Enhancing Mobile Device Security by Security Level Integration in a Cloud Proxy

Thomas Ruebsamen, Christoph Reich
Hochschule Furtwangen University
Faculty of Computer Science
Furtwangen, Germany
{*Thomas.Ruebsamen, Christoph.Reich*}@*hs-furtwangen.de*

***Abstract*—Smartphones, tablets, laptops and other mobile devices dominate our every day life and became indispensable for many businessmen. But at the same time the number of security vulnerabilities have been increasing. To increase the security of such devices the paper proposes a proxy running in a cloud environment that controls the access for mobile device to applications, enterprise services or Internet services. The developed access management system based on the Role Based Access Control (RBAC) model has been extended by 5 security levels. These security levels are determined by a classification of the user, the communication channel, and the device itself.**

***Keywords*-security; cloud computing; mobile device; mobile security**

I. Introduction

Since 2009 a drastic increase of reported security vulnerabilities and exploits in operating systems for mobile devices (e.g., Android, iOS, Windows Mobile, Symbian) can be observed. Attacks, especially those using viruses, worms and similar malware, have been relatively confined to desktop PCs, laptops and servers but are now more and more spreading into mobile platforms [1]. The main reason for this trend is their widely adopted usage and the fact that mobile devices are starting to become more and more similar to classic PC-like computers in terms of performance as well as field of application. A couple of years ago, mobile devices had a limited range of applications. Nowadays, expanding application stores and apps available for download, drastically have changed this. Mobile devices can easily be expanded in their functionality simply by installing new apps. A side-effect is the increased probability of being exposed by malware. With every new generation of mobile devices, especially in the smartphone and tablet sector, the performance regarding CPU, memory and network bandwidth is increased. This makes mobile devices an attractive target for attackers.

Another problem is the lack of security fixes for mobile system software. Manufacturers of mobile devices often fail to provide decent software-related support for their products. This is for example shown by the apparent version fragmentation which can be observed in the Android environment [2]. If the manufacturer does not provide its customers with software patches in time, devices become more vulnerable to exploits. Keeping the operating system and crucial software packages up to date is a well known best practice for securing PCs, yet regarding mobile devices this is often not possible due to lack of support. Using firewalls, anti virus scanners, spyware scanners, rootkit detectors and intrusion detection systems (IDS) on non mobile devices is not a common practice. Adopting such tools to mobile devices proves to be difficult, mainly because of lack of resources like battery longevity, computing power and storage.

Securing mobile devices has become one of the main concerns for companies, because they are adopting mobile devices for improving productivity of their employees. Their major concern is how to prevent attacks originating from compromised devices targeted on their corporate networks and their sensitive data.

To solve the problem of lacking resources on mobile devices, offloading resource intensive tasks to the cloud is one solution [3], [4]. Cloud Computing describes a technique where resources like computation power and storage are provided transparently over a network (usually the Internet). One major advantage of cloud computing is the relatively easy scaling of services. The results presented in this paper rely heavily on leveraging cloud computing especially for enabling scalability and providing sufficient resources to effectively enhance security of mobile devices. Such security mechanisms include but are not limited to anti virus, intrusion detection and application analysis in the cloud [5], [6].

In this paper a proxy, that controls the access of mobile devices to applications and services is proposed. The proxy is operated in the cloud which enables it to perform resource intensive analysis tasks. Also, the proxy is the central control component for evaluating the security as wells as the trustability of users, devices and communication channels. This results in the assignment of security levels which themselves are used to enable a more fine grained access control.

This paper is structured as follows: In this section, we gave an introduction to the security problems which occur with current mobile devices, such as smartphones and tablets in today's enterprise environments. In the next section, a security classification framework for mobile devices will be described. Based on this classification, we propose a security level model. In section IV, we will propose two different approaches for security level integration into the

RBAC model. Section V will highlight evaluation results of the proposed security level model and the classification framework using use cases. The last section includes the conclusion of this paper as well as future work.

II. Related Work

Portokalidis et al. [6] describe a system that implements an intrusion detection for Android based systems called *Paranoid Android*. Paranoid Android is based on a cloud deployment model where intrusion detection is offered as a service. By emulating whole devices in virtual machines in the cloud, it is possible to apply resource intensive anomaly detection mechanisms. This would not be possible to do on mobile devices because of the very limited available resources. The clone is kept in sync with the mobile device. Actions performed on the device are replayed by the emulator. They also show that it is imperative to optimize the synchronization and tracing processes because having to collect and transmit data can very easily lead to disproportionate exhaustion of the battery.

A very similar approach is taken by Zonouz et al. In their framework [7], [8] a lightweight agent is deployed on the mobile device, which collects user and sensor information. Additionally a proxy server is used to duplicate all traffic flowing between device and the internet. The collected traffic gets sent to an emulator in the cloud. Using the collected data from the agent and the proxy an analysis component scans for anomalies. In case of an ongoing attack the system informs the agent about countermeasures which need to be taken.

Andromaly [9] is a framework for detecting malware on mobile devices. Their approach is similar to those of classic host based intrusion detection systems. Android based devices are continuously monitored and attacks are detected using machine learning anomaly detectors. One of the main problems this approach are the limited resources on mobile devices which prevents the use of more sophisticated algorithms.

