Security of the Android operating system

Yury Zhauniarovich

University of Trento, Trento, 38100, Italy
yury.zhauniarovich@unitn.it

1 Summary

Modern smartphones become an everyday part of our life. Checking emails, browsing the Internet, photographing, navigation are successfully carried out with the help of smartphones. Obviously, this happens because mobile phones have been provided with the useful functions.

In the smartphone domain, the Android OS is by far the most popular platform being installed on about 79% of all new mobile devices [3]. Those figures clearly show the pervasiveness of Android, mostly justified by its openness. Being a part of the Open Handset Alliance initiative, Google released most of the Android code under open source licences. Thus, we have an ability to explore this operating system, change platform components and build customized images of the Android OS. Moreover, the third-party applications can be easily developed and tested on this platform without publishing them in an application market. Hence, it is not surprisingly why this platform is so popular nowadays.

Unfortunately, the information about the intrinsics of this operating system is sparse and scattered around different resources. This does not concern Android application programming during the last several years lots of books and web resources appeared describing the process and best practices how to develop Android apps. Moreover, the official documentation about app programming is quite complete and can be treated as a credible source of information on that topic. On the contrary, the official documentation about the system programming is poor and gives you good insights about how to download the Android sources and build them. Additional information only partially covers the topics and does not provide you with the whole picture. The situation in case of security is even more dismal.

In this tutorial we try to close the gap. We consider the layered structure of the operating system and examine the main security mechanisms implemented in Android. In particular, we inspect the sandboxing mechanism implemented at the Linux Kernel level. We will consider how the kernel enforces the isolation of applications and operating system components exploiting standard Linux facilities (process separation and Discretionary Access Control over network sockets and filesystem). Further, we plan to consider the security mechanisms implemented at other layers. In particular, we give an understanding how the security is designed at the Android Middleware level. On this level an IPC Reference Monitor mediates all the communications between processes and controls how applications access the components of the system and other apps. In Android,

IPC Reference Monitor follows Mandatory Access Control (MAC) access control type and is based on permission system. The implementation details of the permission system is also planned to be considered in the tutorial.

Other notable part of the tutorial is dedicated to the limitations in the Android operating system and the state-of-the-art research approaches that close this gap. In particular, we plan to consider such systems as CRePE [1], MOSES [6], FSquaDRA [5], etc.

2 Potential Audience

- Android security researchers
- Researchers working in the field of mobile and desktop operating systems
- Industry researchers and engineers interested in the deep understanding of the security procurement in the Android operating system
- Researchers, faculty and graduate students who explore the limitations of the Android operating system and its ecosystem
- Android application developers

3 Expected Prerequisite Knowledge

- Basic knowledge of the Linux operating system and its security mechanisms
- Basic knowledge of Java/C/C++
- Experience in Android app development is a plus

4 Outline

1. Introduction
2. Android Stack
3. Android Security
   - Linux Kernel Level
   - Native Userspace Level
   - Application Framework Level
   - Application Level
4. Android Security Extensions: open problems and solutions
5. Conclusions and Questions

References

1. Conti, M., Crispo, B., Fernandes, E., Zhauniarovich, Y.: CRêPE: A system for enforcing fine-grained context-related policies on Android. IEEE Transactions on Information Forensics and Security 7(5), 1426–1438 (2012)
2. Fernandes, E.: Instant Android Systems Development How-to. Packt Publishing Ltd (2013)

3. Gartner: Gartner Says Smartphone Sales Grew 46.5 Percent in Second Quarter of 2013 and Exceeded Feature Phone Sales for First Time. Online, http://www.gartner.com/newsroom/id/2573415
4. Zhauniarovich, Y.: Android Security (and Not) Internals. Web (2014), http://zhauniarovich.com/files/asani/asani.pdf
5. Zhauniarovich, Y., Gadyatskaya, O., Crispo, B., La Spina, F., Moser, E.: FSquaDRA: Fast Detection of Repackaged Applications. In: Proceedings of the 28th Annual IFIP WG 11.3 Working Conference on Data and Applications Security and Privacy. pp. 131–146 (2014)
6. Zhauniarovich, Y., Russello, G., Conti, M., Crispo, B., Fernandes, E.: MOSES: Supporting and Enforcing Security Profiles on Smartphones. IEEE Transactions on Dependable and Secure Computing 11(3), 211–223 (May 2014)

5 Instructor Biography

Yury Zhauniarovich is a postdoctoral researcher at the University of Trento (Italy) in Security Research Group. He earned his M.Sc. degree in Computer Science from the Belarusian State University in 2007. From 2007 till 2009, he worked as a SAP Consultant at Itransition. In April 2014, he received his Ph.D. degree in Information and Communication Technology from the University of Trento.

His research interests include design, implementation and evaluation of security enhancements of mobile operating systems, runtime security, smartphone applications security and mobile malware.

Yury Zhauniarovich is the author of the free book "Android Security (and Not) Internals" [4] and a reviewer of the book "Android Systems Development How-to" [2] written by Earlence Fernandes. He has actively participated in the development of several research security enhancements for the Android operating system including CRePE [1] and MOSES [6] systems. He is also an active member of the StackOverflow community.