Environment. Technology. Resources, Rezekne, Latvia
Proceedings of the 10th International Scientific and Practical Conference. Volume III, 54-64

The Myths about and Solutions for an Android OS Controlled and Secure Environment

Imants Gorbans[1, a], Ivans Kulesovs[1, b], Uldis Straujums[1, c], Jānis Buls[2, d]
[1]Faculty of Computing, University of Latvia,
Raiņa bulvāris 19, Riga, LV-1586, Latvia
[2]Faculty of Physics and Mathematics, University of Latvia,
Raiņa bulvāris 19, Riga, LV-1586, Latvia
[a]Imants.Gorbans@lu.lv, [b]Ivans.Kulesovs@gmail.com, [c]Uldis.Straujums@lu.lv,
[d]Janis.Buls@lu.lv

Abstract. **Android is today's most popular mobile operating system for both smartphones and tablets. This fact creates many risks which are not fully recognized. Even advanced users often naively think that by using antivirus software, a firewall, encryption and updates, as well as avoiding potentially risky sites and applications they will be secure. This list is not exhaustive, but nevertheless, in most cases, each item in it only provides the illusion of security. The authors have summarized and pointed out several actual Android security issues and have proposed a number of possible solutions.**

Practical experience as well as direct testing reveals that some Android applications may contain malware. The harmful characteristics of an application often become visible only after it has been run a few times, after an update, or after harmful web content has been downloaded and shown by the application. It has been observed that applications often try to get unauthorized or inattentively authorized access to user data and to send it outside the device.

The situation with Android applications is getting more and more out of control. The authors have proposed a solution for overcoming security issues, while respecting the latest Google solutions. The target group of the proposal is users who use a smartphone or tablet both for private and corporate needs, i.e. a Bring Your Own Device (BYOD) case.

The authors point out and compare four possible Android technical administration solutions based on the unified model for a BYOD case. The authors also propose changes to Android architecture to enhance its security. A look at the mobile operating system, as a web server, has been proposed. Such a principle allows the implementation of a number of security principles taken from web servers solutions.

Keywords: **Android, mobile computing, security, BYOD, smartphones, ICT.**

I INTRODUCTION

Most of the publicly available security manuals for users are quite simple. That is why they usually push users even further away from reality. The advice about asking an ICT specialist for support doesn't always help either, because specialists are often only partly informed. We will be focussing on Android OS, because it dominates about 81% of the world's new smartphone market [1]. We will take a look at the myths or security expectations and into the solutions which may possibly reduce security risks. The authors will focus on users of Android devices who use smartphones or tablets for both private and corporate needs, i.e. a Bring Your Own Device (BYOD) case.

The oldest threat to mobile communication is the interception of calls, but nowadays new threats are coming to the foreground for smartphones. Smartphones already have most of a PCs' functionality, and therefore – the same issues. But smartphones still do not have comparable security solutions to PCs. The everyday user can still only partly reduce the threat of call interception, because it depends mostly on the service provider. But he can choose the right phone with OS and SIM supporting LTE encryption [2] and make important calls only in the 4G network. The call from the phone to the tower will then go only through an encrypted channel.

Another solution is to use additional encryption software at both ends of the call. But such a solution also has an additional technical and legal complexity

ISSN 1691-5402

DOI: http://dx.doi.org/10.17770/etr2015vol3.184

that is outside the scope of this publication. It is worth mentioning that it is the tower which decides on the call encryption mechanism. It means that the mobile service provider can completely disable the encryption [3], without the phone user even being notified. This mostly ends the discussion on the topic of voice calls.

It is possible to get even more information through a smartphone's internet connection than from calls. And there is not enough best practice or verified guidelines on configuring and administering Android devices for a BYOD case. That is why the symbiosis of these two issues is the main topic of this publication.

It is often said that Android is secure, because it is Linux. But, unfortunately, this is not the full Linux in terms of security solutions. There are more than a hundred academic publications and even more technical articles and user manuals on Android security. The number is quite large, because Android began in 2008, but its security issues were only recognized in 2010.

There are more than 18,000 different models of Android devices in the world [4]. OEM and mobile service providers increase the fragmentation of Android even more. The devices running the same Android version are different in most cases, because OEM and mobile service providers do modifications to the OS. Google allows and even encourages this in order to get new ideas for future development.

This policy creates issues for businesses if they want to choose Android as the main platform for their corporate smartphones. According to Gartner's study, less than 10% of companies plan to use Android devices for business purposes [5]. The lack of focus on security, as well as limited user management capabilities, are mentioned as the main reasons for the decision. But the position could change in the future. In the beginning, Android creators did not think about security. The main focus was on the maximal functionality and availability [6]. The situation in the security field is getting better in quite a rapid manner. This is already visible with the most current versions of Android (4.4 and 5).

II THREATS

A. Common Threats to Mobile Security

There are four main types of malware types for mobile devices [7], [8]:

1) Data thieves are the most popular malware. They try to get information about the OS version, product ID, IMEI number, and IMSI number of the infected device. This information can be used for more direct attacks in the future.
2) Rooting-capable malware infects the device in order to get the administrative permission. This allows remote users to access the device's RAM and other resources, e.g. the microphone.
3) Phishing malware. The SMS, or MMS, or email is sent to the infected device and the owner gets subscribed to pay services when opening it or following the link. Of course, the user is not notified. Calls to high-cost services can also possibly be made, or sensitive data sent to third parties while the user thinks that he is communicating with trusted sources. The phishing problem is especially real on Android because of the openness of the platform resulting in easier creation of malware. But, smartphones and tablets are usually connected to purchase and/ or payment systems.
4) Mobile spyware monitors a variety of information which is stored on an infected device like the current location, stored SMS or emails. This type of malware also sends data to third parties through any available channel in the same way as data thieves do. But, spyware concentrates mostly on personal user data retrieval.

