НАУЧНО-ТЕХНИЧЕСКИЙ ВЕСТНИК ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ
январь–февраль 2016 Том 16 № 1 ISSN 2226-1494 http://ntv.ifmo.ru/
SCIENTIFIC AND TECHNICAL JOURNAL OF INFORMATION TECHNOLOGIES, MECHANICS AND OPTICS
January–February 2016 Vol. 16 No 1 ISSN 2226-1494 http://ntv.ifmo.ru/en

(IN-)PRIVACY IN MOBILE APPS. CUSTOMER OPPORTUNITIES

Yu.S. Chemerkin[a], T.I. Kuzmenko[b]

[a] JSC "Advanced Monitoring", Moscow, 127282, Russian Federation
[b] "InfosecService" Group of Companies, Moscow, 109189, Russian Federation
Corresponding author: yury.s@chemerkin.com

Article info
Received 23.11.15, accepted 21.12.15
doi:10.17586/2226-1494-2016-16-1-90-95
Article in English
For citation: Chemerkin Yu.S., Kuzmenko T.I. (In-)privacy in mobile apps. Customer opportunities. *Scientific and Technical Journal of Information Technologies, Mechanics and Optics*, 2016, vol. 16, no. 1, pp. 90–95.

Abstract
Subject of Study. The paper presents the results of an investigation of cross-platform mobile applications. This paper focuses on a cross-platform app data investigation in purpose of creating a database that helps to make decisions from data privacy viewpoint. These decisions refer to knowledge about mobile apps that are available to the public, especially on how consumer data is protected while it is stored locally or transferred via network as well as what type of data may leak. **Methods.** This paper proposes a forensics methodology as a cornerstone of an app data investigation process. The object of research is an application data protection under different security control types among modern mobile OS. The subject of research is a modification of forensics approach and behavioral analysis to examine application data privacy in order to find data that are not properly handled by applications which lead to data leakages, defining protection control type without forensics limits. In addition, this paper relies on using the simplest tools, proposing a limit to examine locally stored data and transmitted over the network to cover all data, excluding memory and code analysis unless it is valuable (behavioral analysis). The research methods of the tasks set in the paper include digital forensics approach methods depending on data conception (at-rest, in-use/memory, in-transit) with behavioral analysis of application, and static and dynamic application code analysis. **Main Results.** The research was carried out for the scope of that thesis, and the following scientific results were obtained. First, the methods used to investigate the privacy of application data allow considering application features and protection code design and flaws in the context of incomplete user awareness about the privacy state due to external activity of the developer. Second, the knowledge set about facts of application data protection that allows making a knowledge database to implement the missing privacy and security protection control and provide the privacy requirements (keeping the users informed about possibility to avoid untrusted usage cases). **Practical Relevance.** Practical relevance of the received results is the following: first, the set of knowledge facts about each examined application to privacy score per application, per application category (IM, travel, etc.), per OS, etc; second, the developed method under the forensics approach can be used to carry out analysis of the application data privacy in relation to the specified requirements including audit, reconfiguring EMM application policies and reasons for their commissioning.

Keywords
mobile security, mobile application vulnerability, data leakage, data privacy, EMM

Acknowledgments
The paper is recommended by the Organizing committee of the International conference "Information Security and Protection of Information Technology 2015" (http://ispit.ifmo.ru/)

УДК 004.056.53

(НЕ-) КОНФИДЕНЦИАЛЬНОСТЬ ИНФОРМАЦИИ В МОБИЛЬНЫХ ПРИЛОЖЕНИЯХ. ВОЗМОЖНОСТИ КЛИЕНТОВ И ПОЛЬЗОВАТЕЛЕЙ

Ю.С. Чемёркин[a], Т.И. Кузьменко[b]

[a] ЗАО «Перспективный мониторинг», Москва, 127287, Российская Федерация
[b] Группа компаний «ИнфоСекьюрити», Москва, 109189, Российская Федерация
Адрес для переписки: yury.s@chemerkin.com

Информация о статье
Поступила в редакцию 23.11.15, принята к печати 21.12.15
doi:10.17586/2226-1494-2016-16-1-90-95
Язык статьи – английский

Ссылка для цитирования: Чемёркин Ю.С., Кузьменко Т.И. (Не-) Конфиденциальность информации в мобильных приложениях. Возможности клиентов и пользователей // Научно-технический вестник информационных технологий, механики и оптики. 2016. Т. 16. № 1. С. 90–95.

