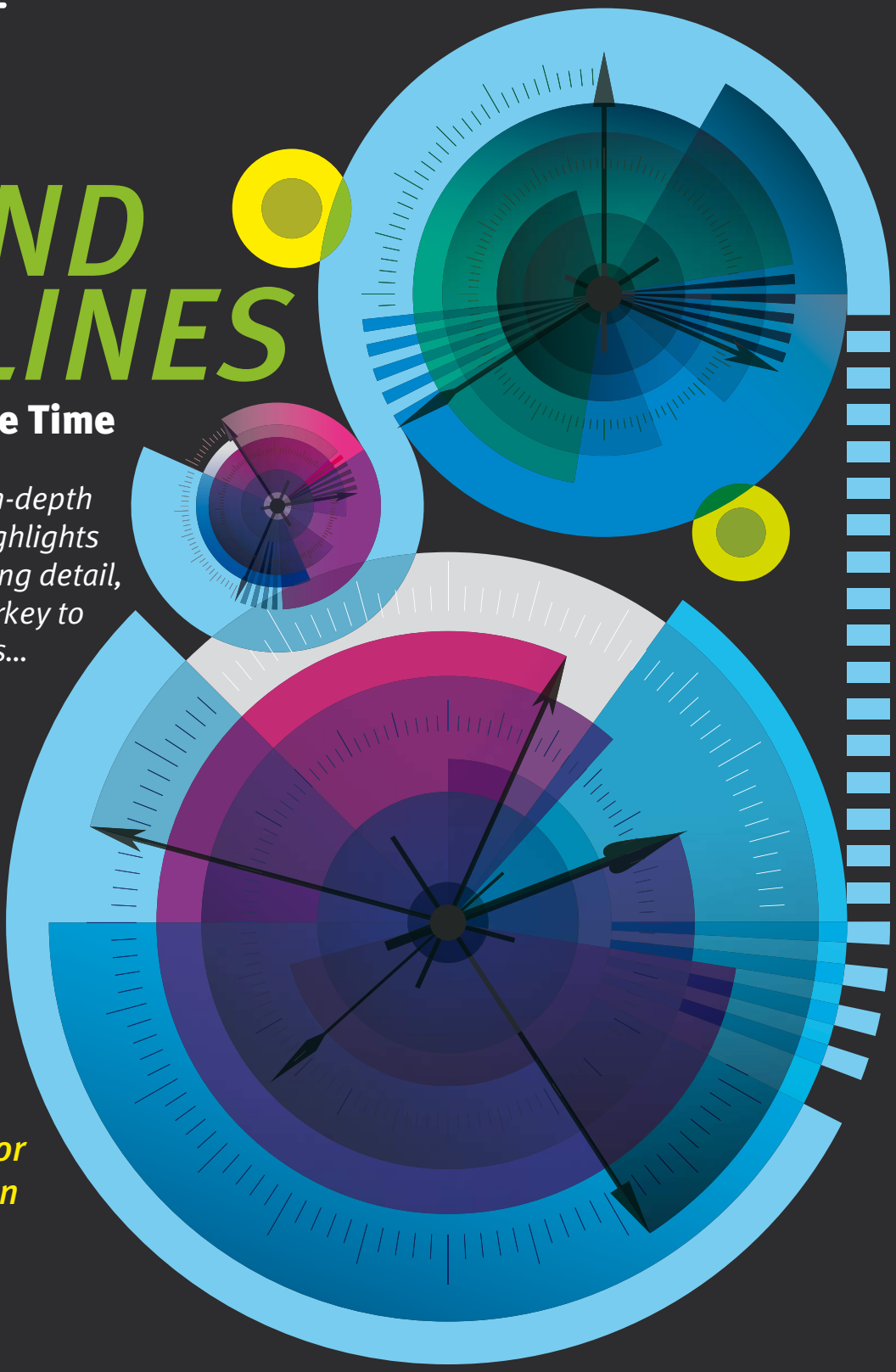
The Quarterly Magazine for Digital Forensics Practitioners

Issue 18 • February 2014

DIGITAL FORENSICS / MAGAZINE

WIN! an iPod Nano



BEYOND TIMELINES

Anchors in Relative Time

Mark Spencer takes an in-depth look at timelines, and highlights the importance of checking detail, using a recent case in Turkey to demonstrate the dangers...

