Bachelor Thesis

Sicherheit von E-Banking auf Smart-Platforms

Diese Thesis beinhaltet eine Analyse möglicher Sicherheitsprobleme bei der Verwendung von E-Banking auf mobilen Endgeräten. Ein mögliches Angriffsszenarios sowie eine Erst-Implementation zeigt das Angriffspotential in seiner ganzen Tragweite auf. Eine Übersicht möglicher Lösungsansätze sowie ein Ausblick runden die Arbeit ab.

Autoren: Simon Klaus (klaus1@bfh.ch), Danijel Brei (breid1@bfh.ch)
Betreuer: Prof. Reto Koenig (reto.koenig@bfh.ch)
Experte: Walter Stucki (walter.stucki.1@postfinance.ch)

Hochschule: Berner Fachhochschule, Fachbereich Informatik
Veröffentlicht: 18.01.2013

E-Banking

Einige Fakten:

- ↘Rasant wachsender Markt
- ↘Häufig genutzte Zahlungsmöglichkeit
- ↘Zahlreiche Anbieter
- ↘komfortabler Service
- ↘ rasante Entwicklung
- ↘Hohe Benutzerzahlen
- ↘Hohe Sicherheit (serverseitig)
- ↘Spannende Zukunft

Einführung

Vorwort

Mit dem rasanten Wachstum des Internets, folgen immer mehr nützliche Online-Dienstleistungen, welche das Leben bequemer und einfacher machen sollen.

Heute kann man sich das Geschäftsleben ohne E-Banking kaum mehr vorstellen. Fast jeder Finanzdienstleister bietet beispielsweise die Möglichkeit an, Zahlungen und Vermögensauszüge online zu tätigen. Der Markt ist rasant gewachsen und hat sich im fast selben Stil weiterentwickelt – doch was ist eigentlich E-Banking?

Egal ob auf PCs, Notebooks, Tablets oder Smartphones – die Bandbreite möglicher Endgeräte für die Zahlungsvorgänge wird immer grösser. Wie steht es dabei mit der Sicherheit beim Ausführen solcher Zahlungen? Ist diese in jedem Fall gewährleistet oder besteht ein Sicherheitsrisiko, falls man seine Zahlung auch unterwegs tätigen möchte?

Kann E-Banking, welches in der Regel über einen Desktop-PC oder ein Notebook getätigt und ein mobiles Endgerät zur Transaktionsbestätigung verwendet wird, durch mobiles E-Banking ersetzt werden? Welche Sicherheitsprobleme tauchen auf, wenn dadurch die Kanäle zusammenfallen: Wenn sowohl das Auslösen, als auch das Bestätigen einer Transaktion (beispielsweise per SMS mittels mTAN) auf demselben Gerät durchgeführt werden?

Wie könnte ein Angriff auf diese neue, mobile Form von E-Banking aussehen? Kann allenfalls auch das klassische E-Banking dadurch negativ beeinflusst werden?

Auf diese und weitere Fragen möchten wir im Rahmen dieser Bachelor Thesis genauer eingehen.

Ausgangslage

Das Modul 7302 (Projekt 2), welches die Grundlagen zur Sicherheit des E-Bankings und Probleme sowohl auf traditionellen Endgeräten (Computer, Notebooks, etc.) als auch auf mobilen Plattformen (Smart-Device: Smartphones, Tables, etc.) aufzeigte, dient als Basis für die Erarbeitung der Bachelor Thesis.

Im Rahmen dieser Arbeit wird ein E-Banking-System analysiert, welches die Identifizierung und Authentifizierung, sowie die Bestätigung jeder Transaktionen mittels alternativen GSM-Kanal (Mobiltelefon, mTAN) durchführt.

Abgrenzung

Allgemeine Abgrenzungen

Die Bachelor Thesis und die Teilprodukte werden wie folgt allgemein abgegrenzt:

- die entwickelten Teilprodukte enthalten jeweils ihre grundlegenden Funktionen (vordergründig und im Kontext des Angriffs)
- die entwickelten Teilprodukte sind nur für Testzwecke und nicht für den produktiven Einsatz gedacht
- die entwickelten Teilprodukte sowie deren Quellcode sind aus Sicherheitsgründen und zum Schutz der bestehenden Infrastruktur nicht zu veröffentlichen
- die entwickelten Teilprodukte sowie deren Quellcode stehen, nebst den Urhebern, der BFH-TI Bern/Biel für Schul- und Ausbildungszwecke zur Verfügung

Technische Abgrenzungen

Die Bachelor Thesis und die Teilprodukte werden wie folgt technisch abgegrenzt:

- auf zusätzliche Sicherheitsmechanismen (SSL, etc.), welche in diesem Fall keine Rolle spielen, wird verzichtet
- die entwickelten Teilprodukte zeigen die technischen Möglichkeiten auf, sollen aber nicht als final angesehen werden

Inhaltliche Abgrenzungen

Die Bachelor Thesis und die Teilprodukte werden wie folgt inhaltlich abgegrenzt:

- auf komplexe Diagramme und programmiertechnische Ausführungen wie beispielsweise Use Case in UML-Notation wird grösstenteils verzichtet, um die Thesis einer breiteren Leserschaft zugänglich zu machen
- der Inhalt baut stetig auf Vorhergehendes auf, Wiederholungen sind daher gewollt zur Auffrischung und als Übersicht gedacht
- die Komplexität des Inhalts soll vordergründig ein Publikum aus der Wirtschaft, insbesondere Personen aus dem Bankenumfeld, ansprechen
- der kommentierte Quellcode wird separat zur Thesis in digitaler Form ausgeliefert

Produkteinsatz

Anwendungsbereiche

Der Prototyp dieser Thesis besteht aus zwei Applikationen, welche unabhängig voneinander auf einem Smart-Device installiert werden können. Beide Applikationen täuschen reguläre Funktionen vor, ermöglichen jedoch im Zusammenspiel einen Angriff auf eine eigens erstelle E-Banking Web-Plattform.

Die Applikationen sowie die E-Banking-Plattform und auch die Command & Control-Plattform sind nicht für den „produktiven“ Einsatz bzw. die Öffentlichkeit gedacht, sondern sollen lediglich als Demonstration dienen.

Zielgruppen

Als primäre Zielgruppe werden Personen aus dem E-Banking-Bereich angesprochen, welche sich mit der Sicherheit der Systeme und Plattformen beschäftigen.

Die Benutzer von E-Banking-Applikationen auf Smart-Platforms werden als sekundäre Zielgruppe in Betracht gezogen. Dabei soll dieser Zielgruppe vor allem das Risiko eines Angriffs bei Benutzung beliebiger E-Banking-Applikationen auf Smart-Platforms aufgezeigt werden, egal welche softwarebasierte Sicherheitsmechanismen im Einsatz sind.

Produktübersicht

Beschreibung

Als Prototyp wurde ein System analysiert, dokumentiert und entwickelt, welches aus zwei Smart-Platform Applikationen, einer E-Banking Web-Plattform (pseudo E-Banking Server) und einer Command & Control-Plattform besteht. Jedes dieser vier Teilprodukte soll einerseits ihren bestimmten Zweck erfüllen und andererseits im Verbund der beiden Applikationen einen Angriff simulieren. Die E-Banking Web-Plattform simuliert lediglich die nötigen Dienstleistungen und ist am Angriff nicht direkt beteiligt. Die Command & Control-Plattform dient zur externen Steuerung des Angriffs. Dieser kann aber autonom durch die beiden Smart-Device Applikationen erfolgen.

Als Smart-Platform wurde Android gewählt. Der Angriff ist jedoch grundsätzlich auf jeder heute verfügbaren Smart-Platform möglich.

Doch zurück zur ersten Frage: Was genau versteht man unter E-Banking?

Definition von E-Banking

Unter E-Banking (Electronic-Banking) oder auch Online-Banking versteht man ein System, welches es erlaubt, online, unabhängig von Ort und Zeit, Bankgeschäfte über ein internetfähiges Endgerät zu tätigen.

Banktypen

Dabei spielt es keine Rolle, ob die Dienstleitung von einer Bank wie beispielsweise UBS, Raiffeisen, Migros-Bank, welche mit einem dichten Filialnetz aufwartet oder von einer virtuellen Bank, wie beispielsweise Swissquote, zur Verfügung gestellt wird.

Bankgeschäfte

Als Bankgeschäfte können Kontoeröffnungen, Zahlungsaufträge, Saldoabfragen aber auch Anträge für ein Darlehen, Hypothek, etc. verstanden werden.

Je nach Umfang der Dienstleistung, können auch Devisen- oder auch Wertschriftengeschäfte getätigt beziehungsweise ausgeführt werden.

Dabei spielt es keine Rolle, ob es sich um eine physisch existierende oder virtuelle Bank handelt, entscheidend ist dabei das Angebot, welches die Bank dem Benutzer der Dienstleistung zur Verfügung stellt.

Echtzeit

Die Dienstleistung erfolgt dabei quasi in Echtzeit. Das heisst: Bei der in Auslösung einer Zahlung, wird der geschuldete Betrag bei genügender Deckung auf dem Lastkonto direkt (beziehungsweise nach einer kurzen Verzögerung) abgezogen und auf dem Empfängerkonto nach einer bestimmten Verarbeitungszeit in der Grössenordnung von ein bis zwei Tagen, als eingehende Zahlung verbucht.

Entwicklung des elektronischen Zahlungsverkehrs

Online banking oder auch electronic banking (E-Banking) genannt, hat den Ursprung in den 80'er Jahren. Eine spannende Entwicklung, denn seither hat sich einiges getan:

Von Tonsignalen…

Der Term 'online' ist auf die verwendeten Medien zurückzuführen. Dabei wurde ein Terminal, eine Tastatur, ein Monitor und eine Telefonleitung benötigt, um mit dem Bankensystem kommunizieren zu können. Zu Beginn wurden Instruktionen mittels Tonsignalen über die Telefonleitung an die Bank gesendet.

…über Videotext-Banking…

Im Jahre 1981 waren es die vier grossen Banken/Geldinstitute Citibank, Chase Manhattan, Chemical und Manufacturers Hanover in New York, die einen Service mittels Videotext anboten. Da Videotext allerdings nicht an Popularität gewann, wurde es von den Systemen, welche über die Telefonleitung operieren, verdrängt.

…über Telefon-Banking…

Ein solches Telefon-Banking System wurde erstmals im Jahre 1983 von der Bank of Scotland angewendet. Das System basierte auf einem Computer, der an das Telefonnetz angeschlossen wurde, einer Tastatur und einem Bildschirm. Dieses System erlaubte es Transaktionen und Zahlungsaufträge online zu sichten. Für einen Zahlungsauftrag musste allerdings ein Schreiben mit dem jeweiligen Auftrag per Post zugesendet werden.

...hin zum heutigen E-Banking

Heutiges E-Banking basiert nicht mehr auf fixe Standorte wie Terminals, Computer, etc. Die heutigen Systeme sind unabhängig von Ort und Zeit und benötigen, nebst den Zugangsinformationen für die jeweilige Bank, lediglich ein Übertragungsmedium wie das Internet.

E-Bankingfähige Geräte

Zu den E-Bankingfähigen Geräten zählen Tablets, Netbooks, Smartphones, Notebooks und so weiter. Und wer weiss, vielleicht wird E-Banking schon bald via Fernseher im Wohnzimmer oder dem Kühlschrank in der Küche möglich sein.

Statistiken

E-Banking gehört für viele mittlerweile zum Alltag. Zwar hinkt die Schweiz im Vergleich zum umliegenden Ausland hinterher, der Trend zeigt aber auch hierzulande klar hin zu elektronischen Zahlungsverfahren. So benutzen fast genau 50% der Schweizer Internetzbenutzer (Stand 2010) das Internet auch für E-Banking. Gemessen auf die ganze Bevölkerung, steht Norwegen mit 83% an erster Stelle, gefolgt von den Niederlanden mit 77%. Die Schweiz ist mit nur rund 29% relativ tief in dieser Statistik vertreten.

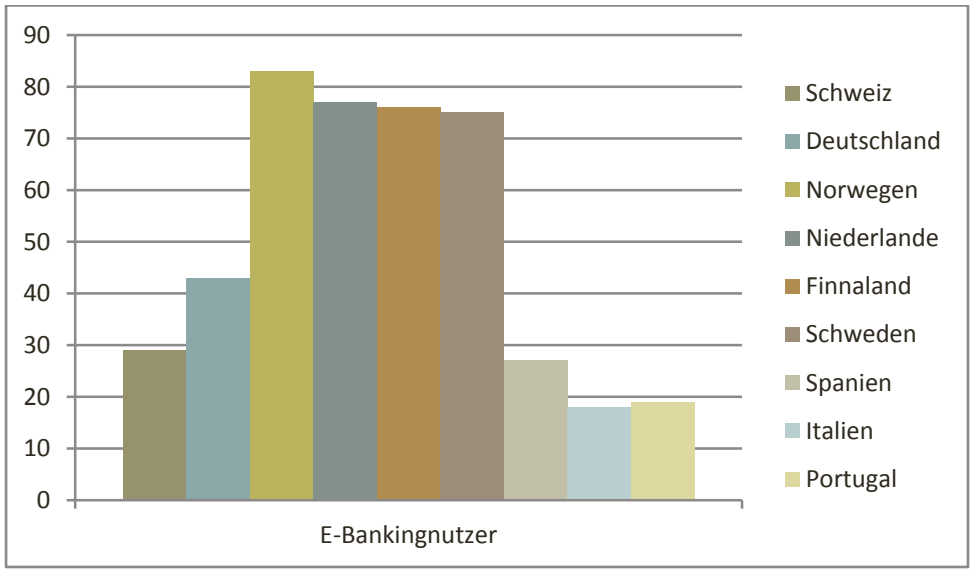

ABBILDUNG 1: E-BANKINGNUTZER (BUNDESAMT FÜR STATISTIK - INTERNETNUTZUNG)

E-Banking System abstrakt

Im Folgenden wird der grundlegende Aufbau eines E-Banking Systems und dessen Akteuren abstrakt aufgezeigt.

Abstrakte Beschreibung der Komponenten

Grundsätzlich werden für eine sogenannte E-Banking Session folgende Elemente benötigt:

- Bank
- Übertragungsmedium
- Computer/Terminal
- Benutzer

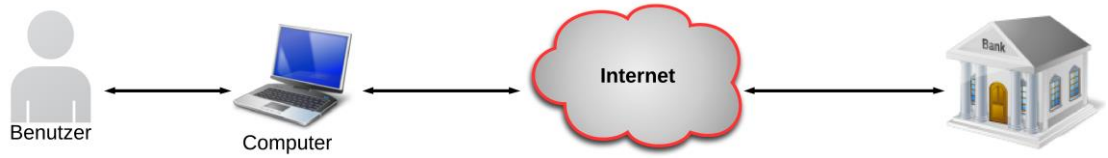

ABBILDUNG 2: E-BANKING ABSTRAKT

Im Folgenden werden die erwähnten Elemente näher erläutert.

Bank

Die Bank stellt die grundlegende Infrastruktur, sprich den E-Banking Server zur Verfügung.

Da sich die Bank ebenfalls um die Instandhaltung kümmert, kann davon ausgegangen werden, dass die E-Banking Server auf dem aktuellen Stand der Technik sind und regelmässig mit Softwareupdates versorg werden.

Grundsätzlich kann die Annahme getroffen werden, dass es ein Angreifer von aussen schwer haben dürfte, einen solchen Server unbemerkt in seine Gewalt bringen zu können. Somit kann angenommen werden, dass die Daten- und Zugriffsicherheit seitens der Bank gewährleistet ist.

ABBILDUNG 3: E-BANKING SERVER

Übertragungsmedium

Als Übertragungsmedium zwischen Computer/Terminal und Bank wird in dieser Arbeit das Internet, sowie das GSM-Netz verwendet. In den nachfolgenden Abbildungen wird das Internet, zur Vereinfachung, als hauptsächliches Übertragungsmedium betrachtet.

Das Internet sowie das GSM-Netz sind unkontrollierte Medien, sprich, weder der Benutzer, noch die Bank haben einen direkten Einfluss darauf. Dabei spielt es keine Rolle, welcher Zugangstyp verwendet wird.

Somit kann die Annahme getroffen werden, dass ein Angreifer die Möglichkeit besitzt, jeglichen Verkehr zu belauschen, mitzuschneiden und manipulierte Daten einzuschleusen. Beide Übertragungsmedien sind somit bei der Übertragung von heiklen Daten als unsicher anzusehen.

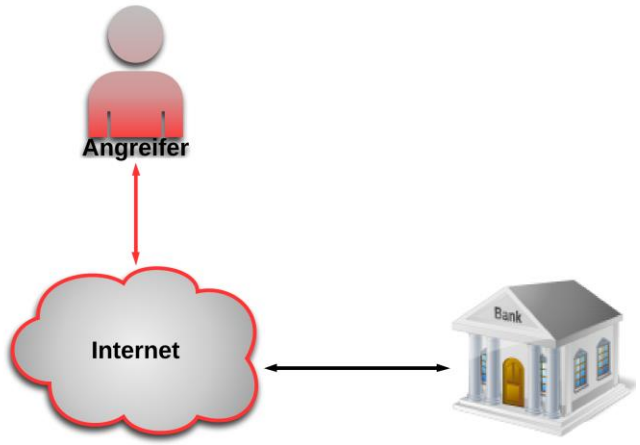

ABBILDUNG 4: ANGREIFER IM INTERNET

Computer/Terminal

Als Computer/Terminal wird hier das Endgerät bezeichnet.

Folgende Endgeräte können für die Nutzung von E-Banking zur Anwendung kommen:

- Smartphone, Tablet-PC (Smart-Device)
- Netbook, Notebook
- Desktop-PC
- etc.

User Agent

Im weiteren Verlauf wird hier der Begriff ***User Agent*** für jegliche Endgeräte sowie die darauf betriebene Software verwendet. Der User Agent stellt die Verbindung zum E-Banking Server über das Internet her.

Dabei kommt eine gesicherte Verbindung über das Internet zur Anwendung, das heisst, der Kanal zwischen dem User Agent und dem E-Banking Server, worüber die Transaktionen gesendet werden, ist als sicher zu betrachten.

Der User Agent befindet sich im Besitz des Benutzers. Es kann allerdings davon ausgegangen werden, dass nicht die aktuellste Software verwendet wird: Sei es beim Betriebssystem, Webbrowser oder auch dem Antiviren-Programm. Dies kann ein Sicherheitsrisiko darstellen, welches jedoch vielen Benutzern weder bewusst ist, noch abverlangt werden kann.

Dadurch kann die Annahme getroffen werden, dass der User Agent durch eine Drittperson (Angreifer) kontrolliert wird, durch Malware infiziert wurde beziehungsweise werden kann. Der Angreifer kann sich somit zwischen User Agent und Server aber auch auf dem User Agent selbst befinden.

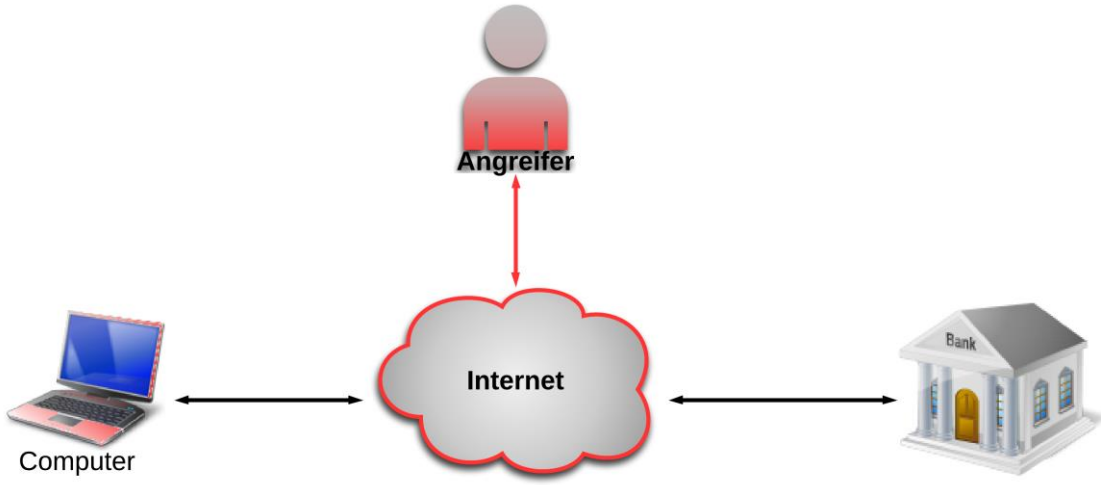

ABBILDUNG 5: ANGREIFER IM INTERNET & USER AGENT

Verbindungsaufbau User Agent zu E-Banking Server

Der Verbindungsaufbau zwischen User Agent und E-Banking Server erfolgt oft direkt oder indirekt, über den Webbrowser beziehungsweise dessen Protokolle.

Zertifikate

Beim Verbindungsaufbau vom User Agent zum E-Banking Server, sendet der Server dem User Agent ein signiertes, erweitertes Zertifikat zu. Auf Grund dieses Zertifikates und den darin enthaltenen Informationen, kann nun der User Agent die Identität des Servers verifizieren.

Authentifizierung

Nachdem der Benutzer seine Credentials wie Vertragsnummer, persönliches Kennwort, etc. eingegeben hat, muss eine sogenannte Challenge zur erfolgreichen Authentifizierung bewältigt werden (siehe auch Abschnitt „Authentisierung“). Diese Challenge kann beispielweise mit einer physikalisch vorhandenen Streichliste (Matrix-Karte), einem eigens dafür verwendeten Gerät (TAN) oder gar einem dedizierten Kanal über ein externes Gerät (USB-Token) gelöst werden. Einige Banken setzen diese Methoden nicht nur beim Verbindungsaufbau, sondern auch zur Bestätigung von Transaktionen ein.

Benutzer

Wie vorhergehend erwähnt, verwendet der Benutzer meist einen Webbrowser oder eine von der Bank zur Verfügung gestellte Software um E-Banking zu tätigen.

Dabei kommen in der Regel Tastatur, Maus und Bildschirm zur Interaktion zwischen dem Benutzer und dem User Agent zur Anwendung. Der Benutzer ist der Überzeugung, dass die Informationen, die ihm während einer E-Banking Session angezeigt werden, auch korrekt, konsistent und gültig sind.

Es muss davon ausgegangen werden, dass der Benutzer ein geringes bis gar kein Sicherheitsbewusstsein besitzt beziehungsweise, der Benutzer sich den Gefahren und dem damit verbundenen Sicherheitsrisiko aus dem Internet nicht bewusst ist.

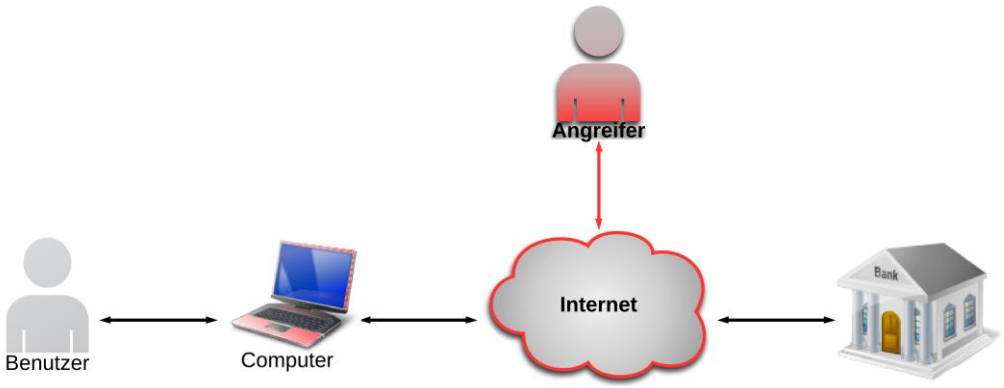

ABBILDUNG 6: KOMPROMITTIERTES SYSTEM

Angreifer

Hierbei handelt es sich um einen monolithischen Angreifer. Das heisst, man betrachtet in den aufgeführten Szenarios einen Angreifer, der sowohl das Internet, als auch den User Agent unter seine Kontrolle bringen kann.

Dabei ist zu beachten, falls der Angreifer die Kontrolle über den User Agent besitzt, dieser auch Einsicht auf die darauf gespeicherten Dateien erlangen kann und somit auch vorhandene private Schlüssel nicht mehr vor dem Angreifer sicher sind.

Absicht

Der Angreifer hat in diesen Szenarios immer finanzielle Absichten und agiert dazu oft in kriminellen Organisationen. Das Risiko bei einem digitalen Angriff erwischt zu werden ist vergleichsweise gering. Aufgrund der breiten Angriffsfläche wie beispielsweise eine Phishing-Attacke, ist auch mit kleinen Beträgen ein hoher Gewinn erreichbar.

Vorgehen bei Phishing

Das Ziel: Der Angreifer möchte Transaktionen im Namen des Benutzers auslösen lassen.

Hier ein möglicher Ablauf:

1. Der Angreifer versendet per E-Mail Schadsoftware an möglichst viele Empfänger.
2. Fällt ein Empfänger darauf rein und verfügt dieser über nicht ausreichende Sicherheitssoftware beziehungsweise aktuelle Software, kann der Angreifer Schadprogramme einschleusen und damit womöglich Zahlungen mitverfolgen sowie tätigen.
3. Oft werden zahlreiche kleinere Geldbeträge auf ein oder mehrere Durchlauf-Konten, welche sich meist im Ausland befinden überweisen.
4. Von diesen Durchlauf-Konten aus, wird das Geld schlussendlich von Strohmännern an den eigentlichen Auftraggeber übergeben, welcher so oft unbekannt bleibt.

Fazit

Aus dem schrittweise aufgezeigten, abstrakten E-Banking System haben sich Möglichkeiten herauskristallisiert, durch die ein Angreifer das System stören oder gar manipulieren kann.

Der E-Banking Server kann sich zwar mittels Zertifikat gegenüber dem User Agent des Benutzers authentisieren, eine Authentisierung des User Agent gegenüber dem Server findet jedoch oft nicht direkt statt.

Eine solche Authentisierung ist zwar zunächst auch nicht notwendig, da der User Agent lediglich sicherstellen will, mit welchem Server er nun tatsächlich kommuniziert und eine allfällige E-Banking Session aufbauen möchte.

Auf ein clientseitiges Zertifikat wird oft verzichtet. Von Seiten des Servers ist somit nur bekannt, dass jemand mit einer gültigen Vertragsnummer angemeldet ist, jedoch nicht, ob es sich dabei um den Bankkontobesitzer handelt.

Falls sich der Angreifer jedoch direkt auf dem User Agent befindet, würde auch ein clientseitiges Zertifikat keine Sicherheit bieten. In diesem Fall hat der Angreifer die volle Kontrolle über die Eingaben sowie die Anzeige über den User Agent.

Der Benutzer ist an die Glaubwürdigkeit des User Agents angewiesen, denn der Anwender kann zu keiner Zeit wissen, ob das System kompromittiert ist.

Secure Platform Problem

Um das grundsätzliche Problem in seiner ganzen Tragweite erfassen zu können, muss an dieser Stelle das „Secure Platform Problem“ kurz aufgenommen werden: (Schläper, 2012)

Allgemeines

Ideale Welt

In der idealen Welt kann davon ausgegangen werden, dass zwei Kommunikationspartner, nennen wir sie Alice und Bob, ungestört miteinander kommunizieren können. Dieser Austausch, ohne einen Dritten, kann direkt oder indirekt über ein elektronisches Gerät erfolgen.

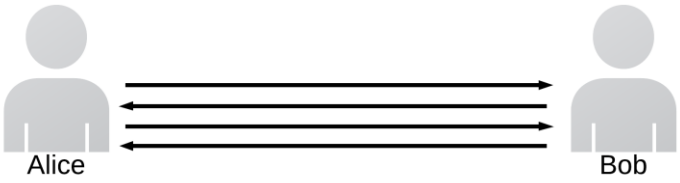

ABBILDUNG 7: KOMMUNIKATION IDEALE WELT

Gefährliche Welt

In der gefährlichen Welt muss davon ausgegangen werden, dass ein Dritter (der Angreifer) im Spiel ist. Dieser Dritte kann die Kommunikation abhören, beeinflussen oder auch unterbrechen, eine andere Identität annehmen, etc.