Аннотация
Предмет исследования. Представлены результаты исследования кросс-платформенных мобильных приложений. Целью работы явилось создание базы данных (знаний) для принятия решений по их защите. Подобные решения

относятся к знаниям о доступных мобильных приложениях и информации о защищенности данных пользователей при хранении, обработке и передаче данных. Исследована защищенность приложений в рамках существующих механизмов безопасности современных мобильных операционных систем. **Метод.** Предложены модифицированные способы криминалистического подхода и поведенческого анализа при изучении защищенности приватных данных и способов поиска ошибок в реализации механизмов защиты информации, хранимой, обрабатываемой и передаваемой приложениями. Анализ механизмов безопасности рассматривается с позиции простых инструментов и исключает применение специальных криминалистических приемов. Применяемые методы опираются на известные подходы цифровой (компьютерной) криминалистики (анализ памяти, трафика, кода), включая поведенческий анализ, статический и динамический анализ кода приложения. **Основные результаты.** Показано, что предложенные методы исследования защищенности данных приложений позволяют сделать выводы о фактическом статусе защиты данных с учетом изменений, вносимых разработчиком. Набор знаний о защите приватных данных мобильных приложений позволяет в дальнейшем реализовать недостающие механизмы защиты с целью поддержания информирования о ситуациях, в которых возможно избежать прецедентов утечек данных. Набор фактов знаний о защищенности данных позволяет вводить и использовать числовые характеристики уровня защищенности приложения, категории приложений и т.д. **Практическая значимость.** Предложенный в рамках криминалистического подхода метод может быть использован для проведения анализа данных приложений в рамках аудиторских работ, реконфигурации EMM-политик и обоснования их введения в эксплуатацию.

Ключевые слова
мобильная безопасность, уязвимости мобильных приложений, утечка данных, защита приватных данных, EMM

Благодарности
Рекомендовано к опубликованию оргкомитетом Международной конференции ISPIT 2015.

Introduction

When a mobile application (further “application” or “app”) processes sensitive information obtained from the user or any other source, it may result in placing that data in an insecure location in the device. This insecure location might be accessible to the other malicious apps running on the same device, leading such device to be at a serious risk. Some of the well-known applications failed to protect data privacy are Android Mail for Exchange and Hotmail, Foursquare, and Groupon [1]. These applications stored the user passcode/password and peace of user information in plain text on the device. That was usual case for applications released around the beginning of 2011. Instagram application was vulnerable to partial eavesdropping and MITM attacks that could lead an evil user to delete or download private media without the victim consent [2]. Recently, Starbucks application was caught storing each user’s username and password in plaintext in the Crashlytics log file (/Library/Caches/com.crashlytics.data/com.starbucks.mystarbucks/session.clslog) [3]. This bug allows attackers to discover usernames, passwords, and e-mail addresses. Some mobile applications use an unencrypted Sqlite database for storing sensitive information, for example, details of customer banking account or transaction history. Some cases lead us to admitting issue by developer by continuously ignoring it, for example, AgileBits insisted that AgileKeychain was still secure, and noted that the format dates back to 2008 when the company was concerned about speed and battery drain problems caused by encryption [4, 5]. At the same time, about 30% customers prefer to uninstall mobile applications because privacy concerns (according to the ITR research results) [6]. Some other publicly available results confirm the same situation for non-official applications [7, 8]. And finally, there are situations when developers officially confirm breaking into the customer’s privacy [9].

