Towards Self-aware Approach for Mobile Devices Security

Nanda Kumar Thanigaivelan(✉), Ethiopia Nigussie, Seppo Virtanen, and Jouni Isoaho

Department of Future Technologies, University of Turku, Turku, Finland
{nakuth,ethnig,seppo.virtanen}@utu.fi

Abstract. We present conceptual design of self-aware security for mobile devices. The design is envisioned to bring self-awareness into the mobile devices security for optimal protection by regulating application activities. The proposed design contains three subsystems: *meta-level* enables self-awareness, *extended meta-level* extends protections to the base-level components through security mechanisms and *base-level* comprises of resources that are essential for applications execution. The presented design enables cooperation among security mechanisms (such as access control and anti-virus) as well as with self-aware agent. The cooperation is intended for better understanding of application activities that leads to recognizing threat patterns in advance. When a threat is predicted/detected, the self-aware agent communicates with the security mechanisms so that they can take the necessary actions. The design of the security mechanisms are elaborated using access control system and anti-virus as example cases.

1 Introduction

Mobile devices are becoming a predominant tool for accessing online information and services. The number of users accessing internet through mobile devices is estimated to reach 6.1 billion by 2022 and it will become the primary interface to access internet for more than 50% of users by 2018 [1,2]. Though mobile market is occupied by numerous manufactures, mobile platform is majorly dominated by two operating systems: Android by Google and Apple's iOS [3]. Both platforms allow developers to build applications and publish in market place by providing APIs [4,5]. The fact that mobile devices handle sensitive information and its resources can be accessed easily makes ensuring security and privacy a priority. In future, mobile devices will play a crucial role in the Internet of Things (IoT) by performing operations such as sensing and actuation, and also acting as an intermediate gateway for IoT devices. These activities will enhance the roles and usage of mobile devices but also increases its vulnerability to threats due to its involvement in various activities.

The emergence of new threats along with the lack of security awareness among the users makes mobile devices a lucrative target for attacks. Existing security

J. Rak et al. (Eds.): MMM-ACNS 2017, LNCS 10446, pp. 171–182, 2017.
DOI: 10.1007/978-3-319-65127-9_14

features in iOS and Android provide privacy controls to protect sensitive information [6,7]. There are a number of ways to overcome the restrictions. For instance, in Android platform, permission verification APIs that are used to verify the state of the required permissions may be exploited by application developers to force users to grant all permissions in order to nullify the advantage of runtime permission revocation control, which is available in Android 6.0 and higher versions. Numerous anti-virus are available to protect mobile devices from exploitation by malicious applications [25–28]. Most anti-virus are signature based and proprietary, failing to protect from new malicious applications behavior and also limiting the users choice. The security mechanism in Android like permission model act as access control system in order to prevent unauthorized access to resources. Several researches have been carried out in improving access control [8–11]. Policies of most access controls are coarse-grained and failed to regulate device level activities. In addition, the use of self-adaptation based on learning neither available nor capitalized in existing access control mechanisms. Hence, there is a need for self-adaptive fine-grained context-based access control for better control over application and device activities. In addition, embedding logic to accomplish the tasks of anti-virus to ensure the permanent existence of anti-virus mechanism into the platform for reliable detection of malicious activities.

In this work, we present a conceptual design that enables building of security mechanisms that can be configured and optimized by a self-aware agent. The main working principle of the design is to configure, control and direct the activities of various security mechanisms that are used to safeguard the mobile devices from applications exhibiting malicious behaviors. Furthermore, security mechanisms help the self-aware agent in the learning process by gathering the required information and communicating with it. This approach improves security of the device by optimizing the controls and detection ability of the anti-virus at runtime with minimal human intervention.

The paper is structured as follows. In Sect. 2, motivation and objectives of the self-aware security are discussed. The details of designing self-awareness and security mechanisms are presented in Sect. 3. The related works on modeling the self-aware agent are discussed in Sect. 4. Finally, future direction for realizing self-aware security in mobile devices and summary are presented in Sects. 5 and 6, respectively.