Schmidt et al. [10], [11] suggest using static analysis of executables as well as the integration of a collaborative system for detecting malware on Android based systems. By inspecting files on the function call level and comparing this data to already known malware files can be classified as harmful or harmless. The analysis can either be performed locally on the device or offloaded to a remote detection server. Additionally, devices can exchange analysis results with each other using the server. This leads to an improved detection rate.

Another approach, specifically targeted on the Symbian platform, is described by Bose et al. [12]. They are relying on behavioral analysis for detecting malware on mobile devices. Their idea is based on the assumption that a single action performed by an application can be classified as harmless, but in relation to other actions, which are performed in the same context, malware behavior can be exposed. Based on this assumption Bose et al. developed a database of behavioral signatures for malware. By training a support vector machine with normal behavior of applications, anomalies such as malware can be detected.

Kim et al. [13] analyze a very specific kind of malware causing battery exhaustion. These kind of attacks have already been described generally by Martin et al. in [14] and more specific by analyzing a security vulnerability in the MMS service by Radic et al. [15]. The core component of Kim's framework is a power monitor which monitors energy consumption and generates a power consumption profile. Using this profile it is possible to extract, analyze and detect attacks.

A very similar framework has been developed by Nash et al. [16]. Their system monitors mobile device parameters like CPU utilization and accesses to local storage to measure the used energy on a per process basis. Using this information they try to detect malware which tries to perform battery exhaustion attacks. This monitoring system is designed to be very lightweight. As an extension they suggest to start a fully-fledged intrusion detection system once an energy depletion attack has been detected.

Another kind of intrusion, especially theft, is the core concern of the work of Gupta et al. [17]. Basis for theft detection are profiles consisting of typing patterns and historic information like the history of made and received calls. Using this information the probability of the device being stolen or accessed by an unauthorized person is calculated. Unless there is no sufficient authentication of the user, data stored on the device remains encrypted. Additionally, if a theft has been detected a central management instance is notified.

The main differences of these approaches are whether the system is deployed on device or offloaded on a separate, dedicated system for analysis and detection. The system described in this paper uses a proxy server in the cloud for offloading most of the security related tasks. Also, the security level concept shares the same goals with the aforementioned projects, to enhance security and data protection in mobile enterprise environments.

III. Enhancing Mobile Security

To increase the security of mobile devices the access to services and data is controlled by a proxy running in a cloud (see Figure 1).

This architecture allows leveraging the advantages of cloud computing having almost unlimited computing power for analyzing the security status of the mobile system. The mobile security of the entire system depends basically on the security and trust level of a) the user, b) the mobile device, c) the communication channel, and d) the backend, the cloud. The proxy, which is under company control, is used to collect as much security related information

ISBN: 978-1-61208-216-5

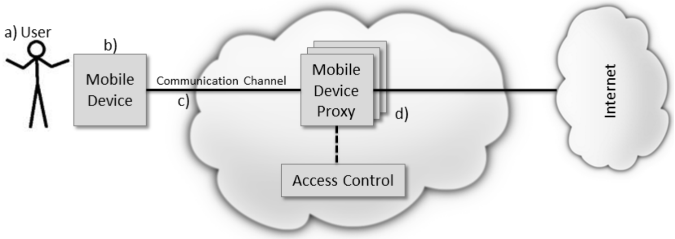

Figure 1. Mobile System Architecture with Cloud Proxy

about the aforementioned components as possible (e.g., by analyzing network traffic, querying company databases for organizational information et cetera). Additionally, the proxy requests information directly from the mobile device, which also monitors the aspects described in the following taxonomy. How this information is to be trusted (e.g., the device may send compromised data) is not in the scope of this paper, but will be part of our future work.

Before a detailed description on security levels will be made (see Section III-B), a mobile security taxonomy classifies the security domain.

A. Mobile Security Taxonomy

Figure 2 illustrates which properties have to be considered for mobile security devices into the categories: user, mobile device, communication channel, and backend.

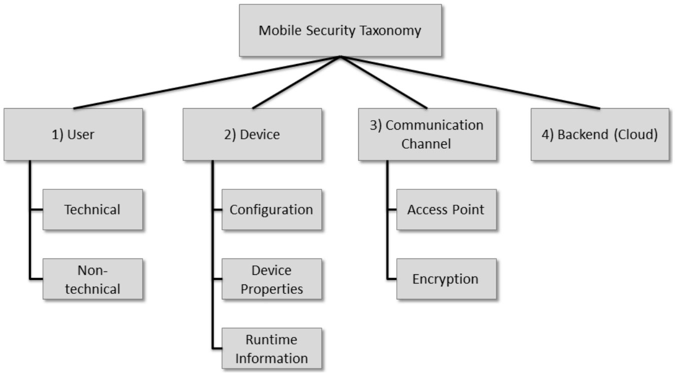

Figure 2. Mobile Security Taxonomy Overview

1) User: The user classification is targeted at the user of a mobile device. The primary function is to determine whether or not an authorized user is using the device. Mobile devices are inherently more prone to theft or unauthorized access because of their portability. Knowing that the owner or at least an authorized user is using the device is therefore crucial, when allowing access to sensitive data.

The second function is to evaluate used authentication mechanisms. For example, having no other authentication mechanisms in place apart from entering a PIN at device startup is very bad. There is no way to distinguish between an authorized user and for instance a thief. If there are other supported mechanisms like biometric identification in use, the user is more trustable, because of the stronger authentication.