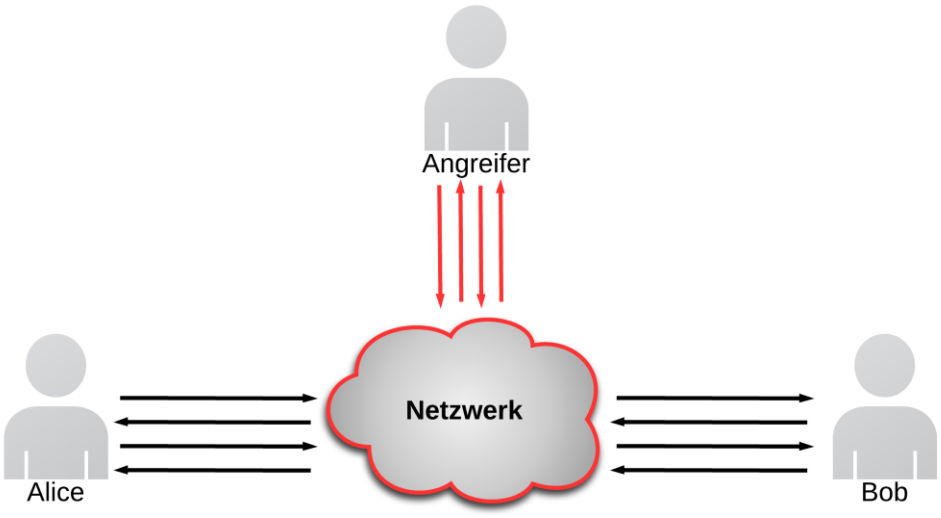

ABBILDUNG 8: KOMMUNIKATION GEFÄHRLICHE WELT

Problembeschreibung

Beim „Secure Platform Problem“ gehen wir davon aus, dass sich der Angreifer auf einem „nichtvertrauenswürdigen“ Endgerät oder auf dem dazwischenliegenden Kommunikationskanal befindet. Die Seite des Servers kann aus diesem Standpunkt als vertrauenswürdig und sicher angesehen werden. Beim Problem geht es somit um das Vertrauen zwischen zwei miteinander kommunizierenden Partnern.

Eine sichere Kommunikation kann entweder über:

- menschliche Kryptographie (bei welcher der Mensch die Verschlüsselung übernimmt, nicht benutzerfreundlich und nicht massentauglich),
- ein Code-Buch (nicht sehr benutzerfreundlich, dafür günstig, kann aber über unsicheren Kanal zum Kommunikationspartner gelangen) oder über
- ein zusätzliches Gerät

erfolgen. Letzteres muss:

- vertrauenswürdig sein und bestimmte, vertrauenswürdige Operationen anbieten
- individualisiert und personalisiert sein (beispielweise über persönlichen Schlüssel)
- eigenständig (beispielsweise Challenge/Response-Tools) oder
- verbunden sein (beispielsweise Secure-Token über das Endgerät)

Damit eine bidirektionale Kommunikation sichergestellt werden kann, muss:

- das vertrauenswürdige Device (Trust-base) Ein- & Ausgabeschnittstellen bereitstellen,
- der Benutzer die Nachrichten direkt an den Server senden und
- die Operationen des Servers zweckmässig überprüfen können

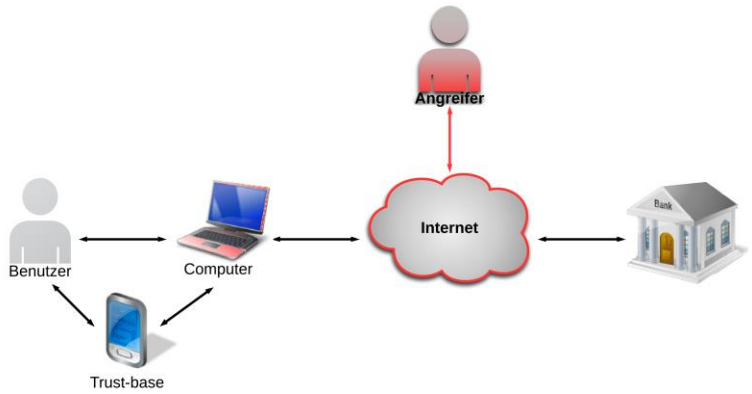

ABBILDUNG 9: SECURE PLATFORM

E-Banking System in der Praxis

Nachdem nun abstrakt aufgezeigt wurde, wie ein E-Banking System aufgebaut ist, welche Akteure involviert sind und worin das grundsätzliche Sicherheitsproblem liegt, soll nun der Ablauf von E-Banking Sessions veranschaulicht werden, wie sie in der Praxis anzutreffen sind.

Traditionelle E-Banking Session

Eine E-Banking Session wird im Normalfall durch den Benutzer beispielsweise per Desktop-PC explizit initiiert. Dabei kann im Allgemeinen von einem alltäglichen Bankgeschäft ohne Angreifer ausgegangen werden.

E-Banking Session ohne Angreifer

Zur Veranschaulichung soll hier der Ablauf einer normalen E-Banking Session ohne einen Dritten aufgezeigt werden. Dabei wird die Behandlung von Fehleingaben, fehlgeschlagenen Anmeldeversuchen, ungültigen Transaktionen, etc. nicht berücksichtigt.

Ablauf im Überblick

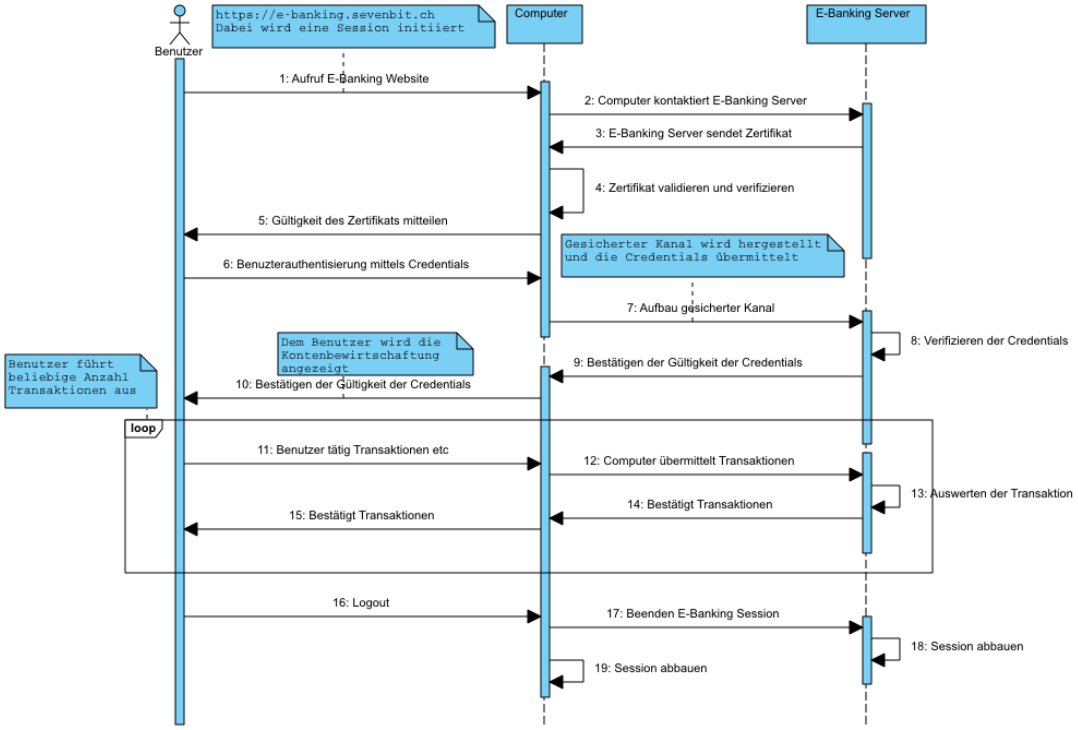

Abbildung 10: Ablauf E-banking Session ohne Angreifer

Ablauf im Detail

1. Der Benutzer gibt über die Tastatur die Web-Adresse der Bank ein, um eine E-Banking Session zu starten.
2. Der User Agent nimmt die Eingabe entgegen und ruft die eingegebene URL der jeweiligen E-Banking Web-Plattform auf.
3. Der Server sendet sein Zertifikat an den User Agent.
4. Der User Agent validiert und verifiziert das erhaltene Zertifikat.
5. Der User Agent teilt dem Benutzer via Webbrowser mit, dass das Zertifikat gültig ist. Dies geschieht meist mittels einer graphischen Ausgabe (beispielweise durch ein Schloss-Zeichen).
   Bei ungültigen oder selbstsignierten Zertifikaten wird eine Warnung ausgegeben.
6. Der Benutzer gibt seine Login-Daten (Credentials) wie beispielsweise Vertragsnummer, Kennwort, etc. ein und schickt diese ab.
7. Der User Agent erstellt nun eine gesicherte Verbindung und übermittelt die Login-Daten an den E-Banking Server.
8. Der E-Banking Server verifiziert und authentisiert den Benutzer.
9. War die Authentisierung erfolgreich, sendet der E-Banking Server eine Übersicht der Konten des Benutzers zurück.
10. Der User Agent zeigt dem Benutzer die Übersicht seiner Konten an, womit dem Benutzer bestätigt wird, dass die Login-Daten korrekt waren.
11. Der Benutzer tätigt nun beliebige Transaktionen.
12. Der User Agent leitet die Transaktionen an den E-Banking Server weiter.
13. Der E-Banking Server verifiziert die Transaktionen.
14. Eine entsprechende Meldung, dass die Transaktion ausgeführt wurde, wird vom E-Banking Server an den User Agent versendet.
15. Der User Agent wiederum informiert den Benutzer über die Ausführung der jeweiligen Transaktion.
    Die Schritte 11-15 können immer wieder durchgeführt werden.
16. Der Benutzer beendet mittels „Logout“ die E-Banking Session.
17. Der User Agent sendet an den E-Banking Server eine entsprechende Meldung, dass die E-Banking Session beendet werden soll.
18. Der E-Banking Server löst die Session mit dem User Agent auf, indem die erstellten Session-Informationen gelöscht werden.
19. Der User Agent löst die Session mit dem E-Banking Server auf, indem dieser das Session-Cookie vom System löscht.

E-Banking Session mit Angreifer im Internet

Der Angreifer befindet sich wie dargestellt im Internet und nicht direkt auf dem Endgerät. Dadurch ist er in der Lage, die Kommunikation zwischen Bank und dem User Agent abzuhören (jedoch nur bis der sichere Kanal aufgebaut ist) und diese zu beeinflussen.

Ablauf im Überblick

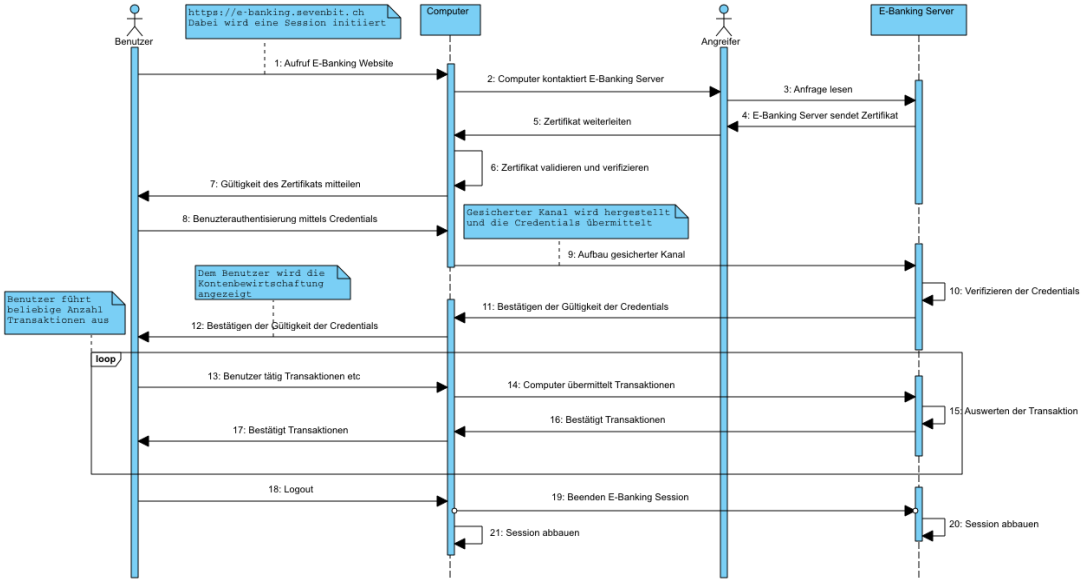

ABBILDUNG 11: E-BANKING SESSION MIT ANGREIFER IM INTERNET

Bevor der Angreifer aktiv in die E-Banking Session eingreifen kann, muss zuerst der Benutzer eine Session mit der Bank beziehungsweise mit dem E-Banking Server aufbauen, wie in der vorhergehenden Abbildung unter Punkt 2 aufgeführt. Der Angreifer lässt die Anfrage an den Server passieren und wartet nun auf die Antwort des Servers.

Der Server sendet sein Zertifikat an den Angreifer. Dieser sendet das Zertifikat an den User Agent weiter (Punkt 5). Konnte der User Agent die Gültigkeit und Echtheit verifizieren, wird (Punkt 9) ein gesicherter Kanal mit dem Server aufgebaut. Der Angreifer hat ab diesem Zeitpunkt lediglich noch die Möglichkeit, die Kommunikation zwischen Server und User Agent zu verzögern beziehungsweise zu unterbrechen. Da die Kommunikation zwischen User Agent und Server gesichert erfolgt, kann er keine Informationen herauslesen oder Schlüsse aus den vom Benutzer getätigten Transaktionen ziehen.

E-Banking Session mit Angreifer auf User Agent

In diesem Szenario konnte sich der Angreifer, zum Beispiel aufgrund einer Sicherheitslücke oder als Malware innerhalb einer gewollten Installation durch den Benutzer, den direkten Zugriff auf den User Agent verschaffen. Der Benutzer selbst hat davon keine Kenntnis und startet wie gewohnt eine E-Banking Session.

Ablauf im Überblick

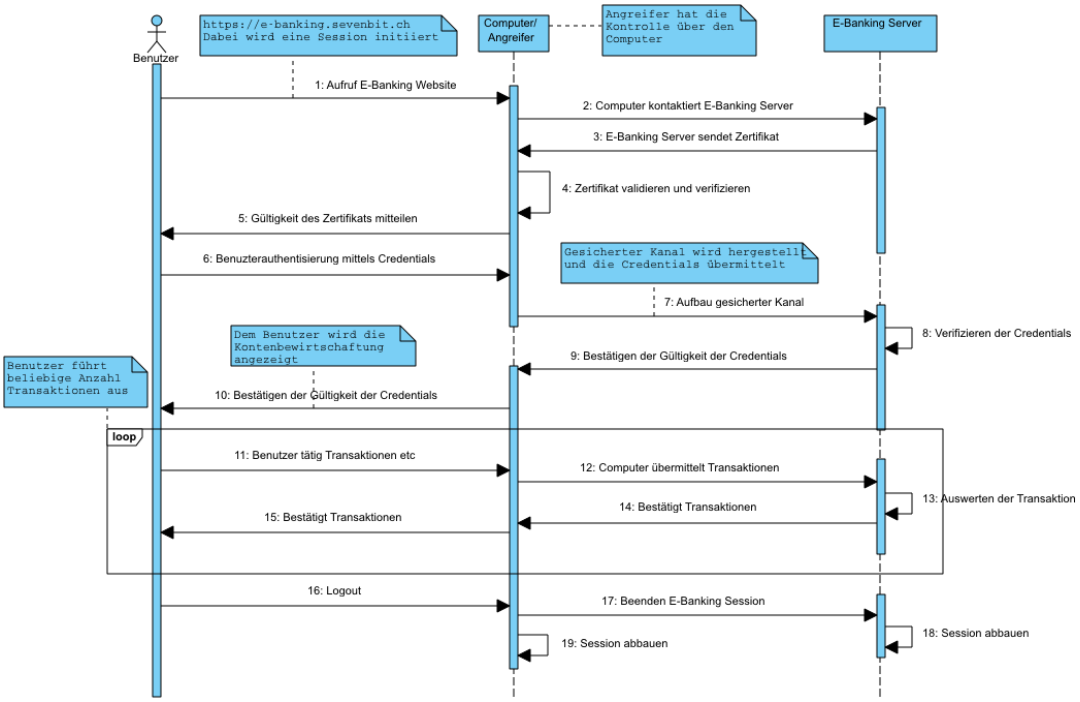

Abbildung 12: Ablauf E-Banking Session mit Angreifer auf User Agent

Der Angreifer kann alle Eingaben einsehen, die der Benutzer tätigt. Somit hat er nun die Möglichkeit passiv alle Vorgänge zu beobachten. Er kann aber auch aktiv werden und die Transaktionen, welche der Benutzer aufgibt, zu seinen Gunsten manipulieren.

Da es sich bei diesem Beispiel um den kritischsten und mächtigsten Fall eines Angriffs handelt und der Angreifer nun grossen Einfluss auf das Geschehen hat, wird in der weiteren Dokumentation dieser Typ von Angreifer beziehungsweise Angriff bevorzugt.

Authentisierung

Im E-Banking Bereich ist nicht zuletzt die korrekte Authentisierung des Benutzers essentiell. Daher soll nun dieser Begriff hier eingeführt werden: Unter Authentisierung versteht man den Vorgang, bei welchem sich ein Benutzer gegenüber dem E-Banking Server authentisiert und somit seine Identität übermittelt.

Der Server wiederum verifiziert diese behauptete Identität und authentifiziert den Benutzer entsprechend. Dadurch erhält der Benutzer Zugriff auf die ihm zugehörigen Daten – er erhält eine Autorisierung.

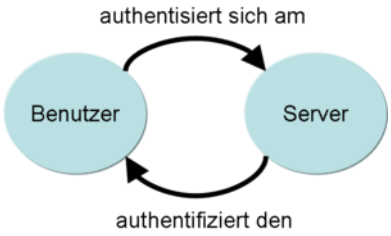

ABBILDUNG 13: AUTHENTIFIZIERUNG & AUTHENTISIERUNG (WIKIPEDIA - AUTHENTIFIZIERUNG)

Zur Authentisierung beziehungsweise als sogenannte Challenge werden im E-Banking Bereich verschiedene Methoden verwendet. Die gebräuchlichsten sind:

- Code-Karte (Matrix-Karte, TAN)
- alternativem GSM-Kanal (SMS-Token, mTAN)
- USB-Token (z.B. UBS Access Key)

Weiter gibt es folgende Verfahren, auf welche wir in dieser Arbeit nicht weiter eingehen werden. Unter anderem:

- Streichliste
- Dynamisches Token (z.B. SecureID)
- Challenge/Response-Tools (z.B. UBS Kobil)
- EMV Card Authentication Protocol
- AXSionics
- SSL Client-Certificate

Mittels Code-Karte (Matrix-Karte, TAN)

Hierbei erhält der Benutzer auf dem Postweg eine Code-Karte, welche für die E-Banking Nutzung erforderlich ist.

Die Codes auf der Karte werden von vielen E-Banking Systemen zur Anmeldung und auch zum Bestätigen einer Transaktion an einen neuen Empfänger oder sogar für jede ausgelöste Transaktion benötigt.

Dabei wird der Benutzer vom E-Banking Server zur Eingabe eines bestimmten Codes (auch Transaktionsnummer TAN genannt) aufgefordert. Die Abfrage der TAN erfolgt zufällig, was ein Erraten der nächsten TAN erschwert. Ein bereits verwendeter TAN verliert nach Eingabe oder einer gewissen Zeit seine Gültigkeit.

Es gibt E-Banking Systeme, welche ebenfalls mittels Code-Karte den Benutzer authentisieren und entsprechend für den Zugriff autorisiert. Im Folgenden wird jedoch nur die Bestätigung jeder Transaktion mittels TAN besprochen.

ABBILDUNG 14: CODE-KARTE IM EINSATZ (123RF)

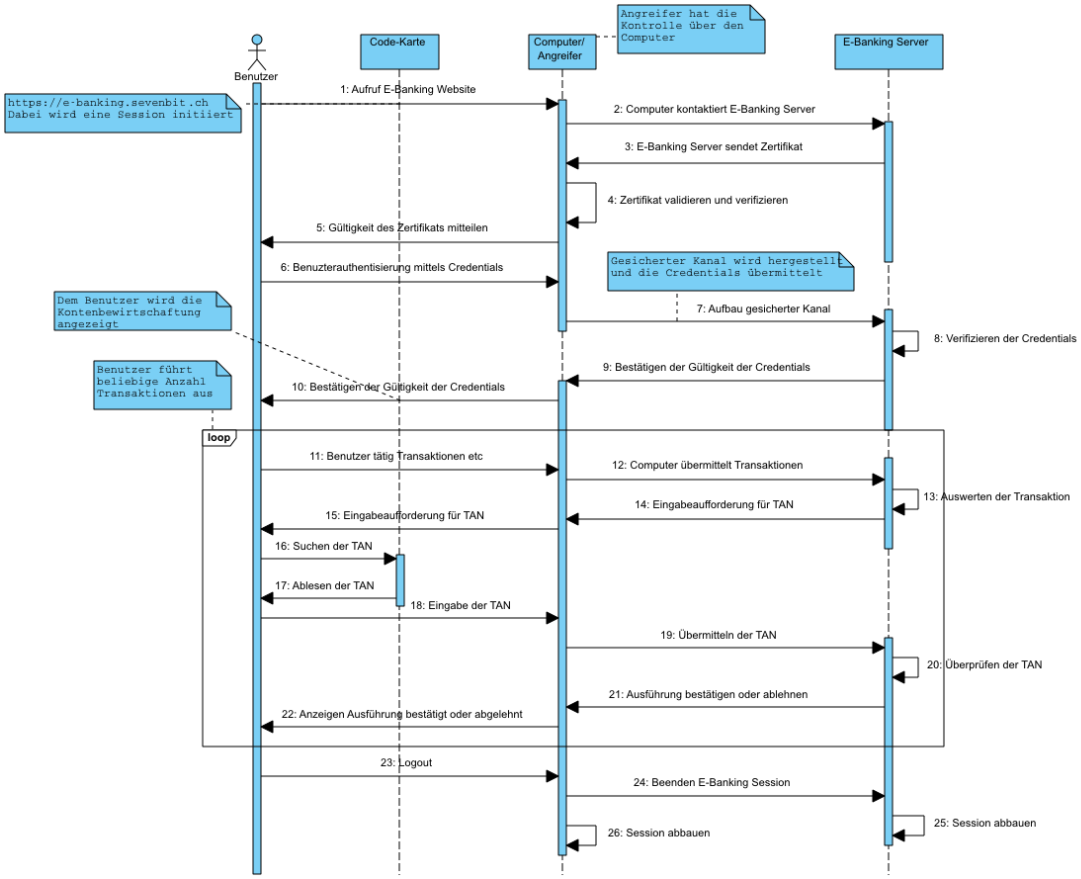

ABBILDUNG 15: E-BANKING SYSTEM MIT CODE-KARTE

Interessant wird es ab Punkt 11: Der Benutzer gibt hier die Daten zur jeweiligen Transaktion ein. Der Angreifer hat nun die Möglichkeit, die Eingaben des Benutzers zu verändern und die manipulierte Transaktion (Punkt 12) an den E-Banking Server zu übermitteln.

Der Server wird nun eine TAN von der Code-Karte des Benutzers abfragen (Punkt 14, 15). Der Benutzer wird die jeweiligen TAN am User Agent eingeben und so die Transaktion bestätigen (Punkt 16-19). Da der Angreifer auch die Möglichkeit besitzt, allenfalls die Anzeige am Bildschirm zu manipulieren, kann er dem Benutzer die gültigen Transaktionsdaten wie beispielsweise Kontonummer anzeigen lassen, obwohl diese im Hintergrund vom Angreifer verändert wurde.

Mittels alternativen GSM-Kanal (Mobiltelefon, mTAN)

Ein weiteres Verfahren: mTAN über einen alternativen GSM-Kanal. Bei diesem Verfahren handelt es sich um ein System, welches ohne physikalische Code-Karte auskommt.

Dabei wird ein mobiles Endgerät verwendet, welches per SMS eine TAN (mTAN, auch mobile TAN genannt) erhält, die der Benutzer anschliessend via User Agent dem E-Banking Server bestätigt.

Ablauf im Überblick

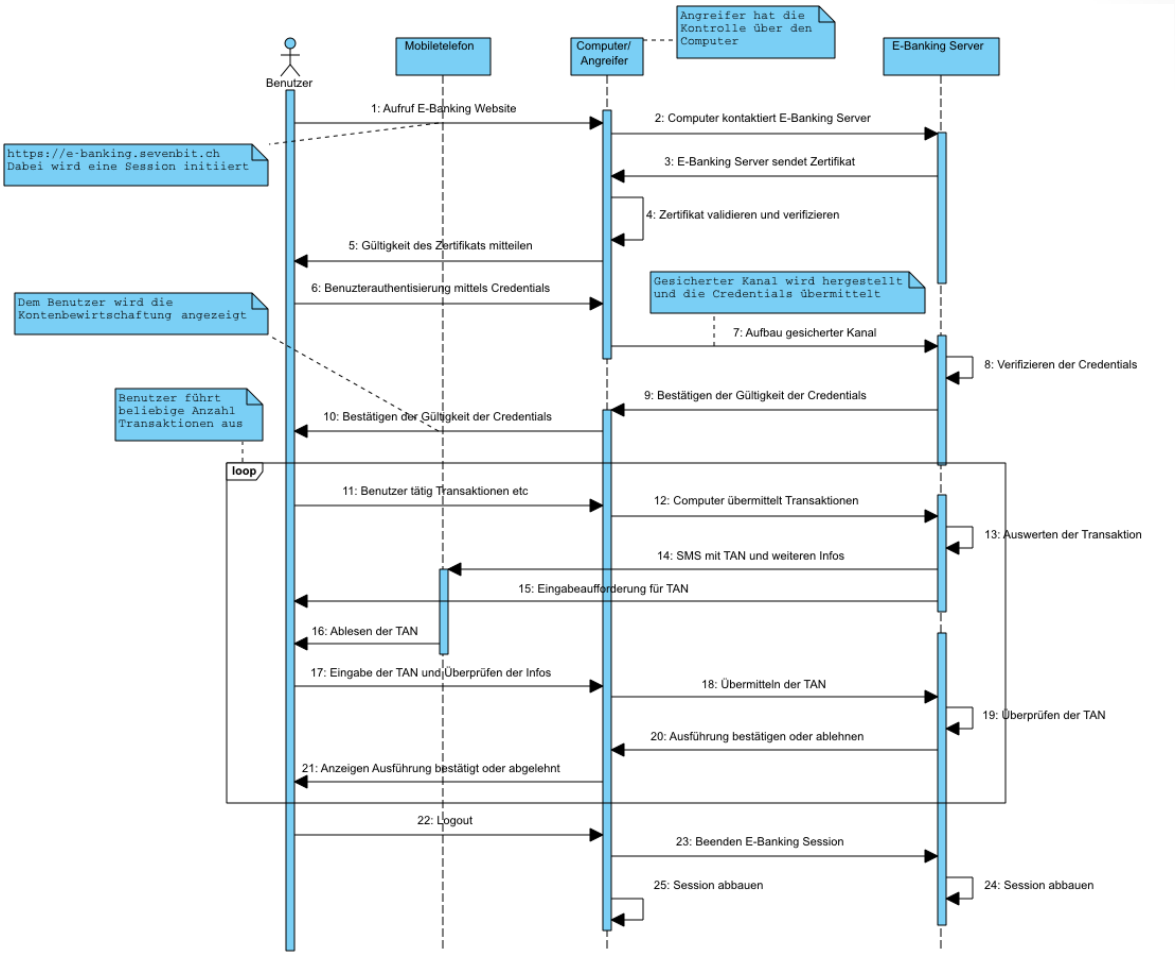

ABBILDUNG 16: E-BANKING SYSTEM MIT ALTERNATIVEM KANAL

Auch hier: Ein allfälliger Angreifer hat unter Punkt 12 die Möglichkeit, die Transaktion zu manipulieren. Allerdings erhält nun der Benutzer die TAN und weitere Informationen wie Kontonummer des Empfängers und den zu überweisenden Betrag über einen alternativen Kanal auf das mobile Endgerät.

Allfällige Manipulationen können so relativ einfach erkannt werden. Mittels dieses alternativen, dedizierten Kanals, hat der Angreifer keine direkte Möglichkeit mehr, eine manipulierte Transaktion einzuschleusen und durch den Benutzer bestätigen zu lassen.

Mittels USB-Token

Bei dieser Technik kommt ein USB-Token zur Anwendung, welcher mit dem User Agent verbunden wird. Ein sogenanntes Security-Token – in dem Fall ein USB-Token, da per USB-Schnittstelle angeschlossen – ist eine Hardwarekomponente, welche die Identifizierung und Authentifizierung eines Benutzers gegenüber dem E-Banking Server übernimmt.

Zur erweiterten Sicherheit können zusätzliche Authentifizierungsmerkmale wie Passwort, PIN oder Fingerabdruckt dem Token zugeordnet werden. Im Fall des USB-Tokens (hier der UBS Access Key) übernimmt dies die persönliche Smartcard (in diesem Fall die UBS Keycard). Das USB-Token selbst besitzt, bis auf wenige Ausnahmen, immer ein Display oder ein anderes Ausgabemedium, sowie eine oder mehre Tasten. Auf dem Ausgabemedium werden dabei die Transaktionsdaten dem Benutzer zur Bestätigung angezeigt. Der Benutzer kann nun über die Tasten am USB-Token die Transaktion bestätigen oder ablehnen.

