Enhancing Password Manager Application Security By Root Detection with Usability and Security Evaluation

Galih Wening Werdi Mukti
Faculty of Computing
President University
Cikarang, Indonesia
galih.mukti@student.president.ac.id

Rusdianto Roestam
Faculty of Computing
President University
Cikarang, Indonesia
rusdianto@president.ac.id

2023 Eighth International Conference on Informatics and Computing (ICIC) | 979-8-3503-4260-4/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICIC60109.2023.10382115

Abstract— **Passwords are commonly used authentication method for many digital applications. The maximum number of passwords that an adult can deal with is four to five passwords. In this time, password manager application is needed to manage passwords, because one person may have many passwords. Password manager is an application used to manage many passwords that are usually owned by someone to access their digital application. However, the use of password manager application will be very unsafe if the mobile device used is rooted/jailbroken. The application is very vulnerable when used on a rooted/jailbroken device, because unauthorized users can see the data in the application. To address this concern, this research focuses on enhancing password manager application security by adding root detection mechanism inside the process of password manager application. The proposed application is designed and developed for Android OS, combining security mechanisms and root detection. To present the results of the proposed application, this research uses usability and security testing. The results are proposed password manager application outperforms most downloaded password manager applications in Playstore such as Keeper, Lastpass, and Dashlane. It can be concluded that proposed password manager application significantly enhances the security of password manager application and provide users with a safer environment for storing their passwords and sensitive information. The contribution of this research is providing a robust and secure password manager by implementing root detection and combining to another security mechanisms from the applications and highlights the essential role of root detection in enhancing the overall security posture of password manager applications.**

Keywords— Mobile Password Manager Application, Root Detection, Security, Usability, OWASP, MobSF, Cybersecurity.

I. INTRODUCTION

In the field of digital applications, passwords are an essential authentication method. Users are required to follow strict guidelines for creating strong and unique passwords [1][2][3], but their ability to remember multiple complex passwords is limited [4][5]. Whereas, storing passwords in online/offline notes might endanger saved passwords and hard to recover [6]. Existing standards and guidelines emphasize the importance of secure password practices, with NIST highlighting password managers as an important defense strategy [3][7][8][9]. Password managers provide the solution by securely storing, generating and retrieving passwords [10]. However, the use of password manager application will be very unsafe if the mobile device used is rooted/jailbroken. These devices are become vulnerable to breaches and malware, requiring strong security measures.

Root detection becomes important for applications that store sensitive data [11], as rooted/jailbroken devices are vulnerable to malicious activity [12]. There are many practical attacks that can be done on rooted device, an article shows that rooted devices can expose Google credentials. In OWASP Mobile Application Security Testing Guide (MASTG) mentioned that it is always possible to easily retrieve the key on a rooted device and then decrypt the values [13]. Based on Symantec Internet Security Threat Report in 2019 [14], rooted or jailbroken were classed as high risk device of the overall number of mobile malware infections.

Previous research analyze security and usage of 14 mobile password manager application and it found that those password manager applications could run in a jailbreak and be the object of reverse engineering [15]. Based on OWASP Mobile Application Security Testing Guide (MASTG), root detection is one of Android Anti-Reversing Defenses. These results clearly indicate that those password manager application haven't implement root/jailbreak detection and it can be conclude as a gap, that is research undertaking security of password manager application haven't implement root/jailbreak detection and restrict the functionalities of the application. NIST recommend possible countermeasures of rooting or jailbreaking are use a mechanisms or other tools to detect rooted/jailbroken devices [16]. By implementing root detection, an additional layer of defense is added against hacker exploits.

The goal of the study are improving the security of the password manager through adding of root detection and evaluate the usability and security of the proposed solution. Root detection is built into the password manager, tested by penetration and malware scanning, and evaluated against OWASP vulnerabilities. To present the results of the proposed application, this research uses usability and security testing. The results highlights the superior security of the proposed solution compared to popular password manager applications. This research enhances the security of password managers through root detection combining with security mechanism. The proposed solution provides strong protection against unauthorized access and data breaches on rooted devices. To ensures a user-friendly experience, drives wider adoption, and promotes better password management practices, this research conduct a usability testing.

II. LITERATURE REVIEW

A. *Password Manager*

Passwords are commonly used authentication method for many digital applications. Previous research found that maximum amount of passwords that normal adults can deal with is four until five [4][5]. Thus, many people decide to use the same passwords across multiple sites. Solution to those password management problems is using a password manager. A password manager is a software application designed to store and manage a user's login details, such as usernames and passwords. Majority password manager application using AES256 for password encryption and PBKDF2 for Key Derivation Function (KDF) [15].

