FAKULTÄT FÜR !NFORMATIK

Faculty of Informatics

Structuring and Enhancing IT Security in Core Mobile Phone Use Cases

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur/in

im Rahmen des Studiums

Software Engineering & Internet Computing

eingereicht von

Andrej Lehotsky
Matrikelnummer 0525176

an der
Fakultät für Informatik der Technischen Universität Wien

Betreuung
Betreuer: Thomas Grechenig

Wien, 05.09.2011

(Unterschrift Verfasser/in)

(Unterschrift Betreuer/in)

Structuring and Enhancing IT Security in Core Mobile Phone Use Cases

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Software Engineering & Internet Computing

eingereicht von

Andrej Lehotsky
0525176

ausgeführt am
Institut für Rechnergestützte Automation
Forschungsgruppe Industrial Software
der Fakultät für Informatik der Technischen Universität Wien

Betreuung:

Betreuer: Thomas Grechenig

Wien, 05.09.2011

Eidesstattliche Erklärung

Ich erkläre an Eides statt, daß ich die vorliegende Arbeit selbständig und ohne fremde Hilfe verfaßt, andere als die angegebenen Quellen nicht benützt und die den benutzten Quellen wörtlich oder inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Wien, am 05.09.2011 --

Andrej Lehotsky

Ein Mensch sah Leiden, unbeschreibbares Leiden. Er sprach zu Gott: "Oh Gott, wie konntest du nur so etwas zulassen. Unternimm doch dagegen etwas!" Und Gott antwortete ihm: "Ich habe bereits etwas dagegen unternommen. Ich habe dich erschaffen..."

Bruno Ferrero

Contents

Abstract

Mobile phones are emerging to personal mobile devices, which can already replace a personal computer in many use cases. But their overall security mechanisms are not sufficient. The data stored in today's devices are much more personal then data stored on any other device we use.

The focus in this thesis are operating systems and additional installed applications. The mobile phone attack surface is examined and based on it, possible and by researchers documented examples of threats are summarized. Since the mobile and fixed environments have their differences, these are outlined as the consideration for improvement approaches. Because of the human factor, some results of awareness surveys are presented and discussed.

Due to limited resources of mobile phones (mainly power supply and size and/or weight), the current research in this field mainly concentrates on off-device (cloud) security services. These off-device security services are now the only solution to save power supply consumption while inferring possible attacks. Another focus is the virtualization and problems interconnected with installation of new applications. Summing up, this thesis proposes a mobile phone attack surface including open problems, which have to be solved to improve the overall security of mobile phones.

Zusammenfassung

Mobiltelefone entwickeln sich zu persönlichen Mobilgeräten, die den PC in vielen täglichen Anwendungsbereichen jetzt schon ersetzen können. Aber deren Sicherheitsmechanismen sind oft nicht ausreichend. Daten, die in den Mobiltelefonen gespeichert werden, sind oft mehr personenbezogen als Daten in einem anderen Gerät, das im Alltag verwendet wird.

Im Fokus dieser Diplomarbeit steht die vergleichende Analyse von Betriebsystemen im Mobiltelefonbereich und zusätzlich installierten Applikationen. Eine Klassifikation der Angriffsfläche von Mobiltelefonen wird vorgeschlagen und darauf basierend werden relevante Sicherheitslücken aufgezeigt, wobei grundlegende Unterschiede zwischen mobilen und fixen Umgebungen berücksichtigt werden. Weil die BenutzerInnen einen menschlichen Faktor darstellen, werden Erhebungen über Sicherheitsbewusstsein präsentiert.

Aufgrund der limitierten Ressourcen mobiler Geräte (meistens wegen der Energieversorgung und Größe) konzentriert sich die aktuelle Forschung in diesem Bereich vor allem auf (Cloud) Sicherheitsdienste außerhalb des Mobiltelefons. Diese Sicherheitsdienste, die außerhalb des Geräts bereitgestellt werden, sind heute mittlerweile die einzige Lösung, um mögliche Angriffe festzustellen, ohne dabei zu viel Energie zu verbrauchen. Weitere Schwerpunkte der Arbeit sind Virtualisierung und Probleme, die mit der Installation von neuen Applikationen verbunden sind. Zusammenfassend bietet diese Diplomarbeit eine Klassifizierung der Angriffsflächen von Mobiltelefonen und zeigt Schwachstellen auf, deren Beseitigung die allgemeine Sicherheit von Mobiltelefonen erheblich verbessern würde.

Acknowledgements

First of all I would like to thank Thomas Grechenig for the possibility to write my master thesis at INSO, for his supervision and for the possibility to work on a such interesting and very crucial topic of security.
Then I would like to thank Christian Schanes for his remarkable discussions during the development of this thesis. I attended his course of Advanced Internet Security as my last course at Vienna Technical University and our discussions on the topics of mobile communication were very useful and inspiriting to me.
I am also pleased to acknowledge the help of Christian Hattinger as my colleague, who took time to read many versions of this thesis and supplied me with many useful hints and considerations for this work. It was a great help to have a computer science colleague, which whom I could always discuss my argumentation and any other questions during the development of this thesis.
Since I am not a native english speaker, I also have to thank to my best friend Michal Valiga and his language improvements. Since computer science is not his focus, it was surely not easy for him to understand the content.
And finally I would like to thank my wife and all the family, since I spent many hours to develop this work (and much more for the whole study), but should have spent many of these hours with them.

Chapter 1

Introduction

In this introduction of the following master thesis, the problem description and motivation, followed by the state-of-the art, are described. After a goal definition, this chapter is finished by describing the composition of the thesis.

1.1 Problem Description

Many users are aware of security issues with their PCs and some of them are trying to secure their devices. As the number of devices connected to the world wide web grows, more users are connected to the Internet trough mobile phones (in some cases, phone is not the right term, maybe "device" would be more precise) and trough regular PCs. Browsing the web with mobile phones raises new possible risks, some of the main threats are the missing awareness of the mobile phone users; missing security standards for producers/sellers of such mobile devices; and the accumulation of sensible, user related data while using mobile phones.

This master thesis will introduce and explain existing problems in mobile phone use case scenarios while discussing the current possible attacks and security mechanisms, trying to search for analogies in the world of personal computers. By discussing these issues, the importance of security in the field of mobile phones will be underlined.

Many now available statistics claim, that the world of mobile devices is now dominating over the world of PCs while examining the connection to the Internet [ITU09]. Some years ago children at primary school started to play with mobile phones. Many of us had already one or more mobile phones carrying around always and everywhere with us. Starting with phone calls and SMS (Short Message Service allows exchange of short text messages between mobile phone devices), following with simple games. Nowadays, children at school are playing java games against each other connected via Bluetooth (an open technology standard for wireless transfer of data over short distances) or Internet directly. Today, mobile devices such as iPhone or iPad have either more possibilities: surfing on the web, working with office-based applications, nearly infinite possibilities through development of new applications.

The necessity of a security model for privacy protection on mobile phones is obvious because of its lack as introduced for example in [SDLC09]. Considering security, following crucial questions raises, which will be discussed in this master thesis: Are todays mobile devices secured to use them for private personal data (like banking, government, e-health)? Have todays devices some kind of security conceptions? How can these mobile devices be misused? How could we achieve some level of security?

The use case of such mobile phone is different to for example a laptop usage: a laptop can be disconnected from the Internet in case of a security threat, but disconnecting a mobile phone from the network would make it useless. Mobile devices are going to replace personal computers in many utilizations of the daily life (mail, chat, instant messaging, viewing media, composing simple documents, etc).

The growing possibilities through development of new applications for mobile devices brings risks. One of the possibilities to control the implementation and usage of such applications is the certification (by third parties) of such applications, proposed for example in [EOM09]. The most spread (in daily use) business approach in this field is the certification process in Apples AppStore. Banking and other financial operations undertaken via Internet are prone to attacks and therefore have to be secured.

The adopted use cases from the well-known world of PCs to the now discussed world of mobile devices are interesting. Starting by the simple use case of borrowing a mobile phone to a friend to make a phone call, as studied in [KBS09]. Following through viewing different multimedia content (news, online newspapers, social networking, videos, podcasts etc.) and other free time activities. And ending with critical tasks as secure sending of emails, as studied for example in [Too08], or working (receiving, editing and sending) with office applications on critical business data.

The main problem is the awareness of the possible security threats. People carry their mobile phones everywhere and the possible functionalities of such devices are growing each day, but the security issues have to be considered. Many of todays mobile phones do not have a firewall or antivirus application like our personal computers, even if there are analogies between the world of mobile phones and PCs. But many users collect confidential data on their mobile devices (sometimes much more than on their PCs) and are not afraid of evil attacks on them.

1.2 Goals

This master thesis will introduce and explain current security issues and possible security mechanisms in mobile phone use case scenarios. The current problems and threats will be discussed and the current security initiatives explained based on general security goals and on a proposed classification. Discussing some possibilities and risks in the future, the necessity of security improvement will be underlined.

From the scientific point of view, these problems have to be studied and some methods and/or models to achieve a higher level of security have to be developed. Not only standards will help, studying the current business models and investigating them for possible future security issues is also necessary and is also done in this thesis. People have to be aware of security issues and therefore the scientists have to examine these existing problems and try to introduce some models to find a solution for them. These problems will be underlined with this master thesis. Trying to find some possibilities to improve the current security and sum up the problem fields, where changes are needed, can raise the current discussion and academic research in this field and by this help to improve the security of mobile devices in the future. The goal of this thesis is to sum up the current problems and possible security issues (inclusive their classification) in mobile phone use case scenarios, propose some global changes and enhance current business models and/or use cases that could help to improve the security while using such devices.

1.3 Composition

After introduction into the topic, the simplified hardware (HW) and software (SW) composition of mobile phones is summarized according to their worldwide utilization statistics like [Metb]. To be able to discuss security threats of these devices, security in general has to be introduced before. This is done by inspecting the attacker types and security protection goals in general.[BA10], [Sch00]
By explaining the used operating systems (OSs) of mobile devices, search for relevant information in the technical knowledge is done. The basic principles of such OSs are explained to introduce some possible future problems in the field of security. [MSS+08]
When introducing and explaining existing and possible future security issues, the undergoing and planned research in security of mobile phone use cases is discussed and the threats are classified. By this, the current sate-of-the-art is examined and some future possibilities are introduced. [SDLC09]
Finally, while summing up the main findings in this research work, some possible (and even necessary) refinements in the field of mobile phone security are proposed.
This is a rather new topic and not so many scientific work has been done (published) yet in the field of the mobile phone use case scenarios security as in the field of personal computers security. Therefore, no dependencies on undergoing scientific research are expected. But the development of such mobile phones expands day by day and therefore the main focus should be on common, more general security issues like the consideration of use case scenarios.

Chapter 2

Introduction into Mobile Phone System Infrastructure

Mobile phones are complex devices. For the security considerations of this thesis, these devices can be divided in three main parts: hardware interfaces, operating system and mobile applications. Hardware interfaces and their drivers accessible to the OS enable the device to communicate with its environment. The OS itself is the main SW part of the device and enables the coordination and management of the whole HW equipment and provides libraries for other, optional, SW equipment. And the third part are the installed applications, which are utilizing the mentioned OS libraries. All these parts are discussed in this chapter with the goal to provide sufficient understanding of the SW design of a mobile phone. Another goal of this chapter is to develop a very simplified general model of such a mobile phone, to provide a base for future security considerations. Therefore details of the HW composition of the mobile device itself are for the discussions of this master thesis not crucial and will be skipped to underline the focus on their use cases.

2.1 Hardware Interfaces

HW interfaces define the interaction and communication possibilities of devices in general. All devices have their versions and not each version contains all possible HW interfaces. Therefore the following discussion concentrates on the most common interfaces, which can be found in many models of mobile devices.

Today's mobile phones are more and more equipped with sensors like GPS (Global Positioning System, a satellite navigation system for establishing the global position on earth) or accelerometer. One of the first built-in sensors of a mobile phone is obviously the microphone. Thanks to this sensor, the possibility of acoustic communication was enabled. But with evolution of these devices and their possible SW equipment, also other interfaces where added. The main intent of these new emerging interfaces is the possibility to communicate with the surrounding environment, for example sensing the light, movement, direction etc.

Each mobile phone consists of above mentioned hardware interfaces (sensors) and their device drivers enabling the OS to communicate with them. To underline their relevance for the later security discussions, the following of them are summarized in this section:

- GPS and A-GPS (Assisted GPS, the improvement lies in utilizing other positioning information such as the used cell tower by cellular phones): retrieving the position
- Accelerometer: retrieving the approximate speed of a moving device
- Microphone: retrieving the sound from the environment

- Compass: additional positioning information for retrieving the orientation
- Light sensor: additional information about the lightness of the environment

Another group of interfaces enables the communication with other devices and systems in the environment. These are communication interfaces on different network layers as for example Bluetooth or WLAN (Wireless Local Area Network, a wireless data distribution method between two or more devices connected in one network). For further details, the different network layers with their concrete protocols and implementation can be found for example in [Tec05], or in [Sto02] with the focus on wireless networks and mobile computing; they will not be deeply discussed in this thesis. While examining the communication possibilities of mobile phones, among others, following communication interfaces are recapitulated in this section for further discussions:

- GPRS (General Packet Radio Service, a packet oriented mobile transmission protocol), UMTS (Universal Mobile Telecommunications System, one of the mobile telecommunication technologies, also known as "3G"), HSCSD (High-Speed Circuit-Switched Data, an enhancement to CSD) and other data transmission protocols enabling receiving of data through the mobile network
- IP (Internet Protocol, a packed-switched transmission protocol based on unique addresses of communicating endpoints) based services enabling IP-address-based communication
- WLAN enabling connection to the wireless network and communication over this channel
- Bluetooth for short distance communication

2.2 Operating Systems of Mobile Phones

A mobile phone can be from the technological point of view divided, like many other electronic devices, in two main parts: the mobile device hardware and the mobile device software. The mobile device software can be than coarse divided into the operating system (OS), a kind of firmware (responsible for communication between mobile device hardware and the OS) and installed applications. According to statistics of the worldwide usage of mobile devices (February 2010), the three most used OSs are: 1. iPhone, 2. Android, 3. Symbian. These are followed by other, not so widely used, OSs: RIM, Windows Mobile, etc. [Metc]
In the following section, each of the three most used OSs will be discussed

with the aim to provide enough information for a general model of a mobile phone OS.

2.2.1 Layers of the iPhone OS 3.0

Development of the iPhone OS is based on the principles of Mac OS X and therefore many similarities exist between these two OSs. A layered view of the iPhone OS, as shown in Figure 2.1 and described in the following section, defines four main layers: [App09]

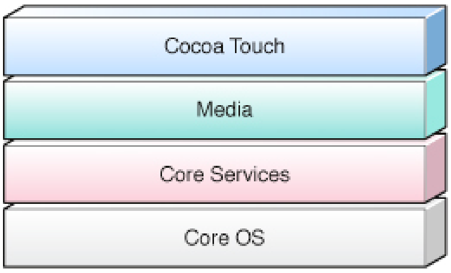

Figure 2.1: Layers of iPhone OS from [App09]

Core OS Layers [App09]

External communication: the CFNetwork framework is based on BSD sockets (network endpoints, which is the IP address combined with the port number, are represented as sockets) and provides an interface to cooperate with following technologies:

- use BSD sockets
- create encrypted connections using SSL (Secure Socket Layer is a cryptographic protocol for secure communication) or TLS (Transport Layer Security is a successor of SSL)
- work with HTTP (Hypertext Transfer Protocol is a standard request-response based protocol for client-server communication), authenticating HTTP and HTTPS (Hypertext Transfer Protocol Secure is a combination of HTTP and SSL/TLS) servers
- work with FTP servers (File Transfer Protocol used for transferring data from one host to another)
- publish, resolve, and browse Bonjour (a service discovery protocol) services

Communication with accessories: The Accessory Support framework allows the connection of accessories to the device, which can be connected either through the 30-pin connector (the only one cable connector of such devices) or via Bluetooth. The connection process is the following: after obtaining the information about the external device, a communication session can be established. In this communication session and through this framework, the external device can be controlled by using its provided commands.

Security: The Security Framework allows the application to secure its data. With the provided interfaces, following technologies are supported:

- managing certificates
- public and private keys
- trust policies
- generation of cryptographically secure pseudo random numbers
- storage of certificates and cryptographic keys in the keychain
- support for symmetric encryption, HMAC (Hash-based Message Authentication Code combines a hash-value and a secret key for message authentication codes), and Digests (functions compatible with the OpenSSL library, the open source implementation of SSL and TLS)
- share key chain items among multiple applications

System: This is the lowest level of the OS and comprise of the Mach (operating system microkernel developed at Carnegie Mellon University) based kernel (central component of each OS architecture establishing the communication of the application part of the OS and the low-level part of the OS like CPU (Central Processing Unit, also known as ”processor”), memory and HW parts), drivers and low-level UNIX interfaces of the OS. Access to this low-level interfaces is limited to the system and the security framework, developed applications can access following of them:

- networking (BSD sockets)
- threading (POSIX threads, ”Portable Operating System Interface for Unix” standard for threads)
- file-system access
- standard I/O (Input/Output)
- Bonjour and DNS (Domain Name Service, a naming service for participants of a network) services

- locale information
- memory allocation
- math computations

Core Services Layer [App09]

Address Book: The Address Book framework supplies interfaces to manage the Address Book of the device - an application can retrieve and modify the user's contact list.
Core Data: The Core Data framework provides special services and simplifications while using a data model in an application.
Core Foundation: The Core Foundation framework is a data management and service feature, providing methods for: management of date, time, string, raw data block, preferences; manipulation of URL (Uniform Resource Locator specifies the address of a resource) and streams; threads and run loops; port and socket communication.
Core Location: The Core Location framework enables a location of the device (latitude, longitude) by utilizing GPS, cell network, WLAN signals and the built-in compass.
In App Purchase: The Store Kit framework provides interfaces for purchases via the iTunes account. An application can request from the user to buy an additional content for the application (for example in case of a game, the application can offer additional game levels for additional costs). This interfaces secure the financial transaction, informing about the successful/unsuccessful payment transaction.
SQLite: The SQLite library enables a lightweight SQL (Standard Query Language, a standard database language) database for applications.
XML Support: with an built in parser, the interface offers methods for retrieving the XML (Extensible Markup Language, rules for encoding of documents) elements, modifying the XML and a simple export to HTML (HyperText Markup Language, dominant markup language of web pages).

Media Layer [App09]

This layer provides many interfaces for audio and video. An application can record, modify and play audio and video content. Beside these main functions, it supports the developer of an application with the needed UI (User Interface) functionality for development of an application UI.

Cocoa Touch Layer [App09]

This is the highest level of the OS, which enables functions and interfaces build upon the other OS layers. An application developer mostly starts developing an application here by utilizing these services, following the deeper levels in case of necessity for specific functions. The provided interfaces are for example: Apple Push Notification Service for notifying a user of an event; In App Email enabling applications to send e-mails; Peer to Peer support for peer-to-peer connectivity and in-game voice features.
An important framework of this layer is the UIKit framework. It allows, besides the development of UI, access to the internal HW(hardware)-devices and enables event driven reactions in the application. To name only some of them: Accelerometer data; built-in camera; user's photo library; device name and model information; battery state information; proximity sensor information; support for handling touch and motion-based events; accessibility support for disabled users; etc.

2.2.2 Layers of the Android OS 2.1

Development of the open source Android OS is based on the Linux kernel and includes the operating system, the middleware and some key applications. Figure 2.2 shows the five layers of the OS, which will be described in the following sections.

Linux Kernel [Goo10]

The communication between the OS and the software layer is based on the Linux kernel 2.6 and includes the main services and interfaces for process management, memory management, network layer, drivers and security.

Android Runtime [Goo10]

Relying on the Linux kernel, register-based Dalvik is used as the VM (Virtual Machine, a virtual implementation of a machine that acts like a physical machine) and the core provided libraries are based on the Java VM. By starting an application under Android OS, the Dalvik VM starts a separate process for each application. Executables of applications are compiled with the Java language compiler and then transformed in the "dex" called Dalvik format.

Libraries [Goo10]

The Android system includes some core C/C++ (programming languages) libraries and provides them to the application developers through the Android application framework.

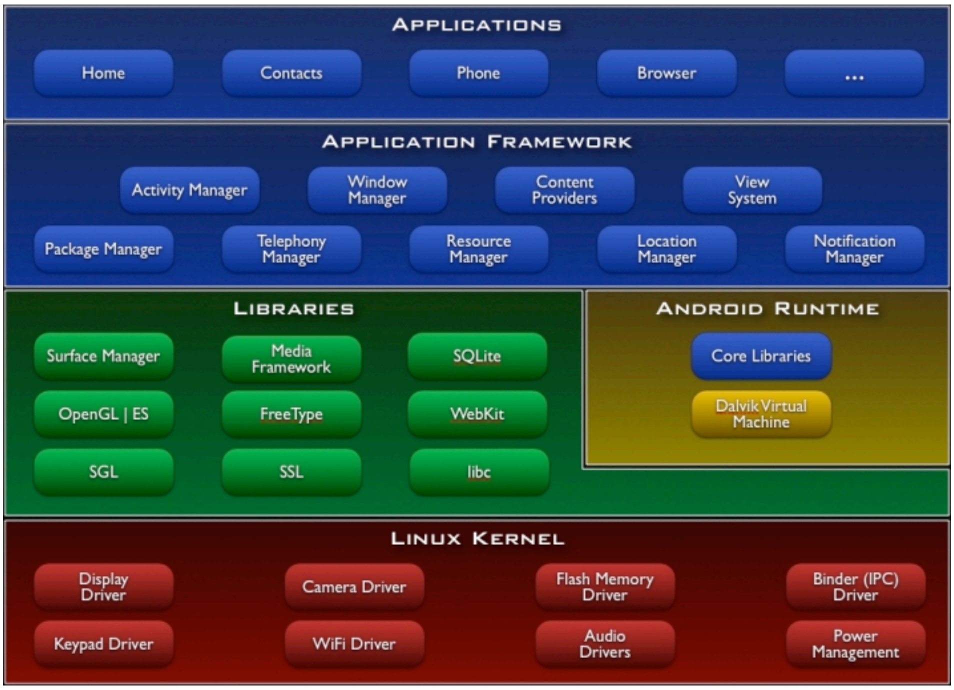

Figure 2.2: Layers of Android OS from [Goo10]

System C library: The standard C system library (libs), a BSD sockets-derived implementation, tuned for embedded Linux-based devices.
Media Libraries: Library for playing and recording of audio and video files, supporting the most popular codecs.
Surface Manager: Access to the display subsystem is provided by this library, composing 2D and 3D graphic layers.
LibWebCore: This is a modern web browser engine for the Android browser, based on the open source WebKit engine (core implementation of classes, that allow the main browser functionality, as displaying homepages in windows, following links, etc.).
SQLite: The lightweight and powerful relational database, which is available to all applications.