In diesem Verfahren, baut das USB-Token eine sichere Verbindung zum E-Banking Server auf. Dabei werden die Daten in beide Kommunikationsrichtungen verschlüsselt und durch den User Agent hindurch geschleust. Eine Manipulation kann aus heutiger Sicht so ausgeschlossen werden.

ABBILDUNG 17: USB-TOKEN DER BANK UBS (UBS AG)

Ablauf im Überblick

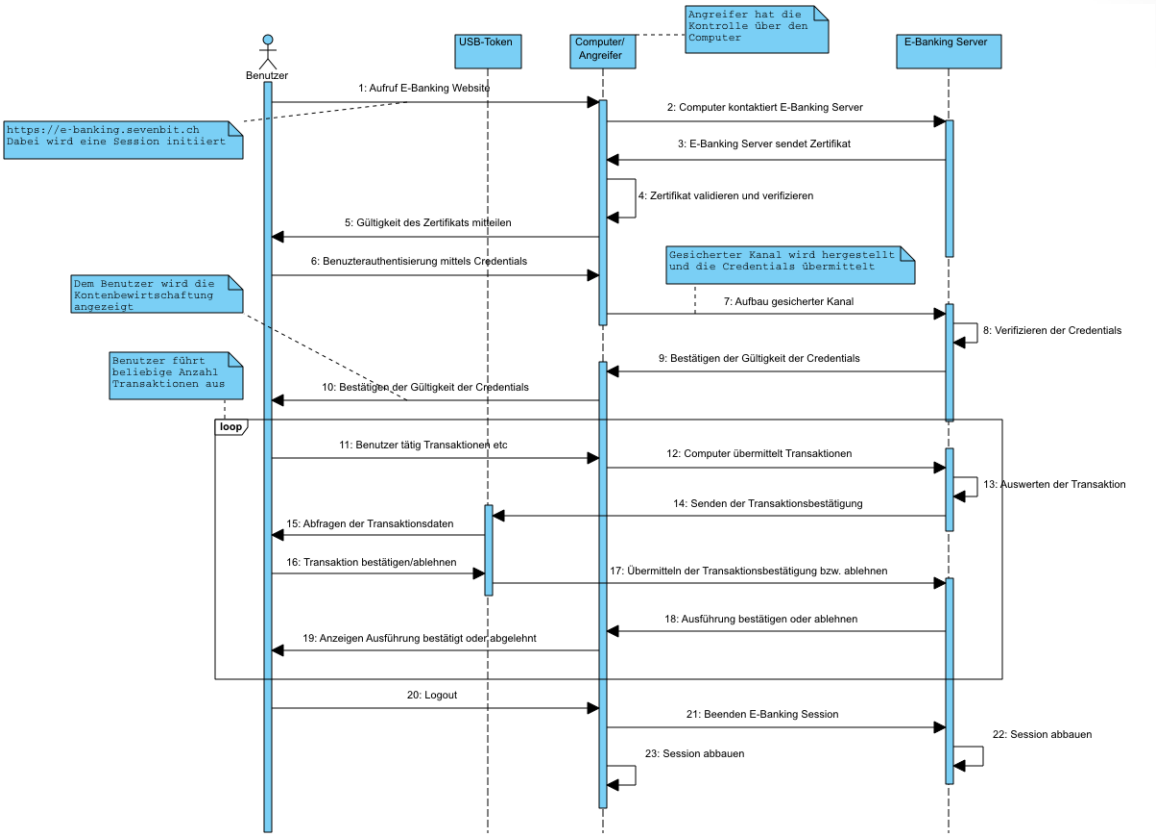

ABBILDUNG 18: E-BANKING SESSION MIT USB-TOKEN

Zwar hat der potentielle Angreifer auch hier weiterhin unter Punk 12 die Möglichkeit, die Transaktion des Benutzers zu manipulieren, allerdings fällt die Manipulation bei der Bestätigung auf.

Da der Kanal vom USB-Token über den User Agent zum Server gesichert ist, kann die Transaktionsbestätigung (Punkt 14) weder abgefangen, noch verändert werden. Das Einschleusen oder Manipulieren einer Transaktion gestaltet sich dadurch sehr anspruchsvoll, ohne dass es der Benutzer bemerkt.

Fazit

Sobald eine Transaktion über einen unabhängigen, alternativen Kanal bestätigt wird (mTAN über SMS, Security-Token, etc.) und der Angreifer diesen nicht auch unter seiner Kontrolle hat, wird das manipulieren von Transaktionen und deren Bestätigung nahezu unmöglich, ohne dass dies der Benutzer bemerkt.

Dies gilt jedoch nur, sofern jede Transaktion an einen neuen Empfänger beziehungsweise jede einzelne Transaktion bestätigt werden muss. Viele Banken haben jedoch standardmässig keinen solchen Mechanismus aktiviert.

Der Angreifer hat zwar weiterhin die Möglichkeit, Transaktionsangaben zu verfälschen, allerdings entsteht kein Nutzen daraus. In jedem Fall würde es dem Benutzer auffallen, dass sein User Agent kompromittiert ist.

Somit kann ein unabhängiger Kanal grundsätzlich als sicher aufgefasst werden, sofern der potentielle Angreifer nicht beide Kanäle kontrollieren kann. Auf diesen besonders mächtigen Fall, konzentriert sich die Thesis in einem der folgenden Kapitel.

Mobile E-Banking

Durch die zunehmende Popularität von Smartphones und Tablets zeichnet sich ein neuer Trend ab: Immer öfters werden mobile Gerät auch zum Tätigen von E-Banking verwendet. Aktuell sind Applikationen für die gängigsten mobilen Betriebssysteme verfügbar, welche beispielsweise das Bezahlen von Rechnungen mittels Scan-Funktionalität sehr vereinfachen. Nachfolgend werden Szenarios behandelt, die sich auf die Verwendung von mobilen Endgeräten im Bereich E-Banking beschränken. Als Bezeichnung für diese Endgeräte wird künftig der Begriff Smart-Device verwendet.

Ablauf mit Smart-Device

Die untenstehende Grafik zeigt den Ablauf einer E-Banking Session auf einem Smart-Device:

Ablauf im Überblick

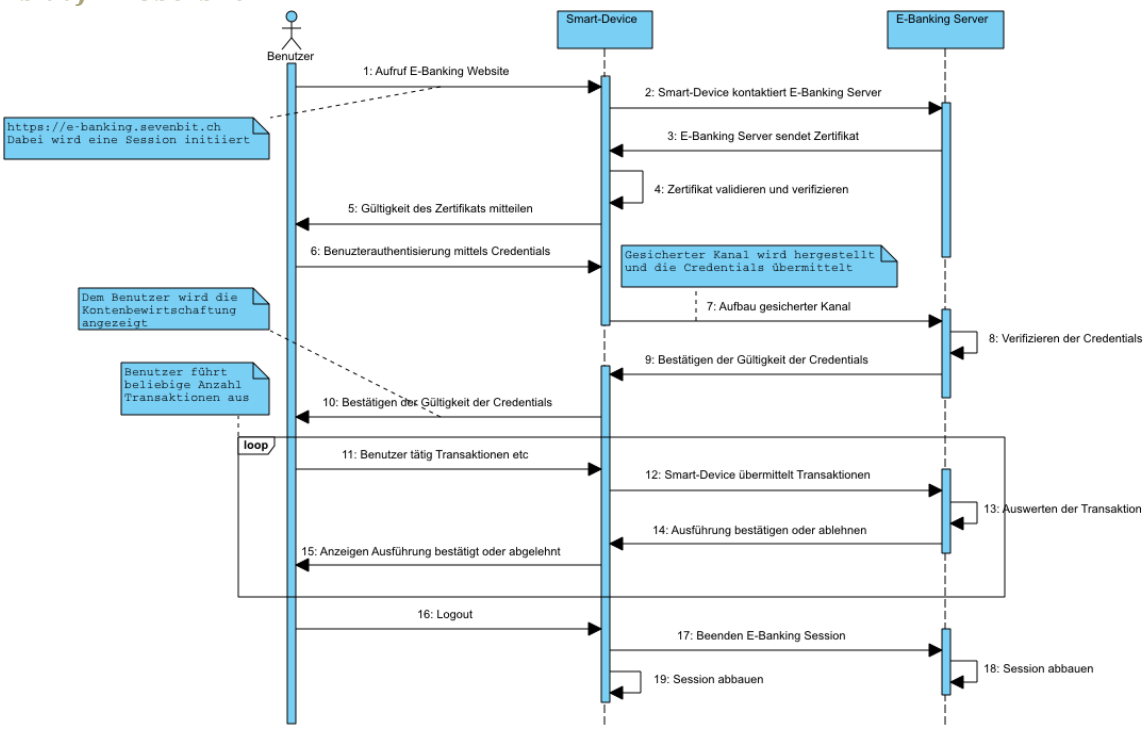

Abbildung 19: Ablauf E-Banking Session über Smart-Device

Hierbei fällt auf, dass der Verlauf der E-Banking Session, demjenigen aus Abschnitt „E-Banking Session ohne Angreifer“ entspricht. Ob also nun klassisches oder mobiles E-Banking betrieben wird, spielt für den Ablauf einer E-Banking Session keine entscheidende Rolle.

Wird nun jedoch das Smart-Device durch Malware kompromittiert, so entspricht das Szenario demjenigen aus Abschnitt „Traditionelle E-Banking Session: E-Banking Session mit Angreifer auf User Agent“.

Ablauf mit Smart-Device & TAN

Um nun Transaktionen bestätigen zu lassen, die via Smart-Device erfasst wurden, kann das bereits erwähnte TAN-Verfahren verwendet werden. Dabei findet beim Ablauf grundsätzlich keine Unterscheidung zwischen mTAN (mittels SMS) und TAN (mittels Code-Karte) statt.

Ablauf im Überblick

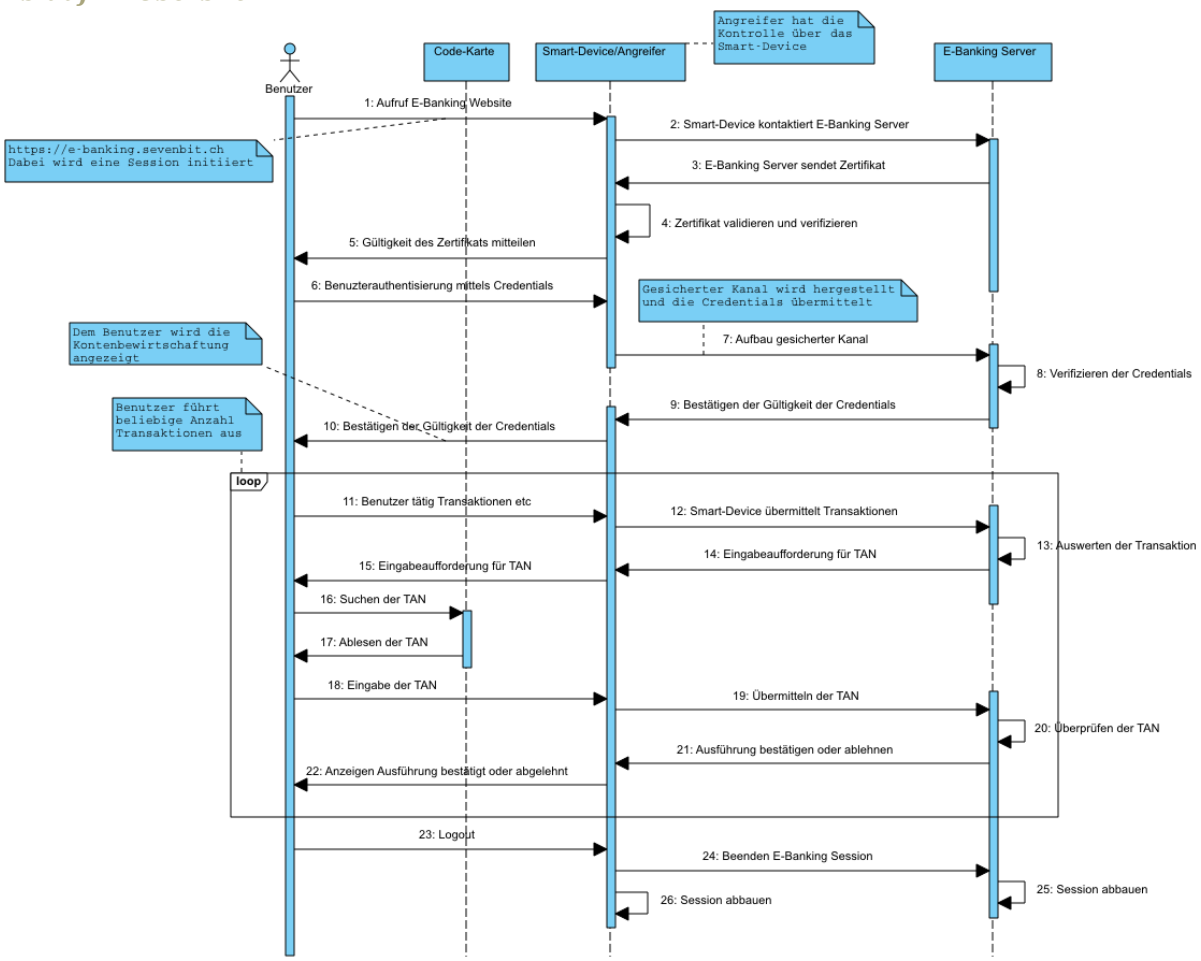

Abbildung 20: Ablauf E-Banking Session über Smart-Device & TAN

Es wird ersichtlich, dass hier der Angreifer analog den Abschnitten „Authentisierung: Mittels Code-Karte (Matrix-Karte, TAN)“ und „Mittels alternativen GSM-Kanal (Mobiltelefon, mTAN)“ weiterhin die Möglichkeit besitzt, Transaktionen zu manipulieren und durch den Benutzer zur Ausführung bestätigen zu lassen.

Da die mTAN nun auf den eigentlichen User Agent geschickt wird, kann nicht mehr von einem unabhängigen, alternativen Kanal gesprochen werden.

Ablauf mit Smart-Device & Secure-Token

Bei diesem Verfahren kommt ein sogenanntes „Secure Mobile Device“ zur Anwendung. Das Prinzip ist vergleichbar mit dem des USB-Tokens, welches bereits im früheren Abschnitt „Authentisierung:Mittels USB-Token“ besprochen wurde.

Das „Secure Mobile Device“ stellt dabei den dedizierten, gesicherten Kanal dar, über welchen Transaktionen angezeigt, bestätigt oder abgelehnt werden können.

Ablauf im Überblick

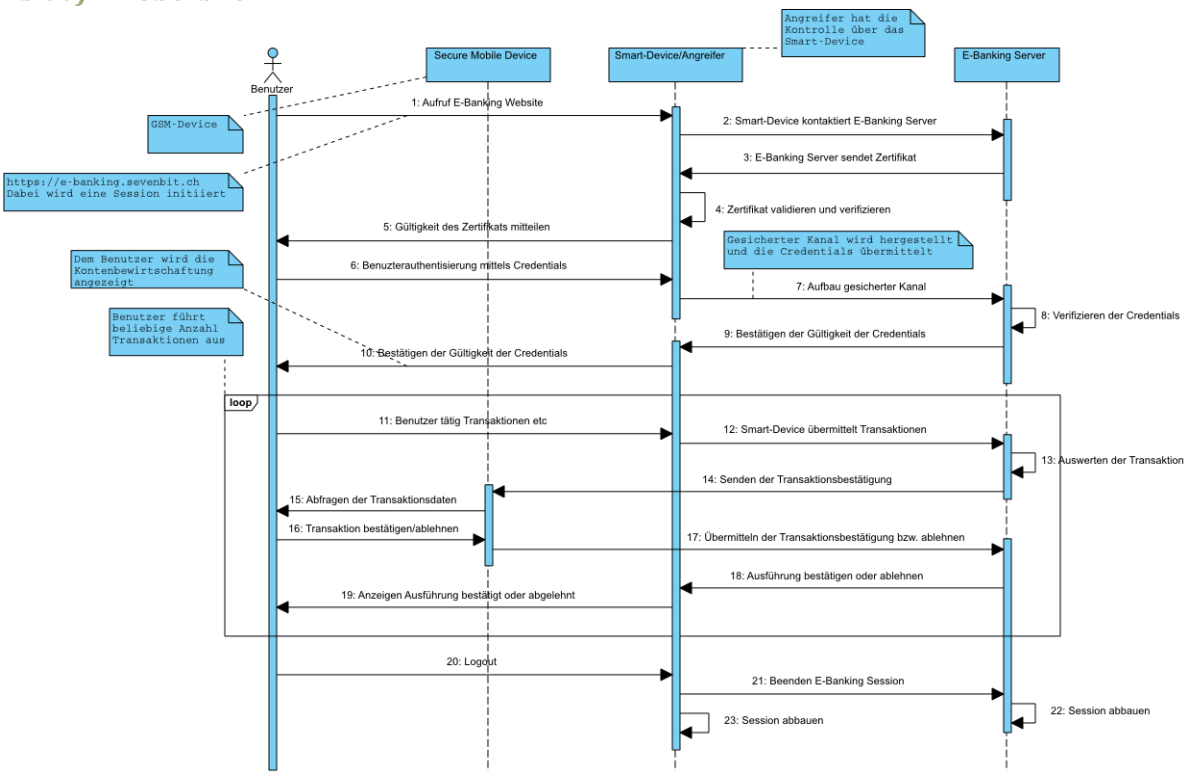

Abbildung 21: Ablauf E-Banking Session über Smart-Device & Secure-Token

Da allerdings bei den heutigen Smart-Devices eine entsprechende Schnittstelle zum Security-Token fehlt und eine solche Schnittstelle vermutlich auch in Zukunft nicht vorzufinden sein wird, muss nach einer anderen Lösung gesucht werden um einen solchen dedizierten, sicheren Kanal herstellen zu können.

Fazit

Der Trend zum mobilen E-Banking nimmt stetig zu. Die neue, mobile Lösung bringt jedoch auch neue Problematiken mit sich: Angreifern ist es auf dieser noch jungen Plattform möglich, eine laufende E-Banking Session einzusehen und allfällige Transaktionen zu manipulieren. Besonders heikel: Ein solches Angriffsszenario inklusive einer implementierten Attacke ist mit relativ wenig Aufwand und Ressourcen möglich.

Es existieren zwar bereits heute Lösungen, welche E-Banking in einem hohen Grad absichern und somit die Beeinflussung einer E-Banking Session nahezu verunmöglichen, allerdings sind diese Lösungen nicht ohne weiteres auf mobile Endgeräte portierbar.

Wie ein solcher Angriff auf einem Smart-Device erfolgen könnte und welche Lösungsansätze möglich wären, werden in den folgenden Kapiteln im Detail beschrieben.

Theoretische Analyse

Ausgangslage

Im Abschnitt „Mobile E-Banking: Ablauf mit Smart-Device & TAN“, wurde der Ablauf einer E-Banking Session mit einem Angreifer auf dem Smart-Device aufgezeigt.

Nun soll in diesem Kapitel der Ablauf unter Verwendung des mTAN-Verfahrens genauer betrachtet und ein praxisnahes Angriffsszenario aufgezeigt werden. Die E-Banking Plattform soll dabei browserbasiert sein. Eine mTAN wird sowohl für die Initialisierung einer E-Banking Session, als auch für die Bestätigung jeder Transaktion benötigt.

Bei der Initialisierung wird ein zufälliger Code generiert und per SMS zugestellt. Um das Login abschliessen zu können, muss die erhaltene mTAN zur Bestätigung eingeben werden. Bei einer Transaktion werden, nebst einem Bestätigungscode, auch alle Details zur ausgelösten Transaktion per SMS übermittelt. Der Benutzer vergleicht die Angaben und kann durch Eingabe des zugestellten Codes die Transaktion bestätigen beziehungsweise abschliessen.

Dieses Verfahren, also browserbasiertes E-Banking mit mTAN, hat die positive Eigenschaft, dass keine weiteren Applikationen und Devices – ausser Smart-Device oder Computer – benötigt werden und daher sowohl mit Desktop-Rechnern, als auch mit mobilen Geräten jederzeit E-Banking über dieselbe Plattform betrieben werden kann.

Es stellt sich die Frage, wie sich denn nun der Angreifer ohne aufzufallen auf dem Smart-Device platziert, um eine E-Banking Session manipulieren zu können. Eine solche Attacke, nennen wir sie „Blended Attack“, erfolgt in unterschiedlichen Schritten. Ein mögliches Szenario wird auf den nachfolgenden Seiten aufgezeigt und soll näher betrachtet werden.

Blended Attack

Allgemeines

Smart-Devices erfreuen sich zunehmender Beliebtheit. Nicht zuletzt aufgrund der hohen Leistungsfähigkeit und grossen Speicherkapazitäten. Auch deren Bedienung ist kinderleicht. Die breite Palette an Applikationen, welche ohne grosse Anforderungen an den Benutzer auf dem Device installiert werden können, macht einem das Leben einfacher oder gar amüsanter.

Allerdings stellt sich hierbei oft die Frage, ob die installierte Applikation auch tatsächlich „nur“ jene Funktionalität beinhaltet und ausführt, welche dem Benutzer vor der Installation im Beschreibungstext angegeben wurde. Von den meisten Benutzern wird die Funktionalität aber auch die dafür nötigen Zugriffsrechte einer Applikation, wohl eher selten angezweifelt. Das Misstrauen in eine Applikation sinkt sogar, je beliebter sie ist.

Das beispielsweise eine SMS-Applikation den Zugriff auf das Adressbuch, die SD-Karte und auf das Internet benötigt, erscheint auf den ersten Blick für die meisten ziemlich plausibel: Schliesslich wird der Zugriff auf das Adressbuch benötigt, da man sich die Kontakte und deren Nummern nicht merken will und kann. Der Zugriff auf die SD-Karte könnte damit begründet werden, dass die Applikation die SMS-Daten auf die SD-Karte sichert. Und, zu guter Letzt: Der Zugriff auf das Internet, kann beispielsweise für die Finanzierung der Applikation, welche gelegentlich dezente Werbung während der Ausführung einblendet, gerechtfertigt werden.

Abbildung 22: Installation auf Smart-Device

Die Verblendung

SMS-Bot

Doch was wäre, wenn nun die SMS-Applikation ohne das Wissen des Benutzers, dessen SMS-Nachrichten mitliest oder gar einzelne Nachrichten dem Benutzer vorenthält?

Was, wenn die Nachrichten automatisch nach bestimmten Mustern untersucht werden und die Applikation plötzlich beginnt, im Namen des Benutzers, selbstständig Nachrichten zu versenden? Oder, dass ein teures SMS-Abonnement abgeschlossen wird? Was, wenn das Smart-Device über einen SMS-Bot überwacht, kontrolliert und gar gesteuert werden kann? Wenn ein vermeintlicher Angreifer per SMS einen Befehl auf dem Gerät ausführt?

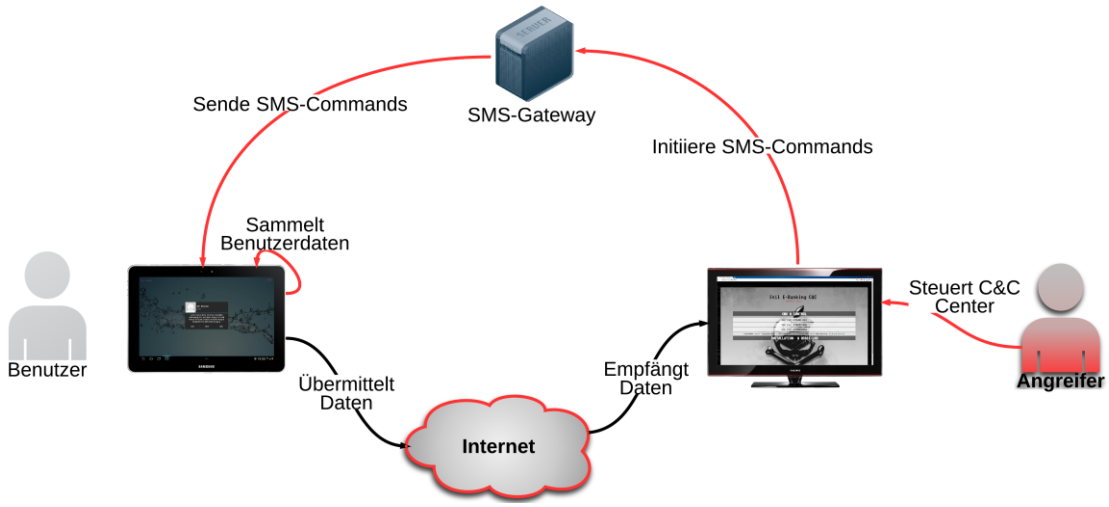

ABBILDUNG 23: SMS-BOT

Angriffsszenario

Überblick

Wie kann nun die „Blended Attack“ auf eine E-Banking Session angewendet werden?

Um die Attacke möglichst unauffällig umzusetzen, ist eine Kollaboration von unterschiedlichen Applikationen auf dem infizierten Device notwendig. Wenn man die Ausführungen im Abschnitt „Ablauf mit Smart-Device & TAN“ betrachtet, so besteht eine solche Session aus folgenden Komponenten:

- SMS-Applikation
- Webbrowser
- E-Banking Server
- Unterschiedliche Übertragungsmedien & -Kanäle
- Smart-Device
- Benutzer

Schematischer Aufbau

Die Abbildung 24: E-Banking Session mit Smart-Device zeigt auf, welche unterschiedlichen Komponenten und Kommunikationskanäle in diesem analysierten Szenario involviert sind.

Dabei verwendet der Benutzer den Webbrowser und SMS-Applikation auf dem Smart-Device. Der Webbrowser visualisiert die Informationen, welche während einer E-Banking Session an- beziehungsweise abgefragt werden, währendem die SMS-Applikation die Codes zur Transaktionsbestätigung erhält.

Das Smart-Device kommuniziert also per mobilem Internet und GSM-Kanal direkt beziehungsweise indirekt mit dem E-Banking Server. Hierbei ist zu beachten, dass das mobile Internet für An- und Abfragen an den E-Banking Server (duplex) und der GSM-Kanal – im klassischen Fall – lediglich für die Code-Übertragung (mTAN) verwendet wird (simplex).

Hierbei wird nun nochmals deutlich, dass die Verifikation einer Transaktion, auf demselben Gerät, sprich auf dem Smart-Device, wie auch das Auslösen und Bestätigen einer Transaktion erfolgt. Dies vereinfacht den analysierten und implementierten Angriff enorm.

Aufbau im Überblick

Folgende Abbildung soll schematisch den Aufbau und das Zusammenspiel der Komponenten verdeutlichen:

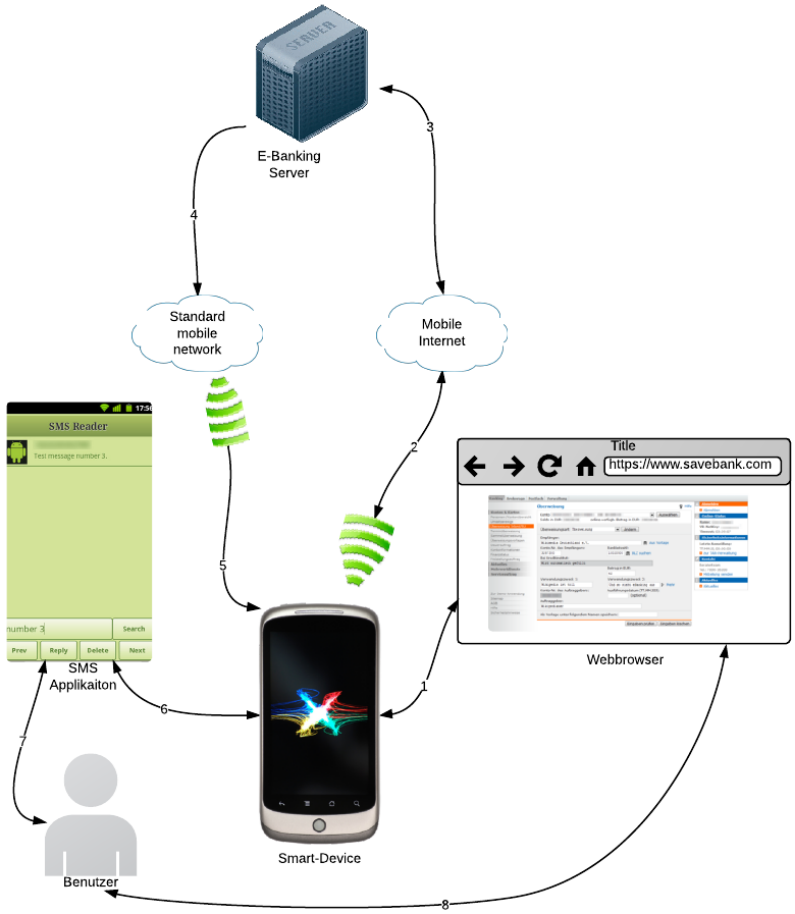

ABBILDUNG 24: E-BANKING SESSION MIT SMART-DEVICE

Der Angriff

Schwachpunkt

Im Falle einer Transaktionsbestätigung werden zwar zwei getrennte Kanäle verwendet, die allerdings auf dem gleichen Endgerät zusammenfallen. Dadurch verschwindet der zuvor dedizierte Kanal. Dies ist unter anderem einer der Schwachpunkte, welcher für den inszenierten Angriff ausgenutzt werden kann. Die SMS-Applikation und der Webbrowser ermöglichen den Zugriff auf das Smart-Device und können dieses nach der Installation beider Applikationen, in einem gewissen Rahmen überwachen und steuern.