While dealing with security issues, many researchers focus only on the data which is transmitted over the network, related to server issues or protocol issues in general (e.g. fuzzing) or memory state related interaction with an application and services through application code. Originally data forensics manages that the data could be accessed within certain limits, e.g. backups, breaking credentials, without any additional interactions with a network that may change data [10, 11]. Also, other approaches focused on code examination like static analyze – analyzing raw mobile source code, decompiled or disassembled code, or dynamic analyze – executing an application either on the device itself or within a simulator/emulator and interacting with the remote services with which the application communicates, including assessing the application local inter-process communication surface, forensic analysis of the local filesystem, and assessing remote service dependencies [12]. Also, there is a type of analysis influenced by behavioral characteristics. Frankly talking, it means finding the facts about that certain data which are stored or transmitted. All approaches and developed tools put the security flaws on the first place. Solid descriptions of different ways to access data which is stored locally, in memory or transmitted over the network were shown in A. Hoog [13 14] researches from the forensics viewpoint. A well-known, digital forensics is the use of scientifically derived and proven methods toward the preservation, collection, validation, identification, analysis, interpretation, documentation and presentation of digital evidence derived from digital sources for the purpose of facilitating or furthering the reconstruction of events found to be criminal, or helping to anticipate unauthorized actions shown to be disruptive to planned operations [15]. On the other side, well known security approaches focused on security issues irrelevant to the privacy concerns (somehow relevant but privacy is not very important in fact; security breaches and flaws are more important), focused on forensics and accessing data have a different score of complexity. Moreover, these approaches did not take the differences into account between self-developed security controls, system protection controls operating by default and system

protection controls must be properly implemented by developers. Thus we need to examine the data under different terms of protection. To reach that goal, we need to make a clear investigation to divide results in regards of protection based on type of security controls.

The goal of this paper is modification of forensics approach and behavioral analysis to examine application data privacy in order to find data which is not properly handled by applications that lead to data leakages defining protection control type without forensics limits mentioned above but focusing on behavioral analysis. In addition, this paper relies on using the simplest tools, proposing a limit to examine locally stored data and transmitted over the network to cover all data, excluding memory and code analysis unless it is valuable (behavioral analysis); according to the practical pre-research results such possibility was proven on small quantity of applications [16] [1]. Also, several independent researches were done at the same time confirming that statement [17, 18].

Related work

To perform an app investigation, we need to examine a data privacy field and particularly forensic methodologies laid under it, forensic features, protection concepts and popular solutions existed by now. Let's talk about forensics methodology. Digital forensic science is a branch of forensic science encompassing the recovery and investigation of material obtained from digital devices. It is divided into several sub-branches relating to the type of digital devices involved: computer forensics, network forensics, forensic data analysis and mobile device forensics. Focusing on forensics analysis techniques, in [13, 14] A. Hoog analyzes many Android and iOS apps. For evaluating the security of these apps, he examined privacy of app data stored locally and transmitted via network channels. Network forensics relates to the monitoring and analysis of network traffic for the purposes of information gathering in our case. Forensic Data Analysis examines data with regard to discovering and analyzing patterns of certain activities. It has a sub-branch referred to Database Forensic that is about a forensic study of databases and their metadata. According to forensics aims, it is possible to find information that is not properly handled by mobile customer apps, which in turn may lead to privacy leakage. It is important to know that the owner's private data will not be accessible for someone else by any kind of unauthorized access (physical, malicious, etc.).

Evaluation and privacy

Data privacy investigation considers data protection concepts and knowledge about mobile security controls that provide protection of data stored or transmitted in/out of the device. To perform a proper privacy assessment, it is important to know all the different states in which data can exist and can be found in a mobile environment. **Data-at-Rest** (DAR) is data recorded on internal memory or external memory like a SD card, in other words, data stored in ROM. **Data-in-Use** (DIU) is all data in the process of being created, retrieved, updated, deleted or manipulated in RAM. **Data-in-Transit** (DIT) is data transferred in a network via Cellular channel or Wi-Fi channels.

Nowadays, there are more than three classic data protection concepts. DAR is referred to as Data-in-Storage; Data-in-Memory is referred to as DIU. Each concept should provide a certain way of protection. DAR, entire app and any local data (sandbox container) should be encrypted. This helps to prevent malware and rogue apps from accessing data. DIU is about encryption & wiping in all possible cases. It helps to prevent fake or non-allowed repackaged apps from running on a device, accessing data. DIT is similar to DAR but refers to network data. This helps to protect a network channel from eavesdropping or MITM.