2 Motivation and Objectives

Mobile platforms like Android, expose device activities and resources through APIs. Several researches have been carried out to address permission overclaiming, abusing granted permission and security issues [8–12]. The platform developers have also strengthened the operating system by introducing encryption, SE Linux and runtime permission revocation [6,7]. However, it is still possible for applications to gather information about the users and devices through various ways. For example, sensors and event broadcasts are not protected in Android, which can be exploited to get user activities and location information [22–24].

A number of solutions have been proposed to thwart the threats in mobile platforms, particularly in Android but most of them concentrated either on access control or prevention of identified threats [8–12]. These solutions failed to recognize the assimilation of mobile devices into internet-of-things (IoT) and the resulting change in the mobile usage landscape. The ongoing efforts in bridging IPv6 and BLE will facilitate easy integration of mobile devices in the IoT ecosystem [13–15]. One potential example of this integration is the capitalization of sensors available in mobile for information gathering through crowd sensing [16–19]. In addition, intercommunication between mobile devices is highly likely in future IoT applications. Once this opportunity is available, applications that try to harness the crowd sensing may exploit the mobile resources. In these scenarios, it is difficult to know the security or privacy threats in advance.

One of the limitations of the existing security measures is that the lack of cooperation between various security solutions though they are operate in the same device. For example, access control system performs its enforcement operations based on the given configuration without sharing or cooperating with the anti-virus applications. In access control, context are necessary to increase the policy granularity. However, the policy granularity and diversification are limited since the existing context based access control systems rely only on location, Wi-Fi, battery and time.

The objective of this work is to specify the concept of a self-aware security design that addresses the limitations of existing security measures and future challenges due to the change in mobile usage landscape. A self-aware system is capable of implementing appropriate measures at run-time in order to achieve and maintain the required performance [20, 21] by observing its environment and operations. The proposed security design is intended to achieve the following:

- To learn application behaviors through self-aware agent and communicate it with the security mechanisms which in turn take the necessary measures.
- To design security measures that are guided by the self-aware agent to achieve fine-grained control with minimal human involvement, leading to improved security.
- To devise cooperation strategy among security measures for improved understanding and detection of new threat types/malicious application activities.

3 Self-aware Security Design

Conventionally, security mechanisms are positioned to intercept the request calls from any applications (see Fig. 1). Based on the given configuration, it will decide further course of action. The configuration has to be performed manually by an administrator or a user depending on the complexity of the security mechanism and the interface provided for configuration. The security mechanisms do not possess any components or subsystems that can perform learning and optimization operations of other subsystems.

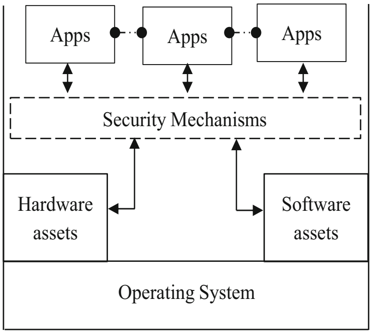


Fig. 1. Traditional mobile platform with security mechanisms

3.1 Self-awareness

Any system that has the capability to learn and adapt itself through continuous monitoring of its own operations and environment is termed as self-aware system. Design and implementation of a self-aware system is not an easy task but they offer numerous benefits upon proper employment. Some of the advantages are improved performance over time, efficient use of resources, reduced complexity, low required maintenance effort/time, detection/prevention of unexpected events, and better functioning of the entire system.