Information which is used to evaluate the trustworthiness of a user and therefore assign a security level can be technical and non-technical:

Technical Information: These kinds of classification characteristics are strongly related to device and mobile operating software properties, especially supported authentication mechanisms.

The most simple is the support and usage of username/password combinations for additional user authentication. Most current mobile operating systems support at least authentication via an user-defined password.

Another way to enhance authentication is to use one-time passwords, which are generated on demand. These passwords are usually generated using special devices which are synchronized (based on the current time) with an authentication service. Requiring the user to be in possession of such a device reduces the risk the mobile device being used unauthorized. Of course, it can happen that both devices get stolen.

Similar to one-time password generators are dongles. Dongles are special devices linked to the mobile phone. By monitoring the proximity of the dongle, a mobile device can be locked and access denied until proximity is re-established.

A more sophisticated technical information is the support of biometric authentication by the mobile device. Devices which posses biometric scanners can achieve a better rating in user classification, assuming the biometric scanner and related software is tamper-proof.

Another way to gather information for user classification characteristics is to monitor location-related information. Many mobile devices have integrated GPS sensors. By tracking the location of a device and comparing it to a database such as an employee's schedule it could be detected whether it got stolen or not. Of course, GPS location and SSID are not 100% accurate, and further information is required.

Another way to identify a user is to make use of implicit authentication. Implicit authentication uses keystroke analysis and user action analysis to identify a user. In [17] a system is described which uses the analysis of typing patterns for theft detection.

Non-technical Information: Non-technical information is collected from internal company sources and usually contains information about the organizational structure. Hierarchical information can be used to classify users. For example temporary employees are usually less trustworthy than permanent ones. Management personnel might be more trustworthy than others and thus are allowed to access more sensitive data and services. Information about employees like the length of the affiliation with the company, professional trainings (e.g., mobile security awareness trainings) taken, can also be used to classify users. These kinds of information can be used to classify users as well as in downstream access control systems.

The combination of different characteristics, technical as well as non-technical ones, enables a more accurate picture of the person using the device.

2) Mobile Device: Security classification of devices is used to evaluate the security and trustworthiness of mobile devices from a technical point of view.

Configuration Monitoring: The configuration of mobile devices includes operating system versions, variants and patch levels as well as information about installed 3rd party apps. Using this information, which is usually supplied by mobile device management systems, it is possible to identify security risks, e.g. non-up-to-date software. An up-to-date system reduces the risks of security vulnerabilities. If there are serious security issues in older software versions of a mobile device a classification in higher security levels could be prohibited.

Device Properties: Mobile devices differ in their hardware configuration. Those features can make a difference in the security of a device, therefore the support and use of such device capabilities is also a factor in device classification. Such device properties include smart card support, which can be used to store digital certificates for authentication purposes, hardware implemented kill pills, for remotely wiping mobile devices, hardware supported encryption, which allows secure storage on mobile devices without putting too much of a burden on the CPU and biometric sensors, which can be used to realize secure and trustworthy authentication.

In the future, virtualization support on mobile devices will be a hot topic in terms of security. With virtualization building distinctly separated environments for parallel personal and business usage of the same device will be possible. This will improve security while handling corporate data and services on mobile devices.

Another characteristic is the mobile device operating systems. iOS and Android, for example, each support different security features and implement them differently. For example the implementation of process isolation or data encryption is done differently on those platforms.

Runtime Information: Runtime information includes collected information about current and historical resource utilization, like CPU load, memory utilization or battery utilization. Using this information, malware could be detected. Additionally, currently running processes and background services should be monitored. This information is sent to the proxy in regular intervals and is accounted for in device security evaluation.

The proxy can be used to collect additional information for security analysis. Network traffic, regardless if it is internal traffic to the corporate intranet or public traffic to the Internet, flows through the cloud-based proxy. This allows for traffic analysis tools to be used. By leveraging deep packet inspection for example, suspicious traffic generated by bots or trojans which communicate with their control instances, can be detected.

The proxy can also be used to create profiles of which network protocols are commonly used and how they are used (e.g., which service is usually used). Deviation from those profiles can be a sign of malware infection.

The proxy is the primary interface of the mobile device to security services like anti virus engines in the cloud. In case those services detect a potential threat, the proxy is informed and uses this information during device security evaluation. One special case is proxy connectivity itself and how it affects device security. It is very likely that there are periods of time where there is no connectivity between proxy and mobile device. This can be because of a GSM/UMTS dead zone or a lengthy stay abroad without data roaming. In these cases the duration between the last connection and the first one after that, must be considered during evaluation, mainly because the mobile device could have been tampered with. Usually, after such a period a mobile device should be regarded as untrusted until a full security check has been performed (either manually or by an automated process).