Our approach

To prove applicability of **forensic techniques** to perform privacy assessment in regards to apps, the forensics capabilities of existing forensic tools should be examined. There are several most popular tools to analyze the data that is stored locally on the device: a software developed by Oxygen Software, Elcomsoft, Compelson, Paraben; several companies NCC Group, Appthority, ViaForensics perform an app analysis to help filling an enterprise app markets by secured apps[2]. Here is a list of apps by categories that are affected by modern capabilities of mentioned forensics tools: Social Networks, Messengers, Business, Navigation/Maps, Travel apps, Finance, Multimedia, Device logs. In our approach, we do the following steps:

– Define a data protection concept (in that paper, only DAR, DIU, DIT were analyzed);

[1] Extended results will be available during the presentation

[2] List of different tools/services and its description/features from sources:
Oxygen Forensics Extractor, http://www.oxygen-forensic.com/en/order/oxygen-forensic-extractor,
Corporate & Forensics Solutions, http://www.elcomsoft.com/products.html,
MOBILedit, http://www.mobiledit.com/forensic,
Paraben Forensics Software, https://www.paraben.com/products.html,
NCC Blog, https://www.nccgroup.com/en/blog/ ,
APP RISK MANAGEMENT RESOURCES, https://www.appthority.com/resources ,
ViaForensics Blog, https://viaforensics.com/blog

- Define target apps to be tested. Among several different categories, we identified the most popular according to AppStore by each category;
- Data gathering and usage cases. We have to create accounts for some apps, use these apps to pass almost all of the use cases by the certain app;
- Data acquisition and Data exploration.

  "Data acquisition" step has the following sub-steps:
  - DAR: Having non-jailbroken iOS[1] device is enough to get an access for app data. Having a rooted Android device is needed to access to /data/data/. Other Android-based OS is required with a root except a BlackBerry Android Simulator that does not require root/jailbreak. Plus, some app data on BlackBerry is stored in shared folders;
  - DIU: Strongly needs a jailbroken iOS device to attach processes in real-time and investigate vars. Android and other Android-based OS require a root too. BlackBerry app is not accessible due to good protection; in case of Android app running on BlackBerry there is no difference, and this case is reduced to Android one;
  - DIT: We need to configure a proxy profile on all devices per Wi-Fi connection. Additionally, we need to install and trust to a web-proxy tool certificate to have an access for https-traffic.

  "Data exploration" step has the following sub-steps:
  - SQLite viewer is needed to simplify examination of sqlite3 databases and non-closed properly db-files;
  - PList viewer is needed to view iOS-based XML file type (plist);
  - Other file types could be opened by plaintext-viewer;
  - CharlesProxy is web debugger for http and https examination. Works as a MITM-tool;
  - Rest of tool for memory access is described in [19].

Results

Since our experiment was over, we could summarize the common data types that were found on through those paths; all information that was typed, processed by app, received via network or prepared to send was found in plaintext in memory; significant amount of data was found in network channel despite of network layer protection. These data types include:

- For **Android**: app data, binary data, data stored in sqlite db files and shared_prefs, and other misc files;
- For **iOS**: data stored in sqlite db files, keychain and plist files, binary cookies, keyboard cache, snapshot files, file cache, iOS local backup data, binary data;
- For **BlackBerry**: BlackBerry backup files, removable accessible mobile device files, removable accessible desktop files, Android internal and external data files;
- For **WindowsPhone**: local, roaming, temporary, local cache, app settings data.

App Type/Protection	In-Rest	In-Use	In-Transit
Built-in apps	Plain-Text	Plain-Text	Rarely Encrypted / SSL/HTTPS
IM apps	Plain-Text	Plain-Text	Weak Encryption
Social app	Plain-Text & Rarely Store some data	Plain-Text	SSL/HTTPS
Geo Apps	Plain-Text	Plain-Text	SSL/HTTPS
Office Apps	Plain-Text	Plain-Text	SSL/HTTPS
Travel Apps	No/weak encryption	Plain-Text	SSL/HTTPS
App with payment features	Plain Text / Weak Encryption	Plain Text	SSL/HTTPS
Bank apps	Rarely Store data / Good Encryption	Plain-Text	SSL/HTTP / Encrypted

Table. App Data Investigation Results

The investigation results are shown in the

Table full depth of research includes about 600 apps. Many apps reveal DIM (DIU concept) that is reasonable because almost of all data should be shown on a mobile device screen to user and it is stored in plaintext at a certain time if it is encrypted the rest of the time. It is not quite difficult to access the data, however an attack should rely on repackaged app (original app overloaded with malicious payload to hook and intercept system