Gelingt einem Angreifer, diese beiden Komponenten mit böswilligen Eigenschaften anzureichern, kann eine Transaktion ohne das Wissen des Benutzers während einer E-Banking Session so verändert werden, als wäre diese Transaktion tatsächlich durch den Benutzer selbst ausgelöst worden.

Ablauf im Detail

Das Szenario einer manipulierten Transaktion würde sich in etwa wie folgt abspielen:

1. Der Benutzer erfasst über den Webbrowser eine Transaktion und schickt diese ab.
2. Der Webbrowser manipuliert die Transaktion, in dem er bspw. die Empfängerkontonummer und den zu überweisenden Betrag verändert und schickt die Transaktion weiter an den E-Banking Server.
3. Gleichzeitig speichert der Webbrowser die veränderten Daten der Transaktion auf einen für Applikationen zugänglichen Speicher ab, wie beispielswiese in einer Datei auf der SD-Card oder in einer Datenbank. Weiter kann der Browser die ebenfalls installierte SMS-Applikation über die Manipulation informieren.
4. Der E-Banking Server erhält die Transaktion und sendet zur Transaktionsbestätigung die mTAN an den Benutzer.
5. Die SMS-Applikation empfängt das SMS mit der darin enthaltenen mTAN. Die Applikation liest ihre erhaltenen Daten aus oder sucht auf der SD-Karte nach der Datei, in welcher die Informationen zur manipulierten Transaktion abgelegt sind, ändert den zu überweisenden Betrag und falls vorhanden den Namen und die Kontonummer des Empfängers und informiert den Benutzer über den Erhalt einer SMS.
6. Der Benutzer überprüft die Transaktionsdaten, bestätigt diese mit der erhaltenen mTAN und hat damit eine Transaktion bestätigt, welche er so nie erfasst hatte.

Ablauf im Überblick

Der Ablauf des Szenarios soll hier noch einmal graphisch dargestellt werden:

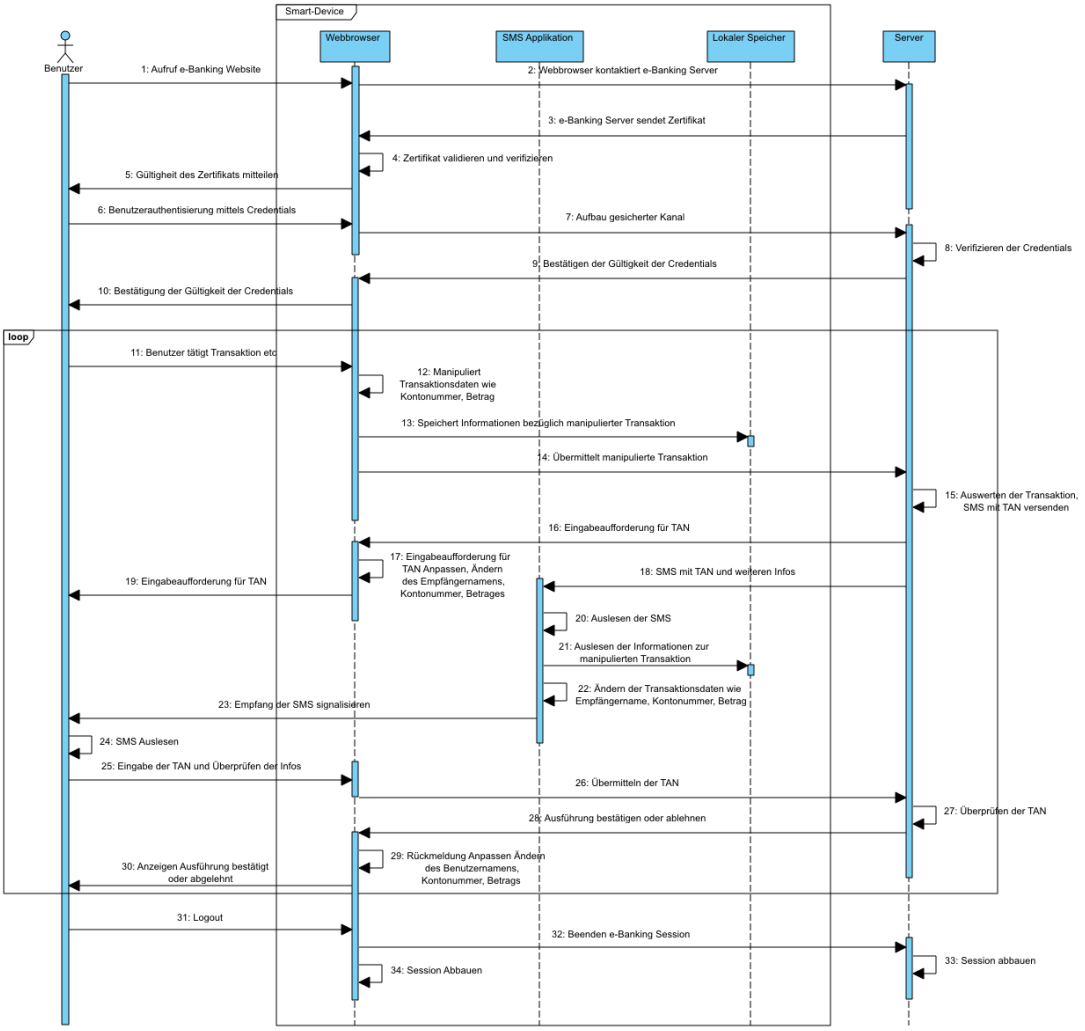

ABBILDUNG 25: E-BANKING SESSION „BLENDED ATTACK“

Fazit des Angriffsszenarios

Angriff durch Verschleierung

Wie im Szenario aufgezeigt, können alltägliche Applikationen wie ein Webbrowser und eine SMS-Applikation eine E-Banking Session negativ beeinflussen oder gar manipulieren.

Für den Benutzer ist die bösartige Kollaboration der Applikationen untereinander nicht ersichtlich und auch das Erkennen solcher Tätigkeiten dürfte sich als schwierig erweisen.

Da solche Applikationen im Kontext des Benutzers ausgeführt werden und des Weiteren vordergründig als gewöhnliche Applikationen agieren, wird ihnen wohl nicht einmal ein Malwarescanner auf die Schliche kommen. Oder erst dann, wenn bereits zahlreiche Geräte übernommen und eine grosse Zahl an manipulierten Transaktionen getätigt werden konnten.

Praktische Umsetzung

Ausgangslage

Im vorhergehenden Kapitel wurde ein mögliches, theoretisches Szenario aufgezeigt, welches eine Manipulation einer E-Banking Session zulassen würde.

Hierbei stellt sich nun die Frage:

- Wie gross ist der Aufwand für die Umsetzung eines solchen Angriffs?
- Wie viel Wissen wird für die Umsetzung der Komponenten auf Android benötigt?
- Wie viel Zeit wird für die Umsetzung des gesamten Szenarios benötigt?
- Welche Ressourcen und Mittel sind dafür nötig?
- Mit welchen Problemen wird man konfrontiert?

Diese Fragen sollen am Ende dieser Arbeit beantwortet werden.

Angriffsszenario

Überblick

Der nun in die Praxis umgesetzte Angriff basiert auf folgenden Komponenten:

- Benutzer
- SMS-Applikation SMS Buddy
- Webbrowser Web Buddy
- E-Banking Web-Plattform
- SMS-Gateway
- Command & Control-Plattform
- Angreifer

Komponenten im Überblick

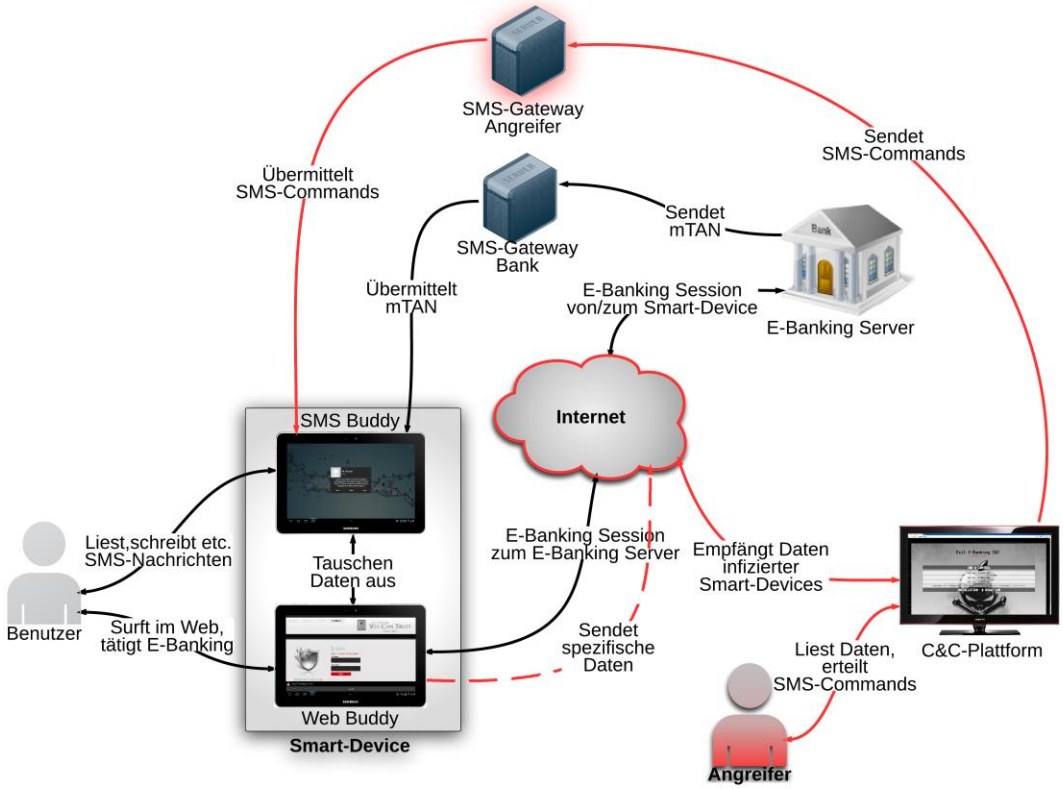

ABBILDUNG 26: KOMPONENTEN IM ÜBERBLICK

Auf den folgenden Seiten wird einerseits der detaillierte Ablauf des Angriffs mit seinen dazu nötigen Komponenten beziehungsweise Teilprodukten aufgezeigt. Andererseits werden die einzelnen Komponenten, um einen groben Überblick zu erhalten, kurz vorgestellt.

Ablauf E-Banking Session ohne Angreifer

Ablauf im Überblick

Die nachfolgende Abbildung zeigt den Ablauf einer E-Banking Session, wie sie per Smart-Device erfolgen würde. Zu beachten ist dabei, dass es sich beim Smart-Device immer um dasselbe Gerät mit zwei separierten Applikationen – dem Web Buddy und dem SMS Buddy – handelt. Zudem zeigt die Grafik abstrakt das Zusammenspiel der Komponenten auf, wobei der Fokus beim Ausführen einer Transaktion innerhalb einer bereits initiierten E-Banking Session liegt.

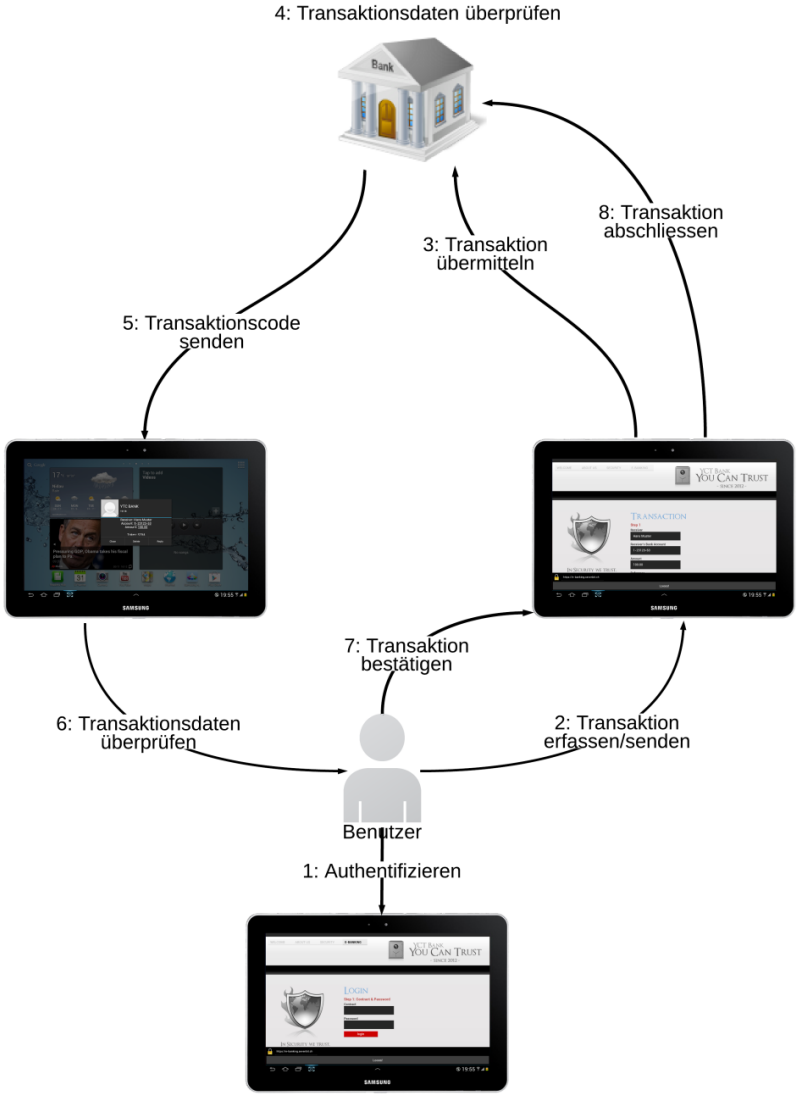

ABBILDUNG 27: E-BANKING SESSION OHNE ANGREIFER

Ablauf im Detail

Im Folgenden wird der Ablauf zum Erfassen und Bestätigen von Transaktionen über ein Smart-Device beschrieben:

1. Um eine E-Banking Session starten zu können, muss der Benutzer seine Credentials eingeben. In diesem Fall handelt es sich um die Vertragsnummer und ein persönliches Passwort.
2. Zum Erfassen einer Transaktion ruft der Benutzer die entsprechende Transaktions-Eingabemaske auf. Darin werden Empfänger, Empfängerkontonummer, der zu überweisende Betrag und eine Referenz eingeben.
3. Über einen entsprechenden Button wird die Übermittlung der Transaktion an die Bank ausgelöst.
4. Die Bank überprüft die Transaktion auf Gültigkeit und Vertrauenswürdigkeit.
5. Gilt die Transaktion als valid (bspw. keine Überweisung über CHF 25'000.-), so wird ein Transaktionscode (mTAN) ausgelöst und an das Smart-Device des Benutzers als SMS-Nachricht gesendet. Die Nachricht kann weitere Informationen zur Transaktion enthalten wie beispielsweise Empfänger, Empfängerkontonummer und den zu überweisenden Betrag.
6. Der Benutzer kann nun die Angaben in der SMS-Nachricht mit den Angaben der E-Banking Plattform vergleichen.
7. Sind die Angaben korrekt, wird die Transaktion mittels Eingabe der mTAN durch den Benutzer bestätigt.
8. Die Bestätigung der Transaktion wird anschliessend an die Bank übermittelt. Seitens der Bank, kann die Transaktion erneut Prozesse durchlaufen, welche die Gültigkeit und Vertrauenswürdigkeit überprüft.

Gültigkeit & Vertrauenswürdigkeit

Unter Punkt 8 werden die Transaktionen auf Gültigkeit und Vertrauenswürdigkeit geprüft. Was ist darunter zu verstehen?

- Als gültige Transaktion (Gültigkeit) sei beispielsweise zu verstehen, dass der Empfänger einer Zahlung mit dem jeweiligen Empfängerkonto assoziiert werden kann.
- Als vertrauenswürdige Transaktion (Vertrauenswürdigkeit) sei zu verstehen, dass Mechanismen verwendet werden, welche getätigte Zahlungen analysieren. So sollte eine Rücksprache mit dem Kontoinhaber gehalten werden, wenn eine Zahlung in einer ungewohnten Höhe getätigt wird. Ein Beispiel: Werden vom Kontoinhaber durchschnittlich Zahlungen in der Höhe von CHF 3000.- getätigt, sollte eine Rücksprache erfolgen, falls plötzlich ein Zahlungsauftrag in der Höhe CHF von 25'000.- ausgelöst wurde.

Ablauf E-Banking Session mit Angreifer

Angriffsmethoden

Wird nun der Angreifer ins Szenario aufgenommen, so sollen zwei mögliche Methoden zur Fälschung von Transaktionen aufgezeigt werden:

- Manipulation einer Transaktion
- Einschleusen bzw. Injektion einer Transaktion

Damit die nachfolgenden Angriffsmethoden erfolgreich durchgeführt werden können, wird als Ausganglage, nebst erfolgreichem Starten einer E-Banking Session durch den Benutzer, vorausgesetzt, dass die Applikationen SMS Buddy und Web Buddy erfolgreich auf dem Smart-Device installiert wurden.

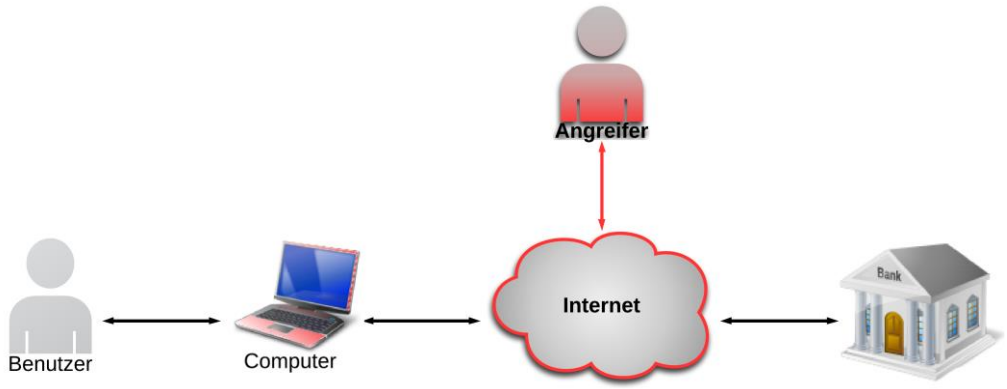

ABBILDUNG 28: KOMPROMITTIERTES SYSTEM

Die in den Angriffsmethoden involvierten Komponenten, werden in den nachfolgenden Kapiteln im Detail betrachtet.

E-Banking Session mit manipulierter Transaktion

Ablauf im Überblick

Im folgenden Szenario soll aufgezeigt werden, wie Transaktionen manipuliert und vor dem Benutzer verschleiert werden können.

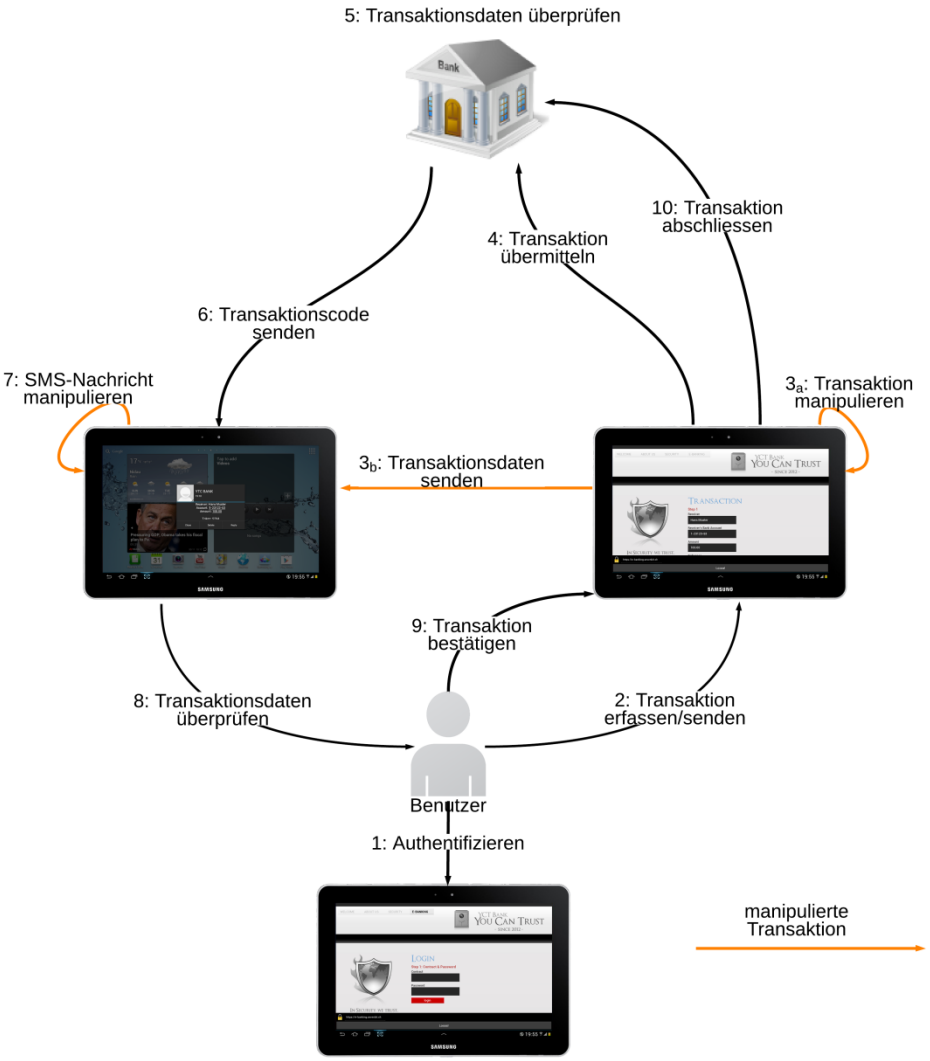

Abbildung 29: E-Banking Session mit manipulierter Transaktion

Ablauf im Detail

Die einzelnen Schritte im Detail:

1. Der Benutzer startet durch Eingabe seiner Credentials eine E-Banking Session.
2. Zum Erfassen einer Transaktion ruft der Benutzer die entsprechende Transaktions-Eingabemaske auf. Darin werden Empfänger, Empfängerkontonummer, der zu überweisende Betrag und eine Referenz eingeben.
3.
    a. Über einen entsprechenden Button wird die Übermittlung der Transaktion an die Bank ausgelöst. Bevor jedoch die vom Benutzer erfasste Transaktion an die Bank übermittelt wird, verändert Web Buddy die jeweiligen Transaktionsdaten.
    b. Die originalen sowie die manipulierten Transaktionsdaten werden an SMS Buddy übermittelt. Jede eintreffende SMS-Nachricht wird analysiert und untersucht, ob diese die manipulierten Transaktionsdaten enthält.
4. Anschliessend wird die manipulierte Transaktion an die Bank übermittelt.
5. Die Bank überprüft die Transaktionsdaten auf Gültigkeit und Vertrauenswürdigkeit
6. Gilt die Transaktion als valid (bspw. keine Überweisung über CHF 25'000.-), so wird ein Transaktionscode (mTAN) ausgelöst und an das Smart-Device des Benutzers als SMS-Nachricht gesendet. Die Nachricht kann weitere Informationen zur Transaktion enthalten wie beispielsweise Empfänger, Empfängerkontonummer und den zu überweisenden Betrag.
7. SMS Buddy verändert nach Erhalt der SMS-Nachricht den Nachrichteninhalt so, dass der Benutzer im Glauben bleibt, dass die erhaltene mTAN für die von ihm erfasste Transaktion gedacht sei.
8. Der Benutzer kann nun die Angaben in der SMS-Nachricht mit den Angaben der E-Banking Plattform vergleichen.
9. Sind die Angaben korrekt, wird die Transaktion mittels Eingabe der mTAN durch den Benutzer bestätigt und abgeschlossen.
10. Die Bestätigung der Transaktion wird anschliessend an die Bank übermittelt. Seitens der Bank, kann die Transaktion erneut Prozesse durchlaufen, welche die Gültigkeit und Vertrauenswürdigkeit überprüft.

E-Banking Session mit injizierter Transaktion

Ablauf im Überblick

Als nächstes Angriffsszenario soll die Möglichkeit veranschaulicht werden, wie eine weitere Transaktion eingeschleust werden kann.

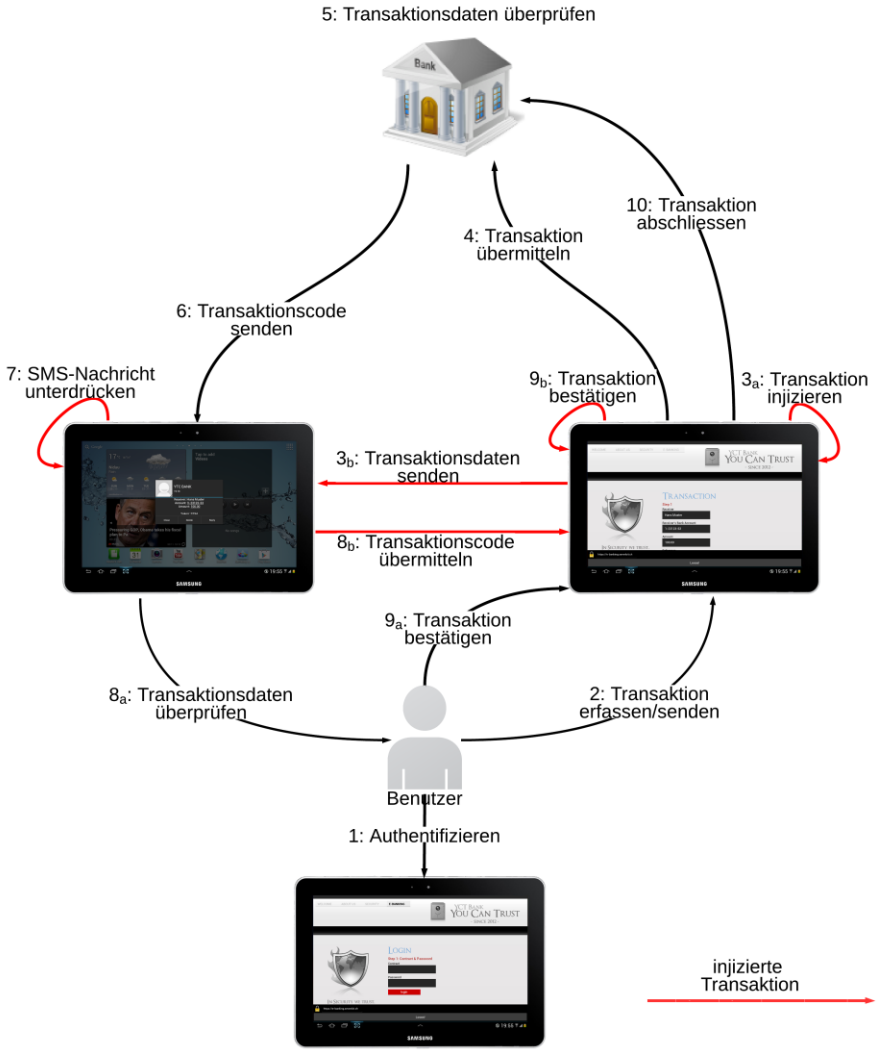

ABBILDUNG 30: E-BANKING SESSION MIT INJIZIERTER TRANSAKTION

Ablauf im Detail

Die einzelnen Schritte im Detail:

1. Der Benutzer startet durch Eingabe seiner Credentials eine E-Banking Session.
2. Zum Erfassen einer Transaktion ruft der Benutzer die entsprechende Transaktions-Eingabemaske auf. Darin werden Empfänger, Empfängerkontonummer, der zu überweisende Betrag und eine Referenz eingeben.
3.
    a. Über einen entsprechenden Button wird die Übermittlung der Transaktion an die Bank ausgelöst. Bevor jedoch die vom Benutzer erfasste Transaktion an die Bank übermittelt wird, injiziert Web Buddy eine weitere Transaktion.
    b. Die originalen sowie die manipulierten Transaktionsdaten werden an SMS Buddy übermittelt. Jede eintreffende SMS-Nachricht wird analysiert und untersucht, ob diese die manipulierten Transaktionsdaten enthält.
    Die vom Benutzer erfasste Transaktion wird bis auf weiteres von Web Buddy zurückgehalten und die Anzeige zwischengespeichert, das Bild wird eingefroren.
4. Die injizierte Transaktion wird an die Bank übermittelt.
5. Die Bank überprüft die Transaktionsdaten auf Gültigkeit und Vertrauenswürdigkeit.
6. Gilt die Transaktion als valid (bspw. keine Überweisung über CHF 25'000.-), so wird ein Transaktionscode (mTAN) ausgelöst und an das Smart-Device des Benutzers als SMS-Nachricht gesendet. Die Nachricht kann weitere Informationen zur Transaktion enthalten wie beispielsweise Empfänger, Empfängerkontonummer und den zu überweisenden Betrag.
7. SMS Buddy fängt die SMS-Nachricht mit den Angaben zur injizierten Transaktion ab und liesst den Transaktionscode aus. Danach wird die SMS-Nachricht vom Smart-Device gelöscht, so dass der Benutzer nichts vom Erhalt der Nachricht bemerkt.
8.
    b. SMS Buddy übermittelt den Transaktionscode an Web Buddy.
9.
    b. Nach Erhalt des Transaktionscodes wird nun Web Buddy selbständig die injizierte Transaktion bestätigen und abschliessen.
10. Die Bestätigung der Transaktion wird anschliessend an die Bank übermittelt. Seitens der Bank, kann die Transaktion erneut Prozesse durchlaufen, welche die Gültigkeit und Vertrauenswürdigkeit überprüft.