3) Communication Channel: The communication channel is a critical part of the security evaluation and the resulting security level classification. It is usually not under control of the company but the mobile network operator (MNO). The MNO's data services are used to connect to company intranets and the Internet. But there are also other communication channels (e.g., public access points) which need to be considered in a security evaluation, when accessing company data and services over such channels. The following characteristics have to be paid attention to:

1) *MNO data services* are generally not under the company's control, thus they are to be regarded as insecure. Connectivity to the proxy is established via the Internet using GSM or UMTS. The actual technical details of the network are hidden and usually there is no detailed technical information about the infrastructure and used technologies (e.g., whether and how NAT is used to connect mobile devices to the Internet) available to the MNO's customer.
2) *Public access points* like WLAN in public facilities are also not under the control of the company. Therefore, this communication channel must also be considered as insecure and untrusted.
3) *Known access points* include access points where there is technical information available and transparency is better than in public access points (e.g., the corporate WLAN infrastructure of a partner company). Depending on the actually available information, a better communication security classification is possible when using such access points.
4) *Internal access points* are under full control by the

company. There is full transparency about the technical infrastructure, technologies in use and implemented security measures. Using such access points allows maximum security.

Examining the access point alone is not a sufficient means of security measurement of the communication channel. In fact, the whole channel between mobile device, its access point, stations in between and the proxy has to be taken into consideration. This is especially the case, if a direct connection to proxy is not possible and the connection has to be established via the internet. Analyzing this problem further is out of the scope of this paper. Therefore, an end-to-end encrypted communication channel between mobile device and the proxy is assumed. Examining the security properties of stations in between becomes unnecessary in this case, if the end-to-end encryption is secure and reliable. End-to-end Encryption can be implemented in two ways:

1) *VPNs* are used to encrypt communication between communication partners. Using VPNs to encrypt traffic between mobile devices and their proxies provides maximum security, even if inherently or possibly insecure access points are used (see access points 1-3). In this case, communication security is depending on the security of the deployed VPN technology.
2) *Message encryption* is an alternative to VPNs. In this case not the whole communication is encrypted, but only relevant messages.

A special case is unencrypted communication between mobile device and proxy using an internal access point. This is the only case where it is possible to pass on using VPN or message encryption and still get a high communication security classification. Nevertheless, this is only possible if the communication channel between proxy and mobile device is fully transparent to the company and secure.

4) Backend (cloud): The backend security of the infrastructure, the cloud, with the proxy and the access control module, must be considered as well, but are traditional data center security issues and will not be considered in this paper.

Continuous evaluation of the mobile system based on the taxonomy, security levels can be assigned to each category which is later used for access control.

B. Security Levels

The aforementioned taxonomy influences the access control on company data and services in the cloud or the Internet. Based on continuous evaluation of the particular mobile system parts, security levels (see Figure 3) are assigned and integrated with the classic access control systems (e.g., RBAC, see section IV for further details) to allow fine grained protection of services and data. The overall security level of the mobile system is determined as following: For each part of the mobile system a security Level $Ln_{system\ part}$ where $n = 0, 1, 2, 3, 4$ is identified. The total mobile system security Level (Ln_{system}) is calculated by the minimum of all three single security levels, as stated in the following formula:

$$Ln_{system} = min(Ln_{user}, Ln_{device}, Ln_{communication})$$

with $n = 0, 1, 2, 3, 4$ levels of security.

The security levels either grant broader access rights or deny them. The main decisions, which need to be made are:

- Is the user of the mobile device authentic (has he been sufficiently authenticated)?
- Does the user have access to the requested data and services?
- Does the used mobile device pose a security risk?
- Is the communication channel between device and proxy or rather the requested data and services sufficiently secured?

In the following, the security level state diagram, transitions between security levels as well as mechanisms for applying security levels on access control decisions are described.

1) Security Level Definition: The following section describes the five identified security levels (see Figure 3), ordered by ascending security and trustability.

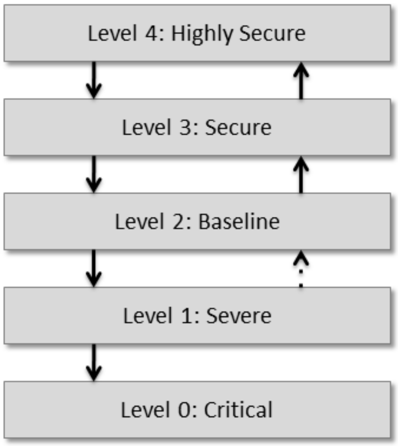

Figure 3. Security Levels

Level 0 (Critical):
The Level 0 security level is the lowest, which can be assigned by the classification process. In this case a highly critical security incident has occurred. If the user classification signals theft or loss of a device, it automatically gets assigned security Level 0. Access to the company's network, services and data is immediately and completely blocked. Further, the removal of all data on the device gets initiated (via a remote wipe of the device), as long there is still connectivity between the proxy and the device. Depending on the company's security policy for lost devices an agent on the device can either be directed to make the device

useless (blocking all communication channels) or to switch into surveillance mode, where GPS location, camera and video information is transmitted to the company for further investigation. This information can be used to to try to re-obtain the device or to initiate legal countermeasures. What has to be done, has to be defined in the company's security policy.

Level 1 (Severe):
The Level 1 security level presumes that the device is in possession of the legitimate owner. Theft or loss can be counted out. Anyhow, there still is a critical security incident. Usually, such incidents will be signaled by security services in the cloud. For example the anti virus system flags the device as compromised because of a detected malware infection or the intrusion detection system throws an alert because of an attempted or succeeded intrusion. Thus, the security classification of the device has failed. A failed security classification means that there is an incident, which is clearly critical. A user installing an unknown app on his device does not per se qualify for Level 1 assignment. Not until the app has been identified as a threat. Just like in Level 0 connectivity to data and services is severely limited and the device is cleaned.

