JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue

JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)

An International Scholarly Open Access, Peer-reviewed, Refereed Journal

A Review on Database Security Techniques used in Android Operating System

[1]Aishwarya Phadtare, [2]Kishor Mane

[1]M.E Student, [2]Assistant Professor
[1]Department of Computer Science and Engineering,
[1]D. Y. Patil College of Engineering & Technology, Kolhapur, India
[2]Department of Computer Science and Engineering,
[2]D. Y. Patil College of Engineering & Technology, Kolhapur, India

Abstract : With increased growth in mobile technology the use of mobile devices has increased eventually. As a result, mobile devices have become portable storage that stores the user information including personal as well as business related. The huge amount of data generated and stored by mobile devices is due to large usage of mobile devices in day-to-day life. Currently, the Android operating system is one of the major operating systems in the mobile phone market, which stores user data on local storage inside the database. When an android device is unlocked the data in the device is easily accessible to anyone because android stores data in the form of plaintext. This paper focuses on the detailed review of the different android techniques of database security, an overview of different security attacks on android devices. Also, explains the different techniques for similarity comparison of applications.

***IndexTerms* - Android Database Security, Attacks, Application Similarity Comparison, Encryption, Decryption.**

I. INTRODUCTION

Android operating system is one of the major operating systems in the mobile phone market which stores user data in the database and provides API's. The APIs are used to get access to android databases and are managed by the system or by the user. Android operating system is free and open source but google prohibits uncertified devices. The data generated by mobile devices may include confidential information of users which needs to be encrypted. Google uses "Full Disk Encryption" (FDE) [6] which performs encryption and decryption. "Full Disk Encryption" safeguards user data only when users' device is locked. When an android device is unlocked the data in the device is decrypted. Android stores data in the form of plaintext. As result data can be leaked easily by anyone who has access to the device. Most applications ask the user of the application to enter password that is used to generate encryption key for data encryption/decryption, else hard coded information is used for encryption/decryption [21]. If an app requires a password for key generation the user must enter password every time which is not convenient. while If user use hard coded information, it is likely to be exposed [6]. Smartphones provide services like text, mail, Wi-Fi, Bluetooth, [12] etc. The data is exchanged through 3G, 4G network, Wi-Fi different wireless medium as a result the chances of data leakage is high [12]. While connecting user device to open hotspot for internet access can divert traffic on device to attacker [12] thus security while using wireless medium is also main concern in android smartphones. There are many android applications developed in market, while most application are misusing user data and privacy to make revenue out of it [21]. Using user data with the device user knowledge, keeping track of user activities, displaying advertisements with links that introduce viruses in user devices are some of the most common challenges in mobile market and these challenges are growing. In this paper section-II discusses Android Database Security, in section-III different security Attacks on android device are discussed. In section-IV Application Similarity Comparison techniques for Android Operating System are discussed.

II. ANDROID DATABASE SECURITY

Android operating system is a mobile operating system. Android OS is used mostly on smartphones and tablets. Mobile devices are used on a large scale as they are small and portable storage. Large amount of user data is stored on mobile devices. Android smartphones are used by many users. The data on android mobile is stored in android database. The data is stored in plaintext and is thus exposed to many vulnerabilities. Different researchers have proposed methods to protect users' data. A security architecture for secure android database [6] is designed. Current android access is based on uid. Application and the database are paired but to secure the data in database the database is separated from the app and the database is no longer in plaintext, it is encrypted. Along with uid one to one pairing is created between database and app. Hence, only the owner of the application is granted access. The android database is encrypted using keys generated by keystore. The secure database daemon requests keystore to generate key which is encrypted by using device specific key which is unique to each device. While the daemon randomly generates an encryption key for each application and database of that application. In the proposed architecture the data is encrypted but there is overhead on the time required for database operation. Android devices are prone to many security threats [7] like device loss, malware attacks, unwanted calls and messages, untrusted applications that can wipe off all information without user's knowledge. In Secure storage application [7] password-based encryption is used to protect data on android device in which it provides facility of encrypted files reversing. This reversing mechanism is not present in encryption system of android. In secure storage the application asks the user to enter password which is used for cryptographic operation. The key generation engine generates encrypted key. The key generation engine generates master key using password provided by the user and then extract key which is used for encryption/decryption. Large number of fraud mobile application target Android based smartphones [12]. Malicious apps download modules from servers thus the attacker can get easy access to user information and personal confidential data stored in the device. Android provides sandbox which protects codes and data of an application from another application. Sandbox isolates code and data of different applications from each other thus, preventing interference and using other apps code from being used. Android provides client-server model based ContentProviders. Even if an application is aware of other applications database location it cannot access it directly hence, the ContentProviders is used to share database with other application when necessary. The app that wants to share database must use URL same as data identifiers. Client ContentResolver communicates with server ContentProviders using queries. But by using reverse engineering user data can be retrieved by attackers. Hence, Android has feature Proguard which deletes code that is not required to optimize program and changes name of class fields to make it difficult and unclear to prevent reverse-engineered results. User grants unnecessary permission to certain application to user data on the devices by agreeing to the terms and condition without reading. Thus, fraud applications get access to irrelevant data from the user. The permission control mechanism [20] has four classes of permission such as Normal which is low risk permission, Dangerous which is high risk permission, Signature is permission requested granted if from app with same signature. SignatureOrSystem is permission requested granted if from app with same signature and image of app is in android system. Android operating system is not effective to control unnecessary access. Today smartphones are connected to the Internet which leads to attacks on the user's data. Due to unprotected access to the Internet, problems like overuse of CPU and memory along with an increase in battery power drain takes place. Thus, dynamic-based solution [20] detects fraud application by analyzing behavior of application. But based on only behavior proper decision cannot be taken. While in Recommendation based method apps based on permission that are requested and responses accordingly are considered from which a ranking algorithm evaluates the request made and a voting algorithm calculate response according to the permission request. To prevent access and privileges to applications the mechanism is designed to detect over-privileged applications [11] by sharing the permission using shared user identity. By using this mechanism single as well as group of such application is detected. Thus, preventing user data from being invaded and exploited.