Application Framework [Goo10]

Through this framework, application developers are able to access device hardware, manage background services, access location information, create notifications etc. With some security restrictions, each application can expose its own services, which can be in turn consumed by another application. Through this mechanism, each component can be replaced by another one.
Views: Extensible set of interfaces for the View-Part of the developed ap-

plications.
Content Providers: Interfaces for exposing and consuming data to/from other applications.
Resource Manager: Via this interfaces, non-code resources can be consumed (layout files, graphics).
Notification Manager: Creating custom notifications is available via this interface.
Activity Manager: This interface manages the life-cycle of an application.

Applications [Goo10]

This is the layer of applications, where new ones, in Java developed, are installed. The Android OS is shipped with some standard applications, such as contact list, email client, browser, calendar, SMS program, maps and others.

Features [Goo10]

Among the mentioned layers, if the hardware supports it, there are many other interfaces provided by the OS, such as interfaces for: GSM (Global System for Mobile Communications, the most used standard for mobile telephony), Bluetooth, EDGE (Enhanced Data rates for GSM Evolution, a mobile phone technology with improved data transmission rates), 3G (a family of standards for mobile telecommunications), WLAN, Camera, GPS, Compass, Accelerometer, etc.

2.2.3 Layers of the Symbian OS

The OS, based on UI platforms S60, UIQ and MOAP, is now (since 04.02.2010) also open source and the current Version is the 3rd release of the Symbian Platform (Q1 2010). In the future, all UI platforms should be integrated into Qt.[Jak09]

Kernel [Bab07]

As the central manager of the whole system, this kernel is responsible for managing processes, memory and privileged access to the CPU. Its functionality can be extended with device drivers and DLLs (Dynamic Link Library, shared libraries that makes some functionality available for the caller).

Base Libraries [Bab07]

Placed upon the kernel, base libraries provide interfaces for the underlying layer functionality for the OS itself, its components and installed applications. These includes the standard I/O, manipulation with data types and error handling.

Application Services and Framework [Bab07]

One layer above, the application services referring on base libraries provide the main interface for installed applications and allows the communication with the device. Among others, this functionality can provide access to the common memory, as for example contacts; using the communication possibilities for sending messages (SMS, email); voice etc.
The Framework concentrates on the UI part of the applications by providing the access via interfaces to the UI of the device represented by its display.

Communication Architecture [Bab07]

Communication interfaces cover internal and external communication facilities. External communication takes place with external devices over for example TCP/IP (Internet Protocol Suite, a set of protocols used for communication over internet), Bluetooth, USB (Universal Serial Bus, a bus for connecting a host with a device), etc. On the other hand, email, SMS and communication between the applications is provided by interfaces for internal communication.

Middleware Feature Architecture [Bab07]

Simple, this category covers all not yet mentioned functionality and interfaces, like security, multimedia, animation and others.

2.3 Mobile Applications

Installing new applications on mobile phones faces one big problem from the viewpoint of security - their approval. Beside this problem, also the current trends in application development are discussed here. This section is finished by definition of a simplified mobile operating system model.

2.3.1 Approval of new Installed Applications

Since the installed applications can be seen as plug-able interfaces for user interaction, their installation process and impacts are very important for the security considerations. Each such application has to be installed on the mobile device, therefore the user has to approve the installation. To support the user, the developed applications are certified (as described below) and the user can then better decide, whether he installs the application or not. Without going into further details of the process, how applications built on the mentioned OSs of mobile devices are deployed, published and sold, used business models for application certification will be described very briefly in the following section. Only legal and official ways of distributing applications are described in this part, that means possibilities offered by each OS owner.

iPhone OS

If a developer wants to publish his application for other iPhone users, he has to publish it in the iTunes store. To achieve the publishing managed by Apple, the developer has to submit the binary of the application, which is in turn checked by Apple. Since this process of checking the application by Apple is not public, one could only assume, that some static and dynamic analyzes are made, maybe in some cases also manual human analysis is utilized. Such an approved application is than available in iTunes for download and installation for all registered iTunes users.

Android OS

The application approval process is handled via certificates, which are used for the identification of the author of such an application. Also self signed certificates are allowed, without using a trusted certificate authority. According to this, a new application developed for Android OS is not forced to be checked by any authority and can be made available without restrictions.

Symbian OS

Symbian OS also utilizes certificates for establishing trustworthiness of new developed applications. As by Android OS, also self signed applications can be published. If a developer wants to increase the credibility of his new developed application, he can also try to get a special certificate, as for example ”Symbian Signed”, proclaiming that his application meets the described internal standard.

Application Approval Model

Summing up the above mentioned application approval models, following two possible models are used today: Approval Process as used by iPhone OS and Apple’s iTunes-Store, where the submitted executable of an application is checked and the publication of such application is approved or denied by an authority. Application Certification as used by Android OS or Symbian OS, where the certificates and signing are used to identify the author. If a developer wants to get some (mostly paid) certificates, he has to meet the certificate requirements, which are then checked and the certificate is assigned. Such additional certificates are mainly for attracting the user for the offered application and are not necessary for the distribution of applications.

2.3.2 Current Trends in Application Development

Nowadays, many developers are aware of the mobile device market. There are plenty of websites and communities for each mentioned platform trying

to support the development of new applications.
The survey in [Meta] illustrates the importance of development of SW for the field of mobile devices and underlines the necessity of considering the security aspects of this topic as done in this work. The authors of [Meta] have chosen 108 developers (selected via author's publisher newsletter) and surveyed them with these main findings:

- Publishers try to develop for multiple platforms
- Developers cross-develop on both main platforms Android and iPhone
- Developers are satisfied with their success in mobile platforms

The developers of applications do not concentrate on one single platform, but they concentrate on the most common used while increasing their possible market success. 31% of surveyed developers already developed for more than one platform, 47% of surveyed developers plan to develop for more than one mobile platform and 58% of asked developers already develop mobile websites. These findings underline the necessity of security concepts not based on OS of the mobile platforms, as developers also do not concentrate on them. For example, more than 70% of iPhone developers plan to develop for Android over the next six months. On the other hand, 48% of Android developers plan to develop for iPhone in the next six months.
As the mobile world is interesting for developers and they plan to develop more for this sector and also for multiple platforms, the necessity of having security concepts for the development is already given. As 64% of surveyed developers rate their success of developed applications as "very" or "somewhat" successful. As known from the world of PCs, there is no possibility to add security to a working system (OS and their applications), the security considerations have to be made beforehand. This also underlines the necessity of considerations proposed in this work. For further information on the exact statistics refer to the used source [Meta] or other statistics made on this topic.

2.4 Simplified Mobile Operating System Model

The very coarse description of the three above OSs of mobile devices has shown, that they have the main concepts in common. But there are also many differences in the detailed implementation of each OS.
By examining the security of mobile devices, the central point of this thesis is not the knowledge of location of an interface/library (whether it is located in the bottom or top layer of the OS), but the knowledge that such an interface exists and is accessible to the OS. The security of this functionality and the security model behind this interface should be reviewed and discussed.
As this thesis is focused on the security considerations of mobile device use

cases, abstraction from such ”small” differences is needed. To focus the future discussion on the security in general, without the necessity of pointing out each difference of an OS implementation, the following simplified assumption is made: the differences between the mentioned three OSs of mobile devices are, from this thesis security viewpoint, not crucial for the future discussions and can bee seen as variations. But for concrete inspection of a specific threat, also the architecture needs to be considered. To underline this assumption, a simplified model of an OS of mobile devices, which will focus the following considerations on the mobile applications, is presented in the next lines from [CJ07] and referred as SMOSM (Simplified Mobile Operating System Model) throughout this thesis.
This model will be used in general discussions about security, valid for each of the discussed OSs in Section 2.2. The other motivation is also to combine many functionality into SMOSM, which are not yet implemented in all of the yet developed OS (e.g.: the real multitasking is not yet available in the iPhone OS 3, but in Android and Symbian). As discussed in Section 2.1, also some HW interfaces are not present in each version of a mobile device. For such cases, it will be assumed that the SMOSM provides the interfaces for these functionality because of today presence in some OSs and some mobile devices. This enables us not to distinguish in such cases between hardware and software versions, but concentrate on the model of such implemented interface/functionality.
According to the discussed OSs in Section 2.2, a layered view of such a SMOSM could look like the Figure 2.3 proposed in [CJ07].

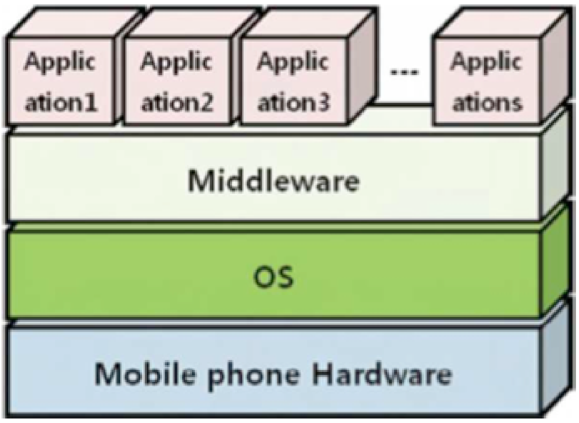

Figure 2.3: Layers of SMOSM from [CJ07]

As can be seen in the model, installed mobile applications are plug-able parts of a mobile device, which can be installed and extend the functionality of the OS by utilizing its interfaces.

Chapter 3

Introduction to IT-Security Areas and Goals

This chapter introduces security and its principles, known also from other devices like personal computers. The many years of development of PCs showed the necessity of considering security from the beginning of development of devices and their infrastructures. Todays problems of the internet originate from some not considered security principles in the past. The main reason is the fast development. Nobody thought of an interconnected world with millions and millions participants when introducing the first connection of PCs.

There is a parallel in this development in the world of mobile devices. When first mobile phones were developed, nobody even thought of the ongoing evolution into a multimedia device, which can replace a PC in many daily situations.

This chapter starts with the discussion about stakeholders of an attack: the attacker and the user of the mobile phone. Some possible attackers are classified with the aim to discuss their objectives and also to clarify the scope of the further discussions about security. The protection goals of security in general are discussed in the last part of this chapter to provide a guideline for systematic threat analysis in next chapters.

3.1 Types of Attackers

Before discussing the protection goals, which each system should provide for it's users, the source of an attack should be considered. It is necessary to be aware of all the stakeholders of an attack. One group of them are the attackers. The most appropriate classification of attackers is based on their motivation and the objective of the attack. Since each attacker is ready to invest resources for an attack directly dependent on his motivation. The following classification of attackers is taken from [Sch00]. Important for this thesis and future discussion of possible security improvements is also the delimitation of treated types of attackers. Therefore each attacker type is considered also from this viewpoint and discussions in next chapters.

3.1.1 Hackers

"I define a hacker as an individual who experiments with the limitations of systems for intellectual curiosity or sheer pleasure; the word describes a person with a particular set of skills and not a particular set of morals." [Sch00]

A hacker enjoys the crossing of borders, getting access to closed systems. Surely there exists bad hackers, but their main intention is not the data gained by the attack, but the ability to break a system. By trying to hack a system, attackers have to gain knowledge of the system, many times is this knowledge much higher than the knowledge of the system designers and originators. Another of their specifics is, that they have relatively enough

time, but are limited in financial resources.
The problem is, that their work is illegal. Breaking into a closed system is illegal. Such a compromised system is no more secure and also if the hacker claims, that he changed nothing in the system, the system is no more secure. Another problem is, that these hackers often provide an automated possibility to break into such systems, enabling other people utilizing these tools.

3.1.2 Lone Criminals

"Lone criminals will target commerce systems because that's where the money is. Their techniques may lack elegance, but they will steal money, and they will cost even more money to catch and prosecute. And there will be a lot of them." [Sch00]
A lone criminal is a small player in the chain of attacks, because he has less money to finance his attack. The difference to the above described hacker is, that lone criminals are interested in stealing money in any possible way and form. This thesis also covers attacks from lone criminals, since their only difference to hackers is, that a lone criminal uses the opportunity of gained access to also make money with it.

3.1.3 Malicious Insiders

"A malicious insider is a dangerous and insidious adversary. He's already inside the system he wants to attack, so he can ignore any perimeter defenses around the system. He probably has a high level of access, and could be considered trusted by the system he is attacking." [Sch00]
The problem by fighting against insiders is, that they deserve trust from the system. The main common security improvements concentrate on external attackers, without the knowledge of internal procedures. It is hard to identify them and also to take action against them, since they reside in the system (because of their assigned role). This group of attackers is not considered in this thesis, since it is nearly impossible for a mobile phone to spot an malicious insider, or to distinguish them from real trusted users.

3.1.4 Industrial Espionage

"The line where investigative techniques stop being legal and start being illegal is where competitive intelligence stops and industrial espionage starts. The line moves from jurisdiction to jurisdiction, but there are gross generalities." [Sch00]
Sometimes it is a problem to distinguish legal and illegal actions in business. Only objective referees can distinguish, but if the decision is worth enough money, the referee can change his mind being subjective. The main objective of such an attack is the advantage in a business competition.

3.1.5 Press

"In industrial countries with reasonable freedoms, the press can bring considerable resources to bear on attacking a particular system or target. They can be well funded; they can hire experts and gain access. And if they believe their motivations are true, they can tolerate risk." [Sch00]
This kind of attacker is very similar to the industrial espionage, the only difference is the objective. Press is prepared to pay for a new story a lot of money to be the first to publish it (keyword for current topic: WikiLeaks). From the viewpoint of this thesis, these two attacker types can be seen as similar.

3.1.6 Organized Crime

"In terms of risk, organized crime is what you get when you combine lone criminals with a lot of money and organization. These guys know that you have to spend money to make money, and are willing to invest in profitable attacks against a financial system." [Sch00]
Organized crime is a global international business. The objectives of organized crime did not change with the emerging of mobile phones, they just got another place of criminality. Since identity theft grows and utilizing mobile phones for electronic theft is more and more valuable, the here discussed devices can become a focused target.

3.1.7 Police

"You can think of the police as kind of like a national intelligence organization, except that they are less well funded, less technically savvy, and focused on crimefighting." [Sch00]
Different to other attackers, police is gathering information for courts. If some police subjects handle against the law, they can be considered as malicious insiders. Police is going to be an attacker, when a police state abuses the possible access of police to secret information and systems.

3.1.8 Terrorists

"This category is a catchall for a broad range of ideological groups and individuals, both domestic and international. There's no attempt to make moral judgments here: One person's terrorist is another person's freedom fighter." [Sch00]
This group of attackers concentrate more on destruction (denial of service), as on gathering information (personal data stored in mobile phones can support their intentions). In the area of IT, breaking into systems like air traffic control, could have terrible impacts. Also their motivation is very high, since they see their actions as a part of a war. Therefore it is also

important to see how helpful would it be for such criminals to gain access to devices like mobile phones.

3.1.9 National Intelligence Organizations

”For most of their adversaries, this is all a game: break into a Web site, gain some competitive intelligence, steal some money, cause a little mayhem. Whatever. For these guys, it’s very real.” [Sch00]
These organizations have enough resources to buy research, hackers, etc. But one of their objectives is to remain secrete, they are not interested in publishing gained information. If such organizations attack a mobile phone to gain access to the ongoing communication, they want to remain secret so the communication participants exchange information without fear to be overheard.

3.1.10 Infowarriors

”An infowarrior is a military adversary who tries to undermine his target’s ability to wage war by attacking the information or network infrastructure. Specific attacks range from subtly modifying systems so that they don’t work (or don’t work correctly) to blowing up the systems completely.” [Sch00]
One main difference to national intelligence organization is the willing to tolerate things, which would be intolerable to them. The second main goal is to lower the ability of the enemy to continue the war. Military had in the past different systems to the commercial products. But today, the development of commercial devices is much more faster than the development of military devices. Therefore, it could be, that military is utilizing the same mobile phones, that are available for the commerce. This is a big problem, since these military devices can be attacked on the same way as the commercial ones.

3.2 The Human Factor for Mobile Phone Security

Attackers are not the only group of stakeholders of an attack. The second important participant in an attack on mobile phones is the user of the device itself. There are many reasons, why users are not successful in blocking attacks, but one of the fundamental problems in the field of user security in general is the unawareness of the involved participants. A good example is the evolution of PCs, where also many users where not aware of many security risks they were exposed to while using their PCs connected to the internet. There are still many users unaware of these risks, but the society has achieved a security standard, where OSs of PCs are shipped with firewalls and some antivirus applications.
An interesting survey regarding user attitudes was made in 2005 [CF05].

One of the results was, that the criteria of security came last for provider selection by mobile phone users. But if they were asked to rank criteria when selecting a model of a mobile phone, security came second after battery life. This could be interpreted, that mobile phone users think, that the security depends on the device and not on the provider.
"Sixty-six percent of respondents reported to use PIN (Personal Identification Number, a secret number shared between a user and a system to gain access to secured content) authentication at switch on, with 18% of users also utilizing the secondary standby mode authentication." [CF05] This means, that nearly 30% of respondents do not use the PIN protection. And additionally, 45% of asked users never changed their PIN, 42% of respondents changed the initial PIN. Only 13% of them have changed their PIN number more than once. Another problem is, that 36% of respondents use as their PIN a number used also in other their services. Therefore the knowledge of such a number would allow the attacker to compromise also other services of the attacked user.
According to the survey [Sea09] from December 2008 in Hong Kong, 48% of respondents think, that the cell network provider is responsible for the security of their mobile devices. 25% of respondents claimed, that the manufacturers of their mobile devices should ensure its security. And 26% of respondents thought, that the user of the mobile phone is responsible for its security. Another interesting topic is the connection to the internet and the related security threats possibilities. 70% of the respondents have their mobile phones connected to the internet, but 66% of respondents do not have any protection software installed on their devices. And 17% of respondents did not have any opinion to the security of their devices. The malware awareness is relatively high, because 86% of the asked users think, that their devices could be infected through Bluetooth. 90% of respondents declared, that a possible malware could gain access to their contacts and send text messages without their notice (and they would have to pay these extra costs). Interesting is, that the respondents were aware of malware, but more than a half of them had no security software installed.
In the survey [Ent09] from December 2008, 66% of worldwide respondents did not have installed any security software, compared with 86% in 2007.
According to the survey [Sta09] from 2009, 89% of asked professionals think that the security level of mobile phones is low. Up to 46% of the asked companies did not have any rules and security restrictions concerning employees mobile phones. Only 25% of asked respondents declared, that the possibility to get illegal access to the company's private network will remain low in 2011 and 40% think, this risk will be high or very high.
Another survey [Hig10] concerning the company's employees and their mobile phones show results, that something about 10% of surveyed companies are running some sort of anti-malware software, but 54% plan to introduce them in the following year (i.e. year 2011). According to this U.K. based

survey, the traffic produced by malware and spam via mail, MMS (Multimedia Messaging Service, improvement of SMS to be able to send also multimedia content to and from a mobile phone) and SMS has risen from 2% up to 20-30%, from which 14-22% was considered malicious. Also important is, that 40% of asked companies want to recruit staff for their security considerations.
Without the knowledge of the dependencies of security awareness of PCs and here discussed mobile phones, the main reason why PC users are aware of security and do not recognize such risks while using their mobile phones, could be, that they do not recognize them as very similar to PC risks from the viewpoint of their OSs (as discussed in Section 2.2). Nowadays, many mobile devices are permanently connected to the internet and therefore many of the well known risks apply also to them. As discussed in Section 3.3, these protection goals have to be implemented into the OSs of mobile devices too.

3.3 Protection Goals

Protection goals evolved through years. New development of functionality and possible use cases made new protection goals necessary. Based on [BA10], the for this thesis relevant protection goals are summarized in the following section and are conducted with possible examples from the world of mobile devices.

3.3.1 Confidentiality

Confidentiality of information is the goal to protect information from unauthorized subjects. In other words, some specific information is accessible only to certain subjects. This covers also the monitoring of access to information parts while transported (access rights), no matter whether the information is stored or only transported. Also additional information about the sending and receiving procedure should remain secret. The behavior of communicating subjects should be secured too, because a trace of their activity could reveal parts of confidential information. One of the main methods to ensure confidentiality are the mentioned access rights and encryption. Encrypted data stays secret for other participants unless they have the key to decrypt the information (either symmetric or asymmetric encryption can be used). This should apply to each communication interface of mobile devices, also for the simple voice transmission between two participants. Detailed discussions about confidentiality issues in mobile phone communication protocols like GSM can be found in [LHY99]. Confidentiality can be divided into following subgoals.

Unlinkability

While using some internet services for example with a mobile device, some servers and services are accessed. The goal of unlinkability is to make it impossible for a third party (e.g. the attacker) to combine multiple single requests of one source together. One possible counteraction to achieve this protection goal are mixes (keyword: Schaum Mix). Some server/communication element mixes the access flow from the mobile device and therefore no linkability of this device is possible unless the mix method is known. Authors of [JLP10] developed for example a novel approach for temporal unlinkability.

Untraceability

The goal of untraceability is somehow similar to the above mentioned unlinkability. But in contrast to linkability, where more actions can be linked together, traceability is possible, when actions can be traced back to a concrete subject. An example would be the ability to trace activities such as a buying behavior of an unknown participant to a concrete person's mobile device. This problem is discussed for example in [SSG99].