B. *Vulnerability related to Rooted Device*

There are many practical attacks that can be done on rooted device, an article shows that rooted devices can expose Google credentials, proven by Samsung's S-Memo application [17]. A detailed procedure to hack and decrypt WhatsApp Database remotely on rooted Android mobile devices was also presented in public [18]. An article about extract androids applications SQLite database from a rooted android device, and manipulate the database was demonstrated and requires only a few steps [19]. It is represent that rooted device is very vulnerable to data breach.

Potential solution is presented for healthcare apps that managing Personal Health Information (PHI), which regulated with strict legislation in many countries. In this case, the simplest solution is to implement a root detection mechanism and prevent the app from running on a rooted device [20]. Most payment and banking apps are hard-coded to stop working if device tampering is detected. This helps prevent identity theft and fraud [21]. The identified vulnerabilities and risks associated with rooted devices underscore the importance of maintaining the integrity and security of mobile platforms. One of the solution is by having a root detection for application that stored sensitive data of user, such as password manager application.

C. *Root Detection*

Rooting an Android device exposes to a range of security vulnerabilities. By rooting an Android device, users can make unauthorized changes to the operating system that may result data loss. Rooted devices might not get security updates, leaving them open to new attacks [22]. OWASP Mobile Application Security Testing Guide (MASTG), recommends implementing root detection as a best practice for mobile app security, and it is also mentioned that root detection can also be implemented through libraries such as RootBeer [23]. Number of applications that using RootBeer are over 6 thousand and finance applications is the highest category in using RootBeer [24]. Rootbeer provide root detection both Java based and native level. Application secured by native code is hard to break [25]. Based on the comparison of existing root detection [26][27][28], this research elevates the existing methods that usually used for root detection. In this research using 7 methods for root detection in Java based and 1 methods using native check. It is outperform most application that only using 2-3 detection methods [26].

D. *Related Works*

In this part explain about research related to password manager and root detection. Based on the Previous Research about Password Manager can be categorized as four different scope of research: Proposed a keyboard for password manager application [29] [30]; Improve security of password manager [31] [32] [33] [34]; Improve usability of password manager [35] [36]; Comparing the usability, security, design, and use case of existing password managers [37] [38] [39] [40]. Based on Previous Research about Root detection can be categorized as two different scope of research: Evaluate Root detection method from existing applications [11][26] [27] [28] [25] [41] [42]; Attack related to rooted device/root detection [43] [44] [45] [46] [61].

Based on the related works, none of researches are improving password manager security by using root detection. This research filled the gap by implementing root detection to improve password manager security. To evaluate the proposed password manager application is involves comparing it with another password manager and conducting usability testing, as indicated in the previous research in this chapter, which typically considers comparisons with existing password manager applications and usability testing for security improvement research. Additionally, to assess the root detection aspect, the study employs static analytics, following the pattern found in the related works for evaluating root detection methods in existing applications through static analysis.

III. Methodology

This research conducted on design and develop a password manager application and integrate with root detection mechanisms then evaluate its usability and security.

A. *Requirement definitions of password manager application*

This process is carried out by analyzing and observing features provided by 3 most popular mobile password manager applications from Play Store including Lastpass, Dashlane and 1Password. From the identification process of three password manager applications, a feature comparison is obtained as shown in Table 1. It can be determined requirements for proposed password manager application such as sign up, login, logout, add data, save data, edit data, view data, copy data, delete data, generate password, multifactor authentication, and import/export.

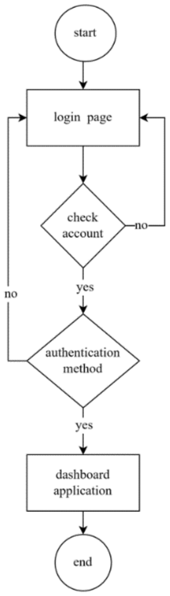
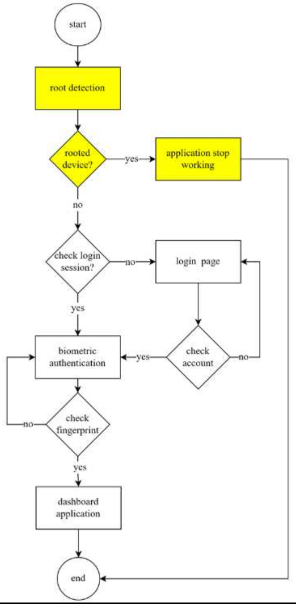
TABLE I. Requirement definitions of password manager applications

