Towards the Principal Applicability of a Zone Concept to Create a User-Friendly Secure Mobile Environment

Peter Riedl
UAS Upper Austria,
Campus Hagenberg
Softwarepark 11
Hagenberg, Austria
peter.riedl@usmile.at

UbiComp '13 Adjunct, Sept 8-12, 2013, Zurich, Switzerland.
ACM 978-1-4503-2139-6/13/09...$15.00.

Abstract
In the last decades mobile devices have evolved to hand-held computers. Their hardware- (processing power, communication interfaces, available memory, etc.) and software- (mobile operating system, third party applications, etc.) capabilities enable functionality far beyond simple phone calls and text messaging. The disadvantage of more complex software is that mobile devices become more vulnerable to attacks. The increasing number of "smart" devices on the market makes them an attractive target for attackers. We address this problem by proposing security zones on mobile devices separated on file, user or operating system level. These zones are used to provide the possibility to implement security critical applications with a strong focus on usability.

Author Keywords
Usability, security zones, virtualization, mobile devices

ACM Classification Keywords
H.5.2 [User Interfaces]: Graphical user interfaces, Interaction Styles.; D.4.6 [Security and Protection]: Access Controls, Informationflow controls.; D.4.8 [Performance]: Measurements.

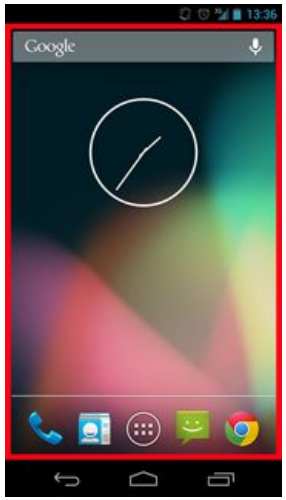

Figure 1: Colored border visualization.

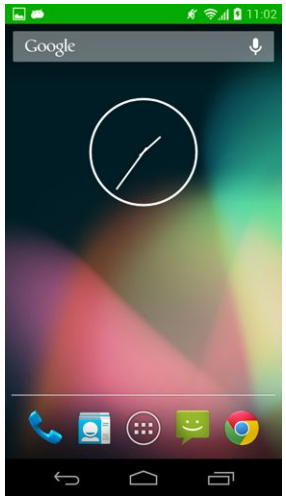

Figure 2: Colored notification bar visualization.

General Terms

Design, security

Introduction

Currently Android is among the most widely used operating systems on mobile devices[1] and it is open source. Therefore Android was chosen as the platform for all prototype implementations. The size of a software system has direct influence on the vulnerability to security attacks. One metric to describe the size of an operating system is the number of source lines of code (SLOC). The number of SLOC for Android is roughly 12000000[2]. This amount of code makes a full security assessment of the whole operating system prohibitively costly. As a consequence of this, many security exploits have been found for the Android operating systems [2, 4, 5] and therefore can not be considered fully secure.

Despite the lack of non-exploitable operating systems, the desire for applications with high security demands (e.g. mobile banking, mobile payment, access to company network, virtual identity documents, etc.) is growing steadily. Another emerging problem is employees' wish to use their own device privately and for business. This brings up security and maintenance problems for system administrators. The commonly used term for this problem is "Bring Your Own Device Problem" (BYOD problem). The approach we take to address these demands is the introduction of security zones as described in the next section.

[1] http://www.idc.com/getdoc.jsp?containerId=prUS23946013

[2] http://www.gubatron.com/blog/2010/05/23/how-many-lines-of-code-does-it-take-to-create-the-android-os/

Approach

Enabling security critical applications on mobile devices and solving the BYOD problem go hand in hand with restrictions. These restrictions could mean to prevent installing applications on the device, disabling parts of the hardware (e.g. bluetooth, camera), permission limitations for applications or only allowing certain network connections such as a virtual private network (VPN) to the company network. All of them would influence user experience negatively. Our approach to address the trade-off between security and usability is the introduction of security zones. The reasoning behind this is very simple: Because it seems very difficult to combine security and usability in one device, we divide one physical device into multiple logical devices for different purposes. Examples for different zones could be: A standard zone without any restrictions (current situation), a private zone for security critical operations in the private domain (e.g. mobile banking) and a business zone which is fully controlled by the company system administrators. This approach brings up the following research questions:

What are the necessary steps to ensure the integrity and authenticity of executed code?

- How to ensure code integrity during boot time (including boot loader, kernel, libraries, and system applications)?
- Which combination of hardware components within the mobile device fulfills the requirements and is realistic for mobile devices (considering size, power efficiency, performance, and cost)?
- How to ensure integrity and authenticity of underlying hardware?

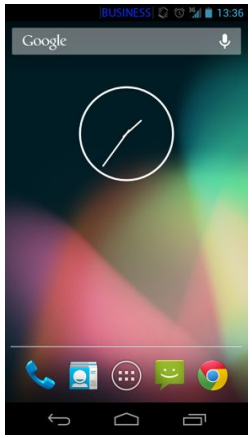

Figure 3: Colored text visualization.

Figure 4: Swipe switching mechanism.

What are the advantages and disadvantages of using a zone concept to create an environment to enable security critical applications on mobile devicess?

- Is it sufficient to separate zones on operating system level or is it necessary to also divide user- and application space into different zones?

Which combination of visualization and switching mechanism provides the best user experience in terms of zone distinguishability, low error rate, cognitive overhead and time spent? Which virtualization mechanism performs best regarding memory consumption, possibility of hardware access control, processing speed, security, HW independence?

