JOHANNES KEPLER
UNIVERSITÄT LINZ | JKU

TNF

Technisch-Naturwissenschaftliche
Fakultät

Sicherheit in Rechenzentren in Bezug auf mobile Geräte – e-Banking mit Android

MASTERARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Masterstudium

NETZWERKE UND SICHERHEIT

Eingereicht von:
Thomas Mayr BSc

Angefertigt am:
Institut für Informationsverarbeitung und Mikroprozessortechnik (FIM)

Beurteilung:
Michael Sonntag, Assoz.Prof. Mag. Dipl.-Ing. Dr.

I. Kurzfassung

Durch die steigende Verbreitung und den erhöhten Leistungsumfang der mobilen Geräte wird auch der Bedarf nach mobilen Anwendungen größer. So ist es wichtig, auch sicherheitskritische Applikationen wie e-Banking für mobile Geräte zu portieren. Ziel dieser Masterarbeit ist es, die Risiken und Gefährdungen für die Anwendung, das System - bestehend aus Gerät, Übertragungsstrecke und Rechenzentrum - und die verarbeiteten Daten unter Android zu ermitteln und zu analysieren.

Seit vielen Jahren existiert die Möglichkeit, die Bankgeschäfte bequem und einfach von zuhause mittels e-Banking vom privaten PC durchzuführen. Die Authentifizierung erfolgt mittels PIN, die Autorisierung zur Überweisung mittels TAN. Die Kommunikation wird verschlüsselt mit Hilfe von SSL/TLS oder MBS/IP durchgeführt.

Es stellen zwar einige österreichische Banken bereits eine e-Banking-Applikation für Android bereit, oft handelt es sich dabei jedoch lediglich um eine Portierung des Webportals für den Computer. Es sind in vielen Fällen keine zusätzlichen Android-spezifischen Sicherheitsmaßnahmen implementiert. Im Zuge dieser Masterarbeit wurden die existierenden Lösungen analysiert und verglichen. Basierend darauf wurde ein Entwurf und ein Prototyp für ein e-Banking System entworfen, das einerseits dem Benutzer die gewünschten Funktionen einfach und komfortabel bereitstellt und andererseits durch geeignete Sicherheitsmechanismen die Sicherheit der Daten, des Gerätes und des Rechenzentrums gewährleistet.

II. Abstract

The number and power of mobile devices is rapidly rising. As a result the need for new mobile applications is increasing, too. Therefore it is important to convert high security services like e-banking to Android. The target of this master thesis is to determine and analyse the risks and dangers related to the application, the system - consisting of the device, the network and the servers - und the data processed by the application..

The possibility to do banking from the PC at home has been available for many years. The authentication is done by the PIN and the TAN. The communication is encrypted by TLS or MBS/IP.

Currently there are applications of e-banking with Android provided by many Austrian banks. In most cases the APPs are only a converted version of the original web portal used at a computer. There are often no additional security features implemented. The existing systems are evaluated and compared within this thesis. A prototyp was designed based on the results of this evaluation. The main focus was on the functions needed by the user and the security. Maximum security must be provided for the data and the devices used during the execution of the application while offering the user a user friendly access to the system. A prototyp of this system shows how it could work.

III. Inhaltsverzeichnis

1. Aufgabenstellung

Ziel dieser Masterarbeit ist es, die Sicherheit von e-Banking auf mobilen Android Geräten zu evaluieren und diese Ergebnisse in Form eines Prototypen zu veranschaulichen und testen. Bei der Evaluierung ist auch ein Vergleich der bestehenden e-Banking APPs zu erstellen.

Im ersten Schritt sind mögliche Sicherheitsanforderungen an die Anwendung „e-Banking“ zu ermitteln und evaluieren. Risiken, die durch die Verwendung dieser Anwendung auf einem Smartphone oder Tablet auftreten, sollen gesammelt und analysiert werden. Herauszuarbeiten sind die Positionen der Schwachstellen und die Bedrohungen, deren sie ausgesetzt sind. (Risikoanalyse und Risikoeinschätzung)

Im nächsten Schritt wird ermittelt, wie derzeit Banktransaktionen auf einem mobilen Gerät durchgeführt werden können und wie diese Möglichkeiten bereitgestellt werden. Auch ist zu ermitteln, welche der im ersten Schritt ermittelten Risiken auf die einzelnen Möglichkeiten zutreffen und wie groß deren Bedeutung ist (Analyse und Beschreibung des Ist-Status).

Nach diesen Analysen und Evaluierungen sollen für die Bedrohungen und Schwachstellen, die im ersten Schritt gefunden wurden, passende Lösungen aufgezeigt werden, die das Risiko eines Informationsverlustes oder eines Schadens für das Endgerät oder das Rechenzentrum minimieren, verhindern oder in andere Bereiche verschieben. Die Lösungsansätze und Lösungen sind zu dokumentieren und begründen. Auch müssen sie auf die Anwendbarkeit und Sinnhaftigkeit überprüft werden. (Risikoabschätzung und Risikoeindämmung)

Nach den Risikobetrachtungen und Analysen ist ein System für die Anwendung „e-banking“ zu entwerfen, das möglichst „sicher“ ist. Wichtige Faktoren dabei sind auch die Umsetzbarkeit und Kundenfreundlichkeit. Es handelt sich dabei um eine theoretische Überlegung, die den gesamten Kommunikationsweg vom Endgerät des Kunden bis zu den Servern im Rechenzentrum abdecken soll. (Systementwurf)

Dieser Entwurf wird im nächsten Schritt mit den bestehenden Systemen verglichen. Die Unterschiede sowie die Vor- und Nachteile der Systeme sollen herausgearbeitet und dokumentiert werden. Gründe für die Verwendung bestimmter Systeme in einzelnen Bereichen sollen basierend auf den Unterschieden gefunden und angeführt werden. (Soll-Ist Vergleich)

Ein Prototyp soll entwickelt werden, der die grundlegenden Funktionen und Sicherheitsmechanismen veranschaulicht und es ermöglicht, Analysen bezüglich Umsetzbarkeit und Benutzerfreundlichkeit des Entwurfes durchzuführen.

2. Begriffsdefinitionen

In der Masterarbeit werden unterschiedliche Begriffe verwendet. Diese werden in folgender Tabelle beschrieben.

Begriff	Definition
Android	Android ist ein Betriebssystem von Google, das für Smartphones verwendet wird. Das System wurde 2005 von Google gekauft und seither als Open Source Projekt entwickelt. Die erste Version wurde 2008 vorgestellt. Aktuell ist die Version 4.0.3 (Stand Jänner 2012).
DalvikVM	Die DalvikVM ist die virtuelle Maschine, in der Programme in Android ausgeführt werden. Sie verwendet die Programmiersprache Java. Im Gegensatz zur normalen JavaVM ist sie jedoch als Registermaschine implementiert.
VM	Die Abkürzung VM steht für "virtuelle Maschine". Sie stellt eine Arbeitsumgebung für Programme dar. Den Programmen, die auf der VM laufen, ist es normalerweise nicht möglich, auf externe Ressourcen zuzugreifen. Der Zugriff auf diese muss von der VM explizit gewährt und in Form von Schnittstellen bereitgestellt werden.
Kellerautomat, Stackmachine	Kellerautomat bzw Stackmachine steht für eine Architektur, die bei der Verarbeitung von Variablen diese zuerst auf einen Stack lädt und bei der Ausführung der Befehle den Stack abarbeitet. Ein Kellerautomat ist einfach umzusetzen und wird zB von der JavaVM verwendet.
Registermaschine	Im Gegensatz zum Kellerautomat werden bei einer Registermaschine "Register" für die Speicherung der Variablen verwendet. Die Register sind sehr nahe der CPU angeordnet und haben somit sehr kurze Zugriffszeiten. Ein Großteil der modernen Prozessoren ist als Registermaschinen ausgelegt. Ein weiteres Beispiel dafür ist die DalvikVM. Am Computer erfolgt die Umwandlung von einem stackbasierten Programm in ein registerbasiertes während der Ausführung.
APP	APP ist die Abkürzung für Applikation im Smartphone-Umfeld. Eine APP ist in den meisten Fällen ein eigenständiges Programm, das ohne einen Browser oder ein sonstiges Programm ausgeführt werden kann.

Webapplikation	Als Webapplikation wird in dieser Arbeit eine Website bezeichnet, die wie eine eigene Anwendung verwendet wird, jedoch für die Ausführung einen Browser benötigt.
Smartphone	Ein Smartphone ist die Kombination eines Computers mit einem Handy. Es ist klein, handlich und transportfähig, besitzt jedoch einen Funktionsumfang, der einem Computer ähnlich ist. Die Basisfunktionalität kann durch Installation von zusätzlicher Software durch den Endanwender erweitert werden. In dieser Arbeit werden Handys und Tablets als Smartphones gleichwertig angesehen. Es wird davon ausgegangen, dass das Gerät sowohl über WLAN als auch über einen Mobilfunkanbieter Zugang zum Internet hat. Ebenso wird die Verfügbarkeit eines GPS Moduls angenommen, da dieses in den meisten Geräten verfügbar ist.
rooting	Als "rooting" wird das Entsperren eines Smartphones bezeichnet. Dabei wird eine Custom-Firmware auf dem Gerät installiert. Der User erhält dadurch Vollzugriff auf das Gerät und kann als Superuser (Root) auf alle Daten des Gerätes und der Speicherkarte zugreifen, diese lesen, verändern und löschen.
Custom-Firmware	Eine Custom-Firmware bezeichnet eine Version eines Betriebssystems, die von der Community, also Anwendern, modifiziert wurde, um Zugriff auf zusätzliche Funktionen und Möglichkeiten zu erhalten. Meistens basiert sie auf den originalen Sourcen des Basisbetriebssystem.
Root	Als "Root" wird im Linux-Umfeld der Superuser bezeichnet. Er hat Zugriff auf alle Funktionen und darf alle Einstellungen ändern.
Hotspot	Im Telekommunikationsbereich spricht man dann von einem Hotspot, wenn WLAN kostenlos angeboten wird. Diese sind oft auf Bahnhöfen und Flughäfen, aber auch in Cafes und Bars zu finden. Sie besitzen meist keine Verschlüsselung.
ELBA	ELBA steht für Electronic Banking und ist eine Anwendung für e-Banking, die von der RACON Linz GmbH entwickelt wurde. Sie wird von mehreren österreichischen Banken für Online Banking eingesetzt.

DoS (Denial of Service), DDoS (distributed DoS)	Ziel eines DoS Angriffs ist das Lahmlegen von Services eines bestimmten Hosts (Servers). Dies geschieht meist durch Überlasten der Infrastruktur durch Senden einer sehr hohe Anzahl von Anfragen, die vom Host verarbeitet werden. Geschieht der Angriff von mehreren Rechnern aus, die dynamisch wechseln und sich von unterschiedlichen Orten verbinden, spricht man von einem distributed DoS Angriff (DDoS). Ein DoS Angriff ist schwer als ein solcher zu erkennen und abzuwehren.
PIN	Die PIN (Persönliche IdentifikationsNummer) wird von verschiedenen Anwendungen wie ein Passwort verwendet. Der Vorteil ist, dass sie nur numerische Zeichen beinhaltet und somit auch auf Nummerntastaturen eingegeben werden kann.
TAN	Die TAN (Transaktionsnummer) ist ein Einmalpasswort und wird bei e-Banking verwendet, um das Senden von Überweisungsaufträgen zu bestätigen. Für die Ermittlung der TAN gibt es unterschiedliche Wege (TAN-Liste, mTAN, cardTAN und Varianten). Welche Methoden verfügbar sind, sind von der Bank und dem Konto abhängig.
Digitale Signatur	Eine digitale Signatur wird verwendet, um die Authentizität und Integrität eines Dokumentes bzw einer Nachricht zu bestätigen. Sie verwendet den privaten Schlüssel des Absenders und verschlüsselt damit den Hashwert der Nachricht. Ein Empfänger kann mit dem öffentlichen Schlüssel die Signatur entschlüsseln und den Hashwert mit dem selber generierten Wert vergleichen. Stimmen diese überein, ist Authentizität und Integrität des Dokumentes bestätigt.
Zertifikat	Ein Zertifikat bestätigt die Verbindung des öffentlichen und privaten Schlüssels zum Namen eines Servers. Damit wird die Authentizität eines Servers, Services oder Benutzers bestätigt. Zertifikate werden von einer CA verteilt und verwaltet. Sie sind eine Voraussetzung für kontrollierbare und sichere Digitale Signaturen. Außerdem wird ein Zertifikat benötigt, um eine sichere SSL/TLS Verbindung aufbauen zu können. (vgl Seite 49, 7.1.3.5 HTTPS)
CA	CA ist die Abkürzung für Certificate Authority. Sie ist zuständig für die Ausstellung, die Kontrolle und den Rückruf von Zertifikaten.

SSL, TLS	SSL (Secure Socket Layer) ist ein Protokoll zur sichern verschlüsselten Übertragung von Daten in einem IP Netz. Es ist der Schicht 6 des ISO OSI Modells zuzuordnen, kann von höheren Protokollen (zB HTTPS) verwendet werden und ist somit system- und anwendungsunabhängig. Ab der Version 3 wird für SSL der Name TLS (Transport Layer Security) verwendet.
Lastverteiler, Loadbalancer	Lastverteiler sind entweder als Soft- oder als Hardware ausgeführt. Sie dienen der Verteilung von Last auf mehrere Server. Abhängig von der Konfiguration wirkt das System wie ein einzelner Server, da dieser nur mit dem Lastverteiler kommuniziert. Letzterer leitet die Anfragen zum entsprechenden Server weiter.
SSL Offloader	Bei einem SSL Offloader handelt es sich um ein Gerät, das in eine verschlüsselte SSL Kommunikation gehängt wird und diese „vorzeitig“ entschlüsselt. Dadurch kann einerseits die Last am Server reduziert werden, da dieser die Datenpakete nicht mehr entschlüsseln muss, und andererseits kann gewährleistet werden, dass die Firewalls und anderen Sicherheitsmechanismen die Pakete beim Eintritt in das interne Netzwerk überprüfen können. Häufig sind SSL Offloader direkt in Firewalls integriert.
NFC	NFC (Near Field Communication) ist ein Kommunikationsstandard für Übertragungen von Daten bei einer Übertragungsstrecke bis 4 cm. Derzeit wird NFC hauptsächlich in Deutschland für kleine Zahlungen mit der EC Karte verwendet. Zurzeit gibt es nur eine begrenzte Anzahl an mobilen Geräten, die NFC unterstützen. NFC basiert auf der Übertragung mittels RFID. Problematisch ist der Einsatz am Smartphone, weil die Sicherheit für jede Softwareversion und für jedes Gerät separat durch die Regeln der Common Criteria geprüft und zertifiziert werden muss.
STUZZA	Die STUZZA (Studiengesellschaft für Zusammenarbeit um Zahlungsverkehr) ist eine Kooperationsplatform von österreichischen Banken, zur Standardisierung und Vereinheitlichung des Zahlungsverkehrs in Österreich.

3. Beschreibung der Komponenten Android, e-Banking

Im folgenden Kapitel wird auf die Systemkomponenten näher eingegangen, die im Laufe der Arbeit abgesichert werden sollen. In dieser Arbeit wird nur Android als Betriebssystem für ein mobiles Gerät behandelt. Die Systeme IOS von Apple und Windows Mobile von Microsoft sowie RIM und Symbian werden hier nicht betrachtet.

3.1 Android

Android ist das Betriebssystem, das von Google weiterentwickelt wird. Es ist ein Open Source Projekt unter der Apache 2.0 Lizenz. [33] Nur der Linux Kernel ist unter der GPLv2 Lizenz lizenziert. Android beinhaltet zwar einen Linux Kernel, jedoch verwendet es andere Programm-bibliotheken und Treiber als eine Standard Linux Distribution, um auf die Hardware zuzugreifen und die Anwendungen auszuführen. Aufgrund dieser Unterschiede ist es nicht möglich, Linux Programme ohne Portierung auf Android laufen zu lassen.

Programme für Android werden in Java geschrieben. Da für die Ausführung die sogenannte DalvikVM verwendet wird, müssen die Programme vorher speziell für diese Architektur kompiliert und gepackt werden.

3.1.1 DalvikVM

Im Unterschied zur normalen JavaVM (Kellerautomat, Stack-Maschine) ist die DalvikVM als Register-Maschine implementiert. Außerdem wird ein anderer Befehlssatz verwendet. Die üblichen Java Befehle haben eine Länge von 8 bit. Das Laden von und zum Stack sind separate Befehle. Bei der DalvikVM sind die Befehle 16 bit lang und bearbeiten direkt die lokalen Variablen. [2] Dies wird auch durch die Verwendung der Register anstatt des Stacks ermöglicht.

Javaprogramme können nur mit einem passenden Interpreter direkt ausgeführt werden. Dieser ist jedoch nicht für Android vorgesehen. Sie müssen vorher für die DalvikVM konvertiert werden. Dabei wird aus dem stackbasierten ein registerbasiertes Programm erstellt. Da diese Umwandlung während der Kompilierung ausgeführt wird, muss sie nicht mehr während der Ausführungszeit durchgeführt werden, wie bei Javaprogrammen am PC in der JavaVM. Dadurch benötigt zwar die Kompilierung länger und ist aufwändiger, jedoch kann das Programm auch auf leistungsschwächerer Hardware eingesetzt werden.

Anwendungen, die unter Android ausgeführt werden, laufen in jeweils einer eigenen sogenannten Sandbox. Diese wiederum wird in einer virtuellen Umgebung (virtuelle Maschine, VM) berechnet. Um auf externe Ressourcen oder auf Daten von anderen Anwendungen zugreifen zu können,

werden Manager benötigt. Ein Bespiel ist der „Content Provider“, der es ermöglicht, auf explizit freigegebene Daten anderer APPs zuzugreifen.[3] In manchen Fällen ist es jedoch erforderlich, dass das Programm noch weitere Berechtigungen erhält. Dafür muss die Anwendung die Umgebung der VM verlassen und direkt im System nativer Code ausgeführt werden. Diese Funktionalität ist wichtig, um zB Firefox oder Flash unter Android auszuführen.[1] Da eine APP unter Android mit einer bestimmten UserID ausgeführt wird, wird auch dieser Code mit den Berechtigungen der APP ausgeführt. Es entstehen keine zusätzlichen Sicherheitsrisiken. Jedoch ist die Fehleranfälligkeit einer APP erhöht, da dies eigentlich nicht vorgesehen war und somit nur bedingt unterstützt wird.

Im Unterschied zu anderen VMs wie der JavaVM bildet die DalvikVM keine weitere Sicherheits-barriere zum System. Sie kann auch direkt nativen Code ausführen, da die Sandbox auf OS Level läuft. [4] Durch das Einbinden von nativem Code kann die Performance und Speichereffizienz auf Kosten von Stabilität und Lesbarkeit verbessert werden.

3.1.2 Sicherheit und Berechtigungen

Um die Sicherheit der Daten auf dem Smartphone zu erhöhen, besitzt Android ein Berechtigungs-system, das festlegt, welche Anwendung welche Tätigkeiten durchführen darf. Die Einstellungen umfassen ua das Wählen von Telefonnummern, Senden von SMS, aber auch den Zugriff auf Internet, Speicherkarte, Telefonnummern oder Kontakte. Welche Berechtigungen eine Anwendung erhält, wird bei der Installation festgelegt. Es erfolgt indirekt, indem die Anwendung dem Benutzer mitteilt, welche Rechte sie benötigt. Durch die Bestätigung bei der Installation akzeptiert der User diese Auflistung. Die Rechte können sich durch ein Update verändern. Bei einer derartigen Änderung kann das Update nicht automatisch eingespielt werden.

3.1.3 Das Android Framework

Der Aufbau ist in Abbildung „Abb. 1 Aufbau des Android Frameworks [3]“ zu sehen. Die An-wendungen (APPs) werden in der obersten Schicht ausgeführt und verwenden die Provider und Manager aus dem Application Framework; um auf die Ressourcen der anderen Applikationen (Content Manager) oder auf die Ressourcen des Gerätes zugreifen zu können. Die Manager aus dem Application Framework sind standardmäßig vorgefertigt und müssen nicht ausprogrammiert werden. Es ist jedoch auch möglich, für eigene Anwendungen die Interfaces neu zu implementieren und somit für weitere Daten und Verwendungszwecke anzupassen. Welcher Manager ausgewählt wird, wird vom System beim Aufruf durch einen Link, eine sogenannte URI, bestimmt. Die Manager müssen nicht manuell gestartet werden. Sie werden vom System verwaltet und bei Bedarf instanziert und der Applikation ein Handle auf die Instanz zurückgegeben. Die Manager verwenden

die Libraries und die Android Runtime für die Ausführung der Programme aus der obersten Schicht. Die Runtime besteht einerseits aus der DalvikVM und andererseits beinhaltet sie auch die Core Libraries, die für die Ausführung der Programme essentiell sind. Die unterste Schicht stellen die Hardwaretreiber dar. Sie verbinden die Libraries mit der Hardware und ermöglichen somit die Hardwareunabhängigkeit des Androidsystems.

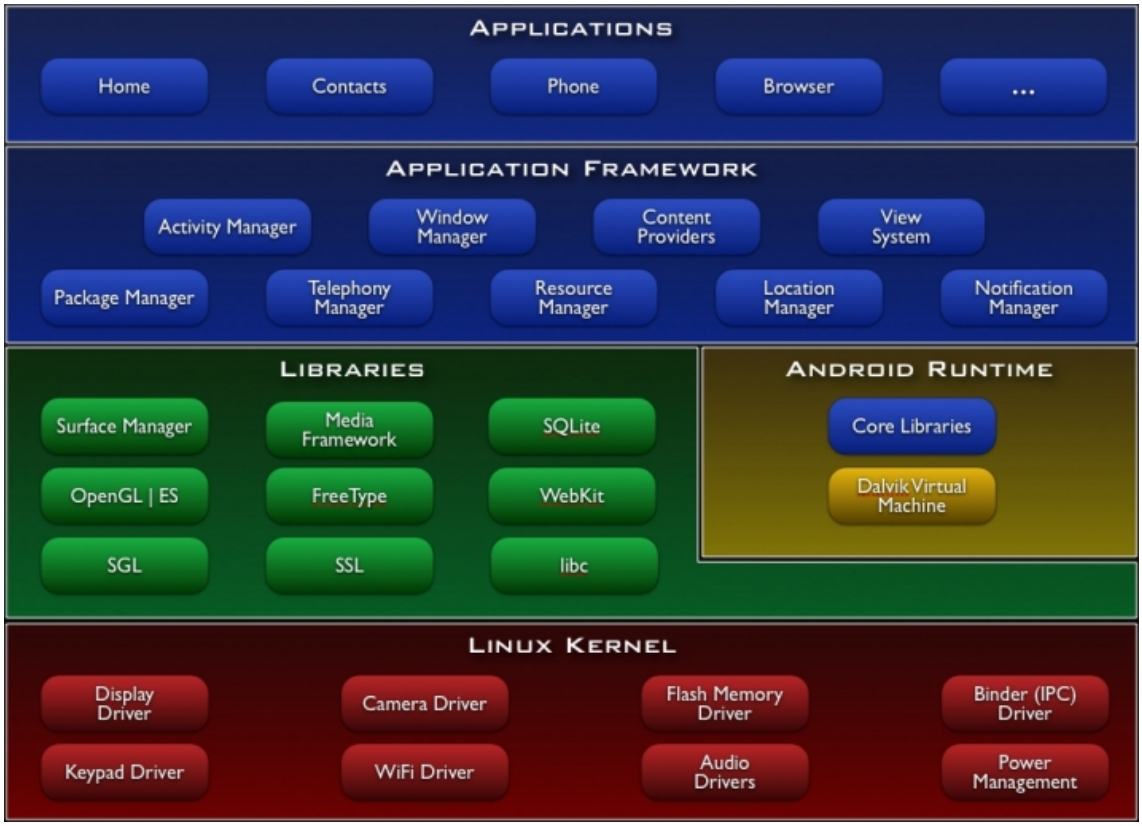


Abb. 1. Aufbau des Android Frameworks [3]

3.2 Funktionsanalyse e-Banking anhand von ELBA mobil

E-Banking bezeichnet das Abwickeln von Bankgeschäften über das Internet. Der Vorteil besteht darin, Bankgeschäfte (Überweisungen, Finanzstatus auslesen, …) rund um die Uhr abwickeln zu können. Um e-Banking nutzen zu können, ist nur ein Internetzugang und die Freischaltung für das System notwendig. Die meisten Banken bieten dieses Service schon seit einigen Jahren an. Da nur ein Browser dafür benötigt wird, kann es auch über mobile Geräte wie Smartphones und Tablets ausgeführt werden. Zur besseren Verwendbarkeit haben die Banken Versionen entwickelt, die mehr Übersichtlichkeit auf den kleinen Bildschirmen bieten.

In dieser Arbeit wird die Webanwendung ELBA-mobil der Firma RACON Software GmbH Linz als Beispiel für e-Banking herangezogen. ELBA-mobil wird ua von der Raiffeisen Landesbank OÖ angeboten. ELBA ist mit https gesichert. (vgl Seite 51, 7.2)

3.2.1 Login

Um das System nutzen zu können, muss sich der Benutzer anmelden. Dazu müssen Bankleitzahl, Konto- oder Depotnummer und Verfügernummer eingegeben werden.

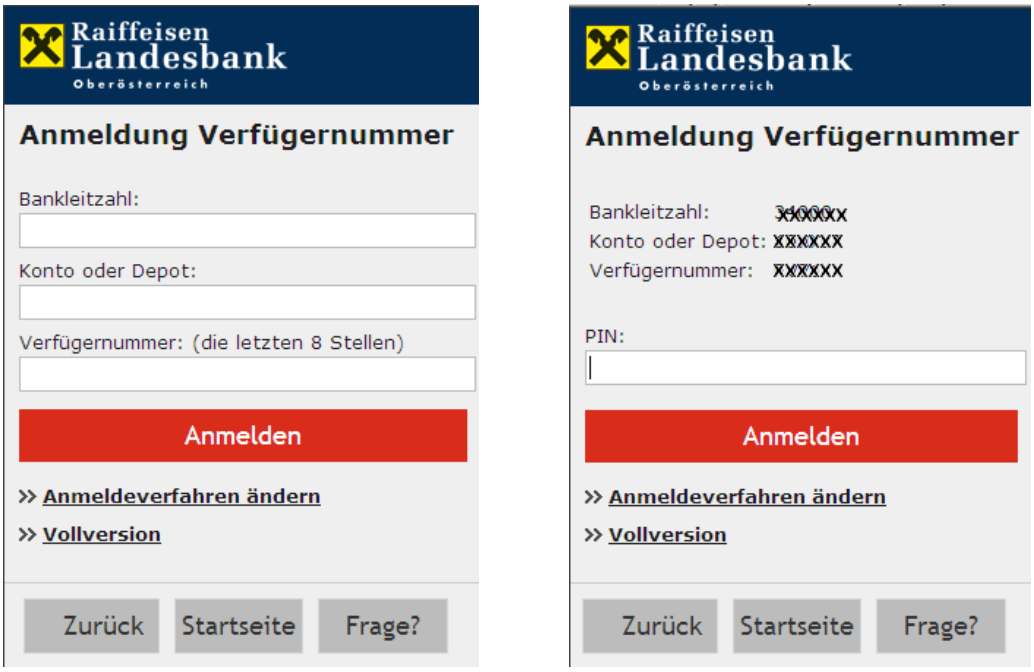


Abb. 2. ELBA-mobil - Logindialog [15]

In einem weiteren Dialog muss die PIN erfasst werden. Durch das getrennte Senden von Anmeldeinformationen und PIN wird einerseits gewährleistet, dass beim Abhören von einem der beiden Loginvorgänge nicht die gesamten Zugangsinformationen erhalten werden. Andererseits kann die Eingabe der Kontoinformationen gespeichert und muss somit nicht für jede Anmeldung erneut erfasst werden. Der Benutzer muss sich dann nur die PIN merken. Die Logindaten (BLZ, Konto- oder Depotnummer, Verfügernummer) werden in einem Cookie gespeichert.

Nach dem Login mit der PIN wird eine Session erzeugt. Diese wird nach 10 Minuten Inaktivität automatisch beendet.

3.2.2 Banknachrichten

Banknachrichten dienen der Information für den Benutzer. Durch Banknachrichten werden aktuelle Informationen an den Benutzer weitergegeben. Sie werden direkt nach dem Login aufgerufen. Durch das Klicken auf Weiter wird die nächste angezeigt. Nach der letzten Nachricht folgt der Finanzstatus (vgl Seite 10, 3.2.4).

Abb. 3. ELBA-mobil - Banknachricht [15]

3.2.3 Navigation

Die Navigationsleiste befindet sich am Ende jeder Seite. Sie beinhaltet die „Menü“-Schaltfläche, über die ua weitere Funktionen ausgewählt werden können (vgl Seite 13, 3.2.6; Seite 16, 3.2.10)

Abb. 4. ELBA-mobil - Navigationsleiste [15]

3.2.4 Finanzstatus

Nach dem Login wird der Benutzer, sofern keine aktuellen Banknachrichten vorhanden sind, automatisch zum Finanzstatus weitergeleitet. Die Übersicht zeigt alle Konten, auf die Zugriff besteht. Nach Auswahl eines Kontos werden dessen Umsatzdaten angezeigt. Von der Umsatzübersicht ausgehend können die Kontodetails angezeigt oder eine Überweisung (vgl Seite 11, 3.2.5) erstellt werden. Bei der Auswahl eines Depots wird die Wertpapierübersicht (vgl Seite 15, 3.2.7) geöffnet.

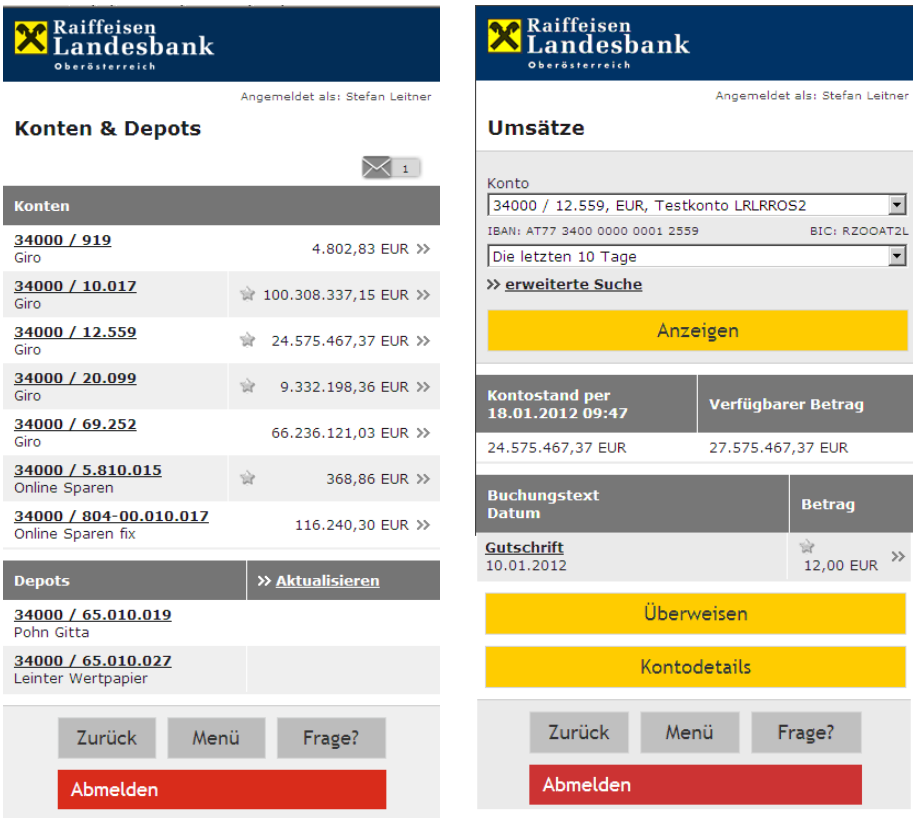


Abb. 5. ELBA-mobil – Finanzstatus, Umsatzübersicht [15]

3.2.5 Überweisung

Durch eine Überweisung wird die Bank beauftragt, Geld von einem Konto auf ein anderes zu transferieren. Das Auftraggeberkonto kann aus der Auswahlliste ausgewählt werden. Es stehen alle Konten zur Verfügung, auf die der angemeldete Benutzer Zugriff hat. Weiters müssen der Empfänger, sowie der Betrag und der Verwendungszweck angegeben werden. Durch „mehr Optionen“ werden weitere Zeilen für den Verwendungszweck und das Datum der Durchführung angezeigt.

Der Überweisungsauftrag wird mit „Überweisen“ und Eingeben der TAN abgeschlossen. Die TAN kann entweder per SMS oder per cardTAN ermittelt werden. Der Auftrag kann auch gespeichert und zu einem späteren Zeitpunkt oder gemeinsam mit anderen Aufträgen abgeschickt werden. (vgl Seite 13, 3.2.6.1)

Abb. 6. ELBA-mobile – Überweisung erstellen [15]

3.2.5.1 iTAN, TAN Liste

Die TAN Liste ist eine gedruckte Liste mit den für dieses Konto generierten TANs, die über den Postweg dem Bankkunden zugestellt wird. Die TANs sind durchnummeriert. Die Gültigkeit wird durch die Eingabe der Ordnungszahl und der TAN gewährleistet. Dieses System ist bereits veraltet und gilt als unsicher gegenüber Abhörattacken. Eine Verbesserung bietet die iTAN. Hier ist bei der Transaktionsbestätgung eine bestimmt TAN einzugeben. Jedoch gilt auch dieses System als unsicher, da keine direkte Relation zwischen der TAN und der Überweisung besteht. Außerdem ist das System sehr anfällig gegenüber Phishing Attacken.

3.2.5.2 mTAN

Bei der mTAN (mobile TAN) wird die TAN per SMS an das Handy des Benutzers gesendet. Damit erfolgt die TAN-Übertragung über einen zweiten Kommunikationskanal, sofern e-Banking auf einem Computer durchgeführt wird. Die mTAN ist im Normalfall 5 Minuten gültig. Wird sie nicht innerhalb dieser Zeit verwendet, muss zum Absenden der Überweisungsaufträge eine neue TAN angefordert werden.

3.2.5.3 cardTAN

Die cardTAN ist eine neue Technologie. Sie ermöglicht die Generierung der TAN ohne Zuhilfenahme des Handys. Es wird mit Hilfe der Bankomatkarte und einer Flicker-Grafik am Display/Monitor die TAN generiert. Der Vorteil gegenüber der iTAN ist, dass diese TAN auftragsbezogen ist und nur für genau eine Überweisung verwendet werden kann. Vorteilhaft gegenüber der mTAN ist zu erwähnen, dass die TAN nicht von einem Angreifer abgehört werden kann, da sie direkt beim Endbenutzer generiert wird. Die Authentizität des Benutzers wird durch die Bankomatkarte und den ELBA-PIN gewährleistet.

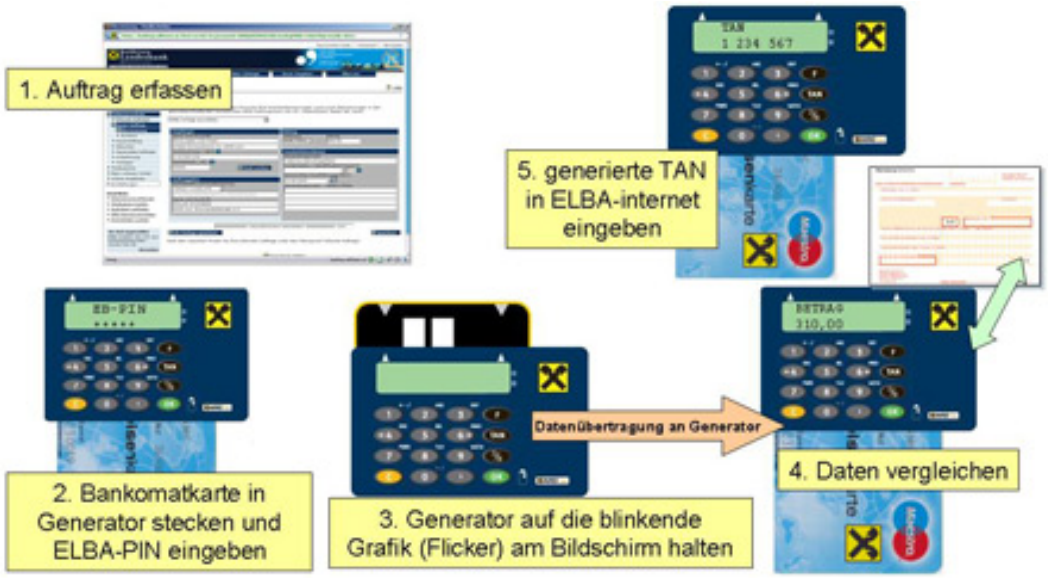


Abb. 7. ELBA-mobil – Workflow cardTAN [14]

3.2.6 Weitere Kontofunktionen

Die weiteren Funktionen können über das Menü erreicht werden. Die Funktionen zum Kontozahlungsverkehr sind im Menü unter Zahlungsverkehr zu finden.

3.2.6.1 Erfasste Aufträge

Unter erfasste Aufträge werden alle erstellten Aufträge aufgelistet, die noch nicht abgesendet wurden. Diese können gesammelt mit einer mTAN oder cardTAN abgesendet werden, wobei maximal 10 Buchungen auf einmal verschickt werden können.

Abb. 8. ELBA-mobil – Erfasste Aufträge [15]

3.2.6.2 Gesendete Aufträge

Unter gesendete Aufträge werden alle Aufträge aufgelistet, die in den letzten 30 Tagen gesendet wurden. Über Filtereinstellungen kann angegeben werden, wie weit zurück die Aufträge angezeigt werden sollen.

Abb. 9. ELBA-mobil – Gesendete Aufträge [15]

3.2.7 Wertpapierdepot

Die Wertpapierübersicht listet alle gekauften Wertpapiere und Informationen zu diesen auf. Weiters werden Daten wie gekaufte Menge und der Kaufwert im Vergleich zum aktuellen Wert (mit Änderungsdatum) und der Veränderung zum originalen Kaufpreis angezeigt.

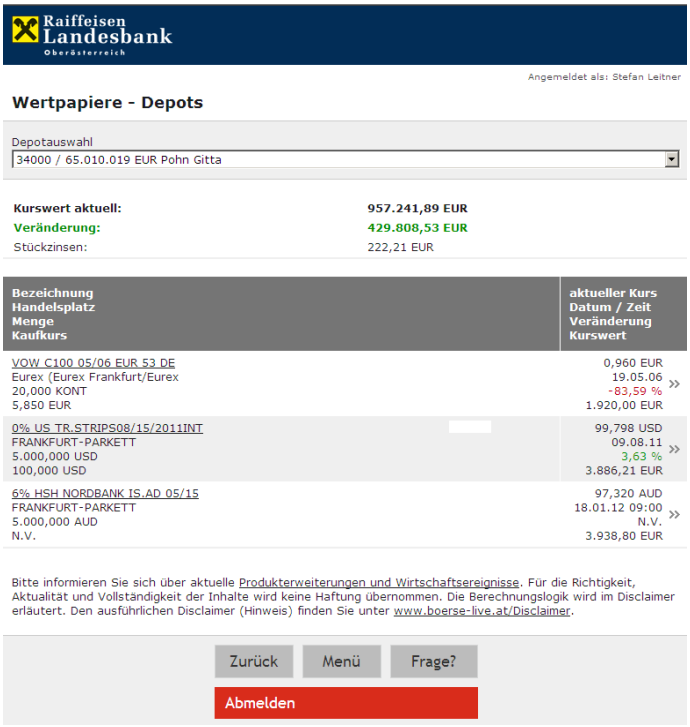


Abb. 10. ELBA-mobil – Depotübersicht [15]

3.2.8 Details zu Wertpapieren

In der Detailansicht eines Wertpapiers gibt es neben den Informationen wie in der Wertpapierübersicht die Möglichkeit zum Kauf und Verkauf.

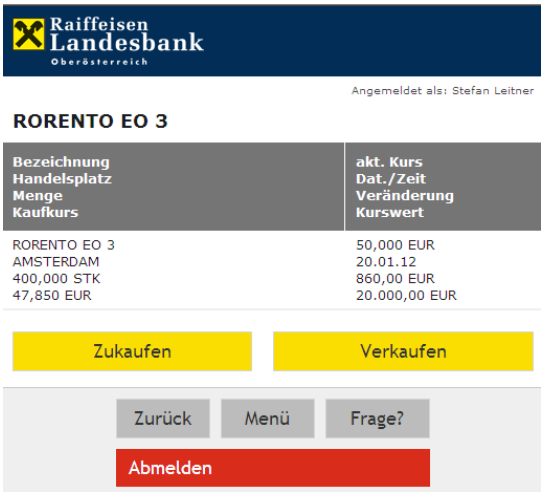


Abb. 11. ELBA-mobil – Details zu Wertpapier [15]

3.2.9 Wertpapierkauf und -verkauf

Beim Kauf oder Verkauf muss einerseits das zu erwerbende oder veräußernde Wertpapier und deren Menge sowie das Depot, das Verrechnungskonto und der Handelsplatz angegeben werden. Beim Verkauf sind weiters das Limit in Börsenwährung und das Gültigkeitsdatum anzugeben.

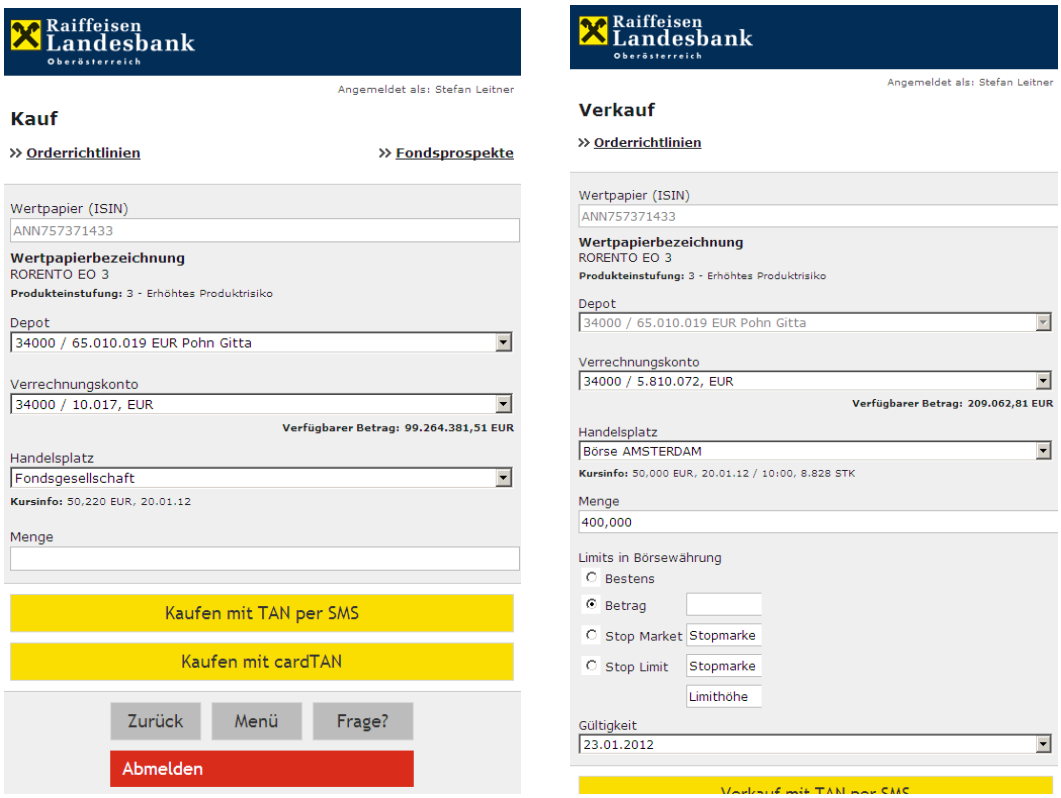


Abb. 12. ELBA-mobil – Kauf und Verkauf von Wertpapier [15]

Der Auftrag wird mittels mTAN oder cardTAN bestätigt. (vgl Seite 12ff, 3.2.5.2; 3.2.5.3) Es ist nicht möglich mehrere Aufträge zu erstellen und diese gesammelt zu senden, jeder Auftrag muss gesondert bestätigt werden.

3.2.10 Weitere Wertpapierfunktionen

Die weiteren Wertpapierfunktionen sind über „Menü - Wertpapier“ erreichbar.

3.2.10.1 Wertpapiersuche

Die Wertpapiersuche ermöglicht die Suche nach Wertpapieren aller Arten über den Namen oder die ISIN (International Securities Identification Number) eine zwölfstellige Buchstaben-Zahlen-Kombination, die ein Wertpapier eindeutig identifiziert. Die Auswahl eines Suchergebnisses führt zur Kaufübersicht. (vgl Seite 16, 3.2.9)

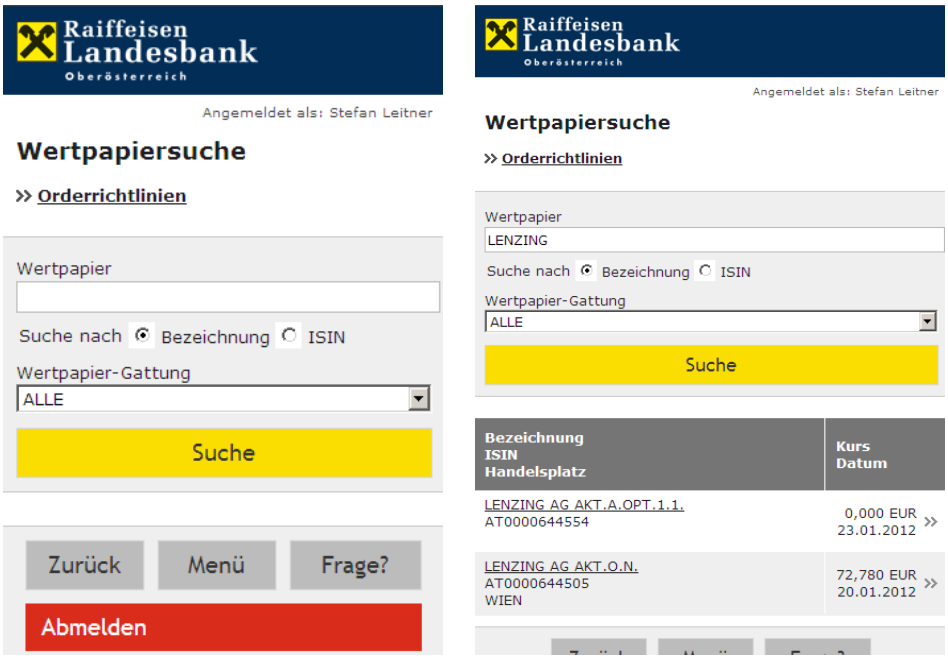


Abb. 13. ELBA-mobil - Wertpapiersuche [15]

3.2.10.2 Orderbuch

Im Orderbuch werden alle durchgeführten Transaktionen mit dem aktuellen Status angezeigt. Bei der Auswahl einer Order werden die Details angezeigt.

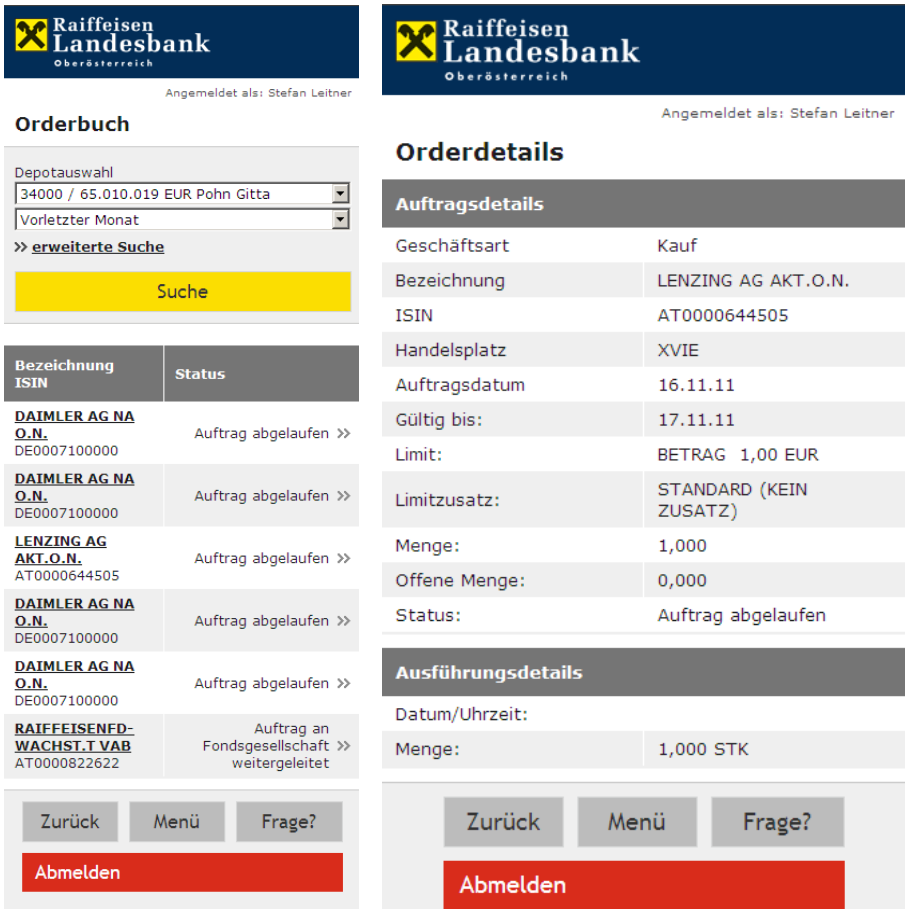


Abb. 14. ELBA-mobil – Wertpapier Orderbuch, Orderdetails [15]

3.2.11 Workflow

ELBA-mobil ist für ein kleines Display optimiert. Es bietet den Zugriff auf fast alle Funktionen der ELBA Oberfläche, wie sie am Computer dargestellt wird. Die wichtigsten Funktionen sind durch einen Workflow erreichbar.

Das Menü enthält alle Funktionen, auch solche, die beim Durchnavigieren nicht erreicht werden. (vgl Seite 13, 3.2.6; Seite 16, 3.2.10)

Anhand folgender Skizze wird der Workflow veranschaulicht. Der Text neben den Pfeilen beschreiben die Buttonbezeichnungen, die Werte in Klammern die Daten, die notwendig sind, um eine Aktion durchzuführen.

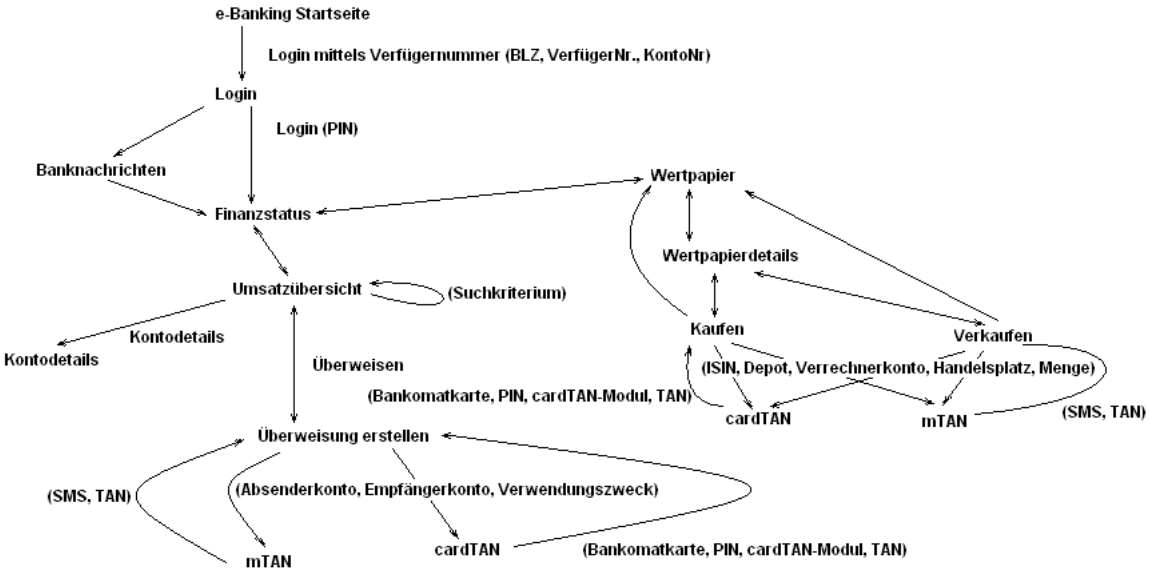


Abb. 15. ELBA-mobil – Workflow ohne Menüfunktionen

4. Gegen Angriff zu schützende Werte

E-Banking mit Android verwendet viele unterschiedliche Daten und Informationen. Es ermöglicht Einblick auf die Finanzen des Benutzers. In den meisten Fällen bestehen bei Verlust der Daten bzw des Zugriffs das Risiko von hohem finanziellem Schaden und dem Eingriff in die Privatsphäre. Somit handelt es sich um eine sehr sicherheitskritische Anwendung.

4.1 Benutzerdaten

Die Handys dienen heutzutage nicht mehr nur zum Telefonieren oder zum Schreiben von Kurznachrichten (SMS), sondern auch zum Abrufen von Mails und Surfen im Internet. Viele verwenden das Smartphone, um in sozialen Netzwerken wie Facebook Postings zu hinterlassen. Somit sind auf dem mobilen Gerät persönliche Daten des Benutzers gespeichert. Weiters können im Speicher

private Daten, wie zB Fotos, Internetverknüpfungen, Cookies und Nachrichten aller Art sein. (e-Mail, SMS, MMS, …)

Es müssen jedoch nicht nur die Daten geschützt werden, die im Speicher abgelegt sind. Viele Informationen werden am Display angezeigt oder per Netzwerkverbindung (GSM, UMTS, WLAN, Bluetooth, …) versendet.

Ein Schutz dieser Daten ist wichtig, da bei Verlust nicht gewährleistet werden kann, wofür die Daten verwendet werden. Im Zuge dieser Masterarbeit sind vor allem folgende Daten wichtig: e-Banking Zugang (PIN, TAN), Finanzmittel (e-Geld und Wertpapiere), Kontoinformationen, Standortdaten.