No	Features	Lastpass (30 day trial)	Dashlane (30 trial)	Keeper (free version)	Proposed PMA
1.	Sign up	✓	✓	✓	✓
2.	Login	✓	✓	✓	✓
3.	Logout	✓	✓	✓	✓
4.	Add data	✓	✓	✓	✓
5.	Save data	✓	✓	✓	✓
6.	Edit data	✓	✓	✓	✓
7.	View data	✓	✓	✓	✓
8.	Copy data	✓	✓	-	✓
9.	Delete data	✓	✓	✓	✓
10.	Autofill	✓	✓	-	-
11.	Launch	✓	-	-	-
12.	Generate password	✓	✓	✓	✓
13.	Multifactor authentication	-	✓	✓	✓
14.	Import/export/backup	-	✓	-	✓

B. *Design the application*

Based on the requirement definitions there are 12 functions in this proposed password manager applications. **Error! Reference source not found.** shows the system flow comparison of existing Password Manager Applications (on the left side), and system flow of proposed Password Manager Application (on the right side). The yellow box indicates the novelty of this research by adding a root detection mechanism. Root detection in this research implemented in front of the application to check whether the device is rooted or not. If the device is detected as a rooted, the application will exit and can't be normally function to prevent unauthorized access.

TABLE II. SYSTEM FLOW COMPARISON OF EXISTING PASSWORD MANAGER APPLICATION AND PROPOSED PASSWORD MANAGER APPLICATION

Existing PMA [47][48][49]	**Proposed PMA**
 start login page check account no yes no authentication method yes dashboard application end	 start root detection rooted device? yes application stop working no check login session? no login page yes biometric authentication yes check account no no check fingerprint yes dashboard application end

C. Develop the application

This research develop a mobile application using Flutter as a framework and Dart as a programming language of Flutter. Flutter is a Google open source framework for creating multi-platform apps from a single codebase.

D. Usability testing

Usability testing conduct based on use case from [40]. That research is to understand how password managers are intended to be used. The usability of proposed password manager application in this research tested based on 10 essential use cases. Essential use cases focus on the core password management lifecycle.

E. Security testing

Security testing in this research conducted by 3 methods:

1) Assess the application's root detection mechanism by installing it on both rooted and non-rooted devices. The goal is to determine how accurately the root detection mechanism can identify rooted devices.

2) Penetration testing on the application and compare with 3 most downloaded password manager applications in Play Store using MobSF. The goal is to assess the application's security posture thoroughly and identify exploit potential security vulnerabilities.

3) Evaluate security based on OWASP Top 10 and compare with 3 most downloaded password manager application in Play Store. It is to identify common security risks and ensures that the password manager application is well-protected against prevalent attack vectors, reducing the likelihood of successful exploitation.

IV. RESULT

A. Proposed Password Manager Application

Proposed password manager application in this research named as RootPass. The name RootPass itself reflects security as the foundation. RootPass combines security mechanism and root detection. Security mechanisms in the RootPass applications, there are root detection, secure storage, password encryption, key derivation, secure password generator, secure database, and biometric authentication.

1) Root Detection

RootPass implement root detection to addresses a critical issue concerning the vulnerability of password manager applications when installed on rooted devices. Because password managers store sensitive information such as login credentials and saved passwords. If a device is rooted, malicious apps or users with unauthorized access could potentially access the password manager's data files, compromising the security of all stored information. Root detection in this research is using flutter jailbreak and root detection plugin, because it is using 7 methods for root detection in Java based and 1 methods using native check. It is outperform most application that only using 2-3 detection methods.

2) Secure Storage

To ensure that sensitive information and variables related to cryptography process on the RootPass is stored in a safe place, this application uses Flutter Secure Storage which provides API to store data in secure storage. Flutter secure storage is utilized KeyStore based solution which using AES encryption. The Android Keystore system securely stores cryptographic keys within a protected container, making it significantly challenging to extract them from the device.

3) Password Encryption

RootPass implements data encryption to encrypt user's passwords using AES-256 as an algorithm that have been approved internationally based on NIST 800-131A. RootPass using AES-256 due to its strong security properties and widespread adoption. Implementing data encryption in RootPass is to ensure that sensitive user data is always protected.

4) Key Derivation

Key derivation functions play a crucial role in turning user master password into encryption keys for securely storing and retrieving data (saved passwords' users). To encrypt saved passwords that have been stored in the application, RootPass using a key that created from user's master password using Key Derivation Function (KDF). RootPass using PBKDF2 for key derivation function because it is a standard algorithm based on international recommendation NIST 800-132.

