PROTECTION OF PERSONAL DATA IN MOBILE DEVICES

Laziz Shirinov
Tashkent university of information technology named after
Muhammad al-Khwarizmi

ANNOTATION

The article discusses the protection of personal data in mobile devices, protective mechanisms, privacy issues, abuse. Recommendations are given on how to protect data in mobile devices.

Keyword: *protection, personal data, mobile devices, Android, iOS.*

Introduction

Even the most primitive cell phone models store so much data that if the device falls into the wrong hands, its owner will be in serious trouble.

Everyone is switching to mobile communications. Our life - not only business, but also personal - is inextricably linked with a cell phone. Even the term “cell phone” no longer captures the essence of devices such as BlackBerry, iPhone and other smartphones, with which many people spend almost as much time as they do with their PCs. Therefore, protecting our mobile devices and the data they contain becomes as important to us as protecting our laptops and desktops. Add to this a direct connection to the Internet (very few users care about its security) and the possibility of physical loss or theft of a mobile device, and the risk of disaster is very real.

Perhaps a set of security features will help you in some way; Some of these packages today include "mobile" features, although we’re not yet sure they’re necessary.

The main mobile operating systems today are iOS, Android, Windows Phone and BlackBerry. All of them are filled with protective functions - almost all the basic concepts of information security are embodied in them, and this is not without reason, because mobile phones store a lot of our personal information. A compact device stuffed with sensors is essentially a treasure trove of vulnerabilities that attackers will try to take advantage of. Mobile devices can run many third-party applications, are used everywhere, are constantly connected to networks, and are often left unattended and lost. They need strong protection measures, but not at the expense of convenience [1].

Fortunately, there is no shortage of computer security research, and the lion's share of it applies to mobile devices. Not long ago, these devices were too slow, drained the battery too quickly, and had too little memory for advanced security features, but today multi-core processors operating at gigahertz frequencies provide ample resources for cryptography, security policy support, address space sharing, and other security requirements. architecture. In many ways, smartphones are even more secure than laptops—many of the fruitful ideas that have proven themselves in the past, such as fine-grained access control, have found their way into the world of smartphones. The OS keeps potentially dangerous applications under control and actively tries to involve users themselves in protecting their personal data [2].

Despite powerful security architectures, smartphones can still pick up malware and put personal data at risk, and compromise can occur in new ways than computers. Users, businesses, hardware manufacturers, and developers need to be aware of privacy risks and how to mitigate them. This is not to say that everyone is blissfully ignorant - according to surveys, about 60% of users deleted an application that could violate the privacy of their data or refused to install it.

Defense Mechanisms

The operating environment of a mobile device is determined by four main categories of players operating in the market: application developers, OS developers, hardware suppliers and operators, with most security measures implemented by

platform creators rightfully focused on third-party application developers. Operating systems of mobile devices deliberately consider applications “hostile” and use powerful mechanisms to isolate them from each other, as well as from the system kernel. These mechanisms (“sandboxes”) take advantage of the processor’s ability to separate task address spaces from each other through access controls implemented at the file system level, as well as cryptography to prevent applications from sharing information locally through standard means. An application may have a file zone, but only it has access to it.

However, buffer overflow errors still occur, including in browsers and other common applications. Such errors are often used as entry points by malware that can compromise a specific application or the OS. To reduce this risk, iOS, and Android use Address Space Layout Randomization technology to place various application functions in randomly selected areas of memory. This makes life difficult for virus writers, since they lose the guarantee of finding the routines they need [3].

Access to almost any services and data outside the application requires privileges, the observance of which is carefully controlled by the operating system. Typically, an application receives privileges from the user upon request (for example, a privilege is permission to work with a camera or obtain geolocation information), and in general, in a typical OS there are more than a hundred different privileges. Once permission is granted, it typically remains for as long as you use the application, but it is possible to revoke the privilege manually.

Sharing privileged information between applications violates a fundamental principle of computer security—the principle of least privilege, which states that processes and users should have no more privileges than are sufficient for the job at hand. This principle minimizes the attack surface for attackers and also minimizes the likelihood of accidental leaks.

Does all this mean that confidential user data cannot be shared without obtaining his permission? Unfortunately, information leaks are inevitable. They can be accidental or caused by an attacker since all systems have their shortcomings. Even

strict isolation has its limitations, and software developers disagree about the wisdom of involving users in decisions about how their information is stored. For example, Apple, as both a hardware and software supplier, strictly controls the applications that users can install on their devices and ensures that personal data is kept safe through checks. But the issue of some particularly important privileges, such as access to geolocation and telephone calls, is decided by the user himself during installation. The Android system, in turn, supports the principle of user awareness and transparency. This leads to certain complications associated with the fact that during the installation process the user must make decisions about all application permissions, and not just the critical ones.

Which method is more effective for protecting personal data? None are perfect, but each has its own characteristics. By studying Android and iOS apps, the researchers found that apps on Android were more compliant with the principle of least privilege than those on iOS. If the need to assign privileges will scare the user away from the application, Android program developers themselves exclude in the code the possibility of using them with unnecessary permissions.

Privacy issues

Despite all the protective mechanisms and warnings, mobile devices can compromise the data of even the most careful users. There are many sources of problems: incorrect factory software configurations, errors in applications, distribution through third-party sites, compromise by attackers, escalation of privileges and leaks because of website hacking.