Latest News, 360 Book Reviews, IRQ
& much more inside!

PLUS!

Forensic Readiness
Malicious use of Android Permissions
Using Fuzzy Hashes for Malware Classification

18
9 772042 061004
Issue 18 / £14.99
TR Media

ANDROID SECURITY: MALICIOUS USE OF ANDROID PERMISSIONS

Nikola Milosevic explains a threat model for analysing android security, and takes a look into android permissions, investigating how they may be used for malicious purposes.

/ ENTRY

Google created a reasonable security model in developing android by getting the best from the Linux user model and advancing it. The Linux security model is based on the user with each user having a set of privileges, or things that a user is allowed to do on the operating system.

Android is built on top of the Linux kernel and uses the Linux features, but in an advanced way. Android creates a unique user for each application that is installed on the system. For every application, that the user installs, the user is prompted to accept a set of permissions that the application requires. A user may of course refuse to accept the permissions resulting in the application not being installed on the user's device.

Using this model the user is warned when installing the applications, what the application will try to do with system functions and with the operating system. For example, if the application requires permission to send SMS messages and the application is a simple game, this might be suspicious and most likely the game is a Trojan that sends premium SMS messages, at a cost to the user. However, the game might not have malicious features if it uses premium SMS for billing of premium items as in game purchases. However, as the user is warned that the application might send SMS messages it is left to user to decide if they will take the risk.

Since for each application the user is prompted to allow permissions, users most often fail to even read what the requested permissions are or relate to. Quite often, users accept everything without reading. This is where the android security model fails; in addition, users are quite often uneducated about risks that relate to certain permissions. In this article we will investigate some of the most common combinations of permissions that may be used maliciously.

/ ANDROID SECURITY MODEL

As described android uses the Linux kernel and at its heart is the Linux security model with users and permissions. The difference being that android creates a new user for each application with the user having a set of permissions that the user accepted allowing the application to access areas on the device's memory. When the application is in installed in the device's memory, it has by default access to read and write to the folder /data/data/app_package_name/files/.

/ PERMISSION CLASSES

Android classifies permissions in four classes:

- Normal
- Dangerous
- Signature, and
- Signature-or-system.

In fact, signature and signature-or-system android permissions cannot be used by custom applications, since only applications signed with the same private key as the operating system are able to use those permissions.

The permissions are set in such a way that no other application is allowed to access this directory. The directory is configured so that only processes with the application's UID (user ID) are able to manipulate this directory on the file system. If an application asks permission, it may also write to other locations such as an external SD card, however any data written to an external location lacks the Linux access based control, thereby allowing other applications to read these files. Developers however, may set files within the application's private directory to be publicly readable, writable or both, by setting the appropriate flags.

Similarly, SQLite databases are stored in private directories on the android platform. An SQLite database is data storage that emulates the behaviour of an SQL database, but all the data is stored in a single file. This file is stored under /data/data/app_package_name/databases/. The SQLite file also has a set of default permissions that only allow processes with the application's UID to manipulate it.

The third data storage area on the android platform, shared preferences, works in the same way as SQLite or files. It stores key-value pairs in XML file under /data/data/app_package_name/shared_prefs/.

By default an android application only has access to these three data stores. In addition the application by default may not access the Internet, access SD card files, send SMS, perform a call, etc. For all of these actions there are permissions that the developer has to ask for and the android user has to accept when installing the application.

/ DANGER CLASSIFICATIONS ON MOBILE DEVICES

Developers are able to add permissions and access features of the mobile device if needed; these features and permissions most often add value to the application. Users are able to store large amounts of data on an SD card so they can use all the sensors their device has, for example they may obtain their location on a map or obtain contact information allowing them to connect with friends using the Internet.