The malware can appear in Google Play despite all preventive measures being taken [9]. It is known that there are fake app stores that replicate very similar content. These stores are made by cyber-criminals in order to fool users and provoke them into installing the malware. The fake stores can appear on a device after some harmful app installation or after some illegal Android version update. There are also apps that fake internet banking apps or try to retrieve user financial information [10], [11]. Statistics shows that 79% of all malware attacks are focused on Android OS [8]. Even if the malware does not steal sensitive enterprise data, it is not appropriate on the device used for business, because the device can deny service when it is needed for fulfilling some critical business task.

According to unofficial, but reliable information, about 44% of Android devices still use Android versions from 2.3.3 to 2.3.7 that have significant security vulnerabilities [8]. These vulnerabilities have already been removed in the latest Android versions. Mobile devices that are older than two years cannot receive the security updates, because often manufacturers do not support them. Many manufacturers already drop the development and support smartphone OS after 12-18 months of release. The only option is to install a newer Android version unofficially, if possible. But this option can often break the warranty. Google does not recommend that manufacturers produce devices with previous Android versions a year after the next Android version is available.

ICT specialists and advanced users can do unauthorized modifications to mobile devices, e.g. "rooting" of Android devices and "jail breaking" of iOS devices. These modifications are done in order to

get rid of operating system limitations. This allows the adding of extra features to the device and to install apps which are not allowed. This also changes the security management principles on the device increasing security vulnerability risks e.g. fake app stores.

Unauthorized modification can also be done to unlock the device from the specific carrier or to install additional security features like a firewall. But in most cases this is done specifically to install apps which are not allowed or/ and pirate apps. This increases the risk of infecting a device with malware, as soon as the built-in app check process is removed or completely modified. The malware now has a greater ability to access system resources and data using administrative permission, while working undetected in the background. This can also deny the ability to get security updates from the manufacturer.

It is possible to limit account permission through installing additional administrative tools on the rooted device. But the user should be aware that the device warranty is lost and the phone can become a "brick" if rooting fails. **That is why we do not recommend rooting the device as the solution for all organizations** in order to increase system security. But this can be an exclusive solution for some special case. In the course of this study, the authors have rooted several mobile devices using the Kingo Android Root or Kingo ROOT free software [1]. This software is thought to be among the most trusted rooting solutions. It allows the installation of such apps as SuperSU, Xprivacy, etc. We gained quite interesting results when using the Xprivacy app. The solution follows the actions of each app on a device in a relatively reliable way. It appeared that many free apps that were downloaded from Google Play do unauthorized actions or actions unwittingly authorized by the user with the user's data, and try to access web resources without any notification or disclosure of what is being sent out. They can sometimes begin to perform in this type of offensive manner, not during the first run, but after several runs, or after apps are updated. Using Xprivacy, user can follow the apps' actions, and can partly or fully block them (both actions and apps), but this solution is not for everyone. Such functionality would be very useful to include in the OS built-in toolbox.

One can rebuke a user for allowing an app to needlessly access data, a network, the GPS, etc. But, as already stated above, a user acts in good faith and the desired app just wouldn't even get installed without providing the relevant permission. Some apps ask for additional permission after an update, and users often, without even reading the question, allow these. The apps that show web content within them are very unreliable. These apps are not harmful in themselves, but they can download and execute some harmful scripts afterwards.

Some apps search for specific information within the user and device data, upload it, try to intercept the data stream, change the sensitive data, etc. One also can rebuke a user for installing some cat petting games, but how can one rebuke users of devices where the Chrome browser switches on a microphone which may be used to stream discussions that take place near the phone. This shows that even the current user practice of Android may be unacceptable and hopeless.

It is obvious that there is only one way: forbid BYOD users from installing additional apps on a mobile device which are able to work with company emails and/ or documents. This does not mean that all apps are malware. This means that only verified apps should be installed on an employee's phone and only by the company ICT specialist (network administrator). But then, why should a user bring his own device, if he cannot install apps for his own entertainment? There should also be an option for a browser to erase all data when it is closed.

B. Security Options in an Android OS

It is self-explanatory that authentication mechanisms as well as the encryption of data storage can help to prevent user data from falling into the wrong hands in a case of theft. The following authentication (screen unlock) mechanisms are available in the Android system: a) a combination drawing on the screen; b) entering a numeric PIN code; c) entering a password; d) biometric recognition, e.g. face recognition that is available from the Android 4.0 version (probably, not the best solution). The authors' recommendation is that **the best method is still a complex password that consists of small and capital letters, digits, and special characters and is at least 14 symbols long**.