Anonymity

Anonymity is given, when the identity of the user can not be revealed and it is the result of the above mentioned unlinkability. This anonymity must be irreversible in contradiction to the below mentioned pseudonymity, that means there is no way to reveal the anonymity backwards to the concrete person. For example, monitoring of behavior of mobile device user for statistical considerations is not a security threat, if there is really a mechanism which guarantees the anonymity of such mobile device user. This problem is discussed for example in [CKK^{+}08], where also a possible implementation is proposed.

Pseudonymity

In contradiction to the above mentioned anonymity, pseudonymity is the creation of new identity and assigning it to a concrete person. The new identity represents a pseudonymity for the concrete person, because the only way to assign the new identity (for example a numerical unique identifier) to the concrete person is to know the assign-rule. This would be for example (in contradiction to the above mentioned example) an assign-rule, where to each monitored person an unique identifier is assigned (for example to pseudonymize the statistical analysis). And in case some suspicious and illegal actions of the monitored person are discovered, this assign rule can be used to reveal the pseudonymized identity. Such an approach while using mobile phones was proposed for example in [CDG05] or [DCG08].

Unobservability

When the protection gaol of unobservability is not guaranteed, the subject sending/receiving data can be identified. Utilization of services or sending/receiving data can then be observed. One such example would be the identification of some person using his mobile device and observing him while using it. The problem for example in packet and data oriented networks (like IP) is, that it is always observable, that a subject is sending/receiving data and therefore the identification of this subject is possible. As mentioned before, mixes can be utilized. Also some dummy data can be produced by the subject and sent somewhere to make the identification harder. Another problem of observability is the environment, where the subject with it's mobile device are located. Some physical observation methods could be used, like camera, voice recorder, another spying person, up to some radiation measurement of the electronic device. This part of observability isn't easy to minimize by the design of mobile devices. Maybe some sort of radiation could be checked and minimized, but the user himself is responsible for the control of physical spying in the user's (physical) environment. Maybe the awareness would force the user to protect himself.

Covertness, Obscurity

In some use cases the security problem is also the knowledge, that a communication even occurs. Steganography is one of such examples achieving covertness, when the secret message is covered into the "unnecessary" data of images or sound files. The security goal of covertness is in general very hard to achieve and the prerequisite is the knowledge about the used transport method/protocol. An example of this goal would be the possibility to cover even the information, that two participants are calling each other via their mobile phones.

3.3.2 Transparency

Transparency can be seen as a contradiction to the above mentioned confidentiality. Used systems and algorithms have to be clear and understandable. Clear information about the processed persons, persons private data and its actions have to be available. A possible way to achieve transparency in mobile devices is such a description of the whole OS (and also its additional applications) of this mobile device, where processed information are clearly defined. Another part of transparency is the utilization transparency, where the usage of a mobile device can be logged (with usage of digital signatures and time stamps), which can also be achieved in a mobile device with an system wide logfile respectively a log database, for example.

Accountability

This protection goal should ensure, that the source of information can be identified, for instance the sender of an email sent via a mobile device. Accountability also means the detectability and the non-repudiation, which are interconnected meanings of this protection goal. These problems occur mostly in e-commerce environments, where the source of the information has to be verified (bank account) and the detectability of realized payment and non-repudiation of an made order are central for the business. With the evolution of internet and remote access to information, the goal of accountability grows and covers today a relatively new aspect of responsibility. The accountability has to be ensured mainly for the reason, that an action (access to information via mobile device) and the non-repudiation of this action has to be secured. For example, authors of [WRP08] discuss the signatures for non-repudiation for service payment.

Authenticity

Authenticity is interconnected with the above mentioned accountability and represents mechanisms, that ensure the identity of a person. On an example of two persons A and B communicating via their mobile devices, the authenticity ensures the identity of A and B and also ensures, that A, B remain A, B for the whole communication session, i.e. that nobody (any person C) could act in some parts of the ongoing communication session as A or B without the notice of them. To ensure authenticity, some private person information (known as credentials) are used, like: passwords, biometric data like iris or fingerprint, presence of chip cards and other authentication devices. The well known example from the beginning of mobile phones and devices is the knowledge of PIN, which ensures a (relatively small) level of authenticity, because theoretically, only the owner of the mobile device should know this PIN (according to "Personal Identifier"), without considering the weakness of such a mechanism. Another authenticity mechanism of mobile phones is the presence of SIM. Evaluation of improvements and implementing a "Software SIM" are discussed for example in [MM08].

Reviewability

One could see the reviewability as a combination of accountability and authenticity. But the reviewability itself as a protection goal should ensure, that each information flow, access to a system and an action in the system itself has to be reviewable, i.e. each information flow should be verifiable/checkable. In other words, this goal should ensure that the flow of the crucial information is reconstructible and a possible misuse is verifiable. In case of information misuse, it should be possible to review, who entered, read, modified or entirely removed the crucial information. Accountability is therefore more concentrated on the identity, but the reviewability is concentrated on the past actions and their documentation. Such an approach was presented in [PS10], where the authors propose a framework for auditable records of transactions.

3.3.3 Availability

Availability covers the possibility to access a service. Mostly measured in percent, 100% availability of a mobile device means, that every time needed, it responds as intended to the user. Simple, this availability would not be guaranteed, if the device would "froze" (not responding or responding not correctly) and would need a restart of the system. The time from the passed availability to the successful reboot and beginning availability is called downtime. In complex systems, like the auction service eBay, redundant systems (more than only one physical server) try to minimize each leverage of the 100% level of availability. Colin Mulliner did some research on SMS fuzzing, where he tried to inject a SMS into the mobile phone to attack it"s availability, as described for example in [MM09].

3.3.4 Integrity

The security goal integrity covers two parts: the integrity and correctness of data (data integrity); the correctness of the system itself, of it's functionality and intended responses (system integrity). Integrity therefore deals with the whole information flow. The data integrity is ensured, when the production of data from the system is correct (no mistakes), when the data was not unintended modified by another subject and presented correctly as a response without modification. Such integrity is often ensured by hash values, which are results of mathematical functions computed with the data. Each modification of this data would then result into a different hash value computed with the modified data. In some systems, also time is a sort of integrity, where the time sequences of request/response messages matters and the order of such messages is crucial. While placing calls with mobile phones in foreign cellular networks, integrity issues arise (roaming), as discussed for example in [CKK$^+$08]. This goal can be divided into following subgoals.

Dependability, Reliability

Dependability is achieved, when the system enters no unintended or undefined state while executing the necessary commands. Reliable are the system functions, when they act exactly as specified. In one simple sentence: Reliability and Dependability ensure, that the target state and the current state of the system are always equal. Example of such problem in a mobile device could be an error in the implementation of cryptographic signing of outgoing e-mails, where the computed hash value of a message would be wrong and such message would be thrown away at the recipient because of the (not happened) modification of the content. Authors of [BC04] composed a framework for identifying dependability problems when developing software for mobile phones.

Controllability

Controllability concentrates on the elimination of possible risks connected with operation of the user. A possible example while using mobile devices would be an unchecked input (number: range, amount; string: length, used characters) to a service request, which could lead to dysfunction of the mobile device itself or the whole service (available for more mobile device users). Such a possibility to cause miss functions and to break the integrity because of not controlled user behavior is against the mentioned controllability as one of the protection goals.

Non-Propagation

Non-propagation of data has to be ensured, because if the temporarily assigned access rules can be copied and used more than once, then the integrity of such a transaction is not ensured. These attacks are known as replay-attacks and are often used in finance transactions. One example would be a session, where a person buys tickets for a performance via his mobile device. Then a delinquent could copy these session credentials and buy some tickets for himself.

Chapter 4

Mobile Phone Use Cases Examples

In this chapter, some today and future use cases utilizing mobile phones and their provided technologies and possibilities are described. Providing possible scenarios, the motivation for security considerations for next chapters should be developed. This chapter binds the discussion of composition and functionality of mobile phones with the before defined protection goals and provides motivation for security considerations of the possible attacks in this use case scenarios. Following chapters refer to these use cases for further security discussions.

4.1 Communication via Messages

By introducing laptops in daily lives usage, the possibility to check, manage and answer e-mail more instantly and also on trips was a huge step for people to be more reachable and not addicted to the workplace. Further, using SMS on mobile phones is common used for its intended purpose - namely short messaging. For detailed studies refer for example to [HC05], teenagers are the focus in [GPE06] and middle-aged users in [SRS06]. New possible scenarios of SMS are discussed for example in [HUJ05].
The development of mobile phones makes it possible to check, manage and answer e-mails directly on the mobile phone. Often, there is no difference in reachability between receiving a SMS or e-mail.

4.2 Storing Personal Data

After introduction of mobile phones into daily life, the necessity to store phone numbers of communication partners into mobile phone's contact list emerged. During the development of these devices, the contact information was more and more precise. Today, a mobile phone user can store many other types of personal data beside the phone number of a person in the contact list: photo, birthday, postal address, private and business contact numbers, various mail addresses, additional web page links, custom notes and much more.
But not only the contact list contains personal data. Also daily used information, like the personal calendar, is often used. Storing past, current and future daily life schedules, possibly interconnected with persons in the contact list, making personal notes attached to these appointments. All these new possibilities allows a mobile phone user to store personal data in the device. Data which is stored, accessed and modified sometimes even more often than it would be on other devices, like for example on a laptop or desktop PC.
Authors of for example [MHM+10] discuss personal data on mobile phones and propose a framework for securing them.

4.3 Location Based Information

Location based services in mobile phones have their origin in the establishment of a cellular network. For a successful connection between two communicating subjects over the cellular network, the operator of such a cellular network has to know where these two participants are located to establish a connection over cell towers. Therefore, each cell phone has to be connected with at least one cell tower to make/receive calls. This was in the past the most granular source of location information about such a mobile phone, because distances between such towers are sometimes several kilometers.

With the development of mobile devices, other, more precise location information, were made available. GPS modules, included in the most todays mobile devices, provide a relatively precise location information. Another today utilized location information are the WLAN access points, since WLAN modules are also more often included into new mobile devices. The combination of the above mentioned location services can provide a relatively precise location information.

To specify the location and its environment even more detailed, other sensors, like for example the acoustic sensor (at least microphone is present in each device), of a mobile phone can be used. By analyzing the sound in the surrounding area, some specific properties of such environment can be identified. The authors of [SXD09] conclude, that in their proposed method by using only the analyses of the audio (without assistive information, as accelerometer, GPS, etc.), a 91% overall recognition rate can be achieved. By only analyzing the sound of the environment, authors can distinguish, if the device is in a classroom, office, vehicle, etc. But such statistical arithmetic-operations are resource intensive and therefore some times not possible in the mobile devices directly. For this reason, some applications send collected information to a server, which evaluates the collected information and returns only results to the mobile phone. Such an application is for example SoundSense and is described in [LPL+09]. Without the disruption of the user, the collected information is sent to the server, evaluated and sent back, without any notice of the mobile phone owner. Also personalized profiles for each user are available and the application is capable of learning user habits in the field of acoustic events. A possible use case in the daily life could look like proposed in [MLF+08], where the authors developed an application, which infers the current social environment similar as described above and is connected with the social network, as for example Facebook, and automatically updates the social status of the user.

Having all these possible context information about the position and environment of such a mobile device, location-based services gain a high interest by developers. The authors of [IMI10] summarize the existing literature in the field of location-dependent query processing. Without going into further details, these services are based on location information about the mobile device (gathered by the before mentioned sensors). The more precise the information about the location is, the more precise the location-based service will be.

Going even further, the authors of [CMP09] combined information of phone calls, sent messages and environment information to develop a platform called SocialCircuits. According to their work, this platform is capable of examine the social circuit of a person: face-to-face and phone-based communication. By utilizing the sensors mentioned above, this platform is capable of examining the face-to-face communication with a participant and draw some conclusions, based on these collected information, regarding the social circuit of a mobile phone user.

In contrast to PCs, many mobile phones today are equipped with a positioning HW. But these devices contain also other sensors, like for example the microphone, accelerometer or light sensors. All of these sensors connected with the positioning HW allow the positioning services to be more precise. And when the position and motion of such a device is known, tracking or even guessing the current moving activity is possible. Such an approach, by combining the accelerometer and the positioning HW, was used by the authors of [RMB$^+$10] to determine the transportation mode. By utilizing this knowledge, an application could improve collective values like the nature. Such an example was proposed in [FDK$^+$09], where the authors developed an application for sensing and revealing the transportation behavior.

4.4 Assisting People with Special Needs

Today, many devices concentrate on assisting people with special needs in their daily life trying to integrate them much more in the society, as it was possible in the past. A mobile device, with its growing technical possibilities, can emerge to the missing platform for disabled people. Voice, camera, positioning and ambience sensors can assist people with special needs.

Authors of [VBSP08] discuss some possible services for patients (from the point of view of an hospital, such people are patients) and hospital, like for example: monitoring patient's condition, location-based monitoring, remote surveillance, alarm-triggered actions, help on demand, direct communication. All these services are possible though and can be combined in one device - the mobile phone.

Combining all sensors, also for example a camera capable of video recording, could be very useful for example for guiding blind people through their daily life procedure. A mobile device capable of indoor and outdoor navigation could guide the person with usage of GPS/A-GPS, cellular network, connection to internet and WLAN hot spots in his whole journey by utilizing voice navigation. The lookup of destination, retrieving additional information, as-

sisting during the whole journey and reacting on unforeseen circumstances could be provided from such mobile devices in a blind friendly way to the user.
The survey from [KGE11] summarizes the development of electronic mobile guides and could be used for adoption for blind people to develop a suitable solution for their disability.

4.5 Automated Alarming

In many daily life situations - not depending on a possible disability of the mobile device user - an assisted technology can provide the needed help. Such a case could be a car accident, as discussed for instance in [BIM+09]. For example, a person connects his mobile phone with the car system. The car system recognizes a possible accident automatically and undertakes the necessary communication with first aid service, police, assurance and other necessary services.

4.6 Network Communication Services

A common use case of laptops is to connect them to the company's intranet and work with a remote data or work via a remote PC. For working on remote PC from the laptop, a network connection to that remote point and a client for working with the remote machine are necessary. Nowadays, a VPN (Virtual Private Network, keeping private a conversation between two participants over a public network like internet) connection is common for this purpose. A mobile phone can be also used for this scenario. Having an internet connection and a suitable client for the remote machine running on the mobile phone, the communication is possible. The anytime access from anywhere in business environments is discussed for example in [POS+01].
According to [EKC08] the usage of a Peer to Peer File Transfer Protocol is another example.

Chapter 5

Threat Analysis for Mobile Phones

Based on the protection goals from Section 3.3, security of mobile phone use cases is systematically inspected. First a classification of possible attack surface is given. Afterwards some of the presented use cases of Chapter 4 will be analyzed and classified. The analysis contains already existing threats as also possible future problems against the presented security goals in scenarios using mobile phones. But not only possible scenarios, even concrete, by researchers documented, examples of threats are discussed.
This classification is based on the simplified description of mobile phones in Chapter 2. The relevant layers of such systems are used to define the classes for further security analysis in this thesis. These are the four identified groups of threats discussed in the following chapter:

- Communication Interfaces
- OS Vulnerabilities
- Installed Applications
- Mobility

5.1 Defining the Mobile Phone Attack Surface

Analogue to [AAB10], where the authors utilize implied scenarios for security testing, this classification is based on in Chapter 4 discussed possible use case scenarios of mobile usage. By analyzing possible threats based on use case scenarios, not a concrete part of a mobile phone is examined, but the whole use case scenario. Focusing the threat analysis on scenarios provides an overall view on possible attacks, allowing combination of more security threats of more components to one security attack. Such analysis is more complete (not bind to a component) and has the focus of this master thesis - analyzing use cases of mobile phones.
To perform a security threat analysis for mobile phone use case scenarios, the possible attack surface has to be considered. Based on the discussions in Chapter 2 and the simplified architecture of the mobile device, this attack surface is proposed, focused on the discussed examples and is not a complete categorization of all possible threats in mobile device scenarios. It is proposed for the scope of this thesis and could be extended in future work. Like in the example of threat analysis of storage systems in [HMLY05], this classification could be also extended to a domain specific classification of mobile phone threats.
Authors from [Bla08] examined the security of electricity grid, which is from the HW point of view very complicated device, and they identified three layers for security discussions: semantic (involving people), logical (computer and networks) and physical (including interactions of all layers). Analogously to this classification based on the layers identified by Neumann in

[NB99], this thesis's classification has a logical layer divided into OS Vulnerabilities and Installed Applications, Communication Interfaces as the physical layer and the interaction with users grouped as Mobility. For more detailed description what is covered in each class refer to the following sections.

Based on the described approach in [HL02], the following threat analysis and the approaches for improvements in Chapter 6 were identified and composed like following:

1. Make a literature research of existing threats
2. Rank the threats by their decreasing priority
3. Make a literature research for possible responding to these threats
4. Propose a classification based on other similar approaches
5. Provide a threat analysis based on before discussed protection goals and grouped with the proposed classification
6. Provide possible improvements also based on the protection goals and on the classification

In next sections, for each mobile surface attack group, some examples are presented to explain the chosen classes.

5.2 Communication Interfaces

Communication interfaces (on different network layers) refers here for a generic term for communication protocols like IP, UMTS, GPS; but also for the wireless kind of communication transfer like WLAN or Bluetooth. These interfaces are used by built in HW interfaces as mentioned in Section 2.1. To handle attacks in this class, the communication interfaces should be improved. The following section describes in detail, which communication interface is covered in this class and on which considerations was this class defined.

These threats originate from the used communication interface and its weaknesses. The most used ones for mobile phones are for example: GSM, GPRS, EDGE, etc. As these protocols often have some vulnerabilities and some threats will be discovered also in the future, there is no possibility to solve these issues in the mobile device itself. But the design and implementation of the OS should see this communication interfaces as possible source of vulnerabilities and allow in the future some possible (temporary) patches introduced also into the OS-Layer. But most of these leaks should be patched in the communication environment of the mobile device and not on the device itself. Clearly a patch of a communication protocol like GSM is not so easy

to roll out and needs some time to even distribute it, therefore also patch possibilities in the mobile phone itself should be enabled.
Another complex group of communication interfaces also covered in this group are IP-based communication protocols. Their vulnerabilities should be positioned as a subclass of the above described interface class, but, as known from the internet, this class covers too many threats to be deeply discussed in this thesis. Since the connection via the IP is possible in the mobile phones, all threats known from the world of PCs connected to the internet via IP should be placed here, like the pitfalls of e-mail communication, browsing the web and utilizing any other IP based services.
Sometimes the used interfaces also predefine the workflow, like for example the SMS. As each device, also mobile phones evolved since their establishment. So were the first mobile phones not designed for the use case to be always connected to the internet. This subclass covers functions, methods, possibilities (described in general as workflows) designed and implemented in the past without the possibilities to consider todays circumstances. So is for example in today's mobile phones and the era of malware, viruses and spam not possible to reject or somehow not receive a SMS or MMS.
Nowadays the current (and new) mobile phones compete on the market in the number of contained communication interfaces such as WLAN, Bluetooth and other (mostly wireless) communication possibilities. These are additional communication interfaces and whole communication technologies with their possible security risks. Each designer and user has to consider such interfaces as possible risks to the mobile device and its OS. As the devices are mobile, they are more and more equipped with additional interfaces and communication possibilities. Therefore the design of the mobile device (incl. OS) and its interfaces has to consider them as possible security risks in the future. For example results the elimination of the infrared communication possibility from the mobile phone in elimination of whole nowadays discovered and future possible security leaks of such communication interface.
The fact, that the communication of a mobile phone has to be mobile, the main part of communication is wireless. The main sort of attacks against this possible security leak is, that an attacker does not have to gain physical access to the network as in case of wired communication, but needs only to somehow sniff, get access, or even manipulate the transmitted data "in the air".

5.2.1 VPN

Two persons are communicating via usual phone call. The first one has a mobile phone (further called "device") which is capable to connect to a PC (further called "PC") over internet. For these purposes, the device has an application (a kind of client for remote connections, further called "client app") for remote connection to the PC. The client app is capable to communicate with the PC over VPN.

Implementation of VPN in the device: security leaks concerning this part are beyond the scope of this thesis and can be classified as Communication Interfaces (see Section 5.2). It is necessary, that the VPN protocol is implemented securely and its violation can not result in possible attacks against the device and its OS.

Credentials and transferred data from and to the client app: this part of the transport path is crucial for the security of the transferred data on the device itself. There should be no possibility for the device OS itself and for no parts of it (other applications) to get access to the content of the transferred data. It could be secured by private memory (with really no access to other applications) of the client app and a kind of encryption, where the data transferred from the client app to the system for further transfer, could not be decrypted. Such an implementation would eliminate the risk of data interception in the device itself by another (maybe harmful) application. These problems can be classified as a combination of OS Vulnerabilities (see Section 5.3) with the Installed Applications (see Section 5.4), since the OS itself is responsible for some part, as well as the client app is responsible for other part of the process.

Wireless data: since the data is transferred wireless, any violations in the transfer method could reveal the transferred data and therefore are a possible risk in this scenario, which is also beyond the scope of this work and can be classified as wireless Communication Interfaces (see Section 5.2). All the communication path from the device to the PC and back is beyond the scope of this thesis.

An attacker could also try to compromise the device by replacing the targeted PC and transfer harmful data to the device. As there is no other possibility to secure this threat from the viewpoint of the device as to secure the client app of the device, this threat can be classified as Installed Applications (see Section 5.4). Here an intuitively possible secure mechanism could be a kind of session management to block communication from other than the intended PC, but this is also beyond the scope of this thesis's security considerations.

Recapitulatory can be concluded, that in this scenario the most crucial part according to the possible deployment of security mechanisms is the transferred data in the device intern between the OS of the device and the applications. Because the deployment of other security mechanisms to other parts of this concrete communication path are not effective from the viewpoint of the mobile device. The Mobility class (see Section 5.5) of threats has also impact on this scenario, since the communicating person is located in the public transport vehicle, but this is not the focus of this discussion.