4.1.1 E-Banking Zugang

Die e-Banking Plattform stellt das Portal zu den Finanzen des Benutzers dar. Die Zugangsdaten bestehen aus Kontonummer, Bankleitzahl, Verfügernummer und PIN. Bei Verlust dieser Daten besteht die Möglichkeit, dass Unberechtigte Zugang zum Konto erhalten und Finanzdaten auslesen können. Wird auch noch die TAN bekannt, können Überweisungen problemlos durchgeführt werden. Die Gefahr besteht vor allem dann, wenn das Smartphone die TAN via mTAN als SMS erhält. (vgl Seite 25, 5.3.3)

4.1.2 Kontoinformationen

Die Kontoinformationen geben die Zahlungsfähigkeit einer Person an. Gehen diese Informationen verloren, entstehen für den Kontoinhaber in anderen Lebensbereichen Risiken. Bei einer Person mit erhöhtem Finanzvermögen ist ein Einbruchs- oder Überfallsversuch lukrativer als bei einer finanzschwachen. Andererseits kann bei letzterer eher mit Zahlungsausfällen gerechnet werden.

Aus der Umsatzübersicht können Rückschlüsse auf den Lebensstil und die Gewohnheiten gemacht werden.

4.1.3 Standortdaten

Ziemlich alle Smartphones und Tablets sind heutzutage schon mit GPS Empfängern ausgestattet. Damit ist es möglich, die Position des Gerätes ziemlich genau zu bestimmen. Das kann sich einerseits vorteilhaft auswirken, wenn man wissen will, wo man sich genau befindet, zB bei der Verwendung eine Navigationssoftware. Andererseits kann diese Information auch von Schadsoftware ausgelesen und zur Ortung von Personen verwendet werden. In Kombination mit dem Bekanntwerden von Kontodaten ist es Dieben nicht nur möglich, den Finanzstatus einer Person

herauszufinden, sondern auch deren Position. Durch die Auswertung von Logs können Rückschlüsse auf den Aufenthaltsorte des Betroffenen gemacht werden (Wohnung, Arbeitsplatz)

4.2 Mobilgerät

Der Anwendungsbereich eines Smartphones hat sich im Vergleich zum normalen Handy, das zum Telefonieren und Versenden von Kurznachrichten verwendet wurde, stark erweitert. Das Installieren von Schadsoftware ist für Hacker bei Smartphones interessanter, da viele Daten von einem Gerät ausgelesen werden können. Die Schwachstelle bei den mobilen Geräten ist, dass die Betriebssysteme teilweise noch nicht so weit ausgereift sind. Außerdem ist das Gefahrenpotential bei den Benutzern noch großteils unbekannt. Die Verwendung von Viren- und Malwareschutz ist noch sehr selten.

4.3 Applikationsdaten

Innerhalb der Anwendung werden bestimmte Werte gespeichert. Beispiele hierfür sind temporäre Internetdateien und Cookies, die benötigt werden, damit die einzelnen Seiten schnell geladen werden können und Downloadvolumen gespart werden kann. Diese Dateien werden im Speicher des mobilen Geräts abgelegt.

4.4 Kommunikation

Die Kommunikation im Internet erfolgt mittels IP Paketen. Diese werden vom Smartphone erstellt und an den Server versandt. Im Falle von e-Banking werden die Anmeldeinformationen, die Kontodaten und auch die Applikationsdaten übertragen. Wird die Kommunikation mitgelesen, kann es zu einer Gefährdung der Benutzerdaten kommen (vgl Seite 18, 4.1)

4.5 Rechenzentrum

Für eine Anwendung wie e-Banking sind mehrere Server sowie Geräte zur Lastverteilung und Ausfallssicherheit notwendig. Damit diese Infrastruktur funktioniert, ist es wichtig, dass keine unberechtigten Zugriffe das System aus dem Gleichgewicht bringen. Nach Außen wirkt das Netzwerk wie ein einzelner Server.

E-Banking benötigt neben den Servern für die Generierung der Oberfläche (Schnittstelle zum Benutzer) auch Server zur Authentifizierung und Datenbanken, in denen die Kontoinformationen ausfallssicher gespeichert werden.

Ein Ausfall des Systems würde die Verfügbarkeit von e-Banking stark beeinflussen, was einerseits einen großen finanziellen Verlust für die Anleger aber auch einen finanziellen und vor allem einen Vertrauensverlust für die Bank bedeuten würde.

5. Potentielle Schwachstellen

In der Kommunikation zwischen einem Smartphone und den Servern im Rechenzentrum sind viele unterschiedliche Komponenten beteiligt. Durch die große Anzahl entstehen auch viele Schwachstellen und Möglichkeiten für einen Angriff auf das System.

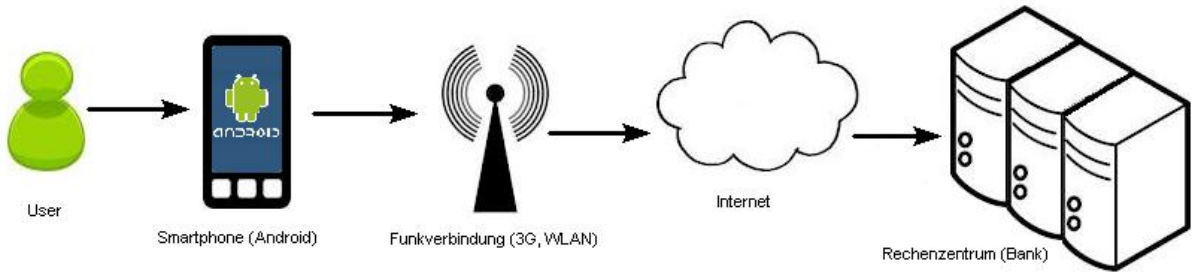


Abb. 16. Vereinfachte Darstellung Kommunikationsweg

In den folgenden Unterkapiteln möchte ich auf die Komponenten näher eingehen.

5.1 User

Der Benutzer ist ein sehr unvorhersehbares Element in der Kommunikationskette. Aufgrund vieler verschiedener Persönlichkeiten ist es bei manchen einfacher, eine bestimmte Sicherheitslücke auszunutzen, während es bei anderen unmöglich ist. Der „Normal-Benutzer“ ist kein Experte und hat wenig Wissen im Computer-Umfeld. Er will die Funktionen des Gerätes möglichst schnell und ohne Umstände nutzen können. Daraus ergeben sich folgende potentielle Schwachstellen.

5.1.1 Unwissen

Aufgrund der vielen Veränderungen im IT Bereich ist es für einen Normalbenutzer nicht mehr möglich, sich mit allen Funktionen und Möglichkeiten, die das Internet bietet, auseinander zu setzen. Dadurch entstehen Sicherheitslücken, die von Angreifern ausgenutzt werden können. Warnungen und Fehlermeldungen werden ungelesen weggeklickt. Die Betriebssysteme werden zwar immer intelligenter und erlauben dem Benutzer bestimmte Aktionen nicht mehr, jedoch können diese Sicherheitseinstellungen durch zB rooting (Smartphones) oder Anmeldung als Administrator oder Root (Computer) umgangen werden.

Oft ist einem Benutzer das Sicherheitsrisiko, das durch die Einstellungen eingegangen wird, nicht bewusst. Es ist vor allem wichtig, dass alles einfach und schnell geht und „passiert“.

5.1.2 Unachtsamkeit mit Berechtigungen

Es wird immer wichtiger, dass die Programme einfach und schnell funktionieren. Langwierige Installationsroutinen sind unbeliebt und verzögern den Start des Programms. Aus diesem Grund werden diese Agenten oft ungelesen durchgeklickt und sowohl AGBs aber auch Berechtigungen für die Applikation einfach akzeptiert. Dadurch erhalten APPs oft mehr Rechte als sie eigentlich benötigen. Ohne Akzeptierung der Berechtigungen wird die Applikation jedoch nicht installiert.

5.1.3 Fehlende Tastensperre

Die Tastensperre schützt in erster Linie vor unabsichtlicher Benützung des mobilen Gerätes, wenn sich dieses in einer Tasche befindet. Die Tastensperre kann aber je nach Ausführung auch als Sicherheitsmechanismus und somit als erste Hürde gegen einen Angreifer verwendet werden. Dies ist dann der Fall, wenn eine PIN oder eine Mustereingabe notwenig ist, um das mobile Gerät zu entsperren. (vgl Seite 70, 8.1.2; 8.1.4)

5.1.4 Benützung des mobilen Gerätes durch Dritte

Oft werden Handys und mobile Geräte für den Telefon- oder Internetzugriff an Freunde und Bekannte weitergegeben. Durch dieses Verleihen von Geräten erhält eine andere Person als der Besitzer direkten Zugriff auf das Gerät. Dieses kann nicht zwischen den beiden unterscheiden. Die andere Person kann sich dem Gerät gegenüber als Eigentümer ausgeben und alle Daten auslesen, bzw die vorhandenen Services nutzen und neue installieren. Damit gilt aus Sicht der Sicherheit ein Verleihen des Gerätes als ähnlich gefährlich wie Verlust oder Diebstahl. (vgl Seite 33, 6.4) Beim Verleih ist dem Besitzer jedoch bekannt, wer das Gerät benützt und im Normalfall besteht auch ein Vertrauensverhältnis zwischen Besitzer und der entleihenden Person. Die Gefahr ist daher eher gering.

5.2 System

Als System wird in dieser Masterarbeit nur Android betrachtet. Android wird von Google weiterentwickelt und basiert auf einem Linux Kernel. (vgl Seite 6, 3.1) Die Versionen auf den Endgeräten wurden jedoch noch von den Herstellern angepasst und auf die entsprechende Hardware zugeschnitten. Folgende potentielle Schwachstellen sind durch das System gegeben.

5.2.1 Gerootete Geräte

Bei Android ist der Benutzer standardmäßig kein Superuser. Er besitzt somit nur eingeschränkte Rechte und kann nicht auf alles zugreifen. Dies dient als Sicherheitsfunktion, da damit sicherheitskritische Vorgänge unterbunden werden. Die Smartphones und Tablets, die von den Providern

verkauft werden, sind gebrandet, wodurch das Gerät nur mit der SIM-Karte des verkaufenden Providers funktioniert. Diese Einschränkungen können durch rooting umgangen werden. Dazu wird eine Customfirmware auf das Handy gespielt. Diese schaltet einerseits den SIM-Lock ab und andererseits wird oft der Benutzer in den Root-Status gehoben.

Dadurch können mehrere Sicherheitsprobleme entstehen: Der Benutzer, der das rooting durchführt, ist in vielen Fällen kein Experte. Er möchte entweder das Gerät mit einer anderen SIM-Karte nutzen oder Zugriff auf weitere Funktionen erhalten. Dass dadurch viele Sicherheitsmechanismen abgeschaltet werden und viele Einstellungen, die Sicherheitslücken öffnen, aktiviert werden, ist dem Benutzer oft nicht bewusst oder bekannt. Gerootete Geräte sind anfälliger für gefährliche Programme und können als Ausgangspunkt für weitere Angriffe dienen. Das Auslesen von Daten ist auf gerooteten Geräten für Malware sehr viel einfacher. So kann im gerooteten Zustand mit der Superuser-Berechtigung auf den Framebuffer zugegriffen und ohne Wissen des Benutzers Screenshots erzeugt und über das Internet versendet werden.

5.2.2 Veraltetes System (fehlende Updates)

Android ist ein offenes Betriebssystem. Es wird von Google entwickelt. Das System wird von den einzelnen Herstellern für die Geräte angepasst. Somit gibt es keine generell gültigen Updates. Der Benutzer ist in Bezug auf Systemupdates vom Hersteller abhängig. Diese verabsäumen oft die Erstellung und Herausgabe eines Updates. Damit besitzen viele Android-Benutzer eine veraltete Version des Betriebssystems. Viele Sicherheitslücken, die in aktuellen Systemen bereits geschlossen sind, bleiben somit offen und bieten weiterhin Angriffspunkte.

5.2.3 Speicherkarte (SD Karte)

In den mobilen Geräten befinden sich zur Speicherung von Daten, Dokumenten oder Multimediadateien wie Fotos, Musik oder Filme Speicherkarten. Bei diesen handelt es sich meist um SD Karten. Dieser Kartentyp ist sehr weit verbreitet und kann von den meisten Kartenlesen gelesen werden. Das Zugriffssystem von Android auf die Daten der SD Karte kann leicht umgangen werden, indem die Speicherkarte von einem anderen Gerät, wie zB einem Computer ausgelesen wird. Auf der Speicherkarte sollten daher nur ungefährliche Daten abgelegt werden. Wichtige oder geheime sollten ausschließlich verschlüsselt gespeichert werden.

5.2.4 Interner Telefonspeicher

Jedes mobile Gerät besitzt intern einen Speicher, auf dem sowohl das System als auch die installierten APPs gespeichert werden. Manche APPs erlauben die Installation auf der SD Karte. Der interne Speicher kann nicht aus dem mobilen Gerät ausgebaut werden, ohne das Gerät zu

zerstören. Somit kann er nicht von einem Computer direkt ausgelesen werden. Es ist jedoch möglich, den Speicher über die Debugkonsole einer Entwicklungsumgebung (zB Eclipse mit DDMS) auszulesen. Dazu muss der Debugmodus aktiviert sein. Der interne Speicher wird zwar durch die Berechtigungen von Android geschützt, jedoch sollten wichtige und geheime Daten auch hier nur verschlüsselt abgelegt werden, da durch Rooting und das Auslesen über die Entwicklungsumgebung auf alle Daten zugegriffen werden kann.

5.2.5 Software und Softwarefehler

In jeder Software treten Fehler auf. In vielen Fällen sind es kleine Bugs, die weder die Funktionsfähigkeit noch die Sicherheit stark beeinträchtigen. Manche Fehler können jedoch schwere Sicherheitsmängel auslösen und die Anwendung oder die Server und Geräte, die für die Ausführung benötigt werden, durch einen Hacker angreifbar machen. Fehler können nur durch Tests und Code-Reviews gefunden werden. Tritt dennoch ein Fehler auf, sollte versucht werden, diesen möglichst schnell zu beheben, da sonst eventuell die Daten oder Server gefährdet werden. Nach einem Update der Software muss erneut getestet werden, da durch das Beheben eines Fehlers neue entstehen können.

5.2.6 Designfehler

Das Design von Android sieht ein Sicherheitssystem vor, das auf Berechtigungen und der Sandbox der DalvikVM basiert. Beides kann einerseits vom Benutzer und andererseits vom APP-Entwickler beeinflusst und verändert werden. Ein Großteil der Sicherheit wird somit vom Benutzer festgelegt.

Das Design regelt die Zugriffe auf die Informationen und Daten. Nicht immer sind diese Möglichkeiten schlüssig und es kann sein, dass auf zu viele Daten zugegriffen werden kann. Dies entsteht oft bei der Erweiterung des Systems, um zusätzliche ursprünglich nicht vorgesehene Funktionen zur Verfügung zu stellen.

5.3 Gerät (Smartphone)

Das Endgerät ist im Eigentum des Servicebenutzers. Dieser ist für die Sicherheitseinstellungen zuständig. Vom Rechenzentrum aus können keine Einstellungen geändert werden. Aus diesem Grund muss das Endgerät als unsicher und nicht vertrauenswürdig angesehen werden.

Weiters werden die Geräte immer leistungsfähiger und haben einen größeren Funktionsumfang. Jedoch sind die verbauten Komponenten nicht immer gleich und somit kann es sein, dass bestimmte Hardwarekomponenten nicht verfügbar sind (zB NFC, GPS)

5.3.1 WLAN, Bluetooth, MMS, NFC

Funkverbindungen können auf mehrere Arten Angriffspunkte darstellen. Alle Funkteilnehmer verwenden zur Übertragung dasselbe Medium. Es ist nur notwendig, in der Nähe zu sein. Damit ist es einfach, die Datenpakete abzufangen und zu analysieren. Werden diese unzureichend oder nicht verschlüsselt, kann der Inhalt einfach ausgelesen werden.

Ein weiteres Problem stellt die Möglichkeit des Eindringens dar. Angreifer müssen sich nicht physikalisch mit dem Gerät verbinden. Eine Funkverbindung ist „unsichtbar“. Nach dem Aufbau der Verbindung zu dem Gerät kann auf das Gerät zugegriffen werden. Fehlen weitere Sicherheitsmaßnahmen ist ein Vollzugriff möglich. [6] [8](Seite 118, G.75)

Sowohl WLAN als auch Bluetooth sind verwundbar gegenüber DoS Attacken. (vgl Seite 37, 6.10)

NFC (Near Field Communication) wird derzeit unter anderem bei Plakaten verwendet, um Interessierten zusätzliche Informationen im Internet bereitstellen zu können. Durch einen modifizierten Tag auf dem Plakat könnte ein Angreifer einen Benützer auf eine falsche Seite weiterleiten und somit einen Phishing Angriff durchführen. (vgl Seite 30, 6.1) [36] NFC unterstützt keine zusätzliche Authentifizierung. Somit kann ein Angreifer bei Verlust oder Diebstahl des Gerätes auf die über NFC freigegeben Ressourcen (zB elektronische Geldbörse, Schlüsselfunktion) zugreifen.

5.3.2 Gebrauchsspuren

Durch die Benützung von Geräten entstehen automatisch an bestimmten Stellen Abnützungen und Gebrauchsspuren. Diese Spuren können einerseits Kratzer auf dem Gehäuse sein, die auf einen ungeschützten Transport in der Hosentasche oder dem Rucksack hinweisen. Andererseits bilden sich auch auf dem Display Kratzer, die durch häufiges Eingeben desselben Musters sichtbar werden. Diese Spuren können ungefährlich sein, wie zB die vom Scrollen, jedoch bildet sich auch das Entsperrmuster am Display ab. Dieses ist wahrscheinlich eines der am häufigsten eingegebenen Muster. Ähnlich verhält es sich auch mit den Tasten, die für die Passwort oder die PIN Eingabe verwendet werden. Das Risiko bei den Tasten ist jedoch geringer, da durch einen Druck weniger Kratzer entstehen als durch das Ziehen bei der Mustereingabe. Außerdem werden die Tasten auch anderwärtig, zB zum Versenden von Nachrichten (e-Mail, SMS) verwendet. [7](Seite 17)

5.3.3 2-Wege Autorisierung auf einem Gerät

E-Banking verwendet beim Absenden eines Transaktionsauftrages eine TAN. Diese wird durch einen weiteren Kommunikationskanal zur Bank übertragen. Die ursprüngliche Version der TAN ist eine gedruckte Liste, die über den Postweg versendet wurde. Das neue System verwendet die Versendung des Codes per SMS. Wird e-Banking am Computer verwendet, sind es weiterhin zwei

Kommunikationswege. Durch das Verwenden von e-Banking auf dem Smartphone befindet sich nun sowohl die SMS-Kommunikation als auch die Banking-Kommunikation auf demselben Gerät. Dies kann zu Gefahren führen, da durch Schadsoftware nun die SMS ausgelesen und Transaktionen automatisch bestätigt werden könnten. Weiters muss ein Angreifer nur noch auf ein Gerät Zugriff bekommen, um den Kontozugang vollständig anwenden zu können. Somit stellt es auch ohne Schadsoftware ein großes Sicherheitsproblem dar.

5.3.4 Hardwarefehler

Die Hardware eines Smartphones ist sehr kompakt verbaut. Ein Großteil des Volumens des Gerätes wird durch den Akku und das Display in Anspruch genommen. Die restlichen Komponenten müssen sehr platzsparend um diese Teile eingebaut werden. Da die Teile und Toleranzen für den Einbau sehr klein sind, kann es leicht zu Fehlern kommen. Die Folge sind Fehlfunktionen einzelner Teile oder auch ganz Komponenten. Dadurch kann die Funktionalität merkbar (zB nicht funktionierende Tasten) oder unmerklich (falsche GPS Koordinaten) beeinträchtigt werden. Eine weitere Fehlerquelle ist die Empfindlichkeit der Teile. Bereits kleine Überspannungen können einen Ausfall von Elementen und Bauteilen hervorrufen.

Es besteht jedoch auch die Möglichkeit, dass Veränderungen an der Hardware von einem Angreifer durchgeführt werden. Wie die herstellerbedingten Fehler können auch diese in den meisten Fällen nicht von einer Software erkannt werden. [7](Seite 17)

5.3.5 Mobilität

Smartphones dienen der mobilen Kommunikation und sind somit nicht ortsgebunden. Sie werden fast immer überall mitgenommen und sind dadurch direkt der Öffentlichkeit ausgesetzt. Weiters werden die mobilen Geräte auch in dieser verwendet, wodurch den nebenstehenden Personen einerseits die Existenz des Gerätes bewusst wird und andererseits sie unter Umständen die Anzeige auf dem Display mitlesen können.

5.4 Funkverbindung

Mobile Geräte verwenden zum Datenaustausch nur Funktechnologien. Die einzige Möglichkeit, um Daten zum und vom Gerät zu senden, ohne dabei eine Funkverbindung zu nutzen, ist die USB Schnittstelle. Diese kann jedoch bei den meisten Androidsystemen nicht für den Datenverkehr ins Internet genutzt werden.

5.4.1 Luft als Übertragungsmedium

Um die Mobilität zu ermöglichen, werden die Daten mittels elektromagnetischer Wellen über die Luft übertragen. Die übertragenen Datenpakete werden somit nicht abgeschlossen in einem Kabel geführt. Alle, die in der nahen Umgebung sind, können theoretisch die Daten abfangen und auslesen. (vgl Seite 25, 5.3.1)

5.4.2 Freie Hotspots

An vielen öffentlichen Orten wird bereits kostenloses WLAN angeboten. Jeder kann sich nach Belieben in das Netzwerk einloggen und an der Netzwerkkommunikation teilnehmen. Oft sind diese Netzwerke ohne Verschlüsselung, wodurch alle anderen Teilnehmer die Kommunikation im Klartext mithören können.

Eine weitere Sicherheitslücke ergibt sich daraus, dass der Benutzer mit einem nicht vertrauenswürdigen Netzwerk kommuniziert. Es besteht die Gefahr einer Man in the Middle Attack. Der Betreiber des AP (Access Point, Hotspot) kann die Daten zwischen dem AP und dem Anschluss zum Telekom Provider zwischenspeichern und leicht auslesen. (vgl Seite 38, 6.11)

5.4.3 GPS Modul

Für viele moderne Anwendungen ist es hilfreich, wenn dem Benutzer bekannt ist, wo er sich gerade befindet. Um dies zu ermöglichen, sind die meisten mobilen Geräte mit einem GPS Modul ausgestattet, das die aktuelle Position des Geräts feststellt.

5.5 Internet

Das Internet verbindet den Mobilfunkanbieter mit dem Rechenzentrum. Es ist ein Konglomerat aus Routern, Switches sowie Proxies und anderen Netzwerkgeräten. Es ist nicht vorhersehbar, wie ein Paket vom mobilen Gerät durch dieses Netz bis zum Rechenzentrum geleitet wird.

5.5.1 Anonymität

Viele Geräte haben Zugriff auf das Internet. Es besteht keine Identifikationspflicht. Somit muss keiner seine Identität preisgeben bzw kann eine falsche vorgeben. Es kann oft nicht oder nur schwer nachvollzogen werden, wer der Kommunikationspartner ist. Endstellen der Kommunikation authentifizieren sich in vielen Fällen bereits mit einem Zertifikat, jedoch bleiben die Geräte zwischen den Endpunkten anonym.

5.5.2 Vertrauen auf öffentliche Services (DNS, Proxies)

Um komfortabel im Internet surfen zu können, sind bestimmte Services wie DNS oder Proxies und Gateways notwendig. Diese Services sind in vielen Fällen öffentlich zugänglich und werden international oder privat betrieben. Die Standardversion von DNS unterstützt keine Authentifizierung. Somit kann nicht sichergestellt werden, ob auch wirklich mit dem vertrauenswürdigen Server kommuniziert wird, oder ob durch einen Hacking Angriff die Informationen von einem falschen, unsicheren Server abgerufen werden. Proxies werden oft verwendet, um die eigentliche Herkunft der Pakete zu verschleiern oder um Werbung und unerwünschte Inhalte zu filtern. Problematisch sind öffentliche Proxies, da in vielen Fällen nicht nachvollzogen werden kann, wer den Proxy betreibt und was mit den Daten, die über diesen laufen, passiert.

5.6 Rechenzentrum

Das Rechenzentrum stellt die Gegenstelle zum mobilen Gerät dar. Hier werden die Anfragen empfangen, verarbeitet und beantwortet. Hier laufen bestimmte Services und Netzwerkgeräte, die einen unterbrechungsfreien und störungsfreien Zugriff auf e-Banking ermöglichen sollen. Dies wird hardwareseitig durch Router, Switches und Loadbalancer und softwareseitig durch Datenbanken, Webserver und Sicherheitssoftware wie Virenscanner ermöglicht. Bestimmte Systeme wie Firewalls und Intrusion Prevention Systeme (IPS) und Intrusion Detection Systeme (IDS) arbeiten sowohl auf Hardware- als auch auf Softwareebene.

5.6.1 Physische Existenz

Die Server des Rechenzentrums müssen irgendwo untergebracht und betrieben werden. Da es immer wieder notwendig ist, die Hardware zu warten und auszutauschen, ist es auch wichtig, dass es Zugänge zu den Systemen gibt. Über diese Zugänge können potentielle Angreifer direkt Zugriff auf das System und somit auch auf die Daten nehmen.

Ein Rechenzentrum benötigt weiters Ressourcen von außen, um funktionieren zu können. Beispiele hierfür sind Strom und klimatisierte Luft. Es ist notwendig, dass die Betriebsumgebung trocken und möglichst staubfrei gehalten wird. Je nach Lage kann das System auch von Umwelteinflüssen (Hochwasser, Erdbeben, Stürmen) betroffen sein. Dies wird jedoch in dieser Arbeit nicht näher behandelt.

5.6.2 Internetzugang

Neben der physischen Erreichbarkeit ist es auch wichtig, dass die Server vom Internet aus erreichbar sind, damit die Benutzer der Services auch auf diese zugreifen können. Ein Internet-

zugang ist somit essentiell für den Betrieb. Dieser Zugang stellt deshalb ein potentielles Risiko dar, da er auch für andere Anwendungen als die vorhergesehenen verwendet werden kann. Die generelle Absicherung des Internetzuganges wird hier nicht weiter behandelt.

5.6.3 Kommunikationstunnel vom mobilen Gerät

Eine sichere Kommunikation basiert in den meisten Fällen auf einem verschlüsselten Tunnel, der zwischen dem mobilen Gerät und der Endstelle, dem Server im Rechenzentrum, aufgebaut wird. Dieser Tunnel kann bei gut gewählter Verschlüsselung von außen nicht abgehört und mitgelesen werden. Somit ist es auch nicht möglich, mit Firewalls den Traffic zu kontrollieren. Ermöglicht eine Schwachstelle in der Anwendung, dass dieser Tunnel von einer anderen Applikation verwendet wird, erhält diese direkt Zugriff in das Rechenzentrum. Der Angriff kann nicht von den Firewall oder IDS und IPS abgefangen werden, da diese nur die verschlüsselten Pakete sehen. In Kombination mit einem Anwendungsfehler auf der Serverseite kann ein Angreifer Zugriff auf das vollständige System erhalten. Ein Beispiel ist die Sicherheitslücke im Apache Webserver vom März 2010. [32]

5.6.4 Softwarefehler

Auf dem Server läuft die Gegenstelle der Kommunikation mit dem mobilen Gerät. Diese verarbeitet die Anfragen und liefert dem Client eine Antwort. Um diese Antwort zu erstellen, ist es notwendig, andere Geräte und Services, zum Beispiel Datenbanken, im Hintergrund zu kontaktieren, um von denen Informationen zu erhalten, oder um diesen Informationen und Daten weiterzuleiten. Im Normalfall gibt es definierte Schnittstellen, über die ein Client mit dem Server kommunizieren darf. Beinhaltet ein Script einen Fehler, kann es zu Bedrohungen für den Webserver aber auch für die Hard- und Software im Hintergrund kommen. Ein Beispiel hierfür wäre ein fehlerhafter Datenbankzugriff durch falsche Ausprogrammierung von Queries (SQL Injection)[17]. Das Vorbeugen und Erkennen von Softwarefehler wird in dieser Masterarbeit nicht weiter behandelt.

6. Mögliche Angriffe

Es existieren viele Angriffe, die die Sicherheit von e-Banking auf dem Endgerät, bei der Übertragung oder im Rechenzentrum gefährden. Die Methoden und Auswirkungen sind unterschiedlich. Mache versuchen die Kommunikation abzuhören, um an geheime Informationen zu kommen. Ein derartiger passiver Eingriff ist schwer zu erkennen, kann jedoch durch Verschlüsselung erschwert werden.

Andere Methoden modifizieren die übertragenen Daten oder fügen neue Daten hinzu. In den meisten Fällen wird versucht, neue Finanztransaktionen zu erstellen oder bestehende zu bearbeiten, sodass ein finanzieller Gewinn für den Angreifer entsteht.

Es gibt jedoch auch Angriffe, deren primäres Ziel die Störung der Kommunikation und somit die Verhinderung der Funktionalität (DoS) ist. Diese Angriffe können zwar erkannt, jedoch teilweise nur schwer unterbunden werden.

Die folgenden Kapitel beschreiben die möglichen Angriffe und Angriffsmöglichkeiten, die sich durch die potentiellen Schwachstellen des Systems ergeben. (vgl Seite 21, 5.)

6.1 Social Engineering

Sozial Engineering ist in der heutigen Zeit ein wichtiges Thema. Durch den Aufbau von Beziehungen zu den Betroffenen oder durch das Vorspielen einer falschen Identität wird versucht, an Daten wie zB Passwörter oder PINs zu kommen. Beispiele hierfür sind anonyme Freundschaftsanfragen und Kontakte in sozialen Netzwerken wie Facebook aber auch phishing Mails und phishing Websites. Gefördert wird dies durch die Anonymität des Internets und, die Unkontrollierbarkeit des Netzes aufgrund seiner Größe. Es ist für einen Normalbenutzer oft nicht möglich, die Identität eines Kommunikationspartners zu ermitteln.

Durch Zertifikate kann die Identität eines Webservers sichergestellt werden. Oft fehlen jedoch beim Benutzer das Wissen und die Geduld, um diese Sicherheitsmerkmale auch zu nutzen. (vgl Seite 21, 5.1.1) Damit kann zwar in gewissen Bereichen phishing vorgebeugt werden, jedoch sind Zertifikate bei Privatpersonen unbekannt oder zu umständlich zu nutzen. Da die gewünschten Services auch ohne zusätzlichen Aufwand (Beschaffung und Einrichtung) verfügbar und nutzbar sind, wird von vielen darauf verzichtet. Dadurch stellen soziale Netzwerke und Mail weiterhin ein Gefahrenpotential dar.

6.1.1 Gefährdete Werte

Durch Social Engineering können alle Daten in Erfahrung gebracht werden, die dem Benutzer auch bekannt sind. Somit sind die Benutzer- und Applikationsdaten gefährdet. (vgl Seite 18ff, 4.1; 4.3)

6.1.2 Genutzte Schwachstellen

Der Gefährdung basiert hauptsächlich darauf, dass der Benutzer dem Angreifer vertraut, unwissend ist, oder ihm die Wichtigkeit der Daten nicht bewusst ist. Der Angriff nutzt vor allem die Unwissenheit und Unachtsamkeit des Benutzers, um an dessen Daten zu kommen. (vgl Seite 21ff, 5.1)

Weiters ist die Tatsache der fehlenden 2-Wege-Autorisierung für diese Attacke zutreffend. (vgl Seite 24, 5.3.3)

6.1.3 Relevanz

Die Relevanz von Sozial Engineering ist sehr hoch, da dadurch leicht wichtige Daten verloren gehen können, bzw private Daten öffentlich bekannt werden. Die Verhinderung ist jedoch schwierig, da das Bewusstsein für die Wichtigkeit der Geheimhaltung bestimmter Daten beim Benutzer nicht vorhanden ist und außerdem dieser in vielen Fällen den Angriff nicht direkt bemerkt. Beispiele hierfür sind phishing Mails oder nachgebaute Homepages.

6.2 Schadsoftware und andere Anwendungen (zB Keylogger)

Von jeder installierten Anwendung kann eine Gefahr ausgehen, da sie Fehler enthalten kann und dadurch unvorhersehbare Aktionen durchgeführt werden können. Auch kann es sein, dass diese absichtlich ausgeführt werden, um Daten auszulesen und den Benutzer auszuspionieren. Ist dieses Programm auch noch mit bestimmten Berechtigungen ausgestattet, kann es mit diesen Berechtigungen auf Daten und Services außerhalb der Sandbox zugreifen und das System und deren Applikationen beeinträchtigen.

APPs können bei schlechter Implementierung eigene Rechte an andere APPs weitergeben. Dadurch können Anwendungen mit wenigen Rechten Zugriff auf Daten erhalten, der nicht vom Benutzer akzeptiert wurde. [12] (zB Zugriff auf Fotos über die Gallerie-APP)

Keylogger laufen im Hintergrund und speichern je nach Konfiguration und Funktionsweise bestimmte oder alle Benutzereingaben. Sie sammeln die Daten, um sie später über das Internet an einen Server zu senden. Von dort aus können diese verwendet werden, um weitere Angriffe auszuführen.

Das Androidsystem pausiert normalerweise Prozesse, die nicht im Vordergrund sind. Um dies zu unterbinden, muss er als Service ausgeführt werden. Auf diese Weise läuft es auch dann weiter, wenn die Applikation selber für den Benutzer nicht oder nicht mehr sichtbar ist. [5]

Mit MMS können auch komplexe Daten übertragen werden. Somit ist es theoretisch auch möglich, APPs mittels MMS zu versenden. [8](Seite 85) Besitzt eine Applikation die Berechtigung MMS zu senden und gleichzeitig auch auf das Adressbuch zuzugreifen, kann sich die Anwendung selbständig an die Kontakte verbreiten. Auf ungerooteten Geräten muss eine derartige Installation noch vom Benutzer bestätigt werden. Damit wird die Gefahr eingedämmt.

6.2.1 Gefährdete Werte

Schadsoftware greift direkt auf das mobile Gerät zu. Damit kann unter Umständen auf alle gespeicherten und verarbeiteten Daten zugegriffen werden. Gefährdet sind somit sowohl die Benutzer- und Anwendungsdaten als auch das Mobilgerät und die Kommunikation. (vgl Seite 18ff, 4.1, 4.2, 4.3, 4.4)

6.2.2 Genutzte Schwachstellen

Schadsoftware kann auf verschiedene Wege auf das mobile Gerät gelangen. Um auf dem Gerät Schaden anrichten zu können, sind im Falle von Android vor allem zusätzliche Berechtigungen notwendig. Die Gefährdung durch Malware kann durch mehrere Schwachstellen begünstigt werden, die vor allem durch den Benutzer (Unachtsamkeit mit Berechtigungen, Benützung des mobilen Gerätes durch Dritte), das System (Gerootete Geräte, Veraltetes System (fehlende Updates), Speicherkarte (SD Karte), Interner Telefonspeicher, Software und Softwarefehler) und das Gerät (WLAN, Bluetooth, MMS, 2-Wege Autorisierung auf einem Gerät, Mobilität) gegeben sind. (vgl Seite 21ff)

6.2.3 Relevanz

Malware kann großen Schaden beim Gerät und der Kommunikation sowie bei den Daten anrichten. Das Verhindern von Schadsoftware ist jedoch nicht einfach, da dazu alle installierten APPs gescannt werden müssen.

6.3 Reverse Engineering

Reverse Engineering beschreibt das Analysieren von existierender Soft- oder Hardware mit dem Ziel, deren Funktionsweise zu ermitteln. Es stellt nicht direkt einen Angriff auf das System dar, jedoch ist es eine Vorbereitung von Attacken, da unter anderem Schwachstellen gefunden werden können. Bei Software wird dies oft durch Dekompilierung durchgeführt. Dabei kann es sein, dass Details zum Programmcode oder den Sicherheitsmechanismen sowie Sicherheitslücken ersichtlich und bekannt werden. Ein Angreifer kann dieses Wissen verwenden, um eine eigene Anwendung zu schreiben, die für den Kommunikationspartner wie die originale Applikation aussieht, jedoch im Hintergrund anders funktioniert, bzw nicht dem vordefiniertem Arbeitszyklus folgt. Diese Anwendung kann unter Umständen andere Befehle über einen sicheren Tunnel direkt ins Rechenzentrum senden. (vgl Seite 29, 5.6.3)

6.3.1 Relevanz

Durch Reverse Engineering kann dem Rechenzentrum eine funktionierende APP vorgespielt werden. Jedoch werden im Hintergrund Sicherheitüberprüfungen, die eigentlich von der APP durchgeführt werden sollten, umgangen. Somit ist es wichtig, dass entweder die Integrität der APP gewährleistet werden kann, oder die Daten, die an das Rechenzentrum gesendet werden, vom Server geprüft werden.

6.4 Physikalischer Zugriff

Wird das Gerät gestohlen, hat ein Angreifer direkten Zugriff auf dieses. Sind keine weiteren Sicherheitsabfragen konfiguriert (Passwort oder Mustereingabe bei Reaktivierung) kann er somit alle auf dem Gerät gespeicherten Daten auslesen und Anwendungen starten. Dateien auf der Speicherkarte können auch bei passwortgeschützten Geräten ausgelesen werden, da das Speichermedium von anderen Kartenlesern gelesen werden kann.

6.4.1 Gefährdete Werte

Durch den physischen Zugriff erhält der Angreifer Zugang auf alle gespeicherten Daten. Die Gefährdung betrifft die Benutzerdaten, den e-Banking Zugang und die Finanzmittel, sowie das Mobilgerät und die Anwendungsdaten. (vgl Seite 18ff, 4.1; 4.2; 4.3)

6.4.2 Genutzte Schwachstellen

Der physische Zugriff auf die Daten nutzt die Schwachstellen des Benutzer (Fehlende Tastensperre, Benützung des mobilen Gerätes durch Dritte), des Systems (Speicherkarte (SD Karte), Interner Telefonspeicher) und des Gerätes (Gebrauchsspuren, 2-Wege Autorisierung auf einem Gerät, Mobilität). (vgl Seite 21ff)

6.4.3 Relevanz

Durch den Diebstahl oder Verlust das Gerätes und wenn sich ein Angreifer das Gerät vom Eigentümer ausleiht, erhält dieser direkten Zugriff auf das Gerät. Da die mobilen Geräte klein sind, ist die Wahrscheinlichkeit, dass eine fremde Person Zugriff erhält, erhöht. Es ist daher wichtig, dass auch bei direktem Zugriff auf das Gerät, keine geheimen Daten ausgelesen werden können.

6.5 Shoulder Surfing

Shoulder Surfing beschreibt das Beobachten eines Benutzers zur Erlangung von privaten oder geheimen Informationen wie Passwörter, PINs und Zugangsdaten. Der Angreifer steht dazu meist im Hintergrund und beobachtet, filmt oder fotografiert den Benutzer. Dies kann sowohl direkt oder

auch mit speziellen Beobachtungsgeräten, wie Überwachungskameras oder Fernrohren, durchgeführt werden.

6.5.1 Gefährdete Werte

Durch Shoulder Surfing sind die Daten, die am Monitor angezeigt oder eingegeben werden, gefährdet. Dies betrifft die e-Banking Zugangsdaten (Passwörter, PIN, TAN) und die Kontoinformationen. (vgl Seite 18, 4.1)

6.5.2 Genutzte Schwachstellen

Shoulder Surfing wird durch Unwissen des Benutzers und der Mobilität des Endgerätes begünstigt. (vgl Seite 21ff)

6.5.3 Relevanz

Der Schutz gegen Shoulder Surfing ist zwar wichtig, jedoch ist es auch aufgrund des kleinen Displays eines Smartphones für einen Angreifer relativ schwierig, wichtige Daten auszulesen. Die Eingabe eines Passwortes auf der Bildschirmtastatur kann nur schwer abgeschaut werden. Anders verhält es sich jedoch bei einem Wischmuster vor allem dann, wenn es sich um eine einfache Kombination handelt.

6.6 Manipulation der Hardware

Wenn ein Angreifer im Besitz der Hardware ist, kann er diese nach Belieben verändern und die Speicher auslesen. Er kann die Hardwarekonfiguration ändern. Gelangt das Gerät zurück zu seinem Besitzer, kann der Angreifer abhängig von den Änderungen in der Konfiguration Zugriffe auf wichtige Daten erhalten. Dabei wird die Sperre des Betriebssystems umgangen, wodurch sich für den Angreifer zusätzliche Hintertüren zu den Benutzerdaten öffnen können.

6.6.1 Gefährdete Werte

Je nach Veränderung oder Fehler der Hardware sind einerseits die Mobilgeräte (Seite 20, 4.2) und andererseits die Kommunikation (Seite 20, 4.4) gefährdet.

6.6.2 Genutzte Schwachstellen

Um Hardwarefehler zu nutzen oder die Hardware zu manipulieren, werden das Unwissen des Benutzers, die Hardwarefehler und die Mobilität des Smartphones ausgenutzt. (vgl Seite 21ff)

6.6.3 Relevanz

Die Relevanz von Hardwaremanipulationen ist eher gering, da es extrem schwierig ist, die Hardwarekonfiguration eines mobilen Gerätes zu verändern. Die Hardware ist dazu viel zu kompakt verbaut. Ein mobiles Gerät bietet keinen freien Platz für zusätzliche Bauteile. Außerdem ist es schwierig, Wissen über die Geräte zu erlangen. Wahrscheinlicher sind hier Designfehlern des Herstellers, die von einem Angreifer für eine Attacke ausgenutzt werden können.

6.7 Lauschangriff

Ein Lauschangriff kann bei einem mobilen Gerät auf mehrere Arten durchgeführt werden. Das akustische Abhören von Gesprochenem wird in dieser Arbeit nicht behandelt, da es die Sicherheit von e-Banking nicht beeinflusst. Hier wird nur auf einen Angriff auf die Kommunikation näher eingegangen. Ein derartiger Angriff kann entweder durch eine Man in the Middle Attacke (vgl Seite 38, 6.11) oder durch Abhören der Funkverbindung ausgeführt werden. Da beim Senden über eine Funkverbindung alle benachbarten Geräte mithören können, können diese die gesendeten und empfangenen Pakete ebenfalls lesen und analysieren.

6.7.1 Gefährdete Werte

Ein Lauschangriff ermöglicht dem Angreifer abgehende und eingehende Pakete (vgl Seite 20, 4.4) mitzulesen.

6.7.2 Genutzte Schwachstellen

Um einen Lauschangriff durchführen zu können, bedient sich ein Angreifer bestimmter Software, sowie Soft- und Hardwarefehlern, der Funkverbindung (Luft als Übertragungsmedium, Freie Hotspots) und dem Internet (Anonymität; Vertrauen auf öffentliche Services (DNS, Proxies)). (vgl Seite 22ff)

6.7.3 Relevanz

Da ein Smartphone nur mittels Funk mit der Außenwelt kommunizieren kann, ist die Gefahr eines Lauschangriffes sehr hoch. Es muss sichergestellt werden, dass keine wichtigen oder geheimen Daten unverschlüsselt übertragen werden, da diese einfach von anderen Netzteilnehmern ausgelesen werden können. (vgl Seite 27, 5.4.1)

6.8 Positionsmessungen

Die einfachste Form von Positionsmessung auf einem mobilen Gerät ist die Verwendung des integrierten GPS-Moduls. Hier handelt es sich um ein Feature, das vom Gerät angeboten wird.

Durch Auswertung der Signalstärke, mit der ein Gerät sendet, bzw die am Zugriffspunkt aufgenommen wird, kann auch bei GSM, UMTS und WLAN ein Rückschluss auf die Position des mobilen Gerätes gezogen werden. Durch die Verwendung von drei Sendern kann die Position schon sehr genau berechnet werden. Mit einem einzelnen Sender kann zumindest ein Bereich in Form eines Kreisringes um den Sender oder bei gerichteten Antennen ein Sektor als Position festgelegt werden. (vgl [8] Seite 103)

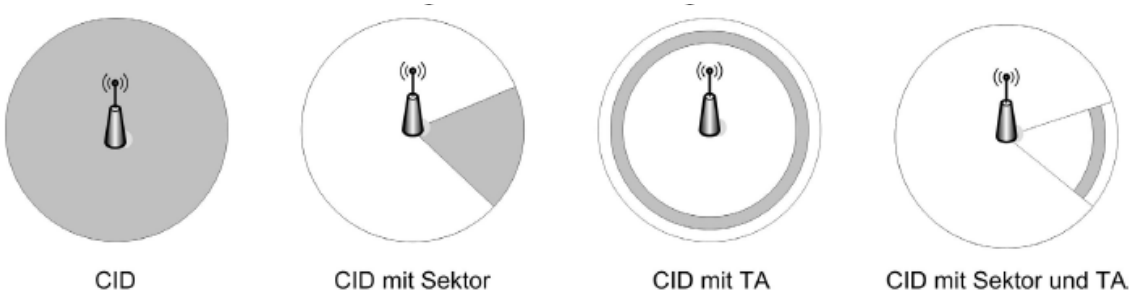


Abb. 17. Positionsmessung mit Cell ID (CID) und Timing Analysis (TA) [8](Seite 105)

6.8.1 Gefährdete Werte

Durch die Positionsmessung kann ein Angreifer die Standortdaten (vgl Seite 19, 4.1.3) auslesen.

6.8.2 Genutzte Schwachstellen

Positionsmessung wird durch die Schwachstellen der Funkverbindung, also das Übertragungsmedium Luft, freie Hotspots und GPS Module ermöglicht (vgl Seite 26, 5.4)

6.8.3 Relevanz

Die Relevanz der Positionsmessung ist eher gering, da sie wie im Fall von GPS sogar erwünscht ist. Die Positionsbestimmung anhand von GSM, UMTS und WLAN ist eher ungenau und kann vor allem bei vielen Personen nicht zur Identifizierung einer bestimmten führen.

6.9 Störsender

Für die Kommunikation wird der elektromagnetische Frequenzbereich in der Luft verwendet. Dieser ist stark begrenzt, jedoch kommunizieren alle Teilnehmer über dasselbe Medium. Sind zu viele Teilnehmer mit einer Basisstation verbunden, stören sich die Geräte gegenseitig und es kann dazu kommen, dass sich ein mobiles Gerät nicht mehr verbinden kann oder die Daten durch Interferenzen zerstört werden. Diese Tatsache kann auch absichtlich ausgenutzt werden. Ein Störsender sendet in einem bestimmten Frequenzbereich lautes Rauschen, wodurch keine Übertragung mehr möglich ist, da es laufend zu Kollisionen und Interferenzen kommt.

6.9.1 Gefährdete Werte

Der Störsender unterbindet jegliche Kommunikation. (vgl Seite 20, 4.4)

6.9.2 Genutzte Schwachstellen

Beim Angriff mittels Störsender sind Hardwarefehler des Smartphones und die Luft als Übertragungsmedium als Schwachstellen ausschlaggebend. (vgl Seite 24ff)

6.9.3 Relevanz

Es ist unmöglich, einen Angriff durch einen Störsender abzuwehren, sofern dieser alle Frequenzkanäle gleichzeitig stört. Ein Verhindern des Störeinflusses kann nicht von der APP übernommen werden. Weiters kann zwar die Kommunikation unterbunden werden, jedoch können vom Angreifer keine Daten ausgelesen und erhalten werden. Aus diesen Gründen ist die Relevanz eines Störsenders für die Sicherheitsbetrachtungen von e-Banking nicht gegeben.

6.10 DoS (Endgerät)

Eine weitere Möglichkeit, eine Kommunikation vom Endgerät zu unterbinden, ist ein DoS Angriff. Dafür gibt es mehrere Möglichkeiten. Durch das Senden einer Wiederverbindungsanfrage kann durchgehend eine Neuverbindung des Gerätes veranlasst werden. Weiters ist es möglich, dass eine Basisstation durch viele wiederholte unautorisierte Anmeldeversuche für andere Kommunikationsteilnehmer unerreichbar wird. Durch eine DoS Attacke auf das Bluetooth Interface kann das mobile Gerät bei schlechter Implementierung des Bluetooth Stacks zum Absturz gebracht werden. [7](Seite 20)

6.10.1 Gefährdete Werte

Da ein DoS Angriff nicht wie ein Störsender nur den Kommunikationskanal verrauscht, sind durch diese Gefährdung einerseits die Mobilgeräte und andererseits die Kommunikation betroffen. (vgl Seite 20, 4.2; 4.4)

6.10.2 Genutzte Schwachstellen

DoS Angriffe bedienen sich unterschiedlicher Schwachstellen. Je nach Art des Angriffs können sie das System (Gerootete Geräte, Veraltetes System (fehlende Updates), , Software und Softwarefehler) das Smartphone (WLAN, Bluetooth, MMS, Hardwarefehler, Mobilität) oder die Funkverbindung Luft als Übertragungsmedium, Freie Hotspots)) betreffen. (vgl Seite 22ff)

6.10.3 Relevanz

Wie der Angriff durch einen Störsender kann ein DoS oder DDoS Angriff auf ein mobiles Gerät nicht durch eine APP verhindert werden. Da jedoch auch durch eine DoS Attacke der Angreifer keine Daten erhalten kann, ist sie für die Sicherheitsbetrachtung von e-Banking nicht relevant.

6.11 Man in the Middle

Da der Kommunikationspfad nicht genau vorgegeben ist, durchlaufen die einzelnen Datenpakete unterschiedliche Teile des Netzes, die weder im Zugriffsbereich des Mobilfunkanbieters noch in jenem des Rechenzentrumsbetreibers liegen. Es ist jederzeit möglich, dass jemand während der Kommunikation die Pakete abfängt und sie analysiert und die enthaltenen Daten weiterverwertet. Da bei schneller Bewegung des mobilen Gerätes (Auto, Zug) die einzelnen Pakete unterschiedliche Wege nehmen können und viel Traffic auf den Leitungen übertragen wird, ist es unwahrscheinlich, dass ein Angreifer sinnvolle Daten auslesen kann.

Problematisch wird es dann, wenn der Angreifer es schafft, dass der gesamte Traffic der Kommunikation über ihn geroutet wird. (zB Proxy, Gateway) Er kann in diesem Fall die gesamten Datenpakete mitlesen. Sind diese verschlüsselt, besteht bei höherer Anzahl von Paketen eine höhere Chance, dass er Schwächen im Verschlüsselungsalgorithmus ausnutzen und sie so entschlüsseln kann. Unverschlüsselte Pakete liegen im Klartext vor. Ein Man in the Middle Angriff kann auf unterschiedliche Arten durchgeführt werden. Je nach Art werden andere Schwachstellen ausgenutzt. In dieser Arbeit werden folgende Möglichkeiten behandelt:

- Zusätzlicher WLAN Zugriffspunkt
- IMSI Catcher
- Proxymanipulation
- DNS-Manipulation

6.11.1 Gefährdete Werte

Alle Methoden der Man in the Middle Attacke zielen darauf ab, die Pakete der Kommunikation (vgl Seite 20, 4.4) abzuhören bzw zu verändern.

6.11.2 Genutzte Schwachstellen

Ein Man in the Middle Angriff basiert immer auf der Anonymität im Internet (vgl Seite 27, 5.5.1). Für die Kommunikationspartner ist die Identität der Geräte und Services, die zusätzlich an der Kommunikation beteiligt sind, nicht bekannt oder basiert auf Vertrauen.

Die folgenden Kapitel gehen näher auf spezielle Varianten eines Man in the Middle Angriffs ein.

6.11.3 Zusätzlicher WLAN Zugriffspunkt

Bei der Speicherung eines WLAN-Zugangs wird nur die SSID gespeichert. Ist ein entsprechendes WLAN aktiv, versucht das Gerät sich sofort zum Netzwerk zu verbinden, sobald dieses in Reichweite ist. Ein Angreifer kann nun einen eigenen Access Point (AP) in den Einzugsbereich des Benutzers bringen. Das Gerät erkennt das Netzwerk als ein bekanntes und verbindet darauf. Ein Smartphone priorisiert eine WLAN Verbindung für das Internet gegenüber der 3G oder der GSM Verbindung. Somit werden danach alle Daten über das Angreifernetz übertragen. Der Angreifer kann nun die Daten ohne WLAN Verschlüsselung mitlesen. Dieser Angriff ist das WLAN Gegenstück zum IMSI Catcher im GSM Bereich. (vgl Seite 39, 6.11.4)

6.11.3.1 Genutzte Schwachstellen

Für den Angriff durch einen WLAN Zugriffspunkt sind zusätzlich zur Anonymität im Internet noch die Schwachstellen Luft als Übertragungsmedium und die freien Hotspots von Relevanz. (vgl Seite 26ff)

6.11.3.2 Relevanz

Da die Kommunikation sehr häufig über WLAN läuft und auch die Verfügbarkeit von öffentlichen Hotspots zunimmt, ist die Relevanz dieser Accesspoints sehr hoch. Es kann vom Benutzer oft nicht mehr genau gesagt werden, wer der Betreiber eines WLANs ist. Bei Hotspots besteht weiters die Gefahr eines Lauschangriffs, auch wenn dieser mit einer Verschlüsselung betrieben wird. (vgl Seite 35, 6.7)

6.11.4 IMSI Catcher

Bei der Kommunikation in einem Mobilfunknetz verbindet sich das mobile Gerät (MS - mobile Station) mit dem Base Station Subsystem (BSS). Dabei wird die am stärksten sendende Station verwendet. Bei der Authentifikation muss sich die MS mit der IMSI und der IMEI bei dem BSS melden, aber nicht umgekehrt. Damit kann die BSS sicher sein, wer sich bei ihr anmeldet, jedoch ist es für das Endgerät nicht klar, ob auch das Netz jenes ist, das es vorgibt zu sein.

Bei einem IMSI Catcher wird diese Eigenheit der Systemarchitektur von GSM ausgenutzt. Der Catcher gibt vor, eine BSS zu sein, und lässt die MS zu sich verbinden. Danach fordert der Catcher das Endgerät auf, die Kommunikation mit dem A5/0 Algorithmus zu verschlüsseln. Dadurch erfolgt die Kommunikation unverschlüsselt und kann problemlos ausgelesen werden. Diese Methode kann

einerseits verwendet werden, um den Datentransfer abzuhören (vgl Seite 38, 6.11), und andererseits, um Geräte zu verfolgen und deren Position zu bestimmen. [8] (Seite 21)

IMSI Catcher werden von polizeilicher Seite verwendet, um Geräte abzuhören und deren Position zu verfolgen, ohne dass dabei der Provider miteinbezogen werden muss.