There are smartphones with double authentication or with repeated authentication if it is not used for some period of time. It is possible to block an Android phone for some period of time when an authorization with the wrong passcode or password takes place. Five incorrect attempts at unlocking a device are possible by default. This blocks the system for 30 seconds. Data storage decryption with a wrong password also blocks the usage of a device for 30 seconds, but 10 wrong attempts are allowed. The recommended policy from PC world would be 5 incorrect login attempts that block a device for 15-30 minutes. But, there are no standard user settings for changing it in Android and we have to look for other solutions.

The next level of Android security allows a configuration of the system to erase all data if the device does not authenticate within a prescribed network within the prescribed period of time [12]. In the latest Android versions, there is a feature which

[1] - www.kingoapp.com/android-root.htm

searches for the lost device, wipes out all of the data and makes the device reset to the factory settings remotely. It is possible to perform a wipe and reset from an Internet portal or by SMS from the registered number if the feature is activated.

One can't overcome the fact that information can be intercepted. It is important that intercepted data cannot be practically decrypted, if the encryption algorithm is unknown, and the key is sufficiently long. That is why one of the most important mobile communication environment criteria is that data decryption is undertaken only by the data receiver. Otherwise, the encryption key is known by a third party which increases the probability of a data leak. If a decryption is made on some proxy server or by some other device at any other transmission step, then it is possible that the decrypted information can be intercepted. This is very important when data is transmitted through different carriers. If there are several different carriers, then the data security control is lost. The same applies to non-encrypted data transmission or when the encryption level is insufficient, i.e. when data fragments can be changed (e.g. a bank account number). The parties that are involved in the communication should be notified that the received message was not changed during the transmission.

Usage of public wireless networks should be minimized. Only new encryption types should be used, i.e. WPAv2 and 802.1x, while WPAv1 and WEP should not. The usage of Bluetooth outside trusted location should be minimized. Only trusted Bluetooth counterparties should be allowed, while others should be denied.

Only the current versions of all software, with the latest security updates, should always be used.

It should be understood that encryption of data storage is not a panacea, but is highly recommended. It will secure data only in cases where the switched off or pass-locked device is stolen. When a user has already entered an encryption key upon starting a device, then all of the activities occur in a non-encrypted way and do not prevent apps from stealing data.

The following issues were discovered whilst testing built-in data encryption tools on Android 4.2 and 4.4 which were installed on a Samsung Galaxy Tab 2 Mini: it wasn't possible to encrypt a device with another password after the previous encryption was removed. The UI was in the Polish language despite the fact that the device was used in only the Latvian and English languages. This should be checked on other Androids. However, it is clear that these solutions still have to develop.

The following useful security configuration options are available through regular Android settings: encrypt data storage, enable remote device blocking and data wipe, setting the device password, change enabled setting "Make passwords visible" by default, disable apps installations from untrusted sources etc. By installing the AppLock app, it is possible to force prescribed apps to run only after entering a passcode.

From the authors' point of view, it would be nice to have better security configuration options in the Android system, like the ones available in the latest Windows OS [13], [14]. Examples of configuration options could be as follows: the number of incorrect unlock/ decryption attempts, the time for which a device should be blocked after a wrong attempts limit is reached, only allowing one to open prescribed URLs, limiting the types of WiFi/ Bluetooth networks, importing security policy templates, choosing a security policy level, viewing logs, etc.

III ARCHITECTURE

A. Main Principles of the Android OS

The main principles of the Android OS are [15], [16], [17]:

1. Android is a processor independent operating system. But, it uses some device specific security features, like ARM v6 eXecute-Never that ensure the separation of user data from processor instructions inside the device's memory.

2. Android has been developed based on the Linux kernel. All device functions, e.g. the camera, GPS, Bluetooth, voice and data transfer is performed using the operating system, not the firmware.

3. Android apps have been developed mostly in the Java programming language and run inside the Dalvik or ART (starting from version 5.0) virtual machine. But many apps, including Android core services, use core libraries. Both, virtual machines and native apps run within the same secure environments – app sandbox that isolates app data and its code execution from other apps. An app gets the prescribed part of a file system to store its data. However, if an app has the appropriate permission then it can access all the device memory (including the SD card). The latest Android versions have an additional layer called SEAndroid that checks all the installed apps on the kernel level [18].

It is possible to create hybrid apps in Android using the WebView component that supports TML, CSS, and JavaScript technologies. These apps are similar to native apps, but they work as web sites with additional options to use the device's camera, accelerometer, etc. Unfortunately, this solution is potentially the most risky, which is why it is not advisable to install such apps from untrusted developers. But it could be a convenient way to develop internal enterprise apps.

B. The Layers of Android Architecture

Android architecture consists of several layers that work one on top of another. The lower layers provide the services for the top levels [15]:

1. The Linux kernel is the basis of the whole system. The Linux kernel enables Android to manage the hardware, memory, security settings, network protocols, and other low level functions. Users and developers do not access this layer directly. This is the layer where the hardware drivers are executed. The basic separation between the apps is also executed at this layer.
2. The core libraries ensure that basic services are available for apps. These libraries are written in C/ C++ language and vary depending on the device hardware. These libraries run as processes inside the Linux kernel.
3. The applications framework layer ensures that the core libraries and virtual machine interfaces are available to the apps.
4. User works at the highest level are called the applications level.

Android apps are divided into two parts:

a) the apps installed by the user;
b) the apps installed by the manufacturer.

It is worth mentioning that the open source Android operating system contains a code from at least three different sources: a) the Android open source project (Android version by Google); b) modifications from mobile device manufacturers; c) third party apps on the market.

