systematic approach to permission removal can be broken down into several steps:

1. Identification : List all permissions requested by the app by examining the AndroidManifest.xml file.

2. Classification: Categorize permissions into necessary and unnecessary based on the app's core functionality.

3. Testing: Simulate app behavior without the unnecessary permissions to ensure it still functions correctly.

4. Risk Assessment: Evaluate the potential risks associated with each permission and prioritize their removal based on the severity of the risk.

5. Implementation: Modify the AndroidManifest.xml file to remove unnecessary permissions.

6. Verification: Test the modified app to ensure it operates correctly without the removed permissions and does not introduce new issues.

7. User Feedback: Incorporate user feedback to identify any permissions that may still be deemed unnecessary or overly intrusive.

During the identification phase, developers must meticulously document each permission requested in the AndroidManifest.xml file. This documentation serves as a reference for understanding the purpose and necessity of each permission. By classifying permissions into necessary and unnecessary categories, developers can prioritize their efforts on permissions that pose the highest risk to user privacy and security.

The testing phase involves creating controlled environments where the app operates without specific permissions. This can be achieved by modifying the AndroidManifest.xml file to exclude certain permissions and running the app to observe its behavior. Developers should conduct extensive testing across different scenarios and use cases to ensure that the app maintains its functionality without the removed permissions.

Challenges and Future Directions

Despite the availability of advanced security enhancement techniques, several challenges remain. These include the complexity of implementing security measures, the need for continuous updates to address emerging threats, and the potential for user resistance to security features. Future research should focus on developing more user-friendly security solutions, integrating advanced technologies such as AI and quantum encryption, and addressing the unique security challenges posed by emerging mobile application use cases.

Conclusion

Enhancing the security of mobile applications is critical in the face of evolving threats. By adopting secure coding practices, encryption methods, multi-factor authentication, and blockchain technology, developers can significantly improve the security of their applications. Continuous innovation and adherence to security frameworks and standards are essential to staying ahead of potential threats and ensuring the safety and reliability of mobile applications.

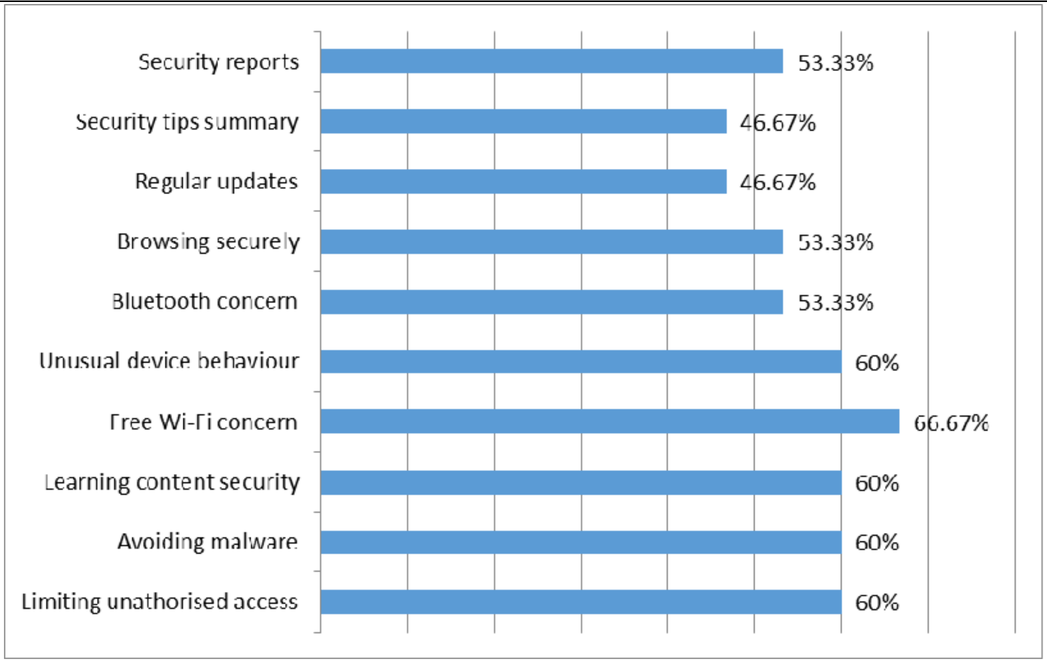

Fig 2: Security concerns of users.

Another critical component of our study involves educating users about the importance of security and privacy. Many novice users are unaware of the potential risks associated with granting app permissions, leading to a lax attitude towards security. We propose the development of comprehensive educational programs that teach users about the dangers of over-permissioning and the steps they can take to protect their data. These programs should be accessible and engaging, using a variety of media formats to reach a broad audience.