Die Ortung wird durch Verwendung von UMTS erschwert, da bei dieser Technologie anstatt der statischen IMSI die dynamische XEMSI oder TEMSI verwendet werden. [8](Seite 44) Diese Werte werden beim Aufbau der Kommunikation zwischen der Basisstation und dem Endgerät basierend auf der IMSI und IMEI berechnet. Sie sind für jede Übertragung unterschiedlich.

6.11.4.1 Genutzte Schwachstellen

Ein IMSI Catcher bedient sich der Luft als Übertragungsmedium (vgl Seite 27, 5.4.1), um sich in die Kommunikation einzuklinken.

6.11.4.2 Relevanz

Der IMSI Catcher funktioniert zwar nur in GSM Netzen effektiv, kann aber mit Hilfe eines Störsenders auch in UMTS Netzen eingesetzt werden. Durch den Störsender wird das UMTS Netz derart gestört, dass das mobile Gerät sich zum GSM Netz verbindet. Diese Abstufung stört zwar nicht bei Telefonaten, führt jedoch zu langsamen Datenverbindungen. Somit ist zwar ein Abhören auch in modernen Netzen möglich, jedoch kann der Angriff von einem Benutzer leichter bemerkt werden. Da die Ausrüstung für diese Attacke teuer ist, ist die Relevanz eher gering.

6.11.5 Proxymanipulation

Ein Proxy ist ein Rechner, der die Daten von einen Gerät (Mobilgerät) annimmt und an den anderen Gesprächsteilnehmer (Rechenzentrum) weiterleitet. Dadurch wird die Kommunikation in zwei Teile aufgeteilt, einerseits in die Kommunikation Gerät-Proxy und andererseits in Proxy-Rechenzentrum. Dadurch entstehen Vor- aber auch Nachteile.

Ein Proxy kann Daten zwischenspeichern, wodurch beim Surfen der Seitenaufbau beschleunigt werden kann, da die Daten nicht erneut vom Webserver abgefragt werden müssen. Bei dynamischen Inhalten kann es jedoch passieren, dass eine veraltete Version vom Proxy zurückgeliefert wird.

Ein weiteres Problem besteht dadurch, dass der Proxy eine Endstelle eines verschlüsselten Kanals ist. Auch wenn die Anwendung mittels https, SSL oder anderen Technologien getunnelt wird, wird dieser Tunnel beim Proxy beendet. Ein Mitlesen und Analysieren der Daten ist somit für den Proxybetreiber ohne Probleme möglich. [8]

Proxies werden oft verwendet, um Webseiten zwischenzuspeichern, oder um sie zu optimieren. Damit kann Transfervolumen gespart werden, da Bilder vorzeitig verkleinert werden können. Von bestimmten Browsern auf den mobilen Geräten werden standardmäßig bereits Proxies verwendet, um Bilder und Videos zu filtern und deren Größe zu reduzieren.

Ein weiterer Grund für die Verwendung von Proxies ist die Verschleierung der eigentlichen Herkunft. Damit kann ein Server nicht mehr genau nachvollziehen, von wo eine Anfrage gesendet wurde. Er sieht nur mehr den Proxy als Sender.

6.11.5.1 Genutzte Schwachstellen

Proxies können auf verschiedene Arten eingesetzt werden. Es können somit unterschiedliche Subsets der Schwachstellen ausschlaggebend sein. Angriffe über Proxies basieren auf dem Unwissen der Benutzer, gerooteten Geräten sowie deren Internetzugang und der Software und deren Fehler. Auch das Vertrauen auf öffentliche Services und freie Hotspots können eine Schwachstelle für einen derartigen Angriff darstellen. (vgl Seite 21ff)

6.11.5.2 Relevanz

Auf mobilen Geräten ist es wichtig, möglichst wenig Daten empfangen zu müssen, um einerseits Strom und andererseits Datenvolumen zu sparen. Somit ist die Relevanz in Bezug auf die Sicherheit von e-Banking sehr hoch, da sie aufgrund der genannten Gründe auf Smartphones häufig eingesetzt werden.

6.11.6 DNS-Manipulation

Ein DNS Server dient der Auflösung des Servernamens in eine IP Adresse. Durch diese Technologie kann einerseits Ausfallssicherheit und Lastverteilung ermöglicht werden und andererseits werden dadurch die URLs der Homepages für den Menschen leichter lesbar und merkbar.

Es besteht jedoch die Möglichkeit, dass die Adressen falsch aufgelöst werden, und der Benutzer so zu einer falschen Seite bzw zu einem falschen Server weitergeleitet wird. Diese Möglichkeit kann genutzt werden, um Login-Informationen und andere Daten vom Benutzer zu erhalten, ohne dass dieser davon etwas bemerkt.

6.11.6.1 Genutzte Schwachstellen

Es werden Schwachstellen des Systems (Gerootete Geräte, Veraltetes System (fehlende Updates), , Software und Softwarefehler), der Funkverbindung (Freie Hotspots) und des Internets (Anonymität, Vertrauen auf öffentliche Services (DNS, Proxies)) ausgenutzt. (vgl Seite 22ff)

6.11.6.2 Relevanz

Das Domain Name Service wird für jede Anfrage benötigt, damit der Namensstring in eine IP-Adresse umgewandelt werden kann. Für einen Benutzer ist es im Normalfall nicht möglich, den DNS Server zu überprüfen. Er muss darauf Vertrauen, dass er vom Provider (Mobilfunkprovider oder WLAN) einen vertrauenswürdigen Server zugewiesen bekommt. Aus diesem Grund ist es vor allem für e-Banking sehr wichtig sich gegen DNS Manipulation zu schützen.

6.12 DoS, DDoS (Rechenzentrum)

Bei einem DoS Angriff auf ein Rechenzentrum werden sehr viele Anfragen gestellt, die von den Servern abgearbeitet werden müssen. Diese Anfragen können verschiedene Teile des Systems lahmlegen. Es kann die Anbindung des Rechenzentrums an das Internet überlastet werden. Aufgrund sehr schneller Anbindungen und redundanter Links ist das heutzutage nur mehr erschwert möglich. Wahrscheinlicher ist es, dass die Firewall oder die IDS oder IPS durch die Vielzahl an Pakete nicht mehr mit der Abarbeitung fertig wird. Eine weitere Möglichkeit ist, dass die Server durch die Anfragen überlastet werden.

Da die Systeme immer leistungsfähiger werden, ist die Chance, dass eine reine DoS Attacke funktioniert, immer geringer. Deshalb wird häufig eine DDoS (Distributed Denial of Service) Attacke durchgeführt. Dabei führen mehrere Clients parallel über mehrere Kanäle DoS Angriffe durch.

6.12.1 Gefährdete Werte

Ein DoS Angriff führt zur Beeinträchtigung der Kommunikation und des Rechenzentrums. (vgl Seite 20, 4.4; Seite 20, 4.5)

6.12.2 Genutzte Schwachstellen

Der Internetzugang und Softwarefehler des Rechenzentrums ermöglichen bzw erleichtern einen DoS Angriff auf dieses. (vgl Seite 28ff)

6.12.3 Relevanz

Ein Ausfall des Rechenzentrums führt zwar nicht direkt zum Verlust von wichtigen oder geheimen Daten, jedoch wird die Verfügbarkeit und somit das Vertrauen der Kunden reduziert. Es ist wichtig, dass ein ausfallsfreier Betrieb gewährleistet werden kann. Es kann jedoch auch sein, dass die DoS Attacke dazu verwendet wird, um den eigentlichen Angriff von dem IDS System zu verstecken. Da dieses durch die Paketabarbeitung überlastet ist, kann die Attacke nicht oder nur schwer aus dem

Traffic gefiltert werden. Somit ist die Relevanz des Schutzes vor DoS Attacken gegen das Rechenzentrum hoch.

6.13 Physischer Zugriff auf Server und Rechenzentrum

Neben der Absicherung gegen Angriffe aus dem Netzwerk bzw dem Internet ist es auch wichtig, den direkten Zugriff auf das Rechenzentrum, dessen Netzwerkhardware und Server abzusichern. Die Speichermedien werden bei steigender Kapazität immer kleiner. Sie können aufgrund ihrer Größe leicht ein- bzw ausgeschleust werden und dabei eine große Anzahl an Datensätze speichern. Bei direktem Zugriff auf den Server können viele Sicherheitsmaßnahmen, die das Einbrechen über das Netzwerk verhindern sollen, umgangen werden.

6.13.1 Gefährdete Werte

Durch den physischen Zugriff auf das Rechenzentrum werden die Kommunikation und das Rechenzentrum gefährdet. (vgl Seite 20, 4.4; 4.5)

6.13.2 Genutzte Schwachstellen

Der physische Zugriff wird durch die Existenz des Rechenzentrums (vgl Seite 28, 5.6.1) ermöglicht.

6.13.3 Relevanz

Ein physischer Zugriff ermöglicht einem Angreifer, direkt auf die Hardware und somit auch direkt auf das System zuzugreifen. Jegliche Sicherheitsvorkehrungen, die vorgenommen wurden, um Angriffe aus dem Internet abzuwehren, können so umgangen werden. Deshalb ist es sehr wichtig, auch die Server selber vor unberechtigtem Zugriff zu schützen.

6.14 Zugriff auf das Rechenzentrum durch den Kommunikationstunnel

Durch die Verschlüsselung der Kommunikation zum Rechenzentrum wird ein virtueller Tunnel aufgebaut, da die Geräte, die sich zwischen den Endstellen befinden, die Kommunikation nicht mitlesen können. Das hat den Vorteil, dass damit ein Angreifer nur verschlüsselte Pakete und keinen Klartext sieht. Es ist für ihn schwieriger bis unmöglich, in einer vernünftigen Zeit zu den transportierten Informationen zu gelangen.

Dieser Tunnel kann jedoch auch von Firewalls nicht ausgelesen werden. Es kann somit nicht kontrolliert werden, ob es sich noch um den zu erwartenden Traffic handelt, oder ob ein Angriff durch den Tunnel stattfindet.

Wird die Verschlüsselung vor dem Server bereits beendet und eine weitere Firewall dazwischengeschaltet, kann diese auch den Traffic im Tunnel überprüfen.

6.14.1 Gefährdete Werte

Wird über den Kommunikationstunnel auf das Rechenzentrum zugegriffen, kann das zu einer Beeinträchtigung der Server und Services sowie zum Verlust von Daten führen. (vgl Seite 20, 4.5)

6.14.2 Genutzte Schwachstellen

Für den Zugriff auf das Rechenzentrum durch den verschlüsselten Tunnel sind der Internetzugang und der Kommunikationstunnel oder ein Softwarefehler notwendig. (vgl Seite 28ff, 5.6.2, 5.6.3, 5.6.4)

6.14.3 Relevanz

Da die Sicherheitssysteme nicht in den Tunnel hineinsehen können, ist es für einen Angreifer möglich, durch diesen Tunnel die Sicherheitsmechanismen zu unterwandern und so direkt auf den Server zugreifen zu können. In Kombination mit Software- oder Konfigurationsfehlern kann ein weiterer Zugriff ermöglicht werden. Somit ist es für die Sicherheit wichtig, dass das Rechenzentrum vor Angriffen aus dem Kommunikationstunnel geschützt wird.

6.15 Social Engineering (Rechenzentrum)

Nicht nur der Endkunde ist anfällig gegen Sozial Engineering, sondern auch der Mitarbeiter im Rechenzentrum. In vielen Fällen ist es nicht direkt ersichtlich, ob Informationen, die weitergegeben werden, wichtig sind oder nicht. Die Mitarbeiter verfügen in vielen Fällen über das nötige Know How, um die üblichen Methoden von Social Engineering erkennen zu können (vgl Seite 30, 6.1). Als problematisch stellen sich jedoch hier Gespräche mit Arbeitskollegen, Freunden oder in der Familie heraus. Auch Telefonanrufer oder Mails, die vorgeben, aus der Firma zu sein und dringend bestimmte Daten zu benötigen, können zu einem Informationsverlust führen.

6.15.1 Gefährdete Werte

Durch Social Engineering bei den Mitarbeitern des Rechenzentrums sind vor allem die Kommunikation und das Rechenzentrum selber gefährdet. (vgl Seite 20, 4.4; 4.5)

6.15.2 Genutzte Schwachstellen

Social Engineering basiert vor allem auf den Schwachstellen des Rechenzentrums (Internetzugang, Physische Existenz, Softwarefehler). (vgl Seite 28ff)

6.15.3 Relevanz

Sozial Engineering ist ein wichtiges Thema und ist auch für die Sicherheit im Rechenzentrum von hoher Relevanz. Ein Grund dafür ist, dass es nicht immer möglich ist, die Gegebenheiten so zu ändern, dass die bekannt gewordene Information ungültig wird. (zB Netzwerktopologie, Sicherheitssysteme) Systeme, die durch das Prinzip von Security by Obscurity geschützt sind, können dadurch sehr leicht angreifbar sein. Dieses Prinzip basiert auf der Verschleierung und Geheimhaltung der Netzwerk- und Topologiedaten.

6.16 Ausnutzen von Serversoftware- und Konfigurationsfehlern

In jeder Software befinden sich irgendwo Fehler. Inwieweit sie die problemlose Ausführung der Applikation beeinträchtigen oder nicht, kann nicht von Anfang an gesagt werden. Ebenso bietet auch jede Konfiguration Angriffsstellen, die ein Eindringen ermöglichen können. Gelingt es einem Angreifer, eine oder beide davon auszunutzen, kann er unter Umständen Zugriff auf das System oder die Daten erhalten. Ein Beispiel für bekannte sicherheitskritische Fehler ist der Apache Webserver in der Version bis 1.3.24 und Apache2 bis 2.0.36.[35] Als Beispiel für einen Angriff auf einen Fehler in der Applikation ist die SQL Injection anzuführen.

6.16.1 Gefährdete Werte

Das Ausnutzen von Fehlern in der Software oder Konfiguration des Rechenzentrums betrifft vor allem die Services, die auf den betroffenen Servern laufen. (vgl Seite 20, 4.5)

6.16.2 Genutzte Schwachstellen

Der Internetzugang und Softwarefehler erleichtern bzw ermöglichen ein Ausnutzen der Fehler. (vgl Seite 28ff)

6.16.3 Relevanz

Für die Stabilität des Systems ist es von hoher Relevanz, dass es für einen Angreifer nicht möglich ist, Software- und Konfigurationsfehler auszunutzen. Es besteht bei einem Angriff ein hohes Potential an Datenverlust. Außerdem ist die Verfügbarkeit der Services gefährdet.

7. Existierende Systeme für e-Banking auf mobilen Geräten

Der Androidmarkt ist noch verhältnismäßig jung. Die normale e-Banking Anwendung, die am Computer im Browser aufgerufen wird, kann auch am Smartphone verwendet werden. Da die mobilen Geräte einen sehr viel kleineren Monitor besitzen, ist die Nutzung der existenten Portale schwierig und meist mit viel Zoomen und Scrollen verbunden. Weiters sind diese Webseiten nicht

für derartige Geräte optimiert und funktionieren aufgrund von fehlenden Ressourcen (Speicher, Rechenleistung, Bandbreite) langsam.

Einige Banken haben schon darauf reagiert und angepasste Versionen veröffentlicht. Diese sind sehr unterschiedlich, lassen sich jedoch in zwei Kategorien einteilen: Browserbasiere Websites oder eigene APPs. Auch der Funktionsumfang ist unterschiedlich. Die Banking-APP der Erste Bank und Sparkassen Österreich unterstützt Netbanking noch nicht. Alle anderen unterstützen zumindest die Basisfunktionen von den e-Banking-Portalen am Computer. Manche nutzen die zusätzlichen Funktionen der Smartphones, um weitere Funktionen wie einen Bankomat- oder Filialfinder anzubieten.

Bei der Verwendung von e-Banking ist es wichtig, dass die Daten sicher vom Client zum Server übertragen werden. Sicherheit bedeutet in diesem Fall:

- Vertraulichkeit der Daten: Die Informationen sollen unlesbar für Dritte übertragen und verarbeitet werden.
- Integrität der Daten: Es ist wichtig, dass die Informationen unverändert beim Server bzw beim Endgerät ankommen. Ansonsten wäre es möglich, dass zwischen den Kommunikationspartnern zB die Zielkontonummer ausgetauscht wird.
- Verfügbarkeit des Systems: Die Services sollten möglichst immer verfügbar sein, um Überweisungen und Ver- und Einkäufe an der Börse zeitgerecht durchführen zu können.
- Zurechenbarkeit der Daten: Es muss möglichst genau bestimmbar sein, wer was in Auftrag gegeben hat.
- Authentizität der Systemkomponenten: Vor allem beim Server ist es wichtig, dass er sich gegenüber dem Client authentifizert. Erst danach kann der Benutzer sicher sein, dass er wirklich mit seiner Bank kommuniziert und nicht mit dem Server von Hackern.
- Zuverlässigkeit: Es muss gewährleistet werden, dass das System den Anforderungen entsprechend korrekt arbeitet und Fehler möglichst ausgeschlossen sind.

7.1 Weboberfläche (ELBA internet)

Die e-Banking Plattform ELBA wird für viele österreichische Banken von der RACON Software GmbH in Linz entwickelt. Für den Computer ist sie seit vielen Jahren verfügbar und beinhaltet viele Sicherheitsfunktionen, um das Kontomanagement für den Benutzer möglichst angenehm und risikofrei zu gestalten. Die Oberfläche ist für große Monitore optimiert und bietet dort eine gute Übersicht und schnelle Navigation.

Abb. 18. ELBA Internet – Übersicht [15]

7.1.1 Funktionen

ELBA bietet passende Funktionen für viele Bankangelegenheiten, sodass ein Besuch in einer Bankfiliale nur mehr selten notwendig ist. Die Funktionen können in drei Bereiche untergliedert werden:

7.1.1.1 Kontofunktionen

In den Kontofunktionen können die aufgelisteten Konten bearbeitet werden. Es können die Kontodaten wie Zinssätze, Kontostand und Währung abgerufen werden. Zusätzlich ist es möglich, die Zahlungsgeschäfte durchzuführen. Die Überweisungen müssen mittels TAN bestätigt werden. Alternativ steht auch die Möglichkeit einer digitalen Signatur zur Verfügung. Dazu wird jedoch in den meisten Fällen zusätzliche Hardware benötigt.

Bei der Verwendung einer TAN gibt es iTAN, mTAN oder cardTAN. Die Möglichkeiten sind von den Einstellungen und Freischaltungen abhängig.

Alle Funktionen sind durch eine Navigationsleiste auf der linken Seite direkt verfügbar.

7.1.1.2 Wertpapierfunktionen

Eine weitere Funktionalität ist die Verwaltung von Wertpapieren in den Depots. ELBA ermöglicht die Ansicht der Kursinformationen und den Handel. Die Transaktionen müssen mit einer TAN bestätigt werden. Hier stehen dieselben Möglichkeiten zur Bestätigung der Aufträge zur Verfügung wie bei den Kontofunktionen.

7.1.1.3 Weitere Funktionen

Zusätzlich zu den genannten Funktionen bietet ELBA auch noch weitere Services. Es ist möglich, spezielle Onlineangebote zu aktiveren und zu nutzen. Dazu zählen Onlinekonten und Depots aber auch Onlinekredite.

Weiters wird ein Nachrichtenservice angeboten, der es ermöglich, mit dem Bankberater über einen geschützten Kanal Nachrichten auszutauschen. Bei Erhalt einer neuen Nachricht wird zwar eine e-Mail an die Mailadresse des Kunden gesendet, jedoch dient diese nur als Mitteilung und beinhaltet keinen Inhalt der Banknachricht.

Eine weitere Funktion ist die Administration des ELBA Zugangs. Hier kann zB die PIN geändert werden, um einem eventuellen PIN-Diebstahl schnell und ohne Umwege entgegenzuwirken. Weiters kann hier ein persönliches Timeout konfiguriert werden. Eine ELBA Session wird standardmäßig nach 10 Minuten Inaktivität getrennt. Durch diese Einstellung kann die Zeit verlängert oder verkürzt werden.

7.1.2 Sicherheitsrisiken bei Verwendung der Bankapplikation am Smartphone

Die Verwendung am Smartphone birgt gewisse Sicherheitsrisiken in sich. Diese Risiken unterscheiden sich jedoch nicht sehr stark von denen am Computer.

Risiken entstehen hauptsächlich durch die Unwissenheit des Benutzers und durch Schadsoftware. ELBA Internet stellt diesbezüglich zwar die Möglichkeit der digitalen Signatur zur Verfügung, jedoch ist sie auf einem mobilen Gerät nicht verwendbar. Den Benutzern ist oft die Gefahr, die durch das vielseitige Betriebssystem Android ausgeht, nicht bewusst. Programme wie Viren- und Malwarescanner sind noch nicht sehr verbreitet.

Für ELBA Internet kann bereits die cardTAN verwendet werden. Diese ist jedoch noch nicht sehr stark verbreitet, bzw für alle Kunden freischaltbar. Durch die Verwendung der mTAN werden beide Kommunikationskanäle auf ein Gerät zusammengelegt. mTAN ist bei ELBA Internet eine als sehr sicher anerkannte Möglichkeit der Auftragsbestätigung, jedoch stellt sie bei der Verwendung am Smartphone ein Sicherheitsrisiko dar. (vgl Seite 25, 5.3.3)

Ein Smartphone wird meist unterwegs und somit in der Öffentlichkeit verwendet. Die Wahrscheinlichkeit eines Verlustes oder von Shoulder Surfing ist erhöht. Letzteres ist jedoch bei ELBA Internet erschwert möglich, da entweder die Schrift sehr klein und somit unlesbar ist oder nur ein kleiner Ausschnitt der Oberfläche zu sehen ist.

7.1.3 Implementierte Sicherheitsmechanismen

ELBA Internet verwendet viele Sicherheitsmechanismen, um die Durchführung der Bankgeschäfte möglichst sicher zu machen.[18] Dennoch wird auf die Usability geachtet und der Benutzer nicht mit unnötigen Sicherheitsmerkmalen genervt.

7.1.3.1 Verfügernummer, PIN

Zur Anmeldung an das System ist es wichtig, neben der Kontonummer und der Bankleitzahl auch die Verfügernummer und die PIN anzugeben. Damit kann festgelegt werden, wer sich an das System anmeldet und wer bestimmte Transaktionen durchführt. Weiters wird damit auch kontrolliert, wer welche Daten sehen darf. Diese Merkmale fördern die Zurechenbarkeit und Authentizität des Benutzers im System.

7.1.3.2 TAN Systeme

Die TAN Nummern sind notwendig, um eine Transaktion durchzuführen. Sie sollten über einen zweiten Kommunikationsweg übertragen werden und dienen der besseren Zurechenbarkeit und Authentizität des Benutzers. Es soll damit verhindert werden, dass Unberechtigte Zahlungen und Überweisungen durchführen. Die 2 Wege Autorisierung ist mit dem mTAN System nicht mehr gegeben, wenn ELBA auf dem Gerät ausgeführt wird, auf dem die SMS empfangen wird. Es besteht jedoch die Möglichkeit des Umstiegs auf cardTAN.

7.1.3.3 Zugangssperre bei mehrfacher Fehleingabe von PIN oder TAN

Wird die PIN oder eine TAN dreimal falsch eingegeben, so wird der ELBA Zugang gesperrt. Dadurch wird ein Brute-Force Angriff auf den Zugang verhindert. Es hilft, dass sich nur Berechtigte Anmelden können.

7.1.3.4 Digitale Signatur

Eine digitale Signatur ist vergleichbar mit einer virtuellen Unterschrift. Sie wird mit einem Zertifikat auf der Clientseite realisiert. Dieses Zertifikat ist auf einer Signaturkarte gespeichert und mit einer PIN geschützt. Sie kann somit nicht so einfach vom Benutzer an Dritte weitergegeben werden. Damit ist dieses System besser für die Zurechenbarkeit und Authentizität des Benutzers als PIN oder TAN. Ein Client Zertifikat ist jedoch meist auch mit Aufwand und Kosten verbunden und findet somit im privaten Bereich wenig Einsatz. Um die Signatur auszulesen, ist ein Kartenleser notwendig. Somit kann dieses Sicherheitsmerkmal nicht auf einem Smartphone verwendet werden. Eine digitale Signatur befindet sich unter anderem auf der Bürgerkarte. Diese muss jedoch vor der Verwendung aktiviert werden.

7.1.3.5 HTTPS

Das Internetprotokoll HTTPS verwendet zur Verschlüsselung das SSL Protokoll. Werden dafür Zertifikate verwendet (Serverzertifikate), ist es für Angreifer fast unmöglich, die Kommunikation abzuhören. Durch das Einspielen eines gefälschten Serverzertifikats kann jedoch eine Man in the

Middle Attack durchgeführt werden. (vgl Seite 38, 6.11) Dadurch wird sowohl die Integrität als auch die Vertraulichkeit der Daten gewährleistet. (vgl Seite 79, 8.4.2)

7.1.3.6 Mehrfache Firewalls, verteilte Systeme und serverseitige Pageerstellung

Die ELBA-Server werden durch mehrere Firewalls geschützt. Die benötigten Services (Webserver, Datenbanken, …) sind auf unterschiedliche Rechner (Server) verteilt. Wird ein Server angegriffen, hat das keinen oder nur wenig Einfluss auf die anderen. Wird ein Webserver von einem Angreifer übernommen, hat er noch keinen Zugriff auf die Datenbanken.

Da zwischen den Servern jedoch oft ein gegenseitiges Vertrauensverhältnis besteht, kann von einem System leichter auf ein anderes zugegriffen werden. In diesem Fall ist es wichtig, dass der Angriff schnell erkannt wird.

Die einzelnen Seiten der Anwendung werden serverseitig zusammengesetzt und als HTML an den Benutzer gesendet. Dynamische Teile sind mit JavaScript implementiert. Durch diesen Aufbau sind keine internen Strukturen und Datenbankzugriffe nach außen ersichtlich. Außerdem muss nur der Webserver auf die Datenbanken zugreifen können.

Durch diese Maßnahmen werden die Integrität und die Zuverlässigkeit des Systems erhöht.

7.1.3.7 Session-Timeout

Durch das Session-Timeout wird verhindert, dass unberechtigte Benutzer eine bestehende Verbindung weiter verwendet oder eine fehlerhaft beendete Session von Unberechtigten fortgeführt wird. Damit wird die Zurechenbarkeit erhöht, da die Wahrscheinlichkeit eines Sessionmissbrauchs verringert wird.

7.1.3.8 Umfangreiche interne Tests, Prüfung durch staatlich anerkannte und autorisierte Stellen

ELBA Internet wird intern umfangreich getestet. Weiters erfolgt eine Prüfung durch eine staatlich anerkannte und autorisierte Stelle.[18] Dadurch wird die Wahrscheinlichkeit eines Fehlers reduziert und eventuelle Sicherheitslücken frühzeitig erkannt, wodurch die Zuverlässigkeit und Verfügbarkeit des Systems erhöht wird.

7.1.3.9 Serverzertifikat

Die Server von ELBA Internet sind mit einem Zertifikat von Verisign, Inc. zertifiziert. Dieses bestätigt die Authentizität des Servers. Der Benutzer kann erkennen, dass er wirklich mit einem Bankserver kommuniziert und kann sich sicher sein, dass der Server derjenige ist, für den er sich ausgibt. Die Zertifizierung erschwert Phishing Attacken und erhöht die Authentizität, Integrität und Vertraulichkeit.

7.1.3.10 Awareness Politik

Der Benutzer wird durch wiederkehrende Mitteilungen und Hinweise auf eventuelle Sicherheitsrisiken hingewiesen. Ein Beispiel dafür ist, dass der Benutzer niemals seine Zugangsdaten oder TAN Liste weitergeben soll. Hält sich der Benutzer an die Vorgaben, führt es zu einer Steigerung der Vertraulichkeit und Zurechenbarkeit. Die Angriffe, die auf Unwissenheit des Benutzers oder auf Social Engineering basieren, können damit abgewehrt bzw verringert werden.

7.1.4 Angeforderte Berechtigungen

Die Applikation wird im Browser ausgeführt und besitzt somit alle Möglichkeiten, die aus dem Browser heraus möglich sind. Der Benutzer muss keine Berechtigungen bestätigen. Diese sind durch die Installation des Browsers schon vorher bestimmt worden.

7.1.5 Handhabung und Benutzerfreundlichkeit

Da ELBA Internet nicht für ein Smartphone oder Tablet entwickelt wurde, ist die Benutzerfreundlichkeit auf einem mobilen Gerät extrem eingeschränkt. Die Displays sind zu klein und somit die einzelnen Schaltflächen schwer zu erreichen. Weiters sind die Schriften sehr klein und die einzelnen Seiten stark verzerrt oder nur durch häufiges Scrollen und Zoomen lesbar.

Derzeit wird der Benutzer automatisch zur ELBA mobile Seite weitergeleitet, wenn ein Androidsystem erkannt wird.

7.1.6 Fazit zu ELBA Internet

ELBA Internet wurde zwar für den Browser am Computer entwickelt, kann aber auch auf einem Smartphone verwendet werden. Es bestehen fast dieselben Sicherheitsrisiken wie auf dem PC. Zu bedenken ist jedoch, dass bei der Verwendung der mTAN, die SMS auf dasselbe Gerät gesendet wird, auf dem sie schlussendlich eingegeben wird. Die Sicherheit der 2 Wege Autorisierung ist nicht mehr gegeben.

Die Benutzerfreundlichkeit ist sehr eingeschränkt, da der Bildschirm viel kleiner ist als der eines Computers.

7.2 ELBA mobil

Um die Benutzerfreundlichkeit von ELBA Internet auf dem Smartphone zu erhöhen, wurde die Website für die mobilen Geräte portiert. Eine neue Weboberfläche ermöglicht den schnellen und einfachen Zugriff auf die wichtigsten Funktionen von ELBA Internet. Im Hintergrund arbeitet

dieselbe Infrastruktur. Teilweise wird auch derselbe Code verwendet. ELBA mobil wird in der Raiffeisen Bankengruppe Österreich verwendet.

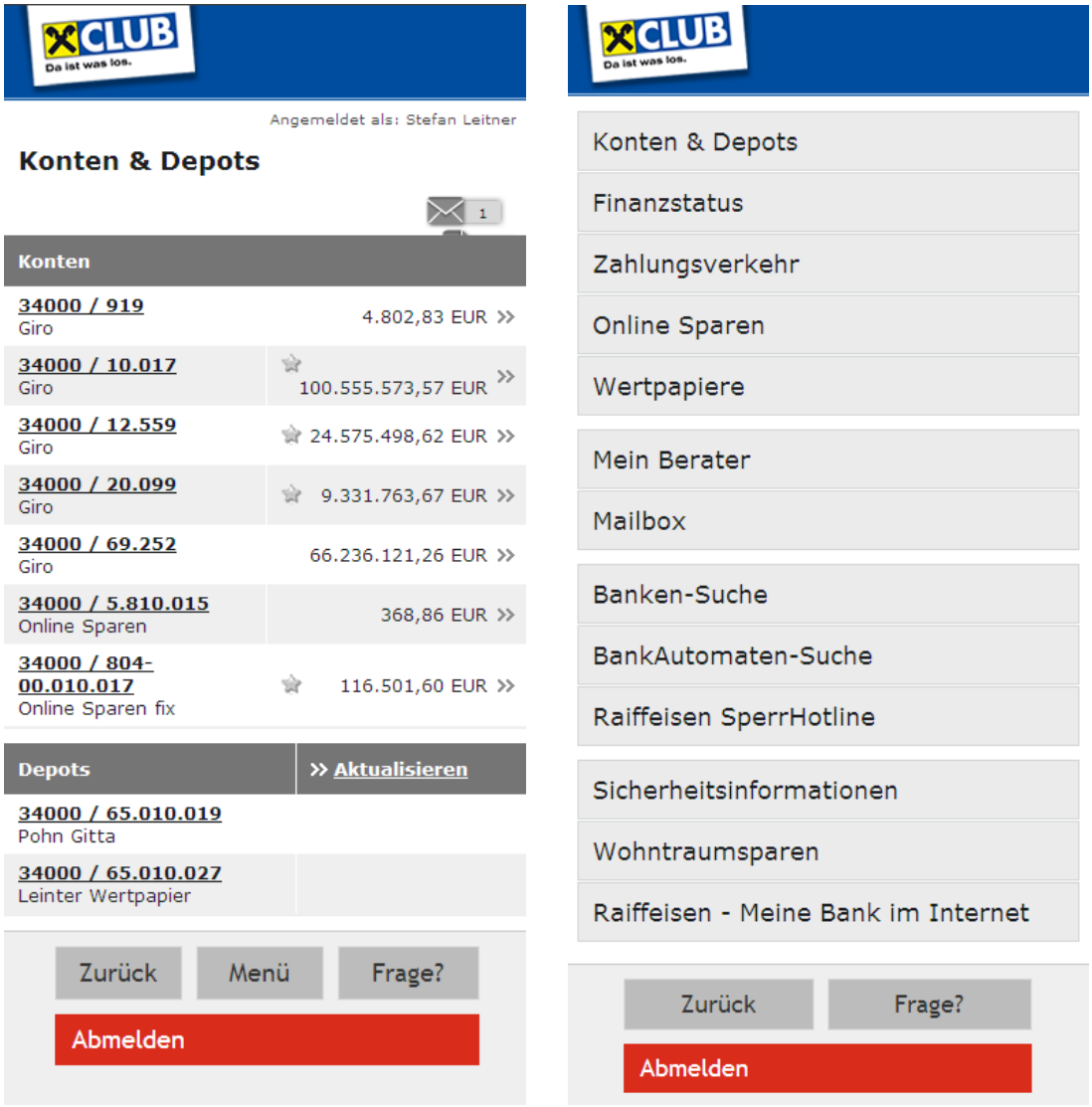


Abb. 19. ELBA mobil – Kontoübersicht, Menü [15]

7.2.1 Funktionen

Die Funktionen sind großteils mit denen von ELBA Internet identisch. Um die Übersichtlichkeit zu erhöhen, wurden zusätzliche Ansichten eingefügt und bestimmte Funktionen in einem Menü zusammengefasst. Funktionen, die eher selten gebraucht werden, sind nur über das Menü erreichbar.

7.2.1.1 Kontofunktionen

ELBA mobil ermöglicht die mobile Verwaltung des Kontos. Die wichtigste Funktion (Überweisung von einem bestimmten Konto) kann direkt aus der Kontoübersicht ausgewählt werden. Es können Überweisungsaufträge erstellt, gespeichert und abgesendet und Kontoinformationen ausgelesen werden. Daueraufträge oder Abbucher können jedoch nicht erstellt werden.

Bei der Umsatzübersicht ist der auswählbare Zeitbereich eingeschränkt. Dies fördert die Übersichtlichkeit und verhindert, dass eine sehr lange Liste geladen werden muss. Der Zeitbereich ist nicht manuell nach Datum auswählbar. Der Benutzer kann aus vier vorgegebenen Einstellungen wählen, die bis maximal zwei Monate vor dem aktuellen Datum liegen.

Für die Absendung von Transaktionen stehen nur mTAN und cardTAN zur Verfügung.

7.2.1.2 Wertpapierfunktionen

Auch die Verwaltung eines Depots ist mit ELBA mobil möglich. Es können bestehende Wertpapiere gekauft, verkauft und deren Kurs verfolgt werden. Auch der Kauf neuer Wertpapiere ist möglich. Es können jedoch keine Reports oder Statistiken erstellt und angezeigt werden. Das Orderbuch ist wie die Umsatzübersicht der Konten zeitlich eingeschränkt.

Für die Absendung von Transaktionen stehen nur mTAN und cardTAN zur Verfügung.

7.2.1.3 Messaging

ELBA mobil beinhaltet auch die Messaging Funktionen von ELBA Internet. Damit kann über einen sicheren Kanal mit dem Bankberater kommuniziert werden. Der Benutzer wird bei Erhalt einer neuen Nachricht zwar per Mail informiert, jedoch beinhaltet das Mail keine Daten aus der Message. Somit kann gewährleistet werden, dass das Gespräch nicht abgehört werden kann.

7.2.1.4 Filialfinder

ELBA mobil bietet dem Benutzer einen Filialfinder. Dieser ermittelt auf Basis von eingegebenen Daten wie Postleitzahl, Ortsname oder Adresse die Bankstellen, für die die Suchkriterien passen. Bei der Auswahl einer Bankstelle werden Informationen wie Telefonnummern, Adresse, Öffnungszeiten, e-Mail und BLZ bzw SWIFT/BIC ausgegeben.

7.2.1.5 Bankomatfinder

Das Smartphone hat Möglichkeiten, die am Computer nicht verfügbar sind. Dazu zählt das GPS Modul. Durch dieses Modul ist es möglich, das Gerät zu orten und somit die Position des Benutzers zu bestimmten. Dies kann einerseits ein Sicherheitsproblem darstellen (vgl Seite 35, 6.8), aber andererseits auch die Basis für weitere Anwendungen sein.

ELBA mobil bietet die Möglichkeit eines Bankomatfinders. Dabei wird die aktuelle Position des Gerätes mittels GPS ausgelesen und alle nahegelegenen Bankautomaten angezeigt.

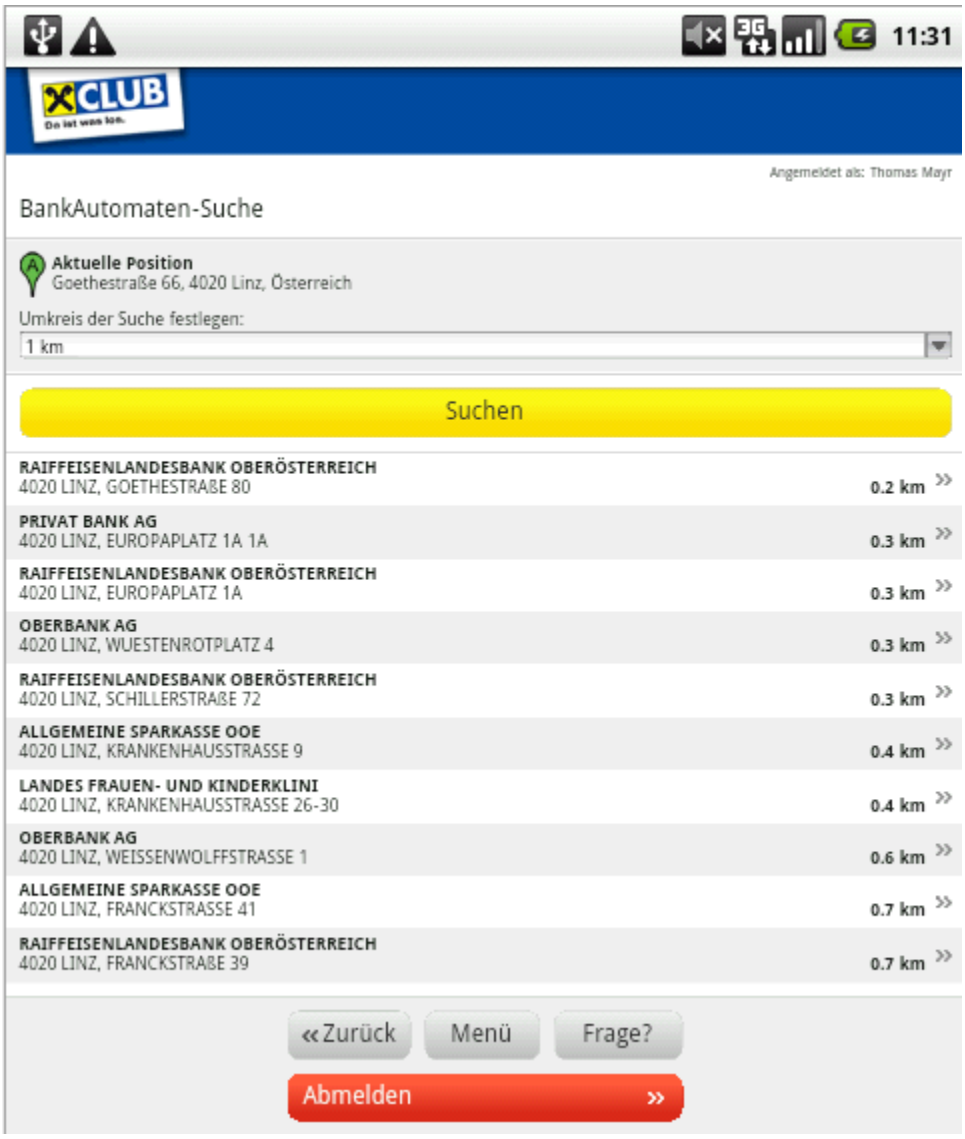


Abb. 20. ELBA mobil - Bankautomatfinder

7.2.2 Sicherheitsrisiken bei Verwendung am Smartphone

ELBA mobil wurde für das Handy entwickelt. Da es sich nicht um eine APP handelt, bestehen dieselben Sicherheitsrisiken wie für eine „normale“ Homepage. Der Funktionsumfang von ELBA mobil ist im Vergleich zu ELBA Internet eingeschränkt, jedoch sind die wichtigsten Funktionen verfügbar. Somit gelten ähnliche Sicherheitsrisiken für ELBA mobil wie für ELBA Internet. Jedoch ist zu beachten, dass die Auflösung für ein kleines Display optimiert wurde und somit die Informationen leichter durch Shoulder Surfing ausgelesen werden können.

7.2.3 Implementierte Sicherheitsmechanismen

Da es sich bei ELBA mobil um eine Portierung von ELBA Internet handelt, sind die Sicherheitsfunktionen von ELBA Internet auch hier verfügbar. Eine Ausnahme bilden die digitalen Signaturen.

Diese sind bei ELBA mobil nicht möglich, da bei einem Smartphone die Infrastruktur für einen Kartenleser fehlt.

7.2.4 Angeforderte Berechtigungen

Die Applikation wird im Browser ausgeführt und besitzt somit alle Möglichkeiten, die durch den Browser gegeben sind. Der Benutzer muss keine Berechtigungen bestätigen. Diese sind durch die Installation des Browsers bereits festgelegt worden.

7.2.5 Handhabung und Benutzerfreundlichkeit

Da ELBA mobil für das Smartphone entwickelt wurde, wurde auch auf eine problemlose Handhabung Rücksicht genommen. Deshalb ist die Nutzung ohne aufwändiges Scrollen oder Zoomen möglich. Umständlich ist jedoch, dass manche selten benutzten Funktionen nur durch das Menü aufrufbar sind, sowie dass nicht die gesamte Funktionalität von ELBA Internet portiert wurde. Diese Einschränkungen sind jedoch notwendig, um die Übersichtlichkeit und somit die Benutzbarkeit zu gewährleisten.

7.2.6 Fazit zu ELBA mobil

ELBA Internet ist nicht für ein mobiles Gerät entwickelt und führt somit zu Problemen und Umständlichkeiten bei der Verwendung auf einem Smartphone. Diese Probleme wurden durch die Portierung behoben.

Da ELBA mobil eine Browseranwendung ist, besitzt sie alle Sicherheitsrisiken, die für eine Homepage existieren. Im Gegensatz zu e-Banking am Computer ist die mTAN Methode nicht als sicher anzusehen, außer die TAN wird auf einem zweiten Gerät empfangen.

7.3 Mobile Banking der Bank Austria

Manche österreichischen Banken haben bei der Umsetzung von e-Banking auf eine eigene APP gesetzt und so die Anwendung browserunabhängig gemacht. Ein Beispiel ist Bank Austria Mobile Banking. Diese APP wurde von der UniCredit entwickelt und kann über den Android Market installiert werden. Bei der APP handelt es sich um eine Webview auf die e-Banking Webanwendung für den Computer. Diese wurde für das mobile Gerät angepasst. Die einzelnen Button rufen keine eigene Activity auf, sondern stellen einen neuen Webrequest an den Server. Die Antwort wird als Content der APP dargestellt.

Die aufgeführten Funktionen stammen aus der Beschreibung der APP.[19] Sie konnte aufgrund eines fehlenden Bankzuganges nicht getestet werden.

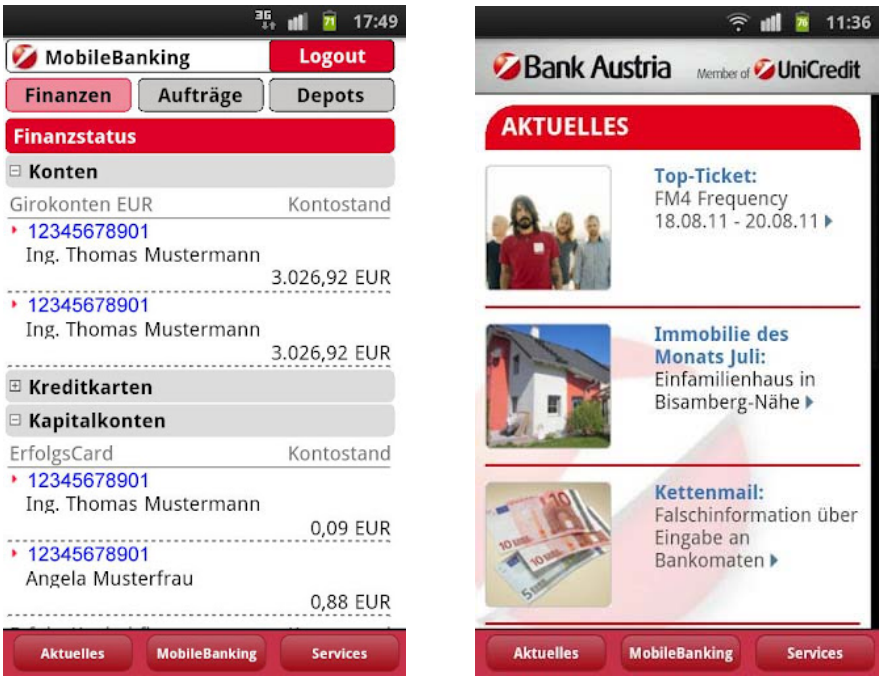


Abb. 21. Bank Austria Mobile Banking – Kontoübersicht, News [19]

7.3.1 Funktionen

Die Applikation Mobile Banking verfügt über einige Funktionen, die die Verwaltung des Kontos und der Depots ermöglichen. Die Navigation zu diesen Funktionen erfolgt über die obere und untere Navigationsleiste.

7.3.1.1 Aktuelles

Unter Aktuelles kann sich der Benutzer über aktuelle Änderungen in der Bank und am Finanzmarkt informieren. Hier werden auch Angebote und Events des Bankinstituts angeführt.

7.3.1.2 Kontofunktionen

Mobile Banking verfügt über die wichtigsten Funktionen zur Verwaltung der Konten. Es können die Kontostände abgefragt, sowie Überweisungen erfasst und abgesendet werden. Wie bei ELBA mobil ist es jedoch nicht möglich, Daueraufträge zu erstellen oder zu bearbeiten.

7.3.1.3 Wertpapierfunktionen

Mit den Wertpapierfunktionen können bestehende Wertpapiere verkauft und neue gekauft werden.

7.3.1.4 Kontaktfunktionen

Über die Kontaktfunktionen können Kontaktdaten wie e-Mail und wichtige Telefonadressen abgerufen werden. Mit diesen Nummern ist es möglich, schnell Kontakt zum Bankberater aufzunehmen,

um Terminvereinbarungen, Hilfestellungen und Kartensperrungen für verlorene Bankomatkarten zu erhalten.

7.3.2 Sicherheitsrisiken bei der Verwendung am Smartphone

Eine APP hat gegenüber einer Webanwendung den Vorteil, dass der Entwickler auf mehr Funktionen des Android Systems Zugriff hat. Bei Mobile Banking wird anstatt des Browsers eine eigene APP genutzt. Dadurch ist es möglich, eigene Buttons für Quicklinks zu verwenden. Die Sicherheitsprobleme, die bei einer Webanwendung entstehen können, bestehen jedoch weiter. An der Infrastruktur im Rechenzentrum sind gegenüber der reinen Webanwendung keine Änderungen notwendig.

Ein Großteil der Risiken entsteht durch die Unwissenheit des Benutzers. Es gibt keine Mechanismen, die gegen Schadsoftware, Shoulder Surfing oder Social Engineering wirken. Als Zugriffsschutz bei Diebstahl oder Verlust des Gerätes wirkt nur die PIN oder das Passwort und gegen das unautorisierte Absenden von Aufträgen die mTAN. Somit sollte bei Verlust des mobilen Gerätes sofort die PIN geändert werden.

Die Abhörsicherheit der Kommunikation wird mittels Serverzertifikat und TLS Verschlüsselung sichergestellt. Da jedoch der Aufbau des Rechenzentrums nicht bekannt ist, kann es hier zu weiteren Sicherheitsrisiken kommen.

7.3.3 Implementierte Sicherheitsmechanismen

Die Anwendung verwendet unterschiedliche Sicherheitsmechanismen. Da kein Bankkonto bei der Bank Austria zur Verfügung steht, kann die APP nicht getestet werden. Deshalb werden hier die Sicherheitsmerkmale der Installationsinformationen des Android Market angeführt.

7.3.3.1 PIN und TAN Kontrolle

Die Basissicherheit wird durch die PIN Eingabe bei der Anmeldung und TAN Eingabe beim Absenden von Aufträgen gewährleistet. Die TAN Ermittlung kann mittels mTAN durchgeführt werden. Durch die mTAN besteht jedoch das Problem, dass die 2 Wege Autorisierung nicht mehr gegeben ist. (vgl Seite 25, 5.3.3)

Diese Kontrollsysteme verhindern einen unberechtigten Zugriff und erhöhen somit die Zurechenbarkeit der Daten.

7.3.3.2 Serverzertifikate

Serverzertifikate bestätigen die Authentizität eines Servers und gewährleisten, dass die APP mit der richtigen Gegenstelle kommuniziert. Wird vom Server das richtige Zertifikat geliefert, kann der

Benutzer davon ausgehen, dass dieser Server derjenige ist, für den er sich ausgibt. Problematisch ist es, wenn das Zertifikat von einem Angreifer gestohlen und für den eigenen Server verwendet wird. (vgl Seite 76, 8.3.1)

Die Zertifikate erhöhen die Authentizität der Systemkomponenten und die Zuverlässigkeit des Systems.

7.3.3.3 Kommunikation mit SSL Verschlüsselung

Die Kommunikation wird mit SSL verschlüsselt. Damit kann eine sichere Übertragung gewährleistet werden. Wichtig ist dabei, dass der Schlüsselaustausch für die Verschlüsselung mit dem public Key des Serverzertifikates verschlüsselt wird. Dadurch kann es nur mit dem privaten Key des Servers gelesen werden. Der Schlüsselaustausch ist für einen Angreifer nicht vollständig lesbar. Damit wird die spätere Übertragung abhörsicher.

Die Vertraulichkeit der Daten kann gewährleistet werden. Werden die Daten bei der Übertragung geändert, können sie nicht mehr korrekt verschlüsselt werden. Somit kann auch die Integrität der Daten gesichert werden. (vgl Seite 79, 8.4.2)

7.3.3.4 Automatischer Logout nach 15 Minuten

Der Benutzer wird nach 15 Minuten automatisch aus dem System ausgeloggt. Damit soll einem unberechtigten Zugriff vorgebeugt werden. Diese Zeit ist im Vergleich zu den ELBA Produkten sehr lange und kann ein gewisses Sicherheitsrisiko darstellen.

Der automatische Logout ermöglicht, die Zurechenbarkeit der Daten bei gleichzeitigem Erhöhen der Benutzerfreundlichkeit zu gewährleisten, da der Benutzer nebenbei Aktionen durchführen kann, ohne sich für jede Bankaktion neu anmelden zu müssen.

7.3.3.5 Speicherung der Verfügerdaten

Standardmäßig speichert die APP die Verfügerdaten für den Login. Dies kann ein Sicherheitsrisiko darstellen, da der Angreifer nur mehr die PIN benötigt, um auf die Konten und Depots zugreifen zu können. Für die Benutzerfreundlichkeit ist es jedoch förderlich, da sich der Benutzer die Kontonummer, Bankleitzahl und Verfügernummer nicht merken muss. Über das Menü der APP kann im angemeldeten Zustand dieses Speichern der Logindaten deaktiviert werden. Ein unberechtigtes Anmelden wird erschwert, da der Angreifer die Daten nicht mehr aus dem Speicher auslesen kann.

7.3.3.6 Android-Version 2.2 notwendig

Die APP benötigt die Android Version 2.2, um sich installieren zu lassen. Damit kann sichergestellt werden, dass die Sicherheitslücken im System, die bis zu dieser Version geschlossen wurden, nicht

mehr ausgenützt werden können. Derzeit ist jedoch die Version 4 aktuell. Versionsmanagement und Updates stellen bei Android jedoch ein Problem dar (vgl Seite 23, 5.2.2). Damit die Anwendung von einem Großteil der Benutzer genutzt werden kann, ist es somit notwendig, eine ältere Version als Systemvoraussetzung zu wählen.

7.3.4 Angeforderte Berechtigungen

Da die APP nur die Berechtigung „Uneingeschränkter Internetzugriff“ benötigt, kann sie das mobile Gerät nicht auf eventuelle Schadsoftware überprüfen oder eine Ortung des Gerätes durchführen.

7.3.5 Handhabung und Benutzerfreundlichkeit

Mobile Banking wurde für Smartphones entwickelt und ermöglicht die Verwaltung der Konten vom mobilen Gerät aus. Um auf einem Handy gute Übersichtlichkeit zu gewährleisten, sind die Buttons am Rand eher klein gestaltet. Dies kann bei der Verwendung auf einem Tablet störend wirken.

7.3.6 Fazit zum Mobile Banking

Mobile Banking der Bank Austria bietet eine sichere Plattform, um Bankgeschäfte abzuwickeln. Sie bietet durch Buttons an der oberen und unteren Leiste eine gute Übersichtlichkeit. Es werden jedoch keine spezifischen Anwendungen für Smartphones wie Bankomatfinder oder Filialfinder angeboten.

7.4 E-Banking APP der easyBank und BAWAG P.S.K.

Sowohl für die easyBank als auch für die BAWAG P.S.K. existieren APPs für die mobile Verwaltung des Kontos. Die beiden Banken bieten zwar im Android Market jeweils eigene APPs an, jedoch unterscheiden sich diese nur im Design. Sowohl die Funktionalität als auch die Sicherheitsmechanismen sind dieselben.

Die aufgeführten Funktionen stammen aus der Beschreibung der APP.[20] Sie konnte aufgrund eines fehlenden Bankzuganges nicht getestet werden.