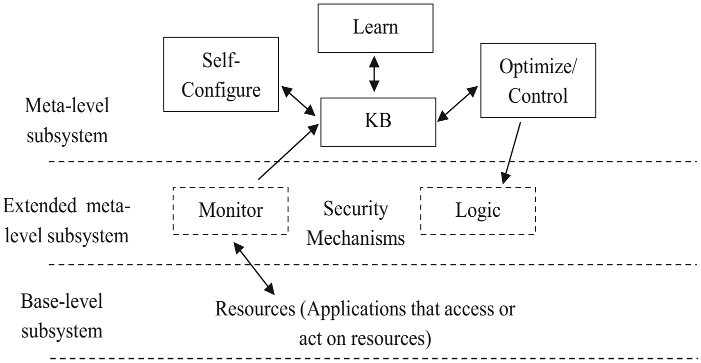


Fig. 2. Proposed self-aware reference architecture for mobile security

A system that shows any self-awareness property is comprised of two subsystems: meta-level that offers protection and base-level that requires protection [29]. A meta-level subsystem is responsible for the realization of self-aware property while base-level subsystem handles the domain functionality and it is protected/managed by the meta-level subsystem. The self-aware architecture of the proposed security design for mobile devices is given in Fig. 2. The main purpose of the given architecture is to secure the available resources in the mobile devices from being abused or exploited. The proposed design is slightly different from the architecture given in [29]. In our case, the monitor functions are categorized as an extended meta-level subsystem along with the security mechanisms since it aides the self-aware realization of meta-level subsystem by gathering information for learning and optimization process. In the proposed design, the security mechanisms are responsible for listening every resource request calls from applications and this is the main reason for the separation of monitor functionality from meta-level subsystem (self-aware agent). Furthermore, meta-level subsystem is not responsible for protecting the security mechanisms. Even the resources are not protected by the self-aware agent directly instead the security mechanisms are responsible for protection of resources under the guidance of the self-aware agent. This is the reason for the introduction of extended meta-level subsystem in the proposed architecture as can be seen in Fig. 2. Since resources do not have the ability to perform operations on their own, applications that use or act on the resources are included in base-level subsystem.

The modified mobile platform with the proposed self-awareness and controllable security mechanism is given in Fig. 3. The introduced components of the modified mobile platform are self-configure, learn and evaluate, controller, and knowledge base.

Self-configure: the purpose of self-configuration component is to update the settings of self-aware agent by considering the previous and current circumstances in order to fulfill the required objectives. It is also responsible for (re)defining the execution period for controller as well as learn and evaluate components.

Learn and evaluate: information gathered by the security mechanisms are analyzed to establish the common system behavior patterns. Upon detection of deviation in behavior pattern after successive execution, the appropriate changes will be deduced and submitted to knowledge base.

Controller: examines the provided recommendations and direct the operations of security mechanisms by changing their settings (for example, access control policies, encryption algorithms and enable/disable the entire security system). It is also responsible for restoring settings to previous states if the applied changes fails or causes abnormal behavior in the security mechanisms.

Knowledge base: act as a central repository for the storage of monitored information (in the form of logs), learning outcomes and execution settings of its components.

Security mechanisms, such as access control system and open anti-virus platform, acts as an extended components of the self-aware agent. The task of gathering information for learning will be handled by the security mechanisms since they have the ability to intercept every request. This will considerably reduce the workload of the self-aware agent and thus, allows it to invest more resources on learning tasks.

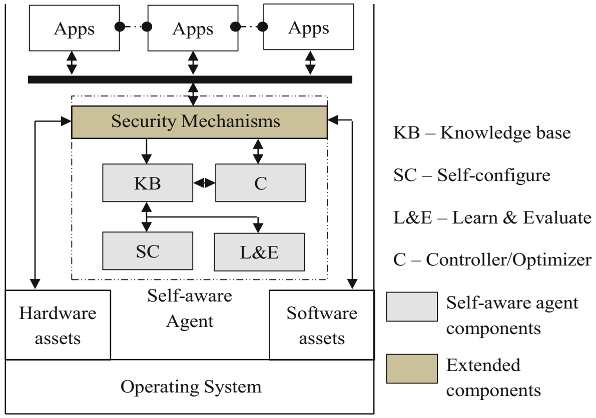


Fig. 3. Mobile platform with self-aware agent and security mechanisms

3.2 Security Mechanisms

In the current version of mobile platforms, each of the security mechanisms are designed as a self-contained system that operates independently and have complete control over its operations. For example, except policy configuration, access control system is completely self-contained. It does not have ability to cooperate with other security mechanisms. In the proposed self-aware security, cooperation among security mechanisms are introduced in order to achieve a holistic protection strategy that strengthen the overall security of the device. In addition, the security mechanisms cooperate with the self-aware agent as they are guided by the self-aware agent for enforcing the security measure. To have these cooperation, all critical operational requirements of the security mechanisms have to be separated from the core logic. In the proposed design, security mechanism contains only the core logic and its own execution parameters or requirements are controlled by the self-aware. The critical operations differ in different security mechanisms and hence, a special care has to be given when designing the security mechanisms and its interaction with the self-aware agent. The design of security mechanisms are elaborated using the following two example cases.