III. SECURITY ATTACKS ON ANDROID DEVICE

Mobile devices store large amounts of user data. These devices use different operating systems. Mostly android operating systems are used as operating systems in smartphones. Attacks are performed by fraud applications to get access to user data thus, security is the main concern. Different security attacks can be performed on android operating systems. Following are the list of some of the security attacks:

1. Mobile privacy- Android mobile device has password lock to protect the data. The data in android database is in plaintext hence if attacker knows the password, users' privacy can be invaded, and users' data can be gained easily [6] [12] [20].

2. Malignant Application- Defamatory applications [12] are present on the google store. Such applications reveal users contacts, messages, location, etc. While some applications download other destructive applications in background that may contain malware and viruses. Also, such malignant applications can make phone calls or send messages in background and increase user data usage and make profit.

3. Network related attacks- Users' data can also be gained if the android mobile device connects to network point set up by attacker and all data on the device and activities can be drawn [12]. Users can connect to open hotspot to use Internet but instead get attacked by attacker by getting users information like contacts, text messages, credit card or bank account details, etc.

4. Device loss- Mobile devices are small and portable thus can be lost. Hence user data can be at risk [12].

5. Colluding Attack- Some applications in android operating system share same uid. Hence this may lead to colluding attack as application sharing user id can get excessive privileges [6] [9]. Attackers can create similar applications to get user information. Collusion detection is required as applications need to limit users resource use and excess flow of users' data to other application. Attackers design graphically similar looking applications and claim as genuine preexisting applications [9]. Users can unknowingly install such an application as it is graphically similar thus getting exploited by attacker.

6. SQL injection- Android SQLite database is used to store user data. This information can be acquired by using SQL queries [15]. Malicious code can be inserted into input strings of SQL query where the condition is always true in database to acquire data from database [10] [15]. By using strings that are stored in a table or as metadata code can be inserted and these codes works as these strings are called by using SQL command [10]. The SQL injection attacks access ordinary Web page. The basic firewall is not able detect SQL injection attacks so artificial detection techniques are required [10] [15].

IV. APPLICATION SIMILARITY COMPARISON

In the android operating system, if the application has bugs that need to be removed or even certain new features need to be added, then in such cases updating the application is required. Fingerprint is unique for each application. It is also called metadata information. Hence when an application is updated, the fingerprint of application is also updated. Malicious application can be installed if a similar looking app is installed as an updated version of an application. Thus, an application similarity comparison is required to check if the newly updated application is a genuine update of the old version of the application or some other fraud app. BloomFilter [17]is used to check if an element does belong to a set defined. The length of BloomFilter is based on the number of elements. Probability is calculated. This is calculated on the bases of hash functions and elements in the set. Some classes may have fewer methods while the updated application classes may have more. Thus, this is one of the ways used to check if the two classes are similar or not. Jaccard similarity coefficient [19]is also used for measuring the similarity between applications by calculating the similarity score. The difference between original and updated code will be represented by addition, deletion, and modification of elements of set. The measure of similar instruction sequences between two applications will be represented as similarity score. Prior detection algorithm [1] is used to check similar applications. Repackaging of the application code creates a privacy threat to the user. Using prior detection technique similar code is identified. Novel tree structure similarity comparison is done in which tree is generated equivalent matrix is traversed to check similarity. This is carried on multidimension sequence. Multidimensional sequence presented as tree have sequential dimension and spatial dimension both similarity is combined to get tree similarity hence similar code is detected which avoid collusion of applications. Cosine similarity method [5] is used to measure whether documents are similar or not. In this method two vectors are compared. A document consists of keywords and phrases. Each document is a vector. Function is defined in the range 0,1 to get similarity. Depending on the similarity function the closeness of object is determined. This method works for large datasize. Word mover's distance [9] is also technique used to check similarity. This method is also used to check documents. A word is converted into vector of numbers. The cooccurrences of the word in sentence is calculated. In word movers distance the similarity is measured by distance one vector of document takes to travel to another vector of another document with which it is compared.