Nach erfolgtem Versand der Transaktionsbestätigung, wird nun die eigentliche Transaktion des Benutzers an die Bank übermittelt (Schritt 4 in der Grafik). Die versandte Transaktion durchläuft nun die im Abschnitt „Ablauf E-Banking Session ohne Angreifer“ beschriebene Prozedere:

4. Die Transaktion wird an die Bank übermittelt
5. Die Bank überprüft die Transaktion auf Gültigkeit und Vertrauenswürdigkeit.
6. Gilt die Transaktion als valid (bspw. keine Überweisung über CHF 25'000.-), so wird ein Transaktionscode (mTAN) ausgelöst und an das Smart-Device des Benutzers als SMS-Nachricht gesendet. Die Nachricht kann weitere Informationen zur Transaktion enthalten wie beispielsweise Empfänger, Empfängerkontonummer und den zu überweisenden Betrag.
   Die zuvor zwischengespeicherte Anzeige wird wieder aktualisiert.
7. Die von der Bank versandte SMS-Transaktionsbestätigung lässt nun SMS Buddy unverändert passieren – der Benutzer wird somit über den Erhalt der SMS-Nachricht informiert.
8.
   a. Der Benutzer kann nun die Angaben in der SMS-Nachricht mit den Angaben seitens E-Banking Plattform überprüfen.
9.
   a. Sind die Angaben korrekt, wird die Transaktion mittels Eingabe der mTAN durch den Benutzer bestätigt und abgeschlossen.
10. Die Bestätigung der Transaktion wird anschliessend an die Bank übermittelt. Seitens der Bank, kann die Transaktion erneut Prozesse durchlaufen, welche die Gültigkeit und Vertrauenswürdigkeit überprüft.

Verschleierung der Transaktionen

Damit nun die eingeschleusten und manipulierten Transaktionen nicht dem Benutzer in der Zahlungsübersicht angezeigt werden, bedarf es einen weiteren Schritt. Ausgehend davon, dass der Benutzer die Übersicht der getätigten Transaktionen via Smart-Device betrachtet, muss Web Buddy die Anzeige so verändern, dass die eingeschleusten Transaktionen erst gar nicht angezeigt werden.

Weiter müssen jene Transaktionen, die manipuliert wurden, so in der Anzeige angepasst werden, als ob es jene Transaktionen sind, die der Benutzer ursprünglich selbst erfasst hat.

Ablauf im Überblick

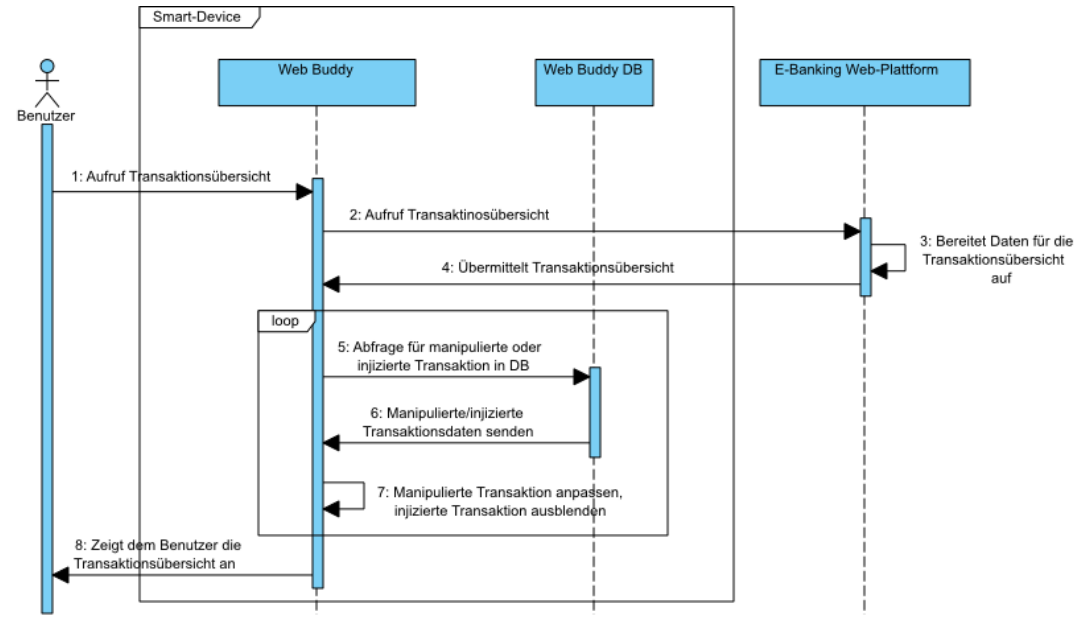

ABBILDUNG 31: VERSCHLEIERUNG DER TRANSAKTIONEN IN DER TRANSAKTIONSÜBERSICHT

Fazit des Angriffs

Die in den Angriffsszenarien aufgezeigten Manipulationsmöglichkeiten sind daher möglich, da der Kanal zum Senden einer Transaktion und der Kanal zum Bestätigen einer Transaktion auf demselben Gerät/Smart-Device zusammenfallen.

Wie in den Szenarien aufgezeigt, können Transaktionen verändert oder aber auch eingeschleust werden, ohne dass dies der Benutzer bemerkt.

Die Tatsache, dass der Webbrowser die vollständige Kontrolle über den anzuzeigenden und auszuführenden Inhalt hat und dadurch eine Website gezielt manipulieren kann, darf dabei nicht ausser Acht gelassen werden. Dadurch könnte es einem Angreifer gelingen, bestimmte Informationen dem Benutzer vorzuenthalten. Als Beispiel sei hier die Bestätigung einer eingeschleusten Transaktion oder das komplette ausblenden von Transaktionen im Bankauszug zu erwähnen.

Allgemein kann das Szenario aber auch auf das klassische E-Banking, bestehend aus beispielsweise einem Desktop-PC und einem Smart-Device, angewendet werden: Können beide Geräte kontrolliert werden, fallen die bis anhin dedizierten Kanäle zusammen – eine Manipulation oder gar eine Einschleusung von Transaktionen wird ermöglicht.

Da bei vielen E-Banking-Plattformen gänzlich auf eine Transaktionsbestätigung verzichtet wird, ist sogar die Kontrolle nur eines Gerätes ausreichend, um ohne das Wissen des Benutzers, Transaktionen zu verändern oder zu einschleusen.

SMS-Applikation

Nach der Devise „not yet another sms app“ wurde auf die komplette Eigenentwicklung einer SMS-Applikation verzichtet. Somit wurde der Entscheid gefällt, eine bestehende Lösung so zu verändern, dass sie für das Angriffsszenario eingesetzt werden kann.

Folgende Kriterien musste die Applikation im Vorfeld erfüllen:

- Eine verhältnismässig hohe Popularität (mindestens 20'000 Installationen) besitzen
- Der Quellcode soll frei verfügbar sein, so dass man auch als Collaborator mitarbeiten könnte

Nach einer Recherche, wurde die passende SMS-Applikation ermittelt. Das Augenmerk fiel auf die Applikation SMS Popup, welche mit über 41'000 Installationen eine gewisse Popularität aufweist. SMS Popup ist eine Entwicklung von Adam K und kann via Googles Play Store bezogen werden. Die für diese Thesis erarbeitete Implementation basiert auf der SMS Pop Version 1.2.4.

Zudem ist der Quellcode frei verfügbar, was somit die Möglichkeit bietet, gewisse Funktionalitäten zu implementieren, welche für das Angriffsszenario nötig sind.

Allgemeine Funktionsübersicht

Vordergründige Funktionalität:

- SMS auf Smart-Device archivieren (Backup)
- Dialog: zum Lesen, Beantworten oder Löschen einer Nachricht

Funktionen im Kontext des Angriffs:

- SMS vor Standard-SMS-Applikation abrufen
- empfangene SMS analysieren
- empfange SMS gezielt manipulieren
- Daten an kollaborierende Applikation Web Buddy weiterleiten

Weitere Funktionen:

- Optionen: z.B. Art der Benachrichtigung (LED, Klingelton)

User-Interface

ABBILDUNG 32: LIZENZVEREINBARUNG

ABBILDUNG 33: DEINSTALLATION

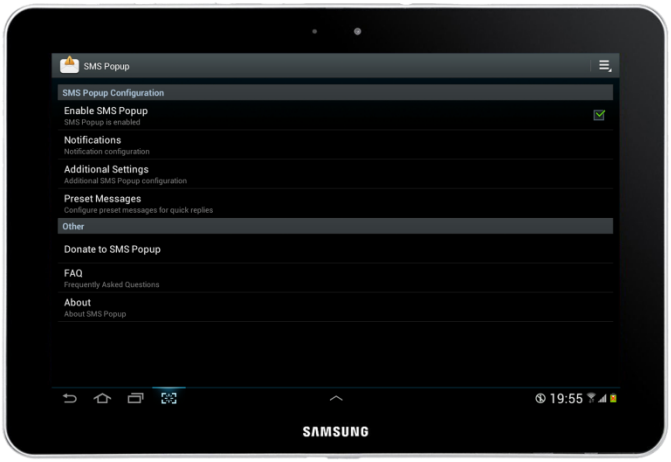

Abbildung 34: Einstellungen

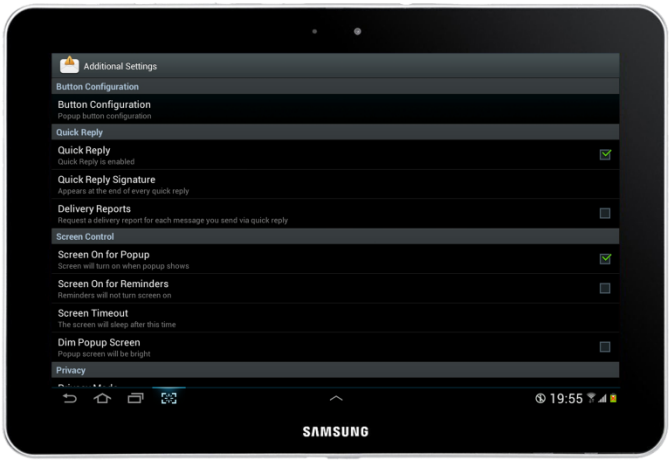

Abbildung 35: erweiterte Einstellungen

ABBILDUNG 36: EINGEHENDES SMS

ABBILDUNG 37: DIREKTANTWORT

Webbrowser

Bei dieser Applikation wurde eine Eigenentwicklung angestrebt.

Die Android-Plattform stellt die sogenannte „WebView“ – Komponente zur Verfügung, die es ermöglicht, Webinhalt auf dem Smart-Device darzustellen. Des Weiteren wird JavaScript vollumfänglich unterstützt.

Diese Eigenschaft mag auf den ersten Augenblick nicht gerade besonders interessant erscheinen. Bei genauerer Betrachtung, wird aber ersichtlich, dass mittels JavaScript der Inhalt einer Website verändert werden kann. Der Benutzer bemerkt nicht, dass die aufgerufene Site vom Browser manipuliert wurde. Zudem besteht die Möglichkeit, dass implementierte Methoden in der jeweiligen Android-Applikation (Webbrowser), direkt aus der Website heraus aufgerufen werden können. Unter gewissen Umständen können so aus dem jeweiligen Smart-Device Informationen ausgelesen werden, die der Benutzer gar nicht preisgeben möchte.

Der Funktionsumfang des Webbrowsers wurde auf das wesentliche reduziert. Auf Funktionen wie Chronik, Lesezeichen, etc. wurde bewusst verzichtet, womit die Applikation schlank gehalten werden konnte.

Allgemeiner Funktionsbeschrieb

Vordergründige Funktionalität:

- Webseite aufrufen & anzeigen

Funktionen im Kontext des Angriffs:

- E-Banking Web-Plattform analysieren
- Inhalt der Web-Plattform dynamisch manipulieren
- Daten der kollaborierenden Applikation SMS Buddy verarbeiten
- Transaktionen zu Gunsten des Angreifers manipulieren und injizieren
- manipulierte Transaktionen ausführen
- manipulierte und injizierte Transaktionen aus Bankauszug (HTML) entfernen
- manipulierter Bankauszug anzeigen

User-Interface

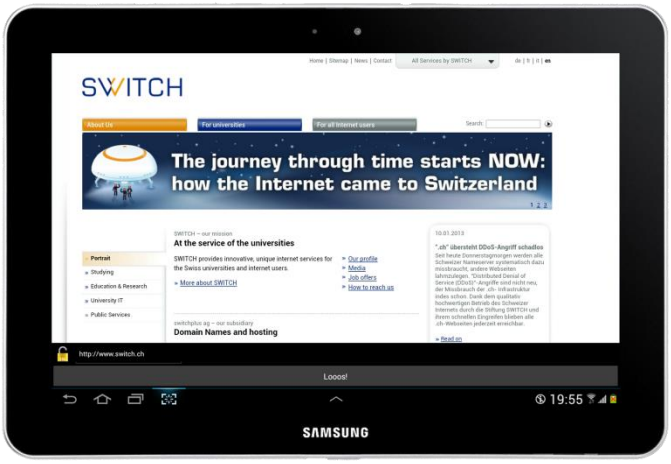

ABBILDUNG 38: BEDIENUNG WEBBROWSER

E-Banking Web-Plattform

Damit der Praxiseinsatz des Angriffsszenarios möglichst realistisch demonstriert werden kann, wird eine entsprechende E-Banking Plattform benötigt.

Für das Szenario wäre der Einsatz einer produktiven, realen E-Banking Plattform am aussagekräftigsten. Jedoch wären wohl die wenigsten E-Banking Plattform-Anbieter erfreut, wenn das jeweilige System einer entsprechenden Attacke nicht Stand halten konnte.

Aus diesem Grund wurde eine eigene E-Banking Web-Plattform entwickelt, welche die grundlegenden E-Banking-Funktionen sowohl für den klassischen Desktop, als auch für Smart-Devices bereitstellt.

Die Seite ist über folgenden Link erreichbar:
https://e-banking.sevenbit.ch (Vertragsnummer: 1111 und 2222 / Kennwort: test)

Allgemeine Funktionsübersicht

Vordergründige Funktionalität:

- Zugriff mittels Vertragsnummer und mTAN-Verfahren
- grundlegende E-Banking-Funktionen (Login, Bankauszug, Transaktionen) bereitstellen
- getätigte Transaktionen per SMS bestätigen (mTAN)
- Export der Transaktionsübersicht (Bankauszug)

Funktionen im Kontext des Angriffs:

- Keine: Es wurden keine „Hintertüren“ oder dergleichen implementiert, jegliche Manipulation einer Website erfolgt über den eingesetzten Webbrowser

User-Interface

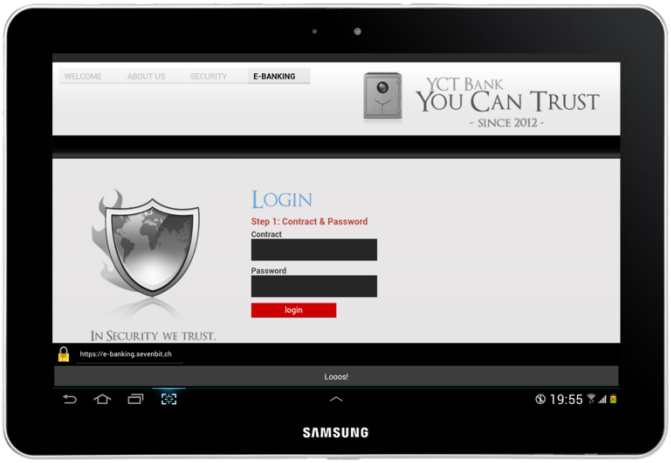

ABBILDUNG 39: E-BANKING LOGIN SMART-DEVICE

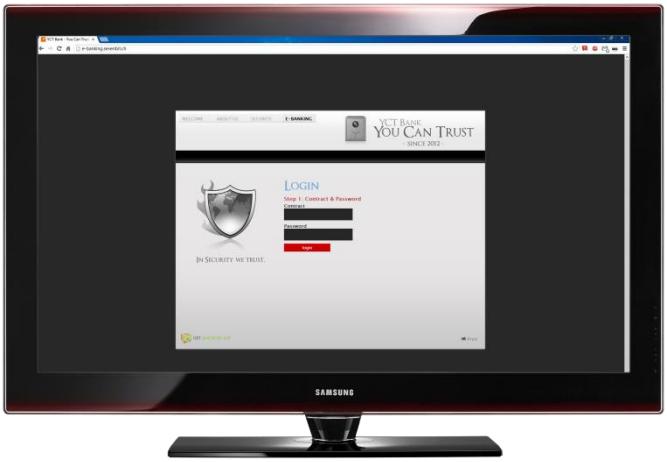

ABBILDUNG 40: E-BANKING LOGIN PC ...

SMS-Gateway

Für den Versand der zu generierenden SMS-Tokens, welche unter anderem für mTANs verwendet werden, wurde ein externer und darauf spezialisierter Dienstleister gesucht.

Nebst dem eigentlichen SMS-Service, welcher auch eine API für verschiedene Script- und Programmiersprachen anbieten musste, sollte der Versand möglichst kostengünstig und zentral erfolgen können.

Die Generierung und der Versand der SMS-Token sollten möglichst einfach und effizient sowie eigenständig sein. Eine eindeutige Zuordnung des Tokens sowie dessen Verifizierung muss jederzeit gewährleistet sein.

Diese Anforderungen wurden durch den Dienstleister ASPSMS.COM erfüllt. Die entsprechende Webpräsenz kann über folgenden Link erreicht werden: www.aspsms.com

Allgemeine Funktionsübersicht

Vordergründige Funktion:

- Versand von Text-SMS
- Generierung und Versand von SMS-Token
- Verifizierung von SMS-Token

Funktionen im Kontext des Angriffs:

- Keine: Stützend auf den allgemeinen Geschäftsbedingungen des Dienstleisters, wird der SMS-Versand lediglich für Demozwecke beansprucht. Eine entsprechende Rücksprache mit dem Dienstleister hat diesbezüglich stattgefunden. Die Manipulation eines SMS wird durch die SMS-Applikation direkt auf dem Gerät vorgenommen.

User-Interface

ABBILDUNG 41: ÜBERSICHTSSEITE

Command & Control-Plattform

Um einerseits eine Übersicht über die infizierten Smart-Devices zu erhalten, auf welchen die beiden Teilprodukte (SMS-Applikation und Webbrowser) installiert wurden und um andererseits die befallenen Smart-Devices steuern zu können, wurde eine Command & Control-Plattform entwickelt.

Die Seite ist über folgenden Link erreichbar: https://evil.sevenbit.ch (Kennwort: bfh)

Weiter sollte es mittels eines Logs möglich sein, die modifizierten, sowie injizierten Transaktionen inklusive dem Zeitpunkt der jeweiligen Transaktion sowie dem erzielten Gesamtertrag pro Smart-Device aufzulisten.

Allgemeine Funktionsübersicht

Vordergründige Funktion:

- Infektionsübersicht inklusive Mobiltelefonnummer und Installationsdatum
- Command & Control der Smart-Devices
- Transaktionslog der modifizierten & injizierten Transaktionen

Funktionen im Kontext des Angriffs:

- Der Angriff läuft autonom auf dem jeweiligen Smart-Device. Die Plattform kann den Angriff jedoch anhand des Logs mitverfolgen sowie per SMS-Commands steuern.

User-Interface

Abbildung 42: Übersichtseite

Funktionsbeschrieb der Teilprodukte

Allgemeines

Nach dieser ersten Übersicht über die Teilprodukte beziehungsweise über die Komponenten, sowie deren vordergründige Funktionen und den Funktionen im Kontext des Angriffs, folgt nun ein detaillierter Beschrieb jeder einzelnen Komponente.

Benutzer

Überblick

Nebst den bisherigen Ausführungen zum Benutzer (siehe Abschnitt „Benutzer“), erfolgt die Kommunikation zwischen dem Benutzer und dem Smart-Device in der Regel mittels Gesten über das Display oder mittels Spracheingabe.

Beim Display handelt es sich dabei um eine Komponente, welches auf Berührungen reagiert, auch als ‚haptic display’ oder Touchscreen bekannt. Bei der Spracheingabe um eine Software, welche die gesprochenen Wörter als Befehle interpretiert.

Die ausgehende Kommunikation kann, nebst der klassischen Ausgabe über das Display, mittels Vibration des Gerätes, durch eine Audioausgabe oder durch Blicken vorhandener LEDs erfolgen.

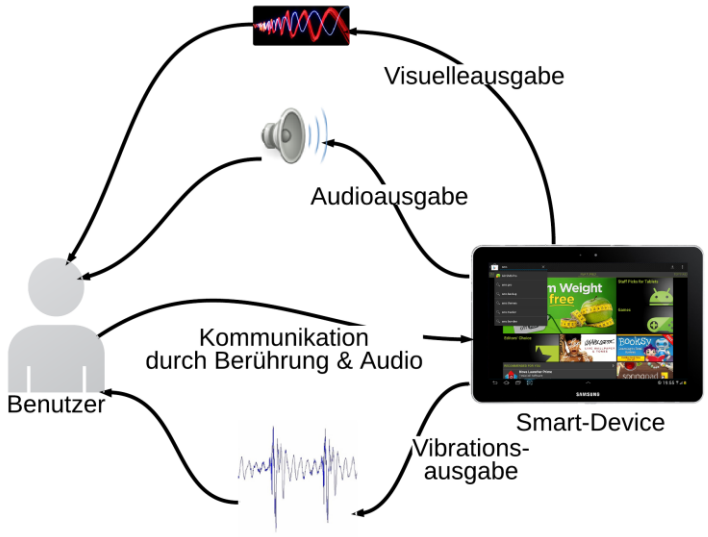

ABBILDUNG 43: KOMMUNIKATION ZWISCHEN BENUTZER & SMART-DEVICE

SMS-Applikation: SMS Buddy

Überblick

Die SMS-Applikation SMS Buddy basiert auf der Open-Source-Lösung SMS Popup und übernimmt all deren Funktionen:

- Benachrichtigung per Popup-Dialog bei eingehenden Nachrichten
- Schnelles Bearbeiten der Nachrichten (löschen, beantworten, schliessen der Nachricht)
- „text-to-speech“, ermöglicht das Vorlesen der Nachricht
- direktes Beantworten der Nachricht mittels Sprachkommando
- Benachrichtigung kann individuell angepasst werden (Audio, Vibration, LED, Benachrichtigung im Infobereich
- Benachrichtigung kann pro Kontakt angepasst werden
- Wiederholung der Benachrichtigung kann angepasst werden
- optionaler „privacy mode“, unterdrückt die Empfängernummer sowie die jeweilige Nachricht

Die frei verfügbare Applikation wurde um einige Funktionen erweitert, ohne jedoch die zur Ausführung benötigten Rechte zu verändern. Der SMS Buddy soll vordergründig, nebst den Funktionen welche bereits SMS Popup bietet, eine Backup-Funktion anbieten, um die vorhandenen SMS auf Knopfdruck zu sichern. Der Button löst jedoch keine Sicherung der SMS aus, sondern ein Backup der Datenbank des Web Buddy Webbrowsers.

Im Kontext des Angriffs fängt SMS Buddy eine eingehende SMS vor der Standard-SMS-Applikation ab, analysiert diese auf bekannte Muster wie z.B. mTAN mit Transaktionsdetails. Bei einem SMS mit Transaktionsdaten, wird dieses, je nach Typ (modifizierte oder injizierte Transaktion) verarbeitet: Entweder so manipuliert, dass der Empfänger glaubt, es sei die Bestätigung seiner ausgelösten Transaktion oder dann, falls es sich um eine eingeschleuste, injizierte Transaktion handelt, werden die Daten (das mTAN) ausgelesen, an den Webbrowser übergeben und das SMS danach gelöscht.

Handelt es sich beim eingehenden SMS um einen, in SMS Buddy implementierten SMS-Befehl, wird dieser entsprechend ausgewertet, an den Web Buddy übergeben und die Nachricht gelöscht. Das Perfide: Der Benutzer bemerkt von alledem nichts.

Nebst den im Abschnitt „SMS-Applikation: Allgemeine Funktionsübersicht“ aufgeführten Funktionen, wurde der Funktionsumfang wie folgt erweitert:

- empfangene SMS gezielt löschen
- empfange SMS gezielt manipulieren
- Verarbeitung von SMS-Befehlen
- Befehle an Webbrowser weiterleiten
- Transaktionsinformationen vom Webbrowser entgegennehmen
- Sicherung der Datenbank des Webbrowsers auslösen

User-Interface

Die Applikation verfügt über mehrere User-Interfaces:

- Lizenzvereinbarung
- eingehendes SMS inkl. Buttons (schliessen, löschen, beantworten)
- Beantworten eines SMS inkl. Buttons (schliessen, Vorlagen)
- Hauptmenü mit Buttons (Einstellungen, Sicherung)

Lizenzvereinbarung

Beim erstmaligen Starten der Applikation wird die Lizenzvereinbarung angezeigt. Diese kann angenommen oder abgelehnt werden, wobei bei letzterem die Deinstallation der Applikation veranlasst werden kann.

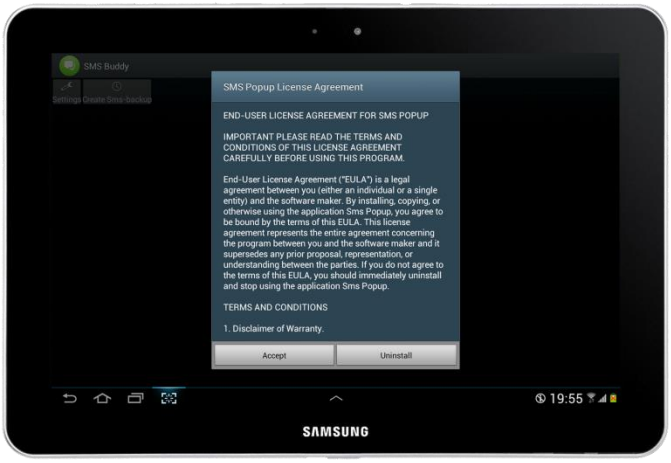

ABBILDUNG 44: LIZENZVEREINBARUNG

Eingehendes SMS

Bei einem eingehenden SMS wird der Inhalt, sowie dessen Absender (als Kontakt) und den Zeitpunkt des Empfangs in einem Popup angezeigt. Der Dialog kann geschlossen, die SMS gelöscht oder direkt beantwortet werden.

Abbildung 45: Eingehendes SMS

Beantworten eines SMS

Über eine, in das Popup integrierte Eingabemaske, lässt sich eine SMS direkt beantworten und an den Absender zurückschicken. Zudem können die Texte diktiert oder aus einer bestehenden Vorlage übernommen werden. Weiter lässt sich auch dieser Dialog per Knopfdruck schliessen.

Abbildung 46: Direktantwort

Hauptmenü

Nach der Installation und der Lizenzvereinbarung erscheint das Hauptmenü. Über dieses lassen sich die Einstellungen öffnen, sowie die Datenbank des Webbrowsers sichern. Letztere Funktion wird hinter dem Button zur Sicherung der SMS versteckt.