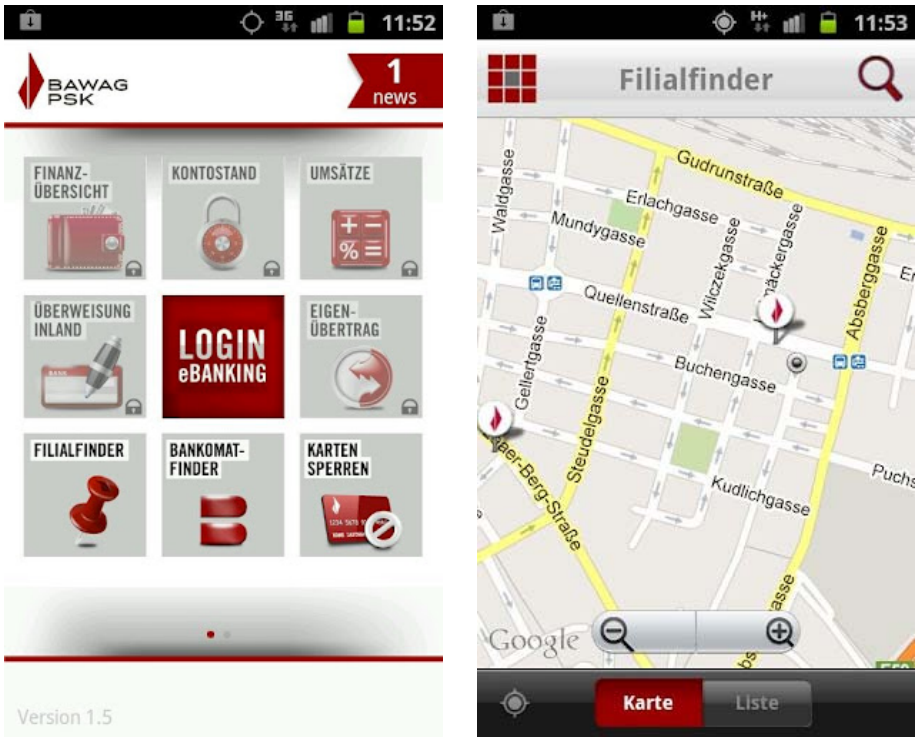


Abb. 22. e-Banking APP easyBank, BAWAG P.S.K. - Loginübersicht, Filialfinder [20]

7.4.1 Funktionen

Die APP bietet mehrere Funktionen, die teilweise auch für unangemeldete Benutzer verfügbar sind. Der Filialfinder, Bankomatfinder und die Funktion Kartensperren können auch ohne Login verwendet werden.

7.4.1.1 Sprachen

Die e-Banking APP ist in sieben Sprachen (Deutsch, Englisch, Kroatisch, Serbisch, Bosnisch, Polnisch, Türkisch) verfügbar, die abhängig vom Smartphone und den Ländereinstellungen ausgewählt werden können.

7.4.1.2 Filialfinder

Der Filialfinder verwendet das GPS Modul, um die aktuelle Position des Benutzers zu finden. Basierend auf diesen Informationen wird der kürzeste Weg zur nächsten BAWAG P.S.K. Filiale berechnet und angezeigt. (Abb. 22 e-Banking APP easyBank, BAWAG P.S.K. - Loginübersicht, Filialfinder)

7.4.1.3 Bankomatfinder

Der Bankomatfinder funktioniert ähnlich wie der Filialfinder. Er ermittelt über GPS die Position des Gerätes und zeigt die nächsten Bankomaten auf einer Karte an. (Abb. 22 e-Banking APP easyBank,

BAWAG P.S.K. - Loginübersicht, Filialfinder) Es werden jedoch nur Bankomaten der BAWAG P.S.K. angezeigt.

7.4.1.4 Kartensperre

Die Kartensperr-Funktion zeigt dem Benutzer die Telefonnummern an, die bei Verlust oder Diebstahl der Bankomatkarte oder Kreditkarte wichtig sind, um diese möglichst schnell sperren zu können.

7.4.1.5 News

Über die News Funktion werden dem Bankkunden wichtige Informationen und Angebote direkt am Smartphone angezeigt.

7.4.1.6 Kontofunktionen

Der Kontostand kann direkt und rund um die Uhr mit der Kontostandsfunktion angezeigt werden. Um alle Konten einzusehen, bietet die APP die Möglichkeit einer Finanzübersicht. Überweisungen können nur im Inland durchgeführt werden. SEPA-Überweisungen sowie Auslandsüberweisungen sind derzeit nicht möglich.

7.4.1.7 Eigenübertrag

Unter Eigenübertrag wird die Umbuchung von Geld zwischen eigenen Konten verstanden. Dies funktioniert nur, wenn beide Konten bei der easyBank bzw bei der BAWAG P.S.K sind. Ein Buchen zwischen unterschiedlichen Banken kann nur über eine Inlandsüberweisung durchgeführt werden.

7.4.1.8 Festlegung eines Überweisungslimits

Es kann ein Maximalbetrag für Überweisungen angegeben werden, bis zu dem eine Überweisung mit der APP angelegt und abgesendet werden kann. Bei einem Limit von 0 ist es nicht mehr möglich, Überweisungen vom Smartphone durchzuführen. Die APP kann dann nur mehr zum Kontrollieren des Kontostandes verwendet werden. Die Gefahr des Überweisens durch Unberechtigte ist damit abgedeckt. Es kann nur der Betrag und nicht die Anzahl der Überweisungen beschränkt werden.

7.4.1.9 Kontrollmappe

In der Kontrollmappe sind alle Transaktionen einzusehen und können hier überprüft werden. Das Absenden und Durchführen von Transaktionen ist nur einzeln möglich. Somit ist das gleichzeitige Absenden mehrerer Aufträge mit einer TAN nicht möglich. Es steht nur das mobileTAN-Verfahren (mTAN) zur Verfügung.

7.4.1.10 Umsätze

Die Umsätze zu den einzelnen Konten können in der Umsätze Funktion eingesehen werden. Durch die Umsatzsuche kann eine bestimmte Transaktion gesucht werden.

7.4.1.11 Mails

Weiters bietet die APP die Möglichkeit, auf das Mailkonto, das ein Bankkunde bei der Bank besitzt, zuzugreifen. Damit kann die gesamte Kommunikation mit dem Bankberater einfach und sicher durchgeführt werden. Da die Mails nur von dieser APP abgerufen werden, sind sie bei Verlust des Gerätes nur dann für den Angreifer lesbar, wenn dieser auch die Logininformationen für den Bankaccount kennt.

7.4.2 Sicherheitsrisiken bei der Verwendung am Smartphone

Die APP der easyBank und BAWAG P.S.K. verwendet zur Anmeldung ein persönliches Wischmuster. Damit können die Gefährdungen, die durch den Benutzer ausgehen, reduziert werden. Jedoch steigt parallel dazu die Gefahr, die durch Shoulder Surfing und der Analyse der Gebrauchsspuren verursacht wird.

Es sind keine Details zu den Sicherheitsmechanismen der Kommunikation vom mobilen Endgerät zum Rechenzentrum und zu den Mechanismen des Rechenzentrums bekannt. Deshalb könnten sich dort weitere Sicherheitsrisiken verbergen.

7.4.3 Implementierte Sicherheitsmechanismen

Die Anwendung verwendet unterschiedliche Sicherheitsmechanismen. Da die APP nicht zum Testen zur Verfügung steht, werden hier die Sicherheitsmerkmale der Installationsinformationen vom Android Market angeführt.

7.4.3.1 „höchste Sicherheitsstandards“

Die APP wird damit beworben, „höchste Sicherheitsstandards“ zu verwenden. Was darunter zu verstehen ist, konnte leider nicht ermittelt werden.

7.4.3.2 PIN und TAN Kontrolle

Durch die PIN Kontrolle bei der Anmeldung wird eine Basissicherheit sichergestellt. Sie ermöglicht die Zurechenbarkeit der Anmeldung. Ein unberechtigtes Auslesen ist nur bei Bekanntwerden der PIN möglich.

Die TAN Kontrolle schützt das Absenden von Überweisungsaufträgen und erhöht ebenfalls die Zurechenbarkeit. Das mobileTAN System ist jedoch für mobile Geräte unsicher, da die 2 Wege Autorisierung umgangen wird. (vgl Seite 25, 5.3.3)

7.4.3.3 Sicherheitsmuster

Beim ersten Login muss der Benutzer ein Muster festlegen, mit dem die Anwendung vor der Verwendung entsperrt werden kann. Damit wird ein zusätzliches Authentifikationsmerkmal festgelegt, wodurch vor allem die Zurechenbarkeit der Daten gewährleistet wird.

Dieses Muster unterliegt jedoch keiner Komplexitätsbeschränkung. Um sich das Muster leichter merken zu können, kann der Benutzer somit auch eine relativ einfache Kombination verwenden. Diese Kombination kann durch Brute Force oder durch Shoulder Surfing einfach durch den Angreifer ausgespäht werden. Da die APP ein eigenes Wischmuster verwendet, muss der Angreifer den Zugriff auf das Konto sehen. Das passiert selten im Vergleich zum Muster für das Entsperren des Gerätes.

7.4.3.4 Speicherung der Verfügerdaten

Standardmäßig werden die Verfügerdaten (Bankleitzahl, Kontonummer, Verfügernummer) im System gespeichert, sodass der Benutzer sie nicht bei jedem Login erneut angeben muss. Dies kann durch eine Einstellung im Menü verändert werden. Damit wird zwar die Sicherheit erhöht, da ein Angreifer dann auch diese Daten kennen muss, jedoch wird die Benutzerfreundlichkeit gesenkt, da der Benutzer die Daten bei jedem Login über die Bildschirmtastatur eingeben muss.

7.4.4 Angeforderte Berechtigungen

Die Applikation benötigt folgende Berechtigungen, um installiert werden zu können:

- Uneingeschränkter Internetzugriff
- Genauer (GPS-) Standort
- Systemtools – Laufende Anwendungen abrufen

Die angeforderten Berechtigungen machen in Bezug auf die Funktionen Sinn. Der Internetzugriff ist für die Kernfunktion e-Banking notwenig. GPS wird vor allem für die Funktionen Bankomatfinder und Bankstellenfinder wichtig. Diese Berechtigung „Systemtools – Laufende Anwendungen abrufen“ erlaubt es, alle derzeit auf dem Gerät ausgeführten Threads anhand ihres Paketnamens zu identifizieren. Daraus kann geschlossen werden, dass die APP eventuell die anderen parallel laufenden Prozesse überprüft und damit auf Malware zu schließen versucht.

7.4.5 Handhabung und Benutzerfreundlichkeit

Die APP wurde speziell für Android entwickelt und verwendet Möglichkeiten, die dieses Betriebssystem bietet, um einerseits die Sicherheit, aber andererseits auch die Benutzerfreundlichkeit zu

erhöhen. Die APP ist übersichtlich, jedoch kann sie nur im Portrait Modus (hochkant) verwendet werden. Es werden auch laufend Patches und Neuerungen über den Android Market veröffentlicht.

7.4.6 Fazit zum e-Banking der easyBank, BAWAG PSK

Die Applikation bietet viele Sicherheitsmerkmale, jedoch wird nur die mobileTAN als TAN Verfahren angeboten. Dieses Verfahren ist für das Smartphone unsicher. Das persönliche Wischmuster hat den Vorteil, dass der Zugang durch ein weiteres Merkmal geschützt wird und somit die Zurechenbarkeit erhöht werden kann. Jedoch ist zu bedenken, dass das Muster leicht durch Shoulder Surfing oder durch die Analyse von Gebrauchsspuren ausgelesen werden kann.

Vom Funktionsumfang her bietet es zwar Komfortfunktionen wie den Bankomatfinder, jedoch können nur Inlandsüberweisungen durchgeführt werden und der Wertpapierhandel ist nicht implementiert.

Die Benutzeroberfläche ist übersichtlich. Alle Funktionen sind durch wenige Klicks erreichbar.

7.5 E-Banking APP der Erste Bank und Sparkassen

Die Erste Bank und Sparkassen habe in Österreich noch keine APP für e-Banking im Android Market veröffentlicht. Die derzeit veröffentlichte Applikation bietet grundlegende allgemeine Bankfunktionen.

Abb. 23. Banking APP der Erste Group [21]

7.5.1 Funktionen

Die Funktionen bieten Informationen rund um das Bankgeschäft. Es ist jedoch nicht möglich, den Kontostand auszulesen oder Überweisungen zu tätigen.

7.5.1.1 Filialen- und Bankomatensuche

Es wird die Möglichkeit einer Filialen- und Bankomatensuche angeboten. Dabei wird der aktuelle Standort mittels GPS erfasst und die Filialen der Erste Bank und Sparkassen angezeigt. In Österreich werden auch Bankomaten von anderen Geldinstituten markiert. Zu den Filialen werden Zusatzinformationen wie genaue Adresse, Öffnungszeiten und Kontaktmöglichkeiten angeführt.

7.5.1.2 Kreditrechner

Mit dem Kreditrechner kann der Benutzer die Laufzeit eines Kredites basierend auf der monatlichen Rate und dem möglichen Kreditbeitrag berechnen.

7.5.1.3 Aktienempfehlungen

Unter dem Punkt Aktienempfehlungen werden jeweils Freitags die Empfehlungen der Analysten der Erste Group Bank AG angezeigt.

7.5.1.4 Aktuelle Wechselkurse

Mit der APP können auch die aktuellen Wechselkurse der wichtigsten Währungen sowie Gold und Silber ausgelesen werden.

7.5.1.5 Prognosen

Unter Prognosen werden die Prognosen des Research Instituts der Erste Group Bank AG für verschiedene Währungen sowie für „10-jährige Rendite“ und 3 Monats Zinssätze angeboten.

7.5.1.6 Research

Research beinhaltet wichtige Informationen zu Aktien, Anleihen, Rohstoffen und Währungen, sowie aktuelle Research Videos.

7.5.1.7 News

Unter News können die aktuellen News der Erste Bank Group AG nachgelesen werden.

7.5.2 Angeforderte Berechtigungen

Die Applikation benötigt folgende Berechtigungen, um installiert werden zu können:

- Uneingeschränkter Internetzugriff
- Genauer (GPS-) Standort

- Telefonstatus lesen und identifizieren
- Telefonnummern direkt anrufen

Die Anwendung muss, um die aktuellen Daten abzurufen, eine Verbindung mit dem Server im Rechenzentrum aufbauen. Dafür wird der Internetzugriff benötigt. Für den Bankomat- und Filialfinder muss der Standort mittels GPS ausgelesen werden können. Die Berechtigungen „Telefonstatus lesen und identifizieren“ sowie „Telefonnummern direkt anrufen“ werden benötigt, damit der Benutzer sich die Telefonnummer eines bestimmten Kontakts nicht merken muss, sondern ihn direkt über die Phone-APP des Systems aufrufen kann.

7.5.3 Sicherheitsrisiken bei Verwendung am Smartphone

Da es sich bei der Applikation um eine reine Informationsanwendung handelt, existieren nur geringe Risiken. Diese betreffen vor allem die Eingaben des Benutzers, da dessen Standort für die Filialensuchfunktion benötigt wird. Hinweise zum Finanzstatus könnten die Daten aus dem Kreditrechner geben, jedoch entstehen dadurch keine bindenden Geschäfte, wodurch keine Sicherungswürdigkeit der Daten entsteht.

Für das Rechenzentrum bestehen jedoch Gefahren, da die Anwendung (für die Aktienkurs und News) offene Ports und Schnittstellen benötigt, um auf die Server zugreifen zu können. Diese Ports können auch ein Risiko für andere Services darstellen, sofern sie nicht ausreichend geschützt sind.

7.5.4 Implementierte Sicherheitsmechanismen

Es liegen keine Informationen über implementierte Sicherheitsmaßnahmen vor. Da jedoch auch keine wichtigen und geheimen Informationen verarbeitet und transportiert werden, sind Sicherheitsmaßnahmen nicht unbedingt notwendig. Es muss jedoch auf die Sicherheit der Server geachtet werden. Für diese reichen die normalen Sicherheitseinstellungen für Homepages aus.

7.5.5 Handhabung und Benutzerfreundlichkeit

Die APP bietet eine gute Informationsquelle für jegliche bankbezogene Informationen. Diese Daten werden übersichtlich und einfach erreichbar dargestellt. Jedoch fordert die APP mehr Berechtigungen als wahrscheinlich notwendig. Weiters sind die fehlenden Bankfunktionen ein großer Nachteil.

7.5.6 Fazit

Die APP bietet dem Benutzer viele Informationen, jedoch können diese auch von anderen Homepages gelesen werden. Der Funktionsumfang umfasst die Funktionen, die bei andern Banken als

Zusatz zum e-Banking geliefert werden. Sicherheit ist aufgrund der fehlenden vertraulichen und geheimen Daten keine notwendig und auch nicht implementiert. [21]

7.6 Vergleich

Derzeit gibt es viele unterschiedliche Anwendungen im Bereich e-Banking für das Android Betriebssystem. Diese Anwendungen unterscheiden sich einerseits in der Art und Weise (Webanwendung oder APP) und andererseits durch die Funktionen und verwendeten Sicherheitsmechanismen. Die Anwendungen ELBA mobil, Mobile Banking und die APP der easyBank; BAWAG P.S.K. zeigen sehr gut die Möglichkeiten, wie e-Banking durchgeführt werden kann. Während ELBA eine reine Website ist, handelt es sich beim Portal der easyBank um eine vollwertige APP. Mobile Banking der Bank Austria ist eine Mischung, die in der Mitte zwischen den beiden anderen angesiedelt ist – eine Webpage in eine APP integriert.

7.6.1 Funktionsvergleich

Die einzelnen Anwendungen unterscheiden sich sehr maßgeblich in den angebotenen Funktionalitäten. ELBA Internet ist für den Computer erstellt und kann mit Hilfe des Browsers am Smartphone aufgerufen werden. Somit wird derselbe Funktionsumfang wie am Computer geboten. Ähnlich dazu ist die Anwendung mobile Banking der Bank Austria. Diese unterstützt jedoch nicht alle Funktionalitäten der Computerversion. Mehr Funktionalitäten sind bei ELBA mobil und der e-Banking APP der easyBank; BAWAG P.S.K. verfügbar. Hier werden zusätzlich Bankomat- und Filialfinder angeboten. Die APP der Erste Group bietet im Gegensatz zu den vorher genannten nur Informationsfunktionen, wie den Bankomat- oder Filialfinder. (Abb. 42 Vergleich e-Banking Systeme Österreich - Funktionalität)

7.6.2 Sicherheitsmechanismen

Die Sicherheitsmechanismen der Anwendungen sind großteils dieselben wie für Anwendungen auf Computersystemen. Die zusätzlichen Möglichkeiten, die Android bietet, werden nur von der easyBank und der BAWAG P.S.K genutzt. Die verwendeten Sicherheitsmechanismen sind das Wischmuster und der Zugriff auf die Liste der laufenden Applikationen. (Abb. 43 Vergleich e-Banking Systeme Österreich – Sicherheitsmechanismen)

7.6.3 Zutreffende Risiken

Die Anwendungen unterscheiden sich bei den Risiken, die durch die Verwendung auftreten, vor allem durch ihren Typ. Die browserbasierten Lösungen haben weniger Zugriffsmöglichkeiten auf

das System. Einerseits sind bestimmte Funktionen aufgrund der Einschränkungen des Browsers nicht möglich, andererseits bietet die Ausführung im Browser weitere Sicherheiten.

Da die Applikationen die Kommunikation mit TLS bzw SSL im Falle von HTTPS verschlüsseln, ist ein Abhören bei gültigem und korrektem Serverzertifikat nicht bzw nur sehr schwer möglich. Die Problematik durch die 2 Wege Autorisierung wurde nur von den ELBA Systemen mit Hilfe der cardTAN gelöst.

Störsender und DoS können von keiner Anwendung abgewendet werden. Für diese Angriffe gibt es jedoch auch keine passenden Abwehrmechanismen, da sie die Konnektivität des Gerätes zum Netzwerk durch Kollisionen und Störungen unterbinden. (Abb. 45 Vergleich e-Banking Systeme Österreich – Risiken und Maßnahmen)

7.6.4 Benötigte Berechtigungen

Die Anwendungen benötigen unterschiedliche Berechtigungen. Die ELBA Produkte fordern keine eigenen Berechtigungen, da sie im Browser ausgeführt werden und somit die Berechtigungen des Browsers mitnutzen.

Der Internetzugriff wird von allen APPs eingefordert und ist auch Voraussetzung für die Hauptfunktion. Zusätzlich dazu benötigt die e-Banking Anwendung der easyBank, BAWAG P.S.K. und die Bank APP der Erste Group Zugriff auf das GPS Modul für den Filial- und Bankomatfinder.

Die easyBank, BAWAG P.S.K. möchte weiters die parallel laufenden Anwendungen auslesen können. Dies könnte zur Erkennung von Schadsoftware verwendet werden.

Dass die Bank APP der Erste Group Telefonnummern anrufen und den Telefonstatus lesen und identifizieren will, erhöht zwar die Benutzerfreundlichkeit, wird jedoch bei manchen Benutzer auf Skepsis stoßen. (Abb. 44 Vergleich e-Banking Systeme Österreich – benötigte Berechtigungen)

8. Sicherheitsmechanismen für e-Banking

Die Programme für Android werden in Java geschrieben, jedoch besitzt ein Smartphone Spezialhardware, die auf einem Computer nicht verfügbar ist. Aus diesem Grund gibt es eigene Klassen und Bibliotheken, um diese Möglichkeiten einbinden zu können. Durch die zusätzlichen Möglichkeiten der mobilen Geräte entstehen auch weitere Mechanismen, die die Sicherheit einer Anwendung erhöhen können.

Im Gegensatz zu den zusätzlichen Funktionen muss bei der Programmierung für Android jedoch auch mitbedacht werden, dass das Gerät einen kleineren Monitor besitzt und es eine geringere Rechenleistung hat als ein Computer.

Die Authentifikation eines Gerätes ist ein wichtiges Thema. Vor allem bei sicherheitskritischen Anwendungen wie e-Banking ist es wichtig, dass sich die Kommunikationspartner gegenseitig authentifizieren. Damit kann gewährleistet werden, dass jeder derjenige ist, für den er sich ausgibt.

8.1 Authentifikation des Benutzers

Dieses Kapitel beschäftigt sich mit den Möglichkeiten der Authentifikation des Benutzers gegenüber dem Gerät oder dem Rechenzentrum.

8.1.1 Gesichtserkennung

Eine Möglichkeit der Identifikation ohne Wissen oder Besitz basiert auf biometrischen Merkmalen. Die meisten Smartphones verfügen über mindestens eine Kamera. Mit dieser kann ein Bild vom Benutzer aufgenommen und dieser anhand der Gesichtsmerkmalen identifiziert werden. Dadurch identifiziert sich der Benutzer gegenüber dem mobilen Gerät.

8.1.1.1 Abgedeckte Gefahren

Die Identifikation basiert auf einem biometrischen Merkmal des Benutzers. Es ist genetisch veranlagt und kann somit nicht verändert werden. Eine Fälschung ist jedoch durch Bilder oder Masken möglich. Durch diese Methode werden die Gefahren Social Engineering (vgl Seite 30, 6.1) und Shoulder Surfing (vgl Seite 33, 6.5) sowie die des physikalischen Zugriffes (vgl Seite 33, 6.4) abgedeckt bzw abgeschwächt.

Die Authentifikation mittels eines biometrischen Merkmals ermöglicht, dass kein Wissen oder Besitz bei der Anmeldung verfügbar sein muss. Es kann der Zugang auf genau eine Person beschränkt werden. Auch der physikalische Zugriff auf das Gerät ermöglicht keinen Zugang.

8.1.1.2 Zutreffende Gefahren

Durch die Verwendung eines biometrischen Identifikationssystems entstehen für das System neue Gefahren, die Angriffe auf dieses ermöglichen. Durch Schadsoftware (vgl Seite 31, 6.2) sowie durch Methoden wie das Austricksen mit Fotos oder Masken, das Einspielen eines gespeicherten Bildes in den Contentmanager des Kamera-APPs oder die Veränderung der gespeicherten Referenzbilder kann das Sicherheitssystem umgangen oder beeinträchtigt werden.

Die Bibliotheken für die Gesichtserkennung sind erst ab Android 4.0 verfügbar. Diese Gesichtserkennung ist jedoch unsicher und kann mit einem Foto leicht ausgetrickst werden. [23]

8.1.1.3 Beteiligte Instanzen

Der gesamte Identifizierungsvorgang wird am Gerät abgewickelt und erfordert keine Eingabe von Passwörtern oder Mustern durch den Benutzer. Der Benutzer nimmt mit Hilfe der Kamera ein Foto seines Gesichtes auf. (Merkmalsaufnahme) Das Gerät vergleicht dann das Bild mit den gespeicherten Referenzbildern und gewährt oder verweigert basierend auf der Ähnlichkeit den Zugriff (Merkmalsauswertung).

8.1.2 Mustereingabe (Wischmuster)

Die Eingabe von Zeichen ist auf mobilen Geräten verhältnismäßig schwierig durchzuführen. Sie ist nur mit der Hilfe der Bildschirmtastatur möglich. Die Eingabegeschwindigkeit ist durch die Hardware begrenzt. Aufgrund des geringen Platzbedarfs kann nicht die gesamte Tastatur auf einmal abgebildet werden. Für Zahlen und Sonderzeichen muss vorher die Anzeige geändert werden, was die Geschwindigkeit weiter reduziert. Außerdem ist die Wahrscheinlichkeit von Eingabefehlern erhöht. Eine Alternative zur klassischen Passworteingabe bietet das Wischmuster unter Android. Hierbei werden mindestens 4 von 9 Punkten in einer bestimmten Reihenfolge miteinander verbunden.

8.1.2.1 Abgedeckte Gefahren

Die Eingabe eines Musters basiert auf dem Wissen des Benutzers. Es ist ein nicht zeichenbasiertes Passwort. Damit können die Gefährdungen, die vom physikalischen Zugriff (vgl Seite 33, 6.4) und von Shoulder Surfing (vgl Seite 33, 6.5) ausgehen teilweise abgedeckt werden.

Die Passworteingabe in Form eines Musters ermöglicht eine erhöhte Benutzerfreundlichkeit gegenüber der umständlichen Eingabe über die Bildschirmtastatur. Bei einem entsprechend komplexen Muster kann auch die Gefahr von Shoulder Surfing verringert werden.

8.1.2.2 Zutreffende Gefahren

Durch die Eingabe des Musters teilt der Benutzer dem Gerät mit, dass er das Muster kennt. Ob es sich um den richtigen Benutzer handelt, kann vom Gerät jedoch nicht bestätigt werden. Die Sicherheit dieses Mechanismus wird hauptsächlich von Social Engineering (vgl Seite 30, 6.1) beeinträchtigt.

Der physikalische Zugriff (vgl Seite 33, 6.4) erhöht die Gefahr, durch Gebrauchsspuren das Muster ablesen zu können. Je einfacher das Muster ist, umso leichter kann es aus den Gebrauchsspuren oder durch Shoulder Surfing (vgl Seite 33, 6.5) ermittelt werden. (vgl Seite 71, Abb. 24 Beispiele für unsichere Muster)

8.1.2.3 Beteiligte Instanzen

Der Benutzer identifiziert sich beim Endgerät. Das mobile Gerät fungiert als Eingabeschnittstelle und benutzt die Authentifizierungsinformation, um die Anwendung mit dem Benutzer auszuführen, dem das Muster zugeordnet ist.

8.1.2.4 Sicherheitsprobleme

Das Muster ist jedoch anfällig für Shoulder Surfing (vgl Seite 33, 6.5) und die Analyse der Gebrauchsspuren. (vgl Seite 25, 5.3.2)

Bei der Konfiguration stehen mehrere Einstellungsmöglichkeiten zur Verfügung. Diese Einstellungen reduzieren jedoch teilweise die Sicherheit des Musters. Durch das Anzeigen des Musters während der Eingabe kann dieses sehr leicht durch Shoulder Surfing ausgelesen werden. Weiters ist es möglich, sehr einfache und somit unsichere Muster zu verwenden.

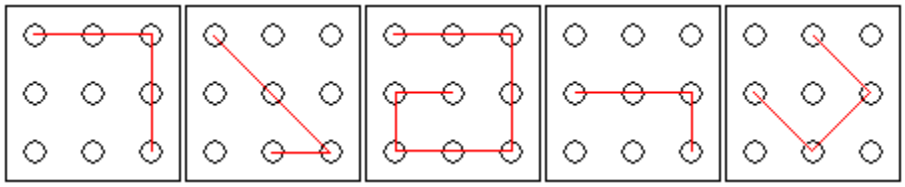


Abb. 24. Beispiele für unsichere Muster

8.1.2.5 Testprogramm

Für die Mustererkennung sind keine eigenen Bibliotheken im Android SDK verfügbar. Die Erkennung des Musters muss manuell ausprogrammiert werden. Somit ist auch die Sicherheit dieser Methode sehr stark von der Implementierung abhängig. Es besteht jedoch die Möglichkeit, das Loginpattern aus den Android Sourcen zu exportieren. Die Sourcen können von https://code.google.com/p/android-lockpattern/ heruntergeladen werden. [28]

8.1.3 Mustereingabe (Touchmuster)

Die Mustereingabe funktioniert ähnlich wie beim Wischmuster, nur dass die Punkte nicht durch Wischen verbunden werden, sondern eine bestimmte Anzahl an Buttons gedrückt werden muss. Aus der Reihenfolge dieser wird ein Code erstellt, der in Form eines Hashwertes mit dem des richtigen Codes verglichen wird.

8.1.3.1 Abgedeckte Gefahren

Die Eingabe eines Musters basiert auf dem Wissen des Benutzers. Damit können die Gefährdungen des physikalischen Zugriffs (vgl Seite 33, 6.4) und von Shoulder Surfing (vgl Seite 33, 6.5) abgedeckt werden.

Die Passworteingabe in Form eines Musters ermöglicht eine erhöhte Benutzerfreundlichkeit, da die Eingabe über die Bildschirmtastatur sehr aufwändig ist. Bei einem entsprechend komplexen Muster kann auch die Gefahr von Shoulder Surfing gebannt werden. Im Gegensatz zum Wischmuster entstehen keine Gebrauchsspuren, da nicht über den Monitor gewischt wird und somit keine Kratzer entstehen können.

8.1.3.2 Zutreffende Gefahren

Durch die Eingabe des Musters teilt der Benutzer dem Gerät mit, dass er das Muster kennt. Ob es sich um den richtigen Benutzer handelt, kann vom Gerät durch die Eingabe nicht bestätigt werden. Die größte Gefahr für diesen Sicherheitsmechanismus geht von Social Engineering aus. (vgl Seite 30, 6.1)

Je einfacher das Muster ist, umso leichter kann es durch Shoulder Surfing (vgl Seite 33, 6.5) ermittelt werden. (vgl Seite 71, Abb. 24 Beispiele für unsichere Muster)

8.1.3.3 Beteiligte Instanzen

Der Benutzer identifiziert sich entweder beim Endgerät oder direkt beim Server. Das mobile Gerät benutzt diese Authentifizierungsinformation, um die Anwendung mit dem Benutzer auszuführen, dem das Muster zugeordnet ist. Alternativ kann auch die Eingabe an den Server über den Kommunikationskanal übertragen und dort überprüft werden.

8.1.4 Passworteingabe, PIN

Durch die Passworteingabe bestätigt der Benutzer, dass er ein bestimmtes Wissen hat. Dieses Wissen kann in unterschiedlicher Form eingegeben werden. Die übliche Variante ist ein textuelles Passwort, das über ein Textfeld eingegeben wird. Möglich wäre aber auch eine numerische PIN, die mit Hilfe eines Touchmusters (vgl Seite 71, 8.1.3) erfasst wird.

8.1.4.1 Abgedeckte Gefahren

Da das Passwort nicht wie ein Wischmuster (vgl Seite 70, 8.1.2) gewischt sondern getippt wird, entstehen keine Gebrauchsspuren. Die Gefährdungen durch reverse Engineering (vgl Seite 32, 6.3) und dem physikalischen Zugriff (vgl Seite 33, 6.4) können vermindert werden.

Abhängig von der Art der Eingabe kann auch die Gefahr von Shoulder Surfing (vgl Seite 33, 6.5) reduziert werden.

8.1.4.2 Zutreffende Gefahren

Da es sich bei einem Passwort um Wissen handelt, kann es weitergegeben werden. Bei einfachen Passwörtern oder einer schlechten Eingabeweise kann dieses leichter von einem Angreifer

mitgelesen werden. Somit sind die Gefahren durch Social Engineering (vgl Seite 30, 6.1) und Shoulder Surfing (vgl Seite 33, 6.5) zu beachten.

Wird die Eingabe vom Server überprüft, muss beachtet werden, dass diese nicht während der Übertragung durch einen Lauschangriff (vgl Seite 35, 6.7) oder einen Man in the Middle Angriff (vgl Seite 38, 6.11) ausgelesen oder verändert wird.

8.1.4.3 Beteiligte Instanzen

Der Benutzer gibt seine Kennung mit Hilfe des Textfeldes oder des Touchmusters ein. Danach wird es vom Gerät zum Server übertragen und dort auf die Gültigkeit überprüft. Aus Sicherheitsgründen sollte nur ein Hash der Eingabe zum Server übertragen werden.

8.2 Authentifikation des Gerätes

In diesem Kapitel werden Methoden erklärt, wie sich das Gerät gegenüber dem Rechenzentrum authentifizieren kann.

8.2.1 Identifikation mittels IMSI oder IMEI

Die Identifikation mittels Wissen oder Besitz (Benutzername, Passwort) ermöglicht einfachen Zugriff von verschiedenen Zugangspunkten aus. Es entstehen damit jedoch auch Probleme, da Wissen leicht weitergegeben oder vergessen werden kann. Eine sicherere Identifikation erfolgt durch eindeutige Merkmale des Gerätes oder des Benutzers. Bei einem mobilen Gerät kann die IMSI oder IMEI verwendet werden.

Bei einer derartigen Identifikation ist der Benutzerzugang hardwareabhängig. Somit ist es bei einem Wechsel der Hardware notwendig, auch den Zugang neu zu konfigurieren.

Die Identifikation mit IMSI und IMEI ermöglicht eine Identifizierung des Gerätes beim Rechenzentrum. Eine Identifizierung des Benutzers ist nicht möglich, da das Gerät auch von anderen Benutzern verwendet werden kann.

8.2.1.1 Abgedeckte Gefahren

Durch die Verwendung einer Identifizierung durch die IMSI oder IMEI können Gefahren durch Social Engineering (vgl Seite 30, 6.1) und Shoulder Surfing (vgl Seite 6.5) abgedeckt oder zumindest verringert werden.

Durch das Verwenden der IMSI oder IMEI bei der Authentifizierung des Benutzers kann der Zugang auf ein bestimmtes Gerät eingeschränkt werden. Die Zugangsdaten des Benutzers alleine ermöglichen einem Angreifer keinen Zugriff auf dessen Finanzen.

8.2.1.2 Zutreffende Gefahren

Da die Identifizierung auf Merkmalen des Gerätes basiert, besitzen die Gefährdungen Reverse Engineering (vgl Seite 32, 6.3) und der physikalische Zugriff (vgl Seite 33, 6.4) sowie eine Manipulation der Hardware (vgl Seite 34, 6.6) ein erhöhtes Risikopotential. Es kann vom Rechenzentrum nicht direkt festgestellt werden, ob die APP die echten Daten des Gerätes gesendet hat. Um diese gesichert zu erfahren, müsste eine zusätzliche Kommunikation mit dem Provider aufgebaut werden.

Der Zugang ist auf bestimmte Geräte reduziert. Nur durch den direkten Zugriff auf das mobile Gerät kann der Zugang genutzt werden. Durch virtuelle Geräte oder die Veränderung der Hardware kann ein Gerät nachgebildet werden und somit der Zugriff auch von einem anderen Gerät durchgeführt werden. Da die IMSI und IMEI vom Rechenzentrum überprüft werden, ist es notwendig, dass diese bei der Übertragung nicht verändert oder abgehört werden. (vgl Seite 35, 6.7; Seite 38, 6.11)

8.2.1.3 Beteiligte Instanzen

Bei der Identifizierung durch IMSI und IMEI werden die Identifikationsmerkmale vom Gerät ausgelesen. Diese werden einerseits vom Hersteller des mobilen Gerätes (IMEI) und dem Telekommunikationsunternehmen durch die SIM Karte (IMSI) festgelegt. Die Daten werden von der APP über den Kommunikationskanal ins Rechenzentrum transportiert und dort mit den Reverenzdaten abgeglichen.

8.2.1.4 IMSI Identifikation

Die IMSI ist die eindeutige Nummer der SIM Karte und ermöglicht eine weltweite Identifikation dieser. Das Ändern der Nummer ist nur durch einen Wechsel der SIM Karte möglich.

8.2.1.5 IMEI Identifikation

Die IMEI ist bei jedem Handy fix in die Hardware codiert. Eine Änderung ist nur durch den Austausch eines Chips möglich.

8.2.1.6 Testprogramm

Die IMEI und IMSI können direkt aus dem TelephonyManager ausgelesen werden. Dazu wird jedoch die Berechtigung „`android.permission.READ_PHONE_STATE`“ benötigt.

Durch einen Stringvergleich können die ausgelesenen Werte mit Referenzdaten verglichen werden.

```
TelephonyManager telephonyManager = (TelephonyManager)
     getSystemService(Context.TELEPHONY_SERVICE);
IMEI = telephonyManager.getDeviceId();
IMSI = telephonyManager.getSubscriberId();
```

Code. 1. IMSI und IMEI auslesen

8.2.2 Google Account

Jedes Android Gerät muss mit einem Google Account aktiviert werden. Dieser Accountname kann ausgelesen werden, um den Benutzer zu identifizieren. Der Vorteil des Accounts ist, dass er für mehrere Geräte verwendet werden kann und somit der Zugang für die Anwendung nicht geändert werden muss, wenn die Hardware getauscht wird. Jedoch ist zu bedenken, dass bei einem Verlust der Google Zugangsdaten auch der Identitätsnachweis für die Anwendungen, die diese Daten für die Authentifizierung verwenden, verloren geht.

8.2.2.1 Abgedeckte Gefahren

Bei der Verwendung des Google Accounts zur Identifizierung des Benutzers wird davon ausgegangen, dass die Zugangsdaten zu diesem Account geheim und nur dem Benutzer bekannt sind. Es kann einerseits ein Googlekonto auf mehreren mobilen Geräten und PCs verbunden und andererseits mehrere Googlekonten auf einem Gerät verwendet werden. Die Gefahr durch Social Engineering (vgl Seite 30, 6.1) und Shoulder Surfing (vgl Seite 33, 6.5) werden aus Sicht der e-Banking APP abgedeckt, jedoch werden sie bei dieser Methode auf die Authentifizierung mit Google verschoben. Das Google Konto ist nicht für eine hohe Sicherheit, wie sie für e-Banking benötigt wird, konzipiert. Ein Vorteil ist jedoch, dass die Zugangsdaten nicht bei jedem Anmeldevorgang eingegeben werden müssen. Damit kann Shoulder Surfing in vielen Fällen vollständig verhindert werden.

8.2.2.2 Zutreffende Gefahren

Die Verwendung des Google Mail Accounts ermöglicht die Verlagerung der Gefährdungen zu Google. Die Gefahr Social Engineering (vgl Seite 30, 6.1) ist davon betroffen. Durch Reverse Engineering (vgl Seite 32, 6.3) und durch einen physikalischen Zugriff (vgl Seite 33, 6.4) können die Accountdaten ausgelesen werden, wodurch eine weitere Bedrohung entsteht. Für einen Angreifer ist es weiters möglich, durch einen Lauschangriff (vgl Seite 35, 6.7) und eine Man in the Middle Attacke (vgl Seite 38, 6.11) zu den Daten zu kommen. Ein weiteres Problem besteht darin, dass das Google Konto für viele weitere Services und auch am Computer verwendet wird (zB YouTube). Dadurch ist es Malware ausgesetzt.

Die Authentifikation sollte nicht nur auf dem Google Account basieren, da dieser nicht genügend Sicherheit bezüglich der Authentizität des Benutzers bietet. Die Accountdaten müssen geschützt vom Endgerät zum Rechenzentrum übertragen werden.

8.2.2.3 Beteiligte Instanzen

Bei der Identifizierung wird der Benutzer durch das Gerät und dem darauf verbundenen Google Account beim Rechenzentrum identifiziert. Als Authentifikationsbasis dient die Anmeldung am Google Server, die zuvor vom Anwender manuell durchgeführt werden muss. Da bei diesem Vorgang die Daten mit Google gegengeprüft werden müssen, erfährt auch diese Firma von der Anmeldung zu e-Banking.

8.3 Authentifikation des Rechenzentrum

Dieses Kapitel zeigt eine Möglichkeit auf, wie sich das Rechenzentrum gegenüber dem Gerät und dem Benutzer authentifizieren kann.

8.3.1 Serverzertifikat

Ein Serverzertifikat dient der Authentifizierung des Servers gegenüber dem mobilen Endgerät bzw dem Benutzer. Nur wenn das Zertifikat aus einer vertrauenswürdigen Quelle stammt, kann diesem vollständig vertraut werden.

Um die Identität des Servers zu bestätigen, verfügt dieser über ein von einer übergeordneten CA ausgestelltes Zertifikat. Damit kann festgestellt werden, ob der Server derjenige ist, für den er sich ausgibt. Befindet sich die CA, die das Zertifikat für den Server ausgestellt hat, in der Liste der vertrauenswürdigen CAs, kann die Identität sofort festgestellt werden. Trifft das nicht zu, wird der Benutzer darauf aufmerksam gemacht, dass der Server nicht identifiziert werden konnte. Dies kann mehrere Gründe habe: Es kann sein, dass die CA noch nicht in die Liste der Vertrauenswürdigen aufgenommen wurde, dass zu einem falschen Server verbunden wurde und deshalb das Zertifikat ungültig ist, oder dass das Zertifikat abgelaufen ist oder zurückgerufen wurde.

Um die Unwissenheit der Benutzer zu umgehen und zu verhindern, dass ein Benutzer ein falsches Zertifikat bestätigt, sollte die Überprüfung durch die APP durchgeführt werden. Dabei muss jedoch bedacht werden, dass bei einem Wechsel des Zertifikates die Anwendung neu ausgerollt werden muss.

8.3.1.1 Abgedeckte Gefahren

Das Serverzertifikat wird für mehrere Sicherheitsmechanismen benötigt. Einerseits wird es zur Authentifizierung des Servers durch den Client verwendet und andererseits kann damit eine sichere Kommunikation mit SSL/TLS durchgeführt werden. (vgl Seite 80, 8.4.3) Da der Server mit dem Zertifikat genau identifiziert werden kann, kann die Gefahr durch eine DNS Manipulation (vgl Seite 41, 6.11.6) abgewehrt werden.

8.3.1.2 Zutreffende Gefahren

Das Zertifikat muss sicher verwahrt werden, da ein Verlust Auswirkungen auf die Authentizität des Servers und auf die Sicherheitsmechanismen, die auf das Zertifikat aufbauen, hat. Somit muss sowohl der physische (vgl Seite 43, 6.13) als auch der virtuelle Zugriff (vgl Seite 43, 6.14) auf das Rechenzentrum verhindert werden.

Allgemein stellen alle Angriffe auf das Rechenzentrum und den Server eine Gefahr für das Zertifikat dar.

Vor allem in den letzten Jahren wurde immer öfter bekannt, dass die Sicherheit der CAs nicht immer gewährleistet ist und es zu gefälschten Zertifikaten kommt. Diese ermöglichen es einem Server, sich glaubhaft für einen anderen auszugeben.[16]

8.3.1.3 Beteiligte Instanzen

Eine CA stellt für den Server im Rechenzentrum das Zertifikat aus. Dieser kann sich dann damit gegenüber dem Client (in diesem Fall dem mobile Gerät) authentifizieren und somit seine Identität bestätigen.

8.4 Kommunikation

E-Banking kann nicht lokal ausgeführt wird. Für den vollen Funktionsumfang ist eine Internet-verbindung notwendig. Diese Verbindung wird entweder über WLAN oder mit Hilfe des GSM oder UMTS Mobilfunknetz durchgeführt. Da sensible Daten übertragen werden, ist es notwenig, dass diese Kommunikation abgesichert wird. Dafür stehen folgende Möglichkeiten bereit:

8.4.1 Webservices (SOAP, REST)

SOAP ist ein Netzwerkprotokoll, das den Austausch von Nachrichten und den Aufruf von Remote Procedure Calls (RPC) ermöglicht. Die Basis für das Protokoll wird durch verschiedene Protokolle (http, https, ftp) gebildet. SOAP definiert nur, wie die Daten, die übertragen werden, formatiert sein müssen, damit die Schnittstellen zur Kommunikation sie richtig lesen und schreiben können. SOAP wird in XML verpackt. Am häufigsten wird die Kombination mit http verwendet.

Der Vorteil von SOAP ist, dass durch die Schnittstellendefinition auf dem Server auf bestimmte Ressourcen zugegriffen, dieser Zugriff jedoch sehr genau definiert werden kann.

SOAP ist derzeit in der Version 1.2 verfügbar und besitzt den Status einer W3C Recommendation.

SOAP definiert keine bzw nur sehr rudimentäre Sicherheitsfeatures. Die Sicherheit wird durch das darunterliegende Protokoll (zB https) gebildet. SOAP bietet durch bestimmte Erweiterungen die

Möglichkeit einer Authentifizierung, jedoch werden sowohl Passwort als auch Benutzername im Klartext übertragen.

REST wurde in der Dissertation von Thomas Roy Fielding vorgestellt und beschreibt einen Architekturstil für das World Wide Web. Es basiert auf den http Methoden GET, POST, PUT, DELETE. Mit diesen Methoden können Dokumente abgefragt, verändert, erstellt und gelöscht werden. Wie die Daten dargestellt werden, ist nicht genauer spezifiziert. Oft wird XML verwendet, da es sowohl für die Maschine als auch für den Menschen leicht zu lesen ist. Die Sicherheit wird wie bei SOAP durch das darunterliegende Protokoll (http, https) gebildet. REST bietet von sich aus keine Sicherheitsmerkmale. [31]

Webservices sind zustandslos. Jede Anfrage steht für sich und ist unabhängig von den vorangegangenen. Somit müssen alle für die Abarbeitung der Nachricht benötigten Informationen in dieser Nachricht enthalten sein. Das würde auch bedeuten, dass Authentifizierungsinformationen bei SOAP in jedem Paket enthalten sein müssten, was deren Auslesen erleichtern kann. [24]

8.4.1.1 Abgedeckte Gefahren

Mit SOAP kann die Schnittstelle auf dem Server genau definiert werden. Damit kann verhindert werden, dass auf Ressourcen zugegriffen wird, die nicht von extern verfügbar sein sollten. REST kann nur auf freigegebene URIs zugreifen. Bei der Verwendung von https kann sowohl ein Lauschangriff (vgl Seite 35, 6.7) als auch eine Man in the Middle Attacke (vgl Seite 38, 6.11) abgewehrt werden. Allgemein kann mit Webservices die Gefahr für das Rechenzentrum, durch den Kommunikationstunnel (vgl Seite 43, 6.14) angegriffen zu werden und das Ausnutzen von Server- und Konfigurationsfehler (vgl Seite 45, 6.16) verringert werden. Wie gut das Gefahrenpotential reduziert werden kann, hängt von der Qualität der Schnittstellendefinition am Server ab.

8.4.1.2 Zutreffende Gefahren

SOAP wird abhängig vom darunterliegenden Protokoll als Klartext im XML Format übertragen. Deshalb sind Webservices sehr anfällig für Lauschangriffe (vgl Seite 35, 6.7) und Man in the Middle Attacken (vgl Seite 38, 6.11), wenn http als Transportprotokoll verwendet wird.

Eine weitere Gefahr geht vom verwendeten XML aus. Es werden zusätzliche Bibliotheken benötigt, die Fehler beinhalten können. Außerdem kann durch spezielle Angriffe ein Absturz oder Datenverlust ermöglicht werden. Ein Beispiel hierfür ist die XML-Bomb. Hier handelt es sich um ein kleines Dokument, das durch Entpacken beim Parser mehrere Terabyte Größe erreicht und somit den Server lahmlegen kann.

8.4.1.3 Beteiligte Instanzen

Der Client stellt eine Anfrage über den vom darunterliegenden Protokoll bereitgestellten Übertragungskanal an den Server. Sowohl die Zugriffssicherheit als auch die Übertragungssicherheit hängt vom Übertragungsprotokoll ab.

8.4.1.4 Fazit

SOAP bietet die Möglichkeit, Schnittstellen zu definieren und somit den Zugriff aufs Rechenzentrum einzuschränken. Die Sicherheit ist jedoch sehr stark von den anderen verwendeten Übertragungsprotokollen abhängig. Weiters benötigt die Konvertierung in XML zur Übertragung viel Rechenzeit und erhöht somit die Laufzeit und Komplexität der Kommunikation.

8.4.1.5 Testprogramm

Webservices werden von Android nicht direkt unterstützt. Es gibt im Unterschied zum „normalen“ JDK keine Möglichkeit, eine WSDL Datei zu importieren. Um Webservices unter Android nutzen zu können, sind eigene Bibliotheken wie „kSOAP 2“ notwendig.[30]

8.4.2 Webzugriffe (http, https)

Die Protokolle http und https stellen im Internet den Standard zur Datenübertragung dar. Sie ermöglichen den Zugriff auf Webpages von unterschiedlichen Anwendungen. In den meisten Fällen wird die Anfrage aus einem Browser gestartet.

Das Protokoll http verwendet bei der Übertragung keine Verschlüsselung und ist damit sehr unsicher. Es sendet alle Daten im Klartext.

Das Protokoll https verwendet denselben Aufbau wie http, nutzt im Transportlayer zusätzlich zu TCP noch eine SSL/TLS Verschlüsselung (vgl Seite 80, 8.4.3) bei der Übertragung. Dadurch werden die Datenpakete sicherer übertragen als bei http.

Der Vorteil gegenüber SOAP ist, dass die Schnittstellen fast so genau definiert werden können, jedoch die Anfrage nicht in XML verpackt werden muss.

8.4.2.1 Abgedeckte Gefahren

Da bei http sowohl die Anfrage als auch die Antwort im Klartext übertragen werden, ist das Protokoll unsicher und sollte nicht für sicherheitskritische Anwendungen verwendet werden. Bei https wird dieses Problem durch die SSL Verschlüsselung behoben. SSL gilt als sicher. Somit können mit https sowohl ein Lauschangriff (vgl Seite 35, 6.7) als auch eine Man in the Middle Attacke (vgl Seite 38, 6.11) abgewehrt werden:

Die Sicherheit basiert auf SSL/TLS. Sie kann durch die Verwendung eines Serverzertifikates weiter erhöht werden. (vgl Seite 80, 8.4.3)

8.4.2.2 Zutreffende Gefahren

Http verwendet keine Verschlüsselung zur Übertragung der Daten. Somit ist es anfällig für Lauschangriffe (vgl Seite 35, 6.7) und Man in the Middle Attacken (vgl Seite 38, 6.11). Https verschlüsselt die Daten und ist somit nicht davon gefährdet. Jedoch kann durch den Verschlüsselungstunnel an der Firewall vorbei direkt auf den Server zugegriffen werden (vgl Seite 43, 6.14). Funktioniert die Hardware des Endgerätes nicht richtig (vgl Seite 34, 6.6), kann es dazu kommen, dass unverschlüsselte Daten ausgelesen werden können. Bei beiden Protokollen kann es, wie zB beim Apache Webserver[35], zu Fehlern kommen, die einem Angreifer Zugriff auf den Server gewähren (vgl Seite 45, 6.16).

Durch die Verwendung von https können viele Gefahren abgedeckt werden, jedoch ist darauf zu achten, dass der Schlüsselaustausch ohne Zertifikat beim SSL/TLS Handshake nicht durch eine Man in the Middle Attacke abgehört werden kann. (vgl Seite 38, 6.11)

Problematisch für die Sicherheit ist auch der sogenannte HTTP Responce Splitting Angriff. Dabei ist es einem Angreifer möglich, durch Einfügen zusätzlicher Zeilen im Header eines HTTP Pakets unter Umständen die Nutzdaten zu beeinflussen. Diese Gefahr kann jedoch durch genaue Überprüfung der mitgesendeten Benutzereingaben verringert werden. HTTP Responce Splitting kann als Vorbereitung für Cross-Side-Scripting und Web-Cache-Poisioning verwendet werden.

8.4.2.3 Beteiligte Instanzen

Bei http und https sind nur die Kommunikationspartner (Server und Client) beteiligt. Der Client stellt eine Anfrage, die vom Server beantwortet wird. Beim https-Protokoll wird vor der eigentlichen Übertragung noch der Sitzungsschlüssel ausgetauscht, wofür ein Serverzertifikat notwendig ist.

8.4.3 TLS (Transport Layer Security)

TLS ist ein Netzwerkprotokoll, das auf dem ISO-OSI Layer 4 (Transportschicht) die Kommunikation verschlüsselt. TLS basiert auf SSL, das ab der Version 3 als TLS bezeichnet wird. TLS verwendet TCP zur Übertragung. Als Verschlüsselungsalgorithmus wird meist AES oder RSA verwendet. Das Protokoll dient als Basis für viele Protokolle aus den höheren ISO-OSI Schichten wie https. (vgl Seite 79, 8.4.2)

Beim TLS-Handshake wird mittels asynchronem Schlüssel (zB Serverzertifikat (vgl Seite 76, 8.3.1)) ein synchroner Sitzungsschlüssel vereinbart. Dieser wird für die weitere Datenübertragung verwendet.