C. The Main Security Features of the Android OS

Android has the following main security features:

1. The mandatory sandboxing of every app. Android uses the Linux Mandatory Access Control (MAC) mechanism (it will be described in detail separately) to force apps to work in a sandbox mode that is a part of the SEAndroid solution. A user unique identifier (UID) is assigned to each app during the installation. All executed app processes are attached to this UID. This allows a system to control access to low level resources. The private data storage inside the internal memory is assigned to each app according to this mechanism.
2. Access permissions. Permission labels are assigned to each app. They are displayed during the app installation and a user must accept them. These labels are checked at the application level when an app tries to use the security critical API. Developers can define new access permission in order to secure the interface of their apps in addition to the standard Android access permission.
3. In order to ensure the integrity and authenticity of apps, they are signed a X.509 certificate.
4. There are now enhanced web browsing security options available since the Android 5.0. It enables TLSv1.2 and TLSv1.1. Some enhancements are made in HTTPS and SSL protocols; Smart Lock has been introduced.
5. Android 5.0. also includes an enhanced FORTIFY_SOURCE feature that should provide security from buffer overflow attacks more efficiently, i.e. when an app tries to overflow the device memory in order to get sensitive data.

In the same way as for other operating systems, the latest stable version should be chosen in the Android case. At the moment of writing, the latest version is 5.0.2., which means that a device with a 5.0 version installed could be bought and updated.

D. SEAndroid

Android is a very rapidly evolving OS. The latest versions (including those, modified by the manufacturer) contain new promising solutions. Several Linux security enhancements were introduced starting from the Android 4.3. The Discretionary Access Control (DAC) was changed to Mandatory Access Control (MAC). MAC implements security control over all processes, objects, and operations. According to the developers, MAC can usually restrict erroneous and malicious access even for apps working with root privileges. This was not possible with DAC. The SE (Security Enhanced) Android was developed by US NSA [18]. SEAndroid should solve the following vulnerabilities: a) the malicious usage of administrative privileges, i.e. root exploits; b) the vulnerability of apps when they want to access or modify data without user authentication. It is worth mentioning that SEAndroid only began working in a permissive enforcing mode from version 4.4 and in full enforcing mode from version 5.0.

There also several threats that are not solved by SE Android:

a) It is not possible to forbid things which are allowed by the security policy. It means that the development of a good security policy is a critical task for SEAndroid to be efficient.
b) SEAndroid stops some core vulnerabilities from restricting the vulnerable code from untrusted apps, or making the impact of vulnerability negligible. But SEAndroid cannot stop all core vulnerabilities. That is why additional core security mechanisms should be used together with a SEAndroid solution.

SEAndroid cannot prevent threats that arise from other platform components. Particularly from components that have direct access to system resources, e.g. the RAM or network card.

E. File System

The Android file system is called the YAFFS (”Yet Another Flash File System”). It is built for Flash memory cards that are used as data storage for mobile devices. The classic limitation of the apps’ and users’ permission on folders and the file level is not the most

efficient in the Android system, because there is typically only one mobile device user [19]. The isolation of the apps, even when running them with different UDID, is a partial solution, because they are still executed by the same physical user and apps can ask for and get extensive permission during an install or update.

F. User Accounts

Starting from version 4.3 for tablets and from version 5.0 for smartphones, Android has some built-in user management capabilities. There are the following User Account types available:

The "Owner" user can add, remove, and configure user, guest, and profile accounts, i.e. can do almost everything (this account should be given to the company ICT administrator);

"User" accounts provide full access to apps and services on a device, while all changes made to system settings (like adding a Wi-Fi network) or updating the apps are applied to all user accounts on the device (!). An "Owner" user can restrict whether a "User" can use the phone for calls and SMS (this account should not be used for BYOD due to extensive permission availability).

A "Restricted profile" account (which is currently available only on tablets) can be restricted to run only allowed apps. Currently there is no built-in ability to fully restrict changes made to system settings. Only some additional restrictions, like disallowing location services when using the profile are possible (this account should be used for a user in a BYOD case).

A "Guest" account is a temporary "User" account. The system asks to reset the account or to continue the previous guest session each time the "Guest" account is used (could be useful when a device is given to another trusted user for some period of time).

G. The Fragmentation of Android

Android has the typical Linux issue: the fragmentation of software, dirty code, missing support for older versions (while almost half of all Android devices still use version 2.x). Furthermore, mobile device manufacturers often don't use the latest updates. This is one of the reasons why there are so much malware in Android.

According to F-Secure, mobile device manufacturers are guilty for Android security breaches in most cases, because devices cannot follow the development progress of apps. Mobile device security should be considered in general, because there are many threats that multiply with new apps, holes in internet browsers, messages (SMS), etc. [20], [21].

IV BYOD

A. What is BYOD?

There are different options available for allowing users to access business content and enterprise IT services. One of the options is when an organization distributes devices to employees with a strict security policy enabled. Another option is the so-called Bring Your Own Device (BYOD) when an employee brings their own private mobile device and it is up to the organization as to how to enforce security in such a case. A compromise between usability and security should be found.

Before selecting Android as an option for BYOD, the stakeholders should be aware that Android currently has an insufficient level of security. Currently, the situation with security and integrity is better on iOS and the Windows Mobile systems [22].