- What are the SW/HW requirements to enable adequate access control?

Planned Implementation

The work described in this paper is carried out within the *u'smile* project[3]. The research vision for this project is to enable applications with very high security requirements on mobile devices. Examples for these would be to have your passport, your health insurance card or your driver's license on your mobile device. Despite the assumption that the Android operating system can never be considered fully secure, we have to create a channel of secure communication from the user to the actual hardware. The need for this additional communication channel arises because the user can install arbitrary (potentially malicious) applications on the device, which could be eavesdropping on input or masquerade as secure applications and fake output. In order to enable this, several adaptations have to be made to the Android operating system.

[3] http://www.usmile.at

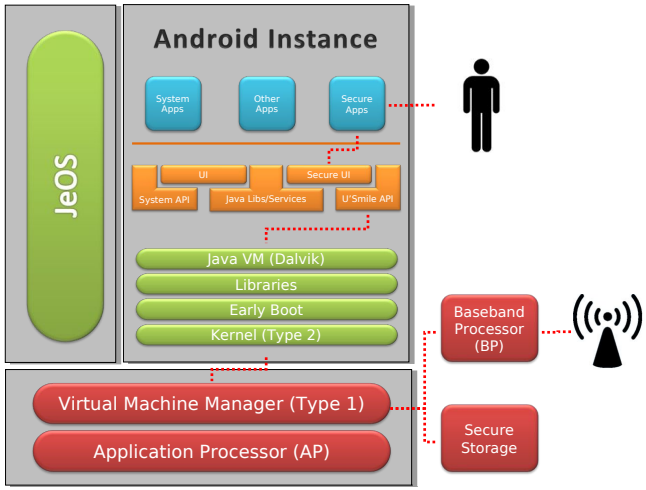

Figure 5: Ways to enable secure user-to-hardware communication.

We are researching different approaches to enable secure user-to-hardware communication as depicted in figure 5 and investigate different virtual machine managers (Type 1: XEN, OpenVirtualization; Type 2: KVM, SELinux) as the basis for our zone concept [1, 3]. Having the constraints of the respective technology in mind, we will create customized versions of Android to interact with the underlying virtual machine manager. One custom Android version will have the same functionality as the official one, and also add support for the zone system. In another version we will reduce the code to the bare minimum needed for operation, to create a "Just enough Operating System" (JeOS).

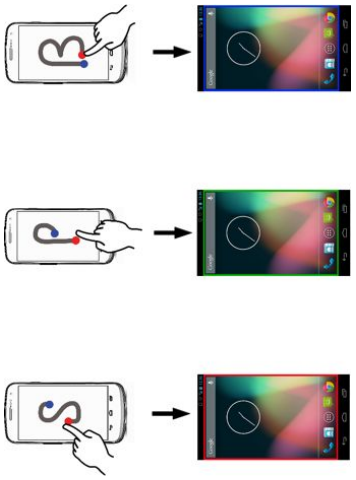

Figure 6: Gesture switching mechanism.

Figure 7: Lock screen switching mechanism.

User Interaction

One very important aspect besides the used technologies is user interaction. We are fully aware that the most solid secure communication is not worth much without proper user notification and a way to apply the proposed security zone system easily for the average user. In order to find out how to bridge the gap between security and simple usage, we are currently conducting a user study to compare different approaches for displaying zones and switching between them. In the user study we are comparing three different ways of visualizing the current zone.

- Colored border: A border in the color of the respective zone is drawn on the screen (see figure 1).
- Colored notification bar: The notification bar is colored according to the current zone (see figure 2).
- Colored text: The name of the current zone is displayed in the notification bar (see figure 3).

We also propose three different ways of switching between security zones.

- Swipe: A three finger swipe is used to cycle through zones (see figure 4).
- Gesture: The first letter of the desired zone is drawn on the screen to switch to the respective zone (see figure 6).
- Lock screen: The switch between zones is executed on the lock screen (see figure 7).

Objective and Biographical Sketch

In the current state of this work there are many open questions besides the research questions stated above. Input from other fellow researchers could potentially bring up even more ideas or help to find the "best" solution within the set of constraints we are currently aware of.

Peter Riedl received both his bachelor and master degree from the University of Applied Sciences Upper Austria, Campus Hagenberg (UAS). Since October 2012 he is enrolled in the PhD program of the Johannes Kepler University, Linz (JKU).

Acknowledgements

We gratefully acknowledge funding and support by the Christian Doppler Gesellschaft, A1 Telekom Austria AG, Drei-Banken-EDV GmbH, LG Nexera Business Solutions AG, and NXP Semiconductors Austria GmbH.

References

[1] Brakensiek, J., Dröge, A., Botteck, M., Härtig, H., and Lackorzynski, A. Virtualization as an enabler for security in mobile devices. IIES '08, ACM (2008).

[2] Davi, L., Dmitrienko, A., Sadeghi, A.-R., and Winandy, M. Privilege escalation attacks on android. In *Information Security*, vol. 6531. Springer, 2011.

[3] Hwang, J.-Y., bum Suh, S., Heo, S.-K., Park, C.-J., Ryu, J.-M., Park, S.-Y., and Kim, C.-R. Xen on arm: System virtualization using xen hypervisor for arm-based secure mobile phones (2008).

[4] Miller, C. Mobile attacks and defense. *Security Privacy, IEEE 9*, 4 (2011).

[5] You, D.-H., and Noh, B.-N. Android platform based linux kernel rootkit. In *Malicious and Unwanted Software (MALWARE), 2011 6th International Conference on* (2011).