5) Secure Password Generator

RootPass utilizing Secure Random class from dart programming language to generate random passwords for user. This feature is to help users create new random password for their accounts. Human-generated passwords tend to have patterns and biases that attackers can exploit. Cryptographically secure random number generators eliminate these patterns, making it more challenging for attackers to guess passwords based on common patterns or behaviors.

6) Secure Database

RootPass encrypt user's passwords before store it to the database to secure the data. Besides user's passwords, application name and username that are associated with user's passwords also stored in the database. Encrypting credentials data in a database is essential for maintaining the confidentiality and security of sensitive user information especially user's passwords. Encrypting credentials ensures that even if an unauthorized person gains access to the database, the stored credentials remain unreadable without the encryption key.

7) Biometric Authentication

RootPass utilizing biometric authentication, specifically fingerprint recognition to provides a highly secure method of verifying user identity. Biometric authentication offers strong non-repudiation. Because each person's fingerprint is unique, making it extremely difficult for unauthorized individuals to gain access to the application.

B. Usability testing

This research tested the usability of a password manager application using 10 essential use cases from [40]. The testing was conducted for three days from July 30th to August 1st, 2023, with participants having more than five digital accounts and Android devices with fingerprint authentication. A group size of 3-20 participants was typically valid used for usability testing [50]. The questionnaire included a pre-questionnaire to identify users' profiles and a post-task questionnaire to complete tasks. Different types of mobile phone and android versions utilized in this usability testing there are 5 types of mobile phone (Samsung, Xiaomi, Realme, Infinix, Oppo) and 3 android versions from 11 until 13 which defined by the respondents. The majority of respondents had multiple digital accounts and sometimes reused passwords, which can pose security risks.

The result from 25 respondents is all use cases meets the expected response according to 100% respondents success completed the tasks based on 10 essential use cases from [40]. It can be concluded that proposed application is appropriate and successful in usability testing. Other than that, several respondents also give responses about the application has been running well, it is a good application, and very useful especially for people who often forget passwords.

C. Security testing

1) Test case: Assess the application's root detection mechanism

For assessment process, this research uses Samsung S7 Edge as a rooted device, the device is rooted using Magisk, then check the rooting status using Root Checker from PlayStore. Otherwise, for non-rooted device this research is using Vivo Y53s. Both rooted and non-rooted devices are android devices that have fingerprint sensor for biometric authentication.

TABLE III. RESULTS OF THE APPLICATION'S ROOT DETECTION SECURITY TESTING

No	Steps	Rooted Device	Non-rooted
1.	Prepare the devices	Samsung S7 Edge, Android 7	Vivo Y53s, Android 13
2.	Checking root status	Rooted	Non-rooted
3.	Install the application	Successfully installed	Successfully installed
4.	Test root detection by executing the application	The application refuse to work and exit	The application works normally
5.	Assess Effectiveness	The application successfully detects rooted device	The application detects non-rooted device

According to the result on **Error! Reference source not found.**, the application's root detection function is effective, it is accurately determining if a device is rooted and responding accordingly. The application terminates execution on rooted devices, indicating the need for security measures. On non-rooted devices, the application runs without restrictions. An effective root detection mechanism is crucial for security-sensitive apps like password managers, as it protects sensitive data from rooted devices and prevents execution or restricts functionalities. To reduce security threats, it is essential to conduct comprehensive security assessments, including penetration testing, to validate the application's overall security posture and ensure resilience against various attack vectors.

2) Penetration testing on the application

Penetration testing was carried out using Mobile Security Framework (MobSF), it is an automated, all-in-one framework for malware analysis, security assessment, and pentesting mobile applications. This research utilize static analysis of MobSF and pentesting 3 most popular password manager applications based on Play Store. Top three of most downloaded password manager applications are Keeper, LastPass, and Dashlane. Keeper and LastPass are installed by more than 10,000,000 users, and Dashlane is installed by more than 5,000,000 users. This research is comparing the proposed application version 1.0 and the latest version of three existing password manager applications.