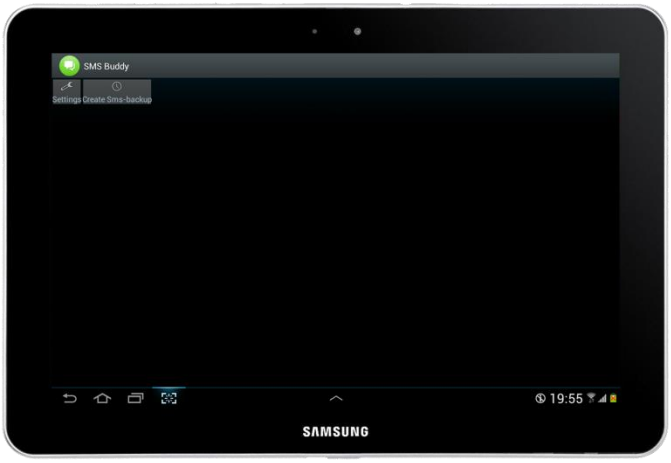

Abbildung 47: Hauptmenü

Funktionen & Programmablauf

Verhalten

SMS Buddy operiert im Verbund mit dem Webbrowser Web Buddy, das heisst, der Webbrowser sendet gezielt Informationen an SMS Buddy. Der Austausch erfolgt über sogenannte Intents: Diese stellen einen Nachrichtendienst dar, mit dem es möglich ist, Activities und Services zu starten und Broadcast-Receiver über Ereignisse zu informieren. Im diesem Fall können dies beispielsweise Transaktionsinformationen sein, um eine Transaktionsbestätigung via SMS-Nachricht abzufangen.

Sendet hingegen der Webbrowser keine entsprechenden Transaktionsdaten an SMS Buddy, so verhält sich SMS Buddy unscheinbar und verrichtet, analog zum SMS Popup seinen Dienst.

Eingehende SMS werden, wie bei der systeminternen SMS-Applikation, in die zentrale Datenbank abgelegt und verwaltet. Bei einer Manipulation wird die Nachricht durch SMS Buddy abgefangen, manipuliert und danach in die Datenbank geschrieben. Die nachfolgende Abbildung soll dies verdeutlichen.

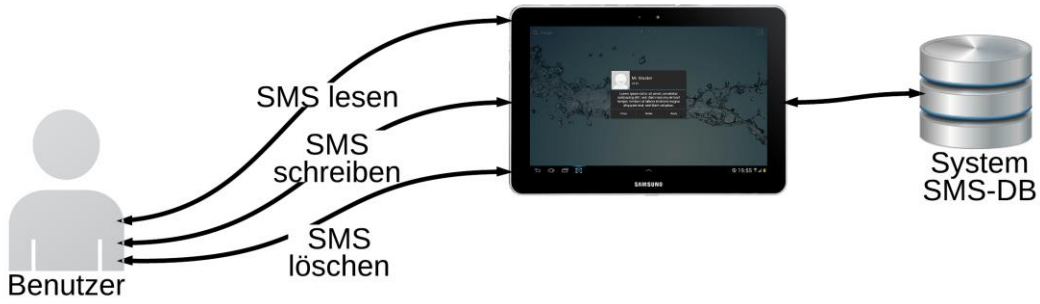

ABBILDUNG 48: SMS BUDDY NACHRICHTENEMPFANG

Empfangene SMS gezielt verändern

Sendet der Webbrowser einen Intent mit der Information, dass eine Transaktion manipuliert (verfälscht) wurde, so analysiert SMS Buddy alle eingehenden SMS-Nachrichten auf deren Inhalt, ob eine Übereinstimmung mit den Transaktionsdaten vorliegt. Liegt eine gewisse Übereinstimmung vor, so wird der Inhalt der SMS entsprechend angepasst.

Dadurch ist es möglich, eine Transaktion vor dem Benutzer zu verschleiern: Der Inhalt wird also so angepasst, dass er mit den Originaldaten, welche der Benutzer eingegeben hat, eins zu eins übereinstimmen.

Ablauf im Überblick

Das folgende Sequenzdiagramm soll dies noch einmal verdeutlichen:

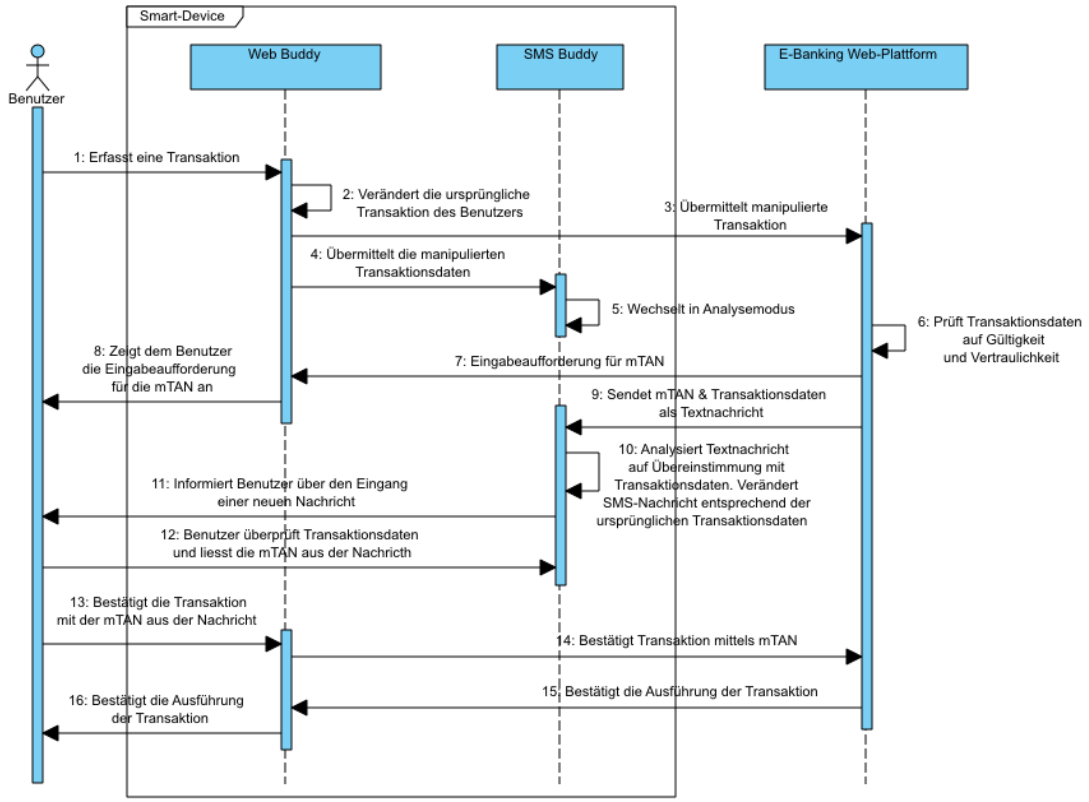

ABBILDUNG 49: E-BANKING SESSION MIT MANIPULIERTER TRANSAKTION

Empfangene SMS gezielt löschen

Sendet der Webbrowser einen Intent mit der Information, dass eine Transaktion injiziert wurde und eine entsprechende mTAN erwartet wird, so analysiert SMS Buddy alle eingehenden SMS-Nachrichten auf deren Inhalt, ob eine Übereinstimmung mit den Transaktionsdaten vorliegt. Liegt eine gewisse Übereinstimmung vor, so wird die mTAN aus der Nachricht extrahiert und die Nachricht umgehend gelöscht. Die mTAN wiederum wird an den Webbrowser übermittelt.

Ablauf im Überblick

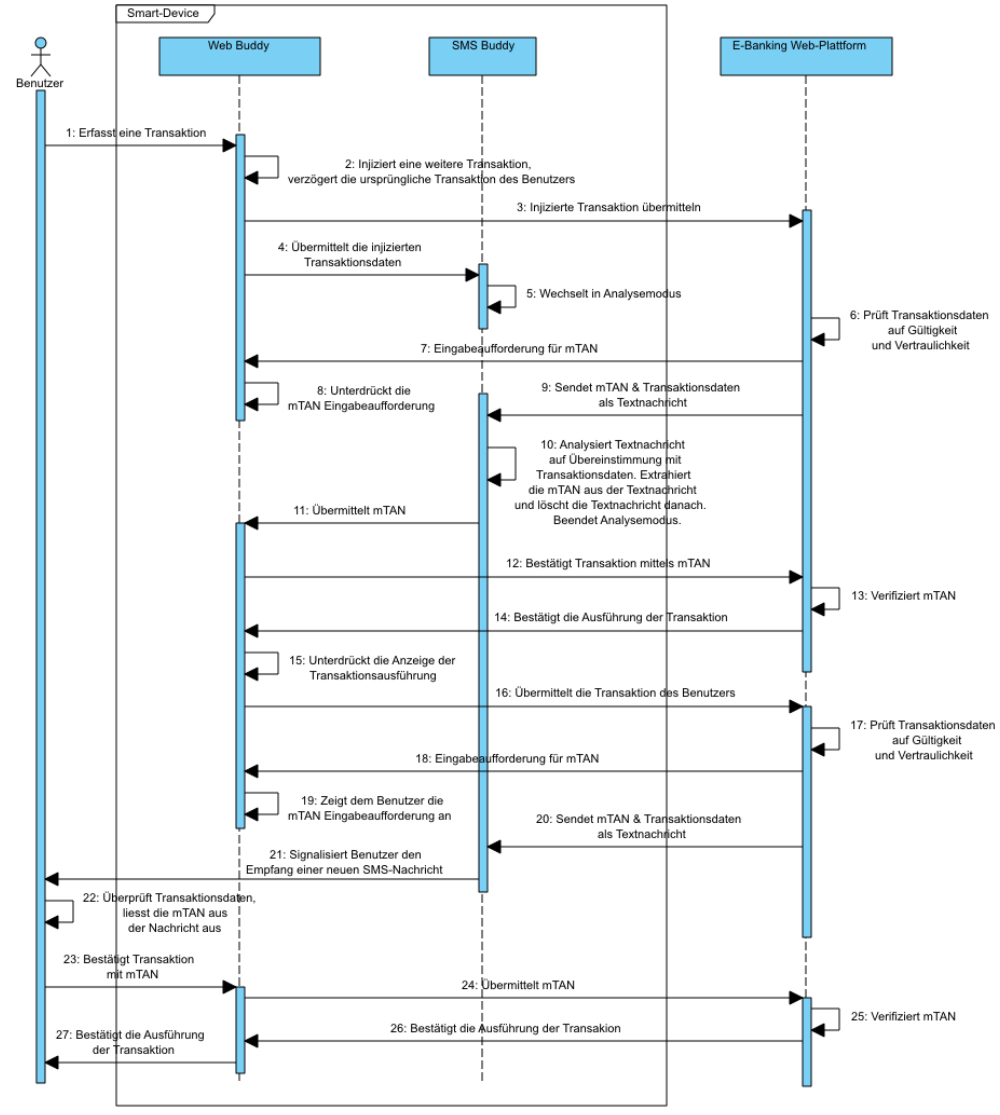

Abbildung 50: E-Banking Session mit injizierter Transaktion

SMS-Befehle verarbeiten

SMS Buddy ist in der Lage vordefinierte SMS-Befehle zu verarbeiten und so zum Beispiel das Verhalten des Web Buddy zu beeinflussen.

Folgende Befehle sind implementiert:

- /@addReceiver: Mit diesem Befehl können weitere Zahlungsbegünstige zur Datenbank des Web Buddy hinzugefügt werden. Diese Empfänger können danach bei einer Injektion oder bei einer Manipulation einer Transaktion zufällig verwendet werden.
- /@disableReceiver: Dieser Befehl setzt einen Begünstigten auf ‚inaktiv'.
- /@enableReceiver: Setzt einen entsprechenden Begünstigten wieder auf ‚aktiv'.
- /@enableCmd: Funktionen können mit diesem Befehl aktiviert werden.
- /@disableCmd: Das Gegenstück zu ‚enableCmd', womit bestimmte Befehle deaktiviert werden können.
- /@clearDB: Die vom Webbrowser administrierte Datenbank kann mit diesem Befehl gelöscht und die böswilligen Aktivitäten des Web Buddys deaktiviert werden.

Folgende Funktionen können aktiviert (enableCmd) oder deaktiviert (disableCmd) werden:

- fraudMode: Ist der Fraud-Mode deaktiviert, verhält sich der Web Buddy wie ein gewöhnlicher Browser und nimmt keine Manipulationen vor. Standardmässig ist der Fraud-Mode aktiviert.
- injectionMode: Ist der Injection-Mode aktiviert, werden Transaktionen nur noch injiziert und nicht mehr modifiziert. Standardmässig ist der Injection-Mode deaktiviert. Der Web Buddy wechselt automatisch zwischen Injektion und Modifikation.

Berechtigungen

Die SMS Buddy verwendet dieselben Berechtigungen, wie SMS Popup. Konkret handelt es sich dabei um folgende Berechtigungen:

- SMS senden
- SMS oder MMS bearbeiten
- SMS empfangen
- MMS empfangen
- SMS oder MMS lesen
- vertrauliche Protokolldaten lesen
- Kontakte lesen
- Telefonstatus und Identität abrufen
- aktive Apps abrufen
- Bildschirmsperre deaktivieren
- Ruhezustand des Tablets deaktivieren
- Ruhezustand des Telefons deaktivieren
- Vibrationsalarm steuern
- Anrufliste lesen

Webbrowser: Web Buddy

Überblick

Beim eigens entwickelten Webbrowser Web Buddy, handelt es sich auf den ersten Blick um eine einfache Applikation, um schnell und unkompliziert eine Webseite anzeigen zu lassen. Der Browser kann natürlich auch für E-Banking genutzt werden. Entsprechend gestaltet, könnte dieser aber auch als eigenständige und vertrauenswürdige E-Banking-Applikation beworben werden. Zahlreiche bestehende E-Banking Lösungen für Smart-Devices kommunizieren im Hintergrund über eine Webschnittstelle, da eine Implementierung relativ einfach und nahezu plattformunabhängig erfolgen kann.

Im Kontext des Angriffs agiert der Webbrowser als zentrale, im Hintergrund tätige Applikation, welche den ganzen Angriff überhaupt erst ermöglicht. Dies beginnt bereits beim Start der Applikation: Es wird um ein angeblich besseres Nutzererlebnis und höhere Sicherheit im E-Banking zu bieten empfohlen, den SMS Buddy zusätzlich zu installieren, sofern dieser noch nicht auf dem Gerät vorhanden ist.

Ist der Webbrowser gestartet, kann eine beliebige Webseite aufgerufen werden. Es können sowohl Cookies, als auch JavaScript und self-signed SSL-Zertifikate verwendet werden. Ein Handler um Links zu E-Mailadressen (startet Mail-Applikation) und Telefonnummern (startet Telefon-Applikation) zu verarbeiten, sowie der Download von PDFs wurde ebenfalls implementiert.

Werden bestimmte Webseiten aufgerufen, wie zum Beispiel unsere E-Banking Web-Plattform oder auch Bing.com bzw. Google.com, reagiert der Browser und verändert spezifisch die Webseiten. Der Benutzer merkt von alldem nichts. Die Webseiten werden mittels JavaScript on-the-fly analysiert und entsprechend modifiziert. Während bei den beiden Suchmaschinen lediglich ein Redirect von Google.com zu Bing.com durchgeführt und bei Bing das Logo durch jenes von Google ersetzt wird, sind die Modifikationen bei der entwickelten E-Banking Web-Plattform wesentlich gravierender: Es wird nicht nur die Mobiltelefonnummer beim ersten Login abgefragt und an die Command-Plattform weitergeleitet, sondern auch unbemerkt Transaktionen injiziert und modifiziert sowie, der Bankauszug auf dem Smart-Device entsprechend verfälscht angezeigt.

Nebst den im Abschnitt „Webbrowser: Allgemeiner Funktionsbeschrieb“ aufgeführten Funktionen, wurde der Funktionsumfang unter anderem wie folgt erweitert:

- Anzeigen von Werbung (für SMS Buddy)
- Verarbeitung von E-Mail- & Telefonlinks sowie PDF-Download
- Verarbeitung von SMS-Befehlen (über SMS Buddy)
- Daten an SMS Buddy weiterleiten
- Sicherung der Datenbank durchführen

User-Interface

Die Applikation verfügt über mehrere User-Interfaces:

- Splashscreen
- Hauptscreen mit und ohne Werbung
- Validierungsscreen

Splashscreen

Vor jedem Start erscheint ein sogenannter Splashscreen mit dem Logo der Applikation. Im Hintergrund könnten beispielsweise grössere Grafiken vorgeladen werden. Es wäre aber auch denkbar, dass im Hintergrund weitere kollaborierende Applikationen gestartet werden.

ABBILDUNG 51: SPLASHSCREEN

Hauptscreen mit Werbung

Beim erstmaligen Starten nach der Installation erscheint, sofern der SMS Buddy nicht vorhanden ist, ein Werbehinweis auf die genannte SMS-Applikation. Wird dieser geschlossen und die Applikation nicht installiert, erscheint die Werbung bei jedem 5. Start.

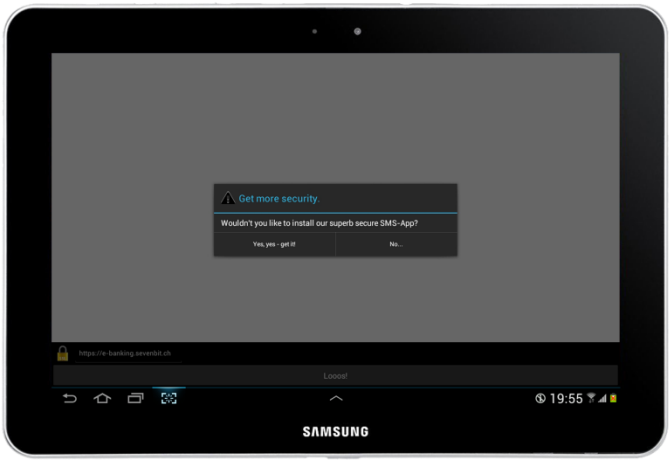

Abbildung 52: Hauptscreen mit Werbung

Hauptscreen ohne Werbung

Der Webbrowser erscheint mit einem einfachen und zweckmässigen User-Interface. Dieses bietet, nebst einer Adresszeile für die gewünschte URL, eine pseudo Anzeige für SSL – also, ob die Verbindung verschlüsselt ist – sowie ein Button, um die eingegebene URL zu laden. Wird eine unzulässige URL eingeben, beispielsweise ohne oder falsch geschriebener Protokollangabe oder unzulässigen Zeichen (gemäss URI-Standard), wird ein entsprechender Hinweis angezeigt.

Abbildung 53: Hauptscreen ohne Werbung

Prozessdialog

Bei jeder über die E-Banking Web-Plattform initiierten Transaktion, erscheint im Web Buddy ein pseudo Prozessdialog, in dem angezeigt wird, dass eine vermeintliche Validierung durchgeführt wird.

Je nach Modus, also ob eine Transaktion modifiziert oder eine weitere injiziert wird, ist der Ablauf im Hintergrund anders: Die Webseite wird bei einer injizierten Transaktion in den Cache geladen, bis alle nötigen Informationen gesammelt, die mTAN für die jeweilige Transaktion eingetroffen und der Vorgang abgeschlossen ist. Damit kann die Webseite im Hintergrund neu gerendert werden, ohne, dass sich die Anzeige verändert. Es wird somit eine Transaktion eingeschleust, ohne dass dies der Benutzer bemerkt.

Bei einer modifizierten Transaktion erscheint derselbe Prozessdialog, die Webseite wird jedoch nicht in den Cache geladen. Der Dialog verschwindet beim Übermitteln der Transaktionsdaten, da nicht auf das Eintreffen einer mTAN gewartet werden muss.

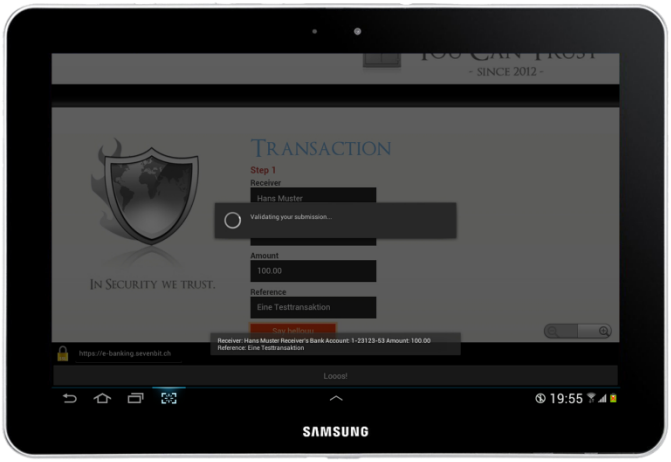

ABBILDUNG 54: PROZESSDIALOG

Funktionen & Programmablauf

Web Buddy operiert im Verbund mit dem SMS Buddy, das heisst, SMS Buddy reagiert auf bestimmte Nachrichten, welche vom Web Buddy gesendet werden. Umgekehrt kann Web Buddy sein Verhalten ändern, falls entsprechende Kommandos von SMS Buddy gesendet werden.

Normales Browserverhalten

Sofern der Fraud-Mode deaktiviert ist, womit beispielsweise Transaktionen manipuliert werden, verhält sich Web Buddy unscheinbar und verrichtet wie ein normaler Webbrowser seinen Dienst.

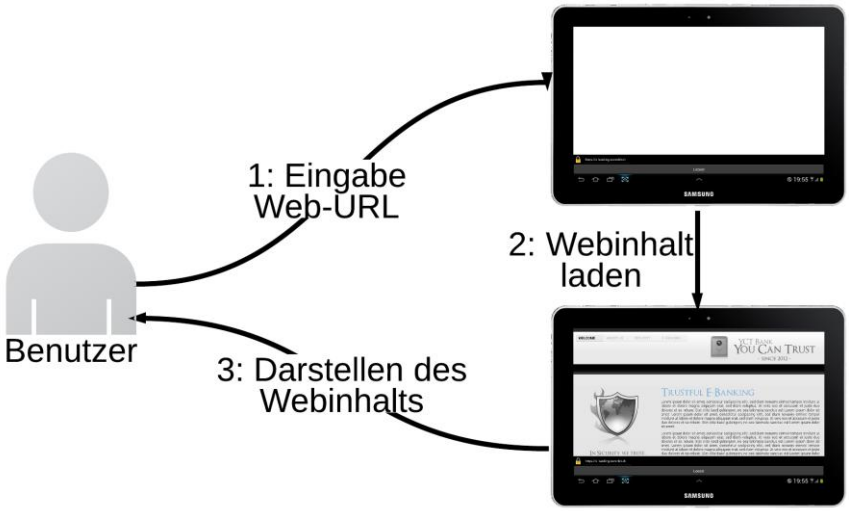

Abbildung 55: Web Buddy Standardverhalten

Fraud-Mode: Transaktion manipulieren & injizieren

Wurde die Funktionalität für die Manipulation von Transaktion aktiviert, beginnt Web Buddy seine tatsächlich vorgesehene Tätigkeit im Kontext des Angriffs aufzunehmen.

Sobald erkannt wird, dass eine E-Banking Session gestartet wurde und der Benutzer damit begonnen hat, Transaktionen zu erfassen, wird Web Buddy selbständig versuchen, Transaktionen zu verfälschen beziehungsweise, Transaktionen zu injizieren.

Manipulierte Transaktion

Web Buddy erkennt, dass der Benutzer soeben eine Transaktion erfasst hat und diese absenden möchte. Dieser nimmt die erfasste Transaktion entgegen, verändert die Transaktionsdaten wie Empfängername, Empfängerkonto, zu den zu überweisenden Betrag und senden die Transaktion an die E-Banking Web-Plattform weiter. Damit eine Transaktion eindeutig identifiziert werden kann, wird das Feld für die Referenznummer, wie sie beim E-Banking verwendet werden (roter Einzahlungsschein: Zahlungszweck, oranger Einzahlungsschein: ESR/ESR+), mit einem zufällig generierten, eindeutigen Wert gefüllt.

Gleichzeitig werden die Transaktionsdaten an SMS Buddy übermittelt, damit beim Eintreffen der SMS für die Transaktionsverifikation, die veränderten Werte mit den ursprünglich vom Benutzer erfassten Transaktionsdaten ausgetauscht werden können. Hierbei ist das Ziel, dass der Benutzer nichts von der manipulierten Transaktion bemerkt und somit die mTAN zur Verifikation selbstständig eingibt. Per Knopfdruck wird die Transaktionsbestätigung an die E-Banking Web-Plattform übermittelt.

Injizierte Transaktion

Der Benutzer hat eine Transaktion erfasst und sendet diese an die E-Banking Web-Plattform. Web Buddy nimmt die erfasste Transaktion entgegen, injiziert selbstständig eine weitere Transaktion und sendet diese an die E-Banking Web-Plattform weiter.

Gleichzeitig werden die injizierten Transaktionsdaten an SMS Buddy gesendet, damit beim Eintreffen der SMS für die Transaktionsverifikation, die mTAN ausgelesen, an Web Buddy übermittelt und die Nachricht gelöscht werden kann. Web Buddy wiederrum nimmt die mTAN entgegen, verifiziert die Transaktion und sendet danach die vom Benutzer erfasste Transaktion an die E-Banking Web-Plattform ab. Hierbei ist das Ziel, dass der Benutzer nichts von der injizierten Transaktion bemerkt.

Das injizieren, beziehungsweise manipulieren von Transaktionen wird durch Web Buddy selbstständig durchgeführt. Das heisst, dass bei jeder dritten Transaktion eine Transaktion injiziert wird und jede zweite Transaktion automatisch manipuliert wird.

Dieses Verhalten kann über das Setzen des Injection-Modes per SMS-Commands gesteuert werden.

Anzeigen der Transaktionsübersicht

Der Benutzer hat jederzeit die Möglichkeit eine Übersicht der ausgeführten Transaktionen online anzeigen zu lassen. Web Buddy erkennt dies und verändert die Anzeige so, dass die injizierten Transaktionen nicht aufgelistet und anstelle dieser, die manipulierten Transaktionen mit den vom Benutzer erfassten Transaktionsdaten angezeigt werden. Nur ein Blick auf den Bankauszug über ein alternatives Gerät oder auf den nächsten Bankauszug in Papierform, lässt die gefälschten Transaktionen für den Benutzer sichtbar werden.

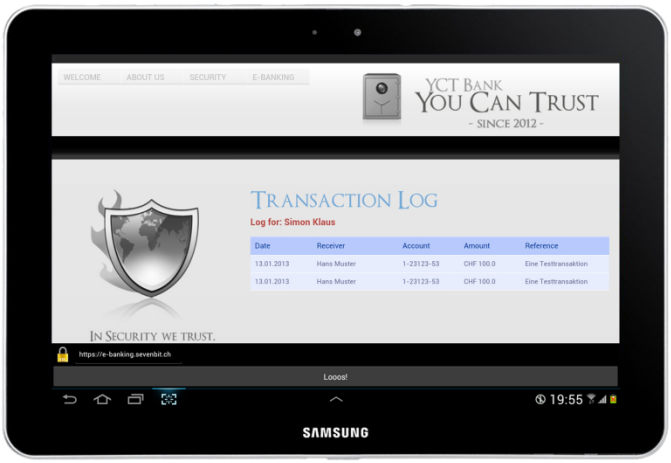

Abbildung 56: Bankauszug Smart-Device

Abbildung 57: Bankzauszug PC

Web Buddy Datenbank-Management

Die Informationen zur Manipulation von Transaktionen bezieht Web Buddy aus einer lokalen Datenbank. Die Datenbank enthält folgende Informationen:

- welche Transaktionen wurden injiziert
- welche Transaktionen wurden manipuliert
- für die Manipulation von Transaktionen werden die Zahlungsbegünstigten verwaltet
- weiter können bestimmte Flags (wie Fraud-Mode, Injection-Mode, etc.) in der Datenbank hinterlegt werden, welche das Verhalten von Web Buddy beeinflussen

Transaktionsdaten werden von Web Buddy selbstständig nachgeführt. Die Verwaltung der Zahlungsbegünstigten erfolgt jedoch in Zusammenarbeit mit SMS Buddy.

Dazu empfängt SMS Buddy bestimmte SMS-Nachrichten, welche als Kommandos interpretiert werden. Über diese Kommandos können beispielsweise Empfänger hinzugefügt, aktualisiert oder auch deaktiviert werden. Weiter besteht die Möglichkeit, dass die Datenbank vom infizierten Gerät per SMS-Kommando komplett entfernt werden kann, um die Spuren einer Attacke zu verwischen.