B. BYOD Threats with Android or How a Mobile Affects Enterprise Security

Quite a few guidelines and suggestions have already been prepared for somehow improving the situation in the field. Here is a list of the typical, but still insufficient suggestions identified for Android BYOD users [23]:

a) Set a device password (Settings / Location & Security / Set up screen lock),
b) Disable Unknown Source to install apps from (Settings / Applications / Unknown sources),
c) Install Anti-Virus protection,
d) Review application permissions,
e) Check for system updates,
f) Turn off wireless features (GPS, Bluetooth, Wi-Fi and Portable Hotspot) when not in use,
g) Do not Root the device,
h) Be aware of Web Security,
i) Back-up data on the device,
j) Turn off Google location.

Despite all these good suggestions, they are not enough to rescue a situation and serve as myths about adequate security. The problem lies within the inadequacies of Android architecture [18] and the low quality control in the Google Play store and the even lower quality in alternative Android stores.

In general, malware is distributed more on personal equipment, due to the lower security policy which is applied. That is why some enterprise network administrators do not allow work from home in order to protect documents from being infected with malware. Some organizations only allow connection to an enterprise IT system through VPN from specially configured workstations. All of this best practice collapses when the same mobile device is used for private and enterprise needs. Mobile phones and tablets are more vulnerable than enterprise workstations.

The PC may have old, outdated antivirus software installed which does not work or has been damaged by malware. That is why it should be checked occasionally using an antivirus CD. The equivalent check should also be done for mobile devices and memory cards [24]. Unfortunately, this is done quite rarely, but a mobile device or MicroSD card can

already contain viruses at the time of purchase. Not all antivirus software is effective enough. There are common enterprise security measures known to each corporate ICT specialist [25]:

1) using the enterprise cloud with integrated mobile secure synchronization capabilities, does not allow non-encrypted data to appear outside the secure enterprise virtual premises;
2) deploying an enterprise container with a special security level into a mobile device should decrease the risk of enterprise data leaks;
3) enabling mobile access to the enterprise document management system (e.g. SharePoint, Alfresco etc.), this decreases the desire for keeping data in some personal cloud;
4) blocking apps like Dropbox and Google Drive from accessing enterprise data;
5) the user's personal data can still be synchronized with Dropbox, Google Drive, etc. if the encrypted enterprise data containerization solutions are in place.

It is still important for an enterprise IT system to be accessed using a secure channel. That is why, when connecting to enterprise data through mobile internet or through an insecure wireless network, the creation of a VPN (Virtual Private Network) connection to one's own workplace or to the trusted internet provider service is recommended. The channel should also be secured using certificates or even more advanced solutions.

The introduction of a BYOD policy highlights new factors when assessing the security risks of an enterprise's IT infrastructure:

1) A user's full access to administer their personal mobile device is in conflict with the enterprise's general policy and their mobile device in the specific security policy. It increases the risk of data leaks and data vulnerability.
2) A user is free to choose any device model. But, there are many devices with a lower security level than desired (about 40 % of Android devices use old versions or OS versions which are not updated) [8]. This makes enterprise security management quite complex. It also becomes more difficult to track all the vulnerabilities and security updates available for the different device models.

The following is important when a mobile device is lost or stolen, especially for a BYOD case [5]:

1) the possibility of erasing the sensitive data that is kept on the device (user credentials, documents, GPS history, etc.) remotely;
2) the possibility of blocking device usage while pretending to be a device user, i.e. the messages receiving/ sending or accessing the network resources;
3) the sensitive information that is kept on the device must be encrypted;
4) the possibility of automatically blocking the device when it appears inside the untrusted Bluetooth or NFC zone.

It could be that the lost or stolen device cannot be accessed via the network and there is no possibility of initiating a remote wipe of the data. That is why encryption of the information on the device is a mission critical exercise.

A BYOD policy influences the costs of not only infrastructure and software, but also corporate risks and the level of client service. It makes the overall costs smaller, but less predictable. The following typical security mechanisms are needed when a BYOD strategy is in place [5]: 1) authentication and authorization; 2) network access control (NAC); 3) mobile device management (MDM); 4) mobile apps management; 5) encryption of both calls and internet data as well as data storage security.

Typical smartphones can ensure connection and data transfer using the following: mobile telecommunication networks (for calls, SMS, internet, and GPS support), WiFi, Bluetooth, USB cable, Micro SD card. All of them bring additional risks. Apparently, in a BYOD case, best practice is to disable all these options except for calls and SMS, and not allowing them to be enabled automatically. A user can enable them knowingly in a manual way, when needed, and disable them again when they are not being used anymore. But this solution does not cover everything, including a user's level of social responsibility.

C. Options to Restrict Android Users from Enhancing Security

There are different ways in which to forbid a user from installing or changing the configuration of an Android OS:

1) The easy one – install the AppLock free app or similar, set the password for settings, for Google Play, for apps installation and for running specific apps, etc. This option can be used by everyone.
2) There is a built-in multiple users feature in the latest Android versions (starting from 4.3 for tablets, and starting from 5.0 for smartphones). However, the feature should be extended with more configurable options per user account.
3) The complex option – to root the device and to create multiple user accounts there. Restricting the installation of new apps is possible, changing the settings, and forcing the usage of predefined networks.
4) A wholesome option would be to connect the device to the enterprise IT system, domain, or special server, e.g. Google for Work or Windows Intune with the Microsoft System Center Configuration Manager. This will be the place where user rights on the mobile device can be administered in a centralized way. It

will also ensure secure access to documents, and email.