Beside the already mentioned literature, following sources discuss the utilization of VPN (and other forms of private networks and tunnels) in use case scenarios of mobile phones: [VBSP08], [MLK+09], [Lan10].

5.2.2 Messages

This is a simple example of polling a remote server (mailbox) for new messages and in case there is one, its content is downloaded to the mobile phone. The parts of this communication path can be identified as follows. The mobile phone (further called "device") connected with the remote server containing new emails (further called "mailbox"). For simplicity we assume, that the functionality of sending/receiving e-mails is covered in the mail client application (further called "mail client") installed on the device.
The communication of the device with the mailbox is realized via wireless communication and it's connected risks can be too classified as wireless Communication Interfaces (see Section 5.2). Also the possible security threats of the used communication protocol (for example IMAP, Internet Message Access Protocol, a standard internet protocol for mail retrieval) are beyond the scope and can be classified as Communication Interfaces (see Section 5.2).
There are two main sources of possible threats in this scenario: The first one is the transmission of the credentials necessary for the mailbox access from the mail client to the communication interface of the device, since it is using the same communication interface, where also the received new mail is transmitted. All possible risks according to revealing the credentials/mail have to be secured here and the access to content of this data should be blocked for the device itself and its OS inclusive other devices. This threat can also be categorized as Installed Applications (see Section 5.4). Second main source of security risk is the transfer of the data from the device to the mailbox and back and these aspects are also beyond the scope of the thesis and can be considered as wireless Communication Interfaces (see Section 5.2).
Also following sources in the literature discuss sending messages in various forms when utilizing mobile phones: [SRS06], [IHS07], [HKL+07], [Too08], [MM09].

5.2.3 BitTorrent

Downloading applications (games are also applications) via various protocols like BitTorrent can result in security threats classified as Communication Interfaces (see Section 5.2). Downloading, installing and executing executable applications from the (unsecured) web has many problematic parts and can in this scenario be classified as Installed Applications (see Section 5.4). The main problem of this use case is downloading the executable content from insecure sources. If the application is known and certified, that it does not contain any harmful code parts, the downloaded version for example via the

BitTorrent protocol can contain additional (modified or inserted) harmful code.
Utilization of BitTorrent and other Peer-To-Peer networks in mobile phone use case scenarios is discussed for example in: [EKC08], [KN09], [OHKY09], [ZKJ+10].

5.2.4 Transportation

In this scenario, where collected personal data (transportation habits and positioning) are transferred for computation to a remote server without anonymisation (pseudonymisation would be still possible, see Section 3.3.1 for explanation of Anonymity and Pseudonymity), the problem of remote evaluation is revealed. In such cases the computation is too resource-intensive, or as in this case, data from many devices has to be collected and then computed (this computation could also be resource intensive). Connected threats of this use case could be classified as Communication Interfaces (see Section 5.2), since the data is sent via communication interfaces. The main security goal should be securing this data transferred from the device to the remote server and backwards. Another part of this scenario, which could be attacked, is the anonymisation of such data, which happens on the device. Caused by some OS vulnerabilities (see Section 5.3), the anonymisation could be backtracked.
Transportation is discussed on mobile phone use case scenarios for example in: [FDK+09], [RMB+10], [CVV+10], [RSD+10].

5.2.5 Cellular Botnets

The authors of [TLO+09] describe a Denial-of-Service attack against the core of the cellular network. These attacks are "using selected service request types on the Home Location Register (HLR), the central repository of user location and profile information in the network, by a botnet composed entirely of mobile phones." [TLO+09] Such a successful attack against HLR destroy the whole service in the attacked part of network, because a Denial-of-Service of HLR "would make all users assigned to this device immediately unreachable." [TLO+09] This explains also the Figure 5.1, where the successful attack on the central HLR causes a Denial-of-Service for large geographic network regions. Such an attack can be classified as Installed Applications (see Section 5.4).

5.2.6 Platform and Network Obscurity

Other security challenges are the obscurity in mobile platforms and the mobile network. Many mobile platforms are based on Linux/Unix, like for example the mentioned Android OS and iPhone OS, but have many different security considerations. As the source code of iPhone OS is not available

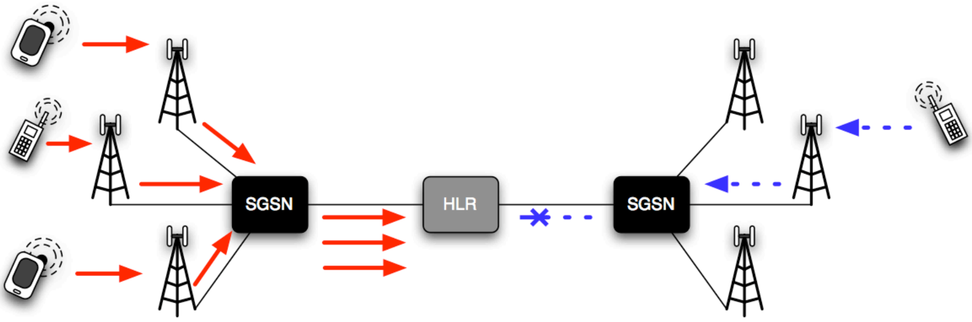

Figure 5.1: Sketch of a possible botnet attack on the cellular network from [TLO+09]

for researchers, it is harder to find the security problems (in contrast to for example the available source code of Android OS). "In addition, platforms are often intentionally restricted from modification and instrumentation due to mobile carrier agreements and regulatory requirements." [OJ10] But some enterprise scenarios need endpoint management: the mobile device has to be identified and managed, which is also a problem because of the platform obscurity. The design of the mobile network also does not obey obscurity and has also impacts on the security considerations. Implementing and deploying security improvements and defense in such obscure environment of unclear mobile operations is also not ideal.

5.3 Operating System Vulnerabilities

The mobile phones, as introduced in the Section 2.2, have their own OSs. Such systems grow by each new system update and version and are becoming more and more complex. A parallel can be seen in the world of PCs, where the operating systems gained their complexity connected with the new hardware resources possibilities. This class is defined for grouping attacks utilizing OS vulnerabilities.

In the era where mobile phones where not connected to the internet, the rollout of released OS patches was not so easy. The user had to physically hand out the device to the manufacturer, who in turn could load the new patch (or in some other way connect the mobile device with for example a PC via cable and load the patch). Today, these patches can easily be deployed also without a physical connection with a PC via the data connection of the mobile phone itself (or via utilizing Bluetooth).

But this new improvement of the patches rollout does not lower the security risk of OS vulnerabilities. These operating systems of mobile devices, as

any other OSs does, contains built in vulnerabilities and possible errors in the development. Therefore future rollout of patches is made necessary. So the first aim should be to develop secure systems without built-in possible future attack surfaces, and not to concentrate on roll outs, which are clearly necessary too. The level of security of the deployed OSs is also connected with their usability and with all protection goals discussed in Section 3.3.
Another subclass also covered here are the additional HW interfaces and first of all their drivers and libraries available for the OS. Since the OS of the mobile phone is the SW component, which utilizes these interfaces and if these interfaces are vulnerable, it is easier to create an attack through the OS. Therefore also attacks on the HW interfaces should be placed in this class.

5.3.1 Location

The analysis of social circuit of a mobile device user with the combination of all environmental sensors (like GPS, compass, microphone, accelerometer, light sensor, etc.) should have a high if not the highest private security. Since confidential and very personally related data combined with the information about persons communication (voice calls, mail, SMS, MMS) can really reveal details about a person much more than it can be inferred from the PCs. The misuse of the integrated sensors can be classified as OS Vulnerabilities (see Section 5.3), because the OS is responsible for securing the access to them. But also installed applications, with access to the sensors, could collect the personal location information and misuse them. Such attacks can be classified as Installed Applications (see Section 5.4), because in this case they contain harmful, not intended functionality.
Location discussions based on mobile phone use case scenarios and the GPS capability can be found for example in: [ZLTG07], [WAB09], [LPL$^+$09], [IMI10], [DCG08], [JLP10].

5.3.2 Medical Assistance

The use case of medical assistance includes highly personal data - information about health diseases and status. But an attack on a device, which communicates also with other medical devices, could have perilous impacts. Denial-of-Service in cases where medical assistance is controlled via the mobile phone, could cause even death. For such examples, mobile devices should be secured on a high level. Attacks in these use cases cover two classification groups. First of all the OS Vulnerabilities (see Section 5.3) have to be secured to provide a secure OS. Then, the additional installed applications have to be checked for security leaks. The class of Installed Applications (see Section 5.4) threats have to be secured, because these applications indirectly communicate with additional medical devices and can also cause harm even if the OS itself is secure enough. [VBSP08] For another example of medical assistance with mobile phones, refer for example to [QLG10].

5.3.3 Rootkits

Kernel-level rootkits, or shortly only rootkits, are well known from the world of PCs. "Rootkits are malware that achieve their malicious goals by infecting the operating system." [BOB^{+}10] This special software can be classified as OS Vulnerabilities (see Section 5.3) and is not easy to detect because of the infection of the OS itself. Detection has to be therefore done somewhere outside the OS - on a separate hardware (a second processor reserved for this purpose, or on a special virtual machine inside the OS). But such techniques where developed for PCs and do not deal with the special environment of mobile phones, where energy should be used wisely and can not be wasted for resource intensive tasks like for example periodic scans of kernel memory snapshots. Authors of [BOB^{+}10] developed three examples of rootkits for mobile phones, which will be described in turn to show the possibilities of such attacks.
Spying on conversation via GSM: after infecting the device, a remote attacker has the necessary functionality to remotely listen or record conversation made on the infected phone (for example by issuing a three-way call with the attacker when the victim is calling someone). A possible scenario by compromising the access to the GSM radio interface of the mobile phone is shown in Figure 5.2. An event (here a notification of calendar event) is intercepted by the rootkit. A malicious phone call via GSM is established and the microphone is turned on. Without the notice of the user, the mobile device acts now as a spying element for each conversation reached by the turned on microphone. Such attack has impacts on the privacy of the mobile device user and could be misused in some crucial meetings, where the participants mostly have their mobile devices turned on while in the meeting. [BOB^{+}10]
Compromising Location Privacy using GPS: in this attack, the rootkit achieves to get the current GPS location of the device and sends it via SMS to the attacker. As each built in interface (as the GPS) of the mobile device has its own buffer for storing necessary information to be read by the system and other applications, the rootkit can easily read this data before the intended application does and then pack them into a SMS and send it to the attacker - without any notice of the mobile device user. Because the rootkit is working in kernel mode, such an attack is also possible when the GPS device is turned off, because a rootkit can temporarily enable and then disable this GPS for its purposes. [BOB^{+}10]
Denial of Service via Battery Exhaustion: since the rootkit runs in kernel mode, it can enable and disable other built in devices like GPS, Bluetooth

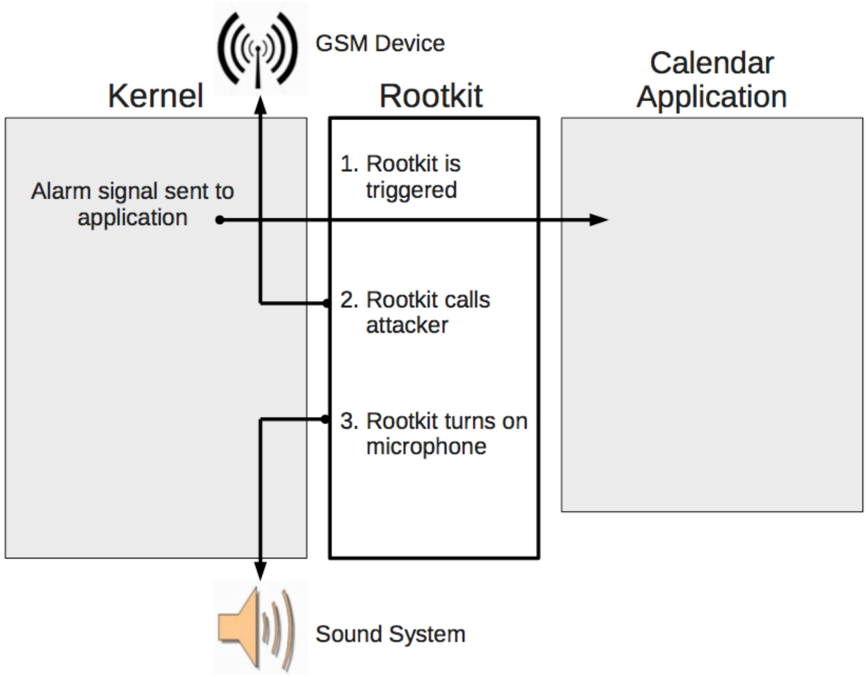
GSM Device
Kernel
Rootkit
Calendar Application
Alarm signal sent to application
1. Rootkit is triggered
2. Rootkit calls attacker
3. Rootkit turns on microphone
Sound System

Figure 5.2: Example scenario of spying a conversation via GSM from [BOB+10]

or WLAN. Such additional services are very resource intensive and consume battery power. The rootkit can also modify the status of the services retrieved by the user and the user has not to be aware that these services are running. But after some minutes, the battery can be empty and the result is a Denial-of-Service attack on the mobile device user. [BOB+10]
Todays mobile devices compose of many sensors like microphone, camera, GPS, accelerometer, etc. The problem of securing these sensors is their usage, because they are often used on behalf of the user. But the access to these data should not be enabled for not intended applications. Another problem is, that mobile devices are often carried unused (the user is not aware whether his device is now working or not), but these sensors can be attacked also during such phases. The presence and the standby mode of these devices make the possible gained private data more valuable for the attacker. All these threats can be categorized as OS Vulnerabilities (see Section 5.3), since these sensors are present in the device and are managed by the OS.

5.3.4 Video-based spyware

The authors of [XZL+09] claim to be the first team, which developed a video-based spyware for mobile devices. In their implementation, the spyware controls the video capable built in camera of the mobile device and covertly records some video sequence and sends it from the mobile device without the notice of the device user. Interesting challenge while developing such a spyware is also the recognition of the moment to record the video, since the controlling of the spyware from remote is not always possible. But for such recognitions, other built in sensors (microphone, light sensor, accelerometer, etc.) can be compromised and used. Such an attack can be categorized as OS Vulnerabilities (see Section 5.3).
The software architecture of the proposed spyware is shown in Figure 5.3 and identifies the necessary parts of such piece of software on the example of the Windows Mobile 5.0/6.0 OS: Video Capture, Trigger Algorithm, File Sending, Camera Driver and Network Connection. The workflow of such a spyware could be implemented as shown in the Figure 5.4. [XZL+09]

5.3.5 Resources Constraints

The main constraint is connected with the size and the mobility of the device. In fixed environments, the size of HW components and the power supply are not a limitation as it is in mobile environments. Malware detection deployed in PCs consume many resources (CPU, memory and power supply) because of the complexity of detection algorithms. As the malware detection becomes more difficult with the time and the evolution of malware (some

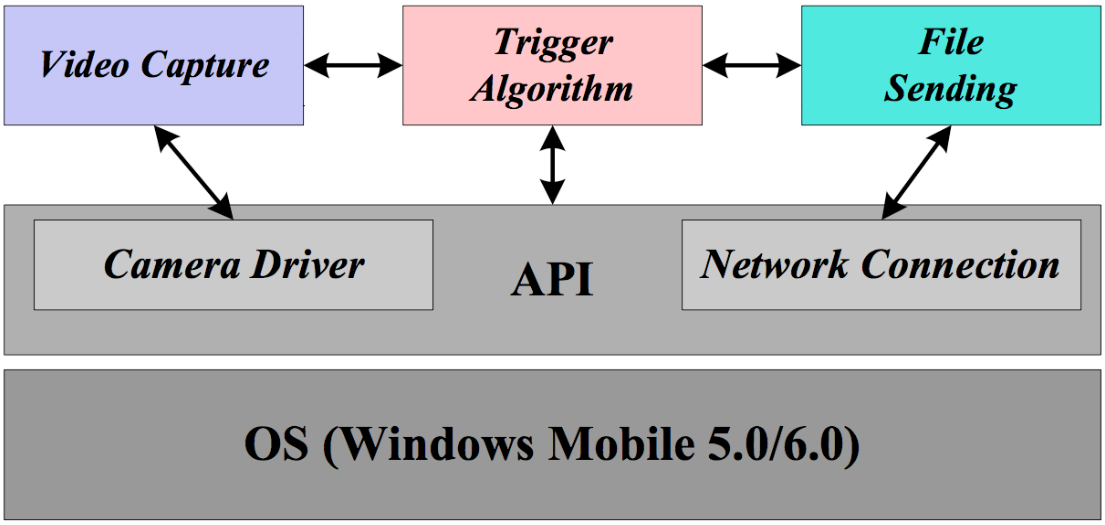

Figure 5.3: System architecture of the video-based spyware from [XZL+09]

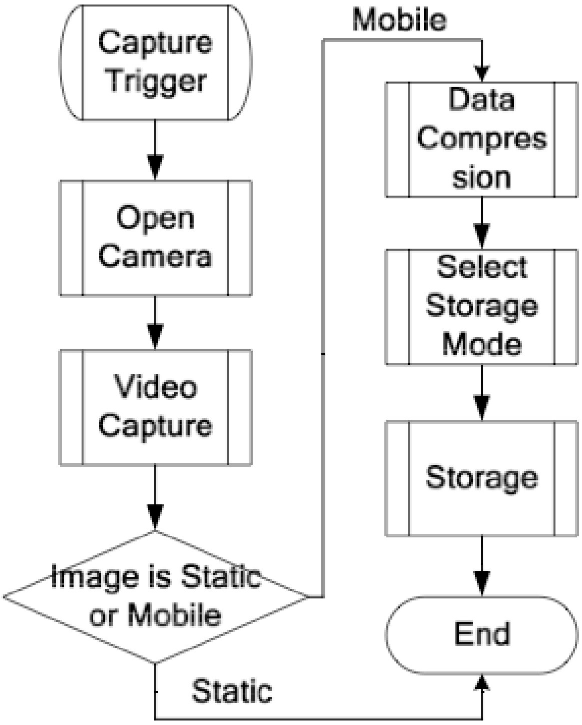

Figure 5.4: Sketch of the workflow of the video-based spyware from [XZL+09]

behavioral monitoring and more complex detection are necessary), more and more resources are utilized for such a detection model. Therefore these algorithms from fixed environments have to be reconsidered to achieve high scalability in mobile environments. "If mobile attacks follow a trend similar to traditional fixed computing, they will may face a deluge of new threats, requiring detection systems that can scale elegantly to handle a diverse and sophisticated threat landscape." [OJ10]

5.4 Installed Applications

Todays mobile devices with their OS structure as discussed in the Section 2.2 allows the user to install additional applications. According to Section 2.3.1, where the problem of application approval was introduced and discussed on OS examples, the problem of new installed applications is their interaction with the mobile device and its OS. All attacks based on the security leaks in installed applications and their interaction with the whole environment can be classified in this group.

The designer of the OS has to consider the new possible applications as a security risk and should secure the OS and other applications (inclusive their common device wide and private application based memory). Because an application can contain malware and attack other applications or the OS itself. But nor only harmful application can be used for attacks. Also vulnerabilities and design mistakes can make simple and useful applications attackers instruments.

A possible example would be an (not originally harmful) application, which deletes or modifies the contact database of the mobile device. It is necessary, that the (not only mobile phone) OS designer sees this new installed applications as extensions of the OS itself. These new features add also new risks, which has to be considered.

5.4.1 Mobile Phone as Secure Entry Point

Mobile devices contain sometimes more personal related data than PCs and are also always carried wherever we move. Since their mobility, using them as authentication keys would be convenient. The todays possibilities of these devices allow to use them as entry points for many daily systems, for example at workplace (authentication into a corporate network) or by payment activities. But such systems build up on mobile devices can be compromised, if they contain security leaks in their deeper layers, like here the mobile devices itself and the utilized applications. Therefore the crucial question should be discussed: are todays mobile devices a secure entry point for the daily used systems?

The authors of [dGL08] proposed a novel approach for an authentication system. It is a combination of "an innovative graphic-based challenge-response authentication mechanism with the security of a standard two factor authentication scheme.” [dGL08] For this system, a secure mobile device is necessary, equipped with a client application and a camera. Such an approach eliminates passive and active phishing attacks and eliminates the necessity of a bank card. But before deploying such a system, the above mentioned question has to be discussed carefully.

Another authentication approach, which utilizes the mobile device, is the proposed Small-group PKI-less (Public Key Infrastructure, a very complex paradigm for assigning keys to concrete subjects to enable encrypted information transfer) Authenticated Trust Establishment from [LSH$^+$09]. The authors proposed a communication protocol for small groups (eight or fewer), where keys can be generated and exchanged for future secure communication. One of the explained use cases is a simple scenario, where a small group of people meets personally, with usage of their mobile phones Bluetooth capability, generate and exchange their communication keys, which can than in turn be used for the future communication. Such a use case also require a secured mobile device to be the proper secure entry point for such a proposed protocol.

The authors of [KEAR09] proposed an approach called ”On-board Credentials”, which ”combines the flexibility of virtual credentials with the higher levels of protection due to the use of secure hardware.” [KEAR09] An example of hardware security token is the, in mobile phones omnipresent, SIM card (Subscriber Identification Module, a removable card for storing the identification of a mobile network subscriber to be independent from the mobile phone device). The deployment of secure hardware with combination of such an approach would make it possible to use the mobile device as a secure point.

5.4.2 Guiding People

This scenario of guiding blind people is similar to the other mentioned examples of special assistance. It is another user setting, where the device is used as usual (installing applications, surfing the web, playing games, etc.), but further for one critical case, namely guiding of a blind person. Possible attacks on this device could have many negative impacts, and could even harm the person physically. A possible scenario of an attack could be one, where an attacker gains access to the mobile device via a security leak and modifies the guiding route. This security risk can also be classified in two groups. Threats classified as OS Vulnerabilities (see Section 5.3) could harm the OS and gain access to critical parts in this scenario. Another group of these threats can be classified as Installed Applications (see Section 5.4), which could somehow influence the guiding procedure.