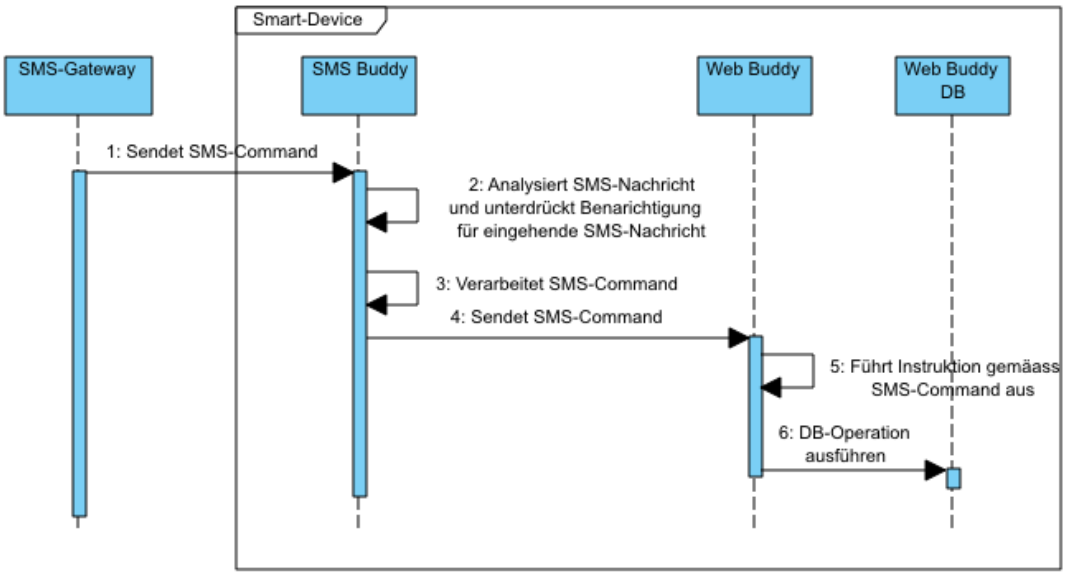

ABBILDUNG 58: VERARBEITUNG SMS-COMMANDS

Datenbanksicherung

Das Erstellen einer Sicherung der Datenbank wurde ebenfalls implementiert. Die Sicherung wird auf die SD-Karte des Smart-Devices geschrieben. Von da aus könnte nun Web Buddy die gesicherten Daten an einen entsprechenden Server zur Auswertung weiter senden. Eine solche Funktionalität wurde nicht implementiert.

SMS-Commands

Neben den bereits erwähnten Kommandos, welche unter anderem dazu dienen, um Zahlungsbegünstigte zu verwalten oder die gesamte Datenbank zu löschen, können weitere SMS-Commands das Verhalten beeinflussen. Konkret heisst dies, dass die Funktionalität zur Manipulation von Transaktionen mittels Kommandos ein- beziehungsweise ausgeschaltet werden kann.

Abbildung 59: Command & Control

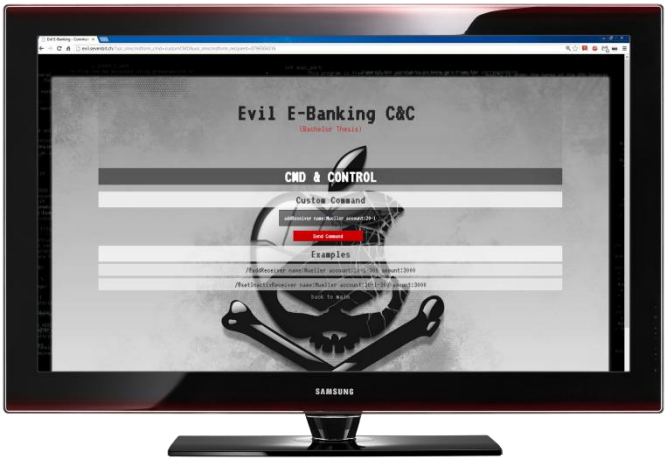

Abbildung 60: Custom SMS-Commands

Werben für SMS Buddy

Wie eingangs erwähnt, kollaboriert Web Buddy mit SMS Buddy. Damit der Benutzer früher oder später SMS Buddy installiert, wird bei jedem fünften Start von Web Buddy der Benutzer damit umworben, die Applikation SMS Buddy zu installieren.

Hat der Benutzer SMS Buddy auf dem Smart-Device installiert, wird die Anzeige der „Werbung“ deaktiviert.

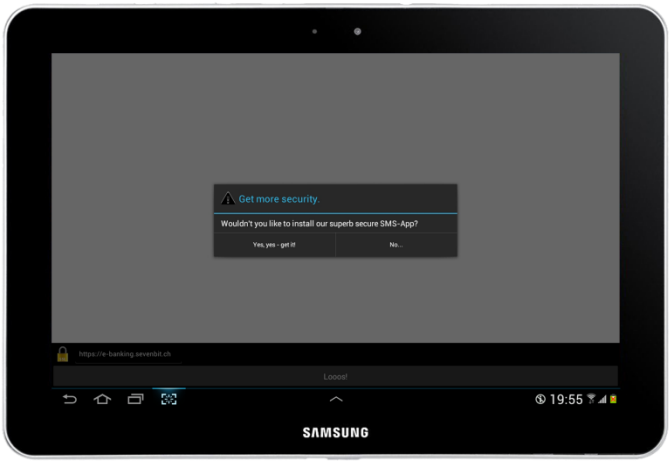

ABBILDUNG 61: WERBUNG

Berechtigungen

Web Buddy wurden folgende Berechtigungen vergeben:

- Schreiben auf SD-Karte
- Informationen zum Netzwerk beziehen
- Netzwerkverbindungen öffnen

E-Banking Web-Plattform

Überblick

Die entwickelte E-Banking Web-Plattform soll, als Demosystem, die nötigsten Funktionen einer klassischen, modernen E-Banking-Lösung bereitstellen.

Im Kontext des Angriffs wird weder die serverseitige Applikation angegriffen, noch die Übertragung der Daten manipuliert. Somit können die Daten – wie dies bei allen E-Banking Web-Plattformen der Fall ist – natürlich auch verschlüsselt übertragen werden. Die Manipulation der Webseiten-Elemente, wie zum Beispiel Formularfelder oder Buttons, aber auch das Ersetzen von Grafiken, erfolgt on-the-fly im entwickelten Webbrowser.

Zur Verdeutlichung: Es werden keine Benutzerrechte und keine Lücken in den Serverscripts ausgenutzt und keine Verschlüsselung gebrochen. Einziger Unterschied zu einem echten Szenario: Da wir in unserem Beispiel mit einem kostenlosen self-signed SSL-Zertifikat arbeiten, wurde bei der Implementation die Zertifikatüberprüfung deaktiviert. Dies hat aber keinen Einfluss auf die Glaubwürdigkeit des Szenarios, da die Verschlüsselung wie erwähnt keine Rolle spielt und für den Angriff belanglos ist.

Die Web-Plattform bietet, nebst einigen Texten zur Fantasiebank „YCT Bank – You Can Trust“ ein Login mittels Vertragsnummer und persönlichem Passwort. Der Ablauf ist bereits bekannt: Bei korrektem Ausfüllen wird eine mTAN an die, dem Vertrag hinterlegte Mobiltelefonnummer geschickt. Dieses wird durch den SMS-Gateway generiert und, nach Eingabe auf der Webseite, wieder durch den Gateway über SOAP verifiziert. Nur wenn diese Phase erfolgreich abgeschlossen wurde, erhält man Zugriff auf das E-Banking-System.

Damit wurde die E-Banking Session eröffnet, so dass man während den Tätigkeiten auf der Web-Plattform eingeloggt bleibt und dementsprechend unter anderem seinen persönlichen Bankauszug ansehen oder Transaktionen tätigen kann. Für jede Transaktion, bestehend aus dem Namen und der Kontonummer des Empfängers, sowie dem zu überweisenden Betrags und einer Referenz (z.B. Zugunsten von oder ESR/ESR+), wird wiederum ein mTAN inklusive Transaktionsdetails generiert und an das Smart-Device gesendet.

Dieses kann anschliessend vom Benutzer in das Bestätigungsformular, welches noch einmal die Transaktionsdetails beinhaltet, eingegeben werden. Der SMS-Gateway verifiziert wiederum sowohl mTAN, Mobiltelefonnummer und ein zufälliger, eindeutiger Wert, um mehrere mTANs an denselben Empfänger unterscheiden zu können. Weiter wurde die Ausgabe eines Bankauszugs implementiert, der eine Übersicht über alle getätigten Zahlungen bietet. Der Einfachheit halber sind nur ausgehende Zahlungen aufgeführt. Zudem kann der Auszug in Druckform gebracht werden.

User-Interface & Ablauf

Die Web-Plattform verfügt über mehrere User-Interfaces:

- Willkommensseite
- Loginseite
- Transaktion auslösen
- Transaktionsübersicht (Bankauszug)

Gleichzeitig entspricht der Ablauf einer E-Banking Session in etwa der hier aufgeführten Reihenfolge der User-Interfaces.

Willkommensseite Desktop-PC

Die Webseite kann sowohl von einem gewöhnlichen Desktop-PC, als auch einem Smart-Device aufgerufen werden. Die Seite erscheint immer im selben Layout.

Abbildung 62: Willkommensseite Desktop-PC

Willkommensseite Smart-Device

Es wurde keine mobil-optimierte Webseite erstellt. Dadurch entsteht einerseits ein positiver Wiedererkennungseffekt und somit auch ein gewisses Vertrauen. Andererseits ist dadurch auch der volle Funktionsumfang auf jedem Endgerät gegeben.

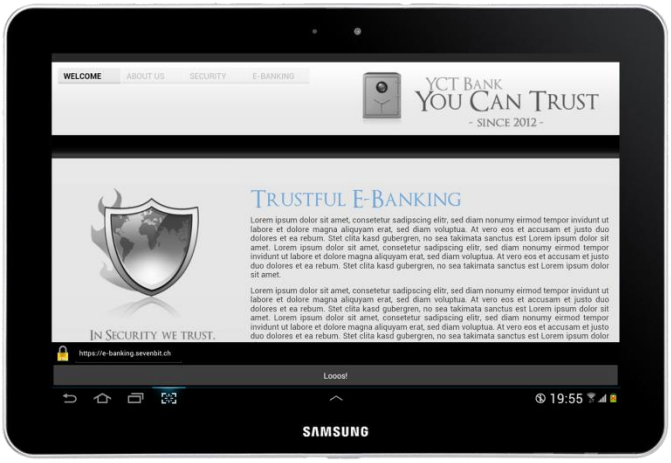

ABBILDUNG 63: WILLKOMMENSSEITE SMART-DEVICE

Loginseite

Normale und manipulierte Loginseite

Mittels Vertragsnummer und persönlichem Passwort, kann sich der Benutzer anmelden. Bereits hier kommt eine erste Manipulation zum Zug: Es wird ein weiteres Formularfeld für die Mobiltelefonnummer hinzugefügt, welche nach der Eingabe zusammen mit der Vertragsnummer und dem persönlichen Passwort an den Webbrowser und danach an die Command & Control-Plattform übergeben wird.

ABBILDUNG 64: NORMALE UND MANIPULIERTE LOGINSEITE

Eingabe mTAN

Nach korrekter Eingabe der Credentials wird ein SMS-Token (mTAN) an die hinterlegte Mobiltelefonnummer versendet. Diese ist 10 Minuten gültig und wird durch den SMS-Gateway generiert sowie verwaltet. Dem E-Banking Benutzer wird die mTAN direkt auf das Smart-Device zugestellt.

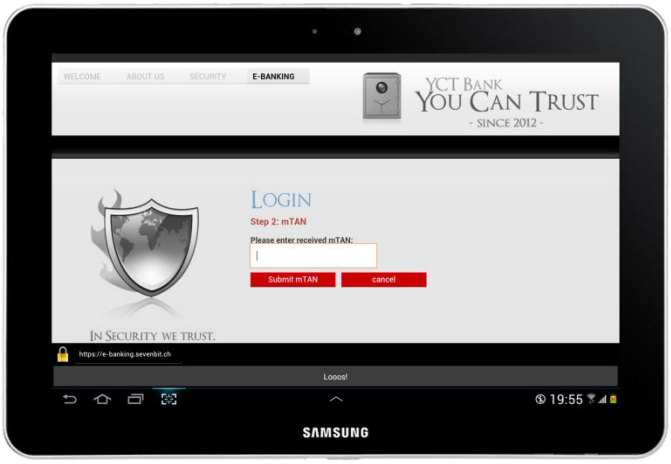

ABBILDUNG 65: EINGABE MTAN

Erfolgreiche Validierung mTAN

Das erhaltene Token wird über das Formular an die Web-Plattform übertragen und dort per SOAP an den SMS-Gateway zur Validierung weitergeleitet. Ist die Antwort positiv, erhält der Benutzer Zugang zu seiner persönlichen E-Banking Plattform.

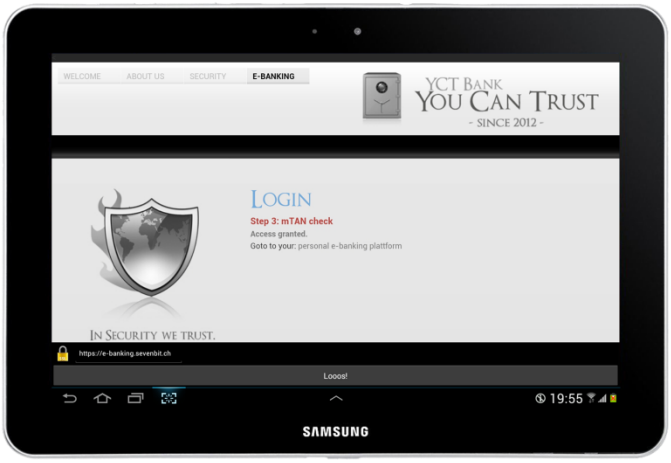

ABBILDUNG 66: ERFOLGREICHE VALIDIERUNG MTAN

Persönliche E-Banking Plattform

War die Antwort des Gateways positiv, erhält der Benutzer Zugang zu seiner persönlichen E-Banking Plattform. Als Begrüssung erscheinen die aktuellen Aktienkurse.

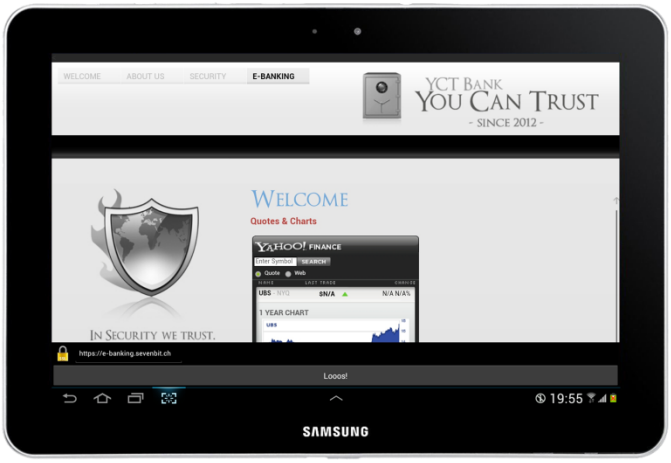

ABBILDUNG 67: PERSÖNLICHE E-BANKING PLATTFORM

Transaktion auslösen

Transaktion starten

Nach erfolgreichem Login kann der Benutzer beliebige Transaktionen auslösen. Dazu werden die entsprechenden Zahlungsangaben eingegeben. Das Formular wird durch den Web Buddy im Hintergrund modifiziert, sodass die Daten nicht direkt an die Web-Plattform, sondern an den Webbrowser übergeben, dort verarbeitet und erst danach weitergeleitet werden.

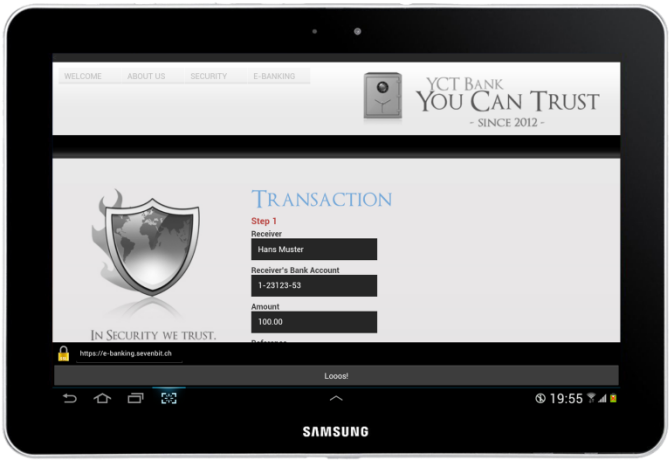

ABBILDUNG 68: TRANSAKTION STARTEN

Prozessdialog

Der Web Buddy löst einen applikationsinternen Prozessdialog aus, um im Hintergrund die Daten verarbeiten zu können. Mehr dazu im Abschnitt „Webbrowser: Web Buddy“.

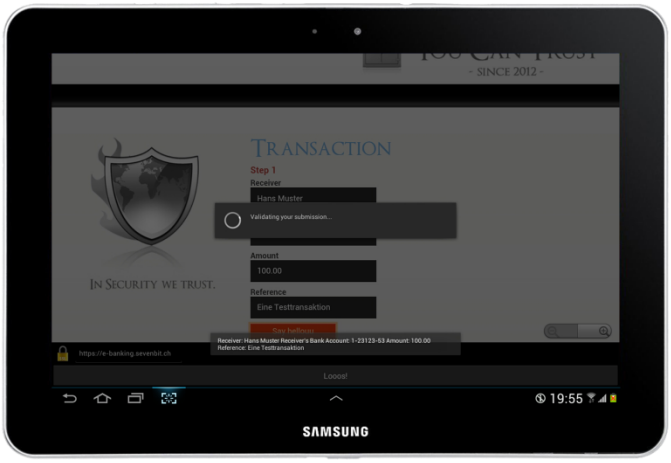

ABBILDUNG 69: PROZESSDIALOG

Eingabe mTAN

Sowohl bei einer injizierten, als auch bei einer modifizierten Transaktion, wird ein Eingabeformular für die mTAN angezeigt. Der per SMS erhaltene Code muss vom Benutzer eingegeben werden. Wie wir bereits gesehen haben: Bei einer injizierten Transaktion werden zwei Transaktionen ausgelöst. Die erste wird im Hintergrund ausgelöst und automatisch bestätigt, die zweite unverändert angezeigt. Bei der modifizierten Transaktion werden die Angaben des SMS entsprechend modifiziert abgespeichert und angezeigt.

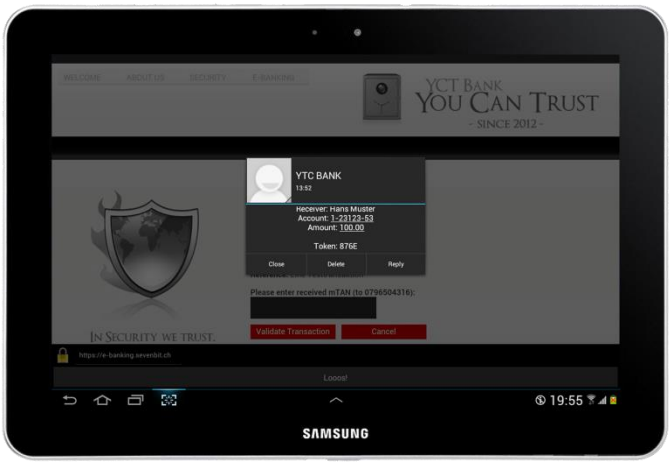

ABBILDUNG 70: EINGABE MTAN

Erfolgreiche Validierung mTAN

Sofern die mTAN korrekt eingegeben wurde, erscheint auf dem folgenden Screen eine Bestätigung. Die Transaktion(en) wurde(n) erfolgreich übertragen, validiert und abgeschlossen.

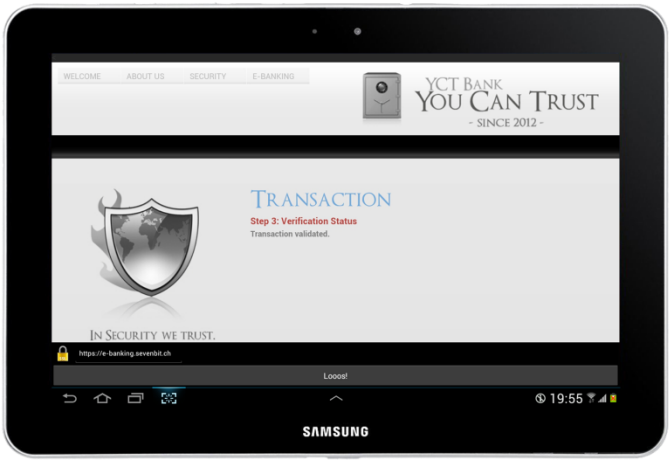

Abbildung 71: Erfolgreiche Validierung mTAN

Transaktionsübersicht

Nebst dem Auslösen einer Transaktion bietet die Web-Plattform die Möglichkeit bereits getätigte Transaktionen anzuzeigen. Web Buddy modifiziert die Ausgabe mit Hilfe einer internen Datenbank, welche alle modifizierten und injizierten Transkationen enthält: Die modifizierten Transaktionen werden durch die ursprünglichen, vom Benutzer gewollten Daten ersetzt, die injizierten ausgeblendet.

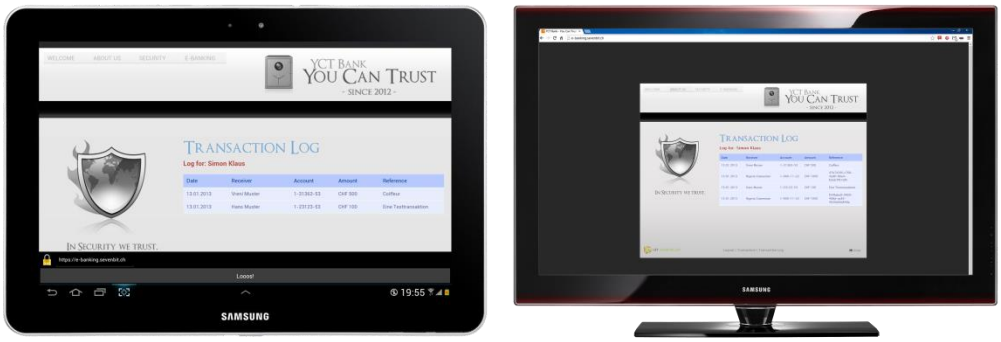

Abbildung 72: Normale und manipulierte Transaktionsübersicht

SMS-Gateway

Überblick

Damit das Szenario so realistisch wie möglich umgesetzt werden konnte, war ein professioneller Gateway zwischen E-Banking Web-Plattform und dem Smart-Device nötig.

Der verwendete SMS-Gateway der Firma ASPSMS.COM bietet dafür zahlreiche Funktionen an:

- Simple SMS
- Multiple SMS
- Delivery Notification
- Alphanumeric Originator
- Binary SMS
- Wap Push
- Arabic & other oriental characters
- Ringtones
- Operator Logos
- vCards
- Message Waiting Indication
- Flashing SMS
- Blinking SMS
- Encrypted Transmission from your host to ASPSMS.COM
- Redundant SMS servers
- Multiple concurrent users
- Multiple SMS accounts on your server
- Balance check

Für das E-Banking-Szenario werden sogenannte SMS-Token benötigt: Dank dem SMS-Token kann sowohl das Login, als auch jede Transaktion über den separaten GSM-Kanal vom Benutzer überprüft werden. Damit entfallen Streichlisten, Karten oder allfällige Zusatzgeräte. Ganz auf der sicheren Seite, darf man sich aber auch mit einem mTAN nicht sein, wie diese Arbeit aufzeigt.

Ein wichtiger Pluspunkt von ASPSMS.COM: Die Token können beispielsweise über SOAP (ursprünglich ein Akronym für Simple Object Access Protocol) mit nahezu jeder Script- und Programmiersprache angefordert, ausgeliefert und überprüft werden.

User-Interface

Übersichtsseite

Um die Dienstleistungen von ASPSMS.COM nutzen zu können, ist eine kostenlose Registrierung nötig. Dadurch erhält man Benutzer-Schlüssel (Userkey) sowie Kennwort.

Mittels dieser Credentials und einem zuvor aufgeladenen Guthaben, kann nun, über Schnittstellen für zahlreiche Programmiersprachen, aber auch mit bereits existierenden Softwarelösungen, auf die zahlreichen Dienstleistungen zugegriffen werden.

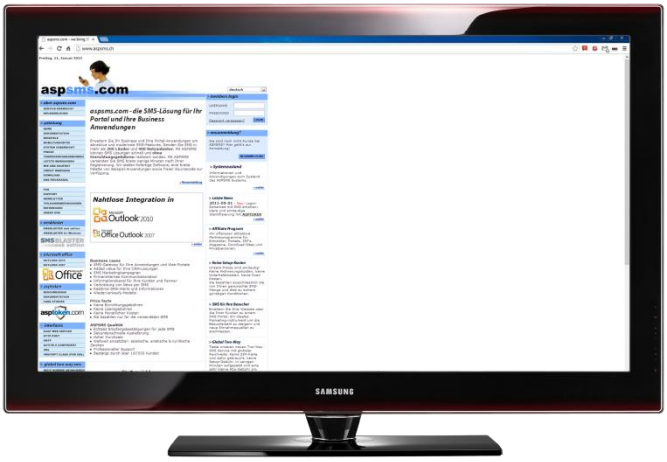

ABBILDUNG 73: ÜBERSICHTSSEITE

Freigeschaltete Absender

Da eine SMS immer einem Absender zugeordnet werden muss, bietet ASPSMS.COM die Möglichkeit an, nummerische Absender zu erfassen. Diese müssen jedoch zuvor mittels eines Freischaltcodes bestätigt und somit für den Versand freigeschaltet werden.

Für Szenario dieser Thesis besonders praktisch: Nebst den nummerischen, sind auch alphanummerische Absender verwendbar. Es können also beliebige Namen mit einer maximalen Länge von 11 Zeichen, anstelle einer Telefonnummer, als Absender gewählt werden.

Durch die Wahl des Absenders darf jedoch kein Missbrauch erfolgen, dieser würde in jedem Fall zivil- und/oder strafrechtlich verfolgt. Die in diesem Fall verwendete Fantasiebank existiert in der Realität nicht und kann somit problemlos verwendet werden.

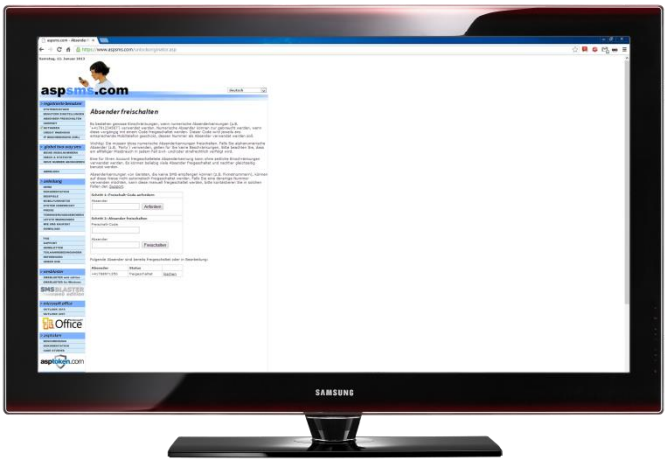

ABBILDUNG 74: FREIGESCHALTETE ABSENDER

Funktionen & Programmablauf

Netzwerkzugriff

Jeder Kunde kann über eine Benutzerauthentifizierung (Benutzer-Schlüssel sowie Kennwort) auf die Dienstleistungen zugreifen. Der Zugriff auf den Gateway beziehungsweise dessen Dienstleistungen, erfolgt redundant über mehrere sogenannte Snatchserver. Diese überprüfen einerseits mit Hilfe der Datenbankservern das Guthaben des jeweiligen Kunden und stellen andererseits die überprüften Daten an den Sendservern zur Verfügung. Die Sendserver wiederum, leiten die verarbeiteten Daten über die SMS Center zu den eigentlichen Empfängern weiter.

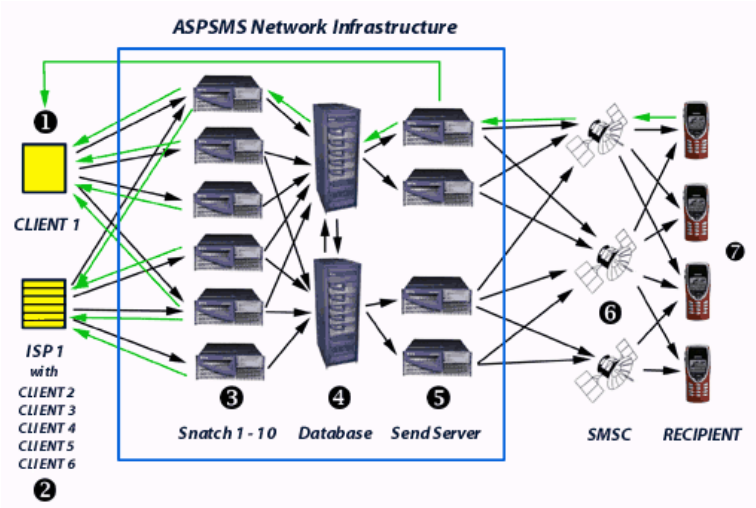

ABBILDUNG 75: NETZZUGRIFF (ASPSMS.COM)

SMS-Token

Der SMS-Gateway bietet, nebst anderen Dienstleistungen, die vor allem im Banksektor verwendeten SMS-Token an. Ein SMS-Token kann zur Authentisierung eines Benutzers, aber auch zur Verifizierung einer Transaktion (mTAN) verwendet werden.