Level 2 (Baseline):
The security level 2 is also known as baseline. User, device and communication classification have not detected a critical problem. All basic services are available and there is connectivity between the mobile device and its proxy in the cloud, but access to data and services is limited, because the full security and trustability of the device cannot be warranted. This can be because the user did not use a sufficiently strong authentication mechanism or there are additional unknown apps installed, which could potentially be dangerous. Another reason for Level 2 assignment is connecting via a public access point without sufficient additional security enhancements, like using a VPN. In Level 2 baseline services like e-mail, calendar and access to non-classified documents are enabled.

Level 3 (Secure):
For accessing confidential services and documents, an elevated security level is required. Level 3 builds upon the properties of Level 2, but requires additional security requirements. This comprises the usage of a VPN, the policy conform configuration of a device (e.g., only explicitly approved apps installed). Of course, the user has to be authenticated using a sufficiently strong mechanism (e.g., user and password combination).

Level 4 (Highly Secure):
The most restrictive security level is Level 4. It can only be assigned if classification attests full compliance to the security policy and additional security mechanisms are used. Such additional mechanisms can be the authentication of the user using biometric information, hardware supported full device encryption and connecting to the network using an internal access point. Only Level 4 allows access to highly confidential internal services and data.

2) Security Level Transitions: The assignment of a security level does not happen linearly. In the following, the transitions between security levels are described:

- L0 → L2 and upwards
  This transition describes the case, where a stolen or otherwise lost device gets regained. In this case, the device is not to be trusted and therefore, has to be classified as insecure and compromised. A full manual audit or a full reset by an administrator is needed for it to be assigned Level 2 or above. This evaluation process must not be automated, but be conducted by a qualified administrator.
- L1 → L2
  A compromised device has to be audited manually. Alternatively a full reset is also possible to reenter a secure state. This process must also be conducted manually by a qualified administrator.
- L2 ↔ L3 ↔ L4
  Transitions between these three security levels can happen automatically. For an assignment to the next higher security level, its security requirements must be fulfilled. For example deinstalling any not explicitly approved apps and connecting to the company's VPN can lead to the automatic upgrade from security Level 2 to 3.
- L2, L3, L4 → L1
  This downgrade usually happens when security services detect critical problem like a virus infection or an intrusion attempt. In this case, the user is informed about the incident and the Level 1 is immediately assigned.
- L* → L0
  Level 0 is assigned if a theft or loss of a device is detected.

IV. Access Control

The aforementioned classification of user, mobile device and communication channels, resulting in a security level of the overall system, has to be integrated into access control systems for services and data. This way classic access control can be enhanced with secure access control for mobile devices. The following section describes two approaches for integrating the security levels into the widely used role based access control model (RBAC) [18].

A. Role-based Access Control

Access control models serve the purpose of limiting access rights of authenticated subjects on certain objects. Subjects can be users, or programs which act on behalf of a user. All access attempts in a system are monitored and evaluated against a rule set of the access control model. This rule set describes which subjects are allowed to perform which actions on objects protected by the system. One important aspect of classic access control models, like RBAC, is the distinction between authentication of a subject and the actual access control. Access control systems assume that a subject is properly authenticated before a decision about the authorization of an action is made [19]. Classic access control models are discretionary access control (DAC), mandatory access control (MAC) and role-based access control (RBAC). While DAC and MAC are important models, RBAC seems to be the more interesting model for the further discussion of integrating security levels. More modern approaches like the usage control model (UCON) have yet to prove their importance in real world systems. The fact that RBAC provides an abstraction of real world organizational structures, its wide adoption in software systems and the possibility to implement DAC as wells as MAC models simply by adjusting the RBAC model [20], made it the candidate of choice for further discussion.

The role-based access control model has been unified in 2000, based on the works of Ferraiolo, Kuhn and Sandhu and formally adopted as an ANSI Standard in 2004 [21]. This ANSI standard serves as a basis for further analysis. The basic elements of RBAC are users, roles, sessions and permissions. Users of a system are assigned one or more roles which they can assume. Depending on the role, access to subjects is either granted or denied. Users are assigned to roles using user assignments. Roles describe a function within an organization and the rights and obligations associated with it. Permissions describe operations which can be executed on RBAC-protected objects [21]. This is the foundation of the core RBAC model. Furthermore, there are some extensions to this core model which make it more flexible. One of these extensions is the RBAC 2 model, also called constrained RBAC. With this model it is possible to implement separation of duty concepts into the RBAC model. So called constraints allow a more fine-grained control over the RBAC model.

Despite of the RBAC model already being released as an ANSI standard, there is still research being conducted. Neumann and Strembeck [22] describe an extension to the RBAC 2 model, called context constraints. This type of constraints is used to evaluate predefined conditions at access control decision time. They allow the integration of RBAC model external conditions into the system. Thus, a context condition must be met, before an operation to which it is linked can be performed. One or more context conditions, which evaluate values of context attributes, form a context constraint. Apart from the roles and operations defined in the RBAC model, an unmet context constraint can prohibit the execution of an operation, which would otherwise be perfectly valid without context constraints. In comparison to the constrained RBAC model, Neumann and Strembeck enhance the concept of constraints in a way that makes them more generally applicable, especially the possibility of evaluating information from external databases (e.g., literally an external company database which contains employee records). An example for a context constraint is that users are allowed to access a certain document only in between 8am and 6pm, regardless of them assuming a role which has enough rights to do so or not.