[1] Works for iOS version less than 8.3 only. A jailbreak is required for iOS 8.3+

calls/methods). Apps usually reveal many data by storing it locally (DIR concept), but usually users did not find any data, because it was already erased from the cache. However, in some cases, it could be easily restored from .db-file as a database file which never cleans itself unless it overwrites or manually cleans itself. A huge amount of data can easily be grabbed from data transferred through the network channels (data in transit concept). Despite being encrypted, methods like https/SSL do not solve man-in-the-middle vulnerability. **iOS** specifics of forensics app investigation are described by the following artefacts. Credentials stored or transferred in plaintext locally, as well as data stored in a keychain without additional protection or encryption; data stored in SQLite databases usually are not encrypted or keys may be hardcoded too. Data is usually stored or transferred as structured file type that simplifies an analysis. In turn, signature-based encryption that helps to quickly decrypt data as well as avoiding protection mechanism in iOS that leads to pure protection eventually. Older iOS versions provide an access to data without even jailbreak. **Android** specifics of forensics app investigation are described by the following artefacts. Credentials stored or transferred in plaintext locally, OS does not provide any protection like a keychain in iOS, and stored in SQLite databases usually not encrypted for internal and external storage as well as keys may be hardcoded or put in data folder. Data is usually stored or transferred as structured file type that simplifies an analysis and signature-based encryption that helps to decrypt data quickly (depends on dynamically linked libraries). **WinRT** specifics of forensics app investigation are described by the following artefacts. Credentials stored or transferred in plaintext locally, data stored in SQLite databases usually not encrypted, and keys may be hardcoded or put in data folder. Data is usually stored or transferred as structured file type that simplifies an analysis; signature-based encryption helps to decrypt data quickly (depends on dynamically linked libraries), also apps could be analyzed on Windows 8 via known methods.

Conclusion

First many apps store data in a way that easily lead to data leakage without special tools, techniques or vulnerabilities. In this paper we relied on simple tools and approaches to find out how to access the data which is stored in work folders on apps or other places. Additionally, we described cases when data could be extracted without breaking security controls by using a root or jailbreak. **Second**, data processed by apps is protected in different ways depending on mobile OS. Also we have the knowledge about each mobile OS, and we know that the apps developed to run under the following OS:

- WinRT are the easiest for debug and memory examination, because these apps can run under full desktop OS (Windows 8);
- iOS share data that is stored in work app folders with PC which is synchronized with iPhone/iPad via iTunes. To access it, you do not need to jailbreak your device;
- any mobile OS do not provide any security controls to block a traffic interception, even https, etc.

Third many up-to-date apps still keep data unsecured. For example, App in Air app was released with a security fix and since middle 2015 for Android, iOS/iPhone but not for iOS/iPad. It proves an importance to track changes in app, about everything we talked above to make a proper decision. It is valuable for enterprise and EMM solutions. Despite relying on an encrypted app-sandbox and app-wrappers, you need to know what app store or transfer data are improper according to security guidelines and corporate IT policy. Moreover, it helps to understand what possible data leaks may happen in user app space on BYOD-devices.

Based on the results of this research, there is a serious issue of data theft whenever IT Policy is wrong, is not configured, work and user space do not exist or improperly configured, or when smartphone is lost. Some other cases may include attacks involving special forensics tools or another one to get access to a device, and data stored on it. For example, you do not need a special tool to upload a custom recovery image to Android device with an already embedded malware, with the purpose of uploading it lately or root device. In addition, restoring original recovery image will hide your tracks. Wrong network IT Policy may lead to the traffic interception, even credentials, private or bank data. Even when the device is still running, the original OS does guarantee a security level you may expect from apps, OS or enterprise solutions.

References

1. 1. Sherman E. *Want to Protect Your Emails? Don't Use these 11 Android and iPhone Email Apps*. Available at: http://www.cbsnews.com/news/want-to-protect-your-emails-dont-use-these-11-android-and-iphone-email-apps (accessed 07.09.15).
2. Reventlov C. *Instagram 3.1.2 For iOS, Plaintext Media Information Disclosure Security Issue*. Available at: http://reventlov.com/advisories/instagram-plaintext-media-disclosure-issue, (accessed 07.09.15).
3. Wood D. *[CVE-2014-0647] Insecure Data Storage of User Data Elements in Starbucks v2.6.1 iOS mobile application*. Available at: http://seclists.org/fulldisclosure/2014/Jan/64 (accessed 07.09.15).
4. Fingas R. *1Password to change file formats after key file found to contain unencrypted data*. Available at: http://appleinsider.com/articles/15/10/20/1password-to-change-file-formats-after-key-file-found-to-contain-unencrypted-data (accessed 23.09.15).