TABLE IV. COMPARISON RESULTS OF PENETRATION TESTING PASSWORD MANAGER APPLICATIONS

No.	Application Name	Version	Security Score	Risk Rating
1.	RootPass	1.0	60/100	Low Risk (A)
2.	KeePer	16.6.90.123801	57/100	Medium Risk (B)
3.	LastPass	5.22.0.14150	48/100	Medium Risk (B)
4.	Dashlane	6.2328.0-arm64-v8a	44/100	Medium Risk (B)

ERROR! REFERENCE SOURCE NOT FOUND. describe the comparison result of penetration testing of password manager applications. Based on the result findings, the proposed application, RootPass, has the highest security score among the listed password manager applications, with a score of 60/100. It indicates that the application has implemented security measures more effectively than the other password managers. The risk rating for this app is "Low Risk (A)," making it the one with the lowest risk among the applications on this list. It suggests that the application's security posture is relatively strong compared with another password manager applications.

3) Evaluate security based on OWASP Mobile Top 10

OWASP's Mobile Top 10 is a compilation highlighting various security risks that mobile apps encounter on a global scale. This evaluation was carried out using ImmuniWeb Mobile App Security Test. There are 4 types checking in ImmuniWeb Mobile App Security Test, such as OWASP Mobile Top 10 Security Test, Mobile App Permissions and Privacy, Mobile App External Communications, and Software Composition Analysis. In OWASP Mobile Top 10 Security Test there is risk rating methodology to approach risk analysis of mobile application based on likelihood and impact of the risks. The combination of likelihood rating and impact rating results overall risk severity. There are five level of risk severity from highest to lowest are critical risk, high risk, medium risk, low risk and warning.

TABLE V. Summary Result of OWASP Mobile Top 10 Security Test

No.	**App Name**	**Summary result of OWASP Mobile Top 10 Security Test**					
		Critical risk	***High risk***	***Med risk***	***Low risk***	***Warning***	***Total risks***
1.	RootPass	0	0	3	2	7	12
2.	KeePer	0	1	2	5	8	16
3.	LastPass	0	2	3	5	9	19
4.	Dashlane	0	0	4	6	9	19

Based on the result findings on **Error! Reference source not found.**, the proposed application, RootPass, has the lowest count of risks with 3 medium risk, 2 low risk, and 7 warning. A lower overall score indicates better security. KeePer has a total of 16 risks, with one classified as high risk, indicate that it is not as secure as RootPass. LastPass has a total of 19 risks, with two of them being high-risk vulnerabilities. LastPass ranks lower in terms of security when compared to RootPass and KeePer. Dashlane also has a total of 19 risks, with no high-risk vulnerabilities. Similar to LastPass, Dashlane has a higher count of security issues compared to RootPass and KeePer. Dashlane's security performance is not as strong as RootPass and KeePer.

V. DISCUSSION

This research aims to enhance the security of password manager applications when installed on rooted devices, as rooted devices are more vulnerable to malware attacks. Root detection can help prevent unauthorized access, compromised security, and data breaches. To address this issue, the research developed a password manager application with root detection and security mechanisms for Android OS. Root detection in this research implemented in front of the overall application process before login, to check whether the device is rooted or not. If the device is detected as a rooted, the application will exit and can't be normally function to prevent unauthorized access. By conducting root detection before granting access to the application, it ensures that potential threats posed by rooted devices are identified and rejected before any interaction with the application's sensitive data or functionalities. Incorporating root detection at the front of the application process, before login, serves as an essential security checkpoint.

Usability testing was conducted on five different mobile phones, including Samsung, Xiaomi, Realme, Infinix, and Oppo, and three different Android versions 11, 12, and 13. The application was tested on various devices and operating systems to ensure compatibility and identify potential issues. The usability testing of the proposed password manager application was successful. All use cases met the expected response and 100% of respondents were able to complete the tasks. The proposed password manager outperformed three popular password manager apps on the Play Store: Keeper, Lastpass, and Dashlane. The root detection mechanism used in the application was found to be effective in determining if the device is rooted and taking appropriate actions based on the detection results. The application demonstrated commitment to security and a strong proactive approach to protecting sensitive data.

To validate the application's security posture, the research conducted penetration testing using MobSF and ImmuniWeb Mobile App Security Test. The results showed that the proposed password manager application (RootPass) had the highest security score among the three most downloaded password manager applications, with a score of 60/100. The risk rating for this app was "Low Risk (A)," indicating a relatively strong security posture compared to other password manager applications. Furthermore, an assessment to identify common security risks and detect the following weaknesses and vulnerabilities based on OWASP also conduct in this research. The result show that proposed password manager application (RootPass) has the lowest security risks among others password manager applications. Highlighting that it has the fewest security issues across all risk levels and warnings compared to the other password manager applications. In conclusion, the proposed password manager application (RootPass) is the most secure option among tested password manager applications.