Guiding people with the use of mobile phones is discussed for example in

[KGE11], blind people are considered for example in [NGR09]. For a prototype implementation utilizing a mobile phone only with acoustic commands, refer for instance to [LBH08].

5.4.3 Malware

Authors of [SSB$^+$09] implemented successfully a malware for the Android OS. For malware a hosting application is necessary, which has some simple functionality and some lines of code and is able to execute the malware. For this purpose, the malware application itself can be included as ”raw source” (e.g. as .png) into the mentioned hosting application. After the hosting application is run, the included raw source is made executable (renaming the .png file and making it executable). In the next step, the malicious code can be executed. The authors successfully included in such a malicious functionality some shell code to reboot the mobile phone. These threats can be classified as Installed Applications (see Section 5.4). [SSB$^+$09]

5.4.4 Mobile HCI and Usability

Many applications for mobile devices are downloaded and installed - each application has its own purpose: working with Facebook, playing a game, etc. The lack of isolation of these applications is also a security challenge in developing mobile devices and should be designed very carefully.
Another problem is the usability and the circumstances, when the mobile devices are used. As deeply discussed in [Soh08], often user utilize mobile devices while working on another task to gain some information. This should be also considered from the viewpoint of usability and special characteristics of mobile devices and their difference to other environments. These aspects are important when securing mechanisms in general.

5.4.5 User is the Enemy

One of the problems of todays proposed security mechanisms is their presentation to the user. For instance the pop-up window in the right bottom corner, known from Microsoft Windows, is a good example how easy it is not to install new security updates. Such a user than becomes the enemy of the system. Therefore, each proposed security mechanism should be also considered from the viewpoint of the usability. Mobile devices are used by many people, containing all possible user groups, such as user without any security knowledge, without the possibility to understand the security model behind, without security awareness, etc. [VCU08]
Discussions in this topic would make the future proposed security mechanisms more usable and therefore enhance the overall security. Since, for example, if all security updates would somehow be made obligatory and not

optional, the overall security of such an interconnected system would be much higher.

5.5 Mobility

This covers two not clearly separable aspects of mobile phone usage and the associated attacks. One aspect is the mobility and the mobile way of usage and the second aspect is the substitution of other devices in some use cases like the PC. For detailed description on this classification group refer to the following section.

One part is the mobility itself, because todays mobile phones are used in a very mobile, always changing environment, like workplace, meetings, lunch, business trips, etc. Since one of the most used communication possibilities of mobile phones is the transfer of voice during a phone call, there are many possible attacks on this aspect of usage like a simple overhearing attack. This subclass of mobility concentrates more on impact of the mobility to the existing and often used workflows, which where not introduced and considered for mobile use cases and the specifics of mobile environments. One possible simple example is the entering of PIN in an overcrowded public transportation vehicle with visible keyboard. Such a four digits long number, which can be overlooked from the next person and remembered easily, does not really gain the intended security level. It is a difference to secure a PC, for example a server in a server room, to securing a mobile phone. If an attacker needs physical access to such server computer, he needs to brake in. But to stole such a mobile device, or only gain access to the data, the attacker has much more possibilities to do that. It can be also seen in daily life examples, how easy it is to loose a mobile device in contrast to loose a server from a server room.

The second aspect of this classification group is the substitution of other devices like the PC in the above mentioned mobile environment. Usually, when a PC user writes mails, browses the web or makes some banking transactions, he chooses the appropriate environment for it. For banking transactions, a PC in an internet cafe is maybe not the securest entry point to such systems. But the problem with mobile phones is, that the environment is changing all the time and it is not so easy to secure it.

5.5.1 Forensics of Mobile Phones

Another factor for mobile phone deployment and their supplement with special hardware and software could be the motivation of forensics. A mobile device can be (relatively more easily in contrast to PCs) stolen or just get lost. When this happens, forensics of such a device can become a security threat.

The main problem of mobile phone forensics is their diversity compared to

PCs - many manufacturers, many operating systems, many interfaces, etc. Mobile devices contain much more personal related data than PCs, but this information is not stored because of memory lack. [LK09] The discussion of forensics future possibilities could be also a motivation for deployment of special hardware.

5.5.2 Attacks on more Personal Related Data

Attacks on mobile phones occur rarely in contrast to PCs, but this can change in the future. The private, person related data, stored in mobile devices (SMS, calendar and contacts, possible storable data like voice communication, sensor sniffing, etc.) are valuable points for attacks. Therefore much more work should be invested in securing this private data to obey the parallel in the world of PCs, where the security was introduced after successful attacks and not before.

As discussed in Section 5.5, mobile environments have some unique conditions in contrast to fixed environments. Therefore the described protection goals (see Section 3.3) should be applied to mobile environments by considering these unique conditions. This section does not concentrate on mobile diversities, but rather on impacts on security considerations resulting from these diversities.

More personal related data in mobile phones are discussed for example in: [POS+01]. [KBS09], [MHM+10].

5.5.3 Mobile versus Fixed Environments

According to [OJ10], these security challenges in mobile environments can be grouped as follows.

Missing architectural considerations in mobile phones are a challenge in mobile environments. Many mobile devices are often used for both private and business concerns. If the virtualization would be considered in the architecture, it could be possible to virtualize the private concerns and isolate them from the business part. For example installation of new games could not harm the business system and data. Or some secure HW components could be introduced for some security algorithms.

The mobility of mobile phones introduces also a new "physical" security threat connected with the (mostly insecure) environment of usage (as classified in Section 5.5). These devices are small and portable and therefore a successful target for any kind of stealing. From the thief's point of view, these are relatively small things, which can be stolen easily because of the frequent usage and also because they can be easily resold without being discovered. According to [Hal04], these mobile devices are more and more targeted in robberies. As mentioned by the authors, also the forgetfulness and oversight of employees causes, that such devices are very often left

somewhere (taxi, public transportation, etc.). This all is connected with the mobility and the size of mobile phones.
Mobile devices often contain much more person related data as usual PCs, because they are used the whole day long in many environments and in many use cases. Therefore the person related content is very interesting for any kind of (legal or illegal) social engineering. The insecure public environments are good opportunities to gain physical access to these personal devices. One could, for example, try to copy the personal data while the owner concentrates on another activity. Without the notice of the owner, the personal data can be stolen and the device has not to be taken.
As also mentioned in [Hal04], the company security is also affected by the deployment of mobile devices. Companies introduce some security schemes including rules for the employee's. But these considerations today have to deal with personal mobile devices, many times owned by the person as a private device (not owned or deployed by the company). Employees often use these devices for private and company's concerns (checking private and company's mailbox, making private and business calls, etc.). This new emerging security threat has to be also considered in the company's security rules and advisements.

5.5.4 Special Assistance

The example of special assistance in a car accident, where the mobile device is interconnected with the car system computer for providing communication services, reveals some threats classified as Mobility (see Section 5.5). In a such scenario, the mobile device is used because of its communication capabilities. But if the unsecured mobile device is compromised, it could attack the car system, because of additional (not necessary for the car system) possible functionality of the mobile phone. In this concrete use case, such an attack on the car system while travelling with the car could also be very critical. The user can not notice any changes in the device while concentrating on the journey. [BIM$^+$09]

5.5.5 Spontaneous Usage

An interesting and considerable observation was made by the authors of [JSB08] and was called "spontaneous usage". They describe a use case of a mobile device, where a user utilizes his device only for seconds during the use case procedure and in special environmental circumstances. A spontaneous user, as defined by the authors, is for example a person on the airport, who has to check new incoming mail within few seconds. In contrast a user, who utilizes his mobile device many times a day (for a relatively long period) for example during his work day, is ready to accept a relatively (maybe some minutes) long login procedure (which repeats maybe some times during the day) to gain access to needed services.

The period of use is, in contrast to the possible login period, relatively short when speaking about spontaneous usage. Therefore it is conceivable, that such spontaneous users are not willing to undertake such relatively long login/logout procedures and are a risk to implemented security mechanisms. This case also has to be considered when designing and deploying security countermeasures and there must be a possibility to undertake such short spontaneous usage without complex login/logout security procedures. If these spontaneous users are not considered in the design and development, they will then became the enemy of a system as discussed in the Section 5.4.5.

Chapter 6

Approaches for Improving Mobile Phone Security

According to the classified attack surface of documented and possible threats, the security improvements for mobile phones are introduced in this chapter. Countermeasures for some specific threats will be discussed, dealing with a kind of state-of-the-art security countermeasures proposed by researchers aligned with the proposed attack surface defined in Section 5.1. Each discussion of a security improvement is followed by the current open problems in mobile systems, discovered during the research and development of this thesis.

6.1 Communication Interfaces

As defined and discussed in Section 5.2, some improvements for this group of attacks are presented and discussed.

6.1.1 Model-based Security Analysis

The authors of [JSB08] proposed a framework, where UMLsec (a security extension in form of a profile to the Unified Modelling Language, a language for modelling specifications in a graphical way) is used for security analysis for mobile communications. However, in their case study of modeling 62 security requirements, they did not find any appropriate representation in UMLsec for 13 requirements. As the authors learned from the case study, such an approach is not feasible for complex security mechanisms as their representation in UMLsec is not possible (because a model always abstracts from the reality, which can be crucial in complex security scenarios) in a feasible way. As an example for that, for 15 requirements the UMLsec had to be modified. The authors also summarize, that the analysis did not find many security problems in unsecured and untested systems, but claim that the reason is, that UMLsec was the first time applied to mobile communication. "Furthermore, by embedding the security analysis directly into the IT development and management process, a better understanding and clearer communication of these issues is made possible." [JSB08] Such an security analysis would minimize all groups of the classified threats and would be very useful for the most complicated of them, like the combination of Communication Interfaces (see Section 5.2), OS Vulnerabilities (see Section 5.3) and Installed Applications (see Section 5.4).

6.1.2 A Social Network Based Patching Scheme

The authors of [ZCZ$^+$09] proposed a methodology, where the distribution of a worm could be blocked by the network operator before the propagation could harm the network. The proposed patching scheme should be based on the social network of an attacked mobile device - it's address book and recent calls. The patch is therefore concentrated on those mobile phones, which connect social clusters together. These are patched first, since such root nodes could infect whole clusters.

Based on these considerations, authors propose to construct a social relationship graph of mobile phones, which contains often the next worm targets after the device itself gets infected. A message (SMS, MMS, e-mail) sent from the device to some node from its social relationship graph is not considered as suspicious by the receiver, since this message comes from the receiver's social network. Therefore the receiver is ready to open (and therefore also execute) the contained worm without his notice.

Since the network provider always collects all generated traffic of the mobile device for accounting reasons, the proposed methodology should be deployed at the network provider. This enables a directed patching in case of worm attacks, without the necessity of patching too many network participants, which could overwhelm the device network. The problem of such patching would be not only the resources, but also the time needed for those patches. A worm spreading has to be patched in seconds and minutes, and not in hours and days. The authors call it targeted patching: "find a small set of nodes with the highest priority for patching, while keeping the infection rate as low as possible." [ZCZ+09] Such a strategy is also useful, when the worm is discovered, but the patch is not yet known. Then the provider could set up some node-limitations (communication, bandwidth) for the infected nodes to slow down the propagation.

Such a framework would lower the success of an attack against communication protocols significantly (classified as Communication Interfaces in Section 5.2).

6.1.3 Network Obscurity

Todays communication protocols for cellular networks (GSM, GPRS, UMTS, etc.) are very obscure and much work has to be done to get rid of this negative architectural design and implementation. The future will necessarily bring evolution of these communication protocols mainly because of data transmission speed and the amount of communication participants. An open, well documented and designed communication protocol would make the improvements easier and faster. [OJ10]

The obscurity of such crucial protocols is connected with future discovered possible security leaks and following attacks. DoS attacks (Denial of Service attack, where available resources of a system are attacked with the result that this service is for some time no more accessible) on cellular networks with todays architecture and daily usage would have meaningful impacts on the service.

Improvements in this fields would improve the securing mechanisms against attacks classified as Communication Interfaces (see Section 5.2).

6.1.4 Virtualized In-Cloud Security Services

The mobility of devices is ensured through resources limitations and therefore centralized security mechanisms deployed off-device should be emphasized. The future work should concentrate on such frameworks from the viewpoint of deployability. A similar approach will be discussed in Section 6.2.7, that is based on behavior of user initiated actions, which is different to them initiated by malware. The problem could be in the future, where this automated and user initiated actions will maybe be distinguishable with much more effort than today. From the viewpoint of centralization, as proposed in Section 6.1.2, also the social based approach could be utilized, but this is a real privacy problem today and has to be solved carefully. But the current hype of could computing could enable future research on such mechanism.

Such improvements would also affect more then one classified group of attacks.

Malware detection on mobile devices consumes many resources, as CPU, memory and therefore also power supply. The authors of [OVC$^+$08] propose a possibility to delegate these resource intensive tasks from the mobile phone to off-device network parts. They claim, that the evolution and future complexity of the OS of mobile phones will also bring the necessity of complex antivirus software on these devices with it.

The proposed framework from [OVC$^+$08] should, according to its authors, bring the following three benefits: the detection of the malware should be better, because the off-device resources can have much more power and complexity as the mobile device itself; the overall on-device resource consumption can be still lowered though data have to be transferred to a network service for analysis, and future extensions off-device also lower the overall consumption; the on-device software complexity can be reduced with a simple mobile device agent and the complex (evolvable) off-device functionality.

The approach of the authors is based on a mobile device agent, which collects files and sends them for future analysis into the network. The network service then analyzes the data and identifies the malware. Such an architecture could be deployed also by third-party vendors and also by a mobile service provider. The simplicity of the mobile device agent is positive for resource consumption on the mobile phone, and the off-device functionality also enables to make the service very complicated and resource intensive. One could also combine many virtualized detection engines without any resource conflicts. By the growing amount of serviced mobile agents, more service container instances are created. Cashing of the remote and local files can also significantly improve and speed up future access. Also virtualization and behavioral analysis in the remote resources would not exhaust the power supply of the mobile device.

The off-device functionality is also interesting from the viewpoint of additional security services. Complicated problems like SMS spam filtering, phishing detection or centralized blacklists could be more effective solved at remote resources. By aggregating information from more mobile users remotely, the spam SMS identification can be done much more precisely. Also phishing detection could be improved by service providers, if they could see this process centralized at the off-device resource. And last but not least, blacklisting of some malicious nodes in the mobile device network is easily possible, if the centralization would be established. And each extension of the described security threat detection system would be possible by simple deployment on the remote off-device resources.

The main limitations of this proposed framework, also mentioned by the authors, are the offline mode and privacy. Since such introduced pattern works only if the mobile device is always connected, in the time where the device can not utilize the network, suspicious actions can not be detected. But as the authors recognize, the time where the mobile devices will be in a disconnected mode, will be minimal in the future. And since for the behavioral detection also some confidential data has to be sent for future detection, users of such framework have to be aware of these privacy implications and it should be possible for them to choose which privacy data can and which can not be transmitted over the network to the remote detection resource.

The authors also conducted some tests and evaluations, which resulted in following concluding statements: "Our results demonstrate that the current model of on-device antivirus software is not scalable. As the number and complexity of mobile threats increase, on-device engines and their signature databases will require more processing, storage, and power. On the other hand, our mobile agent remains constant in its resource requirements and can easily accommodate new signatures and entirely new engines in the virtualized network service." [OVC$^+$08] Other interesting results are, that the proposed architecture really lowers the complexity of such antivirus software and also scales on any platform with a simple adjustment of the on-device agent. Some related work was also done in [SPL$^+$09].

As such services would allow to deploy a very robust anti threat mechanisms (malware, virus, etc.), improving the overall defense ability against all classified attack groups, future work on this topic has to be continued with much greater focus.

6.1.5 Obligatory Updates

Some organizations need policies, where security updates are made obligatory. Therefore a possibility to deploy such a policy to mobile devices should also be available. For example, notification via SMS that updates are available could be pre-payed. Then, organization's centralized management of devices (by admins) could check for done and undone updates and force user

to update (policy restrictions, cancelled salary bonus).
Such obligatory updates would improve the overall security of deployed OS and would therefore improve the resistance from attacks classified as OS Vulnerabilities (see Section 5.3).

6.2 Operating System

As defined and discussed in Section 5.3, some of the improvements for this attack surface group are presented in the following sections. Starting with general design principles for establishing security and following with concrete examples of possible improvements.

6.2.1 Design Principles for Establishing Security

To recapitulate the design principles already known from PCs, the principles formulated in [SSS75] will be considered. The description is based on mobile examples.

Economy of Mechanism

The implemented mechanism should be simple and conceivable, because complex mechanisms are hard to test and prove, that they do not contain any leak. Also the normal use often does not show the security problems of a complex design. Sometimes after some (automated, frequently repeated) testing a possible security threat is discovered, sometimes only when special circumstances occur. Or even the leak could be discovered very easily, but is not tested due to the complexity of the mechanism. On the example of OSs of mobile devices, if some functions are implemented against this principle, the possibility of testing the implementation and discovering the hidden security risks is not possible in feasible time. Therefore some leaks are discovered only after the deployment of the OS and then the problem of fixing rollouts grows.

Fail-Safe Defaults

Maybe the dominant source of security risks is the gained excess to interfaces, which can cause security problems. By disallowing access to all possible interfaces, the possible security problems are minimized. This is not the solution, because then less functionality can be offered, but it is the desired paradigm to start the design. Sometimes developers argue why these or that functionality or interface should not be accessible to other components because of security concerns. But this design principle underlines the necessity to turn such a way of thinking around. The developer should argue, why this or that functionality should be accessible to other components. On the example of security consideration in the mentioned OSs of mobile phones, the access to personal data stored in the device should be disallowed for applications per default.

Complete Mediation

One possible access, which is not mediated by the system, can cause some security problems. Checking only security critical access does not comply with this design principle and has caused many security problems in the development of PCs. The OS of the mobile phone has to ensure, that each access to each component is checked and mediated. Also if saving access information is used to obey repeated checks of access privileges, the up-to-date status has to be ensured and updated after changes. In a possible example of a mobile phone OS, where only (considered as) critical access to critical resources is checked, an attacker could combine these to gain the not intended access to a component.
Another very important aspect is, that if the application developer implement his own checks into an application, diverse checking implementation are produced. These can not be checked so precise and deeply, as if there would exist a central implementation responsible for the complete mediation of security checks.

Open Design

The security of a design should not rely on the fact, that the implementation itself, parts of it, or its design will stay secret. Because then nobody can be sure, that the design and/or implementation is secure. It should be clear for everyone (maybe also provable), that such a concept is based on logical impossibility to brake the mechanism. A good example from the world of PCs are the encryption mechanisms. If such en-/decryption algorithms were based on their secret implementation, sooner or later they were revealed. But if these algorithms were based on the openness and widely accepted logical safety, their usage lasts for many years, as for example the usage of prime numbers in combination with asymmetric cryptographic techniques. A possible example are the above mentioned access checks for privileges. If the mechanism for checking the access rights in mobile phone's OS would be based on its mystery, sooner or later somebody could, for instance with reverse engineering the algorithm, gain access to not privileged components. Open design does not mean that a system is automatically more safe than one without open design. But each open design can be checked more intensively than a closed one. Then regardless of which security algorithm is chosen, it can not be based only on it's nondisclosure.

Separation of Privilege

An example of mishandling this design principle in the world of PCs is the possibility to gain access to the whole system after authentication. This design principle is known for many years before the age of PCs in the environment of finances and banks. If access to critical information has to be assigned, two separate persons have to provide their separate keys to allow the access. This combination of two persons and their two private keys increase the security barrier necessary for access. An example in the field of mobile phones would be no separation of access after submitting the correct Personal Identification Number (PIN), a secret number shared between a user and a system to gain access. If the whole access (usage and modification of data, settings and applications) would be gained, the only necessary thing to compromise the device would be to somehow snatch the PIN.

Least Privilege

Each subject (no matter if user or a component) should have the minimal necessary privilege. If for example the user has only privileges to use a system but not to do any harm to it by misusing them, such a system is secure against the user. In case of disrespecting this principle, a login for example into an OS of a PC as a user would gain access to all levels of security. Also the OSs of mobile phones should follow this principle and separate levels of access and gain only minimal privilege set to each entry level. The applications on mobile phones follow the incorrect pattern from PCs, where each application needs admin rights even for the execution. Admin rights are necessary only for the installation and the connected initial setup. But it is wrong to use them also for the execution. Because in daily life examples, these applications are utilized many times a day (which is the original intent of installing an application into a mobile phone), there is no real need that these applications are executed with rights of an administrator.

Least Common Mechanism

Design of common mechanisms with common used variables should be considered wisely, because it represents a possible information path. On the example of OSs of mobile devices, a designer should minimize functions, which can be used by all users and should also be aware of security risks of such common functions and design them carefully. Not each generalization of functions to make them usable by all users is wise according to this design principle. Some generalizations can be made and library functions can be implemented. Those users, who need different functionality, can use other functionalities implemented only for them.

A prototype example is a use case, where two applications are utilizing the same file. These two applications are reading from and writing to this file.

If one application manages to change the content of the file right before the other application is reading from it, the first application could compromise the second one.

Psychological Acceptability

The whole human interface should be implemented in an for a certain user acceptable way and should force the user to use these security implementations as intended. If the interface would be complicated and therefore misunderstood by the user, it could lower the intended security mechanism. An example could be the requirement of only numerical and 20 characters long secret numbers necessary for daily usage of mobile phones. This would most probably result in having all the users noted these numbers directly somewhere on the device and such a handling would reduce the desired security level.

Reusing and Adapting the Current Security Level

A certain level of security in other than mobile systems is already achieved, like the fixed environments of PCs. Also many experiences have been made with embedded systems, which are sometimes mobile, i.e. changing their location. Therefore the researchers have to adopt the used and evaluated security mechanisms from other fields, but should not forget to adapt them because of the emerging mobile environment, as discussed in Section 5.5.3. [OJ10]

Nowadays exists many platforms and therefore researchers often concentrate only on the most used ones, but there should be an orientation on multiple platforms in the field of mobile device security. The research should concentrate on development of reusable security countermeasures, because such security counteractions could be repeatedly applied (with adaptation) to other than mobile environments. [OJ10]

Researchers, when developing future security mechanisms and use cases, should consider also future possible use cases. A good example thereof is the development of SMS, which was intended for short message communication, but the problem of SPAM and malware propagation was not considered. Therefore, future use cases based on future trends (not only on mobile devices, but also on fixed environments) have to be considered when designing security countermeasures. This is also true for the future possibility to scale such mechanisms.