However, developers, using the same permissions that create value for the applications, may also create malicious applications that can harm users or their devices. There are several known malicious behaviour patterns that almost all malware use. Applications that don't use any permissions could be considered safe. Unfortunately, however, permissions that can harm users' devices may also be used in a legitimate way. The only solution for checking if an application is malicious or not is to carry out a sanity check. The user should read carefully what the application does and check the permission list to identify if the application really does need the permissions to work properly. Many users don't know much about the underlying technology being used and what permissions are needed for certain features. It is therefore extremely important to educate users about how certain permissions may harm their data, money and privacy.

/ 120 PERMISSIONS

Currently, Android defines 120 permissions, where each permission is related to a specific device resource or to a critical operation that may be exploited to harm the user privacy, her money, or the device itself.

"ANDROID IS BUILT ON TOP OF THE LINUX KERNEL AND USES THE LINUX FEATURES, BUT IN AN ADVANCED WAY."

/ PERMISSIONS THAT CAN HARM THE DEVICE

The first malware in the history of computing tried to harm the victim's computer, latterly the destructive kind of malware is not so often found, but when it does surface the impacts may be significant, such as Stuxnet, the intent of which was to damage machines used for uranium enrichment.

In the mobile space this type of malware is still quite rare. This we believe is because causing damage to victims is not main goal of the attacker. The attacker is more interested in the data a victim possesses, than in destroying the victim's mobile device or causing some serious error. Still, malicious applications maybe available on the market that will harm victim's device and there are several permissions available that could be used.

There is a command that disables the device. This is the "BRICK" command, and has associated brick permissions. Fortunately this permission cannot be used by third party applications, as it is a system level permission. In order to use this permission, applications have to be signed with the same signature as the operating system. Fortunately the android developers thought about this and disabled usage of this command by third party applications.

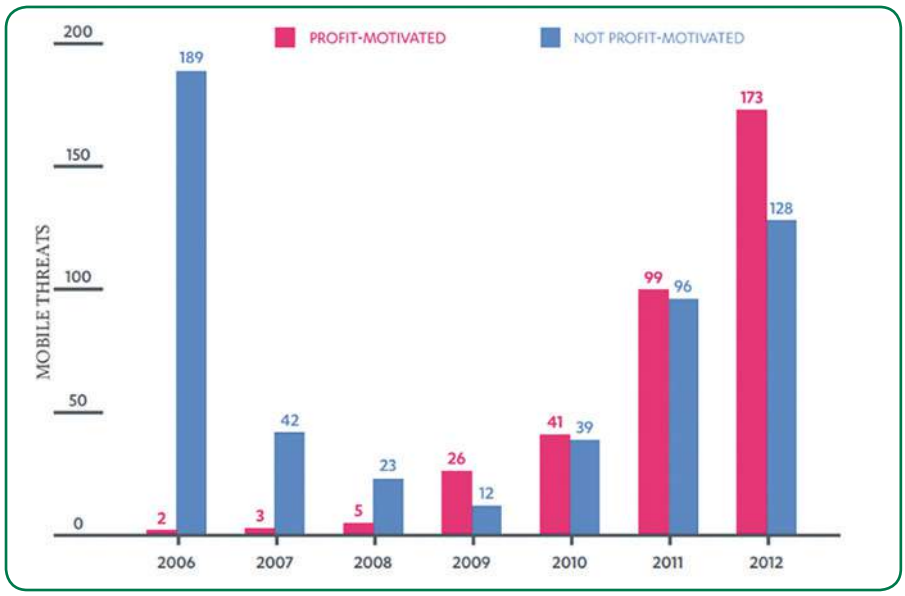
There are more permission's that are not available to third party applications, so the most sensitive parts of operating systems and hardware are protected. However, there are some permission's that can harm the functionality of other applications, operating systems or hardware sensors. The CHANGE_NETWORK_STATE permission for example, can change whether or not the device is connected to a network. Similarly, third party applications may change the WIFI state. Programmers are also able to disable KEYGUARD, this will keep the android device unlocked and unprotected. This may cause unwanted calls from the victim's pocket etc.