VI. CONCLUSION

This research contributes to the security aspect of password management and cybersecurity by providing a robust and secure password manager that adding root detection and combining with security mechanisms. It highlights the essential role of root detection in enhancing the overall security posture of password manager applications. The research also conducts usability testing on RootPass, demonstrating its usability and compatibility with various devices and Android versions. To validate the application's security posture, the research further conducted security testing. The result shows that RootPass is the most secure option among tested password manager applications.

The implications of this research include implementing root detection mechanisms to mitigate risks associated with rooted devices, users being aware of security risks associated with rooted devices, and developers considering root detection as a fundamental layer of defense in their security strategies. Root detection is an important tool for protecting sensitive user information and maintaining the integrity of password manager applications. Recommendations for further research include using more advanced root detection techniques, expanding research scope to cover a wider range of operating systems and devices, improving the usability of password manager applications, and increasing security awareness. By pursuing these recommendations, researchers can further enrich their understanding of password management, root detection, and cybersecurity, ultimately contributing to a safer and more secure digital environment.

References

[1] M. Zviran and W. J. Haga, "Password Security: An Empirical Study," *http://dx.doi.org/10.1080/07421222.1999.11518226*, vol. 15, no. 4, pp. 161–185, 1999, doi: 10.1080/07421222.1999.11518226.

[2] Boston University, "How To Choose a Strong Password : TechWeb : Boston University." https://www.bu.edu/tech/support/information-security/security-for-everyone/how-to-choose-a-strong-password/ (accessed Oct. 16, 2022).

[3] National Center for Education Statistics, "Practical Guidelines for Electronic Education Information Security - Safeguarding Your Technology, NCES Publication 98-297," 1998, Accessed: Oct. 17, 2022. [Online]. Available: https://nces.ed.gov/pubs98/98297.pdf

[4] A. Adams and M. A. Sasse, "Users are not the enemy," *Commun. ACM*, vol. 42, no. 12, pp. 40–46, Dec. 1999, doi: 10.1145/322796.322806.

[5] D. R. Pilar, A. Jaeger, C. Gomes, and L. M. Stein, "Passwords Usage and Human Memory Limitations: A Survey across Age and Educational Background," *PLoS One*, vol. 7, no. 12, p. 51067, 2012, doi: 10.1371/journal.pone.0051067.

[6] N. Adrian, D. Friska, and Y. Kurniawan, “Indonesian Generation Z’s Awareness of Data Privacy in the Use of Social Media,” Mar. 2022.

[7] P. A. Grassi, M. E. Garcia, and J. L. Fenton, “NIST 800-63-3: Digital Identity Guidelines,” *NIST Spec. Publ.*, p. 68, 2017, doi: 10.6028/NIST.SP.800-63-3.

[8] Connie LaSalle, “Cybersecurity Awareness Month 2022: Using Strong Passwords and a Password Manager | NIST,” Oct. 13, 2022. Accessed: Oct. 16, 2022. [Online]. Available: https://www.nist.gov/blogs/cybersecurity-insights/cybersecurity-awareness-month-2022-using-strong-passwords-and-password

[9] National Cybersecurity Alliance, “Cybersecurity Awareness Month - National Cybersecurity Alliance,” Jun. 02, 2022. https://staysafeonline.org/programs/cybersecurity-awareness-month/ (accessed Oct. 16, 2022).

[10] Brandon Lee, “What is NIST guidance on password managers? - Specops Software,” Jun. 10, 2021. https://specopssoft.com/blog/what-is-nist-guidance-on-password-managers/ (accessed Oct. 16, 2022).

[11] A. Kellner, M. Horlboge, K. Rieck, and C. Wressnegger, “False Sense of Security: A Study on the Effectivity of Jailbreak Detection in Banking Apps,” in *2019 IEEE European Symposium on Security and Privacy (EuroS&P)*, Jun. 2019, pp. 1–14. doi: 10.1109/EuroSP.2019.00011.

[12] Cryptomathic, “A Cryptomathic white paper Securing Mobile Apps with MASC,” 2023.

[13] OWASP Mobile Application Security Testing Guide, “Data Storage on Android,” Jul. 2023. https://mobile-security.gitbook.io/mobile-security-testing-guide/android-testing-guide/0x05d-testing-data-storage (accessed Aug. 11, 2023).

[14] Symantec, “ISTR Internet Security Threat Report Volume 24,” 2019.

[15] T. Sean Oesch, “An Analysis of Password Manager Security and Usage on Desktop and Mobile Devices,” The University of Tennessee, Knoxville, 2021.