1) RBAC Security Level Integration: Following the basics of the classic RBAC model this section will describe two ways of integrating security levels for mobile devices in RBAC. The first approach is based on extending the RBAC model with previously described context constraints while the second approach uses a two phase flow of access control.

Integration by RBAC Extension: This approach integrates security level requirements into RBAC using context constraints (see Figure 4 (S)ecurity (L)evel Check = Context Constraints). This means in addition to needing a specific role for accessing certain objects, a certain minimum security level is also required. The minimum security level is a context condition and the currently applied security level for a device is a context attribute. Together they form a context constraint which is bound to an operation. Before an actual access decision, based on the user, his role and the object, is made the context constraint is evaluated. If and only if the context constraint is met, the access control decision is made. During evaluation of the context condition, the current security level is pulled from the proxy (see Figure 4 step 3). The proxy is always informed about the currently applied security level. Is the current security level (context attribute) equal or higher than that defined in the context constraint of the object, the evaluation of the access control decision may continue. If the current security level is less than required, no further evaluation takes place and access is blocked.

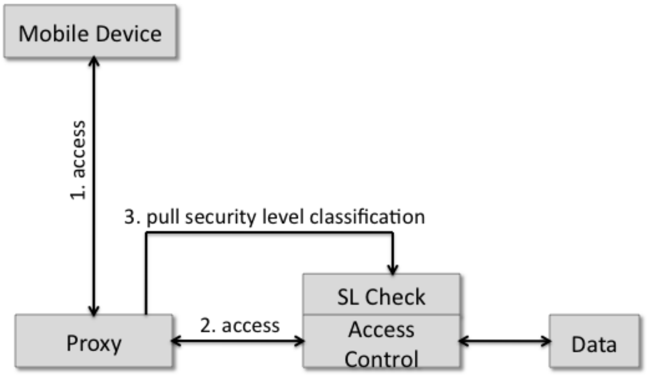

Figure 4. RBAC Security Level Integration - RBAC Extension

The main advantage of this approach is the tight integration of the security level concept into the RBAC model. But, there are also difficulties like the reduced flexibility by having to always integrate security levels into all access control systems which are in use. The pull mechanism during the context constraint evaluation can also be a problem. As previously described, the assignment of a security level can change abruptly because of the continuous security evaluation of the user, device and communication channel. Therefore, with each access control decision, the current security level needs to be determined. This can happen quite often and thus degrade performance significantly depending on the deployment model (e.g., access control decision point needs to communicate with the proxy via a network).

Integration by Two Phase Flow: The two phase flow splits security level evaluation and actual access control decision into two phases. Figure 5 illustrates this concept. The proxy is the central component for accessing any services on the intranet and the internet and also stores the current security level assignment. Because of this it can easily be used to control security level evaluation. Every object (e.g., data object in Figure 5) possesses a minimum security level, which needs to be matched to gain access via a mobile device. This information is stored in the access control system. Usually, this information is very static and does not change too often. The proxy stores a copy of these object/minimum level mappings. If the required minimum level is changed, the updated mapping is pushed to the proxy. Now, if a mobile device is requesting access to access control protected data or services, the proxy first evaluates whether the minimum security level requirement is met or not. If it is, the request passes for further evaluation by the access control system, if not the request is refused by the proxy.

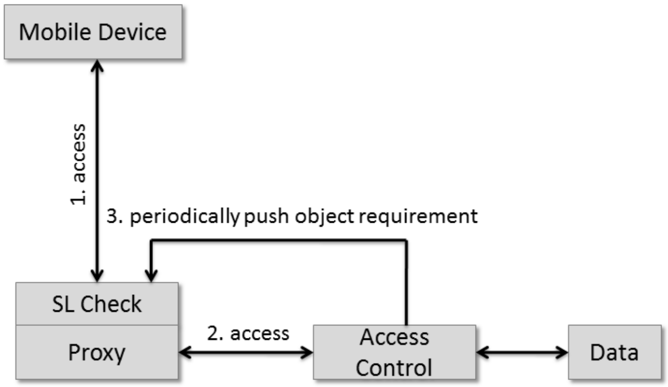

Figure 5. RBAC Security Level Integration - Two Phase Flow

This approach has the advantage of being access control model agnostic. It actually does not matter which system is used for access control, as long as object's requirements are available to the proxy. Also, the proxy can be used to terminate requests even before they reach the access control system located behind it. Pushing object requirements on update or in regular intervals also greatly reduces round trips during fetching of the current security level assignment. A problem of this approach is having to keep the same information (minimum requirements for objects) synchronized in two separate locations.

Both approaches have advantages and disadvantages. Higher flexibility, compatibility and a better communication flow are advantages of the two phase approach.

V. Evaluation by Use Case

In this section an evaluation of the proposed security level and classification concept is performed by using use cases for a better illustration. The general context of these use cases is the usage of mobile devices in a company. Sensitive documents may be stored on mobile devices. There is also an IT security policy in place, which sets the basic rules for using mobile devices (e.g., VPN, user authentication mechanisms, trusted software packages et cetera). Table I presents a selected overview of the most interesting use cases. The first column is used to describe preconditions (the state of security classification *before* a specific incident happens). The second column does the same for postconditions (the state of security classification *after* a specific incident happened). The overall security level classification is evaluated by choosing the current minimum security level of user, device or communication. Certain requirements for reaching a specific security level, e.g. having installed only known apps for reaching level 3 in device classification, is subject to concrete company security policies. Use cases 1 to 3 describe typical scenarios where the security level is lowered because of a security incident detected by the described system, whereas use cases 4 to 6 show how security level upgrades work.