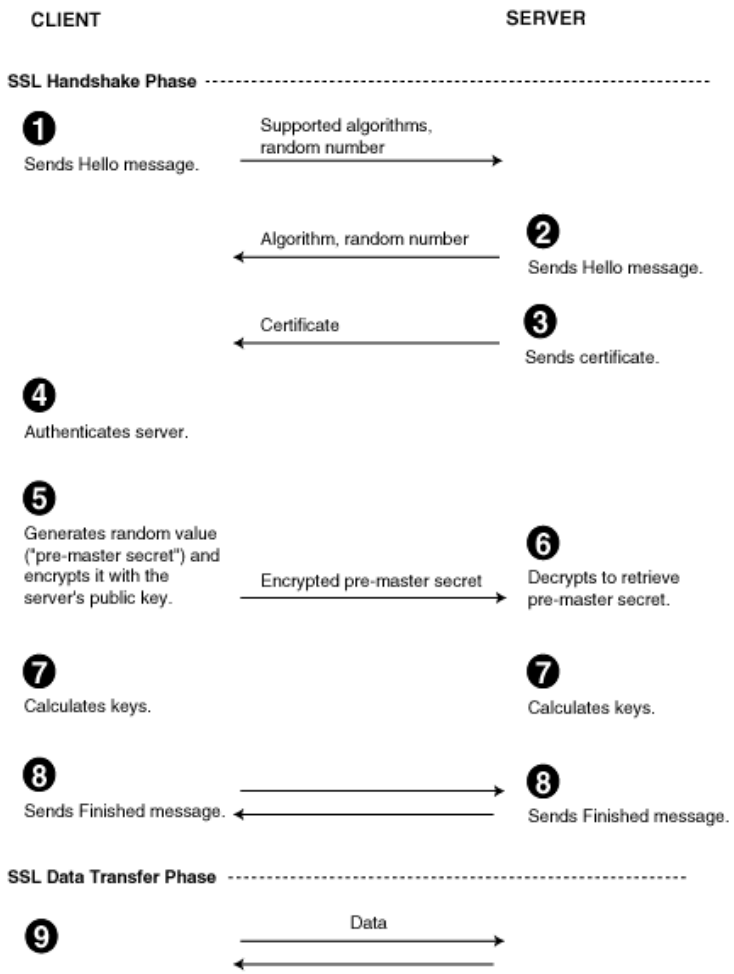


Abb. 25. TLS-Handshake, Schlüsselaustausch [25]

8.4.3.1 Abgedeckte Gefahren

Die kritische Stelle bei TLS ist der Schlüsselaustausch. Er kann durch das Serverzertifikat abgesichert werden. Durch die Übertragung mittels TLS bei der Verwendung eines vertrauenswürdigen Serverzertifikats kann sowohl einem Lauschangriff (vgl Seite 35, 6.7) als auch einer Man in the Middle Attacke (vgl Seite 38, 6.11) vorgebeugt werden.

8.4.3.2 Zutreffende Gefahren

Die Sicherheit der TLS Verschlüsselung kann durch eine Hardwaremanipulation (vgl Seite 34, 6.6) oder das Ausnutzen von Software- und Konfigurationsfehlern (vgl Seite 45, 6.16) beeinträchtigt werden. Da TLS einen kryptografischen Tunnel ins Rechenzentrum aufbaut, kann dieser auch für einen Angriff verwendet werden. (vgl Seite 43, 6.14)

8.4.3.3 Beteiligte Instanzen

Der Client kommuniziert mit dem Server über den TLS Tunnel. Dieser kann als sicher angesehen werden. Wichtig ist, dass der Handshake geschützt wird, was durch das Serverzertifikat sichergestellt werden kann. Da die Sicherheit somit vom Zertifikat abhängig ist, muss dieses und die CA, die es ausgestellt hat, geschützt werden.

8.4.4 VPN (Virtual Private Network)

Eine VPN Verbindung verbindet einen externen Client durch ein anderes Netz mit einem bestimmten Zielnetz. Als Anwendungsbeispiel kann hier ein externer Mitarbeiter einer Firma angeführt werden, der auf interne Ressourcen zugreifen muss. Die Verbindung erfolgt in den meisten Fällen verschlüsselt. Dazu baut VPN auf IPSec, TLS oder andere Protokolle auf. Die Variante mit TLS wird nur für eine End-to-Site Verbindung verwendet. (vgl Seite 80, 8.4.3) IPSec kann zusätzlich zur Verbindung eines Endgerätes mit der Site auch zwei Sites miteinander verbinden.

IPSec arbeitet im Gegensatz zu TLS auf dem Layer 3 des ISO-OSI Modells. Es gewährleistet Vertraulichkeit, Authentizität und Integrität der Kommunikation.

Die Bibliotheken zur Erstellung von VPN Verbindungen sind erst ab der Version 4.0 von Android verfügbar.

8.4.4.1 Abgedeckte Gefahren

VPN verwendet entweder TLS oder IPSec zur verschlüsselten Übertragung. Dadurch kann einem Lauschangriff (vgl Seite 35, 6.7) und einem Man in the Middle Angriff (vgl Seite 38, 6.11) entgegengewirkt werden.

Bei der Verwendung von SSL ist die Sicherheit wie bei Kapitel 8.4.3 TLS (Transport Layer Security) vom Serverzertifikat abhängig. Dies trifft bei IPSec nur bei dem zertifikatbasierten Verbindungsaufbau zu.

8.4.4.2 Zutreffende Gefahren

Die Gefährdungen, die durch VPN für das System entstehen, hängen von den gewählten Methoden ab. Bei der Verwendung von IPSec und einem pre-shared Key zur Verschlüsselung muss vor allem auf Schadsoftware (vgl Seite 31, 6.2), reverse Engineering (vgl Seite 32, 6.3) geachtet und ein physischer Zugriff (vgl Seite 33, 6.4) durch den Angreifer verhindert werden. Allgemein wird die Sicherheit von VPN durch Hardwaremanipulation (vgl Seite 34, 6.6) und Software- und Konfigurationsfehler (vgl Seite 45, 6.16) beeinträchtigt. Da VPN einen verschlüsselten Tunnel ins Rechenzentrum aufbaut, kann dieser auch für Angriffe genutzt werden (vgl Seite 43, 6.14).

Die Sicherheit basiert in den meisten Konfigurationsmöglichkeiten auf Zertifikaten oder einem pre-shared Key. Die Sicherheit dieser Daten muss gewährleistet werden, da bei Verlust die Kommunikation leicht mitgelesen werden kann. Ein pre-shared Key sollte nicht verwendet werden, da dieser oft geändert werden muss, um die Sicherheit gewährleisten zu können. Da bei der Verwendung von Zertifikaten für jede Session ein eigener Schlüssel verwendet wird, besteht ein geringeres Sicherheitsrisiko.

8.4.4.3 Beteiligte Instanzen

Bei einem Virtual Private Network kommuniziert der Client mit dem Server durch einen kryptografisch geschützten Tunnel. Dabei kann sowohl SSL als auch IPSec für die Sicherung des VPN Tunnels verwendet werden. In beiden Fällen sollte der Schlüsselaustausch mit einem Serverzertifikat durchgeführt werden. Somit ist es wichtig, dass die ausstellende CA vertrauenswürdig und geschützt ist.

8.4.5 MBS/IP Multi-purpose Business Security over IP

MBS/IP wurde von der RACON Software GmbH Linz im Auftrag der STUZZA für e-Banking und Telebanking entwickelt. Es ist ein Netzwerkprotokoll, das auf TCP basiert und für die Verschlüsselung SSL verwendet. Die Daten werden ähnlich wie bei FTP in Form von Dateien über den Port 3048 ausgetauscht. Übertragen werden die Daten meistens im EDIFACT Format. Die aktuellste Version sieht die Verwendung von XML Dateien vor.

Das Protokoll wird von Android nicht direkt unterstützt. Die verfügbaren Javabibliotheken können aber für die Verwendung mit Android angepasst werden. Die Bibliotheken sind lizenzpflichtig.

MBS ist ein Standard und wird seit Jahren von den österreichischen Banken verwendet. Der Vorteil ist somit, dass mit einer APP mehrere Banken angesprochen und auch mehrere Bankkonten unterschiedlicher Banken verwaltet werden können. [26][27]

8.4.5.1 Abgedeckte Gefahren

Das Protokoll wurde für die sichere Übertragung von Bankdaten entwickelt. Aus diesem Grund sind alle Gefährdungen, die die Übertragungssicherheit bedrohen, wie der Lausch- (vgl Seite 35, 6.7) und Man in the Middle Angriff (vgl Seite 38, 6.11) und der zweckentfremdete Zugriff durch den Kommunikationstunnel (vgl Seite 43, 6.14) abgedeckt.

8.4.5.2 Zutreffende Gefahren

Da die Sicherheit auf TLS basiert, wird das System von denselben Gefährdungen bedroht. (vgl Seite 80, 8.4.3)

8.4.5.3 Beteiligte Instanzen

Ein Client sendet die Anfrage durch den zuvor aufgebauten MBS/IP Tunnel zum Server. Dieser nutzt denselben Tunnel, um darauf zu antworten. MBS/IP ist jedoch lizenzpflichtig und kann nicht frei verwendet werden. Da TLS zur Verschlüsselung verwendet wird, ist ein Serverzertifikat notwendig. Dieses muss von einer vertrauenswürdigen CA ausgestellt werden.

8.4.6 Erkennung von Zwischenhops (Proxy, Man in the Middle)

Verläuft die Kommunikation zwischen dem Endgerät und dem Server im Rechenzentrum über Zwischenhops wie Proxies oder Gateways, kann es sein, dass die Verschlüsselung unterbrochen wird, und die übertragenen Daten im Klartext auslesbar sind. Um diese Unterbrechung zu verhindern, ist es notwendig, derartige Hops zu erkennen oder Zwischenhops, die die Verschlüsselung aufbrechen, zu vermeiden. Methoden dazu sind einerseits Serverzertifikate, die es ermöglichen, dass nachfolgende Komponenten erkennen, ob der Endpunkt der Kommunikation der richtige ist und andererseits Übertragungsprotokolle, die Verschlüsselung unterstützen und für den Datenverkehr einen sicheren Tunnel aufbauen. (zB https, SSL, VPN, MBS/IP)

Die Gefährdungen, die dadurch abgedeckt oder hervorgerufen werden, sind von den umgesetzten Methoden abhängig (vgl Seite 76ff, 8.3.1; 8.4)

8.5 Applikationssicherheit

Neben der Sicherheit der Identitäten der Kommunikationspartner und der Übertragung muss auch die Sicherheit der Anwendung selber und der von ihr verarbeiteten Daten sichergestellt werden. Das ist für die sensiblen Daten des Benutzers wichtig, jedoch auch für die Integrität des Rechenzentrums. Gelingt es einem Angreifer, eine APP zu schreiben, die an die Schnittstelle der Server andockt, kann er eigene Kommandos an den Server senden und umgeht dadurch eventuelle Kontrollen in der APP.

8.5.1 Schutz der APP gegen Veränderung von Außen

Die Daten der APP, bestehend aus den Binarys der Anwendung und den Daten, die von der Anwendung in den Speicher geschrieben werden, sollen vor unberechtigtem Zugriff und Veränderung geschützt werden.

Die Daten, die in den Speicher geschrieben werden, können nur von der UserId gelesen werden, die die Daten geschrieben hat. Da die UserId für eine APP eindeutig ist, können keine anderen Anwendungen auf diese Daten zugreifen. Eine Ausnahme bilden hier Anwendungen, die als Root

ausgeführt werden. Rootrechte können nur auf gerooteten Geräten vergeben werden. (vgl Seite 22, 5.2.1)

8.5.1.1 Sicherheit des/durch Erstellerzertifikats

Jede APP, die über den Market angeboten wird, muss zertifiziert sein. Das Zertifikat kann jedoch selbstsigniert sein. Die Quelle des Zertifikates ist nicht signiert und somit nicht vertrauenswürdig. Dennoch erhöhen diese unsicheren Zertifikate die Sicherheit einer Androidanwendung. Mit Hilfe des Zertifikates kann überprüft werden, ob bei Anwendungen, die in derselben Sandbox laufen sollen, auch der Entwickler dieser APPs derselbe ist.

Jede Anwendung unter Android wird unter einem eigenen Useraccount in einer eigenen DalvikVM ausgeführt. Die Daten sind somit vor unberechtigtem Zugriff geschützt, solange keine andere Anwendung mit demselben Account oder einem Benutzer mit zusätzlichen Rechten (Root) ausgeführt wird. Der Useraccount wird entweder vom System automatisch erstellt oder kann in der Manifest-Datei als „sharedUserId“ angegeben werden. Bei manueller Angabe wird das Programm unter der „sharedUserId“ ausgeführt und teilt sich somit die Sandbox mit den Programmen derselben UserId. Dadurch ist es möglich, dass die APP auf Daten von anderen Anwendungen in der Sandbox zugreifen kann, auch wenn diese nicht freigegeben wurden. Um dies zu verhindern, wird bei der Installation der APP nicht nur die UserId, unter der die APP laufen wird, überprüft sondern auch das Zertifikat, mit dem die Anwendungen bei der Freigabe signiert wurden. Stimmen beide überein, geht das System davon aus, dass die Anwendungen vom selben Entwickler sind und somit kein Sicherheitsproblem besteht. [22](Seite 26)

Ein Angreifer benötigt somit nicht nur die UserId, unter der die Anwendung ausgeführt wird, sondern auch den privaten Schlüssel des Entwicklers. Aus diesem Grund ist es wichtig, diesen Schlüssel geheim zu halten und nicht den default Key aus dem Entwicklerzertifikat der Entwicklungsumgebung zu verwenden. Der default Key sollte nur für Testzwecke genutzt werden.

8.5.1.2 Signaturüberprüfung

Die Signatur, die für die Signierung der APP verwendet wurde, wird beim Start der Anwendung aus den Binarys ausgelesen und mit der originalen Signatur vom Server gegengeprüft. Damit kann sichergestellt werden, dass die APP nicht verändert wurde. Da eine APP nur mit einer gültigen Signatur ausgeführt werden kann, kann somit die Integrität der Anwendung bis zu einem gewissen Grad gewährleistet werden. Problematisch sind Replayattacken oder eine eigene APP, die mit Hilfe von Reverse Engineering erstellt wurde und die geforderte Signatur als Ressource verwendet.

8.5.1.3 Abgedeckte Gefahren

Durch den Zugriffsschutz und die Integritätsprüfung kann reverse Engineering (vgl Seite 32, 6.3) und einem unerlaubten Zugriff durch den Kommunikationstunnel auf das Rechenzentrum (vgl Seite 43, 6.14) vorgebeugt werden.

Veränderungen an der APP am Gerät sind durch die Überprüfung der Signatur nicht mehr möglich. Es muss jedoch auch sichergestellt werden, dass es nicht sein kann, dass eine gehackte APP nur einen Hashwert sendet und somit eine falsche Integrität bestätigt.

8.5.1.4 Zutreffende Gefahren

Besteht die Möglichkeit, dass die Signatur vor oder während der Übertragung durch eine Man in the Middle Attacke verändert wird (vgl Seite 38, 6.11), so kann die Integrität der Daten nicht mehr überprüft werden. Weiters kann der geforderte Hashwert direkt von einer gefälschten APP an den Server gesendet werden. Das Rechenzentrum kann nicht nachprüfen, ob der Hash hardcoded gesendet wird, oder ob er berechnet wurde (vgl Seite 32, 6.3).

8.5.1.5 Beteiligte Instanzen

Bei der Erstellung der APP wird diese mit dem Erstellerzertifikat signiert. Diese Signatur kann ausgelesen und an den Server gesendet werden. Es muss sichergestellt werden, dass dieser unverändert im Rechenzentrum ankommen, um ihn dort mit dem originalen Hash vergleichen zu können. Weiters wird diese Signatur vom Betriebssystem verwendet, um die APP zu identifizieren und den Zugriff auf die lokal gespeicherten Daten zu gewähren.

8.5.1.6 Testprogramm für das Auslesen des Signaturzertifikates

Das Erstellerzertifikat kann mit Hilfe eines Packagemanagers ausgelesen werden. Dabei werden alle Signaturen eingelesen und danach durch Durchiterieren für jede Signatur das entsprechende Zertifikat gesucht. Aus dem Zertifikat können die Informationen über den Ersteller des Paketes ausgelesen werden.

```
PackageInfo pkgInfo = getPackageManager().getPackageInfo(getPackageName(),
     PackageManager.GET_SIGNATURES);

for (Signature appSignature : pkgInfo.signatures) {
     X509Certificate appCertificate =
     X509Certificate.getInstance(appSignature.toByteArray());
     // can give you the public key:
     appCertificate.getPublicKey();
     // will give you a Principal name:
     appCertificate.getSubjectDN().getName();
}
```

Code. 2. Auslesen des Signaturzertifikates

Die Überprüfung der Integrität der APP mit Hilfe eines Hashwertes ist nicht zielführend, da dieser nicht fehlerfrei erstellt und durch Reverse Engineering (vgl Seite 32, 6.3) leicht manipuliert werden kann.

Die Erstellung eines Hashwertes der Binarys der APP ist auf ungerooteten Geräten unmöglich, da nicht auf die apk-Dateien im Telefonspeicher zugegriffen werden kann.

8.5.2 E-Banking per SMS auf dem Gerät sperren

Wenn das Smartphone verloren geht oder gestohlen wird, erlangt ein potentieller Angreifer physikalischen Zugriff auf das Gerät. (vgl Seite 33, 6.4) Befindet sich darauf ein e-Banking Zugang, hat er auch darauf Zugriff. Mit Hilfe einer SMS, die an das Gerät gesendet wird, könnte die Applikation deaktiviert und eventuelle Daten gelöscht werden. Damit ist es einem Angreifer nicht mehr möglich, sie zu nutzen oder auszulesen. Dazu muss die e-Banking APP ein Service verwenden, das alle SMS liest und auf bestimmte reagiert.

8.5.2.1 Abgedeckte Gefahren

Durch die SMS kann die e-Banking APP auf dem mobilen Gerät gesperrt werden. Damit kann sie dort bis zu dem Zeitpunkt nicht mehr verwendet werden, wo sie wieder durch die Bank aktiviert wird, wodurch die Gefährdung des physikalischen Zugriffes (vgl Seite 33, 6.4) entkräftet werden kann.

8.5.2.2 Zutreffende Gefahren

Es ist jedoch zu bedenken, dass die SMS von jedem gesendet und somit als DoS Attacke (vgl Seite 37, 6.10) von einem Angreifer verwendet werden kann. Weiters ist zu beachten, dass ein Angreifer durch Social Engineering (vgl Seite 30, 6.1) Informationen zu dieser Sicherungsmethode erhalten und somit das Empfangen von SMS deaktivieren kann.

8.5.2.3 Beteiligte Instanzen

Es wird ein Gerät benötigt, von dem aus eine SMS an das Smartphone, auf dem e-Banking läuft, gesendet werden kann. Die Mitteilung wird über das Netz des Providers an das Endgerät gesendet, auf welchem die APP die Nachricht liest und die Zugangsdaten löscht.

8.5.2.4 Sicherheitsproblem

Dadurch, dass der Zugang mittels SMS Versand gesperrt werden kann, muss sichergestellt werden, dass nur SMS vom wirklichen Benutzer zu einer Sperrung führen. Ansonsten führt diese Funktionalität zu einer Sicherheitslücke, die einen DoS Angriff ermöglicht. Im Internet gibt es

mehrere Plattformen, von denen SMS mit einstellbarer Absendernummer versendet werden können. Für den Empfänger ist es nicht nachvollziehbar, woher die SMS wirklich kommt.

Abhilfe schafft hier nur die Verwendung einer elektronischen Signatur im Nachrichtentext der Kurznachricht. Diese wird mit einem privaten Schlüssel des Rechenzentrums erzeugt und ermöglicht eine eindeutig Identifizierung des Absenders. Die Sicherheit dieser Methode wird jedoch durch die geringe Nachrichtenlänge und die Möglichkeit von Replay-Attacken eingeschränkt.

Ein weiteres Problem stellt das Lesen der SMS dar. Um den Zugang zu sperren, muss jede ankommende Nachricht von der APP gelesen werden. Somit muss die APP durchgehend gestartet sein. (Service) Dies benötigt Systemressourcen und führt eventuell zu Verletzungen des Datenschutzes, da auch private SMS analysiert werden müssen.

8.5.2.5 Fazit

Da es nicht möglich ist, den Ursprung einer SMS am Endgerät festzustellen, kann dieser Mechanismus leicht für einen DoS Angriff genutzt werden. Eine Verwendung als Sicherheitsfeature ist grundsätzlich nicht empfehlenswert. Wird jedoch eine passende Authentifizierung mit der SMS mitgesendet, sodass der Absender identifiziert werden kann, wäre eine Implementierung denkbar.

8.5.3 Erkennung von gerooteten Geräten und deren Absicherung

Wird ein Gerät gerootet, ist es möglich, dass APPs eine Superuser Berechtigung anfordern, und damit Vollzugriff auf das System erhalten. Dadurch würden sie auch Zugriff auf Daten von anderen Anwendungen erhalten, die eigentlich in einem sicheren Bereich liegen. (vgl Seite 22, 5.2.1) In diesem Fall sollte der Benutzer zumindest gewarnt werden, dass sein System eventuell unsicher ist, und dass die Sicherheit von e-Banking durch Malware eingeschränkt wird.

Möglichkeiten zur Erhöhung der Sicherheit sind ua das Scannen der aktuell laufenden oder installierten Programme. (vgl Seite 89, 8.5.4) Basierend auf diesem Ergebnis kann ein Verweigern der Funktionalität oder ein Warnen des Benutzers anhand eines Vergleichs mit Softwareblacklisten durchgeführt werden. (vgl Seite 92, 8.5.5)

8.5.3.1 Testprogramm

Die Überprüfung, ob das Gerät gerootet ist, ist verhältnismäßig schwierig, da die Systeme abhängig vom Hersteller unterschiedlich aufgebaut sind. Die einfachste Methode, auf Root zu überprüfen, ist das Anfordern von su- (root-) Rechten.

```
Process proc = Runtime.getRuntime().exec("su");
```

Code. 3. Ausführen eines Prozesses als Root

Das Ausführen eines Prozesses als Root ist nur auf gerooteten Geräten möglich. Bei ungerooteten wird durch die Anweisung aus Code. 3 eine Exception geworfen, bzw bleibt die Variable proc auf dem Wert „null“. Bei einem gerooteten Gerät würde der Benutzer nach der Superuser Berechtigung gefragt werden. Dies wiederum kann Skepsis hervorrufen und dazu führen, dass die APP als Malware markiert wird.

8.5.3.2 Benötigte Berechtigungen

Es sind keine Berechtigungen für diese Funktionalität notwendig.

8.5.4 Erkennung von installierter und/oder laufender Software und deren Berechtigungen

Es wird die Liste der installierten Software (Packages- und Klassennamen) aufgerufen und anhand dieser nach Schadsoftware gesucht. Es werden von jeder installierten APP die angeforderten Berechtigungen abgefragt. Basierend auf diesen Informationen entscheidet die APP, ob eine Anwendung eventuell ein Sicherheitsrisiko darstellt.

Da die installierten Anwendungen anhand der Berechtigungen analysiert werden, kann es dazu führen, dass auch nicht gefährliche APPs, wie die Systemprogramme „Taskmanager“ und „Einstellungen“ als Schadsoftware erkannt werden.

8.5.4.1 Abgedeckte Gefahren

Durch die Erkennung von gefährlicher Software können Malware und Viren erkannt werden. (vgl Seite 31, 6.2) Je nachdem, welche Berechtigungen als gefährlich angesehen werden, muss von einer erhöhten False-Error-Rate oder False-Acceptance-Rate ausgegangen werden. Durch Black- und Whitelisten können diese Raten verringert werden.

Es muss bedacht werden, dass die Überprüfung auf schädliche Software zwar die Sicherheit stark erhöhen kann, jedoch viel Rechenzeit benötigt und somit die Benutzerfreundlichkeit einschränkt.

8.5.4.2 Zutreffende Gefahren

Durch das Scannen nach schädlicher Software wird die Gefahr für die Bankanwendung nicht erhöht. Jedoch muss beachtet werden, dass durch eine gefälschte APP (vgl Seite 32, 6.3) dem Server mitgeteilt wird, dass ein Scandurchgang ohne Fund war, ohne dass dieser es nachprüfen kann.

8.5.4.3 Beteiligte Instanzen

An der Durchführung des Scans ist nur das Endgerät beteiligt. Der Scan kann lokal ausgeführt werden. Es muss jedoch die Liste der möglichen Berechtigungen laufend aktualisiert werden, sodass neue Berechtigungen möglichst früh mit in die Überprüfung einfließen können. Es sollte

auch eine Whitelist existieren, in der bekannte Anwendungen angeführt werden, die kein Sicherheitsrisiko darstellen. Das Updaten dieser Listen kann durch ein APP Update durchgeführt werden. Ob der Scan wirklich durchgeführt wurde, kann vom Rechenzentrum jedoch nicht überprüft werden.

8.5.4.4 Beispiele für Berechtigungen, die eine Bedrohung darstellen können

Um auf bestimmte Ressourcen zugreifen zu können und um bestimmte Funktionen ausführen zu dürfen, ist es notwendig, dass die APP Berechtigungen anfordert. Teilweise fordern APPs mehr Berechtigungen an, als für deren eigentlichen Funktionsumfang notwendig sind. Durch folgende Berechtigungen kann eine Gefahr für das Gerät, die Daten und die anderen installierten APPs ausgehen. [29]

- Durch die Permission „`BRICK`“ wird der APP erlaubt, das System des Gerätes zu zerstören und somit für den Benutzer unbrauchbar zu machen. Diese Berechtigung sollte niemals akzeptiert werden.
- Eine Netzwerkkommunikation (`INTERNET, BLUETOOTH, BLUETOOTH_ADMIN`) ermöglicht das Senden und Empfangen von Daten. Somit können Informationen über das Gerät unbemerkt vom Benutzer an die Umwelt gesendet werden.
- Durch das Lesen von persönlichen Daten (`READ_SMS, READ_CONTACTS, READ_CALENDER`) kann auf das Privatleben des Benutzers zurückgeschlossen werden. Weiters können SMS Informationen zum Bankzugang enthalten (zB mTAN)
- Das Schreiben von persönlichen Daten (`WRITE_SMS, WRITE_CONTACTS, WRITE_CALENDER`) ermöglicht ein Verändern von Informationen, denen der Benutzer vertraut. Eine Telefonnummer wird vor dem Anruf oder Versenden der SMS nicht erneut kontrolliert.
- Durch das Lesen oder Speichern auf der SD Karte (`WRITE_EXTERNAL_STORAGE, READ_EXTERNAL_STORAGE`) kann ein Angreifer Zugriff auf die Daten des Gerätes erhalten, ohne physikalisch auf dieses zugreifen zu müssen. Diese Berechtigung ist nur dann gefährlich, wenn auf der Speicherkarte Daten abgelegt werden.
- Die administrative Verwaltung der SD-Karte (`MOUNT_FORMAT_FILESYSTEMS, MOUNT_UNMOUNT_FILESYSTEMS`) kann für DoS Angriffe verwendet werden, da damit einerseits alle Daten der Speicherkarte durch Neuformatierung gelöscht, oder die SD-Karte getrennt und wieder verbunden werden kann.
- Werden die bevorzugten Anwendungen (`SET PREFERRED_APPLICATIONS`) verändert, kann es vorkommen, dass zB SMS standardmäßig nicht mehr mit der systemeigenen APP abgerufen werden sondern mit einer Malware Anwendung. Somit kann einerseits die Verwendbarkeit des

Gerätes eingeschränkt werden, wenn Dateitypen mit Programmen geöffnet werden, die diesen Typ nicht verstehen und andererseits vom Benutzer unbemerkt die geöffneten Dateien von Malware analysiert und wichtige Informationen an die Außenwelt gesendet werden.

- Mit Hilfe der Berechtigung „`WRITE_SECURE_SETTINGS`“ können zusätzliche Sicherheitslücken geöffnet werden. (zB Aktivieren der Installation von unsignierten APPS)
- Das Verwalten von anderen APPS (`INSTALL_PACKAGES, DELETE_PACKAGES`) ist auf einem ungerooteten Gerät nicht möglich. Diese Berechtigungen dürfen nur von Systemapplikationen angefordert werden. Kritisch ist es jedoch auf Gerooteten, da damit zusätzliche Malware installiert oder Sicherheitssoftware automatisch deinstalliert werden kann.
- Die Permission, das Gerät neuzustarten (`REBOOT`), kann für DoS Angriffe verwendet werden. Durch wiederholte Neustarts wird eine Benützung unmöglich.

Zu beachten ist jedoch immer, dass die Berechtigungen auch notwendig sein können und für die eigentliche Funktionalität wichtig sind. Somit muss von diesen Berechtigungen nicht unbedingt eine Gefahr ausgehen.

8.5.4.5 Testprogramm

Laufende APPs können mit Hilfe des ActivityManagers ausgelesen werden. Für die Liste der installierten APPs und Packages ist ein PackageManager notwendig. Die Berechtigungen sind für die einzelnen Packages direkt festgelegt und können somit auch nur aus einer PackageInfo ausgelesen werden. Dafür kann folgendes Codefragment verwendet werden:

```
final List<PackageInfo> appinstall =
     this.getPackageManager().getInstalledPackages(PackageManager.GET_PERMI
     SSIONS);
for (int i = 0; i < appinstall.size(); i++) {
     //Packagename: appinstall.get(i).packageName
     String[] perm = appinstall.get(i).requestedPermissions;
     if (perm == null) {
        //Keine Berechtigung für dieses Paket benötigt
     } else {
        for (int n = 0; n < perm.length; n++) {
              // Berechtigungsname: perm[n].toString());
}}}
```

Code. 4. Installierte Packages und deren Berechtigungen auslesen

8.5.4.6 Benötigte Berechtigungen

Das Auslesen der installierten Packages, sowie der laufenden Activities (`ActivityManager.getRunningAppProcesses`) und Services (`ActivityManager.getRunningServices`) ist immer möglich. Für die aktuellen Tasks (`ActivityManager.getRunningTasks`) wird die Berechtigung „`GET_TASKS`“ benötigt.

8.5.5 Softwareblacklisting

Beim Softwareblacklisting geht es darum, dass die APP die derzeit installierten Anwendungen ausliest und diese Liste an den Server sendet. Der Server kann diese nun mit bekannter Schadsoftware vergleichen und so potentielle Gefährdungen erkennen. Alternativ können auch die derzeit laufenden APPs für den Vergleich herangezogen werden.

Durch diese Methode kann zwar einfach festgestellt werden, ob Schadsoftware installiert ist, jedoch kann nur nach bekannter Software gesucht werden. Die Liste der gefährlichen Programme muss oft aktualisiert werden, was einen großen Aufwand darstellt.

8.5.5.1 Abgedeckte Gefahren

Durch die Blacklist kann Schadsoftware erkannt und somit die Gefahr von Malware eingedämmt werden. (vgl Seite 31, 6.2) Die Effektivität hängt jedoch sehr stark von der Aktualität dieser Liste ab. Oft werden Programme erst sehr spät als Schadsoftware erkannt.

8.5.5.2 Zutreffende Gefahren

Da die Liste der Programme vom Server überprüft wird, muss die Kommunikation integer durchgeführt werden können. Weiters muss die Liste der Schadprogramme auf dem Server komplett und aktuell sein, um die Sicherheit gewährleisten zu können. Die Funktionsfähigkeit wird durch reverse Engineering (vgl Seite 32, 6.3) und Man in the Middle Angriffe (vgl Seite 38, 6.11) beeinträchtigt.

8.5.5.3 Beteiligte Instanzen

Das Endgerät stellt die Liste der Anwendungen zusammen und sendet diese an den Server. Dieser überprüft die Liste mit einer lokalen Blacklist und kann somit feststellen, ob das Gerät gefährdet ist. APPs, die zur Blacklist hinzugefügt werden, können somit bereits bei der nächsten Überprüfung erkannt werden. Es ist kein Update des Clients notwendig.

Jedoch benötigt dieses Sicherheitsmerkmal Netzwerktraffic und erfordert einen erhöhten Arbeitsaufwand für die Serverwartung. (Blacklisterstellung und Wartung)

8.5.5.4 Testprogramm

Das Auslesen der APPs funktioniert wie in „8.5.4 Erkennung von installierter und/oder laufender Software und deren Berechtigungen“ beschrieben. Für die Kommunikation können die in „8.4 Kommunikation“ beschriebenen Übertragungswege verwendet werden. Je nach verwendetem Kommunikationsprotokoll wird auch die Sicherheit der Funktion beeinflusst. (Abhörsicherheit bei Tunnelung)

8.5.6 Speicherung von Daten im Telefonspeicher

Im Telefonspeicher können Einstellungsdaten der APP einfach als Preferences abgelegt werden. Da bei einem ungerooteten Gerät die APPs nur auf ihre eigenen Daten zugreifen können, sind sie vor unberechtigtem Zugriff sicher.

Die Daten können jedoch nach dem Verbinden des Gerätes mit einem Computer durch eine Entwicklungsumgebung wie Eclipse mit Hilfe von DDMS ausgelesen werden. Bei einem gerooteten Gerät können sie auch von einer APP mit Root-Rechten gelesen werden. Eine Verschlüsselung sollte für wichtige Daten immer verwendet werden. (vgl Seite 96, 8.5.8)

8.5.6.1 Abgedeckte Gefahren

Durch die Speicherung der Daten im Telefonspeicher können keine Gefahren abgewehrt werden. Jedoch können Daten zwischengespeichert werden und müssen nicht erneut vom Benutzer eingegeben werden.

8.5.6.2 Zutreffende Gefahren

Die Daten im Telefonspeicher sind im Normalfall vom Androidsystem geschützt. Nur die eigene Anwendung kann darauf zugreifen. Wird auf das Gerät über eine Entwicklungsumgebung (zB Eclipse mit DDMS) zugegriffen, kann jedoch ein Großteil des Speiches ausgelesen werden (vgl Seite 33, 6.4). Bei gerooteten Geräten ist es teilweise auch für andere APPs möglich, auf diese zuzugreifen (vgl Seite 31, 6.2).

8.5.6.3 Beteiligte Instanzen

Die Daten werden direkt von der APP in den Telefonspeicher abgelegt. In welche Ordner sie gespeichert werden, wird vom Androidsystem festgelegt.

8.5.6.4 Testprogramm

- Für den Zugriff

Der lesende Zugriff auf den Telefonspeicher funktioniert am einfachsten mit Hilfe der shared Preferences. Nach der Definition des Zugriffs kann mit den get-Methoden auf die gespeicherten Werte zugegriffen werden.

```
SharedPreferences settings = getPreferences(MODE_PRIVATE);
String storedText = settings.getString("text", noPW);
```

Code. 5. Zugriff auf den internen Telefonspeicher (shared Preferences)

- Für die Speicherung

Zur Speicherung ist es notwendig, dass zusätzlich ein Editor auf die shared Preferences definiert wird. Werden die shared Preferences im „`MODE_PRIVATE`“ angelegt, kann nur die eigene APP darauf

zugreifen. Mit dem Editor können durch die put-Methoden unterschiedliche Datenwerte in die Datei geschrieben werden. Mit einem Commit wird der Speichervorgang beendet und die Daten endgültig im Telefonspeicher abgelegt.

```
SharedPreferences settings = getPreferences(MODE_PRIVATE);
SharedPreferences.Editor editor = settings.edit();
editor.putString("String", strVar);
editor.commit();
```

Code. 6. Speichern in den Telefonspeicher (shared Preferences)

8.5.7 Speicherung von Daten auf der Speicherkarte

Es gibt mehrere Gründe für die Speicherung von Daten auf der SD Karte. Einerseits können damit die APP Binarys ausgelagert werden und verbrauchen weniger Platz am internen Speicher des mobilen Geräts und andererseits können auf der Speicherkarte Benutzerdaten wie die Verfügernummer oder Standardeinstellungen gespeichert werden.

Zu beachten ist jedoch, dass eine Speicherkarte nicht mehr dem Schreib- und Leseschutz des Systems unterworfen ist. Eventuell geschützte Bereiche können spätestens nach dem Einlegen in einen Computer gelesen werden. Für wichtige Daten sollte eine Verschlüsselung verwendet werden. (vgl Seite 96, 8.5.8)

Weiters ist zu beachten, dass die SD Karte nicht immer verfügbar sein muss, zB wenn das mobile Gerät mit einem Computer verbunden ist oder es keine SD Karte besitzt.

8.5.7.1 Abgedeckte Gefahren

Durch die Speicherung der Daten auf der SD Karte können keine Gefahren abgewehrt werden. Es wird jedoch Speicherplatz auf dem internen Speicher freigegeben, der für andere APPs verwendet werden kann. Somit wirkt es positiv auf die Akzeptanz der Benutzer, wenn die APP selber bereits ziemlich groß ist.

8.5.7.2 Zutreffende Gefahren

Gefährlich für die Daten ist jeder direkte Zugriff auf die Speicherkarte. Andere Anwendungen, die ein Leserecht auf der SD Karte besitzen, haben auch Zugriff auf die Daten (vgl Seite 31, 6.2). Deshalb muss verhindert werden, dass diese die Daten auch weiterverwenden können. Eine Speicherkarte kann aber auch von einem Computer oder anderen Gerät mit Kartenleser gelesen werden. Somit ist auch der reine physikalische Zugriff auf das Smartphone für die Daten gefährlich (vgl Seite 33, 6.4).

8.5.7.3 Beteiligte Instanzen

Die Speicherkarte erweitert den internen Speicher eines Smartphones und ermöglicht somit dem Benutzer, zusätzliche Daten am Gerät zu speichern. Dieser Speicher kann im Gegensatz zum internen jederzeit ausgewechselt werden, um schnell auf neue Daten Zugriff zu haben. Der Speicher kann auch von anderen Geräten wie PCs gelesen und beschrieben werden.

8.5.7.4 Testprogramm

Die Speicherung der Binaries wird in der Manifest-Datei im Tag „manifest“ unter dem Eigenschaftswert „android:installLocation“ festgelegt. Es gibt drei Einstellmöglichkeiten:

- internalOnly
- auto
- preferExternal

Diese Funktionalität ist erst aber Android Version 2.2 verfügbar. In früheren Versionen ist eine APP nur im Telefonspeicher speicherbar. Bei Custom ROMs kann es sein, dass die Funktion schon früher implementiert ist.

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    android:installLocation="auto"
    package=”…
```

Code. 7. Speicherort der APP – Androidmanifest.xml

Durch diese Einstellung wird nur die .apk Datei auf der Speicherkarte abgelegt. Die anderen Daten, die im Normalfall im Telefonspeicher abgelegt werden (Datenbanken, sharedPreferences), werden auch weiterhin dort gespeichert.

Um Dateien auf der SD Karte anzulegen oder von dieser zu lesen, muss ein FileInputStream oder ein FileOutputStream verwendet werden. Außerdem muss die Lese- (`READ_EXTERNAL_STORAGE`) und Schreibberechtigung (`WRITE_EXTERNAL_STORAGE`) für die SD Karte in der Manifestdatei angefordert werden.

```
public String FileSD(boolean mod, String text) { //mod=True->write
     if (mod) {
// Text in eine Datei auf der SD Karte schreiben
       try {
             File txtDat = new
     File(Environment.getExternalStorageDirectory().getAbsolutePath()+
     File.separator + "Test.txt");
             if (!txtDat.exists()) {
                   txtDat.createNewFile();
             }
             FileOutputStream fOut = new FileOutputStream(myFile);
             OutputStreamWriter myOutWriter = new OutputStreamWriter(fOut);
             myOutWriter.append(text);
             myOutWriter.close();
             fOut.close();
```

```
        } catch (IOException e) { e.printStackTrace(); }
        return null;
      }
// Text aus einer Datei auf der SD Karte auslesen
      try {
        File txtDat = new
      File(Environment.getExternalStorageDirectory().getAbsolutePath()+
      File.separator + "Test.txt");
        FileInputStream fIn = new FileInputStream(txtDat);
        BufferedReader myReader = new BufferedReader(new
      InputStreamReader(fIn));
        String aDataRow = "";
        String aBuffer = "";
        while ((aDataRow = myReader.readLine()) != null) {
              aBuffer += aDataRow + "\n";
        }
        myReader.close();
        return aBuffer;
      } catch (IOException e) { e.printStackTrace(); }
      return null;
}
```

Code. 8. Dateizugriff SD Karte

8.5.8 Verschlüsselung der gespeicherten Daten

Werden Daten der APP auf der Speicherkarte gespeichert, können diese leicht durch eine andere Anwendung oder durch einen Computer ausgelesen werden. Daten aus dem Telefonspeicher sind zwar schwieriger auszulesen, jedoch ist es nicht unmöglich. Um das Auslesen zu verhindern, sollten die Daten verschlüsselt abgelegt werden. Die Verschlüsselung kann nur auf Daten und nicht auf die Binaries der APP angewendet werden. Der dafür benötigte Schlüssel wird aus Benutzereingaben generiert (zB Passwort, Touchmuster)

8.5.8.1 Abgedeckte Gefahren

Da durch die Verschlüsselung nur mehr die eigene APP die Daten entschlüsseln kann, wird sowohl der Schadsoftware (vgl Seite 31, 6.2) als auch dem physikalischen Zugriff (vgl Seite 33, 6.4) entgegengewirkt. Auch die Bedrohung durch reverse Engineering (vgl Seite 32, 6.3) wird abgeschwächt.

8.5.8.2 Zutreffende Gefahren

Kryptografie erhöht keine Gefahren, jedoch wird die Laufzeit und Zugriffszeit auf die Daten erhöht, was zu einer verringerten Benutzerzufriedenheit führen kann.

8.5.8.3 Beteiligte Instanzen

Die Sicherheit der Daten hängt sehr stark vom verwendeten Verschlüsselungsalgorithmus ab. Da die Speicherkarte von einem Computer ausgelesen werden kann, besteht die Möglichkeit, auch mit

leistungsstarken Geräten einen Angriff auf die Verschlüsselung durchzuführen. Werden die Daten auf der Speicherkarte gesichert, ist es schwieriger, einen derartigen Angriff zu starten.

8.5.8.4 Testprogramm

Zur vereinfachten Übergabe des Klartextes und des Chiffrates zwischen den Funktionen wird die Hilfsklasse „CipheredText“ verwendet. Diese besitzt drei Variablen:

- `int length`: Speichert die Länge des Textes
- `byte[] cipher`: Speichert entweder den Text oder das Chiffrat
- `byte[] iVector`: Speichert den Initialisierungsvektor eines Chiffrats

Die Methoden haben die Hilfsklasse als Parameter und Rückgabewert. Zusätzlich wird der Schlüssel zur Verschlüsselung mitgegeben. Dieser muss als AES Key aus einem `byte[]` erzeugt werden (`new SecretKeySpec(<byte[]>, "AES");`)

- Symmetrische Verschlüsselung mit AES

```
private CipheredText encrypt(CipheredText value, SecretKeySpec secKey)
     throws Exception {

     byte[] plainText = value.getcipher();
     int ptLength = value.getlength();

     Cipher cipher = Cipher.getInstance("AES/CFB8/PKCS5PADDING");
     cipher.init(Cipher.ENCRYPT_MODE, secKey);
     byte[] cipherText = new byte[cipher.getOutputSize(ptLength)];
     int ctLength = cipher.update(plainText, 0, ptLength, cipherText, 0);
     ctLength += cipher.doFinal(cipherText, ctLength);
     return new CipheredText(ctLength, cipherText, cipher.getIV());
}
```

Code. 9. symmetrische Verschlüsselung (AES)

Beim Aufruf der Methode werden zuerst die Daten aus der Hilfsklasse ausgelesen und ein Cipher initialisiert. Durch die Methode `cipher.update` wird der Verschlüsselungsvorgang durchgeführt. Am Ende wird das Chiffrat mit den zusätzlichen Daten in einen CipheredText gepackt und zurückgegeben.

- Symmetrische Entschlüsselung mit AES

```
private CipheredText decrypt(CipheredText value, SecretKeySpec secKey)
     throws Exception {

     int ctLength = value.getlength();
     byte[] cipherText = value.getcipher();
     byte[] iVector = value.getIV();

     Cipher cipher = Cipher.getInstance("AES/CFB8/PKCS5PADDING");

     cipher.init(Cipher.DECRYPT_MODE, secKey, new
     IvParameterSpec(iVector));
     byte[] plainText = new byte[cipher.getOutputSize(ctLength)];
     int ptLength = cipher.update(cipherText, 0, ctLength, plainText, 0);
```

```
        ptLength += cipher.doFinal(plainText, ptLength);
        return new CipheredText(ptLength, plainText, null);
    }
```

Code. 10. symmetrische Entschlüsselung (AES)

Die Entschlüsselung funktioniert ähnlich wie die Verschlüsselung. Der Cipher muss jedoch im Modus „`Cipher.DECRYPT_MODE`“ initialisiert werden. Bei der Entschlüsselung muss der Initialisierungsvektor angegeben werden.

8.5.9 (GPS-) Positionserkennung

Die (GPS-) Positionserkennung ist nicht direkt als Sicherheitsmechanismus zu sehen, da sie von bestimmten Funktionen der APP benötigt wird. Im Bereich der Sicherheit kann sie benutzt werden, um die Verwendung der APP auf einen bestimmten geografischen Bereich einzuschränken. Damit soll verhindert werden, dass bei Verlust des Gerätes oder der Zugangsdaten der Angreifer Zugriff auf die Kontoinformationen erhält. Voraussetzung dafür ist jedoch, dass der Angriff von einem Gebiet aus durchgeführt wird, das als nicht akzeptabel eingestuft ist.

8.5.9.1 Abgedeckte Gefahren

Durch die geografische Beschränkung kann einerseits die Bedrohung durch Social Engineering (vgl Seite 30, 6.1) und andererseits die Gefährdung eines physikalischen Zugriffs (vgl Seite 33, 6.4) vermindert werden.

Zu bedenken ist jedoch, dass die Positionserkennung nur dann die Gefahren abdecken kann, wenn der Angriff aus einem anderen Gebiet kommt als das, in dem das Gerät normalerweise verwendet wird, und es sich somit außerhalb der definierten Gültigkeitsbereiche befindet. In den meisten Fällen wird nach einem Diebstahl des Gerätes sofort auf die Daten und Services zugegriffen. Der Angreifer befindet sich dann wahrscheinlich noch innerhalb des definierten Gebietes.

8.5.9.2 Zutreffende Gefahren

Es ist wichtig, dass die Positionsdaten bei der Übertragung nicht verändert werden, da sie sonst nicht mehr vertrauenswürdig sind (vgl Seite 38, 6.11). Damit der richtige Standort ausgelesen werden kann, ist es wichtig, dass die Daten der Positionserkennung nicht gefälscht sind und dass die APP nicht verändert wurde (vgl Seite 32, 6.3). Durch eine Veränderung der GPS Hardware (vgl Seite 34, 6.6) ist die Rückmeldung dieser nicht mehr vertrauenswürdig.

Kleine Gültigkeitsbereiche schränken den eigentlichen Benutzer ein. Dies kann die Benutzerzufriedenheit reduzieren und Inakzeptanz auslösen. Weiters ist zu beachten, dass die häufige Ver-

wendung des GPS Moduls die Akkulaufzeit reduziert. Deshalb wird dieses Modul oft vom Benutzer deaktiviert. Ein Aktivieren kann vom User als störend empfunden werden.

Alternativ könnte die Position des Gerätes direkt über den Provider ermittelt werden. Dazu ist jedoch eine separate Anfrage des Rechenzentrums an diesen notwendig. Derartige Anfragen dürfen aber aus Datenschutzgründen nicht ohne Zustimmung des Benutzers beantwortet werden.

Aus diesen Gründen sollte die Positionserkennung als Sicherheitsmerkmal deaktivierbar sein.

8.5.9.3 Beteiligte Instanzen

Das mobile Gerät liest mit Hilfe der Positionserkennung (zB GPS) den aktuellen Standort aus und sendet diesen über das Netzwerk an den Server. Damit dieses Sicherheitsmerkmal zuverlässig funktionieren kann, ist es wichtig, dass korrekte Daten ausgelesen und diese auch unverändert übertragen werden.

8.5.9.4 Testprogramm

Für die Erkennung der Position muss ein LocationManager und ein LocationListener angelegt werden. Es wird die Berechtigung ACCESS_FINE_LOCATION benötigt, um auf genaue Positionsdaten zugreifen zu können.