Access Control System. The access control system in self-aware security cooperates with the self-aware agent in order to perform dynamic policy administration by taking advantage of learning outcomes. Except the core logic, i.e., restriction enforcement, the rest of access control critical parameters has to be separated. The access control system must be classified into two subsystems so that it can handle device level and application level activities separately. The core logic must be extended to record all the activities intercepted during execution. For policy creation, both environment and feature based context have to be employed. In addition, new provisions to control the memory requirement

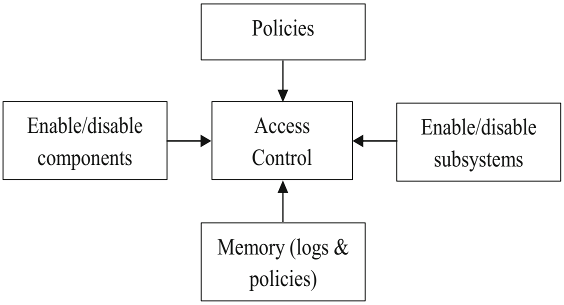


Fig. 4. Security mechanism: access control system security mechanism: access control system

(for logs and policy storage) and controlling of subsystems have to be established. The customized access control mechanism is shown in Fig. 4.

Open Anti-virus/Malware. Several vendors are providing proprietary anti-virus application to mobile devices [25–28]. The signatures will be updated periodically to detect malicious applications. Since they operate as a stand alone system application, it is not possible for any system to oversee the activities and/or cooperate with anti-virus applications. This limits their capacity of identifying new malicious activities. By cooperating with other security mechanisms and the self-aware agent, their detection capacity can be enhanced significantly. Thus, we are proposing a new open anti-virus security mechanism that will be managed by vendors as well as the self-aware agent collectively. The block diagram of the proposed open anti-virus is shown in Fig. 5. It will be designed as a part of the operating system and open to any vendors. The users can choose any vendor according to their preference and change the vendors at later stages. Compared to proprietary anti-virus application, the open anti-virus has to operate with two signature repositories. First repository will be used by vendors to

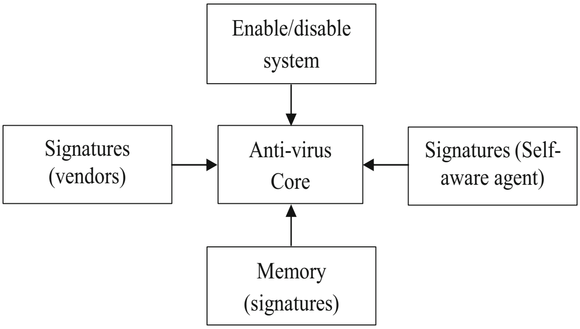
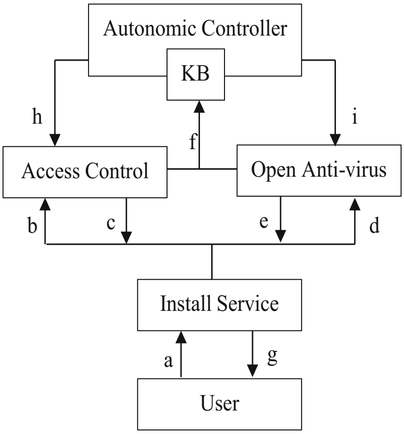


Fig. 5. Security mechanism: open antivirus

update the malicious applications' signatures and the other one by the self-aware agent to maintain its own learned malicious patterns. It will also have a provision to turn on/off the execution of core logic when it is ordered by the self-aware agent.

Overall Conceptual Design. The cooperation between the self-aware agent and security mechanisms is necessary to improve the security of mobile devices. In Fig. 6, the execution flow of application installation event between self-aware agent, access control and open anti-virus is illustrated. It shows the sequence of actions in the mobile platform during application installation event and gathering of information through observation of activities by security mechanisms. In the diagram, malicious verification of the application is initiated by install service but it can also be initiated by the access control system.



a. Install action by user
b. Policy verification
   i. Is install allowed?
   ii. Is application origin allowed?
c. Verification status
d. Malicious application verification
   i. Is malicious by vendor?
   ii. Is malicious by autonomic controller?
e. Verification status
f. Monitored information storage
g. User notification (success or fail with reason for denial)
h. Update policy repository and adapt execution settings
i. Update malicious repository and adapt execution settings