The use cases show, how the security level concept for mobile devices allows to dynamically and continuously adjust their security classification. This allows a more controlled and more secure access of protected data as well as the overall improvement of the security of mobile devices in an enterprise environment.

VI. Conclusion and Future Work

In this paper, we demonstrated why security of mobile devices, like smartphones and tablets, in enterprise environments will be an important issue in the next couple of years. We also proposed a framework based on security levels and classification of user, device and communication which could improve security when handling confidential company data on such devices. Based on an ongoing classification security levels are applied and are evaluated during access on protected data. The integration of these two concepts into the well known RBAC model was also an important issue, discussed in this paper. We provide two possible solutions: one, which integrates tightly with the RBAC model using an extension called context constraints and another approach based on two-phase evaluation. Two-phase evaluation allows the decoupling of classic access control system and additional access control for mobile devices. At

Table I
EVALUATION OF THE SECURITY LEVEL CONCEPT FOR MOBILE DEVICES USING USE CASES

No.	Pre-Incident Security Level	Post-Incident Security Level	Use Case Description
1	min(L3, L3, L3) = L3	min(L0, L3, L3) = L0	The owner of the device is authenticated using PIN and additional username/password. His device adheres to general policies but has a game app installed, which is known to not contain malware but could lead to privacy problems. The connection to the proxy is established via UMTS using the company's VPN. Now, the device is stolen, while the owner is distracted. The thief fails three times to enter username/password correctly upon unlocking the screen. The mobile device agent reports this incident to the proxy which starts countermeasures according to the security policy for stolen devices.
2	min(L3, L4, L3) = L3	min(L3, L1, L3) = L1	The owner of the device is authenticated using PIN and additional username/password. His device configuration adheres strictly to the policies in place. Now, the user installs a new app from the app store. This app is scanned for malware in the cloud. The scanning engine detects a trojan inside the app and reports this incident to the proxy. The proxy starts countermeasures to protect the company's network.
3	min(L3, L4, L3) = L3	min(L3, L2, L3) = L2	Preconditions are the same as in the previous use case. The user installs an unknown app. The app is checked for malware without a positive result. To protect the company's data and network from a potential 0day-attack the security level is lowered.
4	min(L3, L4, L4) = L3	min(L4, L4, L4) = L4	The user is authenticated using username/password. The device's configuration matches security policy 100%. The connection to the proxy is established using the company's internal WLAN and VPN. Maximum security is guaranteed and there are no restrictions due to mobile access. The user now needs access to documents which are highly confidential an therefore require security level 4. The user now chooses to authenticate himself with additional biometric information using the fingerprint scanner. User classification is now upgraded to level 4, which enables an overall classification of 4, allowing access to the protected documents.
5	min(L3, L2, L3) = L2	min(L3, L3, L3) = L3	The user is authenticated using username/password. The device's configuration adheres to general security policy, but an unknown app is installed. The user now needs access to level 3 protected services. To achieve an upgrade, the user uninstalls the app. The proxy now registers that the unknown was removed and upgrades the device classification to level 3, resulting in an overall level 3 classification.
6	min(L4, L4, L2) = L2	min(L4, L4, L3) = L3	The user is authenticated using additional biometric information. The device's configuration matches security policy 100%. The connection to the proxy however is done using a public access point without using the company's VPN only relying on application based communication encryption (e.g. using IMAPS, HTTPS). The user needs access to internal documents requiring him to be classified as level 3. Therefore he establishes a secure VPN connection, which grants communication classification upgrade to level 3, resulting in an overall classification of 3.

last, we provide an evaluation of our approach which is used to demonstrate the feasibility using use cases.

In our future work we will concentrate on the process of securely collecting data (e.g. with the help of trusted infrastructure) about all the participants in our proposed framework and using that data for security classification. The proxy as well as data collected directly on the devices and the trustworthiness of the data will be in the center of our future examination. Another problem to solve will be data privacy protection, because of the very restrictive laws existing in Germany.

REFERENCES

[1] IBM Corporation, "IBM X-Force 2011 Mid-year Trend and Risk Report," https://www14.software.ibm.com/webapp/iwm/web/signup.do?source=swg-spsm-tiv-sec-wp&S_PKG=IBM-X-Force-2011-Mid-year, 2011, [retrieved: May, 2012].

[2] Android Developers, "Android Platform Versions - Current Distribution," http://developer.android.com/resources/dashboard/platform-versions.html, 2012, [retrieved: May, 2012].

[3] B.-G. Chun and P. Maniatis, "Augmented smartphone applications through clone cloud execution," in *Proceedings of the 12th conference on Hot topics in operating systems*, ser. HotOS'09. Berkeley, CA, USA: USENIX Association, 2009, pp. 8–8.

[4] E. Cuervo, A. Balasubramanian, D.-k. Cho, A. Wolman, S. Saroiu, R. Chandra, and P. Bahl, "Maui: making smartphones last longer with code offload," in *Proceedings of the 8th international conference on Mobile systems, applications, and services*, ser. MobiSys '10. New York, NY, USA: ACM, 2010, pp. 49–62.

