Research on the Security of Deleting Data for Android File Manager

Abstract：With the rapid development of mobile Internet, intelligent mobile terminal has become one of the important carrier of personal privacy data. But for the current mobile smart terminal, the system provides data removal interface does not achieve the desired security. There are some unavoidable shortcomings in the existing Android design model, especially the security model, which leads to the risk of these imperceptible residual data, which are ignored by previous security analyzes and even subjectively considered not existing. In this case, the attacker can use the loophole of security protection to launch an attack, and thus more seriously threaten the device-related users’ privacy and data security. By using AOP slicing technology based on the Android system, we study the method of deleting data in various file managers existing in Android system. The research shows that some of these methods have serious security risks. By focusing on the analysis of deleteing data through the file manager, which is a way that users commonly choose to do the delete data task, the experiment successfully recovered a large number of data deleted from the Android device.This proved that the current file manager to delete the data is not safe. We Propose a solution called PureEnc to protect data from Android devices against data retention issues. PureEnc can automatically modify the Android application to achieve the purpose of protecting data. You can easily deploy and effectively protect the application data to deal with data residual vulnerability without modify the Android system.

Keywords: Android, data deletion, data residue

1.introduction

The popularity of mobile smart terminal devices has changed the way people deal with personal information. More and more people choose to use mobile devices and other mobile devices to deal with personal affairs and work services. Mobile application developers offer a variety of programs to help people better manage the kinds of information used in their lives, such as using social applications to keep in touch with friends, using bank applications to handle their own banking transactions, The camera application is used to record the moments of life. This trend make the mobile smart terminal equipments contain more and more information closely related to the user's personal privacy data. As a result, personal privacy data leakage incidents occur more frequently. As the most popular mobile intelligent terminal operating system, in a very long period of time, Android is the main source of personal privacy data leakage occurred [1]. The main reason for this situation is that the Android system and applications for individuals have a security risk in the handling of privacy data, especially when the data deletion operation is performed, which does not satisfy the pre-conceived security. Although the problem of data residue on flash media [2] has been studied a lot, but in the Android system, this problem has still not got the attention from file system and operating system level. In the Android application development document, there is no special explanation for the problem or warnings for developers who use the general data interface to delete data about the existance of security issues, which led to the vast majority of Android applications are using unsafe data removal interface to handle sensitive data. An this eventually leads to the occurrence of data leakage.

2.background

2.1 Android data storage

Different from traditional desktop computers, equipped with Android system, intelligent mobile terminals often choose flash memory as their own storage media. For flash memory, the most important feature is that all the data on the flash memory is written in blocks. In general, the size of a data block is a few hundred to several thousand bytes. When data deletion is performed, even if the data to be deleted is only part of a block, it is necessary to erase all the data on the entire block. Therefore, in order to improve the efficiency of disk read and write, when the data is deleted, the flash control chip will not really put all the data in the data block 0, but the data block is marked as unused. Later when other data is written into the data block, the so called “deleted data” is really erased. This results in the existance of the deleted data in the flash media for a period of time. The length of this period of time depends the size of the media and the frequency of disk read and write [3].

2.2 Data erase

In general, Android system has two main data erasure. The first is by calling the system API to delete the file directly, in this way to delete the file, its content will continue to remain on the disk. Through the full disk image extraction, you can get these residual data content, and then through some means of evidence can be restored to the original data content. In the Android system, the vast majority of data erase operations are in this way, although the interface in the system is not the same, but at the bottom of the system is to delete the file to delete Linux file. Android system, another data erase mode is to restore the factory settings, by triggering the system settings interface interface, Android system will restart to Recovery mode. In Recovery mode, the entire userdata partition is automatically formatted. In general, the system will call the flash media interface to rewrite the entire partition, to ensure that all the data on the partition are safely erased.

2.3 Android data protection

Google, from Android 3.0, has been adding the choice of full disk encryption to Android system. Full disk encryption is modified based on Linux dm-crypt features. Users can choose to open the disk encryption for their own devices in the system settings. After all disk encryption is enabled, the user needs to enter the root key for decryption each time the system starts. Full disk encryption is a one-time irreversible process, which means that once the user has full disk encryption, unless the entire device is formatted, there is no other way to unlock the disk's encryption state.

In the Android system, in order to reduce the impact of full disk encryption on system performance, Google removed the dm-erypt program for all against the forensic security policy, which to some extent weakened the full disk encryption security. On the other hand, the on-the-fly eneryption / decryption feature of the dm-crypt scheme determines that full disk encryption can not confront attacks from within the system. When the system is already running, all data on the disk will no longer receive full disk encryption protection, with data access to any user and App can read the decrypted data. In the Android device root is very common today, full disk encryption and can not effectively protect the security of disk data.

In addition, due to some flaws in the design of the program, resulting in full disk encryption did not bring the expected security. The full disk encryption key in the Android system is generated by the user's lock screen PIN, and the encryption parameters required to encrypt and decrypt the disk are stored in the crypto footer structure at the end of the corresponding partition of the local disk. By parsing this structure, an attacker can get the parameters needed to initiate a violent exhaustive attack on a disk that is protected by full disk encryption, that is, an exhaustive PIN. At the same time, as a daily use of handheld devices, the vast majority of Android users will not use too complex PIN to lock their phone. So for the Android default 6-bit PIN, any ordinary home computer can be completed within 1 to 2 hours to crack, and if you use a professional large computer to attack, crack the PIN range will be greatly improved.

These insecure features of full disk encryption are greatly amplified under the influence of disk data residue. For Android devices that are abandoned or accidentally lost, even if full disk encryption is turned on, attacker can still use the remaining disk Data to carry out the above attacks.and even found in the use of third-party recovery of the equipment, due to the lack of recovery developers, you want to format all the disk disk has become very difficult to become encrypted. And it’s found in some of the use of third-party recovery of the equipment, duing to the recovery of the developer's negligence, formating all disk encrypted disk has become very difficult.

To attack a device that has full disk encryption, you first need to extract the relevant partitions of the disk, including the data partition where the encrypted data is stored and the special partition where the crypto footer is stored. After finishing analysis on crypto footer , attacters obtain the necessary parameters to calculate the master key. Then thet use some violent crack script on the user's PIN code for the exhaustive attempt to use a different PIN code to decrypt the data partition of the first block. If the decrypted data conforms to the standard format of the ext4 file system, then the correct PIN code is found, and then the PIN code can be used to decrypt the entire data partition and get plaintext data.

It is worth mentioning that the data residue on the disk is the probability of existence, not every byte will remain on the disk, which means that the residual information extracted from the disk may be damaged, incomplete, if these If the incomplete happens to the head of the data partition or the part where the crypot footer is saved, the missing key data will cause the violent crack to fail.

For the above reasons, this paper believes that the traditional Android full disk encryption program can not achieve its expected security strength, can not exist in the case of data residue to protect the safety of disk data.

Reference: [1]Baekes M，Gerling S，HammerC，et a1．AppGuard-enfordng user requirements on android apps[C]//International Confe-rence on Tools and Algorithms for the Construction and Analysis of Systems(TACAS’2013)．Rome：Springer Berlin Heidelberg，2013：543-54

[2] Flash memory[OL]．[2016-5-17]．http://en.wikipedia.org/w/index php?title=

Flash\_memory

[3] Bugiel S，Heuser S，Sadeghi A R. Flexible and fine-grained mandatory access control on Android for diverse security and privacy policies[C]//Usenix Conference on Security．2013 :131—146

[4]scrypt[OL].[2016-3-14].http://www.tarsnap.com/scrypt.html