6.2.2 Secure Sensors

According to [CMC09], there exists three types of application access to the sensors which makes securing of them even harder. There are applications dominated by sensors, where at the start of the application, the sensor is

turned on (for example making a photo) and after the application is closed, the sensor is turned of. The second group are applications supported by sensors, where the sensors are used only in some of the use cases. In this examples, the sensor is turned on for a short time (for instance turning the microphone on for recognizing a speech command) and immediately after that turned off. And the last group of applications are those, which use only the context provided by such sensors. One example could be the usage of camera for sensing the light characteristics of the environment. [CMC09]
A solution for these security concerns would be a prevention system without much interference with the mobile device user. From the viewpoint of security, this framework should not allow recording sensor data to malicious applications. No user interaction would be the ideal solution from the viewpoint of usability, but if such interaction have to occur, the user has to be able to decide in line with his security policy and has to be therefore informed and provided with enough information. Because of many nowadays implemented and deployed mobile device environments, the proposed solution should not be connected with re-implementation of each until today implemented application. The solution should also have high performance and be suitable for all the todays and future platforms and different sensors.
The authors of [CMC09] proposed a framework of a defense system against sensor sniffing consisting of: policy engine, interceptor and user interaction. This framework is shown in Figure 6.1 and concentrates on protecting sensors from unintended access and disallows sending this information from the device (the problem of physical access to the device is obeyed in the framework). "The policy engine determines whether to allow each access, based on the input from user interaction and application monitoring/profiling. The interceptor enforces the decision by the policy engine." [CMC09] Based on the framework description by the authors, possible solutions and their drawbacks will be discussed in the rest of this section.

The decisions made by the framework should be based on a policy guided by application monitoring and profiling, not by intervention of the user for his decisions. One of the solutions could be the whitelisting/blacklisting of applications, which could be also later on (maybe frequently) updated. Applications dominated and supported by sensors can be white-listed, but for the context provided applications, more granular approaches are needed.
Another part of such a framework could be tracking of the information flow, where the collected sensor data could be tracked. It is not enough to simple allow each application, which does not access the network, to collect sensor data, because another application could send such personal information via the network. Collected sensor data could be therefore tracked and each usage of this kind of data could be monitored by the framework. The problem is, that such monitoring is connected with performance degradation and an application could also compromise this tracking. And last but not least, there exists applications, which need to collect sensor data and then access

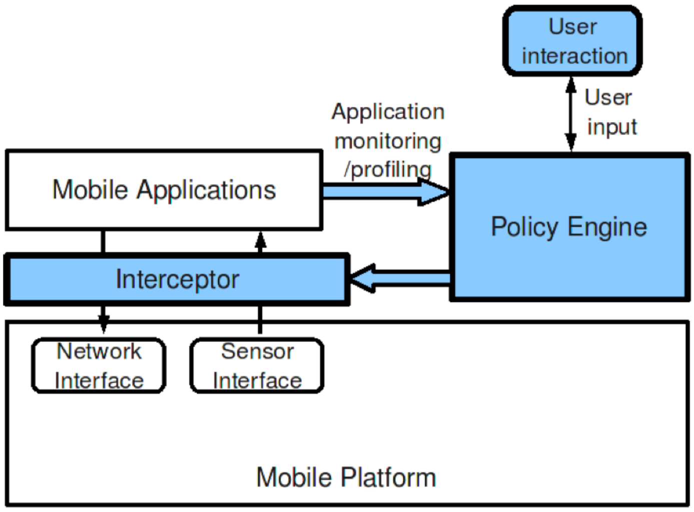

Figure 6.1: The proposed framework defending sensor sniffing attacks from [CMC09]

the network, as for example a simple application for uploading taken photos.

Policy decisions are not always possible and therefore user interaction is necessary in some cases. The authors argue, that a user decision for sensor access is something extraordinary, because each user understands the purpose of these sensors and is aware of problems connected with for example microphone access. Another improvement proposed in [CMC09] is the "Sensor-In-Use notification", where the user would be notified about the sensor activity and could avoid confidential actions (speech, movement, camera view, etc.). The sensors are something different than persistent data, because (if we do not concern caching of sensor data) this confidential data can be used (or somehow recorded) only when the sensor is active. But this assumption only holds, if the notification could not be attacked, as for example mentioned in Section 5.3.4 based on [XZL$^+$09].

An important part of this proposed framework is the interceptor between the sensor interface and the mobile application which needs to access the sensor. It is necessary that the interceptor intercepts each access and the OS itself is capable of some kind of sensor lock, where the access to the sensor interface can be blocked for a certain application (it would not be wise to block the whole interface for all applications).

The authors of [CMC09] also provide some novel solutions by leveraging mobile platform. These are based on the context aware platform, where the OS of the mobile device can also infer the current context thanks to this sensors and better support the decision of such a defense framework.

The user could for example define special places based on GPS coordinates, where all sensors have to be locked unless the user unlocks them. Another example of context aware platform is sensing the user activity. If the sensor is turned on on behalf of the user, then something is going to happen in the next seconds, for example if the user turns on the microphone, then there should be some voice to record. But if the sensors are turned on not on the behalf of the user, then some sniffing context could be recognized (no sound in there for a long time). Another proposed possibility would be the encryption of sensor data at the sensor. This data could then be decrypted only by suppling the necessary key and the malware application could not use this encrypted data. Such an approach would also help to avoid loosing of sensed data because of the current lack of access rights.

As discussed in this thesis, current mobile devices need a security mechanism, which really secures all sensors and checks each access. These sensors are something critical, each compromised person's sensor in a business meeting could do harm many companies in a second. They should be therefore motivated to invest in this research. A similar mechanism as firewall should have been integrated into sensors years before.

The problem of user authorization approach while securing sensors is, that many users are not aware of the connected problems. On the other hand, notification for current used sensors is not feasible in cases, where applications utilize the sensors very often. As the authors of [CMC09] recognized, "designing such a solution requires research into mobile user behavior, algorithms for automatic context inference, and operating system primitives such as information flow." [CMC09]

Securing sensors, which are mostly located as HW physically in the device and directly communicate with the OS, would lower the possibilities of attacks classified as OS Vulnerabilities (see Section 5.3). These sensors should not be directly connected with the installed applications and therefore such classified attacks should not reach the interfaces. This kind of access should be blocked by the OS itself.

6.2.3 Virtualization

Virtualization is a security improvement, which is connected with consumption of today's already available resources. Nowadays mobile phones have enough memory capacities to handle the additional memory consumption because of virtualization. The power supply should not be affected remarkable, since the virtualization is not connected with much additional computing operations.

On many PCs including laptops, virtualization is used to simulate other OSs. Many persons have their workstation at work, a different PC at home and sometimes an additional laptop. There is no need to separate business and private concerns via virtualization, because many times these two use

cases are handled on different devices. But the personal mobile phone is the place, where private and business sphere meets very often and every user would appreciate to have these two parts combined in one physical device. But this is mostly not possible because of the necessity having two SIMs.
If the virtualization of the OS, its applications and communication interfaces (including sensors) would be deployed securely, the motivation for one all-in-one personal mobile device would be increased. The main motivation to separate business and private concerns is for both subjects interesting. The employer wants the employee to work in a secure environment, and the employee on the other hand wants to handle his own private actions separated from the business part.
An example of such use case could look like the following. A user has his personal device capable of virtualization. At startup, he can choose between business and private environment. By starting one of them (in our example the business environment), the other one (here the private part) can not be accessed and nothing of this part can be executed. Because of a private concern, which occurs suddenly, the user is forced to start the private part. Thanks to the virtualization capability, the business part is suspended (the resources are freed) and the virtualized private environment is started. If the virtualization is deployed securely, there is no need to be afraid of security attacks on the business environment.
In such a scenario, where the user has the certainty, that due to virtualization, no security attacks can be made on the business part, the user is ready to pay with some resource loss (mostly power supply) for it.
This virtualization would mainly improve the security concerns with attacks classified as OS Vulnerabilities (see Section 5.3) and Installed Applications (see Section 5.4).

6.2.4 A Virtual Machine Monitor

The authors of [YLH+08] proposed a framework for virtualization in mobile phones, which allows a kind of mobile application classification and their separation thanks to the proposed virtualization. By utilizing this virtualization, application integration and isolation will be possible.
The virtualization itself, as it is known from the world of PCs, allows the portability of data and applications. If a user changes his mobile phone, other applications and settings can not be transferred automatically. But with the utilization of virtual machines, the suspended VM is just moved to another mobile device and everything is ported. Such VM copies could then also be used for backup and restore.

The authors claim, that their proposed framework is power and resource efficient. "MobiVMM will suspend or shutdown those VMs which have been idle for certain amount of time. MobiVMM monitors the battery condition and will prompt the user to shutdown some functionalities in low battery,

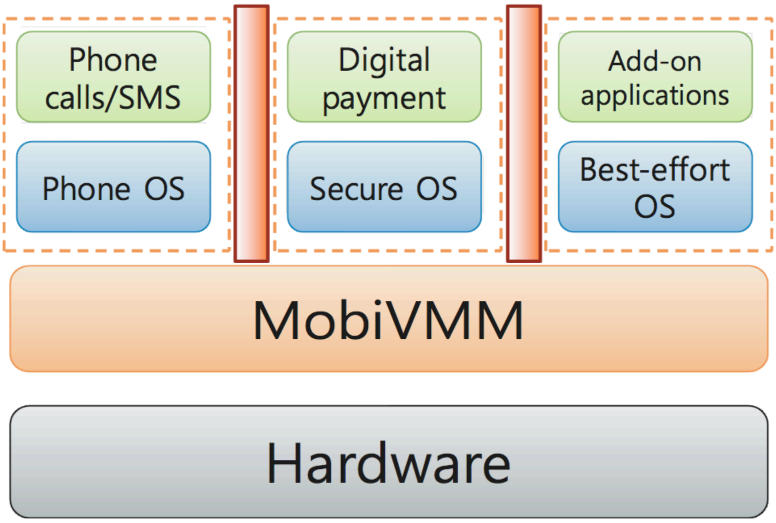

Figure 6.2: The design of the proposed framework MobiVMM from [YLH+08]

so that the core services keep running. Profiles are supported thus users are able to configure that low priority functions may be automatically shut down when battery is too low." [YLH+08]
Thanks to such virtualization, it would be possible to have more than one and personalized OS, which could be than started and suspended according to available resources. The authors conclude, that the "real-time property is achieved by priority-based preemptive scheduling and pseudo-polling mechanism." [YLH+08] Also future work is necessary, because the current implementation only works for Linux. Authors point out, that due to the fact of the variety in mobile phone models, the virtualization interface has to be defined very precise as the deployability on many mobile devices must be considered.
Such a virtualization would be very helpful against the attacks classified as OS Vulnerabilities (see Section 5.3) and Installed Applications (see Section 5.4), since the virtualization would improve the possibilities of separation.

6.2.5 Web Service Description for Viruses

The authors of [LCCC09] proposed a web service description of mobile phone virus behaviors in an ontology based environment. This conducted knowledge of virus behaviors could be then distributed through a web service. Because of the expression possibility and the model-based powerful knowledge of the ontology, also very complex mobile virus kinds could be described. While web services and their technology stack are used, the provided knowl-

edge is platform independent shareable. ”Ontology can not only present the essence of field knowledge, and can describe the concept, attribute of the knowledge content, and the relation between the concepts. Thus, it can present the knowledge structure through the theoretical foundation of Ontology and clearly describes the knowledge content.” [LCCC09]
Since this would improve the virus detection, this countermeasure can be classified for attacks from the OS Vulnerability (see Section 5.3) and the Installed Applications (see Section 5.4) group.

6.2.6 Intelligent Virus Detection

”In this paper, we have described a method to detect mobile viruses based on common functionality. Specifically, we used DLL import sets as features to detect viruses. Our evaluation on Symbian-OS platform shows that with a small set of training data, our system detects most existing viruses with a 0% of false positive. Our result shows that many virus families have common core functionality and our detection method is capable of determining this functionality and using it to detect virus.” [VHR06]
Same as above, this would lower the power of attacks classified as OS Vulnerabilities (see Section 5.3) and the Installed Applications (see Section 5.4).

6.2.7 A Behavior-Based Malware Detection

The authors of [XZSZ10] proposed a framework based on the difference between user initiated and malware compromised actions while using mobile devices. This approach concentrates in recognition of non-human activities and not in known or unknown malicious code. There is no need to train the system with false positives and it is capable of detecting also now unknown malware. The framework therefore can concentrate on state transitions of the cell phone and its applications itself, but also on the user activity. These two aspects, connected together, can be used to distinguish a user activity from a malware activity.
The framework considers a process as malicious, if the difference between the expected and the current state is remarkable, as can be seen in the model of the proposed framework from [XZSZ10] in Figure 6.3.
A malware tries to somehow compromise the mobile device and/or tries to propagate itself to another device. In both cases, resources are utilized in a way which is not conform to user initiated actions. ”This can be done through observing the key / display event pairs in the graph and extracting unique features including both the users personal operational patterns (e.g., time between keystrokes, keystroke durations, and pressure on touch-screen) and the sequence of process state transitions.” [XZSZ10] Theoretically, the proposed framework could also have user profiles, when a mobile device is used by more users, but often the personal mobile devices have only one

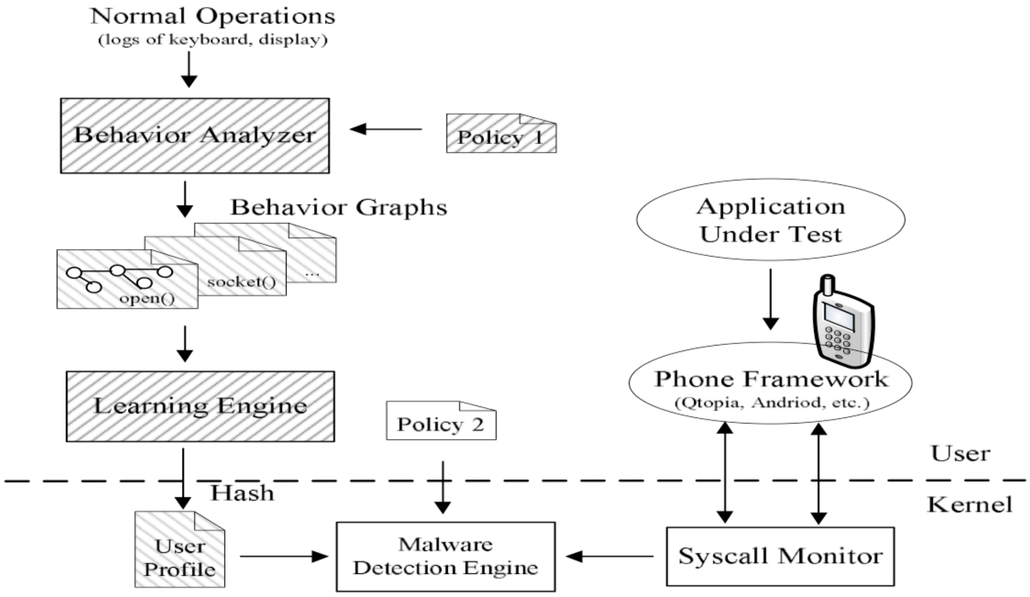

Figure 6.3: Behavior-based malware detection system in cell phones from [XZSZ10]

user.

Crucial part is the positioning of monitoring points for the anomaly detection. Mobile devices have different workflows and inputs as most of the PCs: flexible input methods, event-driven displays and limited number of keyboard elements. With too many monitoring points, the behavioral analysis would be too complex. A well recognized problem of this framework is the possible attack on the monitoring points (which would be a kind of kernel level attack).

The proposed framework utilizes graphs for the representation of the monitored actions and can therefore learn from new actions (modelling them as new nodes and vertices). Another problem can emerge, when some user actions defer from the, as graph modeled, expected workflow. Such a recognition would result in false negatives in the engine. One of the motivations of the authors for this framework was, that ”all existing cell phone malware do not even simulate users input events, their abnormal behaviors which cause abnormal process state transitions can be easily detected by our system.” [XZSZ10] The authors also argue, that the future evolution of mobile device UIs will still result in simpler UI of mobile devices than of PCs.

The problem of automatic generated workflows through other applications can be manually excluded, because the proposed framework allows exceptions. The authors conclude, that ”human intelligence-based defenses to differentiate malware from human beings hence becomes one of the most promising solutions for smartphones." [XZSZ10] Their framework is capable of detecting also nowadays unknown malware, but the remaining issues as kernel level attacks and the diversity of the OS of mobile devices have to be considered in the future work.

Since the malware, which could be detected, could result from all of the classified groups of attack, such a framework would improve the overall security and is not limited to one of the classes.

6.2.8 Deployed Secured HW

For some security countermeasures, a deployed secure HW in the mobile device would enhance the defense algorithm. It is understandable, that manufacturers do not include secured HW into their mobile devices because of additional costs, resource problems (considering also the size, weight and the layout design), etc. But a user would be ready to pay some extra costs (the employer would join him) for having a more secure device.

Therefore manufacturers should be motivated via a kind of secured HW certificates to integrate also this improvements. Then a user/deployer (mainly employer) could decide to invest in a more secure device for example for business concerns. The deployer could also be interested in future forensics possibilities to persecute crimes, which could also enhance the motivation for secured HW in such deployed devices.

Deployment of secured HW would improve the overall security of the device and would allow more robust securing mechanisms for all classified attacks.

6.3 Installed Applications

As defined and discussed in Section 5.4, the attack surface grouped under Installed Applications is discussed by presenting some improvement possibilities.

6.3.1 Application Delivery

This is the process of signing an application for the installation on mobile devices. It is necessary to identify the developer of the application, for which signatures are used. The main challenge is to find the way between restriction and allowing of all (possible harmful) applications.

The authors of [OJ10] classify the iPhone OS with "high" security of the application delivery, because each application can only be downloaded and installed from the Apple iTunes Store. But before an application can be published at the store, it is inspected and the store "maintains a remote kill switch that allows Apple to blacklist applications that may have already been installed on a device." [OJ10]

”The Android platform is given a medium rating due to its default setting to only allow applications through the official Android Marketplace, ” [OJ10] but by changing the default setting, also applications from other sources can be installed, which emerges the possible security risk.

6.3.2 Trust Levels

By installing an application and executing it, the user should have the possibility to define trust levels for the application. The main problem here is to find the middle way between simple approve/dismiss access to the device and between too coarse-grained trust levels, where the usability would be to low.
”For example, Googles Android platform is rated at high as it has a permission-based model that strikes a good balance of trust level granularity.” [OJ10] When trying to install the application, the user is shown a list of device parts to which the application requires access and the user can allow/deny the installation of such an application.
”The iPhone is rated at a low because it has very course grained permissions that only protect a few services such as the location of the user.” [OJ10] Windows Mobile OS for instance provide a classification of privileged, normal and blocked categories, which allows the user to group his applications in these three classes, which are connected with certain access rights.

6.3.3 Isolation

The isolation is connected with the before mentioned virtualization (for details see Section 6.2.3) and enables and/or enriches the proposed possible concept. If each SW part of the device is isolated properly, it is possible to suspend currently unused services.
The main OS functionality should be isolated from other applications and these additional applications should also be isolated among each other. The possible consequences of an attack could be minimized, if the isolation would be implemented correctly.
Since this improvement is interconnected with the before mentioned virtualization, attacks classified as OS Vulnerabilities (see Section 5.3) and Installed Applications (see Section 5.4) would be touched by this countermeasure.

6.3.4 System Isolation

System isolation is the possibility to execute an application in a sandbox-based environment, since a discovered vulnerability in such an environment would cause much less harm than without any isolation.
”For example, the iPhone platform is rated at low for system isolation as many of the applications run at the same privilege level.” [OJ10] A discovered vulnerability in an application could be utilized to access any other

part of the system.
"While a vulnerability within an Android application may allow an attacker to steal data owned by that application (e.g., steal cookies by exploiting a browser), other applications and the underlying system is isolated from the compromise since each app is executed as a unique UID." [OJ10] Therefore Android OS is rated as "high" in this category.

Table 6.1: A ranking summary of the security attributes (e.g., high is a positive ranking) adapted from [OJ10]

Mobile Platform	**Application Delivery**	**Trust Levels**	**System Isolation**
iPhone OS	high	low	low
Android OS	medium	high	high
Symbian OS	medium	high	medium
Windows Mobile OS	medium	medium	medium
RIM OS	low	medium	low

6.3.5 Secure Software Installation

The authors of [HA07] proposed using community feedback to help users to decide whether an application is harmful or not. The authors claim, that the most important drawback of the code signing procedure is, that it is cost intensive and therefore not feasible for non-profit usage. And the disadvantage of the user decision is, that the user is not aware of the possible consequences.
In this proposed framework, users can add details to an application located in a sort of database and this data can be updated. The framework considers each application as safe until something suspicious is reported by a user and can be in turn checked by some professionals.
Such a framework would improve the security against attacks classified as Installed Applications (see Section 5.4), since the discussed application approval would be improved.

6.3.6 Detecting Malware by Static Function Call Analysis

Detecting malware trough static function call analysis is discussed in [SCCA09] mainly theoretically without a prototype implementation. The question is, how it would contribute to overall security, but this must be first evaluated. Such analysis in general could improve application portals, where certification could be based on automated (static) analysis. Such a framework would then indirectly improve countermeasures against Installed Applications (see

Section 5.4) attacks, since it would lower the problem of application certification and approval.

6.3.7 Common Security Models

The problem of todays OSs of mobile phones is their usage of private security models. Researchers and academic society should concentrate on common security models and evaluate their usability on many platforms. If widely applicable and adaptable security models would be developed and the adaptation would be discussed and/or proposed, some security certificates could be introduced and the manufacturers could be forced to meet these certificates. Both the user and the manufacturer should understand, that these models ensure the security of private confidential data and a misuse can therefore be impeached.
Such improvement would have impact on the overall security and can not be scoped for some specific group of classified attacks.