[5] J. Oberheide, K. Veeraraghavan, E. Cooke, J. Flinn, and F. Jahanian, "Virtualized in-cloud security services for mobile devices," in *Proceedings of the First Workshop on Virtualization in Mobile Computing*, ser. MobiVirt '08. New York, NY, USA: ACM, 2008, pp. 31–35.

[6] G. Portokalidis, P. Homburg, K. Anagnostakis, and H. Bos, "Paranoid android: versatile protection for smartphones," in *Proceedings of the 26th Annual Computer Security Applications Conference*, ser. ACSAC '10. New York, NY, USA: ACM, 2010, pp. 347–356.

[7] S. A. Zonouz, K. R. Joshi, and W. H. Sanders, "Cost-aware systemwide intrusion defense via online forensics and on-

demand detector deployment," in *Proceedings of the 3rd ACM workshop on Assurable and usable security configuration*, ser. SafeConfig '10. New York, NY, USA: ACM, 2010, pp. 71–74.

[8] A. Houmansadr, S. Zonouz, and R. Berthier, "A cloud-based intrusion detection and response system for mobile phones," in *Dependable Systems and Networks Workshops (DSN-W), 2011 IEEE/IFIP 41st International Conference on*, june 2011, pp. 31 –32.

[9] A. Shabtai, U. Kanonov, Y. Elovici, C. Glezer, and Y. Weiss, "Andromaly: a behavioral malware detection framework for android devices," *Journal of Intelligent Information Systems*, pp. 1–30, 2011.

[10] A.-D. Schmidt, R. Bye, H.-G. Schmidt, J. Clausen, O. Kiraz, K. Yuksel, S. Camtepe, and S. Albayrak, "Static analysis of executables for collaborative malware detection on android," in *Communications, 2009. ICC '09. IEEE International Conference on*, june 2009, pp. 1 –5.

[11] A.-D. Schmidt, F. Peters, F. Lamour, and S. Albayrak, "Monitoring smartphones for anomaly detection," in *Proceedings of the 1st international conference on MOBILe Wireless MiddleWARE, Operating Systems, and Applications*, ser. MOBILWARE '08. ICST, Brussels, Belgium, Belgium: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2007, pp. 40:1–40:6.

[12] A. Bose, X. Hu, K. G. Shin, and T. Park, "Behavioral detection of malware on mobile handsets," in *Proceeding of the 6th international conference on Mobile systems, applications, and services*, ser. MobiSys '08. New York, NY, USA: ACM, 2008, pp. 225–238.

[13] H. Kim, J. Smith, and K. G. Shin, "Detecting energy-greedy anomalies and mobile malware variants," in *Proceeding of the 6th international conference on Mobile systems, applications, and services*, ser. MobiSys '08. New York, NY, USA: ACM, 2008, pp. 239–252.

[14] T. Martin, M. Hsiao, D. Ha, and J. Krishnaswami, "Denial-of-service attacks on battery-powered mobile computers," in *Pervasive Computing and Communications, 2004. PerCom 2004. Proceedings of the Second IEEE Annual Conference on*, march 2004, pp. 309 – 318.

[15] R. Racic, D. Ma, and H. Chen, "Exploiting mms vulnerabilities to stealthily exhaust mobile phone's battery," in *Securecomm and Workshops, 2006*, 28 2006-sept. 1 2006, pp. 1 –10.

[16] D. Nash, T. Martin, D. Ha, and M. Hsiao, "Towards an intrusion detection system for battery exhaustion attacks on mobile computing devices," in *Pervasive Computing and Communications Workshops, 2005. PerCom 2005 Workshops. Third IEEE International Conference on*, march 2005, pp. 141 – 145.

[17] A. Gupta, D. Gupta, and N. Gupta, "Infosec-mobcop - framework for theft detection and data security on mobile computing devices," in *Contemporary Computing*, ser. Communications in Computer and Information Science, S. Ranka, S. Aluru, R. Buyya, Y.-C. Chung, S. Dua, A. Grama, S. K. S. Gupta, R. Kumar, and V. V. Phoha, Eds. Springer Berlin Heidelberg, 2009, vol. 40, pp. 637–648.

[18] D. Ferraiolo and R. Kuhn, "Role-based access control," in *In 15th NIST-NCSC National Computer Security Conference*, 1992, pp. 554–563.

[19] R. S. Sandhu and P. Samarati, "Access control: Principles and practice," *IEEE Communications Magazine*, vol. 32, pp. 40–48, 1994.

[20] S. Osborn, R. Sandhu, and Q. Munawer, "Configuring role-based access control to enforce mandatory and discretionary access control policies," *ACM Trans. Inf. Syst. Secur.*, vol. 3, pp. 85–106, May 2000.

[21] American National Standard Institue Inc., "American National Standard for Information Technology - Role Based Access Control," http://profsandhu.com/journals/tissec/ANSI+INCITS+359-2004.pdf, 2004, [retrieved: May, 2012].

[22] G. Neumann and M. Strembeck, "An Approach to Engineer and Enforce Context Constraints in an RBAC Environment," in *In Proc. of the 8th ACM Symposium on Access Control Models and Technologies (SACMAT)*. ACM Press, 2003, pp. 65–79.