```
LocationManager locMan = (LocationManager)
     this.getSystemService(Context.LOCATION_SERVICE);

     LocationListener locList = new LocationListener() {
       public void onStatusChanged(String provider, int status, Bundle
     extras) {}
       public void onProviderEnabled(String provider) {}
       public void onProviderDisabled(String provider) {}
       public void onLocationChanged(Location location) {
             UseNewLocation(location);
       }
     };

     String locProv = LocationManager.NETWORK_PROVIDER; //bzw GPS_PROVIDER;
     Location loc = locMan.getLastKnownLocation(locProv);
     UseNewLocation(loc);
     locMan.requestLocationUpdates(locProv, 0, 0, locList);
```

Code. 11. Positonsabfrage

Mit der Methode `getLastKnownLocation(<LocationProvider>)` kann eine schon ermittelte Position aus dem Speicher abgerufen werden. Diese ist unter Umständen jedoch veraltet.

8.5.10 Auftragsautorisierung mittels cardTAN

Ein großes Sicherheitsproblem bei e-Banking am Smartphone stellt die Verwendung der mTAN dar. (vgl Seite 25, 5.3.3)

Um dies zu vermeiden, ist es notwendig, dass die TAN auf einem anderen Gerät, als jenem, auf dem e-Banking betrieben wird, generiert bzw empfangen wird. Dazu kann das neue cardTAN, chipTAN oder smartTAN-plus Verfahren genutzt werden. Bei den letzten beiden Verfahren wird noch in ein manuelles und ein optisches Verfahren unterschieden. Beim manuellen müssen die Transaktionsdaten manuell in den TAN Generator eingegeben werden. Das optische Verfahren funktioniert wie die cardTAN. Dabei werden die Transaktionsinformationen mit Hilfe eines Flickerbildes auf den TAN-Generator übertragen. Dort wird mit der Bankomatkarte und der PIN des e-Bankingzuganges die TAN generiert, die der Benutzer nun manuell zur Bestätigung der Durchführung der Transaktion am e-Banking Portal bzw in der APP eingeben muss. (vgl Seite 13, 3.2.5.3)

8.5.10.1 Abgedeckte Gefahren

Durch die Verwendung der cardTAN anstatt der mTAN kann Social Engineering (vgl Seite 30, 6.1), Malware und Viren (vgl Seite 31, 6.2), reverse Engineering (vgl Seite 32, 6.3) und Shoulder Surfing (vgl Seite 33, 6.5), aber auch dem physischen Zugriff (vgl Seite 33, 6.4) und der Hardwaremanipulation (vgl Seite 34, 6.6) zumindest teilweise vorgebeugt werden.

8.5.10.2 Zutreffende Gefahren

Die Verwendung der cardTAN führt zu keiner zusätzlichen Erhöhung der Gefährdungspotentiale. Jedoch muss beachtet werden, dass, wenn der Angreifer Zugriff auf die Bankomatkarte hat und die PIN kennt, er unbemerkt eigene TANs generieren und somit eigene Transaktionen bestätigen kann. Dies kann teilweise sehr schnell passieren, wenn zB die Handtasche oder der Rucksack gestohlen wird.

8.5.10.3 Beteiligte Instanzen

Bei der cardTAN wird mit dem TAN-Generator das Flickerbild eingelesen und auf dem Generator die eingelesen Daten mittels e-Banking PIN und Bankomatkarte bestätigt. Dieser erzeugt dann eine TAN, die wiederum in der APP eingegeben werden muss, um die Überweisung zu bestätigen. Danach wird die TAN über das Netzwerk zum Server gesendet, der sie auf Gültigkeit überprüft.

8.6 Sicherheit im Rechenzentrum

Ein weiterer wichtiger Punkt in der Absicherung von e-Banking ist der Schutz des Rechenzentrums und der Server. Hier werden alle Daten gespeichert und aufbewahrt. Es stellt einen zentralen Punkt dar, der bei einem Ausfall das gesamte System lahmlegen kann.

8.6.1 Allgemeine Sicherheitsvorkehrungen

Um die Sicherheit in einem Rechenzentrum zu gewährleisten, gibt es bestimmte Grundregeln, die eingehalten werden sollen. Abhängig von den angebotenen Services und verwendeten Systemen ist die Wichtigkeit und Effektivität unterschiedliche einzustufen.

Durch SSL/TLS wird ein verschlüsselter Tunnel ins Rechenzentrum aufgebaut. Dieser Tunnel kann auch von anderen Anwendungen genutzt werden, um an der Firewall vorbei mit einem Server zu kommunizieren. Die Firewall kann den Inhalt des Traffics meist nicht überprüfen, da der Tunnel kryptografisch gesichert ist. Aus diesem Grund ist es wichtig, dass durch den Tunnel Zugriff auf möglichst wenig Ressourcen und Daten gewährt wird. Durch geeignete Maßnahmen kann der Tunnel aufgebrochen werden, um den Sicherheitssystemen wieder Zugriff auf die Daten zu verschaffen. Das ist jedoch mit anderen Problemen verbunden. (vgl Seite 106, 8.6.5)

In einem verteilten System (vgl. Abb. 26 Aufbau verteiltes System) existieren mehrere Server und Geräte, die für unterschiedliche Aufgaben zuständig sind. Beim Ausfall eines der Geräte ist nicht das gesamte System unerreichbar. Für einen Benutzer sieht es aus, als würde seine Anfrage von einem einzigen Server beantwortet werden, jedoch kommuniziert er nur mit einem Grenzsystem (zB Webserver in der DeMilitarized Zone (DMZ)). Dieses wiederum verwendet eigene Services, um die Anfragen an dahinterliegende Server weiterzuleiten oder um die Daten zur Beantwortung der Anfrage des Endbenutzers zu sammeln.

Wird ein verteiltes System eingesetzt, kann ein Angreifer nur auf den Server oder das Gerät zugreifen, mit dem der Tunnel aufgebaut wurde (Grenzsystem). Alle anderen Systeme (zB Datenbanken) sind von außen nicht erreichbar und können nur von Anwendungen des Grenzsystems angesprochen werden.

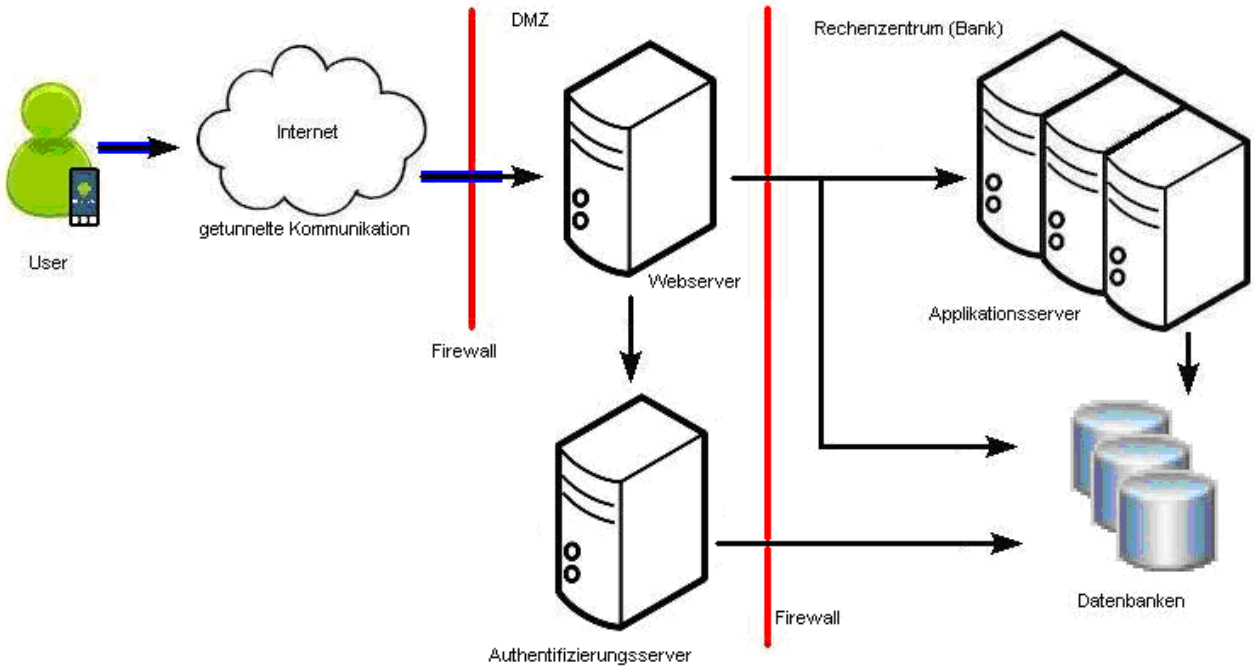


Abb. 26. Aufbau verteiltes System

Je mehr von der Infrastruktur unbekannt ist, umso schwieriger wird es für einen Angreifer, eine passende Schwachstelle zu finden und diese auszunützen. Somit ist zu beachten, dass Daten bezüglich Infrastruktur und Sicherheitsmechanismen zumindest als wichtig einzustufen sind. Passwörter von Netzwerkgeräten, Servern und Services müssen geheim gehalten werden und dürfen nicht weitergegeben werden.

Jegliche Absicherungen gegen Angriffe von außen sind wirkungslos, wenn der Angreifer bereits im Rechenzentrum sitzt. Die physikalische Abschottung ist wichtig, um Personen ohne Zugriff von den Systemen wie Servern und Routern fernzuhalten. Dazu sind ein Zutrittssystem bei den Türen zu den Geräten und möglichst wenig alternative Zustiegsmöglichkeiten (wie Fenster) notwendig. Es sollen nur diese Personen Zutritt zum Serverraum haben, die diesen betreuen. Eine Aufteilung in mehrere Räume ermöglicht die Granularisierung der Zutrittsberechtigungen, erhöht jedoch den Aufwand für die Verwaltung.

Server müssen gegen Ausfall geschützt werden, damit sie ausfallsfrei ihre Services anbieten können. Dafür sind eine alternative Stromversorgung und eine gute Klimatisierung der Räumlichkeiten notwendig. Auch die Sicherung dieser Versorgungseinrichtungen ist von hoher Wichtigkeit, da beim Ausfall dieser das Rechenzentrum direkt betroffen ist. (Überhitzung, Stromausfall)

Um für Notfälle vorbereitet zu sein, sollten auch Gefährdungen wie Feuer und Wasserschäden beachtet werden. Um Datenverlust und langen Ausfallszeiten vorzubeugen, ist sowohl ein Backup

von Servern, Netzwerkgeräten und Services notwendig, als auch eine Dokumentation der Konfiguration, sollten Services neu eingerichtet werden müssen.

8.6.1.1 Abgedeckte Gefahren

Durch die Einhaltung der beschriebenen Sicherheitsrichtlinie kann die Gefahr durch physischen Zugriff (vgl Seite 43, 6.13), einer DoS Attacke (vgl Seite 42, 6.12) oder von Social Engineering (vgl Seite 44, 6.15) stark reduziert werden. Zusätzlich erfolgt eine Abschwächung der Bedrohung durch den Angriff aus dem verschlüsselten Kommunikationstunnel (vgl Seite 43, 6.14) und durch das Ausnutzen von Software- und Kommunikationsfehlern (vgl Seite 45, 6.16).

Weiters kann die Ausfallssicherheit und Zuverlässigkeit gesteigert werden.

8.6.1.2 Zutreffende Gefahren

DoS Angriffe können nur bis zu einer gewissen Grenze abgefangen werden. Erfolgt der Angriff von zu vielen Geräten (DDoS), kann trotz Lastverteilung das System lahmgelegt werden (vgl Seite 42, 6.12). Einige Teile des Sicherheitskonzeptes basieren auf der Geheimhaltung von Informationen oder dem Vertrauen zu den Mitarbeitern. Erhält ein Angreifer diese Daten über das System oder dessen Aufbau, kann er Schwachstellen leichter finden und ausnützen. (vgl Seite 44, 6.15).

Durch die serverseitigen Anwendungs- oder Konfigurationsfehler kann es sein, dass bei einer Übernahme des Grenzsystems ein Angreifer leichten Zugriff auf die dahinterliegenden Systeme erhält, da diese zueinander in einem Vertrauensverhältnis stehen (vgl Seite 45, 6.16). Damit das System zusätzliche Sicherheit bieten kann, ist es wichtig, dass einer Anfrage aus dem Grenzsystem nicht automatisch vertraut wird. Eine Firewall zwischen der DMZ und den restlichen Systemen kann den Traffic mitlesen und somit gefährliche Anfragen herausfiltern.

8.6.1.3 Beteiligte Instanzen

Von den allgemeinen Sicherheitsvorkehrungen ist das gesamte Rechenzentrum betroffen. Dazu zählen vor allem interne und externe Firewalls, Gateways und Proxies sowie die Server und Datenbanken. Neben den Computerelementen sind noch die Räumlichkeiten des Rechenzentrums und die Mitarbeiter von der Umsetzung und Einhaltung betroffen.

8.6.2 Konsistenzüberprüfung der Anfrage

Die Überprüfung der Anfrage ermöglicht eine frühzeitige Erkennung, ob die Kommunikation von einem Endgerät den normalen Verwendungszweck darstellt, oder ob sie ein Angriff ist. Es muss gewährleistet werden, dass nur gültige Anfragen ins System kommen und verarbeitet werden.

8.6.2.1 Abgedeckte Gefahren

Durch die Anfragenüberprüfung können gefälschte Clients entdeckt werden, wenn sich diese unerwartet verhalten (vgl Seite 32, 6.3). Die Kontrolle erlaubt nur bekannte Befehle. Somit kann Angriffen entgegengewirkt werden, die aus dem Tunnel (vgl Seite 43, 6.14) kommen oder auf die Software und Konfiguration mit dem Ziel, dessen Fehler auszunutzen, wirken. (vgl Seite 45, 6.16)

In bestimmten Fällen können auch DoS Attacken (vgl Seite 42, 6.12) abgewehrt werden, da die Anfrage bereits beim Check abgewiesen wird.

8.6.2.2 Zutreffende Gefahren

Die Verwendung einer Überprüfung ist ein weiterer Zwischenschritt in der Kommunikation. Sie benötigt eine gewisse Rechenzeit und verzögert somit die Durchlaufzeit. Die zusätzlich benötigte Rechenleistung kann bei sehr vielen Anfragen den Server überlasten und somit eine DoS Attacke ermöglichen (vgl Seite 42, 6.12). Ein weiteres Problem ist die Erweiterbarkeit und Flexibilität des Systems. Je genauer eine Anfrage geprüft werden muss, umso mehr Aufwand ist es, neue Anfragetypen zu erstellen. Weiters wird auch durch eine genauere Prüfung noch länger für den Kontrollvorgang benötigt. Fehler in der Anwendung und der Definition können dazu führen, dass korrekte Anfragen abgewiesen oder fehlerhafte angenommen werden.

8.6.2.3 Beteiligte Instanzen

Die Anfragenüberprüfung sollte möglichst früh durchgeführt werden. Dies kann entweder direkt beim Proxy, der Firewall oder dem Gateway sein. Eine genaue Kontrolle ist meist jedoch erst am Server möglich. Für die Durchführung ist eine Überprüfungsvorschrift notwendig.

8.6.3 Authentifikation und Autorisierung des Benutzers bzw des Gerätes

Eine Autorisierung kann nur nach einer erfolgreichen Authentifizierung durchgeführt werden. Erst wenn der Benutzer bzw das Gerät bekannt ist, kann festgelegt werden, worauf zugegriffen werden darf. Die Authentifikationsmöglichkeiten unter Android werden in Kapitel „8.1 Authentifikation des Benutzers“ und „8.2 Authentifikation des Gerätes“ behandelt.

Durch eine erfolgreiche Authentifizierung kann einem Benutzer Zugriff auf das System gewährt werden. Es ist jedoch wichtig, dass er nur darauf Zugriff erhält, wofür er auch berechtigt ist. Für e-Banking bedeutet das, dass ein User nur auf die ihm zugeordneten Konten zugreifen darf.

8.6.3.1 Abgedeckte Gefahren

Durch die Vergabe von Berechtigungen können Angriffe auf die Ressourcen von innen (vgl Seite 43, 6.13) und von außen (vgl Seite 43, 6.14) abgewehrt werden.

Fehler in den Anwendungen können durch ein gutes Berechtigungssystem abgefangen werden. Schafft es ein Angreifer, aus der Anwendung auszubrechen, kann er nicht auf das System zugreifen, da ihm die Berechtigungen fehlen. Bei einem verteilten System (vgl Seite 100, 8.6) können mit dieser Methode auch die Systeme hinter der DMZ geschützt werden.

8.6.3.2 Zutreffende Gefahren

Das System muss vor allem gegen den physikalischen Zugriff abgesichert werden. Weiters kann ein Angreifer durch Insiderinformationen leichter Schwachstellen im System finden. Deshalb soll der Verlust an Informationen möglichst vermieden werden (vgl Seite 44, 6.15).

8.6.3.3 Beteiligte Instanzen

Für die Durchführung der Authentifizierung und Autorisierung ist es notwendig, dass der Benutzer seine Authentifizierungsinformation zur Verfügung stellt. Diese werden über die Kommunikationskanäle zum Authentifikationsserver übertragen, der die Daten mit einer Datenbank, die die Berechtigungs- und Autorisierungsinformationen enthält, abgleicht und so den Benutzer dementsprechend autorisiert.

Es ist wichtig, dass die Anmelde- bzw Authentifikationsinformationen geheim und integer übertragen werden.

8.6.4 Lastverteilung (Loadbalancing) und redundante Systeme

Lastverteilung ermöglicht, dass mehrere Server die Anfragen gemeinsam abarbeiten. Damit wird Redundanz erzeugt und somit kann gewährleistet werden, dass auch beim Ausfall eines Servers das Service weiterhin verfügbar bleibt. Für die Implementierung der Lastverteilung gibt es mehrere unterschiedliche Ansätze. Es gibt softwaretechnische Lösungen, bei denen die Server miteinander kommunizieren müssen. Alternativ dazu stehen die Hardware-Loadbalancer, die durch kontinuierliche Abfragen und Logs die Belastung der einzelnen Server kennen und somit die Anfragen dementsprechend weiterleiten.

8.6.4.1 Abgedeckte Gefahren

Aufgrund der Redundanz und der zusätzlichen Rechenleistung mehrerer Server ist es schwieriger, das System durch eine DoS Attacke (vgl Seite 42, 6.12) lahm zu legen. Die Schwachstelle verlagert sich jedoch bei schlechter Implementierung auf die Lasterverteiler. Deshalb sollten auch diese möglichst redundant ausgelegt sein.

8.6.4.2 Zutreffende Gefahren

Die Lastverteilung benötigt zusätzliche Hard- oder Software. Somit besteht die Gefahr von Fehlern und Bugs sowohl in diesen als auch in der Konfiguration dieser. (vgl Seite 45, 6.16) Weiters erhöht sich der administrative Aufwand für die Infrastruktur stark, da mehr Geräte gewartet und verwaltet werden müssen. Die Lastverteilung ist auch beim Design einer Anwendung zu berücksichtigen, da nicht davon ausgegangen werden kann, dass sich ein bestimmter Client immer zum selben Server verbindet.

8.6.4.3 Beteiligte Instanzen

Die Anfrage kommt vom Gateway oder der Firewall direkt zum Lastverteiler. Dieser leitet sie zu dem aus seiner Ansicht nach am wenigsten belasteten Server weiter. Welcher sie wiederum abarbeitet. Dafür sind Datenbanken notwendig, welche unter Umständen nicht redundant ausgeführt sind, oder selber wieder durch Lastverteiler verwaltet werden.

Weiters ist zu beachten, dass bei bestimmten Konfigurationen das Serverzertifikat vom Lastverteiler bereitgestellt werden muss, da der Client die SSL/TLS Verbindung mit diesem und nicht mit dem Server aufbaut.

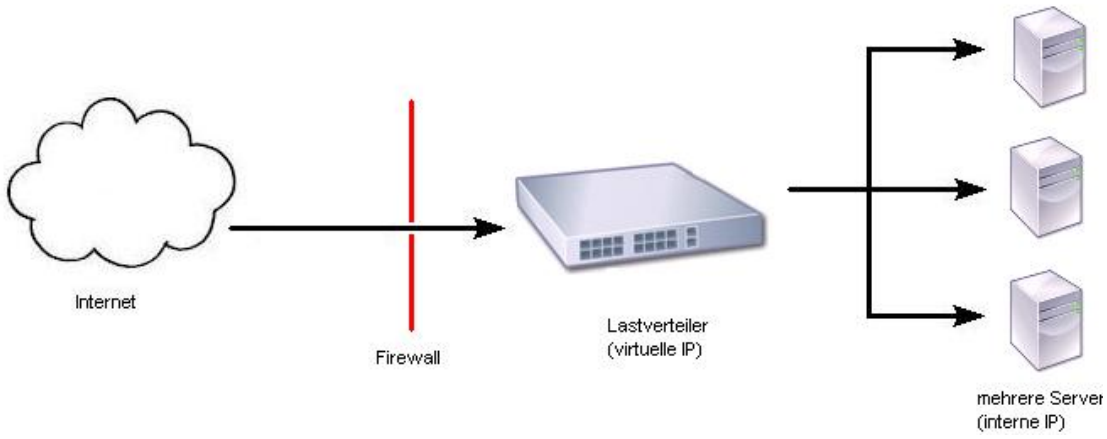


Abb. 27. Möglicher Systemaufbau mit lokaler Lastverteilung

8.6.5 SSL Offloading

Die Verschlüsselung durch SSL oder TLS hat mehrere Nachteile. Einerseits benötigt es zusätzliche Rechenleistung beim Server und beim Client, andererseits können die Firewalls den Traffic nicht mehr mitlesen. SSL Offloader können den verschlüsselten Tunnel vorzeitig beenden und somit ermöglichen, dass die Firewall und das Intrusion Detection System (IDS) oder Intrusion Prevention System (IPS) die Datenpakete mitlesen können.

SSL Offloading wird von den Sicherheitssystemen durchgeführt. Dabei gibt es zwei Möglichkeiten. Es kann entweder der Traffic nach der Überprüfung erneut verschlüsselt und als neue SSL Kommunikation an den Server gesendet oder unverschlüsselt weiterübertragen werden. [34]

8.6.5.1 Abgedeckte Gefahren

Ein SSL Offloader ermöglicht, dass die Datenpakete von Firewalls und anderen Sicherheitsvorkehrungen überprüft werden. Außerdem kann es die Ent- und Verschlüsselung der Pakete für den Server übernehmen.

Durch den Offloader endet der Kommunikationstunnel vor dem Server im Rechenzentrum. Damit erhalten Angreifer keinen direkten Zugriff auf diesen durch den Tunnel (vgl Seite 43, 6.14). Die Last einer DoS Attacke (vgl Seite 42, 6.12) verteilt sich dann auf den Offloader (Ver- und Entschlüsselung), eventuell die Firewall (Überprüfung aus Konsistenz und Richtigkeit; vgl Seite 45, 6.16) und den Server (Anfragenabarbeitung).

8.6.5.2 Zutreffende Gefahren

Sendet der Offloader die Anfragen unverschlüsselt zum Server weiter, kann ein Angreifer, der sich innerhalb des Rechzentrums befindet, die Kommunikation im Klartext mitlesen (vgl Seite 43, 6.13).

8.6.5.3 Beteiligte Instanzen

Durch den SSL Offloader kommuniziert der Client nicht mehr direkt mit dem Server. Er sendet die Pakete durch den SSL Tunnel bis zum SSL Offloader. Dort werden sie entschlüsselt, geprüft (Virusscan, …) und an den Server weitergeleitet. Das Zertifikat, das dem Client vorgelegt wird, um die Identität des Servers zu bestätigen, muss somit von dem Offloader bereitgestellt werden.

9. Entwurf für ein Singlebank e-Banking System

In den folgenden Kapiteln wird beschrieben, wie ein e-Banking System aussehen kann, das sich nur mit einer Bank verbindet. Die Auswahl der verwendeten Funktionen basiert auf Rückmeldungen von den Benutzern zu den bestehenden APPs. Ziel ist es, ein System vorzustellen, bei dem eine vergleichbare Sicherheit geboten wird, wie bei bestehenden Webportalen für den Computer. Die Zielgruppe für die APP sind hauptsächlich Privatkunden von Banken, die von einem Smartphone aus die eigenen Konten verwalten wollen. Pro Anmeldung kann nur mit einem Verfüger auf ein Konto zugegriffen werden.

Das Ziel der Implementierung des Entwurfes (vgl. Seite 134, 9.8) ist das Testen des Zusammenspiels und der Funktionsweise der Sicherheitsmechanismen, sowie der Benutzerfreundlichkeit in Bezug auf den Workflow.

9.1 E-Banking Funktionalitäten

Folgende Funktionen sollten von einer Singlebank e-Banking APP mindestens zur Verfügung gestellt werden. Sie stellen die grundlegenden Funktionen von e-Banking dar. Die Daten werden nicht auf dem Endgerät zwischengespeichert. Sie werden für jede Ausgabe separat vom Server abgerufen. Somit wird zwar mehr Transfervolumen benötigt, jedoch erhält ein Angreifer auf diese Weise nie Zugriff auf Kontoinformationen wie den Kontostand oder die Kontodaten der verfügbaren Konten eines Verfügers.

9.1.1 Bankomat-, Filialfinder

Durch GPS und das GSM Modul ist es möglich, eine genaue bzw ungefähre Position des mobilen Gerätes zu ermitteln. Diese Funktion kann genutzt werden, um dem Benutzer einen Bankomat- und Filialfinder anzubieten. Diese Funktion steigert zwar nicht die Sicherheit oder hilft bei der Verwaltung der Finanzmittel auf den Konten und Depots, dafür erhöht sie die Benutzerakzeptanz der APP. Sie sollte auch ohne Login zum e-Banking System möglich sein.

9.1.2 Kartensperre

Der Verlust der Bankomat- oder Kreditkarte kann schwerwiegende Folgen haben. Deshalb ist es wichtig, dass diese nach dem Bemerken des Verlustes so schnell wie möglich gesperrt wird. Dafür sollte eine Kartensperrfunktionalität angeboten werden. Für diese Funktion ist ein erfolgreicher Login erforderlich, da sonst die Möglichkeit der Sperrung auch von einem Angreifer genutzt werden kann.

Die entsprechende Activity kann entweder eine Liste von verfügbaren Karten, die zur Sperrung möglich sind, anzeigen und diese nach der Auswahl automatisch sperren oder sie zeigt die notwendigen Telefonnummern und Daten an, die für die Kartensperre notwendig sind.

9.1.3 Login

Der Login wird mit den Zugangsdaten der Bank durchgeführt. Zu den Zugangsdaten zählen: Bankleitzahl, Kontonummer, Verfügernummer und PIN. Die PIN sollte in einem eigenen Fenster abgefragt werden. In diesem werden zusätzlich noch die vorhergegangenen Daten angezeigt, damit diese vom Benutzer überprüft werden können.

Die Daten Bankleitzahl, Kontonummer und Verfügernummer können im Telefonspeicher abgelegt werden. Diese Funktion sollte deaktivierbar sein. Sind die Daten im Telefonspeicher abgelegt, kann die APP mit Hilfe eines Touchmusters (vgl Seite 71, 8.1.3) entsperrt werden. Sind keine Daten im

Telefonspeicher gespeichert, startet die APP mit einem Eingabefomular für die Kontozugangsdaten. Eine eventuelle Speicherung kann der Benutzer durch Auswahl der Checkbox bestätigen.

9.1.4 Banknachrichten

Der gesicherte Kommunikationskanal zum Erhalten und Senden von Mitteilungen, die Konto- und Kontoführungsinformationen beinhalten, sollte als eigene Funktion bereitgestellt werden. Damit kann der Kunde über Neuerungen und Angebote der Bank informiert werden. Außerdem ist eine Kommunikation mit dem Bankberater möglich, ohne dass dabei sensible Daten über unsichere Kanäle wie e-Mail fließen. (vgl Seite 9, 3.2.2)

9.1.5 Kontofunktionen

Die Kontofunktionen dienen der Verwaltung der einzelnen Konten. Pro Verfüger und Kontonummer können mehrere Konten eingebunden sein. Diese können auch von unterschiedlichem Typ sein. (zB Girokonto, Sparbuch, Bausparer)

9.1.5.1 Kontoübersicht

In der Kontoübersicht werden alle Konten angezeigt und deren Kontostand ausgegeben. Es sollte möglich sein, ein Konto auszuwählen und die Daten wie Zinsen und Inhaber anzuzeigen. Die Zusatzdaten sollten jedoch in einem eigenen Fenster aufgelistet werden, um die Übersicht gewährleisten zu können.

9.1.5.2 SEPA Überweisung

Die SEPA Überweisung ermöglicht das Überweisen von Geld innerhalb der EU. Aus Sicherheitsgründen sollten nur SEPA Überweisungen und keine Nicht-EU-Auslandsüberweisungen angeboten werden. Nach dem Ausfüllen der Überweisungsdaten werden diese noch einmal in einer neuen Activity zur Bestätigung angezeigt. Erst nach dem erfolgreichen Eingeben einer gültigen TAN wird der Auftrag abgesendet. Als TAN Verfahren sollte nur die cardTAN möglich sein. (vgl Seite 117, 9.2.8) Das gesammelte Absenden von Aufträgen sollte aus Gründen der Benutzerfreundlichkeit ebenfalls implementiert werden, jedoch sollte nach dem Erstellen einer Überweisung standardmäßig das Formular zum Absenden angezeigt werden. Wird die Eingabe der TAN abgebrochen, können die vollständig erfassten Aufträge im Anschluss über einen Unterpunkt im Menü abgeschickt werden.

9.1.5.3 Daueraufträge

Die Einbindung von Daueraufträgen stellt zwar ein gewisses Gefährdungspotential dar, da damit kontinuierlich Geld von einem Konto auf ein anderes gebucht werden kann. Von Konten, die nur

selten vom Benutzer kontrolliert werden, kann auf diese Weise von einem Angreifer automatisiert viel Geld abgebucht werden.

Diese Funktion wird jedoch in den Foren der bestehenden APPs von den Benutzern gefordert. Um das Sicherheitsrisiko umgehen zu können, sollte es möglich sein, diese Funktion zu deaktivieren. (vgl. Seite 121, 9.2.14)

9.1.6 Wertpapierfunktionen

Beim Handel mit Wertpapieren ist vor allem der richtige Zeitpunkt von Kauf und Verkauf wichtig. Deshalb ist es auch essentiell, dass entsprechende Funktionen von der e-Banking APP angeboten werden. Da viele Kunden keinen Handel mit Wertpapier betreiben, zählen diese Funktionen aber nicht zu den Kernfunktionen der APP.

9.1.6.1 Depotübersicht, Orderbuch

Die Depotübersicht dient als Sammlung aller derzeit im Besitz des Benützers befindlichen Wertpapiere. Sie zeigt neben der Anzahl der Papiere auch deren ursprünglichen Einkaufswert und den aktuellen Kurs an. Die Veränderung sollte ebenfalls angezeigt werden. Eine farbige Hinterlegung soll auf einen möglichen Gewinn oder Verlust beim Verkauf zum aktuellen Kurs hinweisen.

Das Orderbuch stellt eine Historie von abgesetzten Transaktionen dar. Damit kann verfolgt werden, welche Aktionen in den letzten Tagen und Wochen durchgeführt wurden. Um die Übersichtlichkeit zu wahren sollte nie mehr als ein Monat angezeigt werden.

9.1.6.2 Kursübersicht

Die Kursübersicht kann für jedes Wertpapier direkt ausgewählt werden. Sie zeigt in einem Diagramm den Kursverlauf an. Ein grafischer Vergleich unterschiedlicher Wertpapiere ist denkbar, jedoch sollte die Maximalanzahl der angezeigten Kurse begrenzt sein.

9.1.6.3 Kauf und Verkauf

Die Funktion zum Kauf und Verkauf von Wertpapieren dient nicht wie die anderen Funktionen zur Information sondern rein dem Handel. Sie enthält ein Feld für die Auswahl des Wertpapiers, welches schon in einer vorhergegangen Anzeige ausgewählt wurde, und weitere Felder für die Anzahl, das Verrechnungskonto und beim Verkauf den zu erzielenden Preis.

Die Autorisierung zur Durchführung der Transaktion wird mit einer TAN durchgeführt. Wie bei der Überweisung sollte auf dem Smartphone keine mTAN möglich sein. Als TAN Verfahren sollte nur die cardTAN verfügbar sein. (vgl Seite 117, 9.2.8)

Auch die Möglichkeit der Sperre der Wertpapierhandelsfunktionen ist anzubieten. Es kann sein, dass ein Benutzer zwar eine Übersicht über die Wertpapiere am mobilen Gerät verwenden möchte, jedoch aus Sicherheitsgründen auf den Handel mit diesen verzichten will.

9.2 Verwendete Sicherheitsmerkmale

Dieses Kapitel beschäftigt sich mit den Sicherheitsmerkmalen, die verwendet werden, um die oben erwähnten Funktionalitäten abzusichern. Genaue Beschreibungen zu den Funktionen und Hinweise bezüglich der abgedeckten Gefährdungen sind in dem jeweiligen Unterkapitel des Kapitels „8 Sicherheitsmechanismen für e-Banking“ zu finden.

9.2.1 Mustereingabe mittels Touchmuster

Das Touchmuster unterstützt die Eingabe der PIN und TAN. (vgl Seite 71, 8.1.3) Damit kann theoretisch auch ein Passwort erstellt werden, das nicht auf Buchstaben und Zahlen basiert sondern auf Icons, die auf den einzelnen Buttons abgebildet werden.

9.2.1.1 Vorteile

- Übersichtliche Eingabe

Bei der Verwendung des Touchmusters muss das Passwort nicht mehr über die Bildschirmtastatur eingegeben werden. Die Tastatur ist vor allem bei kleineren Geräten sehr unübersichtlich und mit den Fingern ist eine fehlerfreie Eingabe nur schwer möglich.

- Keine Gebrauchsspuren

Im Gegensatz zum Wischmuster wird der Identifizierungscode nicht über den Bildschirm gewischt sondern getippt. Beim Wischen kann es zu Kratzern und Spuren am Monitor kommen, die von einem Angreifer ausgelesen werden können. (vgl Seite 25, 5.3.2) Das Tippen auf Buttons erzeugt wenig Spuren und kann somit nicht so einfach gelesen werden.

- Leichter zu merken

Für einen Benutzer ist ein Tippmuster auch als Muster merkbar. Damit ist es nicht mehr notwendig, sich eine Zeichenkombination zu merken. Lediglich die Reihenfolge der Buttons ist ausschlaggebend.

- Eingabegeschwindigkeit

Da sich die Eingabe auf weniger Buttons beschränkt als bei einer Bildschirmtastatur und die Eingabe als Muster merkbar ist, kann von einer höheren Eingabegeschwindigkeit ausgegangen werden. Ein Mitlesen der Eingabe wird für eine nebenstehende Person erschwert.

9.2.1.2 Nachteile

- Einfache Muster

Ein Problem stellen einfache Muster dar. Je einfacher das Muster ist, umso leichter kann es von anderen mitgelesen werden. (vgl Seite 71, Abb. 24 Beispiele für unsichere Muster)

- Identifikation durch Wissen

Ein Muster ist wie ein Passwort ein spezielles Wissen, das der berechtigte Benutzer besitzt. Wissen kann weitergegeben werden. Dies geschieht bewusst oder unbewusst.

- Verringerter Zeichensatz

Durch die Reduzierung der Bildschirmtastatur auf eine kleinere Anzahl an Buttons wird auch die Kombinationsmöglichkeit drastisch verringert. Für einen Angreifer ist es leichter alle Kombinationen durchzutesten. Jedoch ist es für die Verwendung der cardTAN sowieso notwendig, dass die PIN des e-Banking Zuganges nur aus numerischen Zeichen besteht. Somit kann der Nachteil der Zeichensatzreduzierung ignoriert werden. Weitere Sicherheit kann hinzugefügt werden, wenn die Position der einzelnen Button für jeden Aufruf unterschiedlich ist. Somit ist die Beschriftung bzw das Symbol und nicht die Position wichtig.

9.2.2 Loginauthentifikation

Die Zugangsdaten zum e-Banking Account werden im Telefonspeicher abgelegt und müssen somit nicht bei jedem Zugriff eingegeben werden. Für den Login wird ein eigenes, vom Benutzer vergebenes Passwort verwendet. Dieses wird mit Hilfe des Touchmusters eingegeben. Das Passwort wird mit dem Hashwert im Telefonspeicher abgeglichen. Um einen Brute-Force Angriff auf den Hashwert zu erschweren, ist der Wert mit Salt gehasht.

Die Zugangsdaten werden verschlüsselt im Telefonspeicher abgelegt. Zur Verschlüsselung dient ein Schlüssel, der aus dem Klartext des Passwortes generiert wird.

9.2.2.1 Vorteile

- Schutz der Zugangsdaten

Die Zugangsdaten müssen nur beim ersten Login eingegeben werden. Danach wird nur mehr nach dem persönlichen Passwort gefragt. (vgl Seite 72, 8.1.4) Dieses wird mit der Referenz im Telefonspeicher abgeglichen. Schafft es ein Angreifer, das Passwort durch Änderung des Referenzhashes zu manipulieren, kann er die Zugangsdaten weiterhin nicht entschlüsseln. Der Zugang ist nur mit dem richtigen Passwort möglich. (vgl Seite 116, 9.2.7)

- Benutzerfreundlichkeit durch Passwort anstatt Zugangsdaten

Die Zugangsdaten (Kontonummer, BLZ, Verfügernummer) sind lang und schwer zu merken. Durch das Passwort ist es für den Benutzer einfach, sich eine für ihn leicht merkbare Kombination zu erstellen.

9.2.2.2 Nachteile

- Daten im Telefonspeicher abgelegt

Die Reverenzdaten für den Vergleich und die Zugangsdaten werden im Telefonspeicher abgelegt. Somit kann unter Umständen auf die Daten von Extern zugegriffen werden. Da jedoch die Verschlüsselung der Zugangsdaten mit dem Passwort verknüpft ist, ist die Gefahr reduziert.

9.2.3 Identifikation mittels IMSI, IMEI

Nach der Anmeldung wird die IMSI und IMEI mit dem Server abgeglichen. Damit soll gewährleistet werden, dass die Verbindung von einem bestimmten registrierten Gerät gestartet wurde. Aufgrund der Einschränkungen der Benutzerfreundlichkeit sollte diese Funktion deaktivierbar sein.

9.2.3.1 Vorteile

- Einschränkung des Zuganges auf bestimmte Geräte

Der e-Banking Zugang wird auf ein paar Geräte beschränkt, die am Server eingetragen sein müssen. Dadurch ist die alleinige Kenntnis der Zugangsdaten nicht mehr ausreichend, um das Konto von einem anderen mobilen Gerät nutzen zu können. Im Rechenzentrum kann somit auch mitgeloggt werden, von welchem Gerät die Verbindung aufgebaut und auf das Konto zugegriffen wird.

9.2.3.2 Nachteile

- Notwendigkeit einer Konfiguration des Zuganges

Die Identifikation muss konfiguriert werden. Wird standardmäßig die IMSI oder IMEI überprüft, ist ein Login vor der Konfiguration bei einer Bankstelle oder vom Computer aus nicht möglich.

- Handywechsel benötigt eine Rekonfiguration des Zuganges

Da die IMSI für die Simkarte und IMEI für das Gerät einzigartig sind, ist der Zugang ausschließlich von den konfigurierten Geräten aus möglich. Verwendet der Kunde ein neues Gerät, ist es notwendig, dass der Zugang neu konfiguriert bzw das neue Gerät dem Zugang zugeordnet wird.

9.2.4 Serverzertifikat

Durch das Serverzertifikat kann sich der Server gegenüber dem Client authentifizieren. (vgl Seite 76, 8.3.1) Bei Android kann das Zertifikat entweder mit einem eigenen Zertifikatsspeicher verglichen werden oder mit dem unter Android Vorimplementierten. Ein eigener Speicher hat den

Vorteil, dass auch eigene unsignierte Zertifikate verwendet bzw die Zertifikatsliste auf die einige Notwendige reduziert werden können.

9.2.4.1 Vorteile

- Authentizität des Servers wird gewährleistet

Durch das Zertifikat kann die Authentizität des Servers gewährleistet werden, wenn die Vertrauenskette bis zu einer bekannten und vertrauenswürdigen CA nachvollzogen werden kann.

- Zertifikat wird für sicheres TLS benötigt

Die Verschlüsselung mit TLS kann auf unterschiedliche Weisen aufgebaut werden. Durch die Verwendung des Zertifikates kann auch der Verbindungsaufbau abhörsicher durchgeführt werden. (vgl Seite 80, 8.4.3)

9.2.4.2 Nachteile

- Signierte Zertifikate für den Server sind teuer

Das Ausstellen von Zertifikaten von einer offiziellen CA verursacht jährliche Kosten. Die Höhe der Kosten hängt von der ausstellenden CA und dem auszustellenden Zertifikat ab. Zertifikate können bei Firmen wie Verisign oder DigiCert gekauft und ausgestellt werden.

9.2.5 Verwendung von TLS für die Kommunikation

TLS verschlüsselt die Pakete vor dem Absenden und verhindert somit ein Mitlesen der Informationen durch Dritte. Beim Verbindungsaufbau wird der öffentliche Schlüssel des Serverzertifikates verwendet, damit auch der Handshake und der Austausch des symmetrischen Schlüssels nicht abgehört werden kann. (vgl Seite 80, 8.4.3)

9.2.5.1 Vorteile

- Abhörsicherheit

Bei der Verschlüsselung werden alle gesendeten Pakete mit AES verschlüsselt und so über das Netzwerk übertragen. Der Schlüssel für die Übertragung wird bei jedem Verbindungsaufbau neu erstellt, wodurch es für einen Angreifer fast unmöglich ist, die Übertragung im Klartext mitzulesen.

- Verbreiteter Standard

TLS ist ein weit verbreiteter Standard und wird sehr häufig eingesetzt. Somit ist er gut getestet und besitzt abhängig von der Implementierung wenige Sicherheitsprobleme. Durch häufige Sicherheitstests von staatlichen und privaten Institutionen kann der Sicherheitsstandard bestätigt werden.

9.2.5.2 Nachteile

- Übertragung von Sicherheitssystemen nicht lesbar

Der Übertragungstunnel verhindert nicht nur das Mitlesen durch Angreifer. Auch für Sicherheitssysteme wie Firewalls oder IDS, IPS ist die Überprüfung der Pakete nicht möglich. Abhilfe schafft hier ein vorzeitiges Beenden des TLS Tunnels. Dies kann unter anderem mit einem SSL Offloader (vgl Seite 106, 8.6.5) durchgeführt werden.

- Erhöhter Rechenaufwand

Die Verschlüsselung der Datenpakete benötigt im Vergleich zur Übertragung von Klartext viel Rechenleistung. Dieser zusätzliche Rechenaufwand für die Ver- und Entschlüsselung kann einen DoS Angriff erleichtern. Durch einen SSL Offloader kann diese zusätzliche Last vom Server ausgelagert werden.

9.2.6 Speicherung der Zugangsdaten und lokalen Einstellungen im Telefonspeicher

Um die Benutzerfreundlichkeit zu erhöhen und somit die Handhabung der APP zu erleichtern, müssen bestimmte Daten und Einstellungen gespeichert werden. Zur Speicherung stehen zwei Medien zur Verfügung: der Telefonspeicher und die SD Karte. Der Telefonspeicher ist fix im Gerät integriert und kann nicht gewechselt werden. (vgl Seite 93, 8.5.6)

9.2.6.1 Vorteile

- Schwieriger auszulesen als SD Karte

Der Telefonspeicher ist fix im Gerät verdrahtet und kann somit nicht direkt von einem Computer ausgelesen werden. Wird das mobile Gerät mit dem Computer verbunden, ist bei ungerooteten Geräten kein Zugriff auf den internen Speicher möglich, da dies vom Androidsystem unterbunden wird.

- Durch Berechtigungen geschützt

Im Telefonspeicher hat jede APP ihren eigenen Speicherbereich. Dieser kann nicht von anderen Anwendungen ausgelesen oder beschrieben werden.

- Wiederkehrende Eingaben und Einstellungen speicherbar

Eingaben, die wiederholt durchgeführt werden müssen (zB Zugangsdaten wie BLZ und Kontonummer), können zwischengespeichert und automatisch ausgefüllt werden. Damit kann die Benutzerfreundlichkeit gesteigert werden.

9.2.6.2 Nachteile

- Mit einer Entwicklungsumgebung auslesbar

Wird das Gerät über die USB Schnittstelle mit dem Computer verbunden und wird es im Entwicklungsmodus verwendet, ist es möglich, dass auch auf den Telefonspeicher zugegriffen wird. Daten, die dort abgelegt sind, können dann problemlos von einem Angreifer ausgelesen werden.

- Begrenzte Speicherkapazität

Der Telefonspeicher ist sehr begrenzt und wird auch zur Speicherung des Systems sowie aller APPs verwendet. Wenn der Speicher voll ist, können einerseits keine Daten mehr geschrieben werden und andererseits wirkt es sich negativ auf die Benutzerzufriedenheit aus, wenn die APP schuld an der Überfüllung des Speichers ist.

9.2.7 Verschlüsselung der Daten im Telefonspeicher

Bei der Verwendung des Telefonspeichers zur Speicherung von Daten muss berücksichtigt werden, dass unter bestimmten Umständen auch dieser ausgelesen werden kann. Somit sollten die Daten niemals im Klartext abgelegt werden. Durch eine Verschlüsselung wird der Zugriff auf die Informationen erschwert. (vgl Seite 96, 8.5.8)

9.2.7.1 Vorteile

- Auslesen der Daten erschwert

Durch die Verschlüsselung ist es für einen Angreifer nicht mehr so leicht möglich, die Daten, die im Speicher abgelegt wurden, auszulesen und sie zu verstehen.

9.2.7.2 Nachteile

- Erhöhter Rechenaufwand

Jede Verschlüsselung verursacht zusätzlichen Rechenaufwand für das Gerät. Beim Schreiben oder Lesen muss zuerst der Text ver- oder entschlüsselt werden, damit die Daten von der APP korrekt verstanden werden. Da die Rechenleistung von mobilen Geräten stetig zunimmt, ist dieser Nachteil vernachlässigbar.

- Kryptografischer Schlüssel notwendig

Für die Verschlüsselung muss ein Schlüssel generiert werden. Diese muss irgendwo zwischengespeichert werden, damit später wieder auf die verschlüsselten Daten zugegriffen werden kann. Eine Möglichkeit ist die Verwendung eines Passwortes, aus dem der Schlüssel generiert wird, das zu jeder Entschlüsselung (zB beim Login) vom Benutzer eingegeben werde muss.

9.2.8 Transaktionsautorisierung mittels cardTAN

Beim Absenden einer Transaktion muss diese mit Hilfe einer Transaktionsnummer (TAN) bestätigt werden. Bei dieser TAN ist es wichtig, dass sie von einem zusätzlichen Kommunikationskanal stammt, damit diese nicht vom System selber verändert wird oder die Transaktion automatisiert bestätigt werden kann. Die höchste Sicherheit in diesem Bereich bietet derzeit das cardTAN Verfahren. (vgl Seite 13, 3.2.5.3)

9.2.8.1 Vorteile

- Zweiter Kommunikationsweg

Da die TAN immer vom separaten TAN Generator erstellt wird, kann davon ausgegangen werden, dass sie immer über einen eigenen Kommunikationskanal erstellt wird. (vgl Seite 25, 5.3.3) Eine Veränderung der TAN oder Automatisierung der Authorisierung ist nicht möglich.

- Überprüfung der Überweisung am TAN Generator möglich

Es kann sein, dass die Überweisungssumme vor der Übertragung zum TAN Generator verändert wurde. Deshalb ist es wichtig, dass die Überweisungsdetails am Generator noch einmal vom Benutzer überprüft werden können. Dazu muss dieses Gerät über ein eigenes Display verfügen.

- Zusätzliche Sicherheitseingaben (PIN)

Die TAN wird nicht nur aus den Überweisungsdaten generiert. Auch die PIN der Bankomatkarte und die Karte selber werden in die Berechnung mit einbezogen.

9.2.8.2 Nachteile

- TAN Generator und Bankomatkarte notwendig

Für jede Überweisung ist es notwendig, dass der Benutzer nicht nur die Bankomatkarte sondern auch den TAN Generator bei sich hat. Beides wird für die Autorisierung benötigt. Dies kann zu Inakzeptanz bei den Kunden führen, da beim aktuellen System nur das Handy benötigt wird.

- Flickercode muss eingelesen werden

Die Überweisungsdaten werden mittels einer blinkenden Grafik (Flicker) zum TAN Generator übermittelt. Da die Bildschirme der Androidgeräte unterschiedlich groß sind, kann es zu Problemen beim Einlesen dieser Grafik kommen. Es ist wichtig, dass die Grafik entweder immer gleich groß oder die Größe vom Benutzer einstellbar ist.

9.2.9 Allgemeine Sicherheit (Rechenzentrum)

Für das Rechenzentrum sind allgemeine Sicherheitsmechanismen, wie in Kapitel 8.6.1 Allgemeine Sicherheitsvorkehrungen beschrieben, zu implementieren. Diese Mechanismen stellen die Standardsicherheit in einem Rechenzentrum dar.

9.2.9.1 Vorteile

- Basissicherheit

Durch diese Mechanismen wird die Sicherheit erhöht und der Zugriff auf die Systeme stark eingeschränkt. Für einen Angreifer werden dadurch Hürden gegen einen Zugriff erstellt und man verhindern somit viele Angriffe.

- Ausfallssicherheit

Durch die Verteilung der Services auf unterschiedliche Server hat ein Ausfall eines Gerätes nicht einen Totalausfall des Systems zur Folge. Durch weitere Sicherheitsmechanismen (Lastverteilung, SSL Offloading) kann die Ausfallssicherheit und die Möglichkeit, Angriffe abzuwehren, weiter gesteigert werden. (vgl Seite 119)

9.2.9.2 Nachteile

- Wartungsaufwand

Da der Server nicht direkt an das Internet angeschlossen wird, entstehen für die Geräte, die den Traffic vom Internet zum Server filtern, zusätzliche Wartungs- und Instandhaltungsaufwände. Jedoch erhöhen die zusätzlichen Sicherheitsmechanismen die Sicherheit und auch die Skalierbarkeit des Netzes und der Services enorm.

9.2.10 Konsistenzüberprüfung (Anfrage)

Es ist wichtig, dass die Konsistenz und Syntax der einzelnen Anfragen genau geprüft wird. Es dürfen nur vollständige Anfragen vom Server beantwortet und verarbeitet werden. Auch die Reihenfolge der Anfragen ist wichtig. Zu beachten ist jedoch auch, dass durch Probleme in der Kommunikation Pakete verloren gehen können. Die normale Ausführung der APP darf nicht gestört werden.

9.2.10.1 Vorteile

- Vorzeitiger Abbruch der Verbindung möglich

Ist eine Anfrage unvollständig und zum gegebenen Zeitpunkt unpassend bzw unmöglich, kann der Server die Verbindung trennen. Diese Anfragen können vor der Verarbeitung abgefangen und somit vorzeitig beendet werden. Die Effektivität einer einzelnen Anfrage kann in Bezug auf einen DoS

Angriff reduziert werden. Weiters kann eine veränderte Version der APP erkannt werden, sofern sich diese nicht an den definierten Kommunikationsablauf hält.

9.2.10.2 Nachteile

- Aufwändige Überprüfung

Jede Überprüfung kostet Rechenzeit und ist mit Aufwand verbunden. Je genauer die Anfragen überprüft werden, umso mehr Rechenzeit ist dafür notwendig. Diese Tatsache bremst nicht nur die APP selber, sondern kann auch bei einer DoS Attacke verwendet werden. Wichtig ist in diesem Fall jedoch, dass der Angreifer den Kommunikationsablauf kennt und die Syntax der Anfragen stimmt.

9.2.11 Lastverteilung, SSL Offloading

Durch die Lastverteilung wird es ermöglich, dass ein Serververbund für die Anwendung wie ein einzelner Server verwendet werden kann. Die Anfragen werden auf die einzelnen Server verteilt. (vgl Seite 105, 8.6.4)

Eine weitere Entlastung für einen einzelnen Server stellt SSL Offloading dar. Dabei wird der SSL Tunnel vorzeitig beendet. Der Server muss sich somit nicht um die Verschlüsselung kümmern. (vgl Seite 106, 8.6.5)

9.2.11.1 Vorteile

- Entlastung der einzelnen Server

Durch diese Maßnahmen werden die Server entlastet und können die Rechenleistung für die Verarbeitung der Anfragen verwenden. Die Ausfallssicherheit wird dadurch erhöht, da mehrere Server parallel die Anfragen abarbeiten können. Fällt ein Server aus, werden zwar die anderen stärker belastet, jedoch bleibt das Service verfügbar. DoS Attacken werden erschwert.

- Firewalls können Traffic mitlesen

Da der TLS Tunnel vom SSL Offloader vorzeitig beendet wird, kann dem Offloader eine Firewall nachgeschaltet werden, die den Traffic mitlesen kann. Damit können Angriffe, die ansonsten im TLS Tunnel versteckt bleiben würden, erkannt und abgewehrt werden.

9.2.11.2 Nachteile

- Kommunikation nach SSL Offloader unverschlüsselt

Der SSL Offloader entschlüsselt die Kommunikation und leitet sie unverschlüsselt an die Firewall, den Loadbalancer oder den Server weiter. Diese Kommunikationsstrecke muss sehr kurz gehalten werden, um ein Abhören zu verhindern. Alternativ kann sie auch erneut mit SSL verschlüsselt werden. Dann wird jedoch der Server wieder mit der Verschlüsselung der Kommunikation belastet.

- Zusätzlicher Konfigurationsaufwand

Die Verwendung eines SSL Offloaders und Loadbalancers stellt einen zusätzlichen Konfigurationsaufwand dar. Es ist notwendig, dass diese Geräte redundant ausgeführt werden, um die Ausfallssicherheit gewährleisten zu können. Auch ist es wichtig, dass ein bestimmtes Endgerät während einer Anmeldung immer mit demselben Server kommuniziert.