From the above literature review, it is observed that following are the limitation in the current system-

1. Android application uses user identification to get access to data. But, only using uid for granting access is not safe.
2. A fraud similar looking app can get access to data in database which is stored in plaintext and can be exploited.
3. If two applications are similar or not need to be checked by comparing to avoid collusion.
4. New features are added by updating app but if a fraud similar looking app is installed as updated version of a pervious installed app, access can be granted to user data for the fraud application.

To remove the above limitations following are the proposed solutions as –

1. Android database stores data in plaintext so to prevent it from being access the data should be encrypted using effective encryption algorithm like Advance Encryption Algorithm. A safe ecosystem needs to be designed to secure devices from any attacks. The application and the database of the application should be separated and access to the database should be granted only after a security check of identity of application is performed.

2. Android databases are used to store user data. This user data can be acquired by using SQL queries [15]. Malicious code can be inserted into input strings of SQL query where the condition is always true in database to acquire data from database. By using the IP address of user, logs record of file that are access, log file size and content it can be detected if attackers have done SQL injection attacks or not. Validating user input like assumptions of type, size or content of the data should not be made [10] [15] can help to protect against SQL injection.

3. Some android applications share the same uid which causes the colluding attack. It can be avoided by comparison techniques.

V. CONCLUSION

In this paper, the discussion has been made for different security issues and techniques for android database security parameters like key generation for database encryption, pairing of apps with its database. The different threats to android devices include mobile privacy, malignant application attacks, network related attacks, colluding attacks, SQL injection attacks are considered. To check similar applications, different methods like bloomfilter, Jaccard similarity, prior detection algorithm, novel tree comparison, cosine similarity method, word movers' distance are used. Thus, there is a need for a system framework that can separate android database from application and android database can be protected by encrypting data in database and certain validation to prevent attack on database. Also, prevents the colluding attack using similarity comparison techniques.

REFERENCES

[1] Zhiyong Song, Liquan Chen,"MDSDroid: A Multi-level Detection System for Android Repackaged Applications", 6th International Conference on Signal and Image Processing, 2021.

[2] Sen Chen, Feng We, Tianming Liu, "ATVHunter: Reliable Version Detection of Third-Party Libraries for Vulnerability Identification in Android Applications", ACM 43rd International Conference on Software Engineering, 2021.

[3] Sujata Chakravarty, Ravi Kiran verma, Santosh K,"Feature Selection and Evaluation of Permission-based Android Malware Detection", 4th International Conference on Trends in Electronics and Informatics,2020.

[4] Igor Zavalyshyn, Nuno Santos,"Flowverine: Leveraging Dataflow Programming for Building Privacy-Sensitive Android Applications", 19th International Conference on Trust, Security and Privacy in Computing and Communications,2020.

[5] Noor, Alfirna, "Cosine similarity to determine similarity measure" International Conference on Cyber and IT Service Management, 2016.

[6] Jin Hyung Park, Seok-Man Yoo, In Seok Kim and Dong Hoon Lee, "Security Architecture for a Secure Database on Android", J. H. Park, S. Yoo, I. S. Kim and D. H. Lee, "Security Architecture for a Secure Database on Android," in IEEE transaction of multidisciplinary open access, vol. 6, pp. 11482-11501, pages 1-20, 2018.

[7] Poonguzhali P, Prajyot Dhanokar, M. K. Chaithanya, and Mahesh U. Patil," Secure Storage of Data on Android Based Devices", IACSIT International Journal of Engineering and Technology, Vol. 8, No. 3, June 2016, pages 1-6.