Some leaks may be caused by the negligence of the device manufacturer or OS developer. For example, log files saved in plain text may contain personally identifiable information or telephone numbers, as well as geolocation data. This information is sometimes available even to unprivileged applications.

The source of the most difficult problems is data exchange: local and through external sites. It is difficult to develop security policies that allow applications to share limited data without the risk of full disclosure [4].

All operating systems give applications the ability to communicate through an interprocess communication mechanism, but despite the limitations, breakthroughs sometimes occur. Communication between applications, even under OS control, can sometimes lead to privilege escalation. If Android application A is given privilege P and application B is not, they can still establish inter-process communication. Application B could request privileged data from A, which, using its privilege P, would gain access to it and send it to Application B. Research shows that this is a very real scenario, although, in theory, the misuse of privileges by Application A should attract the attention of mobile app store operators , which, after checking, should remove the offending program from it, but this does not always happen.

There is another problem - iOS and Windows Phone allow applications that are certified by the same certificate to communicate without restrictions. There is a certain logic in this - the developer signs several of his applications with the same key since he has no doubt about their reliability. A major threat comes from the methods by which applications identify other installed instances of themselves. Games, for example, store usernames, high scores, and other information on servers on the Internet. Accidental or malicious disclosure of this data should not violate the user's privacy, but the developer can repeatedly gain privileges through individual applications, allowing them to share data, which ultimately leads to a complete loss of privacy.

The threat also comes from the negligence of the programmer - when choosing an identifier to use in the database of their server, many developers choose the one that is most convenient for them: a unique device identifier, a real username, a phone number given by the provider, etc. In general, multiple identifiers can be used for "security" and then all the developer's application data is tied to the selected identifier. It's not surprising that over time the server falls victim to malware—two independent application developers sharing the same credentials for their databases could compromise the user's privacy if both databases are compromised. By simply combining databases, you can correlate information and create a dossier on users.

Eventually it will be possible to find out their true identities, even though the developer has formally taken measures against it.

Flagrant Abuse

Here's a particularly striking example of an application that violates user privacy. Many operators believe they need detailed information about mobile network usage, including phone numbers, call and SMS times, website visits and GPS readings. It is claimed that the analysis of this data allows optimizing network performance, therefore, for these reasons, a special Carrier IQ application is installed on devices at the factory, without informing the user about it. For example, as the US Federal Trade Commission found, HTC at one time installed Carrier IQ on its Android smartphones without taking "appropriate security measures." Thus, information from Carrier IQ became available to third-party applications without the user's consent. As a result of countermeasures taken by a special commission, HTC had to agree to undergo external security audits for 20 years. In theory, this should serve as a lesson in the need to put privacy at the forefront, but only time will tell whether there will be any changes in practice.

Recommendations

Some software developers are skeptical about involving users in managing their own privacy settings, reasoning something like this: "If you ask a user to give up all their personal information in exchange for the opportunity to watch a video of a cute cat, nine out of ten will readily click the "Yes" button. However, it is user complaints that are currently holding back the wave of privacy violations, and users who care about the safety of personal data are doing an important job for the entire ecosystem of mobile devices.

Techies and tech enthusiasts can help by doing their research on the risks, but in the meantime, the least that users can do is be careful when installing apps on their mobile devices. If an app asks for permissions unreasonably or asks for too many privileges, don't install it. Read the privacy policies, and if there are any indications that your data will be used without restrictions, including shared with third parties, do

not install the application. Users should also regularly review application privileges and the amount of data they store locally. Unnecessary data should be deleted from time to time. Report any violations you find and what you think could be changed.

Developers should carefully analyze the privilege needs of their applications and adhere to the minimum privilege rule with each new release. Examine third-party libraries for privilege dependencies, and if they are not required, replace offending functions with code that does not require additional privileges. Unique identifiers should be generated from an unpredictable value stored on an external site or use a hash of the device identifier generated using unique data, such as a public key. Try to remove personal data that could "contaminate" the user database remove geolocation information from pictures before they are used in profiles or published on social networks. The principle of least privilege should be your credo.

The creators of leading mobile operating systems recognize the importance of user privacy and have invested a lot of effort into helping programmers respect it. However, not all of them adhere to the necessary rules, so the vigilance of the users themselves remains the most important condition for the secure storage of personal data.

LITERATURE

1. Website www.phonearena.com, article "Study finds that Android lock patterns tend to be too simple, just like passwords." [Electronic resource]. Access mode: http://www.phonearena.com/news/Study-finds-that-Android-lock-patterns-tend-to-be-too-simplejust-like-passwords_id72948/

2. Security Information Portal, article "Is it difficult to guess a PIN code?" dated 09/13/2012. [Electronic resource]. Access mode: http://security-corp.org/os/android/5826-slozhno-li-ugadat-pin-kod.html/

3. Online news publication RT News, article "Hackers talked about ways to steal fingerprints from Android device owners" dated 08/07/2015. [Electronic resource]. Access mode: https://russian.rt.com/article/107807/

4. Information web resource http://gsmpress.ru, article “Fingerprint scanner in a smartphone: pros and cons” dated November 11, 2016 [Electronic resource]. Access mode: http://gsmpress.ru/articlesitem/2511-skaner-otpechatkov-palcev-vsmartfone%3A-pljusy-iminusy.html#sthash.jUJWFjfX.dpuf/