9.2.12 Beschränkung des Überweisungsbetrags

Viele Benutzer verwenden die APP, um kleine Überweisungen schnell und mobil durchführen zu können. Es ist nicht notwendig, dass das maximale Überweisungslimit, das durch das Authorisierungsverfahren gegeben ist, voll ausgeschöpft wird. Durch die Begrenzung kann verhindert werden, dass hohe Beträge mit wenig Aufwand von einem Angreifer überwiesen werden können. Außerdem können bei einem Überweisungsmaximum von 0€ Überweisungen vollständig deaktiviert werden. (vgl Seite 121, 9.2.14)

9.2.12.1 Vorteile

- Nur kleine Überweisungsbeträge möglich (benutzerangepasst)

Das Überweisungslimit ist an den Benutzer angepasst. Dadurch kommt es zu einer Erhöhung der Sicherheit, ohne dabei die Benutzerfreundlichkeit zu beeinträchtigen.

9.2.12.2 Nachteile

- Hohe Beträge müssen durch mehrmalige Transaktionen überwiesen werden

Beträge, die größer sind als das Limit, müssen durch mehrere Transaktionen überwiesen werden. Das führt zu Mehraufwand für den Benutzer und beeinträchtigt somit die Benutzerfreundlichkeit. Bei passender Konfiguration des Limits sollte das jedoch nicht oft vorkommen und somit dem Benutzer zumutbar sein.

9.2.13 Überweisungsanzahl beschränken

Wird der Überweisungsbetrag beschränkt, kann nur mehr eine bestimmte Summe pro Transaktion überwiesen werden. Jedoch ist es weiterhin möglich, einen hohen Betrag durch mehrere Transaktionen vom Konto auf ein anderes zu transferieren. Dies kann nur durch eine Beschränkung der Anzahl der möglichen Transaktionen in einem bestimmten Zeitintervall verhindert werden. (vgl Seite 121, 9.2.14)

9.2.13.1 Vorteile

- Sicherheit ohne Einschränkung der Benutzerfreundlichkeit

Die Anzahl der möglichen Überweisungen wird vom Benutzer manuell festgelegt. Es sollte sich an dem normalen Benützungsverhalten orientieren. Trifft das zu, kommt es zu keiner Einschränkung der Benutzerfreundlichkeit, jedoch kann ein Angreifer nur mehr einen maximalen Betrag von Überweisungslimit mal Überweisungsanzahl in der definierten Zeit vom Konto stehlen.

9.2.13.2 Nachteile

- Überweisungsbetrag zeitlich begrenzt

Große Überweisungen, die nicht im alltäglichen Verlauf vorgesehen sind, können nicht innerhalb einer Zeitperiode vom mobilen Gerät mit der APP durchgeführt werden. Der Benutzer muss die Überweisung entweder vom Computer oder bei einer Bankstelle durchführen oder sie über einen größeren Zeitraum verteilen. Dies kann zu großen Einschränkungen der Benutzerfreundlichkeit führen, jedoch sollten solche Überweisungen eher eine Seltenheit darstellen. Hohe Beträge sollten aus Gründen der Sicherheit weiterhin nicht vom mobilen Gerät beauftragt werden.

9.2.14 Deaktivierung von Funktionen

Die Deaktivierung von bestimmten Funktionen ermöglicht eine Erhöhung des Sicherheitsstandards, da bestimmte Informationen nicht gespeichert werden und somit von einem Angreifer nicht ausgelesen werden können oder bestimmte Aktionen von einem Angreifer nicht mehr durchgeführt werden können. Dadurch wird jedoch die Benutzerfreundlichkeit reduziert.

Weiters kann durch die Deaktivierung von anderen Funktionen die Benutzerfreundlichkeit auf Kosten der Sicherheit gesteigert werden. Beispiele hierfür sind die Identifikation mittels IMSI, IMEI, die Speicherung der Zugangsdaten und lokalen Einstellungen im Telefonspeicher und die Positionserkennung (GPS, WLAN).

Die Deaktivierung der Funktionen sollte (sofern möglich) vom Server durchgeführt werden und keine lokalen Konfigurationsdateien benötigen. Dadurch wird verhindert, dass der Angreifer die Konfiguration ändert und somit Zugriff auf die Daten und Funktionen erhält.

Die Konfiguration der Einstellungen sollte nur in einer Bankstelle oder über ein e-Bankingportal am Computer möglich sein. Alternativ kann eine Einstellungsmöglichkeit in der APP angeboten werden. Das Ändern muss mit einer TAN bestätigt werden, damit nicht ein Unberechtigter sich den Zugang so einstellen kann, wie er ihn benötigt.

9.2.14.1 Speicherung der Zugangsdaten

Die Zugangsdaten zum e-Banking Account bestehen aus Bankleitzahl, Kontonummer und Verfügernummer sowie der PIN. Die PIN darf auf keinen Fall gespeichert werden. Die anderen Daten sind für einen Benutzer schwer zu merken und umständlich einzugeben. Aus diesem Grund sollte

zur Erhöhung der Benutzerfreundlichkeit eine Möglichkeit zur Speicherung dieser Daten angeboten werden. Ist jedoch dem Angreifer die PIN bekannt, kann er sich mit dieser Zugang zum e-Banking Account verschaffen. Aus diesem Grund sollte die Speicherung nur nach expliziter Bestätigung durch den Kunden durchgeführt werden. Diese Daten sollten im Telefonspeicher abgelegt werden, da sie vor dem Verbindungsaufbau zum Bankserver benötigt werden. Im Telefonspeicher kann nur auf die eigenen Dateien zugegriffen werden, sofern das Gerät nicht gerootet ist und der Zugriff nicht mit root-Rechten durchgeführt wird. Somit sind die Daten auf einem ungerooteten Gerät sicher, sofern dieses nicht von einem Betriebssystembug betroffen ist.

9.2.14.2 Überweisungen

Viele Benutzer wollen die e-Banking APP nur zum Kontrollieren des Kontostandes und der Umsätze nutzen. Für diese ist die Überweisungsfunktion nicht notwendig sondern stellt nur ein Sicherheitsrisiko dar. Aus diesem Grund sollte es möglich sein, die Funktion abschalten zu können.

9.2.14.3 Daueraufträge

Daueraufträge stellen, wie unter Punkt 9.1.5.3 Daueraufträge erklärt, ein Sicherheitsproblem dar. Für viele Benutzer ist es nicht notwendig, Daueraufträge vom mobilen Endgerät zu erstellen. Aus diesem Grund sollte diese Funktion nur nach dem ausdrücklichen Wunsch des Kunden bei der Einrichtung des e-Banking Zuganges für das mobile Gerät aktiviert werden.

9.2.14.4 Vorteile

- Benutzer kann zwischen Sicherheit und Benutzerfreundlichkeit wählen

Der Benutzer kann durch die Konfiguration selbst bestimmen, welche Komfortfunktionen (zB Speicherung der Zugangsdaten) er nutzen will und welche nicht. Wichtig ist, dass er bei der Konfiguration der Einstellungen auf die möglichen Risiken hingewiesen wird.

- Benutzer kann Zugang auf seine Bedürfnisse beschränken

Manche Benutzer benötigen nur bestimmte Funktionen. Durch die Deaktivierung der nicht benötigten kann die Sicherheit erhöht werden, ohne dass dadurch die Benutzerfreundlichkeit eingeschränkt wird.

9.2.14.5 Nachteile

- Kein fixes Sicherheitsniveau unter allen Benutzern

Da sich die einzelnen Benutzer mit unterschiedlichen Einstellungen ins Rechenzentrum verbinden, kann nicht davon ausgegangen werden, dass alle mit maximaler Sicherheit die APP verwenden. Für die Sicherheit im Rechenzentrum sind diese Einstellungen egal, da sich dadurch nur ändert, wie auf die APP und das Konto zugegriffen werden kann.

- Konfigurationen müssen am Server abgelegt und abgerufen werden

Um die Einstellungen vor unberechtigter Veränderung zu schützen, müssen sie im Rechenzentrum abgelegt werden. Dadurch ist es notwendig, dass sie beim Login vom Server abgerufen werden. Die zusätzlich notwendige Datenübertragung ist jedoch vernachlässigbar.

9.2.15 Erstkonfiguration

Wenn der Zugang erstmals von einem Smartphone verwendet wird, wird die Erstkonfiguration gestartet. In dieser können die Sicherheitsmechanismen, die in Kapitel 9.2.14 Deaktivierung von Funktionen beschrieben sind, eingestellt werden. Die Konfiguration muss mittels cardTAN abgeschlossen werden.

9.2.15.1 Vorteile

- Konfiguration über die Bankstelle ist nicht notwendig

Die Konfiguration und Einrichtung des Zuganges kann vom Benutzer selbständig durchgeführt werden. Es ist nicht notwendig, dass er für die Ersteinrichtung eine Bankstelle aufsucht oder den Zugang über das e-Banking Portal am Computer einrichtet. Ob es sich um eine Erstkonfiguration handelt, wird vom System dadurch erkannt, dass der Zugang noch nicht in der Serverdatenbank konfiguriert ist.

9.2.15.2 Nachteile

- Denial of Service Angriff möglich

Kennt ein Angreifer die Zugangsdaten zum e-Banking Account, kann er den Zugang auf seinem Gerät vorkonfigurieren. Versucht nun der echte Benutzer die Verbindung aufzubauen, kann er den Zugang nicht mehr einstellen, da die Konfiguration bereits vom Angreifer vorgenommen wurde. Dieser DoS Angriff wird jedoch dadurch verhindert, dass die Konfiguration mittels cardTAN abgeschlossen werden muss. Der Angreifer kann diese TAN nicht liefern. Damit wird die e-Banking Session geschlossen.

Es besteht jedoch die Möglichkeit, dass ein Angreifer immer wieder versucht die Erstkonfiguration durchzuführen, wodurch das Service für den eigentlichen Benutzer unerreichbar wird. Um das zu verhindern, sollte die Erstkonfiguration von einer Bankstelle oder dem Webinterface am Computer eine höhere Priorität besitzen und eine Session von der APP abbrechen können.

9.3 Nicht verwendete Sicherheitsmechanismen

Bestimmte Sicherheitsmechanismen können zwar die Sicherheit der APP erhöhen, jedoch wird dadurch entweder die Benutzerfreundlichkeit stark beeinträchtigt oder die Ressourcen des mobilen

Gerätes übermäßig strapaziert. Die zusätzlich gewonnene Sicherheit ist in einigen Fällen auch eher gering.

9.3.1 Positionserkennung (GPS, WLAN)

Beim Login zum e-Banking Portal wird die Position des Benutzers überprüft, ob sie in einem bestimmten vorkonfigurierten Bereich liegt. Ein Angreifer müsste somit ebenfalls von diesem Gebiet aus den Angriff ausführen. Diese Einschränkung erhöht zwar teilweise die Sicherheit, sollte aber aufgrund der Nachteile nicht eingesetzt werden.

9.3.1.1 Vorteile

- Geografische Einschränkung der Verwendung des Zugangs

Der Zugang wird auf ein mehr oder weniger kleines Gebiet eingeschränkt. Das Rechenzentrum erlaubt nur dann die Verbindung, wenn die Anfrage auch aus dem vorkonfigurierten Gebiet durchgeführt wird.

- Sicherheitsmerkmale Gebietsabhängig

Basierend auf den Positionsinformationen können zusätzliche Sicherheitsabfragen und Mechanismen verwendet werden. Ein Beispiel hierfür wäre eine zusätzliche Passwortabfrage und die Überprüfung des Google-Accounts, wenn die Position nicht in Österreich ist. Bei einer Position außerhalb der EU wird der Zugang verweigert.

9.3.1.2 Nachteile

- Position nicht immer verfügbar

In vielen Fällen ist es nicht möglich, die Position des mobilen Gerätes zu bestimmen. Dies ist dann der Fall, wenn sich ein Benutzer zB innerhalb eines Gebäudes befindet oder die Positionserkennung im Androidsystem abgeschaltet wurde. Weiters kann es sein, dass beim Start der APP noch eine alte Position im System gespeichert ist und somit falsche Positionsdaten an den Server übermittelt werden.

- Zusätzliche Berechtigung notwendig

Für die Abfrage der Positionsdaten ist eine zusätzliche Berechtigung notwendig. Diese Berechtigung ist jedoch auch für den Bankomatfinder (vgl Seite 108, 9.1.1) erforderlich und steht daher ohnedies zur Verfügung.

- Hoher Stromverbrauch

Die Verwendung des GPS Moduls benötigt sehr viel Leistung. Dadurch wird der Akku sehr stark belastet, sodass die Laufzeit des Handys stark reduziert wird. Deshalb sollte die Positionsabfrage nur selten durchgeführt werden.

- Ungenau

Je nachdem wie die Position ermittelt wird, ist mit einer hohen Ungenauigkeit zu rechnen. Außerdem kann es sein, dass die Position nicht ermittelt werden kann und somit eine veraltete Position verwendet werden muss.

- Konfigurationsaufwand

Für die Zugangskontrolle mittels Positionsmessung ist es notwendig, dass vor der Verwendung für jeden Benutzer ein Gebiet festgelegt werden muss, in dem der Zugang genutzt werden darf. Diese Daten sollten im Rechenzentrum abgelegt werden, um eine Veränderung durch den Angreifer zu verhindern. Somit müsste die Konfiguration in einer Bankstelle oder über ein eigenes Portal vom Computer aus durchgeführt werden.

9.3.2 Sperrung des Zuganges mittels SMS

Bei Verlust des mobilen Gerätes ist es wichtig, dass der Zugang zu den e-Bankingdaten nicht mehr möglich ist, um den Datenverlust möglichst gering halten zu können. Die Löschung der Daten kann mittels SMS angestoßen werden. Jedoch ist es dazu wichtig, dass einerseits ein Service auf dem mobilen Gerät läuft, das die SMS lesen kann und andererseits muss der Absender der SMS genau authentifiziert werden können.

9.3.2.1 Vorteile

- Einfache Sperrung des Zuganges

Durch das Versenden der SMS ist es einfach, die gespeicherten Daten auf dem Smartphone zu löschen. Nach der Löschung sieht ein Angreifer nur mehr, dass die APP installiert ist. Der Zustand der Daten ist so, als wäre sie frisch installiert worden.

9.3.2.2 Nachteile

- Aufwändige Kontrollen zur Ursprungsauthentifizierung

Der Standard für SMS sieht keine Authentifizierung des Absenders gegenüber dem Empfänger vor. Eine gesendete SMS kann nicht zurückverfolgt werden, da die Absendernummer vom Absender verändert werden kann. (zB SMS von bestimmten Internetanbietern)

- Möglichkeit eines DoS Angriffs

Kann der Absender nicht authentifiziert werden, besteht die Möglichkeit, dass der Auftrag zur Löschung der Daten von irgendjemandem gesendet wird. Sendet ein Angreifer oft den Löschauftrag kann der Benutzer die APP nicht mehr sinnvoll nutzen, da er für jeden Login erneut seine Verbindungsdaten eingeben muss.

- Service zur SMS Auswertung ist notwendig

Um eingehende SMS-Nachrichten lesen zu können, muss entweder die APP durchgehend aktiv sein oder ein Service im Hintergrund laufen, das beim Empfang einer SMS aktiv wird. Dieses Service benötigt Systemressourcen und kann die Ausführung des Androidsystems verlangsamen. Die Benutzerakzeptanz sinkt dadurch.

- Zusätzliche Berechtigung wird benötigt

Um auf die eingehenden SMS reagieren zu können, ist es notwendig, dass die APP auch die Berechtigung erhält, SMS lesen zu dürfen. Dabei handelt es sich um eine sicherheitskritische Berechtigung, die von einer APP nicht angefordert werden sollte, da sie unter Umständen die Privatsphäre beeinträchtigt.

9.3.3 Überprüfung installierter Pakete, laufender APPs und Services

Beim Start der APP oder vor dem Durchführen einer Transaktion werden die derzeit laufenden APPs bzw Services auf Malware überprüft. Alternativ kann auch überprüft werden, welche Pakete derzeit auf dem mobilen Gerät installiert sind. Die Erkennung von Malware erfolgt entweder mit Hilfe der Liste der angeforderten Berechtigungen eines Paketes oder durch eine Blacklist, die mit dem Server synchronisiert werden muss.

9.3.3.1 Vorteile

- Früherkennung von Schadprogrammen

Dieser Mechanismus ermöglicht eine Erkennung von Schadprogrammen und kann somit einen Daten- und Informationsverlust verhindern.

9.3.3.2 Nachteile

- Aktualität der Vergleichslisten

Um Malware zu erkennen, ist es notwendig, die installierten oder laufenden Pakete mit einer Liste bekannter Schadsoftware zu vergleichen. Diese Listen müssen aktuell gehalten werden und sollten für jeden Vergleich mit dem Server synchronisiert werden. Das Aktuellhalten der Liste ist aufwändig.

- Erkennungsrate

Wird die Schadsoftware mit Hilfe der angeforderten Berechtigungen erkannt, ist es in vielen Fällen schwierig, diese von den nützlichen Programmen auseinander zu filtern. Manche Programme benötigen potentiell gefährliche Berechtigungen, um deren eigentliche Aufgabe durchführen zu können. Das Führen von Akzeptanzlisten ist nicht zweckmäßig, da es zu viele APPs gibt und die Akzeptanz oft im Ermessen des Benutzers liegt (vgl Seite 89, 8.5.4)

- Laufzeit

Das größte Problem stellt jedoch die Laufzeit dar. Das Auslesen der Softwarepakete und die Synchronisation mit dem Server verursachen langsame Speicherzugriffe und Rechenzeit. Dadurch wird die Ausführung der APP stark gebremst und somit die Benutzerakzeptanz reduziert.

9.3.4 Schutz der APP gegen Veränderung von Außen

Für die Sicherheit des Zuganges ist es wichtig, dass die APP nicht verändert wird. Wie im Kapitel „8.5.1 Schutz der APP gegen Veränderung von Außen“ beschrieben ist es jedoch nicht möglich, dass die Konsistenz der APP fehlerfrei überprüft wird. Es können zwar unterschiedliche Daten von der APP angefordert werden, die nur schwer fälschbar sind, jedoch kann die Antwort hardcoded zurückgeliefert werden und somit die Sicherheitsüberprüfung umgangen werden.

9.3.4.1 Vorteile

- Verhinderung von Reverse Engineering

Kann die Integrität der APP positiv und vertrauenswürdig bestätigt werden, können damit auch die Gefährdungen von Reverse Engineering verhindert werden. Es ist jedoch nicht möglich, die Integrität vertrauenswürdig zu testen, da das System selber als nicht vertrauenswürdig anzusehen ist.

9.3.4.2 Nachteile

- Genaue Überprüfung der APP auf Veränderungen ist nicht möglich

Wie bereits erwähnt, ist es nicht möglich, die APP gegen Veränderungen zu prüfen. Es ist nicht nachvollziehbar, woher die gesendeten Daten kommen, und ob es sich um einen Replay mit den geforderten Daten handelt.

- Konsistenzprüfung ist teilweise sehr aufwändig

Bestimmte Methoden der Konsistenzprüfung sind sehr aufwändig. Das Auslesen der Binarys der APP und das Erzeugen eines Hashstrings benötigen viele Speicherzugriffe und viel Rechenleistung. Damit würde dies auch zu einer erhöhten Laufzeit führen. Der Wert der Berechnung kann jedoch auch durch eine gefälschte APP vordefiniert an den Server übermittelt werden und somit diesem eine unveränderte Applikation vorspielen.

9.3.5 Erkennung eines gerooteten Gerätes

Wenn das mobile Gerät gerootet wird, hat nicht nur der Benutzer vollen Zugriff auf das Gerät, sondern auch die Anwendungen, die die Root-Berechtigung angefordert hat. Dieser Zugriff kann für das Gerät aber auch für andere Anwendungen gefährlich werden, da diese nicht mehr alleine auf ihre Daten zugreifen. Durch die Erkennung, ob das Gerät gerootet ist, kann der Benutzer gewarnt werden, dass sein Gerät potentiell unsicher ist.

9.3.5.1 Vorteile

- Warnung bei Sicherheitsrisiken

Ist auf dem Gerät ein Root-Zugriff möglich, kann der Benutzer gewarnt werden. Ob das Sicherheitsrisiko eingegangen wird oder nicht, sollte jedoch dem Benutzer überlassen werden. Ein automatisches Sperren des e-Banking Zugriffs von diesem Gerät hätte fatale Folgen für die Benutzerfreundlichkeit, da viele Geräte gerootet sind.

9.3.5.2 Nachteile

- Erkennung oft schwer bis unmöglich

Die Erkennung, ob ein mobiles Gerät gerootet ist oder nicht, ist sehr schwierig und in manchen Fällen nicht möglich. Es ist somit vor allem bei neueren Androidversionen mit einer hohen False-Error- und False-Positive-Rate zu rechnen, da einerseits die originalen Android-Versionen den APPs mehr Möglichkeiten zur Verfügung stellen, und andererseits die Custom-Firmwares nicht mehr von den anderen zu unterscheiden sind, da auch diese immer öfter keinen direkten Root-Zugriff mehr erlauben.

9.4 Benutzerfreundlichkeit

Die APP muss möglichst benutzerfreundlich sein, um eine hohe Benutzerakzeptanz zu erhalten. Dabei ist es wichtig, dass die Benutzerfreundlichkeit nicht zu sehr auf Kosten der Sicherheit eingeschränkt wird.

Die Übersichtlichkeit wird dadurch gewährleistet, dass pro Anzeige nur ein kleiner Ausschnitt der Daten angezeigt wird. Für das Menü existiert eine eigene Activity. Der Workflow ist für Kontrollen des Finanzstatus und Überweisungen eingerichtet. Die weiteren Funktionen sind über das Menü erreichbar.

Die Konfiguration der Sicherheitseinstellungen (Beschränkung des Überweisungsbetrags, Überweisungsanzahl beschränken, Deaktivierung von Funktionen) kann vom Menü aus durchgeführt werden (vgl Seite 120ff). Zur Bestätigung ist eine TAN erforderlich. Alternativ

können diese Einstellungen vom e-Banking Portal am Computer und in einer Bankstelle geändert werden.

9.5 Schematische Infrastruktur

Das Rechenzentrum ist in drei Zonen eingeteilt. Die Erste wird durch den Internetanschluss gebildet. Sie ist nicht vertrauenswürdig. Die Datenpakete werden zu einer Firewall weitergeleitet. Diese filtert den Traffic und blockt Angriffe aus unverschlüsselten Kommunikationskanälen. Sie kann nicht in verschlüsselte Tunnel eingreifen.

Die zweite Zone liegt hinter der ersten Firewall. Sie gilt als teilweise sicher, da die kryptografisch gesicherte Kommunikation noch nicht auf Angriffe geprüft wurde. Die Verschlüsselung wird mit Hilfe von SSL Offloader geöffnet und die Datenpakete durch die zweite Firewall geleitet. Diese kann nun alle Daten auf Angriffe und Malware scannen. Offloader und Firewall können auch in einem Gerät ausgeführt sein. Sie müssen redundant ausgeführt werden, um im Falle eines Ausfalls weiterhin Anfragen abarbeiten zu können.

Die letzte Zone wird als voll vertrauenswürdig angesehen und ist durch einen high availability Cluster gebildet. Dieser besteht aus redundanten Lastverteilern und mehreren nachgeschalteten Servern, die vom Aufbau her ident sind. Die Lastverteiler sind in einem Active/Active Betrieb, wodurch sie auch die Last der ankommenden Anfragen untereinander verteilen. Die Weiterleitung zu den Servern erfolgt mittels SLB (Server Load Balancing). Da die Verschlüsselung durch die Offloader beendet wurde, ist es wichtig, dass hier keine weiteren Geräte, die für e-Banking nicht benötigt werden, ins Netz eingebunden werden. Um die Anfragen von den Clients zu beantworten, greifen die Server auf weitere Services und Datenbanken zu.

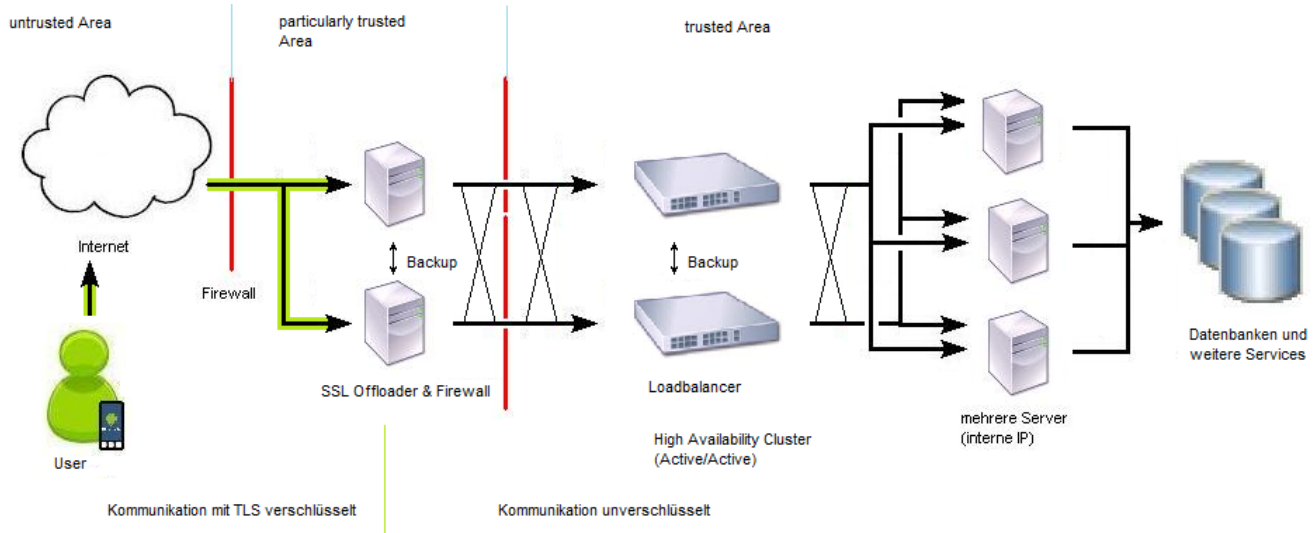


Abb. 28. Infrastruktur Rechenzentrum

9.6 Benötigte Berechtigungen

Um die Funktionen und Sicherheitsmerkmale implementieren zu können, sind folgende Berechtigungen notwendig:

- Uneingeschränkter Internetzugriff
- Genauer (GPS-) Standort
- Telefonstatus lesen und identifizieren

Der uneingeschränkte Internetzugriff ist notwendig, um die Kernfunktionalität (e-Banking Zugriff) gewährleisten zu können. Ohne diese Berechtigung ist es nicht möglich, mit den Bankservern zu kommunizieren

Der (GPS-) Standort ermöglicht die Ortung des Gerätes. Diese Berechtigung wird vor allem für den Bankomat- und Filialfinder benötigt.

Die Identifikation mittels IMSI oder IMEI kann nur mit der Berechtigung „Telefonstatus lesen und identifizieren“ durchgeführt werden. Diese Identifikation ermöglicht eine bessere Zuordnung der Transaktionen zum Kontoinhaber.

9.7 Mögliche Angriffsszenarien

In diesem Kapitel werden Möglichkeiten aufgeführt, wie bestimmte Angriffe oder Schwachstellen möglichst einfach und sicher abgewehrt bzw geschlossen werden, oder welche Sicherheitsmechanismen diese Schwachstelle verhindern sollen.

9.7.1 PIN Verlust

Geht die PIN verloren oder wird öffentlich bekannt, ist es wichtig, dass sie möglichst bald nach dem Bekanntwerden geändert wird.

Je mehr über den Bankzugang bekannt wird, umso wahrscheinlicher ist, dass der Angreifer Zugriff auf Bankdaten erhält. Damit das nicht eintritt, wird vom System der Login überprüft (vgl Seite 112, 9.2.2) und die Identität des Gerätes mittels IMEI und IMSI (vgl Seite 113, 9.2.3) sichergestellt. Gespeicherte Zugangsdaten sind im Telefonspeicher verschlüsselt abgelegt (vgl Seite 116, 9.2.7). Auf sie kann nur durch korrekte Eingabe des Touchmusters (vgl Seite 111, 9.2.1) zugegriffen werden.

Die PIN ermöglicht nur den lesenden Zugriff auf das Konto. Transaktionen können nur mit der TAN durchgeführt werden (vgl Seite 117, 9.2.8). Zusätzlich schützen noch eine Beschränkung der Überweisungsanzahl (vgl Seite 120, 9.2.13) und des Überweisungsbetrages (vgl Seite 120, 9.2.12) sowie die Deaktivierung von Funktionen (vgl Seite 121, 9.2.14) die finanziellen Mittel.

Durch den Verlust oder das Bekanntwerden der PIN hat ein Angreifer abhängig von seinem weiteren Wissen Zugriff auf den e-Banking Zugang und die Kontoinformationen. (vgl Seite 18, 4.1)

Der Verlust der PIN führt zwar dazu, dass der Angreifer privaten Daten einsehen kann, sofern auch die restlichen Zugangsdaten (BLZ, Kontonummer, Verfügernummer) bekannt sind. Ein finanzieller Schaden ist jedoch ohne das Bekanntwerden der TAN nicht möglich.

9.7.2 Verlust des mobilen Gerätes

Durch den Verlust des mobilen Gerätes gehen im Normalfall keine Daten verloren. Sind jedoch die Zugangsdaten (BLZ, Kontonummer, Verfügernummer) gespeichert, können diese vom Telefonspeicher ausgelesen werden (vgl Seite 115, 9.2.6). Die ausgelesenen Daten sind ohne den dazugehörigen Code des Touchmusters (vgl Seite 111, 9.2.1) jedoch wertlos, da sie verschlüsselt sind (vgl Seite 116, 9.2.7).

Werden mit dem Verlust des Gerätes zusätzlich die Zugangsdaten oder das Touchmuster und die PIN bekannt, hat ein Angreifer Zugriff auf den Finanzsstatus. Vor der unberechtigten Durchführung von Transaktionen und dem Abbuchen von Geld schützt die TAN (vgl Seite 117, 9.2.8), aber auch die persönliche Sicherheitskonfiguration im Hinsicht auf die Beschränkung des Überweisungsbetrags (vgl Seite 120, 9.2.12) und Anzahl (vgl Seite 120, 9.2.13) und der Deaktivierung von Funktionen (vgl Seite 121, 9.2.14).

Durch den Verlust des mobilen Gerätes sind der e-Banking Zugang (vgl Seite 18, 4.1), das Mobilgerät (vgl Seite 20, 4.2) und die Applikationsdaten (vgl Seite 20, 4.3) gefährdet. Der Zugriff auf den e-Banking Zugang erfordert jedoch zusätzliches Wissen (Touchmuster, PIN, TAN) beim Angreifer.

Der Verlust des Smartphones stellt in erster Linie keine Gefahr für das Konto dar. Die Zugangsdaten können nur ausgelesen werden, wenn der Angreifer das Passwort kennt und die Daten vorher vom Benutzer gespeichert wurden.

9.7.3 Malwareverdacht am Endgerät

Befindet sich auf dem Smartphone eine Schadsoftware, kann diese nicht von der APP erkannt werden. Jedoch werden die Zugangsdaten, die im Telefonspeicher (vgl Seite 115, 9.2.6) abgelegt sind mit kryptografischen Mitteln geschützt (vgl Seite 116, 9.2.7) und sind nur mit dem passenden Touchmuster (vgl Seite 111, 9.2.1) zugänglich. Die Übertragung wird durch TLS abhörsicher (vgl Seite 114, 9.2.5). Um die Überweisungen vor einem automatischen Absenden zu schützen, wird eine TAN (vgl Seite 117, 9.2.8) ver-wendet.

Auf einem ungerooteten Gerät ist es für Schadsoftware schwierig, auf die Daten der APP zuzugreifen. Ist das Gerät jedoch gerootet, kann unter Umständen auf viele interne Daten wie dem internen Telefonspeicher oder den Framebuffer zugegriffen werden. Damit könnten im Hintergrund Screenshots erzeugt und über das Internet versendet werden. Da Buttons, die gerade benützt werden, von Android farbig hinterlegt werden, kann durch diese Methode auch das Passwort oder die PIN ausgelesen werden.

Malware erhält unter Umständen nicht nur Zugriff auf Benutzer- und Anwendungsdaten sondern auch auf das Mobilgerät und die Kommunikation. (vgl Seite 18ff, 4.1; 4.2; 4.3; 4.4)

Malware ist vor allem auf gerooteten Geräten ein Problem, da es für einen Angreifer leichter ist, Zugriff auf das Konto und die Kontodaten zu erhalten, als auf einem Ungerooteten. Schadprogramme können nur schwer vom Benutzer erkannt werden. Um Datenverluste zu vermeiden, sollte unbekannten APPs niemals Root-Zugriff gewährt werden. Da die Transaktionen mittels cardTAN bestätigt werden müssen, kann keine Finanztransaktion durchgeführt werden.

9.7.4 DoS Attacke auf das Rechenzentrum

Das Ziel einer DoS Attacke ist es, die Erreichbarkeit von Services und Servern zu unterbinden. Es soll die Verfügbarkeit gestört werden. Eine DoS Attacke nutzt die normalen Schnittstellen zum System, jedoch mit dem Unterschied, dass eine Anfrage zwar gestellt wird, dem Anfragensteller die Antwort jedoch egal ist. Da sehr viele Anfragen, die der Server oder das Service abarbeitet, in kurzer Zeit gestellt werden, führt dies zu einer Überlastung der Infrastruktur.

Um die Herkunft eines DoS Angriffs zu verschleiern und noch mehr Anfragen stellen zu können, verwenden Angreifer heutzutage ein Bot-Netz mit vielen Rechnern und führen eine DDoS Attacke durch. Diese kann nur durch sinnvolles Verteilen der Last auf mehrere Server abgefangen werden (vgl Seite 119, 9.2.11). Weiters ist es wichtig, eine fehlerhafte Anfrage möglichst früh zu verwerfen. Das kann durch die Firewall oder vom Server beim Überprüfen der Benutzereingaben durchgeführt werden. (vgl Seite 118, 9.2.9; 9.2.10)

Während eines DoS oder DDoS Angriffs sind die Kommunikation (vgl Seite 20, 4.4) und das Rechenzentrum (vgl Seite 20, 4.5) gefährdet.

9.7.5 Verwendung des SSL/TLS Tunnels zum Einschleusen von Schadcode

Damit die APP sicher mit dem Server im Rechenzentrum kommunizieren kann, müssen die Datenpakete verschlüsselt werden. Dies wird mit Hilfe eines TLS Tunnels erledigt. Wird dieser Tunnel jedoch für andere Zwecke missbraucht, erhält ein Angreifer Direktzugriff auf den Server. Eine Firewall kann nicht direkt in den Tunnel eingreifen ohne ihn zu verändern.

Durch diesen Angriff ist nur das Rechenzentrum betroffen und gefährdet. (vgl Seite 20, 4.5)

Die Zweckentfremdung des TLS-Tunnels kann durch die verwendeten Sicherheitsvorkehrungen (Allgemeine Sicherheit (Rechenzentrum), Konsistenzüberprüfung (Anfrage), Lastverteilung, SSL Offloading) gut unterbunden und die Gefährdung reduziert werden, da einerseits der Traffic noch durch die Firewalls gescannt wird und andererseits mit dem Zielserver nur über bestimmte Ports kommuniziert werden kann. (vgl Seite 118ff)

9.7.6 Fälschung der IMEI oder IMSI durch die Verwendung eines eigenen Programms für den Zugriff (Replay Attacke)

Die Authentifikation mittels IMEI und IMSI ist optional und kann deaktiviert werden. Die Deaktivierung kann vorgenommen werden, um dem Benutzer die Möglichkeit zu geben, die APP von mehreren mobilen Geräten aus zu nutzen. Es ist als zusätzliches Sicherheitsfeature zu interpretieren.

Durch das Fälschen der IMEI oder IMSI kann ein Angreifer die Einschränkung des Zuganges auf ein bestimmtes Gerät umgehen. Da die Authentifikation des Benutzers gegenüber dem System durch die IMSI und IMEI unterstützt wird und nicht ausschließlich auf diesen Werten basiert, entsteht durch das Fälschen dieser nicht unbedingt eine Gefährdung für die Werte des Systems. Sie wirkt unterstützend zur Loginauthentifikation mittels Zugangsdaten und PIN (vgl Seite 112, 9.2.2). Die Finanzmittel sind außerdem noch durch die TAN geschützt (vgl Seite 117, 9.2.8).

Sind dem Angreifer jedoch neben der IMSI und IMEI auch die Zugangsdaten und die PIN bekannt, kann er von einem beliebigen Gerät Zugriff auf das Konto aufnehmen, wenn er es schafft, dass dieses mit der gefälschten Geräteidentifikation sendet.

9.7.7 Verwendung eines Proxys für Man in the Middle Attacke (Aufbrechen der SSL Verschlüsselung)

Damit ein Man in the Middle Angriff durchgeführt werden kann, muss der Angreifer die Daten entweder vor der Verschlüsselung oder nach der Entschlüsselung abfangen, oder die Verschlüsselung aufbrechen Das ist nur möglich, wenn der Angreifer von Anfang an an der Kommunikation beteiligt ist. Die Verbindung wird mit dem Serverzertifikat aufgebaut, wodurch kein unverschlüsseltes Mithören möglich ist. Die Verschlüsselung kann jedoch aufgebrochen werden, indem der Angreifer dem Client sein Zertifikat als das vom Server ausgibt, und er dann die Verbindung zum Server aufbaut. (vgl Seite 40, 6.11.5)

Dieser Angriff wird durch das Serverzertifikat (vgl Seite 113, 9.2.4) und die Verschlüsselung der Kommunikation mittels TLS (vgl Seite 114, 9.2.5) erschwert.

Android erlaubt im Normalfall nur vertrauenswürdige Zertifikate. Wird in der APP ein eigener CertStore verwendet, können die akzeptierten Zertifikate weiter eingeschränkt werden. Sollte ein Angreifer dennoch die Kommunikation mithören können, ist es ihm nicht möglich, Überweisungen zu verändern. Dies wird durch die Transaktionsautorisierung mittels cardTAN (vgl Seite 117, 9.2.8) verhindert, da die TAN nur für eine bestimmte Überweisung gültig ist.

Durch einen Man in the Middle Angriff wird vor allem die Kommunikation (vgl Seite 20, 4.4) gefährdet.

Da die Kommunikation alle e-Banking Daten überträgt, sind im Falle des Aufbrechens der Tunnelverschlüsselung auch die e-Banking Zugangsdaten und die Kontoinformationen betroffen. (vgl Seite 18, 4.1)

Der Angreifer kann zwar die Konto- und Finanzinformationen mitlesen, jedoch kann er keine passende TAN liefern, um Überweisungen durchzuführen. Er kann aber auf die Unachtsamkeit des Benutzers hoffen und die Überweisungen zu seinen Gunsten abändern. Eine vom Benutzer eingegebene TAN könnte dann eine geänderte Transaktion bestätigen. In diesem Fall ist es wichtig, dass der Bankkunde die Überweisung bei der Generierung der TAN auch am TAN-Generator kontrolliert. Unregelmäßigkeiten sollten dem Geldinstitut gemeldet werden, um die Sicherheit weiterhin gewährleisten zu können.

9.8 Implementierung des Prototyp für den Singlebank Entwurf

Der Prototyp wurde in Java entwickelt und soll die grundlegenden Funktionen und Sicherheitsmechanismen veranschaulichen. Er ermöglicht es, Analysen bezüglich der Umsetzbarkeit und der Benutzerfreundlichkeit des Entwurfes für einen Singlebank e-Banking Systems durchzuführen.

9.8.1 Aufbau und Komponenten

Die Bestandteile des Prototyps sind einerseits ein Server, der die Anfragen vom Client entgegennimmt und abarbeitet und andererseits einem Client, der das Frontend für den Benutzer darstellt. Die Daten bezüglich Autorisierung und Konto sind in einer Datenbank (Derby) abgelegt. Der Zugriff auf die Datenbank erfolgt mittels prepared Statements direkt in SQL. Für die Kommunikation wurde ein eigenes Kommunikationsprotokoll entwickelt, das nur bestimmte Befehle akzeptiert. Unbekannte Befehle werden nicht weiterverarbeitet und führen zu einem Logout aus dem System (vgl. Seite 143, 9.8.13).

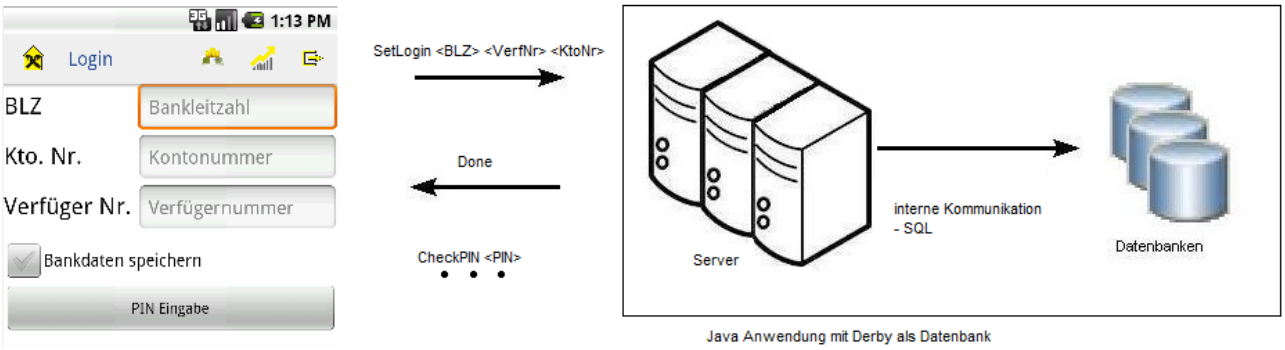


Abb. 29. Aufbau und Komponenten des Prototyps

Der Client ist unter Android programmiert und baut erst beim Login eine Verbindung zum Socket des Servers auf. Sie wird als Session verwendet und wird vom Server verwaltet. Wird sie unterbrochen, folgt ein automatischer Logout. Eine Wiederaufnahme ist nicht möglich.

Im Gegensatz zu bestehenden APPs handelt es sich hier um eine vollständige Java Applikation. Es werden keine Webpages angezeigt oder abgerufen und später interpretiert. Der Funktionsumfang ist vergleichbar mit ELBA mobil, jedoch erhöhen die zusätzlichen Sicherheitsmechanismen die Sicherheit. Ein großer Unterschied ist die individuelle Gestaltbarkeit des Zuganges. Damit kann vom Benutzer festgelegt werden, auf welche Funktionen er mobil Zugriff haben will und auf welche nicht.

Der Prototyp besitzt im derzeitigen Stadium noch nicht alle Banking-Funktionen, da das Hauptaugenmerk auf die Sicherheitsmechanismen gelegt wurde.

9.8.2 Flussdiagram und Navigation

Beim Start der APP wird die Activity „singlebank“ aufgerufen. Sie beinhaltet alle Funktionen, die für e-Banking notwendig sind. Bei bestimmten Funktionen werden zusätzliche Activites aufgerufen, um die volle Funktionalität gewährleisten zu können. (zB „callTouchmuster()“ öffnet „touchmuster_singlebank“)

Um eine schnelle Navigation gewährleisten zu können, kann durch die „ActionBar“ zum Menü, der Kontoübersicht und der Depotübersicht gewechselt werden. Im Menü sind alle Funktionen aufgeführt, die verfügbar sind (vgl Seite 141, 9.8.9).

Die vollständige Navigation des Prototyps ist im Anhang in Abb. 48 Funktionen und Activities des Prototyps ersichtlich.

9.8.3 Login

Der Login besteht aus zwei Masken in der Activity „singlebank“. In der ersten muss die Bankleitzahl, die Kontonummer und die Verfügernummer eingegeben werden. Es besteht die Möglichkeit, diese Zugangsdaten lokal am Gerät im Telefonspeicher abzulegen, um sie bei späteren Loginvorgängen nicht erneut eingeben zu müssen. Dazu muss bei der Eingabe der Verbindungsdaten die Checkbox „Bankdaten speichern“ ausgewählt werden. Voraussetzung dafür ist weiters die Aktivierung dieser in der Konfiguration. Ist das erfolgt, wird nach Drücken des „PIN Eingabe“ Buttons das Touchmuster abgefragt (vgl Seite 136, 9.8.4) und nach erfolgreicher Eingabe zur PIN-Eingabe verzweigt. Ansonsten wird diese direkt aufgerufen.

Sind Verbindungsdaten bereits im Telefonspeicher abgelegt, wird direkt beim Programmstart nach dem Touchmuster gefragt. Nach erfolgreicher Eingabe dieses Musters wird zur PIN-Abfrage weitergeleitet.

Nach dem Absenden der PIN wird der Login durchgeführt und eventuell verfügbare Nachrichten (vgl Seite 137, 9.8.5) angezeigt.

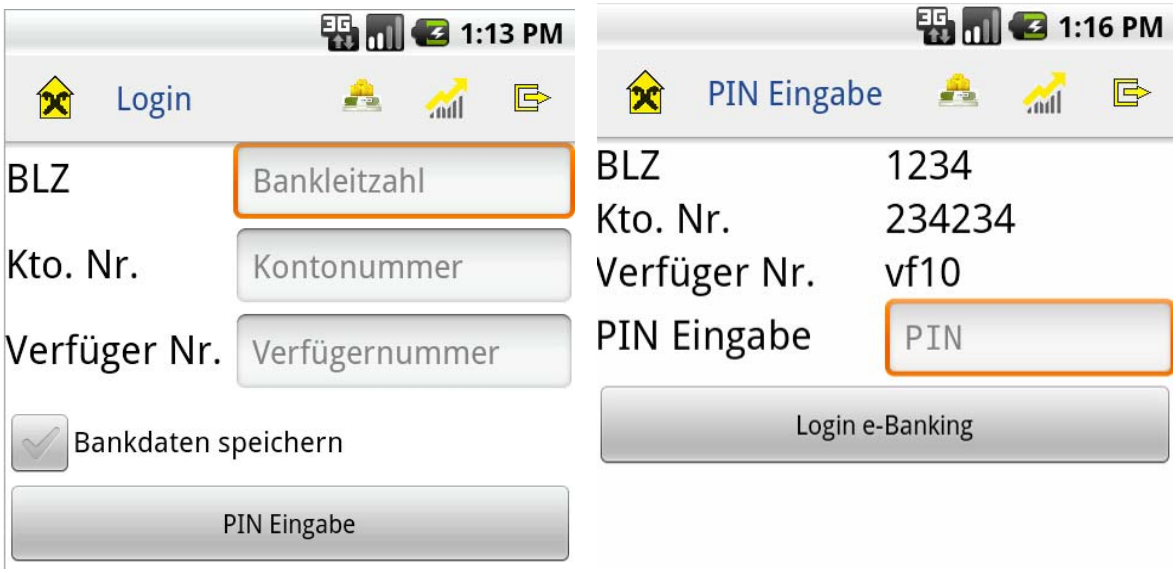


Abb. 30. Login und PIN-Eingabe (Prototyp)

9.8.4 Touchmuster

Das Touchmuster (vgl Seite 71, 8.1.3) dient der Eingabe des Passworts für die Zugangsdaten, die verschlüsselt im Telefonspeicher abgelegt sind. Es wird durch eine bestimmte Kombination von Buttons eingegeben, die in der richtigen Reihenfolge das Passwort ergeben. Jeder Button besitzt intern einen eigenen Wert. Beim Drücken wird dieser zum aktuellen Passwortstring hinzugerechnet. Derzeit erfolgt eine einfache Stringkonkartenation, jedoch kann diese durch einen komplexeren

Algorithmus ausgetauscht werden. Nach vollständiger Eingabe (drücken von „PIN Eingabe“) wird dieser String an die Activity zurückgegeben, die das Muster aufgerufen hat.

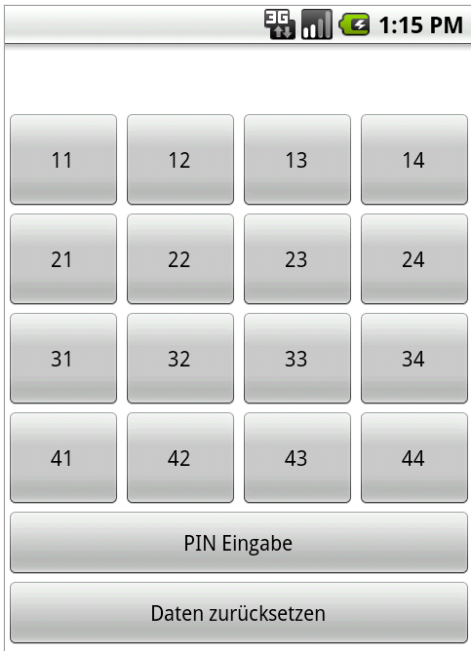


Abb. 31. Touchmuster (Prototyp)

In der Eingabemaske des Touchmusters kann auch das Passwort zurückgesetzt werden. Dabei werden alle Daten aus dem Telefonspeicher gelöscht. Die Zugangsdaten für die e-Banking Applikation müssen danach wieder vollständig eingegeben werden.

Das Touchmuster ist als eigene Activity implementiert und kommuniziert nicht mit dem Server.

9.8.5 Nachrichten

Nach dem Login werden die Nachrichten angezeigt, die in der Datenbank für den angemeldeten Benutzer gespeichert sind. Durch die Buttons „Vorherige“ und „Nächste“ kann durch die Nachrichten navigiert werden. Der Button „Weiter“ ruft die Kontoübersicht (vgl Seite 138, 9.8.6) auf. Die Anzeige der Nachricht besteht aus dem Namen des Absenders und dem Nachrichtentext.

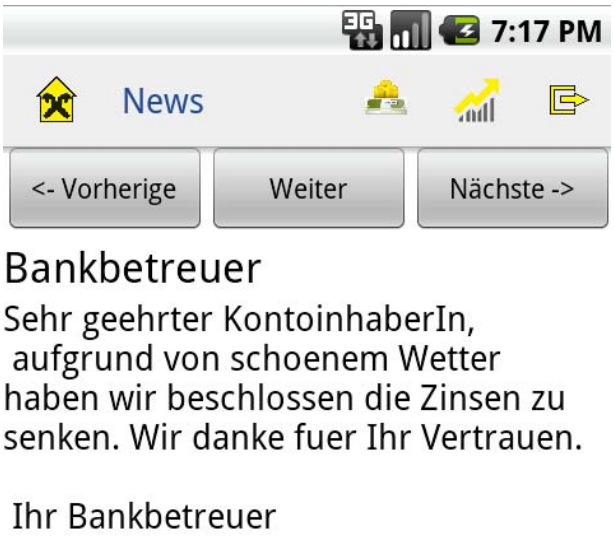


Abb. 32. Nachrichten (Prototyp)

9.8.6 Kontoübersicht

In der Kontoübersicht werden alle Konten angezeigt, für die der angemeldete Bearbeiter zeichnungsberechtigt ist. Die Konten werden mit der Kontonummer und deren Kontostand dargestellt.

Die Darstellung erfolgt als Liste und wird direkt vom Server abgefragt. Die Antwort erfolgt als String und wird am Client formatiert. Durch Auswählen eines Listeneintrages werden die Kontodetails für die gewählte Kontonummer aufgerufen (vgl Seite 138, 9.8.7).

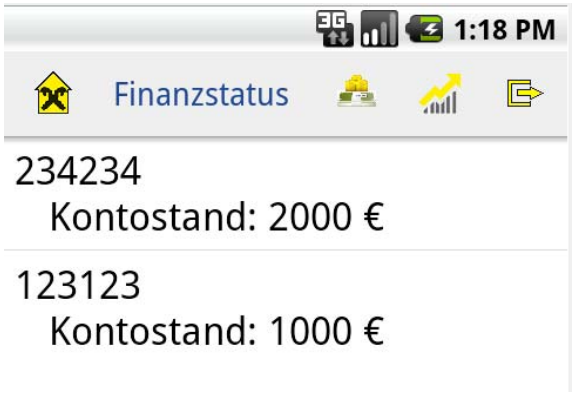


Abb. 33. Kontoübersicht (Prototyp)

9.8.7 Kontodetails und Überweisungserstellung

In den Kontodetails werden die Überweisungen zu diesem Konto innerhalb eines bestimmten Zeitraums angezeigt. Standardmäßig wird beim aktuellen Datum gestartet und die Transaktionen bis zwei Monate vorher ausgegeben. Der angezeigte Zeitraum kann vom Benutzer verändert werden. Von der APP wird keine maximale Zeitspanne vorgegeben, jedoch sollte der Zeitraum nicht zu groß gewählt werden, um die Übersichtlichkeit wahren zu können.

Abb. 34. Kontodetails (Prototyp)

Bei der Auswahl eines fehlerhaften Zeitraumes (zB Startdatum liegt nach Enddatum) wird eine Fehlermeldung ausgegeben.

Mit Hilfe des Buttons „Neue Überweisung“ kann eine neue Transaktion angelegt werden. Dazu wird eine neue Maske geöffnet. Das Referenzkonto ist vorausgewählt, kann jedoch nachträglich auf ein anderes Konto, auf das der angemeldete Verfüger Zugriff hat, geändert werden. Das Zielkonto wird derzeit noch nicht auf Gültigkeit überprüft.

Wird der Auftrag durch Auswählen von „Erstellen“ angelegt, wird der Betrag mit der Konfiguration verglichen. Er muss kleiner sein als der dort eingestellte Maximalbetrag für Überweisungen. Schlägt die Überprüfung fehl, wird eine Fehlermeldung ausgegeben, ansonsten folgt die Weiterleitung zur TAN-Eingabe.

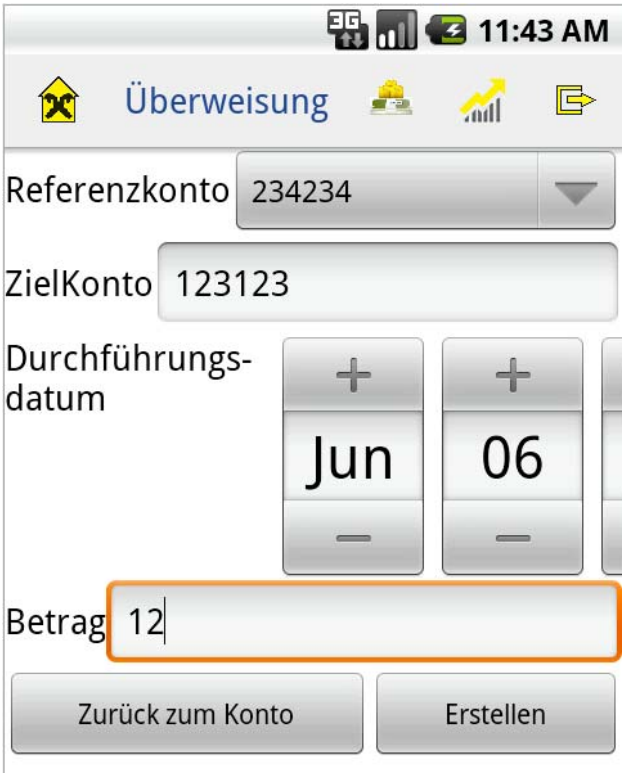


Abb. 35. Überweisung erstellen (Prototyp)

9.8.8 TAN Abfrage

Die TAN Abfrage bestätigt eine Transaktion und autorisiert die Bank, die Überweisung durchzuführen. (vgl Seite 13, 3.2.5.3). Die TAN-Abfrage ist beim Prototyp nicht vollständig implementiert, da die Implementierung und der verwendete Algorithmus für die Erstellung der cardTAN geheim sind.

Abb. 36. TAN-Eingabe (Prototyp)

Die Maske ist als eigene Activity ausgeführt und kommuniziert somit nicht direkt mit dem Server. Die TAN wird als Rückgabewert an die „singlebank“ Activity zurückgegeben, die die Überprüfung übernimmt.

9.8.9 Menü

Das Menü stellt eine Sammlung aller implementierten Funktionen dar, auf die der Benutzer je nach aktuellem Loginzustand Zugriff hat. Da ohne Login die Bankfunktionen nicht verwendet werden können, werden diese auch nicht angezeigt. Das Menü ist nur über die ActionBar (vgl Seite 142, 9.8.12) erreichbar.

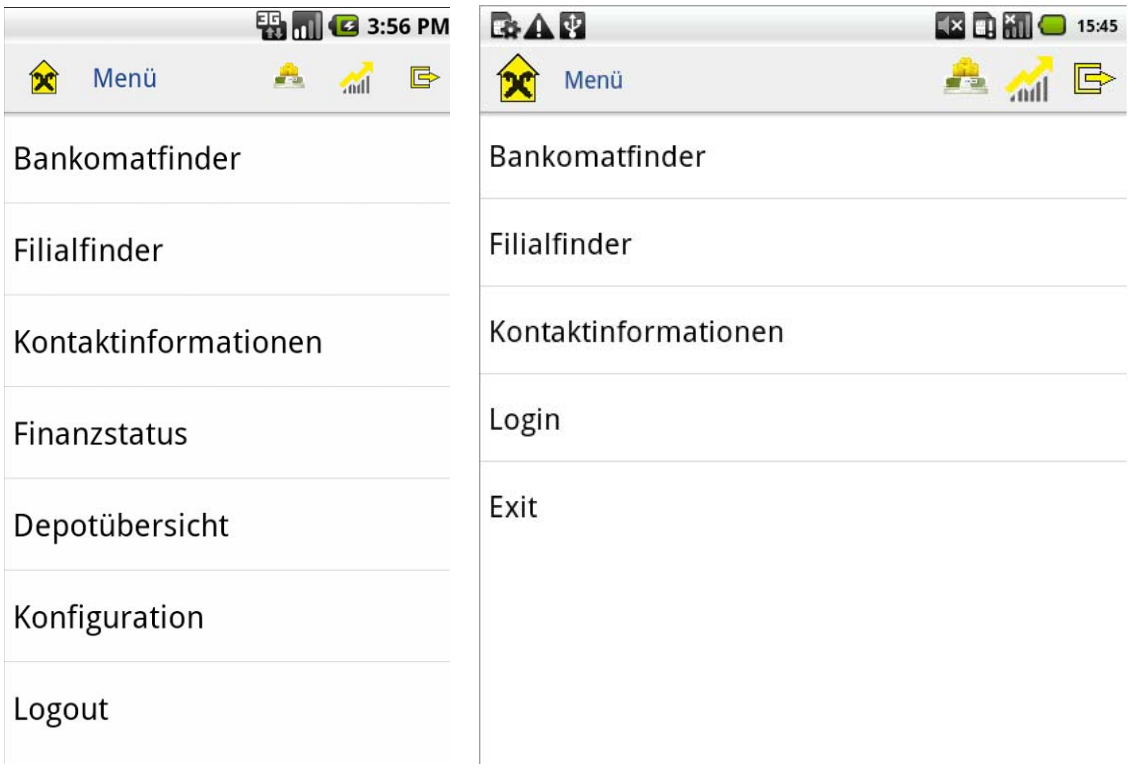


Abb. 37. Menü im eingeloggten und ausgeloggten Zustand (Prototyp)

9.8.10 Erstellte Aufträge

In der Übersicht „Erstellte Aufträge“ werden alle Überweisungen für ein bestimmtes Referenzkonto aufgeführt, die erstellt, aber noch nicht mit einer TAN bestätigt wurden. Die Konto-Auswahlliste zeigt nur Konten an, für die der eingeloggte Verfüger berechtigt ist.

Durch Auswahl einer Transaktion wird die TAN-Eingabe aufgerufen und der Auftrag kann abgeschlossen werden. Es darf immer nur ein Auftrag pro TAN abgesendet werden.

Abb. 38. Erstellte Transaktionen (Prototyp)

9.8.11 Konfiguration

Die Eingabemaske zur Konfiguration des e-Banking Zugangs kann nur über das Menü aufgerufen werden. Sie ermöglicht persönliche Einstellungen in Bezug auf Sicherheit (vgl Seite 121, 9.2.14).

Die neuen Einstellungen werden mittels „Speichern“ Button an den Server gesendet. Um die Sicherheit gewährleisten zu können, wird vorher noch nach einer TAN gefragt.

Abb. 39. Konfigurationsübersicht (Prototyp)

9.8.12 Actionbar

Die Actionbar basiert auf dem Beispiel ActionBarCompat von Android Developers[37]. Sie ermöglicht eine schnelle Navigation zwischen den einzelnen Funktionen. Da ab Android 3.0 der Menübutton nicht mehr Standard ist, wird diese Menüleiste in den Designrichtlinien für APPs vorgeschrieben.

Abb. 40. Actionbar (Prototyp)

Das Symbol am linken Rand öffnet das Menü (vgl Seite 141, 9.8.9). Im Anschluss daran ist der Name der aktuell angezeigten Maske zu lesen. Die Buttons am rechten Rand rufen die Kontoübersicht (vgl Seite 138, 9.8.6) und die Depotübersicht auf. Der letzte Button dient dem Logout und Schließen der APP.

9.8.13 Kommunikation

Die Kommunikation basiert auf einem einfachen selber entwickelten Protokoll. Es verwendet eine Socketverbindung, um zwischen dem Server und dem Client die Daten auszutauschen. Der Client sendet die Anfrage als String zum Server. Dieser empfängt, interpretiert und verarbeitet ihn. Datenbankabfragen werden bei Bedarf direkt aus der Verarbeitung heraus vom Server erstellt. Die Antwort wird wieder über die Socketverbindung an den Client zurückgesendet.

Es werden nur bekannte Befehle akzeptiert. Sie haben einen einheitlichen Aufbau. Es wird mit einem Kommando begonnen, das festlegt, was getan werden soll. Danach folgen die Parameter, deren Anzahl abhängig von der Anweisung unterschiedlich sein kann. Die Befehlserkennung ist casesensitiv. Weiters wird die Anzahl der Parameter sowie deren Gültigkeit vor der Weiterverarbeitung geprüft. Eine unbekannte Anweisung führt zu einem Logout aus dem System, da damit auf einen Angriff geschlossen wird.

Das eigene Protokoll hat den Vorteil, dass keine aufwändigen Parser, wie bei HTML oder XML benötigt werden. Dadurch kann ein Angriff mit XML Bombs verhindert werden. Weiters werden nur sehr wendige Daten übertragen, die ihrerseits wiederum keine oder nur sehr wenig wiederholende Teile (Header, Trailer) haben, die eine Verschlüsselung angreifbar machen können, wenn die Implementierung schlecht oder fehlerhaft ist. Dadurch wird nebenbei die Paketgröße reduziert.