Fig. 6. Conceptual design of self-adaptive security

4 Related Work

A number of researches have been carried out on self-aware computing for realization of autonomous systems. There are also models provided to design self-aware property based systems [30,31]. MAPE-K [30] is one of the earliest and most adopted architecture for modeling autonomic systems. It has five components: Monitor (M), Analyser (A), Planner (P), Executor (E) and Knowledge (K). The knowledge component is shared by MAPE components. SEEC is another framework for self-aware computing that employs Observes (O), Decide (D) and Act (A) components for modeling self-awareness [31]. The O and A components are similar to M and E components of MAPE-K and the tasks of A and P components are collectively carried out by D component in SEEC framework. In [32], a Self-Adaptive Authorization Framework (SAAF) is used to manage access control automatically for computers. It is specifically meant for systems having multi-user and access restrictions based on authority (e.g., administrator, researcher and supervisor). The same framework can be usable in multi-user mobile platforms like Android but it will apply restrictions at user level on applications accessibility. It may not work at resource feature level (e.g., deny network association if security is WEP) in mobile environment and also lacks context based restrictions. In case of mobile security, the self-awareness is primarily used in malware detection techniques [33–36].

The proposed security design of this work has components similar to MAPE-K and SEEC frameworks but it differs in operations. In our work, the tasks are shared between the self-aware agent and security mechanisms. The task of monitoring and enforcing security measures are performed by the security mechanisms while the self-aware agent handles learning and optimization tasks. Since security mechanisms possess greater ability to regulate the mobile device activities, they can perform monitor functions better than any other components. The security measure enforcement is similar to Executor (E) in MAPE-K. The separation of tasks allows to distribute the workload among the components of the security design and enables the cooperation of security mechanisms that results in achieving better security. The interaction between the single self-aware agent and multiple security mechanisms helps to construct holistic view of application behavior patterns distinctively. The security mechanisms are uniquely designed as such that their operations can be controlled easily through the deployment of self-aware agent.

5 Future Directions

Currently, we envisioned self-configuration as part of the self-aware agent. It is responsible for configuration of the learning and optimization/control processes. These processes cannot run continuously due to: (1) sufficient number of logs are required for optimal learning; (2) initiating learning process will result in resource wastage if there is no considerable number of logs recorded between successive learning sessions; and (3) the configuration process should not use resources when the device is busy. Determining sufficient number of logs that are required for optimal learning is necessary for achieving appropriate level of accuracy in pattern detection. The availability of logs depends on the successive time period for learning process execution. Care should be taken in determining successive time period for learning process execution and the period should be adaptable at runtime depending on several factors including environment, available resources and device usage history. If the interval is long, optimal learning can be achieved due to huge number of logs but device may fail to detect/prevent unexpected new events that are occurred during the waiting period. If it is short, it leads to unnecessary resource consumption and inefficient learning. Hence, an algorithm that is able to dynamically adapt the learning interval and the required number of logs will be developed.

Learning algorithm plays a decisive role in the detection of malicious activities and threats. The algorithm has to have high detection accuracy with very low false positives. Given that the operating environment is mobile platform, the algorithm should be light-weight that requires acceptable level of resources with low energy consumption. Thus, accuracy and resource requirement have to be taken into account when selecting and customizing the learning algorithm.

To create a secure ecosystem and to harness the self-aware agent benefits running in multiple devices, establishing communication among agents is a necessity. The self-aware agent has to be designed to operate as distributed intelligent system in order to recognize, cooperate, assist and share knowledge among its peers. In addition, they have to possess competitive traits to handle malicious agents. Therefore, self-aware agent needs to be designed to operate in hybrid mode by combining cooperative and competitive traits.

Finally, the security of knowledge base is critical as it always resides within the mobile devices. There are possibilities that the knowledge can be targeted by the adversaries with an intention of obtaining/spoofing learned behavior patterns or for preventing the execution of tasks by the self-aware agent. It is compulsory to design a dedicated subsystem to handle the any request targeted to knowledge base for authenticity verification before allowing to perform any manipulation.