5. Beekhuis W. *Misleading Headline Popularity Rises 200%.* Available at: http://timedoctor.org/2015/10/misleading-headlines-popularity-rises-200 (accessed 23.09.15).
6. Ahmad I. *Why Do People Uninstall Mobile Apps?* Available at: http://www.digitalinformationworld.com/2015/09/infographic-why-mobile-apps-are-being-uninstalled.html (accessed 23.09.15).
7. Unuchek R. *Stealing to the Sound of Music*. Available at: https://securelist.com/blog/incidents/72458/stealing-to-the-sound-of-music (accessed 07.09.15).
8. Clover J. *Malicious App 'InstaAgent' Sends Instagram Passwords to Unknown Server, Posts Spam in Users' Feeds*. Available at: http://www.macrumors.com/2015/11/10/malicious-instaagent-instagram-app (accessed 07.09.15).
9. Grachev E. *Viber Moved their Servers to Russia*. Available at: http://appleapple.top/viber-moved-their-servers-to-russia (accessed 07.09.15).
10. Egele M., Kruegel C., Kirda E., Vigna G. Pios: detecting privacy leaks in ios applications. *Proc. 18th Annual Network Distributed System Security Symposium, NDSS'11*. San Diego, USA, 2011.
11. Schrittwieser S., Fruehwirt P., Kieseberg P., Leithner M., Mulazzani M., Huber M., Weippl E. Guess who's texting you? Evaluating the security of smartphone messaging applications. *Proc. 19th Annual Network Distributed System Security Symposium, NDSS'12*. San Diego, USA, 2012.
12. *OWASP Mobile Security Project, OWASP*. Available at: https://www.owasp.org/index.php/OWASP_Mobile_Security_Project#tab=M-Security_Testing (accessed 07.09.15).
13. Hoog A., Strzempka K. *iPhone and iOS Forensics Investigation, Analysis and Mobile Security for Apple iPhone, iPad and iOS Devices*. Waltham, Syngress, 2011, 336 p.
14. Hoog A. *Android Forensics: Investigation, Analysis and Mobile Security for Google Android*. Waltham, Syngress, 2011, 432 p.
15. Palmer G. A Road Map for Digital Forensic Research (DFRWS). *Technical Report DTR-T001-01* Final, Air Force Research Laboratory. Rome, New York, 2001.
16. Chemerkin Y. *Mobile Hacking. EMM Limits & Solutions*. 2014. Available at: http://www.slideshare.net/EC-Council/hh-yury-chemerkin (accessed 23.09.15).
17. Shankland S. *Researchers find data leaks in Instagram, Grindr, OoVoo and more*. Available at: http://www.cnet.com/news/researchers-find-data-leaks-in-instagram-grindr-oovoo-and-more (accessed 07.09.15).
18. Muntaha M., Su J., Ahmad F. *Another Popular Android Application, Another Leak*. Available at: https://www.fireeye.com/blog/threat-research/2015/08/another_popular_andr.html (accessed 07.09.15).
19. Xiao C. *iOS and Android Tools for Dynamic Analysis*. Available at: http://wiki.secmobi.com/tools:android_dynamic_analysis (accessed 05.09.15).

Yury S. Chemerkin – researcher, JSC "Advanced Monitoring", Moscow, 127287, Russian Federation, yury.s@chemerkin.com

Tatiana I. Kuzmenko – Head of Department, "InfosecService" Group of Companies, Moscow, 109189, Russian Federation, tata.kuzmenko@gmail.com

Чемёркин Юрий Сергеевич – исследователь, ЗАО «Перспективный мониторинг», Москва, 127287, Российская Федерация, yury.s@chemerkin.com

Кузьменко Татьяна Ивановна – руководитель направления, Группа компаний «ИнфоСекьюрити», Москва, 109189, Российская Федерация, tata.kuzmenko@gmail.com