6.3.8 Trust Levels and Access Control

Much more concentration should be on the application description and the necessity to define which resources an application uses and what data are processed. These definitions could be then also checked and a possible misuse could be discovered earlier. Maybe some authorities could be founded, where organizations would pay for checking this or that application for achieving such certificates. Also a company, which want that their application meets some certificates to get sold for business, might also pay to achieve these security certificates. Also some automated reports for applications, which do not comply with these policies, could be deployed to improve the certificate compliance procedure. All this, sometimes more complex, countermeasures could also require some security admins. Because companies are today forced to pay for security of their information technologies, they could integrate these additional costs spent on mobile phones into their budget. Such improvement in the context of new installed applications would improve the resistance of attacks classified as Installed Applications (see Section 5.4).

6.3.9 Application Store and Delivery

The certification process of the Apple's App Store for new developed applications and their delivery is something which really enhance the security. On the other hand, the iPhone unlock (a hacking procedure, where any restrictions implemented by the iPhone developer Apple are removed via a SW change) is something, which really lowers the security of the mobile device. Based on this approach, companies could sponsor such a payed service for having more secured mobile devices, where applications are actively checked

for security concerns.
But todays application certification has many limitations. The development of malware is made a little bit harder for developers, but is not minimized with application certification. The malware developer has to satisfy the security principles of the certification authority, but not the security policy of the user, which can be very different. Problem of hidden malware deployed in other applications (like for example games) is not eliminated with the application certification. The application behave as intended, with no obvious security risks, but the hidden functionality can harm the system. And last but not least, the problem of the amount of daily developed applications connected with the current model of application certification makes it nearly impossible to check and verify each line of code of new applications. Therefore "applications are often certified based on organizational trust relationship rather than technical verification." [CMC09]
These improvements would lower the problem of attacks classified as Installed Applications (see Section 5.4), since it would enhance the application approval process.

6.4 Mobility

As defined and discussed in Section 5.5, the mobility of a mobile phone generates unique impacts on the security and it's implementation in such environment.

6.4.1 Security Awareness of Users

The problem of security awareness is implicitly connected with secured systems - it is not enough to secure the system. It is commonly known and assumed, that the security of a system is as high as its weakest link, which is often the end-user. [Sch00] If the user acts against the system (because of the unawareness), such a system will never be secure enough. Therefore, many organizations concentrated on awareness enhancements at their workplaces. Users where often thought on main security problems and took this knowledge also into their private environment. Such knowledge improved the security of users also at home, but indirectly. There were less initiatives to target the home users directly. [TCF10] But regardless if the users are home users or business users, their awareness has to be improved for mobile phones.
The learning process at school, from advertisements, etc. is done on low level, without true motivation and not continuous. Employees can be taught periodically and an "information security culture" can be developed within the company (and therefore also among the home users, as these employees are in their free time also home users). The employer should provide the basis of security awareness and the users could then improve this knowledge

in their free time. For example if the user is aware of the Phishing-Attack, he is also sensitive to thoughts about this attack in his free time (advertisement, news, friends) and became also more sensitive to other possible risks. The learning process has to be done according to didactic knowledge of learning processes. The employees can not be simple forced to read some policies in their free time, the learning process has to satisfy their attitude to these problems and choose the appropriate learning method.[TCF10]
It does not mean, that automatically each unemployed person has the lowest security knowledge and each business user is the most aware one. Every user group has it's unique constraints and therefore also the according didactic procedures have to be chosen. The way, how these users will be targeted in such awareness improvements can be drawn from their role, i.e. the improvements for business users can be adapted on the business security policies, the home users can be reached by advertisements, etc.
One current survey [KS10], where 304 young people aged between 18 and 25 were questioned, showed that awareness of security threats is not enough. These young undergraduate students grew up with the internet and mobile devices and one could therefore claim, that these people are more aware of security threats connected with mobile devices.
24% of the respondents where not aware of privacy connected with usage of mobile devices which means, that 3/4 of them where aware of security threats in general. Further they where aware of the possibility to lock their phones with a PIN, 80% of them did not use this primitive authentication mechanism, which is more as twice lower than by the respondents (not concentrated on young undergraduate students) surveyed by [CF05] in 2005. The authors propose two possible answers to the question concerning why these young people are not utilizing security countermeasures (as for example the mentioned usage of the simplest way of authentication) on their mobile devices. These respondents where generally aware of security problems like Worms, Trojan Horses or Virus, but did not have the necessary technical skills and knowledge (50% of respondents thought their technical knowledge is low or low to intermediate) to really understand how the stored information can be misused to their disadvantage. This argumentation is also supported by the fact, that questioned students rated the problem of loosing their devices (via theft, or forgotten) as the most likely cause for loosing their private data, and not other, more sophisticated security attacks without physical contact. Another reason of such behavior could be the willingness to expose private data to friends via todays possibilities (for example on social sites). Todays young people are obviously not aware of the problem of identity theft, since they publish many of their private data. [KS10]

6.4.2 Improving Security Awareness

Employers should be motivated to have security aware employees, because they work with company's confidential data. Organizations should be motivated therefore to spent money on teaching their employees the security principles and improve their awareness.

The goal is to improve the security awareness, employers should therefore think about motivation of the employees to learn also in their free time. An example of a motivation process could look like the following: "You can get the newest iPhone for your business and private concerns, but you have to pass these certificates in your free time. You are not allowed to use any other mobile device for the combination of private and business concerns, only the for us adopted and certified iPhone model."

Employers are also motivated to have always reachable employees and maybe are also motivated to bring these company's mobile devices to their daily life beyond the work time. If the employer can ensure, that all his employees have a base knowledge of security principles and are aware of connected problems, such employees are more sensitive to future possible threats. They will possibly learn new threats, maybe also only for home utilization, because they now understand the problem.

Since the awareness is a global problem of security threats in general, it's connection with any of the classified attacks would be possible.

6.4.3 Necessity of Private Data Protection

As discussed in Section 6.4.1, many of the todays young people do not understand how their private data can be misused. Many of them publish their current social status with minute updates on wide spread social sites. Therefore the above mentioned awareness of security threats is not enough, it must be connected with the problem of private data misuse. If young people are not aware of the problems connected with publishing private data, they will not be aware of these threats and might not be able to protect them against these attacks.

Same as above, this improvement is connected with all classified groups of attacks, since it would improve the awareness of the private data protection.

6.4.4 Remote Medical Treatment

The authors of [VBSP08] present a concrete implementation of a framework for remote medical treatment. From the security viewpoint, they only concentrate on the wireless communication and make use of WLAN encryption. They consider the Bluetooth transmission as secure and for communication over IP, they suggest the use of VPN tunnel with IPsec encryption (Internet Protocol Security, a protocol for securing the IP by authenticating

and encrypting each packet). The remote medical authority (e.g. the doctor) has to authenticate his actions in the remote system. For initiation of the communication process of the doctor with the patient (through the remote medical server), a private-key based encryption is used, but then a shared-key is transferred, because the encryption/decryption process would consume too many resources on the doctor's mobile device. The private patients health data is transferred via TCP/IP with SSL encryption.
Such an approach is just utilizing a combination of todays known countermeasures, but since the health of a patient could be harmed by a successful attack, such frameworks should be much more evaluated and higher security mechanisms should be deployed.

6.5 The Proposed Mobile Attack Surface

The proposed mobile phone attack surface classification for mobile devices in Section 5.1 should be discussed and evaluated. The classification may not be complete, or some overlapping may be present. This classification should be a base for future classifications and should underline the problem of such classifications of device threats and their environment, which are more and more interconnected with other complex systems and in the future with many cluster services.
Also interesting for future work would be the inspection and analysis of consequences and effects of security threats from each group of the proposed classification. For this purpose, the todays state-of-the-art of security countermeasures of each class have to be inspected, possible other security attacks discovered and their impact on mobile devices would have to be noted and evaluated.
Such future work connected with the afterwards prioritization of these possible classified attacks and security leaks could bring a very important security improvement for mobile devices. With such works, also the security awareness in the field of mobile device security could be enhanced. The priorities could then motivate the researchers to search for possible countermeasures and security improvements. Such future work would underline the problem also discussed in this thesis, that mobile devices suffer from lack of design for security, which could produce some disasters in the future when not considered now.

6.6 Future of Mobile Devices

One of the future development of mobile devices is their ongoing possibility to replace PCs including laptops. Todays applications, which can be installed onto mobile devices, are a sign for the future replacement of PCs in

many daily life situations. [BD10]
Another future development is the interconnectivity and the connected devices. Two devices connected via relatively slow Bluetooth will lose their importance in the future. Mobile devices, which will be connected with huge servers providing many services to them, allowing the mobile devices to connect private and business information, knowing the personal needs and preferences, predicting the future move of the mobile device user - all these carried out by using a fast data connection. Maybe a PC-like cluster can be formed with future mobile devices to allow extra resources to be leased. [BD10]
Social networking became a hype these days, but with the evolution of mobile devices, these connections of personal data with automated solutions will change the way we use social networking today. Changing our status on Facebook will be no more necessary, because this happens automatically, as the location aware mobile device will infer the social status by utilizing available sensors. Services based on social status will emerge and the "Minority Report-style marketing will explode." [BD10]

Chapter 7

Conclusion

Mobile phones are a unique kind of daily used devices. Many HW interfaces are included to allow many possibilities of interconnection and intercommunication. Since mobile phones are on the way to become personal communication devices and entry points to daily used systems, also their usage differs from other, in functionality similar, devices like PCs.

Operating systems of mobile phones are also analogical to OSs of personal computers, but have unique considerations, mainly based on resources limitation emerging from limited power supply. Not all operating systems were designed for the todays interconnected world and some crucial security concepts are missing. But the structure of these devices from the SW point of view are akin to other devices: HW interfaces have their drivers and libraries utilized by the OS. The functionality can be extended with installed applications. In contrast to PCs , it is not simple to embed other HW components into a mobile phone. This is one of the main reasons, why there is a remarkable motivation for development of new applications. But also the development of embedded HW interfaces is therefore much faster than in other devices.

The protection goals and possible attackers were already identified based on current use cases (see also the world of PCs), therefore the focus should be now on the security improvements. Also many possible and now in development use cases of mobile usage were documented by the researches. According to new trends, many developers are concentrated on development of new applications for mobile phones, without limitations to one operating system. Many new possible use case scenarios will be developed and this also underlines the necessity of robust security models.

Complete classification of all possible threats should be focused in the research, since this would build up a remarkable base for systematical research on security improvement of mobile phones. Categorization into Communication Interfaces, Operating System Vulnerabilities, Installed Applications and Mobility is surely only the start of a possible classification. But thanks to many documented examples of attacks on mobile phones, there are enough resources to develop a really robust classification as future work.

Mobile devices have their uniqueness in the difference between mobile and fixed environments. These remarkable discrepancy has to be considered while proposing possibilities to improve the security of them. Since beside the device, also a human being is involved in these new emerging systems, the human factor has to be considered too. Thanks to clearly defined design principles for secure systems in general, the main goals are already set. The only necessary adaption is the way how these design principles will be moved to reality. Also some security models were already proposed and can be utilized as a base for future work. The current research in the field of mobile phones security improvement already started. Key factors, that have to be considered next, are amongst others: off device security services (cloud), virtualization, secure installation of new applications.

But there are also many open questions and problems while following the goal to improve mobile phones security. Virtualization, isolation, awareness improvement, necessity of data protection and sensor security and many others are identified and discussed in this thesis as the most important one. Similar to the approach presented here, a prioritized list of open problems could be proposed in some future work, based on a complete classification of possible threats. This would be a remarkable guideline for security improvement of mobile phones.

The current security level of mobile phones is not satisfactory. If, for any reason, a group of attackers would now try to attack as many as possible mobile phones, the mankind would have a very serious problem. Denial-of-Service in mobile phones and their networks could paralyze the world. Since the now developed mobile phones are evolving to mobile devices, combining many before separate devices, the aim should be a systematical approach, like the one proposed in this thesis, to improve the overall security of today's mobile phones.

This thesis is only an introductory research into the topic of possible security improvements of mobile phones. First of all, some future work on the proposed security threat classification is necessary, since it should be a fundament for any discussion of this topic. Then, the proposed security improvements, based on the mobile phone attack surface classification, should be developed and evaluated. For both of these, the current security level from other fields should be adapted.

Chapter 8

Terms and Abbreviations

Glossary

3G A family of standards for mobile telecommunications 18

A-GPS Assisted GPS, the improvement lies in utilizing other positioning information like the used cell tower by cellular phones 11, 38

BitTorrent A peer-to-peer file sharing protocol 48

Bluetooth This is an open technology standard for wireless transfer of data over short distances 6, 12, 13, 18, 19, 29, 41–44, 52, 80, 82

Bonjour Is a service discovery protocol for ”zero configuration”: devices not yet connected to each other discovering the possible services of each other 13, 14

BSD sockets The BSD socket interface is the most popular method of TCP/IP programming, when the network endpoints, which is the IP address combined with the port number, are represented as sockets. [GN98] 13, 14, 17

CPU Central Processing Unit, also known as ”processor” 14, 18, 54, 61, 89

CSD Circuit Switched Data, the original data transmission protocol developed for GSM 12, 88

DLL Dynamic Link Library, shared libraries that make some functionality available for the caller 18, 72

DNS Domain Name Service, a naming service for participants of a network like Internet or any other private network 14

DoS Denial of Service attack, where available resources of a system are attacked with the result that this available service is for some time no more accessible 60

EDGE Enhanced Data rates for GSM Evolution, a mobile phone technology with improved data transmission rates 18, 41

firmware As in the world of PCs the BIOS, in the world of mobile devices, this is the part between the OS and the mobile device hardware. This part of SW is executed upon startup of the device. 12, 89

FTP File Transfer Protocol used for transferring data from one host to another 13

GPRS General Packet Radio Service, another packet oriented mobile transmission protocol 12, 41, 60

GPS Global Positioning System, a satellite navigation system for establishing the global position on earth 11, 15, 18, 37, 38, 41, 49, 51, 52, 69, 87

GSM Global System for Mobile Communications, the most used standard for mobile telephony 18, 41, 50, 51, 60, 87

HMAC Hash-based Message Authentication Code combines a hash-value and a secret key for message authentication codes 14

HSCSD High-Speed Circuit-Switched Data, an enhancement to CSD 12

HTML HyperText Markup Language, dominant markup language of web pages 15

HTTP Hypertext Transfer Protocol is a standard request-response based protocol for client-server communication 13, 88

HTTPS Hypertext Transfer Protocol Secure, a combination of the HTTP and SSL protocols to support the secured authentication on a web server and the ongoing encrypted communication 13

HW Abbreviation for hardware 8, 11, 14, 16, 20, 38, 41, 43, 52, 56, 69, 74, 84, 89

I/O Abbreviation for Input/Output 14, 18

IMAP Internet Message Access Protocol, a standard internet protocol for mail retrieval 47

IP Internet Protocol, a packed-switched transmission protocol based on unique addresses of communicating endpoints 12, 13, 31, 41, 42, 80, 87, 89

iPhone unlock a hacking procedure, where any restrictions implemented by the iPhone developer Apple are removed via a SW change 77

IPsec Internet Protocol Security, a protocol for securing the IP by authenticating and encrypting each packet 80

IT Abbreviation for Information Technology 27

kernel Central component of each OS architecture establishing the communication of the application part of the OS and the low-level part of the OS like CPU, memory and HW parts 14, 16–18, 50, 73, 74

Mach Operating system microkernel developed at Carnegie Mellon University 14

MMS Multimedia Messaging Service, improvement of SMS to be able to send also multimedia content to and from a mobile phone 29, 42, 50, 60

mobile device hardware The hardware part of the mobile device. This term is used as an abstraction in contrast to the mobile device software (focus of this thesis) and include all HW components of a device. 12, 88, 89

mobile device software The software part of the mobile device consisting of the OS, firmware and installed applications. This term is used as an abstraction in contrast to the mobile device hardware (not focused in this thesis). 12, 89

OpenSSL The open source implementation of SSL and TLS 14

OS As in other devices like PCs, mobile devices has their own operating systems. 8, 11–16, 18–22, 28, 29, 32, 41–43, 46–48, 50, 52, 54, 59, 61, 63–65, 68–72, 74, 75, 77, 84, 88, 89

PC Abbreviation for the term Personal Computer, which includes Desktop PCs and Laptop PCs. 6, 7, 19, 22, 24, 28, 29, 36, 38, 39, 42, 43, 45, 46, 50, 54, 56, 63–66, 69, 70, 73, 81, 82, 84, 88, 89

PIN Personal Identification Number, a secret number shared between a user and a system to gain access to secured content 28, 32, 33, 45, 65, 79

PKI Public Key Infrastructure, a very complex paradigm for assigning keys to concrete subjects to enable encrypted information transfer 44

POSIX threads "Portable Operating System Interface for Unix" standard for threads 14

SIM Subscriber Identification Module, a removable card for storing the identification of a mobile network subscriber to be independent from the mobile phone device 44, 70

SMOSM Simplified Mobile Operating System Model, discussed in Section 2.3 19, 20

SMS Short Message Service allows exchange of short text messages between mobile phone devices 6, 18, 19, 29, 36, 42, 45, 50, 51, 60–62, 66, 89

SQL Standard Query Language, a standard database language 15

SSL Secure Socket Layer is a cryptographic protocol for secure communication 13, 14, 81, 88–90

SW Abbreviation for the term software, meaning a kind of executable program code 8, 11, 22, 43, 75, 77, 84, 88

TCP/IP Internet Protocol Suite, a set of protocols used for communication over internet and similar network, consisting of two main protocols: Transmission Control Protocol and Internet Protocol 19, 81, 87

TLS Transport Layer Security is a successor of SSL 13, 14, 89

UI Abbreviation for User Interface 15, 16, 18, 19, 73

UMLsec A security extension in form of a profile to the Unified Modelling Language, a language for modelling specifications in a graphical way 59

UNIX Operating system developed originally at Bell Labs 14

URL Uniform Resource Locator specifies the address of a resource 15

USB Universal Serial Bus, a bus for connecting a host with a device 19

VM Virtual Machine, a virtual implementation of a machine that acts like a physical machine 17, 70

VPN Virtual Private Network, keeping private a conversation between two participants over a public network like internet 39, 46, 80

WebKit Some core implementation of classes, that allow the main browser functionality, as displaying homepages in windows, following links, etc. 17

WLAN Abbreviation for Wireless Local Area Network, a wireless data distribution method between two or more devices connected in one network 12, 15, 18, 37, 38, 41, 42, 52, 80

XML Extensible Markup Language, rules for encoding of documents 15

Chapter 9

List of Tables

List of Tables

Chapter 10

List of Figures

List of Figures

Chapter 11

Bibliography

Bibliography

[AAB10] Sarah Al-Azzani and Rami Bahsoon. Using implied scenarios in security testing. In *Proceedings of the 2010 ICSE Workshop on Software Engineering for Secure Systems*, SESS '10, pages 15–21, New York, NY, USA, 2010. ACM.

[App09] Apple. *iPhone OS Technology Overview.* Apple Inc., 2009-10-19 edition, 10 2009.

[BA10] Mark Bedner and Tobias Ackermann. Schutzziele der it-sicherheit. *Datenschutz und Datensicherheit (DuD)*, 34(5):323–328, 2010.

[Bab07] Steve Babin. *Developing Software for Symbian OS 2nd Edition: A Beginner's Guide to Creating Symbian OS v9 Smartphone Applications in C++.* Wiley Publishing, 2007.

[BC04] Paola Bracchi and Vittorio Cortellessa. A framework to model and analyze the performability of mobile software systems. In *Proceedings of the 4th international workshop on Software and performance*, WOSP '04, pages 243–248, New York, NY, USA, 2004. ACM.

[BD10] John Brandon and Digitaltrends. The future of smartphones: 2010-2015 and beyond, February 2010.

[BIM+09] Oliver Baecker, Tobias Ippisch, Florian Michahelles, Sascha Roth, and Elgar Fleisch. Mobile claims assistance. In *MUM '09: Proceedings of the 8th International Conference on Mobile and Ubiquitous Multimedia*, pages 1–9, New York, NY, USA, 2009. ACM.

[Bla08] Clive Blackwell. A multi-layered security architecture for modelling complex systems. In *Proceedings of the 4th annual workshop on Cyber security and information intelligence research: developing strategies to meet the cyber security and information intelligence challenges ahead*, CSIIRW '08, pages 35:1–35:4, New York, NY, USA, 2008. ACM.

[BOB$^+$10] Jeffrey Bickford, Ryan O'Hare, Arati Baliga, Vinod Ganapathy, and Liviu Iftode. Rootkits on smart phones: attacks, implications and opportunities. In *HotMobile '10: Proceedings of the Eleventh Workshop on Mobile Computing Systems &#38; Applications*, pages 49–54, New York, NY, USA, 2010. ACM.

[CDG05] Thibault Candebat, Cameron Ross Dunne, and David T. Gray. Pseudonym management using mediated identity-based cryptography. In *Proceedings of the 2005 workshop on Digital identity management*, DIM '05, pages 1–10, New York, NY, USA, 2005. ACM.

[CF05] N.L. Clarke and S.M. Furnell. Authentication of users on mobile telephones - a survey of attitudes and practices. *Computers and Security*, 24(7):519 – 527, 2005.

[CJ07] Yun Chan Cho and Jae Wook Jeon. Current software platforms on mobile phone. In *Control, Automation and Systems, 2007. ICCAS '07. International Conference on*, pages 1862 –1867, oct. 2007.

[CKK$^+$08] Cory Cornelius, Apu Kapadia, David Kotz, Dan Peebles, Minho Shin, and Nikos Triandopoulos. Anonysense: privacy-aware people-centric sensing. In *Proceeding of the 6th international conference on Mobile systems, applications, and services*, MobiSys '08, pages 211–224, New York, NY, USA, 2008. ACM.