5) Another option is to choose a device with already OEM enabled OS extra features. It is possible to limit such Android versions to only install the apps, for example, from the Nokia or Samsung stores. The manufacturers say that these stores contain apps which have been verified more thoroughly.

There are different options available for ensuring the security on an Android device depending on the selected administration method. These relations are shown in Table 1.

TABLE I

THE WAYS OF ADMINISTRATING AN ANDROID DEVICE FOR A BYOD CASE

Nr.	Administration Object	Original Android Settings	Rooted Android	Connected to the company mail server	Connected to the company workplace management server
1	Data storage encryption	+	+	+	+
2	Login password	+	+	+	+
3	Multiple user support	+ (from Android 4.3 tablets and 5.0 phones)	+	+	+
4	The ability to work from a restricted user account (forbidding the installations)	- (+ if AppLock)	+	+	+
5	Monitoring the apps' activity		+	+	+
6	Requirements for password strength and change frequency		+	+	+
7	Connect to email only through the secure channel			+	+
8	Connect to documents only through the secure channel			+/-	+
9	Centralized administration of user permission				+
10	Centralized administration of apps to run				+
11	Centralized remote apps installation				+
12	Centralized remote device update, antivirus check, backup				+

Apparently, company administrators have the broadest variety of administrative options when a mobile device is connected to a company workplace management server with mobile device management (MDM). For example, Google Android for Work allows for the administering and restricting of user account settings. Placing a restriction on a user to use and install only the allowed apps can be set within this solution as well. Windows Intune together with the Microsoft System Center Configuration Manager has similar capabilities.

The Samsung KNOX solution allows users to use a device for both personal and enterprise needs through separating these two environments. KNOX is a special Android version that is a part of "Samsung for Enterprise" (SAFE). An employee can only use the predefined and monitored apps in the enterprise environment of the device. This environment is administered by the enterprise IT department. An employee can also switch to the personal environment where he can access personal photos, the calendar, games, etc. This data is not available to the enterprise IT department. The IT department can wipe all the data from the enterprise environment when needed, while the personal environment is not affected. If the device is infected by malware, it cannot access the enterprise data and apps. [26], [27].

There are some similar services available from other vendors like Airwatch from Vmware, Blackberry Enterprise Server, Citrix, MaaS360 from IBM, MobileIron, SAP, SOTI, Motorola AME 2000, Huawei AnyOffice Mobile Security Solution, LG Electronics Enterprise Mobility Solution, etc.

The workstations within an organization are typically administrated in a centralized way by adding them to the organization domain. **The centralized management of mobile devices from the workplace management server (using MDM in particular) is potentially one of the most effective solutions for administering a mobile device in BYOD and other business cases too**. The further development and extension of such solutions and their alternatives is one of the tasks for IT in the near future.

V ARCHITECTURAL PROPOSALS FOR BETTER BYOD SUPPORT

Proposal 1 – Android as the Web Server concept

The authors propose viewing the apps on the Android platform as isolated websites on a web

server. There is much more experience accumulated on web server security than there is on Android's one. A single physical web server can host hundreds and thousands of websites from different authors. Each website can be isolated through, for example, Apache <virtualhosts> directive, allowing it (website) to exist only in the prescribed folder. It is bad practice to run a web server with one default user www-data afterwards. In such a case, the creator of any hosted website can still access other folders.

Good practice on the Linux Web server is to run the exclusive Apache instance per each website as a different guest user. Access to the website's folder should be given only to this guest user. There is no exceptional opportunity to get access to browse other folders when the system is configured in such a way. In the proposed solution, a small source of possible vulnerability could be the common RAM.

In Android, this approach has been only partially implemented – each app is executed with a different system user but apps can ask for and get extensive permission during the installation, and access other data that does not belong to them. A user is also often asked for additional permission from apps during the app update process and provides this without even reading and thinking about the content of the dialogue, which creates a messy situation.

This means that Android with default installation is not currently suitable for a BYOD, as apps isolation is insufficient. If it is not possible to isolate apps, then the only option is to forbid the user to install, update, and configure. Furthermore, the new Android feature "User Accounts" is not the final solution, because the app is updated for all user accounts at once which means that the app is the same for all users. The only difference is the profile data per each user inside the app. This means that installing or updating an infected app in the private account can also harm the data inside the user account that is meant for work.

The authors propose conducting a thorough analysis of all web server experience to look for solutions that could be carried over on Android for BYOD.

Proposal 2 - Absolute Virtual Machine Isolation per each User

Each Android app runs on its own virtual machine (VM) process, but apps isolation is not absolute, because Android uses the process VM, not the system VM. The last is more secure, so we are proposing some hybrid models. **Even though VM on Android has not been created as the security solution, it is possible to convert it to be suitable for this purpose.** The ideal conditions for a BYOD case would be if each VM used the isolated storage, isolated part of RAM, and used its own separated processor core. This could be useful for a BYOD case. The performance of the latest mobile devices with 4 and more cores and 1 GB or more RAM allows the authors to predict that it could be implemented.