In addition to user education, it is essential to involve app developers in the effort to improve smartphone security. Developers should be encouraged to adopt best practices for secure coding and to minimize the number of permissions their apps request. We recommend the implementation of stricter guidelines for app permissions and more rigorous review processes for apps submitted to app stores. By holding developers accountable for the security of their apps, we can reduce the prevalence of over-permissioning and enhance the overall security of the Android ecosystem.

Our research also emphasizes the need for ongoing collaboration between researchers, developers, and users to address the evolving challenges of smartphone security. By fostering a community of practice, we can share knowledge, develop new solutions, and continually improve the security framework. This collaborative approach is essential for staying ahead of emerging threats and ensuring that the security measures in place are effective and up-to-date.

One of the innovative techniques we propose is the development of a layered security approach. This approach involves implementing multiple layers of security measures, each designed to address different aspects of smartphone security. For example, the first layer could involve enhancing the permission system to provide more granular control over app permissions. The second layer could focus on continuous monitoring and anomaly detection, using machine learning algorithms to identify suspicious behavior. The third layer could involve user education and awareness programs to ensure that users understand the importance of security and privacy.

Furthermore, our research highlights the potential benefits of integrating biometric authentication methods into the security framework. Biometric methods, such as fingerprint recognition and facial recognition, provide an additional layer of security by ensuring that only authorized users can access the device and its data. By combining biometric authentication with other security measures, we can create a more robust and secure environment for smartphone users.

In addition to these techniques, we also explore the potential of using blockchain technology to enhance smartphone security. Blockchain technology offers a decentralized and transparent approach to data security, making it more difficult for malicious actors to compromise the system. By leveraging blockchain, we can create a more secure and trustworthy environment for app transactions and data sharing.

Our study also examines the role of artificial intelligence (AI) in improving smartphone security. AI can be used to develop advanced threat detection systems that can identify and respond to security threats in real-time. By analyzing patterns of behavior and detecting anomalies, AI-powered systems can provide early warnings of potential security breaches and help mitigate risks before they escalate.
In conclusion, our research provides a comprehensive analysis of the current state of smartphone security, with a particular focus on the Android operating system. We have identified significant vulnerabilities in the existing security framework and proposed a series of novel techniques to address these issues. Our findings underscore the importance of user awareness, continuous monitoring,

developer accountability, and community collaboration in enhancing smartphone security. We believe that our study has the potential to significantly improve the state of practice in smartphone security and provide a safer digital environment for users worldwide. Through ongoing research and collaboration, we can continue to develop innovative solutions to address the evolving challenges of smartphone security and ensure that users can enjoy the benefits of mobile technology without compromising their privacy and safety.

References

1. OWASP. (2023). Mobile Security Project. Retrieved from [OWASP Mobile Security Project](https://owasp.org/www-project-mobile-security/)

2. SonarQube. (2023). Continuous Inspection. Retrieved from [SonarQube](https://www.sonarqube.org/)

3. Fortify. (2023). Application Security Solutions. Retrieved from [Fortify](https://www.microfocus.com/en-us/cyberres/application-security/fortify)

4. Checkmarx. (2023). Software Security Platform. Retrieved from [Checkmarx](https://www.checkmarx.com/)

5. OWASP ZAP. (2023). Zed Attack Proxy. Retrieved from [OWASP ZAP](https://www.zaproxy.org/)

6. Burp Suite. (2023). Web Vulnerability Scanner. Retrieved from [Burp Suite](https://portswigger.net/burp)

7. MobSF. (2023). Mobile Security Framework. Retrieved from [MobSF](https://mobexler.com/)

8. Metasploit. (2023). Penetration Testing Software. Retrieved from [Metasploit](https://www.metasploit.com/)

9. ProGuard. (2023). Java and Android Obfuscator. Retrieved from [ProGuard](https://www.guardsquare.com/en/products/proguard)

10. DexGuard. (2023). Android Protection Suite. Retrieved from [DexGuard](https://www.guardsquare.com/en/products/dexguard)

11. SSL Labs. (2023). SSL/TLS Best Practices. Retrieved from [SSL Labs](https://www.ssllabs.com/)

12. OWASP. (2023). API Security Top 10. Retrieved from [OWASP API Security](https://owasp.org/www-project-api-security/)

13. National Institute of Standards and Technology (NIST). (2023). Security and Privacy Controls for Information Systems and Organizations. Retrieved from [NIST](https://csrc.nist.gov/publications/detail/sp/800-53/rev-5/final)

14. An Efficient Common Substrings Algorithm for On-the-Fly Behavior-Based Malware Detection and Analysis Authors: J. C. Acosta, H. Mendoza, B. G. Medina Year: 2012

15. A Proposal to Realize the Provision of Secure Android Applications - ADMS: An Application Development and Management System Authors: H. Agematsu et al.Year: 2012

16. Security Comparison of Android and IOS and Implementation of User Approved Security (UAS) for Android Authors: D. M. Ahmad, P. Javed Year: 2016