Permissions that may harm the functionality of the mobile device include MODIFY_AUDIO_SETTINGS or SET_TIME_ZONE, these permissions may affect the usability of the device or impact other applications. Writing external storage and writing contacts permissions may result in a number of affects being realised including harming the actual memory of the device; many writes and deletes may break memory segments for example, in addition an application may fill the device's memory storage such that a victim would be unable to add more data or install required applications. By adding contacts malware could trick the user into calling unwanted numbers, change phone numbers of certain contacts or by adding contacts malware could fill the space for contacts on a SIM card.

Consider how annoying it might be if malware were able to kill processes, for example, consider an application that is some kind of task manager and it used the permission KILL_BACKGROUND_PROCESSES, SET_WALLPAPER or VIBRATE. These are not necessarily malicious and would not do much harm, but it could impact the usability of the device.

/ TOP TIP

Carefully examine what permissions the application that you are about to install uses. Check those permissions with the feature list. If an application requests permissions that are not needed for any of the application's features, the application should be treated as suspicious.



F-Secure 2006 - 2012 Mobile Threats Motivated by Profit

/ COSTING MONEY

Currently the major focus of malware developers has changed to developing malicious applications that are able to generate revenue, with a huge underground economy behind the development of android malware. There are actually two models of earning money from developing malware. The first is to generate revenue directly, for example by calling a premium number, sending premium SMS/MMS, or executing some purchases with a credit card. The other one is by stealing and selling user data on the black market. This second model will be covered later.

Most often android malware comes as Trojans pretending to be a useful application, they can mimic some game, productivity application or other type of application whilst in the background they are actually generating revenue for the developer/attacker.

Permissions that are considered dangerous in this category are CALL_PHONE, PROCESS_OUTGOING_CALLS, SEND_SMS, SEND_MMS and WRITE_SMS. The CALL_PHONE permission is used to call a phone number and is one of the most used permissions used in revenue generating malicious applications.

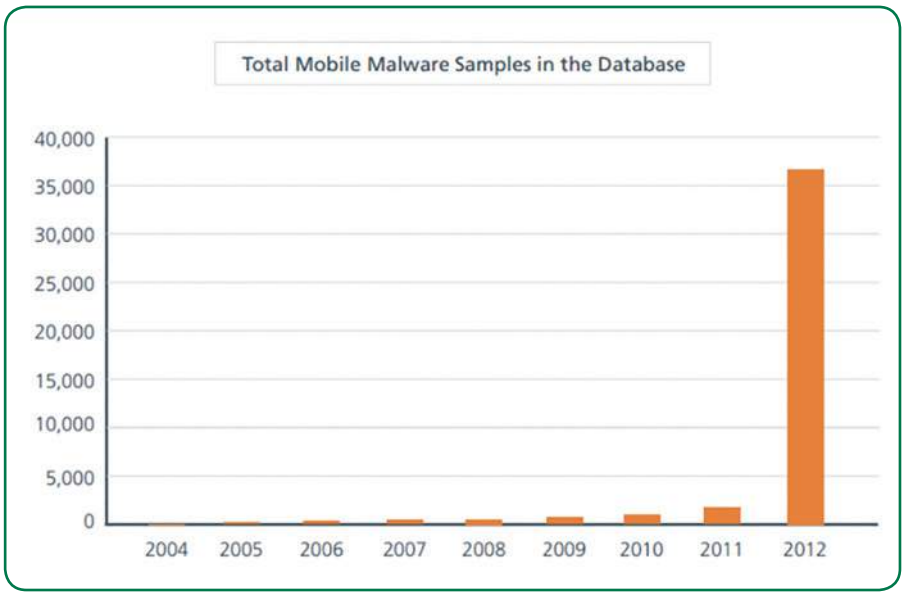
Malware will just randomly or on a certain event, call a premium number. The PROCESS_OUTGOING_CALLS permission is able to intercept an outgoing call and change it. This could also be used maliciously to redirect all calls to some premium number.

Sending messages, ether SMS or MMS could also cost the victim money. For this there are the SEND_SMS and SEND_MMS permissions that the attacker could use in his application and send messages to premium numbers from which he gets the revenue. There is also the permission WRITE_SMS, which may be used for writing SMS, however the SMS may not be sent without the user's confirmation. It is safer if an application has this permission, but the victim could still be tricked to execute the sending of the message.