[16] National Institute of Standards and Technology (NIST), “STA-1 · Mobile Threat Catalogue.” https://pages.nist.gov/mobile-threat-catalogue/stack-threats/STA-1.html (accessed Jul. 13, 2023).

[17] J. Angelo Racoma, “Rooted devices can expose your Google credentials, as proven by Samsung S-Memo database flaw,” Nov. 13, 2012. https://www.androidauthority.com/rooting-security-samsung-s-memo-130895/ (accessed Aug. 11, 2023).

[18] Pierluigi Paganini, “How to Hack and Decrypt WhatsApp Database on rooted devices,” *Security Affairs*, 2015. https://securityaffairs.com/40727/hacking/hack-decrypt-whatsapp-database.html (accessed Aug. 11, 2023).

[19] Piyush Kumawat, “Extract Android Application’s Database,” *Security Cipher*, 2022. https://securitycipher.com/2022/09/10/how-to-extract-database-from-android-application-rooted-devices/ (accessed Aug. 11, 2023).

[20] VLADIMIR IVANOV, “The risks of letting your app run on a rooted device,” 2022. https://vvsevolodovich.dev/risks-letting-app-run-on-a-rooted-device/ (accessed Aug. 11, 2023).

[21] Domenic Molinaro, “What Is Rooting? The Risks of Rooting Your Android Device,” *Avast*, 2023. https://www.avast.com/c-rooting-android (accessed Aug. 12, 2023).

[22] Blue Cedar, “Root Detection.” https://www.bluecedar.com/mobile-app-security-technical-glossary/root-detection (accessed Jul. 14, 2023).

[23] OWASP Mobile Application Security Testing Guide, “Android Anti-Reversing Defenses,” 2023. https://mobile-security.gitbook.io/mobile-security-testing-guide/android-testing-guide/0x05j-testing-resiliency-against-reverse-engineering (accessed Jul. 17, 2023).

[24] AppBrain Statistics, “RootBeer,” Jun. 13, 2023. https://www.appbrain.com/stats/libraries/details/rootbeer/rootbeer# (accessed Jul. 14, 2023).

[25] L. Nguyen-Vu, N.-T. Chau, S. Kang, and S. Jung, “Android Rooting: An Arms Race between Evasion and Detection,” *Secur. Commun. Networks*, vol. 2017, pp. 1–13, 2017, doi: 10.1155/2017/4121765.

[26] S.-T. Sun, A. Cuadros, and K. Beznosov, “Android Rooting: Methods, Detection, and Evasion,” in *Proceedings of the 5th Annual ACM CCS Workshop on Security and Privacy in Smartphones and Mobile Devices*, Oct. 2015, pp. 3–14. doi: 10.1145/2808117.2808126.

[27] D. Geist, M. Nigmatullin, and R. Bierens, “Jailbreak/Root Detection Evasion Study on iOS and Android,” 2016.

[28] N. S. Evans, A. Benameur, and Y. Shen, “All your Root Checks are Belong to Us,” in *Proceedings of the 13th ACM International Symposium on Mobility Management and Wireless Access*, Nov. 2015, pp. 81–88. doi: 10.1145/2810362.2810364.

[29] S. Fahl, M. Harbach, M. Oltrogge, T. Muders, and M. Smith, “Hey, You, Get Off of My Clipboard,” 2013, pp. 144–161. doi: 10.1007/978-3-642-39884-1_12.

[30] F. Boukayoua, B. De Decker, and V. Naessens, “A keyboard that manages your passwords in Android,” in *2014 International Conference on Privacy and Security in Mobile Systems (PRISMS)*, May 2014, pp. 1–4. doi: 10.1109/PRISMS.2014.6970592.

[31] S. Agholor, A. S. Sodiya, A. T. Akinwale, and O. J. Adeniran, “A secured Mobile-Based Password Manager,” in *2016 Sixth International Conference on Digital Information Processing and Communications (ICDIPC)*, Apr. 2016, pp. 103–108. doi: 10.1109/ICDIPC.2016.7470800.

[32] L. Wang, Y. Li, and K. Sun, “Amnesia: A Bilateral Generative Password Manager,” in *2016 IEEE 36th International Conference on Distributed Computing Systems (ICDCS)*, Jun. 2016, pp. 313–322. doi: 10.1109/ICDCS.2016.90.

[33] M. Shirvanian, S. Jareckiy, H. Krawczykz, and N. Saxena, “SPHINX: A Password Store that Perfectly Hides Passwords from Itself,” in *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*, Jun. 2017, pp. 1094–1104. doi: 10.1109/ICDCS.2017.64.