[8] Alexander Uskov, Adam Byerly, Colleen Heinemann, "Advanced Encryption Standard Analysis with Multimedia Data on Intel AES-NI Architecture" International Journal of Computer Science and Applications, Technomathematics Research Foundation Vol. 13, No. 2, pp. 89 – 105, 2016

[9] Jiawei Zhu, Zhengang Wu, Zhi Guan, and Zhong Chen, "Appearance Similarity Evaluation for Android Applications", 2015 Seventh International Conference on Advanced Computational Intelligence (ICACI), Wuyi, 2015, pp. 323-328, pages 1-6, 2015.

[10] Li Qian, Zhenyuan Zhu, Jun Hu and Shuying Liu, 1"Research of SQL injection attack and prevention technology," 2015 International Conference on Estimation, Detection and Information Fusion (ICEDIF), Harbin, 2015, pp. 303-306, pages 1-4, 2015.

[11] Iman Kashefi, Maryam Kassiri, and Mazleena Salleh, "Preventing Collusion Attack in Android", Department of Computer Science, Universiti Teknologi Malaysia, Department of Computer and Information Technology, Islamic Azad University, Iran the International Arab Journal of Information Technology, Vol. 12, No. 6A, 2015.

[12] P. D. Meshram Dr. R.C. Thool, "A Survey Paper on Vulnerabilities in Android OS and Security of Android Devices", 2014 IEEE Global Conference on Wireless Computing & Networking (GCWCN), Lonavala, 2014, pp. 174-178., pages 1-5, 2014.

[13] Qian Li, Xueli Hu, Hao Wu, "Database Management Strategy and Recovery Methods of Android", 2014 IEEE 5th International Conference on Software Engineering and Service Science, Beijing, 2014, pp. 727-730, pages 1-4, 2014.

[14] T. Cooijmans, J. de Ruiter, and E. Poll, ''Analysis of secure key storage solutions on Android,' in Proc. 4th ACM Workshop Security Privacy Smartphones Mobile Devices, 2014, pages 1-10.

[15] Kumar, P., & Pateriya, R. K. (2012)," A survey on SQL injection attacks, detection and prevention techniques", 2012 Third International Conference on Computing, Communication and Networking Technologies (ICCCNT'12), pages 1-5.

[16] David McGrew Cisco Systems, Inc., "Efficient authentication of large, dynamic data sets using Galois/Counter Mode (GCM)", Third IEEE International Security in Storage Workshop (SISW'05), San Francisco, CA, 2005, pp. 6 pp.-94.

[17] B. H. Bloom, ''Space/time trade-offs in hash coding with allowable errors,' Commun. ACM, vol. 13, no. 7, pp. 422–426, Jul. 1970.

[18] Y. Qiao, T. Li, and S. Chen, ''Fast Bloom filters and their generalization,'' IEEE Trans. Parallel Distrib. Syst., vol. 25, no. 1, pp. 93–103, Jan. 2014.

[19] P. Jaccard, ''Étude comparative de la distribution florale dans une portion des alpes et des jura,'' Bull. Soc. Vaudoise Sci. Naturelles, vol. 37, pp. 547–579, 1901

[20] Bahman Rashidi∗ and Carol Fung Virginia "A Survey of Android Security Threats and Defenses" Commonwealth University, Richmond, Virginia, USA, pages 1-33

[21] Igor Khokhlov, Leon Reznik, "Android System Security Evaluation", 15th IEEE Annual Consumer Communications & Networking Conference (CCNC)

[22]Khokhlov, Leon Reznik, "Colluded Applications Vulnerabilities in Android Devices", IEEE 15th Intl Conf on Dependable, Autonomic and Secure Computing, 15th Intl Conf on Pervasive Intelligence, 2017.

[23]Li, Tegawende, Klein, "SimiDroid: Identifying and Explaining Similarities in Android Apps" Interdisciplinary Centre for Security, Reliability and Trust, 2017.

[24] Min Jiangbo Guo, BaoShuai, Pengpeng, "The Design and Implementation of Security Defense System Based on Android", school of Information Science and Engineering, Hebei University of Science and Technology, 2017.

[25]Umar M, "Android Malware Detection Based on a Hybrid Deep Learning Model", Hindawi Security and Communication Networks Volume.

[26] Kantola, Erika Chin, Wagner "Reducing Attack Surfaces for Intra-Application Communication in Android", Proceedings of the second ACM workshop on Security and privacy in smartphones and mobile devices.

[27] Bacis, Simon, "AppPolicyModules: Mandatory Access Control for Third-PartyApps",10th ACM Symposium on Information, Computer and Communications ACM.

[28] K.M. Nver, and Ghanem, "The Android malware static analysis:techniques, limitations, and open challenges," 3rd International Conference on Computer Science and Engineering.

[29] P. Rahul, Enck Xie, "WHYPER: towards automating risk assessment of mobile applications," in Proceedings of the 22nd Usenix Security 2013.