/ STEALING DATA & HARMING VICTIM'S PRIVACY

Malware that steals user's data are the second model of malware that is revenue motivated, applications that steals user's data can harm a victim's privacy, it may just steal data from a victim's memory cards (like documents and files) or it may be used for espionage on either the victim or the victim's organization. In order to send data to the attacker an application that steals user data almost always uses the Internet permission. The Internet permission is not dangerous by itself, it allows access to the Internet for applications, but can be used for transfer of personal data, documents or files of the victim.

The permissions most often used for stealing victim's data are GET_ACCOUNTS, MANAGE_ACCOUNTS, MANAGE_



Android Malware: The Rise

"CURRENTLY, THE FOCUS OF MALWARE DEVELOPERS HAS CHANGED TO THAT OF CREATING MALICIOUS APPLICATIONS THAT CAN GENERATE REVENUE FOR THE ATTACKERS."

DOCUMENTS, and READ_EXTERNAL_STORAGE. The GET_ACCOUNTS permission allows access to the list of accounts in the accounts service. This permission is a lower risk permission than the MANAGE_ACCOUNTS permission, as when using GET_ACCOUNTS, an application can just get the basic properties of an account, such as user name. However using the MANAGE_ACCOUNTS permission an application can do much more and is able to do all the needed actions with existing accounts and to add new accounts. There is, however, a security measure that Google put in place to prevent stealing of account data; an application may only delete/modify an account it created itself, an application may of course create any new account and manage that. The USE_CREDENTIALS permission is used to log into an account, in most cases, "credentials" just means the corresponding authenticator creates a fitting token and hands that over (though, how to deal with that is left to the authenticator). When using an account for the first time, the Account Manager should make sure that the user is asked whether he permits this. The MANAGE_DOCUMENTS permission allows an application to manage access to documents, usually as part of a document picker. This permission can give access to locations of documents that can be then read. For reading the READ_EXTERNAL_STORAGE permission is used.

There are also some other permission's that can harm a victim's privacy. Applications could given permission to read SMS, process calls, read the calendar, call logs, read profile, social stream, record audio and video. Also an attacker may get a list of running tasks or history bookmarks from the browser. To be able to send data to the attacker again an application needs to use the Internet permission. For reading SMS there are two permissions that are required, the READ_SMS permission, which allows access to SMS log and RECIEVE_SMS, which enables the application to be notified and read the incoming SMS message.

Similarly for calls, there are permissions that allow reading of the call log, READ_CALL_LOG and the permission that can process outgoing calls, PROCESS_OUTGOING_CALLS. The READ_CONTACTS permission allows an application to read the list of contacts from the mobile device and the READ_PROFILE permission allows an application to read the user's personal profile data.

In addition there are permissions for recording video or audio. This permission is not often used for stealing data or harming victim's privacy, but there are reports that especially military and police services created and used malware that was able to record voice or video to help them in investigations.

CONCLUSION

Google has made a great effort to secure the android operating system. Many of the permissions that can harm users are not easily used in applications. All other permissions have to be requested by the developer in a manifest file and when the user installs the application, the user has to read and allow the application access to the permission's. There is a trend, however, that user's are not reading what permissions they are giving to the applications they install. Because they don't know what a harmful thing some permissions may do they can easily become victims and their data stolen. It is very important to make more effort on educating users about android security and the permission model.

REFERENCES

1. Jeff Six (2011.), Application Security for the Android Platform, O'Reilly Media
2. Google (2013.), Manifest. Permission's class reference, http://developer.android.com/reference/android/Manifest.permission.html

AUTHOR BIOGRAPHY

Nikola Milosevic was born in Bratislava, Slovakia in 1986. He finished his bachelor and master studies at the School of Electrical Engineering, Department of Computer Science, University of Belgrade (Serbia). Currently, Nikola is a PhD research student at the University of Manchester, School of Computer Science. He is also a project leader on OWASP SeraphimDroid project and he was a founder and leader of OWASP local chapter in Serbia. Nikola has also a professional experience as an android and web developer.