There are two potential solutions available: A) a user will use the same account for private and work needs; B) a better option is when a user uses two accounts with two different profiles.

If the first option with one account is selected, then there should be a possibility for isolating potentially harmful apps from good ones and from the user data inside them. The current approach of running the VM instance per app is not suggested for the isolation described above. However, to build a totally separated VM (storage, RAM, data) for each app would take up too many system resources.

We propose running the Android VMs in two absolutely isolated processes with two different accounts where one VM is for private less trusted apps, but the second one – for work apps. The existing app separation mechanism should be kept inside these VMs. There should be no chance for private apps to access data outside their VM.

Private apps can also be divided into two more VMs, based on the trust level. Then there would be 3 VMs in total, and each could run on its own processor core. Android VMs were not originally planned as a security solution, but could become so, if apps permissions were managed not on the OS, but on a VM level. **The apps of a VM for work must be managed by the organization MDM server**, see Fig. 1.

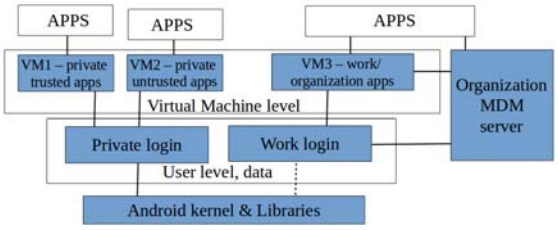

Fig. 1. The proposed *Android* architecture security model for a BYOD.

If there are two different accounts created for the BYOD case, then it is important to separate not only the user data by isolating the data folders, but also to really separate the apps themselves. It means that the same app for two different accounts should be installed twice.

Currently the security issues for enterprises allowing BYOD are very important especially considering the alarming facts published in Veracode analytics in March, 2015 – about 2,400 unsafe mobile apps are installed on employee devices in an average global enterprise [28].

VI CONCLUSIONS AND IMPLICATIONS FOR FURTHER WORK

The Android project has succeeded in some respects. But, today it is not yet ready for BYOD by

default – for secure work and private entertainment with the same device. It is not possible to force all people in an organization to follow the security rules they have signed without some technical restrictive administration tools.

Apps that are indented to reach the masses, like governmental ones, should be developed for at least three platforms: Android, iOS, and Windows Phone. On the other hand, it could be possible to agree on only one platform for a company's internal needs. But it is worth remembering that the Android operating system is not superior to its competitors with regards to security.

When buying a smartphone or tablet, it should be a recent model: its SIM should support 4G/ LTE voice encryption; there should be a possibility to limit voice calls to only use this network type (VoIP is not the preferred encryption solution); it should be updated to the latest OS version; it should have a at least 4 cores processor and at least 1 GB RAM; there should be a possibility of encrypting data storage and setting the device login password; there should be the possibility to administer the device in a centralized way (using the MDM server) in order to minimize the organization's security risks.

The same device should not be allowed for work use and private needs if the device is not specially prepared for this. Connecting a mobile device to the company email system is not enough, because it does not provide the management of all system settings on an Android device. We suggest that any organization should ensure the use of the centralized mobile device management (MDM) server both in a BYOD case or when the organization distributes the device to employees. The following are examples of MDM systems: Google Android for Work, Windows Intune with Microsoft System Center Configuration Manager, Samsung's KNOX within Samsung Approved For Enterprise (SAFE), etc.

Looking into the future, it is desirable to continue to improve the Android architecture, taking over the positive experience of Linux Apache web servers and the experience of virtual machines for cloud services systems.

If it is decided to use the same user account both for work and for private needs on a mobile device, then an organization needs to ensure that potentially harmful apps absolutely never face the good ones and their data. We propose the running of the Android virtual machines as at least two absolutely isolated processes where one is for good (work) apps, while the other is for less trusted (private) apps. Private apps could even be separated between two more VMs. It should not be possible for private apps to access the data outside their VM.

If the decision is to use two different user accounts (which is more preferable), then it is important to isolate not only the user data from each account, but also the apps themselves. In order to achieve the isolation, a VM with its own part of RAM and its own processor core per each user should be run.

The set of apps per each user should also never face the apps and data of another user. It means that the same app from two different accounts should be installed twice. The existing app separation mechanism should be kept inside these VMs.

The proposed solutions can be helpful both for a BYOD case and for the situation where an organization distributes the devices itself. The implementation of these suggestions could make the IT world a bit safer.

VII ACKNOWLEDGEMENTS

This research is part of a project „Competence Centre of Information and Communication Technologies" run by IT competence centre, contract Nr. L-KC-11-0003, activity Nr.1.22, co-financed by European Regional Development Fund.

VIII REFERENCES

[1] D. Kerr. Android dominates 81 percent of world smartphone market. [Online]. Available: http://news.cnet.com/8301-1035_3-57612057-94/android-dominates-81-percent-of-world-smartphone-market/ [Accessed: Dec. 11, 2013].

[2] P. Beuth, W. Merkels. Handy abgehört werden konnte. [Online]. Available: http://www.zeit.de/digital/datenschutz/2014-12/umts-verschluesselung-umgehen-angela-merkel-handy [Accessed: Dec. 25, 2014].

[3] K. Nohl. Attacking phone privacy. Berlin: Security Research Labs. 2010.