6 Summary

In this paper, we presented the design of self-aware security for mobile devices at concept level. The presented design has three interactive subsystems that are tasked to provide maximal protection by taking advantage of self-awareness. For seamless cooperation among security mechanisms as well as with self-aware agent, new design approach for the security mechanism has been proposed, as existing designs results in non-cooperative self-contained security mechanisms. The cooperation enables the self-aware agent to have a holistic view of mobile device activities, leading to better prediction of threats and enforcement of appropriate measures by security mechanisms. The design of access control and anti-virus security mechanisms are presented as case studies.

Acknowledgement. The authors gratefully acknowledge Tekes (the Finnish Funding Agency for Innovation), DIMECC Oy, and the Cyber Trust research program for their support.

References

1. Ericsson. Ericsson Mobility Report, November 2016. https://www.ericsson.com/assets/local/mobility-report/documents/2016/ericsson-mobility-report-november-2016.pdf. Accessed 22 Feb 2017
2. Gartner. Gartner says by 2018, more than 50 percent of users will use a tablet or smartphone first for all online activities (2014). http://www.gartner.com/newsroom/id/2939217. Accessed 22 Feb 2017

3. International Data Corporation. IDC: Smartphone OS Market Share, Q3 (2016). http://www.idc.com/promo/smartphone-market-share/os. Accessed 22 Feb 2017
4. Apple Developer. API Reference Apple Developer Documentation. https://developer.apple.com/reference. Accessed 22 Feb 2017
5. Android Developer. Package Index Android Developers. https://developer.android.com/reference/packages.html. Accessed 22 Feb 2017
6. Apple. iOS6 Software Update (2012). https://support.apple.com/kb/DL1578?locale=en_US. Accessed 22 Feb 2017
7. Android. Security Enhancements in Android 6.0. (2016). http://source.android.com/security/enhancements/enhancements60.html. Accessed 22 Feb 2017
8. Bugiel, S., Davi, L., Dmitrienko, A., Fischer, T., Sadeghi, A.-R.: Xmandroid: a new android evolution to mitigate privilege escalation attacks. Technische Universität Darmstadt, Technical report TR-2011-04 (2011)
9. Enck, W., Gilbert, P., Han, S., Tendulkar, V., Chun, B.-G., Cox, L.P., Jung, J., McDaniel, P., Sheth, A.N.: Taintdroid: an information-flow tracking system for realtime privacy monitoring on smartphones. ACM Trans. Comput. Syst. (TOCS) **32**(2), 5 (2014)
10. Heuser, S., Nadkarni, A., Enck, W., Sadeghi, A.-R.: ASM: a programmable interface for extending android security. In: Proceedings of 23rd USENIX Security Symposium (2014)
11. Wang, X., Sun, K., Wang, Y., Jing, J.: Deepdroid: dynamically enforcing enterprise policy on android devices. In: Proceedings of 22nd Annual Network and Distributed System Security Symposium (NDSS 2015). The Internet Society (2015)
12. Conti, M., Crispo, B., Fernandes, E., Zhauniarovich, Y.: Crêpe: a system for enforcing fine-grained context-related policies on android. IEEE Trans. Inf. Forensics Secur. **7**(5), 1426–1438 (2012)
13. Nieminen, J., Savolainen, T., Isomaki, M., Patil, B., Shelby, Z., Gomez, C.: RFC 7668 - IPv6 over BLUETOOTH® low energy. https://tools.ietf.org/html/rfc7668. Accessed 23 Feb 2017
14. Wang, H., Xi, M., Liu, J., Chen, C.: Transmitting IPv6 packets over Bluetooth low energy based on BlueZ. In: 2013 15th International Conference on Advanced Communications Technology (ICACT), PyeongChang, pp. 72–77 (2013)
15. Andersen, M.P., Fierro, G., Culler, D.E.: System design for a synergistic, low power Mote/BLE embedded platform. In: 2016 15th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN), Vienna, pp. 1–12 (2016)
16. Skorin-Kapov, L., Pripužić, K., Marjanović, M., Antonić, A., Žarko, I.P.: Energy efficient and quality-driven continuous sensor management for mobile IoT applications. In: 10th IEEE International Conference on Collaborative Computing: Networking, Applications and Worksharing, Miami, FL, pp. 397–406 (2014)
17. Angelopoulos, C.M., Evangelatos, O., Nikoletseas, S., Raptis, T.P., Rolim, J.D.P., Veroutis, K.: A user-enabled testbed architecture with mobile crowdsensing support for smart, green buildings. In: 2015 IEEE International Conference on Communications (ICC), London, pp. 573–578 (2015)
18. Zhao, D., Ma, H., Liu, L.: Frugal online incentive mechanisms for mobile crowd sensing. IEEE Trans. Veh. Technol. **PP**(99), 1 (2016)
19. Shu, L., Chen, Y., Huo, Z., Bergmann, N., Wang, L.: When mobile crowd sensing meets traditional industry. IEEE Access **PP**(99), 1 (2017)
20. Guang, L., Nigussie, E., Rantala, P., Isoaho, J., Tenhunen, H.: Hierarchical agent monitoring design approach towards self-aware parallel systems-on-chip. ACM Trans. Embedded Comput. Syst. (TECS) **9**(3), 1–26 (2010)