[CMC09] Liang Cai, Sridhar Machiraju, and Hao Chen. Defending against sensor-sniffing attacks on mobile phones. In *MobiHeld '09: Proceedings of the 1st ACM workshop on Networking, systems, and applications for mobile handhelds*, pages 31–36, New York, NY, USA, 2009. ACM.

[CMP09] Iolanthe Chronis, Anmol Madan, and Alex (Sandy) Pentland. Socialcircuits: the art of using mobile phones for modeling personal interactions. In *ICMI-MLMI '09: Proceedings of the ICMI-MLMI '09 Workshop on Multimodal Sensor-Based Systems and Mobile Phones for Social Computing*, pages 1–4, New York, NY, USA, 2009. ACM.

[CVV$^+$10] Gayathri Chandrasekaran, Tam Vu, Alexander Varshavsky, Marco Gruteser, Richard P. Martin, Jie Yang, and Yingying Chen. Vehicular speed estimation using received signal strength from mobile phones. In *Proceedings of the 12th ACM international conference on Ubiquitous computing*, Ubicomp '10, pages 237–240, New York, NY, USA, 2010. ACM.

[DCG08] Cameron Ross Dunne, Thibault Candebat, and David Gray. A three-party architecture and protocol that supports users with multiple identities for use with location based services. In *Proceedings of the 5th international conference on Pervasive services*, ICPS '08, pages 1–10, New York, NY, USA, 2008. ACM.

[dGL08] R. d'Alessandro, M. Ghirardi, and M. Leone. Sc@cco: a graphic-based authentication system. In *EUROSEC '08: Proceedings of the 1st European Workshop on System Security*, pages 8–15, New York, NY, USA, 2008. ACM.

[EKC08] P. Ekler, I. Kelenyi, and H. Charaf. Bittorrent at mobile phones. In *Consumer Communications and Networking Conference, 2008. CCNC 2008. 5th IEEE*, pages 1214 –1215, jan. 2008.

[Ent09] Mobile Enterprise. Mobile phone security survey indicates most users are unprepared, March 2009.

[EOM09] William Enck, Machigar Ongtang, and Patrick McDaniel. On lightweight mobile phone application certification. In *CCS '09: Proceedings of the 16th ACM conference on Computer and communications security*, pages 235–245, New York, NY, USA, 2009. ACM.

[FDK$^+$09] Jon Froehlich, Tawanna Dillahunt, Predrag Klasnja, Jennifer Mankoff, Sunny Consolvo, Beverly Harrison, and James A. Landay. Ubigreen: investigating a mobile tool for tracking and supporting green transportation habits. In *CHI '09: Proceedings of the 27th international conference on Human factors in computing systems*, pages 1043–1052, New York, NY, USA, 2009. ACM.

[GN98] Ivan Griffin and John Nelson. Linux network programming, part 1. *Linux J.*, page 5, February 1998.

[Goo10] Google. What is android? Android Basics, April 2010.

[GPE06] Rebecca E. Grinter, Leysia Palen, and Margery Eldridge. Chatting with teenagers: Considering the place of chat technologies in teen life. *ACM Trans. Comput.-Hum. Interact.*, 13:423–447, December 2006.

[HA07] Andreas P. Heiner and N. Asokan. Secure software installation in a mobile environment. In *SOUPS '07: Proceedings of the 3rd symposium on Usable privacy and security*, pages 155–156, New York, NY, USA, 2007. ACM.

[Hal04] Benjamin Halpert. Mobile device security. In *InfoSecCD '04: Proceedings of the 1st annual conference on Information security curriculum development*, pages 99–101, New York, NY, USA, 2004. ACM.

[HC05] Jonna Häkkilä and Craig Chatfield. ¡i¿'it's like if you opened someone else's letter'¡/i¿: user perceived privacy and social practices with sms communication. In *Proceedings of the 7th international conference on Human computer interaction with mobile devices &amp; services*, MobileHCI '05, pages 219–222, New York, NY, USA, 2005. ACM.

[Hig10] Kelly Jackson Higgins. Survey: 54 percent of organizations plan to add smartphone antivirus this year, jan. 2010.

[HKL$^+$07] Tuomo Hyyryläinen, Teemu Kärkkäinen, Cheng Luo, Valdas Jaspertas, Jouni Karvo, and Jörg Ott. Opportunistic email distribution and access in challenged heterogeneous environments. In *Proceedings of the second ACM workshop on Challenged networks*, CHANTS '07, pages 97–100, New York, NY, USA, 2007. ACM.

[HL02] Michael Howard and David E. Leblanc. *Writing Secure Code*. Microsoft Press, Redmond, WA, USA, 2nd edition, 2002.

[HMLY05] Ragib Hasan, Suvda Myagmar, Adam J. Lee, and William Yurcik. Toward a threat model for storage systems. In *Proceedings of the 2005 ACM workshop on Storage security and survivability*, StorageSS '05, pages 94–102, New York, NY, USA, 2005. ACM.

[HUJ05] Richard Harper, Axel Unger, and Marcel Jansen. Txtboard: From text-to-person to text-to-home. In *Ext. Abstracts CHI '05, ACM Press*, pages 1705–1708. ACM Press, 2005.

[IHS07] Yoshiro Imai, Yukio Hori, and Yuichi Suigiue. A web-based surveillance system for mobile phones. In *Proceedings of the 1st international conference on MOBILe Wireless MiddleWARE, Operating Systems, and Applications*, MOBILWARE '08, pages 35:1–35:6, ICST, Brussels, Belgium, Belgium, 2007. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).

[IMI10] Sergio Ilarri, Eduardo Mena, and Arantza Illarramendi. Location-dependent query processing: Where we are and where we are heading. *ACM Comput. Surv.*, 42(3):1–73, 2010.

[ITU09] ITU. The world in 2009: Ict facts and figures, 2009.

[Jak09] Andreas Jakl. Symbian os: Overview, possibilities and the community. University of Applied Sciences, Hagenberg, Austria, March 2009.

[JLP10] Wen Jin, Kristen LeFevre, and Jignesh M. Patel. An online framework for publishing privacy-sensitive location traces. In *Proceedings of the Ninth ACM International Workshop on Data Engineering for Wireless and Mobile Access*, MobiDE '10, pages 1–8, New York, NY, USA, 2010. ACM.

[JSB08] Jan Jürjens, Joerg Schreck, and Peter Bartmann. Model-based security analysis for mobile communications. In *ICSE '08: Proceedings of the 30th international conference on Software engineering*, pages 683–692, New York, NY, USA, 2008. ACM.

[KBS09] Amy K. Karlson, A.J. Bernheim Brush, and Stuart Schechter. Can i borrow your phone?: understanding concerns when sharing mobile phones. In *CHI '09: Proceedings of the 27th international conference on Human factors in computing systems*, pages 1647–1650, New York, NY, USA, 2009. ACM.

[KEAR09] Kari Kostiainen, Jan-Erik Ekberg, N. Asokan, and Aarne Rantala. On-board credentials with open provisioning. In *ASIACCS '09: Proceedings of the 4th International Symposium on Information, Computer, and Communications Security*, pages 104–115, New York, NY, USA, 2009. ACM.

[KGE11] Michael Kenteris, Damianos Gavalas, and Daphne Economou. Electronic mobile guides: a survey. *Personal Ubiquitous Comput.*, 15:97–111, January 2011.

[KN09] Imre Kelényi and Jukka K. Nurminen. Bursty content sharing mechanism for energy-limited mobile devices. In *Proceedings of the 4th ACM workshop on Performance monitoring and measurement of heterogeneous wireless and wired networks*, PM2HW2N '09, pages 216–223, New York, NY, USA, 2009. ACM.

[KS10] Stan Kurkovsky and Ewa Syta. Digital natives and mobile phones: A survey of practices and attitudes about privacy and security. In *Technology and Society (ISTAS), 2010 IEEE International Symposium on*, pages 441 –449, 7-9 2010.

[Lan10] Max Landman. Managing smart phone security risks. In *2010 Information Security Curriculum Development Conference*, InfoSecCD '10, pages 145–155, New York, NY, USA, 2010. ACM.

[LBH08] Kevin A. Li, Patrick Baudisch, and Ken Hinckley. Blindsight: eyes-free access to mobile phones. In *Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, CHI '08, pages 1389–1398, New York, NY, USA, 2008. ACM.

[LCCC09] Chien-Yuan Lai, Hsiu-Sen Chiang, Ching-Chiang Chen, and Shih-Hao Chou. Web service description for mobile phone virus. In *ICIS '09: Proceedings of the 2nd International Conference on Interaction Sciences*, pages 951–956, New York, NY, USA, 2009. ACM.

[LHY99] Chii-Hwa Lee, Min-Shiang Hwang, and Wei-Pang Yang. Enhanced privacy and authentication for the global system for mobile communications. *Wirel. Netw.*, 5:231–243, July 1999.

[LK09] Nena Lim and Anne Khoo. Forensics of computers and handheld devices: identical or fraternal twins? *Commun. ACM*, 52(6):132–135, 2009.

[LPL$^+$09] Hong Lu, Wei Pan, Nicholas D. Lane, Tanzeem Choudhury, and Andrew T. Campbell. Soundsense: scalable sound sensing for people-centric applications on mobile phones. In *MobiSys '09: Proceedings of the 7th international conference on Mobile systems, applications, and services*, pages 165–178, New York, NY, USA, 2009. ACM.

[LSH$^+$09] Yue-Hsun Lin, Ahren Studer, Hsu-Chin Hsiao, Jonathan M. McCune, King-Hang Wang, Maxwell Krohn, Phen-Lan Lin, Adrian Perrig, Hung-Min Sun, and Bo-Yin Yang. Spate: small-group pki-less authenticated trust establishment. In *MobiSys '09: Proceedings of the 7th international conference on Mobile systems, applications, and services*, pages 1–14, New York, NY, USA, 2009. ACM.

[Meta] AdMob Mobile Metrics. Admob publisher survey march 2010.

[Metb] AdMob Mobile Metrics. December 2009 mobile metrics report.

[Metc] AdMob Mobile Metrics. February 2010 mobile metrics report.

[MHM$^+$10] Min Mun, Shuai Hao, Nilesh Mishra, Katie Shilton, Jeff Burke, Deborah Estrin, Mark Hansen, and Ramesh Govindan. Personal data vaults: a locus of control for personal data streams. In *Proceedings of the 6th International COnference*, Co-NEXT '10, pages 17:1–17:12, New York, NY, USA, 2010. ACM.

[MLF$^+$08] Emiliano Miluzzo, Nicholas D. Lane, Kristóf Fodor, Ronald Peterson, Hong Lu, Mirco Musolesi, Shane B. Eisenman, Xiao Zheng, and Andrew T. Campbell. Sensing meets mobile social networks: the design, implementation and evaluation of the cenceme application. In *SenSys '08: Proceedings of the 6th ACM conference on Embedded network sensor systems*, pages 337–350, New York, NY, USA, 2008. ACM.

[MLK$^+$09] Mark Manulis, Damien Leroy, Francois Koeune, Olivier Bonaventure, and Jean-Jacques Quisquater. Authenticated wireless roaming via tunnels: making mobile guests feel at home. In *Proceedings of the 4th International Symposium on Information, Computer, and Communications Security*, ASIACCS '09, pages 92–103, New York, NY, USA, 2009. ACM.

[MM08] Keith E. Mayes and Konstantinos Markantonakis. Mobile communication security controllers an evaluation paper. *Inf. Secur. Tech. Rep.*, 13:173–192, August 2008.

[MM09] Collin Mulliner and Charlie Miller. Injecting sms messages into smart phones for security analysis. In *Proceedings of the 3rd USENIX conference on Offensive technologies*, WOOT'09, pages 5–5, Berkeley, CA, USA, 2009. USENIX Association.

[MSS$^+$08] Divya Muthukumaran, Anuj Sawani, Joshua Schiffman, Brian M. Jung, and Trent Jaeger. Measuring integrity on mobile phone systems. In *SACMAT '08: Proceedings of the 13th ACM symposium on Access control models and technologies*, pages 155–164, New York, NY, USA, 2008. ACM.

[NB99] Peter G. Neumann and Anthony Barnes. Practical architectures for survivable systems and networks: Phase-one final report, 1999.

[NGR09] Priya Narasimhan, Rajeev Gandhi, and Dan Rossi. Smartphone-based assistive technologies for the blind. In *Proceedings of the 2009 international conference on Compilers, architecture, and synthesis for embedded systems*, CASES '09, pages 223–232, New York, NY, USA, 2009. ACM.

[OHKY09] Zhonghong Ou, Erkki Harjula, Otso Kassinen, and Mika Ylianttila. Feasibility evaluation of a communication-oriented p2p system in mobile environments. In *Proceedings of the 6th International Conference on Mobile Technology, Application &#38; Systems*, Mobility '09, pages 43:1–43:8, New York, NY, USA, 2009. ACM.

[OJ10] Jon Oberheide and Farnam Jahanian. When mobile is harder than fixed (and vice versa): demystifying security challenges in mobile environments. In *HotMobile '10: Proceedings of the Eleventh Workshop on Mobile Computing Systems &#38; Applications*, pages 43–48, New York, NY, USA, 2010. ACM.

[OVC$^+$08] Jon Oberheide, Kaushik Veeraraghavan, Evan Cooke, Jason Flinn, and Farnam Jahanian. Virtualized in-cloud security services for mobile devices. In *MobiVirt '08: Proceedings of the First Workshop on Virtualization in Mobile Computing*, pages 31–35, New York, NY, USA, 2008. ACM.

[PGT09] Frank S. Park, Chinmay Gangakhedkar, and Patrick Traynor. Leveraging cellular infrastructure to improve fraud prevention. *Computer Security Applications Conference, Annual*, 0:350–359, 2009.

[POS$^+$01] Mark Perry, Kenton O'hara, Abigail Sellen, Barry Brown, and Richard Harper. Dealing with mobility: understanding access anyti anywhere. *ACM Trans. Comput.-Hum. Interact.*, 8:323–347, December 2001.

[PS10] Michael Paik and Lakshminarayanan Subramanian. Signet: low-cost auditable transactions using sims and mobile phones. *SIGOPS Oper. Syst. Rev.*, 43:73–78, January 2010.

[QLG10] Islam Qudah, Peter Leijdekkers, and Valerie Gay. Using mobile phones to improve medication compliance and awareness for cardiac patients. In *Proceedings of the 3rd International Conference on PErvasive Technologies Related to Assistive Environments*, PETRA '10, pages 36:1–36:7, New York, NY, USA, 2010. ACM.

[RMB$^+$10] Sasank Reddy, Min Mun, Jeff Burke, Deborah Estrin, Mark Hansen, and Mani Srivastava. Using mobile phones to determine transportation modes. *ACM Trans. Sen. Netw.*, 6(2):1–27, 2010.

[RSD$^+$10] Sasank Reddy, Katie Shilton, Gleb Denisov, Christian Cenizal, Deborah Estrin, and Mani Srivastava. Biketastic: sensing and mapping for better biking. In *Proceedings of the 28th international conference on Human factors in computing systems*, CHI '10, pages 1817–1820, New York, NY, USA, 2010. ACM.

[SCCA09] Aubrey-Derrick Schmidt, Jan Hendrik Clausen, Seyit Ahmet Camtepe, and Sahin Albayrak. Detecting symbian os malware through static function call analysis. In *Proceedings of the*

4th IEEE International Conference on Malicious and Unwanted Software (Malware 2009), pages 15–22. IEEE, 2009.

[Sch00] Bruce Schneier. *Secrets and lies : digital security in a networked world / Bruce Schneier.* New York ; Chichester : John Wiley, 2000. Includes index.

[SDLC09] Julian Seifert, Alexander De Luca, and Bettina Conradi. A context-sensitive security model for privacy protection on mobile phones. In *MobileHCI '09: Proceedings of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services*, pages 1–2, New York, NY, USA, 2009. ACM.

[Sea09] SearchSecurityAsia. F-secure survey shows continued lack of security awareness from hk mobile phone users, March 2009.

[Soh08] Timothy Youngjin Sohn. *Addressing the needs of mobile users.* PhD thesis, La Jolla, CA, USA, 2008. Adviser-Griswold, William G.

[SPL$^+$09] Aubrey-Derrick Schmidt, Frank Peters, Florian Lamour, Christian Scheel, Seyit Ahmet Çamtepe, and Sahin Albayrak. Monitoring smartphones for anomaly detection. *Mob. Netw. Appl.*, 14(1):92–106, 2009.

[SRS06] Christine Soriano, Gitesh K. Raikundalia, and Jakub Szajman. Middle-aged users' experience of short message service. In *Proceedings of the 7th Australasian User interface conference - Volume 50*, AUIC '06, pages 109–112, Darlinghurst, Australia, Australia, 2006. Australian Computer Society, Inc.

[SSB$^+$09] A.-D. Schmidt, H.-G. Schmidt, L. Batyuk, J.H. Clausen, S.A. Camtepe, S. Albayrak, and C. Yildizli. Smartphone malware evolution revisited: Android next target? In *Malicious and Unwanted Software (MALWARE), 2009 4th International Conference on*, pages 1 –7, oct. 2009.

[SSG99] Stuart G. Stubblebine, Paul F. Syverson, and David M. Goldschlag. Unlinkable serial transactions: protocols and applications. *ACM Trans. Inf. Syst. Secur.*, 2:354–389, November 1999.

[SSS75] Jerome H. Saltzer, Jerome H. Saltzer, and Michael D. Schroeder. The protection of information in computer systems. 1975.

[Sta09] SC Staff. Confidence in awareness for mobile phone security still at a low ebb, October 2009.

[Sto02] I. Stojmenović. *Handbook of Wireless Networks and Mobile Computing*. Wiley series on parallel and distributed computing. Wiley-Interscience, 2002.

[SXD09] K. Subbu, Ning Xu, and R. Dantu. iknow where you are. In *Computational Science and Engineering, 2009. CSE '09. International Conference on*, volume 4, pages 469 –474, aug. 2009.

[TCF10] S. Talib, N.L. Clarke, and S.M. Furnell. An analysis of information security awareness within home and work environments. In *Availability, Reliability, and Security, 2010. ARES '10 International Conference on*, pages 196 –203, 15-18 2010.

[Tec05] Javvin Technologies. *Network Protocols Handbook*. Javvin Technologies, 2005.

[TLO$^+$09] Patrick Traynor, Michael Lin, Machigar Ongtang, Vikhyath Rao, Trent Jaeger, Patrick McDaniel, and Thomas La Porta. On cellular botnets: measuring the impact of malicious devices on a cellular network core. In *CCS '09: Proceedings of the 16th ACM conference on Computer and communications security*, pages 223–234, New York, NY, USA, 2009. ACM.

[Too08] M. Toorani. Smemail - a new protocol for the secure e-mail in mobile environments. pages 39 –44, dec. 2008.

[VBSP08] D. Vassis, P. Belsis, C. Skourlas, and G. Pantziou. A pervasive architectural framework for providing remote medical treatment. In *PETRA '08: Proceedings of the 1st international conference on PErvasive Technologies Related to Assistive Environments*, pages 1–8, New York, NY, USA, 2008. ACM.

[VCU08] S. Vidyaraman, M. Chandrasekaran, and S. Upadhyaya. Position: the user is the enemy. In *NSPW '07: Proceedings of the 2007 Workshop on New Security Paradigms*, pages 75–80, New York, NY, USA, 2008. ACM.

[VHR06] Deepak Venugopal, Guoning Hu, and Nicoleta Roman. Intelligent virus detection on mobile devices. In *PST '06: Proceedings of the 2006 International Conference on Privacy, Security and Trust*, pages 1–4, New York, NY, USA, 2006. ACM.

[WAB09] J. Whipple, W. Arensman, and M.S. Boler. A public safety application of gps-enabled smartphones and the android operating system. In *Systems, Man and Cybernetics, 2009. SMC 2009. IEEE International Conference on*, pages 2059 –2061, oct. 2009.

[WRP08] Zhiguo Wan, Kui Ren, and Bart Preneel. A secure privacy-preserving roaming protocol based on hierarchical identity-based encryption for mobile networks. In *Proceedings of the first ACM conference on Wireless network security*, WiSec '08, pages 62–67, New York, NY, USA, 2008. ACM.

[XZL$^+$09] Nan Xu, Fan Zhang, Yisha Luo, Weijia Jia, Dong Xuan, and Jin Teng. Stealthy video capturer: a new video-based spyware in 3g smartphones. In *WiSec '09: Proceedings of the second ACM conference on Wireless network security*, pages 69–78, New York, NY, USA, 2009. ACM.

[XZSZ10] Liang Xie, Xinwen Zhang, Jean-Pierre Seifert, and Sencun Zhu. pbmds: a behavior-based malware detection system for cellphone devices. In *WiSec '10: Proceedings of the third ACM conference on Wireless network security*, pages 37–48, New York, NY, USA, 2010. ACM.

[YLH$^+$08] Seehwan Yoo, Yunxin Liu, Cheol-Ho Hong, Chuck Yoo, and Yongguang Zhang. Mobivmm: a virtual machine monitor for mobile phones. In *MobiVirt '08: Proceedings of the First Workshop on Virtualization in Mobile Computing*, pages 1–5, New York, NY, USA, 2008. ACM.

[ZCZ$^+$09] Zhichao Zhu, Guohong Cao, Sencun Zhu, S. Ranjan, and A. Nucci. A social network based patching scheme for worm containment in cellular networks. In *INFOCOM 2009, IEEE*, pages 1476 –1484, april 2009.

[ZKJ$^+$10] Zhenyun Zhuang, Sandeep Kakumanu, Yeonsik Jeong, Raghupathy Sivakumar, and Aravind Velayutham. Mobile hosts participating in peer-to-peer data networks: challenges and solutions. *Wirel. Netw.*, 16:2313–2333, November 2010.

[ZLTG07] Keshu Zhang, Haifeng Li, Kari Torkkola, and Mike Gardner. Adaptive learning of semantic locations and routes. In *Proceedings of the 1st international conference on Autonomic computing and communication systems*, Autonomics '07, pages 3:1–3:10, ICST, Brussels, Belgium, Belgium, 2007. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).