```
Anfrage: GetConfSaveZugang
Antwort: 1

Anfrage: SetLogin 1234 vf10 234234
Antwort: login set

Anfrage: CheckPIN 11
Antwort: logged in

Anfrage: GetNews
```

```
Antwort: Bankbetreuer User'Sehr geehrter KontoinhaberIn,
 aufgrund von schoenem Wetter haben wir beschlossen die Zinsen zu senken.
     Wir danke fuer Ihr Vertrauen.

 Ihr Bankbetreuer'Bankbetreuer User'Testnachricht2'Bankbetreuer
     User'Testnachricht3'

Anfrage: GetFinanz
Antwort: 234234 2000
123123 1000
```

Abb. 41. Beispiel einer Kommunikation (Login, News und Kontoübersicht; Prototyp)

9.8.14 Fehlende Funktionen

Im Vergleich zum Entwurf wurden nicht alle Funktionen und Sicherheitsmechanismen umgesetzt. Bei den Bankfunktionen fehlt vor allem die Wertpapierverwaltung.

Bei den Sicherheitsmechanismen konnten aufgrund von fehlendem Equipment nur Funktionalitäten auf der Clientseite umgesetzt werden. Aufbau und Konfiguration des Rechenzentrums wurde nicht ausgeführt. Die cardTAN wurde nicht umgesetzt, da zur Sicherung der aktuellen e-Banking Portale der TAN Algorithmus geheim gehalten wird.

Während der Entwicklung wurde für die TLS Verschlüsselung ein selbst signiertes Zertifikat verwendet. Durch die Verwendung eines eigenen Trustmanagers konnte die Kontrolle mit dem Android CertStore umgangen werden.

10. Vergleich des Entwurfs mit ELBA Internet

Dieses Kapitel beschäftigt sich mit dem Vergleich des in Kapitel „9 Entwurf für ein Singlebank e-Banking System“ vorgestellten Entwurfs zur Implementierung eines e-Banking Systems unter Android zu einem bestehenden System. Als Referenzsystem wird ELBA Internet verwendet, da es gut ausgereift ist und breite Akzeptanz besitzt.

Verglichen werden einerseits der Umfang der Funktionen und andererseits die Sicherheit in Form der implementierten Sicherheitsmechanismen.

10.1 Funktionsumfang (Bankfunktionen)

Der Funktionsumfang des Entwurfs ist ähnlich umfassend wie der von ELBA Internet. Letzteres stellt jedoch noch mehr Möglichkeiten im Bereich der Auswertung des Kontostandverlaufs und der Kontoverwaltung bereit. Das ist vor allem dadurch gegeben, dass es auf leistungsstärkeren Computern ausgeführt wird und mehr Platz für die Anzeige gegeben ist. Außerdem kann ein Benutzer vor einem Computer leichter mehr Zeit in die Verwaltung des Kontos investieren als am Smartphone.

Im Bereich der Unterstützung unterschiedlicher Bankprodukte ist ELBA Internet weiter entwickelt als der Prototyp. Das Anlegen und Verwalten von bankspezifischen Produkten wie Fixzinskonten ist im Entwurf nicht vorgesehen.

10.2 Sicherheitsmechanismen

Die verwendeten Sicherheitsmechanismen sind sehr ähnlich. Vor allem im Bereich der Server gibt es keine Unterschiede. Die Sicherung des Clients und der Applikation ist aufgrund der unterschiedlichen Betriebssysteme und davon ausgehenden Möglichkeiten verschieden. ELBA Internet ist eine Webanwendung und vertraut darauf, dass die Verbindungsdaten und die PIN geheim gehalten werden. Alternativ kann hier auch ein Clientzertifikat in Form einer Digitalen Signatur verwendet werden.

Beim Prototypen muss aufgrund der Mobilität des Endgerätes mehr Rücksicht auf gespeicherte Daten gelegt werden. Die Daten, die im Telefonspeicher abgelegt werden, sind verschlüsselt und können nur mit dem Touchmuster entsperrt werden. Eine Identifikation mittels IMSI und IMEI ist vorgesehen, um nicht nur den Benutzer sondern auch das Gerät erkennen zu können.

10.3 Evaluierung durch den Benutzer

Die Benutzerakzeptanz wird derzeit hauptsächlich durch die Umständlichkeit des Touchmusters beeinträchtigt. Auf mobilen Geräten ist der Benutzer es gewöhnt, mit Wischgesten die Funktionen zu bedienen. Außerdem ist das implementierte Muster optisch nicht ansprechen. Als Verbesserung wurde die Verwendung eines Bildmusters vorgeschlagen. Dieses ermöglicht es dem Anwender, ein eigenes Foto in den Hintergrund zu laden und auf diesem Punkte zu markieren. Durch diese Punkte wird das Passwort beschrieben.

Ein weiterer Kritikpunkt war die Art und Weise der Datumseingaben in den Eingabemasken „Kontodetails“ und „Überweisungserstellung“ (vgl. Abb. 34 Kontodetails (Prototyp), Abb. 35 Überweisung erstellen (Prototyp)). Durch die großen Auswahlfelder wird die Übersichtlichkeit stark beeinträchtigt.

Störend wurde weiters empfunden, dass durch den Zurück-Button des Gerätes die APP verlassen, die Verbindung getrennt und somit ein Logout durchgeführt wurde. Das ist jedoch durch die Architektur der APP gegeben, da diese aus einer Activity besteht. Weiters tritt dieses Problem nicht bei allen Geräten auf, da die Hardware Buttons (Home, Menu, Back) in neueren Android-Versionen nicht mehr existieren, sondern von den APPs selber in Form einer Action Bar (vgl. Seite 142, 9.8.12) geliefert werden sollten.

11. Zusammenfassung

Der Stellenwert von Mobilität im alltäglichen Leben wird immer höher. Von den Benutzern unterschiedlicher Software wird gefordert, dass sie möglichst einfach und überall zu bedienen ist. Ihnen ist auch die Sicherheit wichtig, jedoch werden Gedanken an diese oft erst nach dem ersten Verlust von Daten oder Eigentum gestellt.

Android ist ein sehr junges Betriebssystem, das für mobile Geräte wie Smartphones und Tablets entwickelt wurde. Im Vordergrund der Entwicklung stand, dass das System von jedem möglichst einfach verwendet werden kann, und es Möglichkeiten gibt, dass jeder seine eigene Software problemlos installieren kann.

Android basiert vom Kernel her auf Linux und verwendet viele Sicherheitsmechanismen, die von diesem System gegeben sind. Jedoch sind viele davon leicht zu umgehen und können somit das Gerät und auch die Laufzeitumgebung der Software gefährden.

Aufgrund der steigenden Nachfrage nach Mobilität ist es auch wichtig, Möglichkeiten für e-Banking für mobile Geräte anzubieten. Dadurch werden sehr hohe Sicherheitsanforderungen an diese, die Software und die Übertragungswege gestellt. Das Empfinden, was Sicherheit ist und wie sicher ein System sein soll, ist subjektiv. Manche Menschen würden bestimmte Anwendungen nie nutzen, weil sie ihnen zu unsicher und angreifbar erscheinen, während diese für andere zum Alltag gehören.

Derzeit gibt es bereits von einigen österreichischen Banken Implementierungen von eigenen e-Banking Anwendungen für das Android Betriebssystem. Diese verwenden großteils dieselben Sicherheitsmechanismen wie die Appliaktionen für den Heim-PC. Mit Ausnahme des kleineren Displays wird auf die speziellen Anforderungen der mobilen Geräte wenig Rücksicht genommen (vgl Abb. 42 Vergleich e-Banking Systeme Österreich - Funktionalität)

Die Kombination von Android und e-Banking bietet viele Angriffsmöglichkeiten. Es ist sehr wichtig, dass diese Gefährdungen eingedämmt und verhindert werden. Vollständige Sicherheit ist nicht möglich und würde außerdem die Anwendung unbenutzbar machen. Es muss ein Mittelweg zwischen Sicherheit und Benutzbarkeit gefunden werden, um eine gute Benutzerakzeptanz zu erreichen. Durch den eigenen Entwurf eines e-Banking Systems und der teilweisen Entwicklung von diesem war es mir möglich zu zeigen, wie ein derartiges System aussehen kann.

12. Erkenntnisse und Ausblick

Sowohl e-Banking als auch Android waren mir schon vor der Arbeit bekannt und ich hatte auch beide schon mehrmals als Anwender benutzt. Die Sicherheitsaspekte habe ich dabei jedoch nie im Detail betrachtet und hinterfragt. Durch die nähere Beschäftigung mit diesem Thema bin ich auf einige für mit interessante Erkenntnisse gestoßen:

12.1 E-Banking ist mehr als nur ein Bankzugang

Wenn jemand von e-Banking spricht, meint er meistens ein Webportal, das die Verwaltung des Kontos vom Heim-PC ermöglicht. In vielen Fällen wird es genutzt, um zu kontrollieren, wieviel Geld verfügbar ist und um Überweisungen durchzuführen. Es wirkt für den Endanwender wie eine einfache Homepage.

Durch die Arbeit wurde mir jedoch bewusst, dass einerseits hinter dieser Seite viel mehr Funktionen stecken als erwartet. Andererseits ist es auch möglich, Wertpapierdepots zu verwalten und Kredite anzuzeigen. Neben dem Zugang über den Browser gibt es noch ein eigenes Programm (ELBA Business), das hauptsächlich von Firmen genutzt wird. Es ermöglicht den Zugriff auf mehrere Banken und kann somit sämtliche Konten des Unternehmens der meisten Geldinstitute Österreichs ansprechen. Die Kommunikation erfolgt in diesem Fall über ein eigenes Protokoll, das eigens für die Übertragung von Daten mit einem Bankrechner entwickelt wurde.

12.2 Verwendung und Programmierung von Android

Das Betriebssystem Android ist meistens auf einem Handy installiert. Es bietet viele Funktionen. Dass die Funktionalität und Rechenleistung beinahe einem Computer entspricht, ist vielen nicht bewusst, wodurch auch der Sicherheitsbedarf unterschätzt wird. Die rasante Entwicklung geht oft auf Kosten einzelner als unwichtig empfundener Details. Mir sind bei Android einige aufgefallen, die dringend einer Nachbesserung bedürfen. Ein Beispiel hierfür sind fehlende Bibliotheken für das Wischmuster und Gesichtserkennung sowie fehlende Möglichkeiten um die Integrität und Authentizität einer APP zu prüfen.

12.2.1 Activities und Java 1.6

Die Activities sind für die Anzeige und Durchführung einzelner Aufgaben einer APP unter Android zuständig. In den meisten Fällen besteht eine Applikation aus mehr als einer Activity. Um Daten von einer zur nächsten weiterzugeben, gibt es eigene Funktionen. Diese können jedoch nur einfache und serialisierbare Datentypen verarbeiten. Wird zB ein Socket verwendet, müssen alle Aufgaben, die diese Verbindung benötigen, in einer Activity ausgeführt werden.

Weiters stellt die Verwendung von Java 1.6 Hindernisse dar. Normaler Java Code kann nicht einfach eingebunden werden, da die Version, die vom Android SDK verstanden wird, nicht die neueste ist. Manche Funktionen sind nicht verfügbar und teilweise können ganze Bibliotheken nur durch Verändern eingebunden werden.

12.2.2 Der Android CertStore

Android verwendet einen internen Zertifikatsspeicher. Es ist nicht möglich, neue Zertifikate zu diesem Speicher hinzuzufügen. Wenn man versucht, zu einer Seite mit einem Zertifikat zu verbinden, deren Ursprung nicht verifiziert werden kann, wird der Zugriff von Android geblockt. Dieses Sicherheitsfeature verhindert jedoch teilweise den Zugriff auf echte zertifizierte Seiten, da ein veraltetes oder kein Zertifikat einer RootCA im Speicher ist. Dieses Problem könnte behoben werden, wenn auch für alte Android-Versionen Updates von den Herstellern zur Verfügung gestellt werden würden.

Die Verwendung eines eigenen CertStores würde dieses Problem umgehen. Jedoch ist das kompliziert, da dieser mit der APP ausgeliefert und bei Änderungen erneuert werden muss. Weiters ist er durch die Auslieferung mit der APP angreifbar.

12.2.3 Fehlende Bibliotheken

In Android werden viele Funktionen vom System verwendet. Ein Beispiel ist das Wischmuster. Will man dieses nun jedoch in einer eigenen APP verwenden, muss man es selber schreiben, da keine vordefinierten Bibliotheken verfügbar sind. Dasselbe gilt auch für die ActionBar, die durch den Wegfall des Menübuttons auf den Geräten ab Version 3 wichtiger wird. Im Internet können nur Beispielprogramme gefunden werden, wie diese Funktion implementiert werden kann.

12.3 Unbekannte Gefährdungen und Schwachstellen

Während der Arbeit ist mir aufgefallen, dass e-Banking unter Android vielen Einflüssen ausgesetzt ist und somit auch unbekannte Gefahren auf das System lauern. Erst durch die Aufteilung des Systems in kleinere Teile werden potentielle Schwachstellen ersichtlich.

Es gibt zwar für einen Großteil der Gefährdungen passende Gegenmaßnahmen, jedoch rufen viele dieser Maßnahmen neue Gefahren hervor. Mir wurde dadurch bewusst, dass Sicherheit immer subjektiv ist und es keine vollständige Sicherheit geben kann. Außerdem werden durch die Erhöhung der Sicherheit durch zusätzliche Maßnahmen oder Vorschriften zwangsläufig die Benutzerfreundlichkeit und somit auch die Akzeptanz der Benutzer eingeschränkt.

12.4 Ausblick

Android ist ein sehr junges System und noch häufigen Neuentwicklungen unterworfen. Es ist wichtig, dass diese Neuerungen analysiert und in das Sicherheitskonzept eingebunden werden. In dieser Arbeit wurde ein Entwurf für eine Singlebank APP geliefert. Der Prototyp zeigt, dass es möglich ist, e-Banking bei guter Benutzbarkeit und Übersichtlichkeit möglichst sicher anbieten zu können. Für einen Einsatz im realen Betrieb müssen noch die fehlenden Funktionen hinzugefügt und der Server auf Belastbarkeit und Stabilität geprüft werden.

In Zukunft werden nicht nur Privatkunden ihre Bankgeschäfte über das Smartphone abwickeln wollen, sondern auch Firmen und deren Vorstände. Eine Entwicklung einer Multibank APP ist zukunftsweisend. Dazu müssen die Bibliotheken des Protokolls MBS/IP für Android vorbereitet und überprüft werden, inwiefern sich die Sicherheitsanforderungen von Privat- und Firmenkunden unterscheiden.

Mit steigender Verbreitung ist es auch wichtig, kontinuierlich das Sicherheitskonzept zu verbessern und weiter zu entwickeln. Ein wichtiger Punkt ist die Sicherstellung der Authentizität der APP. Wie kann gewährleistet werden, dass eine gefälschte APP möglichst schnell vom Server erkannt wird und somit der Zugang gesperrt wird. Gibt es in den zukünftigen Versionen von Android neue Möglichkeiten, die das ermöglichen?

Auch die Steigerung der Benutzerfreundlichkeit ist wichtig. Gibt es in Zukunft sichere Methoden der Gesichtserkennung auf Android, sodass eventuell auch die PIN durch einen Blick in die Kamera eingegeben werden kann. Da die mTAN eine bekannte Sicherheitslücke ist, wird die cardTAN verwendet. Diese hat den Nachteil, dass ein zusätzliches Gerät mitgeführt werden muss. Gibt es eine Möglichkeit, dass die TAN auch ohne zusätzliche Geräte sicher und mobil zu erstellen?

Eine letzte große Frage stellt sich noch: Ist es überhaupt notwendig Bankgeschäfte mobil zu tätigen? Sind die Unterschiede zwischen einem Smartphone und einem Notebook in ein paar Jahren noch so groß, dass diese aus Sicht der Sicherheit getrennt betrachtet werden müssen? Sollten nicht Notebooks auch schon als mobile Geräte angesehen werden?

IV. Quellen

[1] (2011, ZDNeT) ZDNet, Android Architektur: Wie viel Linux steckt in Googles OS? http://www.zdnet.de/magazin/41553061/android-architektur-wieviel-linux-steckt-in-googles-os.htm (aufgerufen am 9.1.2012)

[2] (2012, Wikipedia) Wikipedia, Dalvik (Software) http://en.wikipedia.org/wiki/Dalvik_(software) (aufgerufen am 9.1.2012)

[3] (2012, android.com) Android Developers, What is Android http://developer.android.com/guide/basics/what-is-android.html (aufgerufen am 9.1.2012)

[4] (2012, android.com) Android Developers, Designing for Security http://developer.android.com/guide/practices/security.html (aufgerufen am 9.1.2012)

[5] (2012, android.com) Android Developers, Services http://developer.android.com/guide/topics/fundamentals/services.html (aufgerufen am 11.1.2012

[6] (2006, F-Secure) F-Secure, Going around with Bluetooth in full safety http://www.securenetwork.it/ricerca/whitepaper/download/bluebag_brochure.pdf (aufgerufen am 11.1.2012)

[7] (2006, BSI) Bundesamt für Sicherheit in der Informationstechnik, Mobile Endgeräte und mobile Applikationen: Sicherheitsgefährdungen und Schutzmaßnahmen https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/Broschueren/MobilEndgeraete/mobile_endgeraete_pdf.pdf?__blob=publicationFile (aufgerufen am 11.1.2012)

[8] (2008, BSI) Bundesamt für Sicherheit in der Informationstechnik, Öffentliche Mobilfunknetze und ihre Sicherheitsaspekte https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/Broschueren/OeffentlMobilfunk/oefmobil_pdf.pdf?__blob=publicationFile (aufgerufen am 11.1.2012)

[9] (2009, BSI) Bundesamt für Sicherheit in der Informationstechnik, Drahtlose Kommunikationssysteme und ihre Sicherheitsaspekte https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/Broschueren/DrahtlosKom/drahtkom_pdf.pdf?__blob=publicationFile (aufgerufen am 11.1.2012)

[10] (2012, BSI) Bundesamt für Sicherheit in der Informationstechnik, Mobile Banking https://www.bsi-fuer-

buerger.de/BSIFB/DE/MobileSicherheit/MobileBanking/mobileBanking_node.html (aufgerufen am 11.1.2012)

[11] (2011, Heise) Heise Security, Cookie-Klau durch Lücken im Android-Browser http://www.heise.de/security/meldung/Cookie-Klau-durch-Luecken-im-Android-Browser-1318495.html (aufgerufen am 18.1.2012)

[12] (2011, Heise) Heise Security, Sicherheitslücken durch vorinstallierte Android-Apps http://www.heise.de/security/meldung/Sicherheitsluecken-durch-vorinstallierte-Android-Apps-1389329.html (aufgerufen am 18.1.2012)

[13] (2011, Heise) Heise Security, Forscher demonstriert Schwächen des Android-Rechtesystems http://www.heise.de/security/meldung/Forscher-demonstriert-Schwaechen-des-Android-Rechtesystems-1399337.html (aufgerufen am 18.1.2012)

[14] (2011, RLBOOE) Raiffeisen Landesbank OÖ, Die Zukunft - die cardTAN http://www.rlbooe.at/eBusiness/rlbooe_template2/15752112992436962-580769275027790860-626698178384610185-NA-25-NA.html (aufgerufen am 18.1.2012)

[15] (2011, RACON) RACON Software GmbH Linz, ELBA-mobil Testumgebung https://rbgtest.elbi.webapps.local/mobil/login.wf?channel=Mobile (aufgerufen am 18.1.2012 für Screenshots)

[16] (2011, Heise) Heise Security, CA-Hack: Noch mehr falsche Zertifiakte http://www.heise.de/security/meldung/CA-Hack-Noch-mehr-falsche-Zertifikate-1334098.html (aufgerufen am 25.1.2012)

[17] (2006, BSI) Bundesamt für Sicherheit in der Informationstechnik, G 5.131 SQL-Injection https://www.bsi.bund.de/ContentBSI/grundschutz/kataloge/g/g05/g05131.html (aufgerufen am 25.1.2012)

[18] (2011, Raiffeisen) Raiffeisen Bankengruppe OÖ, Das Sicherheitssystem von Raiffeisen ELBA-internet https://banking.raiffeisen.at/html/german/info/sicherheit.xhtml;jsessionid=00003SZsn8RbSyrXJH2QHFeVL-Z:15ond35ba (aufgerufen am 25.1.2012)

[19] (2011, UniCredit) Bank Austria, Bank Austria Mobile Banking https://market.android.com/details?id=com.bankaustria.android.olb&feature=related_apps#?t=W251bGwsMSwxLDEwOSwiY29tLmJhbmthdXN0cmlhLmFuZHJvaWQub2xiIl0. (aufgerufen im Android Market am 30.1.2012)

[20] (2011, PSK) BAWAG und PSK, e-Banking App im Android Market https://market.android.com/details?id=at.bawag.mbanking&feature=search_result#?t=W251bGwsMSwyLDEsImF0LmJhd2FnLm1iYW5raW5nIl0. (aufgerufen im Android Market am 30.1.2012)

[21] (2011, Erste Group) Erste Group Bank AG, Erste Bank und Sparkassen Android BankAPP https://market.android.com/details?id=com.erstegroup&feature=search_result#?t=W251bGwsMSwxLDEsImNvbS5lcnN0ZWdyb3VwIl0 (aufgerufen im Android Market am 30.1.2012)

[22] (2009, Becker) Arno Becker, Marcus Pant, Android Grundlagen der Programmierung, Auflage 1 ISBN 978-3-89864-574-4 http://dpunkt.de/buecher/3436.html (heruntergeladen am 6.2.2012)

[23] (2011, Heise) Heise Security, Android 4.0: Entsperrung durch Gesichtserkennung ist nicht sicher http://www.heise.de/security/meldung/Android-4-0-Entsperrung-durch-Gesichtserkennung-ist-nicht-sicher-1378934.html (aufgerufen am 13.2.2012)

[24] (2007, W3C) W3C, SOAP Version 1.2 Part 1: Messaging Framework (Second Edition) http://www.w3.org/TR/soap12-part1/ (aufgerufen am 14.2.2012)

[25] (2004, Oracle) Oracle Inc., Overview of Secure Sockets Layer (SSL) in Oracle Application Server http://docs.oracle.com/cd/B15897_01/core.1012/b13995/ssl_intro.htm (aufgerufen am 14.2.2012)

[26] (2012, GRZ) GRZ IT Center Linz GmbH, MBS/IP - hochsicherer Datentransfer http://www.mbsip.com/ (aufgerufen am 15.2.2012)

[27] (2012, Wikipedia) Wikipedia Foundation Inc, Multi-porpose Business Security over IP http://de.wikipedia.org/wiki/Multi-purpose_Business_Security_over_IP (aufgerufen am 15.2.2012)

[28] (2012, Google) Google Inc, Sourcen für Android dargestellt über github.com https://github.com/android/platform_frameworks_base (aufgerufen am 12.03.2012)

[29] (2011, MobiFlip) MobiFlip.de, Android Permissions - Apps und ihre Berechtigungen http://www.mobiflip.de/android-permissions-apps-und-ihre-berechtigungen/ (aufgerufen am 29.02.2012)

[30] (2010, BlogSpot) Blogspot.de, SOAP on Android http://android-devblog.blogspot.com/2010/06/soap-on-android.html (aufgerufen am 7.3.2012)

[31] (2008, OIO) Orientation in Objects, REST Web Services http://www.oio.de/public/xml/rest-webservices.htm (aufgerufen am 7.3.2012)

[32] (2010, ZDNeT) ZDNet, Kritische Sicherheitslücke im Apache Webserver entdeckt http://www.zdnet.de/news/41528652/kritische-sicherheitsluecke-in-apache-webserver-entdeckt.htm (aufgerufen am 14.3.2012)

[33] (2012, Google) Google Inc., Android Open Source Project - Licenses http://source.android.com/source/licenses.html (aufgerufen am 19.3.2012)

[34] (2005, Shinder) Deb Shinder – WindowsSecurity.com, SSL Acceleration and Offloading: What are the security implications? http://www.windowsecurity.com/articles/SSL-Acceleration-Offloading-Security-Implications.html (aufgerufen am 21.3.2012)

[35] (2002, Heise) Heise Security, Sicherheitslücke im Apache Webserver http://www.heise.de/newsticker/meldung/Sicherheitsluecke-im-Apache-Webserver-65448.html (aufgerufen am 11.4.2012)

[36] (2012, Heise) Heise Security, Phishing per NFC http://www.heise.de/security/meldung/Phishing-per-NFC-1446774.html (aufgerufen am 25.4.2012)

[37] (2012, android.com) Android Developers, ActionBarCompat - Action Bar Compatibility http://developer.android.com/resources/samples/ActionBarCompat/index.html (aufgerufen am 14.5.2012)

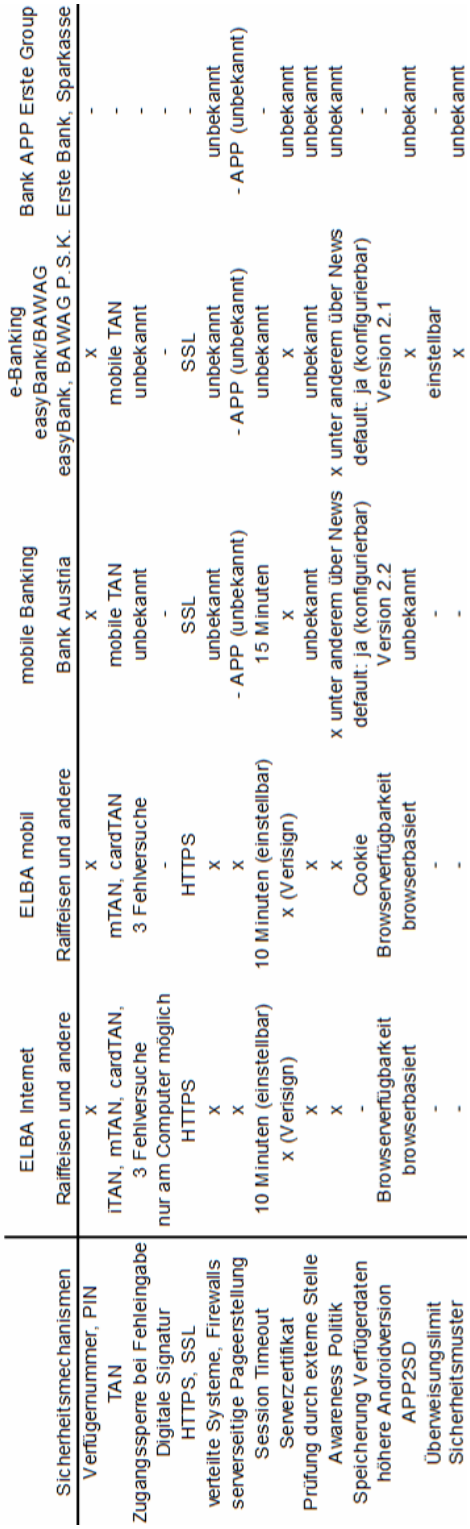
V. Bilderverzeichnis

VI. Codefragmente

VII. Anhang

Funktionen	Bank	ELBA Internet	ELBA mobil	mobile Banking	e-Banking	Bank APP Erste Group
		Raiffeisen und andere	Raiffeisen und andere	Bank Austria	easyBank, BAWAG P.S.K.	Erste Bank, Sparkasse
Allgemeine Funktionen	Bankomatfinder	-	basiert auf GPS	-	basiert auf GPS	basiert auf GPS
	Filialfinder	-	basiert auf Benutzereingaben	-	basiert auf GPS	basiert auf GPS
	Kartensperre	-	Telefonnummernsammlung	über Kontaktfunktion	Telefonnummernsammlung	-
	Kontaktinformationen	-	über Filialensuche	Menüpunkt	über Filialensuche	über Filialensuche
	Mail (Messaging)	ELBA intern	ELBA intern	-	internes Mailing	-
	News / Neuigkeiten	Weiterleitung nach Logout	als Weblink im Menü	Menüpunkt	Menüpunkt	möglich
	Aktienempfehlungen	Weiterleitung nach Logout	-	-	-	möglich
	Kreditrechner	Weiterleitung nach Logout	-	-	-	möglich
	Wechselkurse	Weiterleitung nach Logout	-	-	-	möglich
	Prognosen	Weiterleitung nach Logout	-	-	-	möglich
Konto-funktionen	Finanzstatus	Alle Konten	Alle Konten	Konten und Kreditkarten	Alle Konten	-
	Kontoübersicht	Konten mit Details	Konten mit Details	unbekannt	Anzeige des Kontostands	-
	Inlandsüberweisung	möglich	möglich	möglich	möglich	-
	SEPA Überweisung	möglich	möglich	möglich	nicht möglich	-
	Auslandsüberweisung	möglich	möglich	möglich	nicht möglich	-
	Dauerauftrag	möglich	nicht möglich	nicht möglich	nicht möglich	-
	TAN Verfahren	iTAN, mTAN, cardTAN,	mTAN, cardTAN	mobile TAN	mobile TAN	-
Depot-funktionen	Depotübersicht	Alle Depots mit Informationen	Alle Depots mit Informationen	möglich	-	-
	Suchen	möglich	möglich	unbekannt	-	-
	Kaufen	möglich	möglich	möglich	-	-
	Verkaufen	möglich	möglich	möglich	-	-
	Orderbuch	möglich	möglich	unbekannt	-	-

Abb. 42. Vergleich e-Banking Systeme Österreich - Funktionalität



Sicherheitsmechanismen	ELBA Internet Raiffeisen und andere	ELBA mobil Raiffeisen und andere	mobile Banking Bank Austria	e-Banking easyBank/BAWAG easyBank, BAWAG P.S.K.	Bank APP Erste Group Erste Bank, Sparkasse
Verfügernummer, PIN	x	x	x	x	-
TAN	iTAN, mTAN, cardTAN,	mTAN, cardTAN	mobile TAN	mobile TAN	-
Zugangssperre bei Fehleingabe	3 Fehlversuche	3 Fehlversuche	unbekannt	unbekannt	-
Digitale Signatur	nur am Computer möglich	-	-	-	-
HTTPS, SSL	HTTPS	HTTPS	SSL	SSL	-
verteilte Systeme, Firewalls	x	x	unbekannt	unbekannt	unbekannt
serverseitige Pageerstellung	x	x	- APP (unbekannt)	- APP (unbekannt)	- APP (unbekannt)
Session Timeout	10 Minuten (einstellbar)	10 Minuten (einstellbar)	15 Minuten	unbekannt	-
Serverzertifikat	x (Verisign)	x (Verisign)	x	x	unbekannt
Prüfung durch externe Stelle	x	x	unbekannt	unbekannt	unbekannt
Awareness Politik	x	x	x unter anderem über News	x unter anderem über News	unbekannt
Speicherung Verfügerdaten	-	Cookie	default: ja (konfigurierbar)	default: ja (konfigurierbar)	-
höhere Androidversion	Browserverfügbarkeit	Browserverfügbarkeit	Version 2.2	Version 2.1	-
APP2SD	browserbasiert	browserbasiert	unbekannt	x	unbekannt
Überweisungslimit	-	-	-	einstellbar	-
Sicherheitsmuster	-	-	-	x	unbekannt

Abb. 43. Vergleich e-Banking Systeme Österreich – Sicherheitsmechanismen

benötigte Berechtigungen	ELBA Internet, ELBA mobil Raiffeisen	mobile Banking Bank Austria	e-Banking easyBank / BAWAG easyBank, BAWAG P.S.K.	Bank APP Erste Group Erste Bank, Sparkasse
Internetzugriff	browserbasiert	x	x	x
Genauer (GPS-) Standort	browserbasiert	-	x	x
Laufende Anwendungen abrufen	browserbasiert	-	x	-
Telefonstatus lesen, identifizieren	browserbasiert	-	-	x
Telefonnummern direkt anrufen	browserbasiert	-	-	x

Abb. 44. Vergleich e-Banking Systeme Österreich – benötigte Berechtigungen

Gefährdung	APP/Bank	ELBA Internet	ELBA mobil	mobile Banking	e-Banking	Bank APP Erste Group
		Raiffeisen und andere	Raiffeisen und andere	Bank Austria	easyBank, BAWAG P.S.K.	Erste Bank, Sparkasse
Social Engineering (Endgerät)	Awareness Politik	Awareness Politik	Awareness Politik	Awareness Politik	Awareness Politik	-
Schadsoftware	- (browserbasiert)	- (browserbasiert)	- (browserbasiert)	-	unbekannt, Wischmuster	-
Reverse Engineering	- (serverseitige Pageerstellung)	- (serverseitige Pageerstellung)	- (serverseitige Pageerstellung)	unbekannt	kein APP2SD	-
Physischer Zugriff (Endgerät)	PIN, TAN, Sessions	PIN, TAN, Sessions	PIN, TAN, Sessions	PIN, TAN	PIN, TAN	-
Shoulder Surfing	zu kleines Display	angepasst -> problematisch	Anwendung an Display angepasst -> problematisch	Anwendung an Display angepasst -> problematisch	Anwendung an Display angepasst -> problematisch	Anwendung an Display angepasst -> problematisch
Hardwaremanipulation	Kontrolle nicht möglich. Hardware kann nicht von Software kontrolliert werden.					
Lauschangriff	HTTPS	HTTPS	HTTPS	SSL	SSL	-
Störsender	Störsender verhindern Kommunikation -> Wechsel des Kommunikationskanals - Meist nicht möglich					
DoS, dDoS (Endgerät)	Ein DoS Angriff auf das Funkinterface des mobilen Devices kann nicht durch die Software abgefangen werden.					
Positionsmessung	Positionsmessung ist ein Service das mobilen Gerätes -> Feature not Bug					
Man in the Middle (Internet)	HTTPS, Serverzertifikat	HTTPS, Serverzertifikat	HTTPS, Serverzertifikat	SSL, Serverzertifikat	SSL, Serverzertifikat	-
Zusätzlicher WLAN AP	HTTPS	HTTPS	HTTPS	SSL	SSL	-
IMSI Catcher	HTTPS	HTTPS	HTTPS	SSL	SSL	-
DNS Manipulation	HTTPS, Serverzertifikat	HTTPS, Serverzertifikat	HTTPS, Serverzertifikat	SSL, Serverzertifikat	SSL, Serverzertifikat	-
Proxy Manipulation	HTTPS, Serverzertifikat	HTTPS, Serverzertifikat	HTTPS, Serverzertifikat	SSL, Serverzertifikat	SSL, Serverzertifikat	-
DoS, dDoS (RZ)	Leistungsfähige Hardware			unbekannt	unbekannt	unbekannt
Tunnelzweckentfremdung	Firewallsysteme, Verteilte Systeme, definierte Zutrittssicherheitssysteme			unbekannt	unbekannt	unbekannt
Physischer Zugriff (RZ)				unbekannt	unbekannt	unbekannt
Social Engineering (RZ)	Awareness Politik	Awareness Politik	Awareness Politik	unbekannt	unbekannt	unbekannt
Softwarefehler, Konfigurationsfehler	Prüfung durch externe Stelle			unbekannt	unbekannt	unbekannt

Abb. 45. Vergleich e-Banking Systeme Österreich – Risiken und Maßnahmen

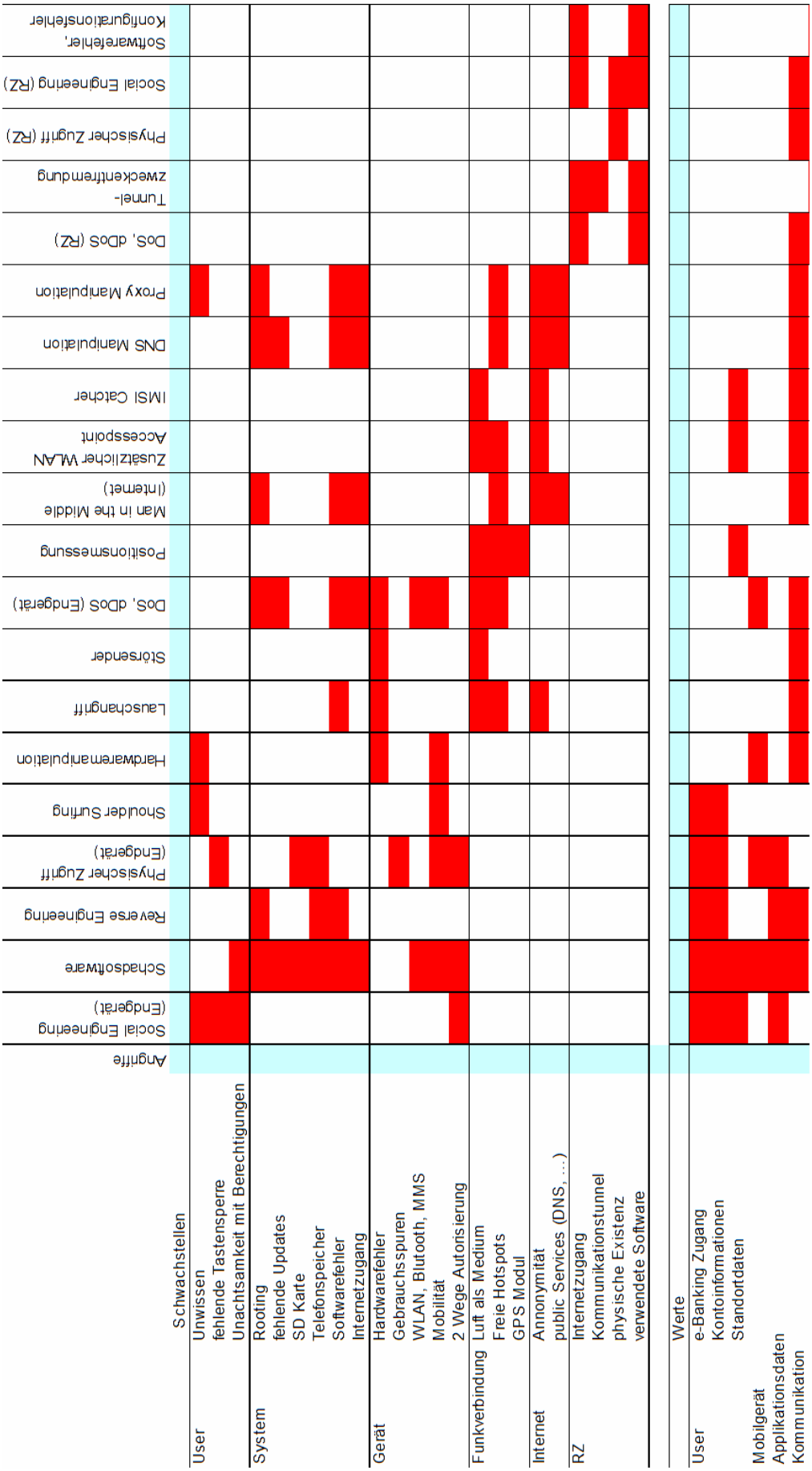


Abb. 46. Zusammenhang zwischen Gefährdung, Schwachstelle und Werte

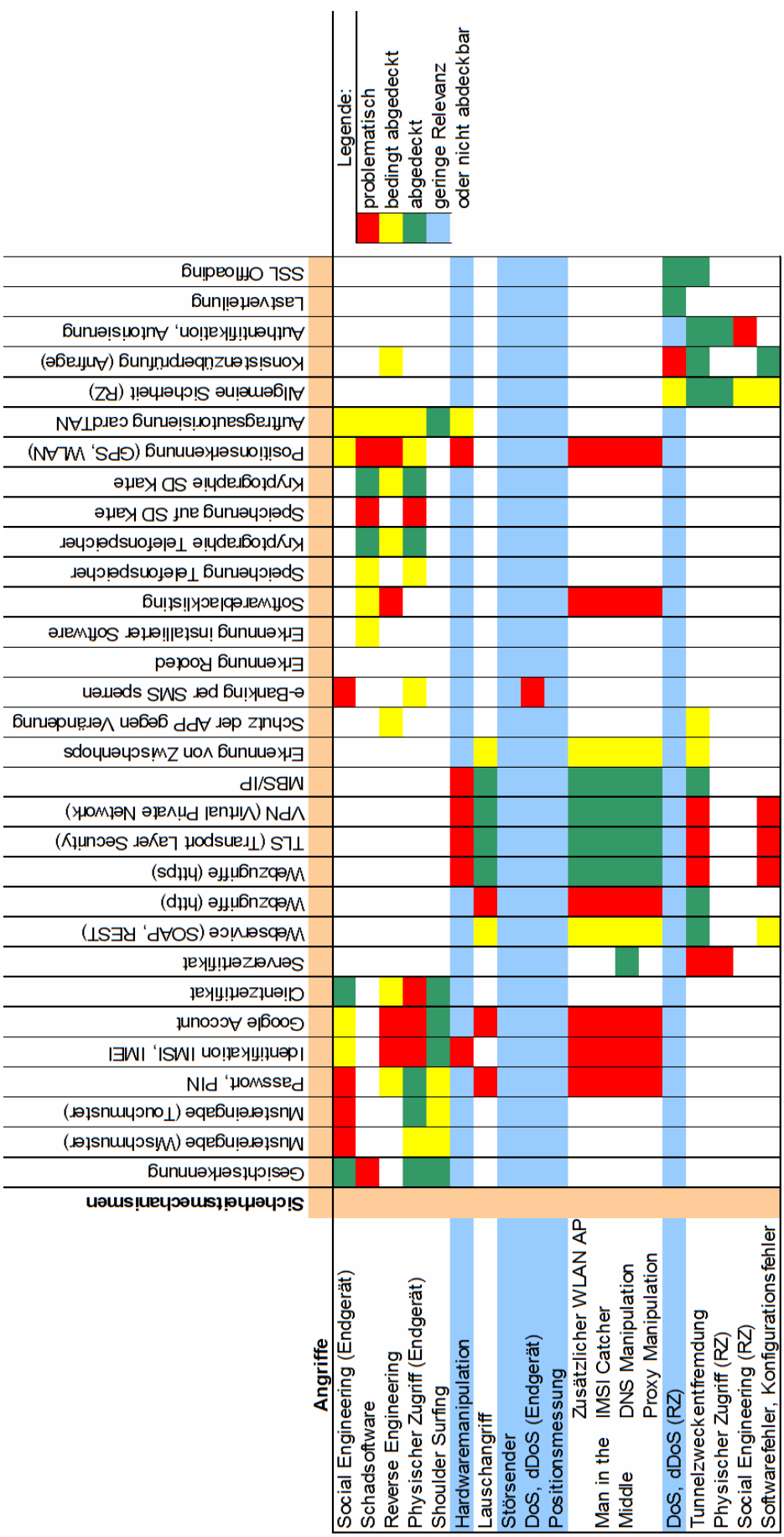


Abb. 47. Vergleich der Sicherheitsmechanismen in Bezug auf die Gefahren

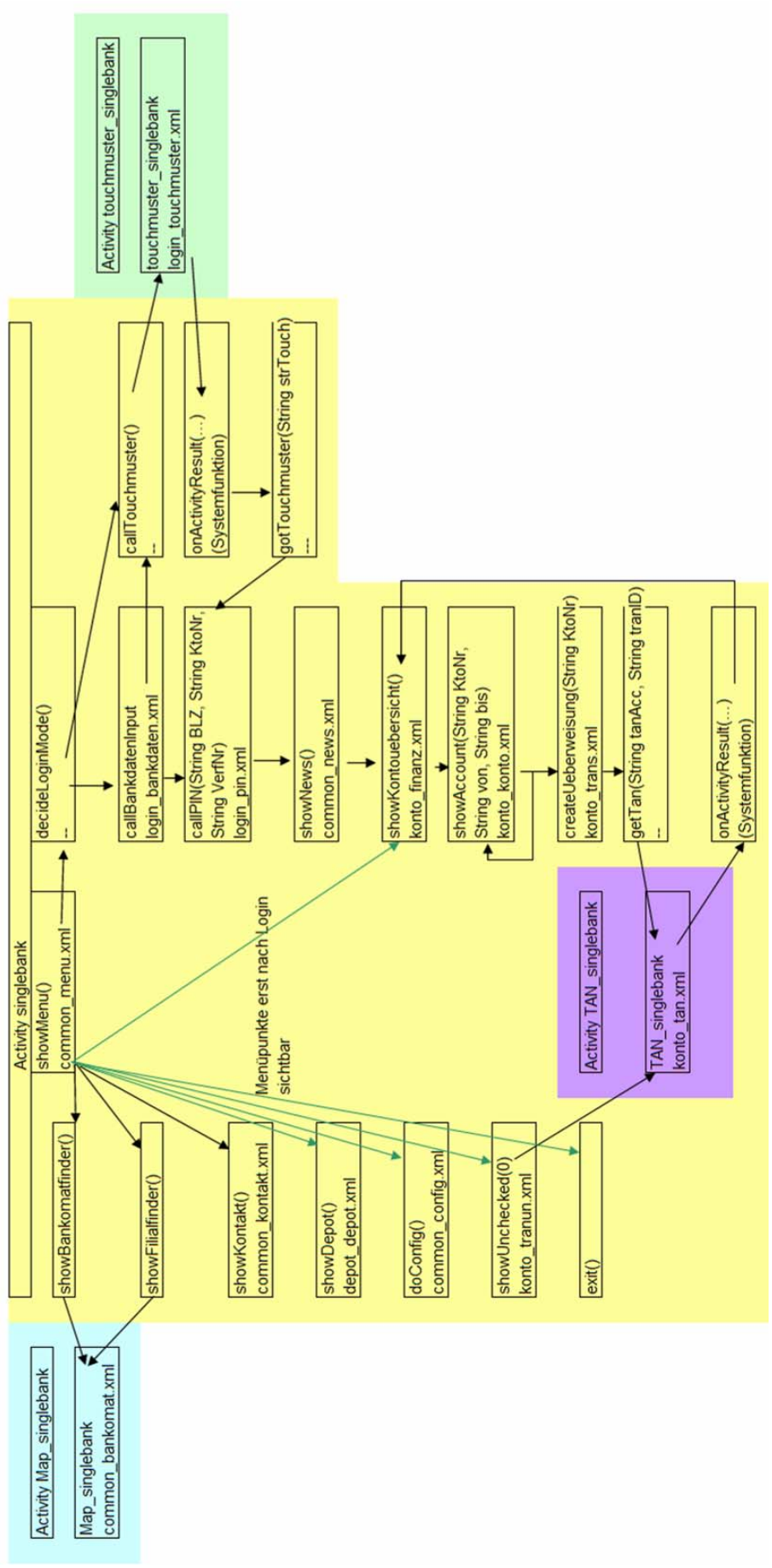


Abb. 48. Funktionen und Activities des Prototyps

VIII. Lebenslauf

Angaben zur Person	
Nachname(n) / Vorname(n)	Mayr Thomas
Adresse(n)	Vorstadt 16, 4840 Vöcklabruck, Österreich
Telefon	Mobil: +43 660 554 44 34
E-Mail	thomasmayr14@gmx.at
Staatsangehörigkeit	Österreich
Geburtsdatum	29.04.1987
Geschlecht	männlich
Berufserfahrung	
Daten	9.Februar 2010 bis 31.Mai 2010, August bis September 2010, Jänner bis Juli 2012
Beruf oder Funktion	Praktikant
Wichtigste Tätigkeiten und Zuständigkeiten	Schichtenübergreifendes Monitoring über die 7 Schichten des OSI- Netzwerkmodells. Ausarbeiten der Erkenntnisse in Form der Bachelorarbeit. Evaluieren und Implementieren eines Programmroboters. Masterarbeit Sicherheit in Rechenzentren in Bezug auf mobile Geräte
Name und Adresse des Arbeitgebers	LOGIS IT Service Gmbh, Goethestrasse 80, 4020 Linz
Tätigkeitsbereich oder Branche	IT
Daten	21.Juli 2009 bis 30.September 2009
Beruf oder Funktion	Praktikant
Wichtigste Tätigkeiten und Zuständigkeiten	Überlagerung des Microsoft Flugsimulator 2002 mit einem vom Matrox Helios Framegrabber eingelesenen Videostream
Name und Adresse des Arbeitgebers	EADS Defense and Security MEA15, Rechliner Straße, D-85077 Manching
Tätigkeitsbereich oder Branche	Luftfahrt
Daten	Mai 2007 bis August 2007 und August 2008
Beruf oder Funktion	Praktikant
Wichtigste Tätigkeiten und Zuständigkeiten	Programmierung von VBA Scripts in Excel, CAD (AutoCAD)
Name und Adresse des Arbeitgebers	Lenzing AG, Werkstraße 2, 4860 Lenzing
Tätigkeitsbereich oder Branche	Zellstofffasern
Daten	August 2005 bis Mai 2006
Beruf oder Funktion	Diplomarbeit für HTBLA Matura
Wichtigste Tätigkeiten und Zuständigkeiten	Erstellung einer Datenbank mit Visualisierung zur Speicherung von Daten von Extruderschnecken inklusive Zugriffsberechtigungen
Name und Adresse des Arbeitgebers	SML, Bundesstraße 1a, 4860 Lenzing
Tätigkeitsbereich oder Branche	Extruder für den Kunststoffbereich

Schul- und Berufsbildung

Daten	seit Wintersemester 2010
Bezeichnung der erworbenen Qualifikation	Informatiker (Dipl. Ing. bzw. MSc)
Hauptfächer/berufliche Fähigkeiten	Grundlagenfächer: Algorithmen und Datenstrukturen, Informationssysteme 2, Praktikum Softwareentwicklung 2, Formale Modelle Fachspezifische Fächer: Systemadministration, Einführung in die IT Sicherheit, Sicherheitsmanagement, IT Recht und Computerforensik, Netzwerkmanagement, Praktikum Netzwerke und Sicherheit (Loadbalancing)
Name und Art der Bildungs- oder Ausbildungseinrichtung	Johannes Kepler Universität Linz – Netzwerke und Sicherheit
Daten	Wintersemester 2007 bis Sommersemester 2010
Bezeichnung der erworbenen Qualifikation	Telematiker (BSc)
Hauptfächer/berufliche Fähigkeiten	Grundlagenfächer: Mathematik, Englisch Fachspezifische Fächer: Informatik, Netzwerkplanung und Konfiguration, Programmieren, Lichtwellenleitertechnik, Elektrotechnik, Microprozessortechnik, Automatisierungstechnik
Name und Art der Bildungs- oder Ausbildungseinrichtung	FH Kärnten - Telematik / Netzwerktechnik
Daten	2001 bis 2005
Bezeichnung der erworbenen Qualifikation	Betriebsinformatiker
Hauptfächer/berufliche Fähigkeiten	Grundlagenfächer: Mathematik, Deutsch, Englisch Fachspezifische Fächer: Informatik, Programm und Projektmanagement, Netzwerkkunde, Werkstoffkunde, Maschinenelemente, Konstruktionsübungen Werkstätte: Fräsen, Drehen, Bohren, NC, CNC, CAM, Elektronik,
Name und Art der Bildungs- oder Ausbildungseinrichtung	HTBLA Vöcklabruck Betriebsinformatik

Persönliche Fähigkeiten und Kompetenzen

Muttersprache(n)	Deutsch
Sonstige Sprache(n)	Englisch (First Certificate in English (ESOL) ausgestellt am 12.05.05), Spanisch, Russisch

Selbstbeurteilung	Verstehen				Sprechen				Schreiben	
Europäische Kompetenzstufe	Hören		Lesen		An Gesprächen teilnehmen		Zusammenhängendes Sprechen			
Englisch	B2	Selbstständige Sprachverwendung	B2	Selbstständige Sprachverwendung	B2	Selbstständige Sprachverwendung	B2	Selbstständige Sprachverwendung	B2	Selbstständige Sprachverwendung
Spanisch	A1	Elementare Sprachverwendung	A1	Elementare Sprachverwendung	A1	Elementare Sprachverwendung	A1	Elementare Sprachverwendung	A1	Elementare Sprachverwendung
Russisch	A1	Elementare Sprachverwendung	A1	Elementare Sprachverwendung	A1	Elementare Sprachverwendung	A1	Elementare Sprachverwendung	A1	Elementare Sprachverwendung

Soziale Fähigkeiten und Kompetenzen	Arbeiten im Bereich der Behindertenbetreuung (Zivildienst Lebenshilfe Tagesheimstätte2 Wels)
Organisatorische Fähigkeiten und Kompetenzen	Erhebung der Ist-Situation, Anforderungsanalyse, Umsetzung des Projektes, Arbeiten im Team (Durchführung der Bachelorarbeit, Diplomarbeit, Arbeiten an verschiedenen Projekten in Firmen)
Technische Fähigkeiten und Kompetenzen	Fundierte Kenntnisse im Bereich der Informatik (HTBLA, Studium) Grundkenntnisse aus dem Bereich Maschinenbau (HTBLA)
IKT-Kenntnisse und Kompetenzen	Netzwerkkentnisse: Cisco CCNA Semester 1 bis 4 abgeschlossen im Wintersemester 2009/10 Programmiersprachen: C, C++, C#, Java, PHP, SQL, Visual Basic
Sonstige Fähigkeiten und Kompetenzen	Segeln (Segelschein A), Modellbau, Latein
Führerschein(e)	Klasse B seit 06.07.2004

IX. Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Masterarbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt bzw die wörtlich oder sinngemäß entnommenen Stellen als solche kenntlich gemacht habe.

Die vorliegende Masterarbeit ist mit dem elektronisch übermittelten Textdokument identisch.