21. Isoaho, J., Virtanen, S., Tenhunen, H.: Current challenges in embedded communication systems. In: Innovations in Embedded and Real-Time Systems Engineering for Communication. IGI Global (2012)
22. Zhou, X., Demetriou, S., He, D., Naveed, M., Pan, X., Wang, X., Gunter, C.A., Nahrstedt, K.: Identity, location, disease and more: inferring your secrets from android public resources. In: 2013 ACM SIGSAC Conference on Computer & Communications Security, pp. 1017–1028. ACM (2013)
23. Narain, S., Vo-Huu, T.D., Block, K., Noubir, G.: Inferring user routes and locations using zero-permission mobile sensors. In: 2016 IEEE Symposium on Security and Privacy (SP), pp. 397–413. IEEE (2016)
24. Zhou, Y., Jiang, X.: Dissecting android malware: characterization and evolution. In: 2012 IEEE Symposium on Security and Privacy (SP), pp. 95–109. IEEE (2012)
25. Avira Holding GmbH. Avira Mobile Security (version 2.1). https://itunes.apple.com/us/app/avira-mobile-security/id692893556. Accessed 28 Feb 2017
26. NortonMobile. Norton Security & Antivirus (version 3.17.0.3205). https://play.google.com/store/apps/details?id=com.symantec.mobilesecurity. Accessed 28 Feb 2017
27. Trend Micro Incorporated. Trend Micro Mobile Security (version 5.2.1089). https://itunes.apple.com/us/app/trend-micro-mobile-security/id630442428. Accessed 28 Feb 2017
28. Avast Software. Mobile Security & Antivirus. https://play.google.com/store/apps/details?id=com.avast.android.mobilesecurity. Accessed 28 Feb 2017
29. Weyns, D., Malek, S., Andersson, J.: FORMS: unifying reference model for formal specification of distributed self-adaptive systems. ACM Trans. Auton. Adaptive Syst. **7**(1), 61 (2012)
30. Kephart, J.O., Chess, D.M.: The vision of autonomic computing. Computer **36**(1), 41–50 (2003)
31. Hoffmann, H., Maggio, M., Santambrogio, M.D., Leva, A., Agarwal, A.: SEEC: a framework for self-aware computing (2010)
32. Bailey, C., Montrieux, L., de Lemos, R., Yu, Y., Wermelinger, M.: Run-time generation, transformation, and verification of access control models for self-protection. In: Proceedings of the 9th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS 2014), pp. 135–144. ACM (2014)
33. Guo, D.F., Sui, A.F., Shi, Y.J., Hu, J.J., Lin, G.Z., Guo, T.: Behavior classification based self-learning mobile malware detection. J. Comput. **9**(4), 851–858 (2014)
34. Shabtai, A., Tenenboim-Chekina, L., Mimran, D., Rokach, L., Shapira, B., Elovici, Y.: Mobile malware detection through analysis of deviations in application network behaviour. Comput. Secur. **43**, 1–18 (2014)
35. Li, F., Clarke, N., Papadaki, M., Dowland, P.: Behaviour profiling on mobile devices. In: International Conference on Emerging Security Technologies, Canterbury, pp. 77–82 (2010)
36. Tong, F., Yan, Z.: A hybrid approach of mobile malware detection in Android. J. Parallel Distrib. Comput. **103**, 220–31 (2016)