[4] OpenSignal. Android Fragmentation Visualized. [Online]. Available: http://opensignal.com/reports/2014/android-fragmentation/ [Accessed: Dec. 20, 2014].

[5] Samsung Electronics Co. Ltd., featuring Gartner. Strategies to Solve Challenges of BYOD in Enterprise. 2013.

[6] A. Goodloe, S. Person. NASA Formal Methods: *4th International Symposium, NFM 2012, Norfolk, VA, USA, April 3-5, 2012*, Proceedings, Springer, 465 lpp.

[7] C. Osborne. Android app malware rates jump 40 percent. [Online]. Available: http://www.zdnet.com/android-app-malware-rates-jump-40-percent-7000019093/ [Accessed: Aug. 7, 2013].

[8] Z. Whittaker. Millions of Android users vulnerable to security threats, say feds. [Online]. Available: http://www.zdnet.com/millions-of-android-users-vulnerable-to-security-threats-say-feds-7000019845/ [Accessed: Aug. 26, 2013].

[9] N. A. Staff. How to check the legitimacy of Android apps. [Online]. Available: http://networksasia.net/article/how-check-legitimacy-android-apps-1324343340, 11.10.2013.

[10] P. Ducklin. Naked security. [Online]. Available: http://nakedsecurity.sophos.com/2013/05/31/android-malware-in-pictures-a-blow-by-blow-account-of-mobile-scareware/ [Accessed: May 31, 2013].

[11] C. Castillo. McAfee Blog Central. [Online]. Available: - http://blogs.mcafee.com/mcafee-labs/phishing-attack-replaces-android-banking-apps-with-malware [Accessed: Jun. 3, 2013].

[12] Android Smartphone Security. [Online]. Available: http://latestandroids.wordpress.com/2013/06/10/android-smartphone-security/ [Accessed: Jun. 10, 2013].

[13] Technet. Security Options. [Online]. Available: http://technet.microsoft.com/en-us/library/jj852268.aspx [Accessed: Dec. 25, 2014].

[14] Technet. Windows Server Security. [Online]. Available: http://technet.microsoft.com/en-us/windowsserver/windows-server-security.aspx [Accessed: Jan. 12, 2015].

[15] Android Community. Android Security Overview. [Online]. Available: http://source.android.com/devices/tech/security/ [Accessed: Jan. 8, 2015].

[16] Android Community. Dashboards. [Online]. Available: http://developer.android.com/about/dashboards/index.html [Accessed: Nov. 12, 2014].

[17] Android Community. Session Initiation Protocol. [Online]. Available: http://developer.android.com/guide/topics/connectivity/sip.htm [Accessed: Nov. 12, 2014].

[18] L. Chanhee, K. Jonghwa, C. Seong-je, C. Jongmoo, P. Yeongung. Unified security enhancement framework for the Android operating system. *Supercomput 67:738–756, DOI 10.1007/s11227-013-0991-y, Springer Science+Business Media New York 2013*, Published online: 6 August 2013 [Accessed: Dec. 29, 2014].

[19] YAFFS. Overview. [Online]. Available: http://www.yaffs.net/yaffs-overview [Accessed: Dec. 19, 2014].

[20] F-Secure Labs. Whitepappers. Mobile Threat Report. [Online]. Available: http://www.f-secure.com/static/doc/labs_global/Research/Mobile_Threat_Report_Q3_2013.pdf. [Accessed: Nov. 12, 2014].

[21] F-Secure. (2014). Mobile Threat Report Q1 2014. [Online]. Available: https://www.f-secure.com/documents/996508/1030743/Mobile_Threat_Report_Q1_2014.pdf. [Accessed: Feb. 8, 2015].

[22] Microsoft Corporation. Low cost devices in government and education- Windows vs. Android. [Online]. Available: http://blogs.technet.com/cfs-file.ashx/__key/communityserver-components-postattachments/00-03-59-75-35/Low-Cost-Devices-in-Government-and-Education-_2D00_-Windows-vs-Android.pdf [Accessed: Jan. 11, 2015].

[23] T. Oh, B. Stackpole, E. Cummins, C. Gonzalez, R. Ramachandran. Best Security Practices for Android, BlackBerry, and iOS. *The First IEEE Workshop on Enabling Technologies for Smartphone and Internet of Things (ETSIoT)*, 2012.

[24] CERT.lv. Datorvīrusu ierobežošana. [Online]. Available: https://cert.lv/uploads/uploads/Seminari/Datorvirusu_ierobezosana.pdf [Accessed: Dec. 11, 2014].

[25] B. Rossi. Mobile content management and BYOD: the Dropbox catch-22. [Online]. Available: http://www.information-age.com/technology/mobile-and-networking/123457826/mobile-content-management-and-byod-dropbox-catch-22 [Accessed: Mar. 20, 2014].

[26] B. X. Chen, I. Austen. Samsung Armors Android to Take On BlackBerry. The New York Times. 2013.

[27] Samsung Electronics Co., Ltd., Enterprise Mobility Solutions. White Paper: An Overview of Samsung KNOX 2013.

[28] Average Large Enterprise Has More Than 2,000 Unsafe Mobile Apps Installed on Employee Devices. Veracode Press Release, March 15, 2015. [Online]. Available: http://www.veracode.com/average-large-enterprise-has-more-2000-unsafe-mobile-apps-installed-employee-devices [Accessed: Mar. 18, 2015].