17. Android vs. iOS: The Security Battle Authors: F. Al-Qershi et al. Year: 2014

18. Enhancing Stealthiness & Efficiency of Android Trojans and Defense Possibilities (EnSEAD): Android’s Malware Attack, Stealthiness and Defense: An Improvement Authors: M. Ali, H. Ali, Z. Anwar Year: 2011

19. A Framework for GPU-Accelerated AES-XTS Encryption in Mobile Device Authors: M. A. Alomari, K. Samsudin Year: 2011

20. Oily Residuals Security Threat on Smart Phones Authors: K. AlRowaily, M. AlRubaian

Year: 2011

21. Performance Evaluation of Multi-Pattern Matching Algorithms on Smartphone Authors: A. Amamra, C. Talhi, J. M. Robert Year: 2012

22. Smartphone Malware Detection: From a Survey Towards Taxonomy Authors: A. Amamra, C. Talhi, J. M. Robert Year: 2012

23. Why is My Smartphone Slow? On the Fly Diagnosis of Underperformance on the Mobile Internet Authors: C. Amrutkar et al. Year: 2013

24. Permission-Based Android Malware Detection Authors: Z. Aung, W. Zaw

Year: 2013

25. Malware Detection Method Based on the Control-Flow Construct Feature of Software

Authors: J. Bai, Z. Zhao, J. Wang Year: 2014 Secure Software Installation on Smartphones

Authors: David Barrera, P. Van Oorschot Year: 2011

26. Automatically Securing Permission-Based Software by Reducing the Attack Surface: An Application to AndroidAuthors: A. Bartel et al. Year: 2012

27. Elliptic Curve Cryptography in Practice Authors: J. W. Bos et al. Year: 2014

28. Android Malware Past, Present, and FutureAuthor: C. A. Castillo Year: 2010

29. Malwise-an Effective and Efficient Classification System for Packed and Polymorphic Malware Authors: S. Cesare, Y. Xiang, W. Zhou Year: 2013

30. A Scalable Approach for Malware Detection Through Bounded Feature Space Behavior Modeling Authors: M. Chandramohan et al. Year: 2013

31. Offloading Android Applications to the Cloud Without Customizing Android

Authors: E. Chen, S. Ogata, K. Horikawa Year: 2012

32. A Lightweight Virtualization Solution for Android Devices Authors: W. Chen et al. Year: 2015

33. Identifying Smartphone Malware Using Data Mining Technology Authors: H. Sen Chiang, W. J. Tsaur Year: 2011

34. Security Assessment of Code Obfuscation Based on Dynamic Monitoring in Android Things Authors: T. Cho, H. Kim, J. H. Yi Year: 2017

35. Design and Development of a New Scanning Core Engine for Malware Detection

Authors: L. L. Chuan et al. Year: 2012

36. Android Application Development to Promote Physical Activity in Adolescents

Authors: D. Clark et al. Year: 2012

37. CRePE: Context-Related Policy Enforcement for Android Author: M. C. T. N. N. Crispo Year: 2010

38. Analysis of Zitmo (Zeus in the Mobile) Author: D. Desai Year: 2011

39. A Novel Strategy to Enhance the Android Security Framework Authors: M. Dar, J. Parvez Year: 2014

40. Role of Smartphone in Rural Development: A Case Study of Kashmir Author: M. A. Dar Year: 2016

41. Enhancing Security of Android & iOS by Implementing Need-Based Security (NBS)

Authors: M. A. Dar, J. Parvez Year: 2014

42. Smartphone Operating Systems: Evaluation & Enhancements Authors: M. A. Dar, J. Parvez Year: 2014

43. A Live-Tracking Framework for Smartphones Authors: M. A. Dar, J. Parvez Year: 2015

44. Novel Techniques to Enhance the Security of Smartphone Applications Authors: M. A. Dar, J. Parvez Year: 2016.

45. Security Enhancement in Android Using Elliptic Curve Cryptography Authors: M. A. Dar, J. Parvez Year: 2017

46. Emerging Security Threats for Mobile Platforms Authors: G. Delac, M. Silic, J. Krolo

Year: 2011 iPhone 2.0 SDK: The No Multitasking Myth Author: D. E. Dilger Year: 2008

47. Power Based Malicious Code Detection Techniques for Smartphones Authors: B. Dixon, S. Mishra Year: 2013

48. Enhancing User Privacy on Android Mobile Devices via Permissions Removal

Authors: Q. Do, B. Martini, K. K. R. Choo Year: 2014

49. A Flexible and Lightweight ECC-Based Authentication Solution for Resource-Constrained Systems Authors: N. Druml et al. Year: 2014

50. TaintDroid: An Information-Flow Tracking System for Realtime Privacy Monitoring on Smartphones Authors: W. Enck et al. Year: 2014