[34] D. McCarney, “Password Managers: Comparative Evaluation, Design, Implementation and Empirical Analysis,” Carleton University, Ottawa, Ontario, 2013. doi: 10.22215/etd/2013-09932.

[35] M. Aliasgari, N. Sabol, and A. Sharma, “Sesame: a secure and convenient mobile solution for passwords,” in *2015 First Conference on Mobile and Secure Services (MOBISECSERV)*, Feb. 2015, pp. 1–5. doi: 10.1109/MOBISECSERV.2015.7072879.

[36] N. M. Barbosa, J. Hayes, and Y. Wang, “UniPass: Design and Evaluation of a Smart Device-Based Password Manager for Visually Impaired Users,” in *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, Sep. 2016, pp. 49–60. doi: 10.1145/2971648.2971722.

[37] P. Arias-Cabarcos, A. Marin, D. Palacios, F. Almenarez, and D. Diaz-Sanchez, “Comparing Password Management Software: Toward Usable and Secure Enterprise Authentication,” *IT Prof.*, vol. 18, no. 5, pp. 34–40, Sep. 2016, doi: 10.1109/MITP.2016.81.

[38] S. Chaudhary, T. Schafeitel-Tähtinen, M. Helenius, and E. Berki, “Usability, security and trust in password managers: A quest for user-centric properties and features,” *Comput. Sci. Rev.*, vol. 33, pp. 69–90, Aug. 2019, doi: 10.1016/j.cosrev.2019.03.002.

[39] P. Gasti and K. B. Rasmussen, “On the Security of Password Manager Database Formats,” 2012, pp. 770–787. doi: 10.1007/978-3-642-33167-1_44.

[40] J. Simmons, O. Diallo, S. Oesch, and S. Ruoti, “Systematization of Password ManagerUse Cases and Design Paradigms,” in *Annual Computer Security Applications Conference*, Dec. 2021, pp. 528–540. doi: 10.1145/3485832.3485889.

[41] J. Jin and W. Zhang, “System Log-Based Android Root State Detection,” 2017, pp. 793–798. doi: 10.1007/978-3-319-68542-7_69.

[42] R. Bialon, “On Root Detection Strategies for Android Devices,” Dec. 2020, [Online]. Available: http://arxiv.org/abs/2012.01812

[43] I. Gasparis, Z. Qian, C. Song, and S. V Krishnamurthy, “Detecting Android Root Exploits by Learning from Root Providers”, Accessed: Jul. 14, 2023. [Online]. Available: https://www.usenix.org/conference/usenixsecurity17/technical-sessions/presentation/gasparis

[44] B. Soewito and A. Suwandaru, “Android sensitive data leakage prevention with rooting detection using Java function hooking”, doi: 10.1016/j.jksuci.2020.07.006.

[45] Z. Zhang, Y. Wang, J. Jing, Q. Wang, and L. Lei, “Once Root Always a Threat: Analyzing the Security Threats of Android Permission System,” 2014, pp. 354–369. doi: 10.1007/978-3-319-08344-5_23.

[46] L. Casati and A. Visconti, “The Dangers of Rooting: Data Leakage Detection in Android Applications,” *Mob. Inf. Syst.*, vol. 2018, pp. 1–9, 2018, doi: 10.1155/2018/6020461.

[47] S. Patil, K. Wasnik, and S. Bagade, “Pen-Drive Based Password Management System for Online Accounts,” 2019, pp. 693–704. doi: 10.1007/978-981-13-1951-8_62.

[48] Zhelyazko Petrov and Razvan Ragazan, “Password Manager with 3-Step Authentication System,” *Natl. Informatics Conf. Dedic. to 80th Anniv. birth Profr. Petar Burn. Collect. reports*, 2016, Accessed: Aug. 13, 2023. [Online]. Available: http://www.math.bas.bg/infres/cib80/book/cib80-p14.pdf

[49] TIJANI SAHEED OLAWALE, “DEVELOPING A PASSWORD MANAGEMENT SCHEME.” Accessed: Aug. 13, 2023. [Online]. Available: https://www.academia.edu/25357880/DEVELOPING_A_PASSWORD_MANAGEMENT_SCHEME_BY_TIJANI_SAHEED_OLAWALE

_CST_12_COM_00207_CHAPTER_ONE

[50] R. Macefield, “How To Specify the Participant Group Size for Usability Studies: A Practitioner’s Guide,” 2009.