Um ein Token anzufordern, löst der jeweilige Benutzer über seinen User Agent, in dem Fall über den Web Buddy, eine Anfrage an die E-Banking Web-Plattform aus. Diese wiederum kommuniziert über SOAP over HTTPS mit dem SMS-Gateway. Zur Authentisierung wird der Userkey sowie das Passwort des Benutzers verwendet. Als Absender kann die Web-Plattform einen beliebigen Namen verwenden.

Es wird eine Referenznummer generiert und gemeinsam mit Empfängernummer(n) und dem generierten Token auf dem Gateway für eine gewisse Zeit (standardmässig 10 Minuten) abgelegt. Das Token wird per SMS an den bzw. die Empfänger versandt. Der Empfänger bestätigt den Erhalt, indem er das Token in ein entsprechendes Formular eingibt, welches wiederum an den Gateway zur Verifikation abgeschickt wird. Dieser meldet als Antwort den Überprüfungsstatus.

Command & Control-Plattform

Überblick

Die extra für diese Arbeit entwickelte Command & Control-Plattform dient hauptsächlich dazu, einen Überblick über die zu steuernden Smart-Devices zu erhalten.

Im Kontext des Angriffs spielt die Plattform eine eher passive Rolle. Der Angriff beziehungsweise die Modifikation und die Injektion von Transaktionen erfolgt vollautomatisch und autonom auf jedem Smart-Device, sofern die beiden Applikationen SMS Buddy und Web Buddy installiert und aktiv sind.

Über die Plattform lassen sich jedoch alle Mobiltelefonnummern, Vertragsnummern sowie die dazugehörigen persönlichen E-Banking-Kennwörter der befallenen Smart-Devices auslesen. Dies, sofern der Benutzer des Web Buddy die E-Banking Web-Plattform besucht und beim ersten Login seine Nummer angegeben hat. Um die Daten zu erhalten, werden also keine Rechte ausgenutzt, sondern reines Social Engineering durch Manipulation angewendet. Zudem wird das Installationsdatum geloggt.

Jede über den Web Buddy modifizierte und injizierte Transaktion wird mit dem Namen und der Kontonummer des Empfängers, sowie dem überwiesenen Betrag und einer generierten, eindeutigen Referenz an die C&C-Plattform übertragen, sodass pro Smart-Device eine Transaktionsübersicht sowie ein Transaktionstotal ersichtlich ist. Damit könnte man allenfalls ein Limit setzen und die Fraud-Aktivitäten, also die Modifikation und Injektion von Transaktionen, auf dem Smart-Device über die Plattform deaktivieren.

Um die Smart-Devices, oder genauer, den Web Buddy zu steuern, wurden einige SMS-Commands implementiert: So können unter anderem der Fraud-Modus sowie der Injektions-Modus verändert beziehungsweise aktiviert und deaktiviert oder die Datenbank des Web Buddys gelöscht werden.

Weiter können beliebige Custom-Commands gesendet werden. Dazu gehören zum Beispiel das Hinzufügen von neuen Zahlungsbegünstigten, neuen Kontonummern und Beträgen, aber auch das Aktivieren beziehungsweise Deaktivieren von Begünstigten. Natürlich könnte der Funktionsumfang beliebig erweitert werden.

User-Interface & Funktionen

Die Plattform verfügt über mehrere User-Interfaces:

- Login
- Hauptmenü
- Command & Control
- Custom SMS-Commands
- Installation- & Usage-log

Login

Beim ersten Besuch der Plattform, erscheint eine Passwortabfrage. Sobald hier das gültige Passwort eingegeben wurde, wird der Zugriff gewährt und eine Session erstellt.

ABBILDUNG 76: LOGIN

Hauptmenü

Beim eigentlichen Hauptmenu können nun durch Anwählen der Titel, die jeweiligen Inhalte und Funktionen aufgerufen werden. So lässt sich unter anderem das Command & Control Center sowie das Installations- & Verwendungslog öffnen.

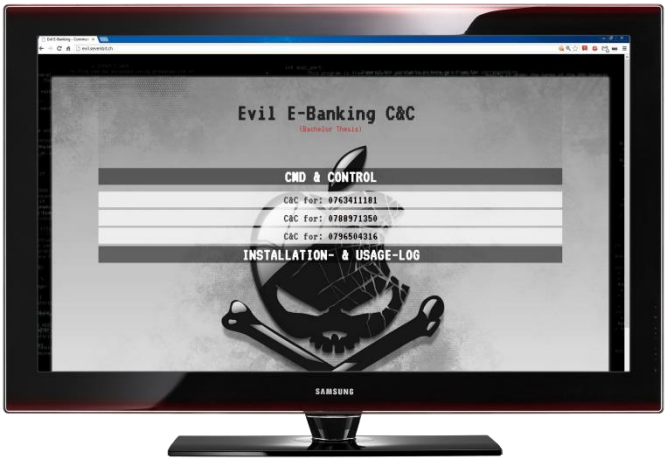

Abbildung 77: Hauptmenü

Command & Control Center

Über das geöffnete Command & Control Center können zahlreiche Befehle per SMS an das jeweilige Smart-Device geschickt werden: Fraud-Mode sowie Injection-Mode ein- bzw. ausschalten oder auch die Datenbank des Web Buddys löschen. Zudem ist der Zeitpunkt der letzten Aktivität ersichtlich und es können Custom SMS-Commands ausgelöst werden.

Abbildung 78: Command & Control

Custom SMS-Commands

Nebst den vordefinierten Commands, können weitere Befehle und Daten übertragen werden. Unter anderem ist es möglich weitere Zahlungsbegünstigte per SMS in die Datenbank des Web Buddys hinzuzufügen oder aber auch bereits eingetragene Begünstigte zu deaktivieren beziehungsweise zu reaktivieren.

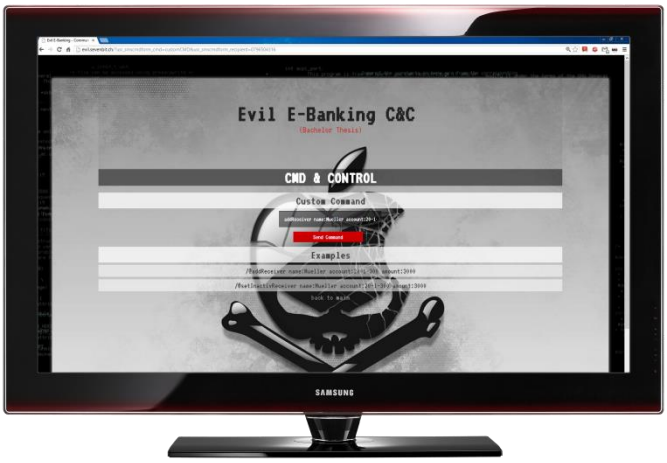

ABBILDUNG 79: CUSTOM SMS-COMMANDS

Installation- & Usage-Log

Dank dem Installations- und Verwendungs-Log, erhält man einen schnellen Überblick über alle infizierten Geräte sowie deren Transaktionsverlauf. Es erscheinen alle manipulierten sowie injizierten Transaktionen des jeweiligen Geräts.

ABBILDUNG 80: INSTALLATION- & USAGE-LOG

Schnittstellen & Erweiterungen

Überblick

Allgemeines

Die SMS-Applikation SMS Buddy und der Webbrowser Web Buddy verfügen bereits über einige implementierte Schnittstellen.

Zur Verschleierung aber auch zum Ausbau der Funktionalität beziehungsweise zum Einsatz in einem alternativen Szenario, wurden einige Erweiterungen und Schnittstellen angedacht.

Implementierte Schnittstellen

Allgemeines

Die beiden Applikationen SMS Buddy und Web Buddy tauschen ihre Daten intern auf dem Smart-Device per Intent aus. Der Vorteil: Damit werden, nebst einer Internetverbindung für den Web Buddy und dem Zugriff auf das GSM-Netz für den SMS Buddy keine weiteren Netzzugriffe benötigt.

Der SMS Buddy kann, nebst den Intents an den Webbrowser, vollkommen eigenständig agieren und SMS über das GSM-Netz empfangen. Der Web Buddy stellt die interne Kommunikation zur SMS-Applikation ebenfalls über Intents her.

Intent

Eine Applikation kann auf eingehende Intents reagieren, sofern ein entsprechender Listener für die Applikation in der zugehörigen Manifest-Datei registriert wurde.

In den von uns entwickelten Applikationen werden die benötigten Listener nicht alle in der jeweiligen Manifest-Datei aufgeführt. Gerade Web Buddy verwendet die Möglichkeit, Listener on-the-fly zu „registrieren“. Somit ist auf Anhieb nicht erkennbar, welche Applikation, auf welche Intents reagiert. Auf diese Art wird eine erste Analyse durch Inspizieren der Manifest-Datei etwas erschwert.

Modulare Erweiterungsmöglichkeiten

Allgemeines

Ein grosses Manko besteht: Die Namen der Intents, sowie die eigentlichen Daten, werden als Plaintext übertragen – dies könnte bei genauerer Analyse auffallen. Auch die SMS-Commands werden im Klartext übermittelt, jedoch nicht auf dem Smart-Device angezeigt, da sie gleich nach dem Erhalt und der anschliessenden Auswertung gelöscht werden.

Auch hier wäre es jedoch für einen erfahrenen Analysten möglich, dies zu erkennen. Eine mögliche Lösung: Security through obscurity. Zudem, wenn man das Szenario wieder auf zwei Geräte, also beispielsweise die klassische Desktop und Smart-Device mit mTAN Situation ausweiten würde, wäre eine externe Kommunikation, zum Beispiel über das Internet, anstelle der internen Intents nötig. Ein infiziertes Smart-Device (mit SMS Buddy) und ein kompromitierter Desktop-Browser (zum Beispiel durch ein manipuliertes Browser-Plugin), könnten über diesen zusätzlichen Kanal kommunizieren und so auch die Sicherheit des klassischen E-Banking stark herabsetzen.

Daher wären unter anderen folgenden Erweiterungsmöglichkeiten möglich und sinnvoll:

- Obfuscation
- IRC-Channel
- Jabber-Channel
- PHP-MySQL-Gateway
- SMS-Channel

Des Weiteren besteht die Möglichkeit, dass die Kommunikation zwischen den Applikationen auch ohne Internet- und GSM-Kanal zwischen den Geräten erfolgen kann. Beispielsweise über:

- Ultraschall
- Bluetooth, IrDA & WiFi direct

Obfuscation

Um die Kommunikation, so wie sie in den aktuellen Prototypen implementiert wurde, zu verschleiern, wären folgende Lösungen bei den verwendeten Intents und SMS-Commands denkbar:

- Beim Start zufällig generierte oder verschleierte Intent-Namen
- Verschleierung des Intent-Contents
- Verschleierung der SMS-Commands

Zur Verschleierung könnten folgende kryptographische Methoden eingesetzt werden:

- Hash-Funktionen: SHA1/256/384/512, MD5, etc.
- Einfache Cipher: ROT13, Ceasar, Shift, Substitution, etc.
- Symmetrische Block Cipher: DES, 3DES, AES, etc.
- Asymmetrische Cipher: RSA, DSA

IRC-Channel

Um allenfalls die interne Kommunikation durch eine externe zu ersetzen oder um auch das klassische Desktop- und Smart-Device-Szenario abdecken und angreifen zu können, wäre die Kommunikation über einen externen IRC-Server mit einem eigenen IRC-Channel pro infiziertes Gerät (wie im Szenario dieser Arbeit) beziehungsweise pro infizierten E-Banking-Kunden (Desktop / Smart-Device) sinnvoll.

Für das zweite Szenario könnte mittels geschicktem Social Engineering sowohl der Desktop, als auch das Smart-Device mit einer Applikation und/oder einem Browser-Plugin versehen werden. Dadurch könnte die Kontrolle über beide Geräte erlangt werden. Der zweite, vermeintlich sichere Kanal wäre somit auch in dieser Konstellation keiner mehr. Besonders leicht wäre ein Angriff bei einem aktiven Google-Account und einem auf Android basierenden Smart-Device: die Installation der Applikation könnte vom infizierten Desktop aus gestartet werden.

Jabber-Channel

Wie beim obengenannten IRC-Channel, wäre auch eine Kommunikation mittels des Jabber-Protokolls möglich, so dass der zweite Kanal mit der generierten mTAN – auch beim klassischen E-Banking – hinfällig und somit unsicher werden würde.

PHP-MySQL-Gateway

Alternativ zu den obigen Kanälen, wäre auch ein PHP-MySQL-Gateway denkbar. Über diesen könnte sowohl das Smart-Device, als auch ein anderes, zweites Gerät wie ein Desktop-PC miteinander kommunizieren. Dies hätte den weiteren Vorteil, dass die Kommunikation kaum auffällt, da alles über Port 80, also der Default-Port für Webanfragen, laufen würde. Anmerkung: Auch IRC und Jabber könnten dank Proxylösungen über diesen Port kommunizieren.

SMS-Channel

Weiter könnte eine reine Kommunikation über SMS angedacht werden. In diesem Szenario müsste lediglich der Webbrowser beziehungsweise ein Browser-Plugin Zugriff auf einen SMS-Kanal und -Rückkanal haben. Der Austausch der Daten könnte auch hier über eine Verschlüsslungstechnik verschleiert werden.

Ultraschall, Bluetooth, IrDA & WiFi direct

Um ganz auf einen Internet- oder GSM-Zugriff zu verzichten, könnten die klassischen Techniken wie Bluetooth, IrDA und WiFi direct zur Kommunikation zwischen den einzelnen Geräten eingesetzt werden.

Eine besondere Methode stellt die Kommunikation über Ultraschall dar: Damit lassen sich Daten über hochfrequentierte Töne übertagen. Unter anderem von einem Smart-Device zum anderen: Somit würde sich beispielsweise ein E-Banking-Botnetz realisieren lassen, in welchem die einzelnen Applikationen physisch voneinander getrennt und unabhängig sind. Aber auch ein Angriff im klassischen Szenario, mit einem Desktop-PC und einem Smart-Device wäre möglich.

Lösungsansätze

(Un)sicheres E-Banking

Überblick

Es stellt sich nun die Frage: Gibt es überhaupt sicheres mobile E-Banking? Die Frage lässt sich nicht direkt mit einem Ja oder Nein beantworten – es muss ausgeholt werden:

Die Analyse und die entwickelten Prototypen haben gezeigt, dass ein Angriff mit relativ einfachen Mitteln und Ressourcen möglich ist.

Der Angriff funktioniert nicht nur beim hier beschriebenen Verfahren, bei welchem die beiden Kanäle – Auslöser einer Transaktion sowie der Empfang der mTAN – auf dem Smart-Device zusammenfallen. Auch beim klassischen E-Banking, bei welchem die Transaktion beispielsweise über einen Desktop-PC ausgelöst und die zu überprüfende mTAN per GSM an ein Smart-Device gesendet wird, ist der in dieser Arbeit beschriebene Angriff möglich (siehe auch „modulare Erweiterungsmöglichkeiten“).

Problematik

Worin besteht das eigentliche Problem, nebst den zusammengefallenen Kanälen?

Zum einen hat es mit dem abnehmenden Sicherheitsbewusstsein der Anwender zu tun. Einige Applikationen, die man auf seinem Mobilgerät installieren kann, fordern mehr Berechtigungen ein, als für die Benutzung notwendig sind und dadurch steigt die Gefahr für Datenverlust, erhöhte Telefonkosten und Schäden am Mobilgerät oder Gefährdung der Privatsphäre (gdata).

Sowohl auf dem Desktop, als auch auf einem Smart-Device kann dies einen grossen Einfluss auf die Sicherheit haben. Viele Benutzer sind sich dieser Gefahr nicht oder zu wenig bewusst.

Andererseits ist auch das Sicherheitskonzept auf nahezu allen Endgeräten fragwürdig: Zwar werden auf den meisten Smart-Devices die für die Applikation nötigen Rechte dem Benutzer angezeigt – dieser kann jedoch oftmals kaum abschätzen, ob diese nun wirklich auch alle benötigt werden. Zudem können, wenn beispielsweise die Leserechte für die SD-Card gesetzt wurden, alle Daten und nicht nur applikationsweite Daten ausgelesen und verändert werden. Oft benötigen die Applikationen zu viele oder unnötige Rechte, ohne, dass man den Grund dafür eruieren könnte.

Weiter fühlt man sich oftmals auf der sicheren Seite: Auf dem Desktop mit den Sicherheitslösungen, welche trügerische Sicherheit versprechen, auf den Smart-Device mit der (noch) geringen Anzahl und Verbreitung von Schadsoftware. Ist eine Applikation erst einmal mit den nötigen Rechten installiert, können die Aktivitäten im Hintergrund nur sehr schwer nachvollzogen und überwacht werden. Dies wird auch in unseren Prototypen ausgenutzt: Während im Vordergrund scheinbar alles mit geregelten Dingen zu und her geht, werden im Hintergrund – ohne irgendwelche Rechte auszuhebeln – relevante Daten ausgetauscht und die Ausgabe manipuliert.

Mögliche Lösungen

Überblick

Es existieren zwar bereits einige sichere E-Banking-Lösungen auf dem Markt. Alle Lösungen haben jedoch gewisse Vor- aber auch Nachteile, welche es vor der Wahl zu beachten gibt.

Folgend eine Auswahl der aktuellen Sicherheitslösungen und Methoden, welche grundsätzlich als sicher erachtet werden können:

- USB Secure-Token
- Zahlungsbegünstigter bestätigen
- Cronto
- chipTAN
- Carberp link

USB Secure-Token: ZTIC

Eine solche Lösung wurde von IBM unter dem Namen „Zone Trusted Information Channel“, kurz ZTIC entwickelt.

ABBILDUNG 81: ZONE TRUSTED INFORMATION CHANNEL

Es handelt sich dabei um einen Sicherheits-Stick, der über die USB-Schnittstelle am privaten Rechner direkt angeschlossen werden kann. Der Stick stellt nun eine sichere Verbindung zwischen dem Rechner und dem Bank-Server her. Die gesamte E-Banking Session erfolgt über das USB-Token und dessen sicheren, verschlüsselten Kanal.

Wird nun eine Transaktion ausgelöst, wird sie durch den ZTIC-Stick an den E-Banking Server gesendet. Die Transaktionsbestätigung direkt auf dem integrierten Display des ZTIC-Sticks angezeigt. Der Benutzer kann nun die Transaktion über entsprechende Bedienelemente auf dem ZTIC-Stick bestätigen oder ablehnen.

Somit kann zwar weiterhin ein Angreifer die Transaktion mit Hilfe des Webbrowsers verändern, allerdings wird die Manipulation spätestens auf dem ZTIC-Stick sichtbar. Es ist somit anzunehmen, dass diese Lösung sicher und eine Manipulation der Anzeige nahezu unmöglich ist.

Obwohl aktuelle Smart-Devices über eine USB-Schnittstelle verfügen, kann der ZTIC-Stick (noch) nicht über diese USB-Schnittstelle am Smart-Device betrieben werden (ZTIC).

Zahlungsbegünstigter bestätigen

Bei der Bestätigung des Begünstigten handelt es sich um eine zusätzliche Sicherheitslösung im Online-Zahlungsverkehr, die verhindert, dass von Kriminellen manipulierte Zahlungen ausgeführt werden.

Die Sicherheitslösung verlangt eine Zusatzprüfung, wenn bei einer Zahlung ein Begünstigter erfasst wird, an welchen zuvor noch nie eine Überweisung getätigt wurde. Der Zahlungsbegünstigte muss vor der Übermittlung mittels eines Codes freigeschaltet und somit einmalig bestätigt werden. Dieser Code wird in den meisten Fällen aus der Kontonummer des Begünstigten generiert und ist somit als sicher anzusehen.

Die Bestätigung ist jedoch nicht bei allen neuen Begünstigten notwendig. Von der Zusatzprüfung ausgenommen sind Begünstigte, die von der Bank bereits freigegeben sind so zum Beispiel die meisten Telefongesellschaften, Elektrizitätswerke, Krankenkassen und Versandhäuser etc. (UBS AG).

Voraussetzung, dass diese Lösung als sicher erachtet werden kann: Die Rechnung sollte physisch vorliegen. Wird beispielsweise E-Payment verwendet – die Rechnung wird als Zahlungsaufforderung in einem Postfach auf der E-Banking Plattform angezeigt – so hat der Angreifer wiederrum die Möglichkeit, die Rechnung mit Hilfe des Webbrowsers zu manipulieren oder möglicherweise gar den Erhalt einer Zahlungsaufforderung dem Benutzer vorzutäuschen.

Ein weiteres Problem beim E-Payment: Wurde der Begünstigte einmal freigegeben, wird eine weitere Bestätigung nicht mehr verlangt. Konkret heisst dies: Gelingt es einem Angreifer, mittels Social Engineering den Benutzer zu überzeugen, dass es sich bei der Bestätigung einer Zahlung um eine plausible und routinemässige Zahlungsbestätigung handelt, wird der Benutzer unter Umständen die Zahlung „freigeben“.

CrontoSign

Bei der Lösung der Firma Cronto, handelt es sich um ein sogenanntes photoTAN-Verfahren. Der Benutzer erfasst eine Transaktion auf der E-Banking Plattform via Webbrowser und erhält als Transaktionsbestätigung ein codiertes „Bild“.

ABBILDUNG 82: CRONTOSIGN (CRONTO)

Der Benutzer fotografiert im Anschluss die Grafik mit dem Smart-Device. Voraussetzung dafür ist, dass die CrontoSign-Applikation auf dem Smart-Device vorgängig installiert und mit einem persönlichen Schlüssel freigeschaltet wurde.

Die Software entschlüsselt die Bilddaten und zeigt – zur Kontrolle – den Betrag und das Konto des Empfängers sowie eine sechsstellige Transaktionsnummer an. Mit der Eingabe dieser Ziffernfolge wird die Transaktion bestätigt und abgeschlossen.

Als Sicherheit für das photoTAN-Verfahren, wird die die Signatur und die Verschlüsselung der Bank sowie die unverfälschte Übertragung auf das Smart-Device angegeben (bank, 2009).

Wenn man dieses Verfahren auf mobile E-Banking anwenden möchte, leidet jedoch die Benutzerfreundlichkeit: Die generierte Grafik müsste vom einen Smart-Device, welches die Transaktion auslöste, mit einem weiteren Smart-Device fotografiert werden.

Zudem speziell: Abgesehen von den Banken, die dieses Verfahren einsetzen, ist nichts Weiteres in Bezug auf dieses photoTAN-Verfahren bekannt. Wie der Grundsatz von Kerckhoffs besagt, darf die Sicherheit eines Verschlüsselungsverfahrens nicht auf der Geheimhaltung des Algorithmus beruhen. (Kerckhoffs)

Somit sei dahingestellt, ob ein Angreifer im Webbrowser nicht doch noch die Möglichkeit besitzt, das System zu manipulieren. Attacken auf ähnliche Systeme konnten bereits erfolgreich durchgeführt werden (heise, 2012).

ZTIC NextGen

Das nachfolgende System beschreibt den Einsatz von ZTIC (IBM) in Kombination mit der kabellosen Datenübertragungstechnik Bluetooth. Diese Lösung existiert (noch) nicht auf dem Markt, wäre aber wohl ein Lösungsansatz für sicheres, mobiles E-Banking.

Bluetooth

Bei Bluetooth handelt es sich um eine von Bluetooth Special Interest Group (SIG) entwickelter Industriestandard. Die Entwicklung reicht in die 90er Jahre zurück, wurde seither ständig weiterentwickelt und optimiert.

Die Sendereichweite liegt dabei zwischen ca. 10 (Klasse 3) bis ca. 100 Meter (Klasse 1). Bluetooth wird heutzutage in einer Vielzahl von Bereichen eingesetzt: Das Einsatzspektrum reich von Wirtschaft-, Industrie-, Medizinbereich bis hin zum heimischen Wohnzimmer, um einige Beispiele zu nennen. Diese Funktechnologie erlaubt es zudem, unterschiedliche Geräte miteinander zu verbinden und Daten austauschen zu lassen. (Bluetooth SIC)

Wollen zwei Bluetooth-Einheiten miteinander in Kontakt treten, müssen sie den Bluetooth Pairing-Prozess durchlaufen. Ergebnis dieses Prozesses ist ein Verbindungsschlüssel, der in einer Tabelle für weitere Kontaktaufnahmen gespeichert oder nach der Kommunikation wieder verworfen wird. Speichern die Geräte den Schlüssel, brauchen sie bei erneuter Kontaktaufnahme den Pairing-Vorgang nicht nochmals durchlaufen. Der Verbindungsschlüssel ist Eingangsparameter für alle sicherheitsrelevanten Mechanismen. (Institut für Internet-Sicherheit)

Sicherheit

Wie steht es allerdings mit der Sicherheit bei dieser breit eingesetzten Funktechnologie? Diese Frage lässt sich nicht auf die Schnelle beantworten und würde den Rahmen dieser Arbeit sprengen. Daher soll hier ein mögliches Konzept vorgestellt werden, welches eine mögliche Verwendung von ZTIC in Kombination mit einem Smart-Device erlaubt.

Ablauf im Überblick

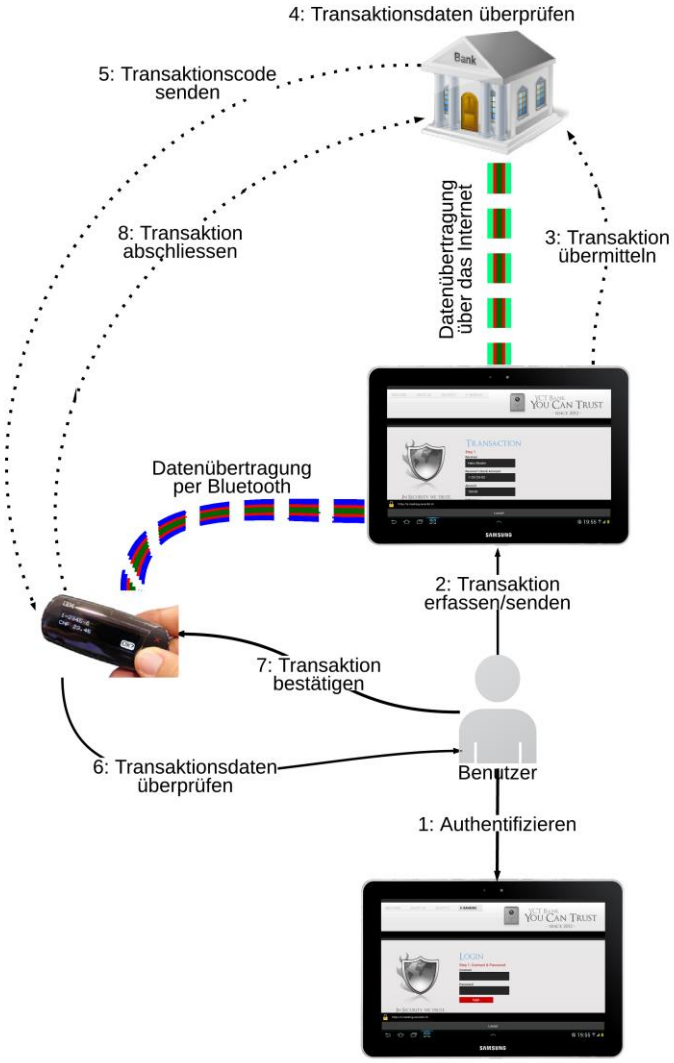

ABBILDUNG 83: PRINZIPIELLER ABLAUF ZTIC NEXTGEN

Die angedachte Lösung ZTIC NextGen basiert auf dem ZTIC, wird jedoch mit der Bluetooth-Technologie erweitert. Dies ermöglicht die Kommunikation mit Smart-Devices, ohne dass eine weitere Hardware-Schnittstelle am Smart-Device benötigt wird.

Als nächstes soll hier der Ablauf eines möglichen E-Banking Szenarios unter Anwendung des ZTIC NextGen aufgezeigt werden.

Ablauf im Detail

1. Zum Starten einer E-Banking Session stellt der Benutzer eine Bluetooth-Verbindung zwischen dem Smart-Device und dem ZTIC NextGen her (Pairing).
   ZTIC NextGen stellt über das Smart-Device eine sichere Verbindung zum E-Banking Server her. Diese umfasst die Kommunikation zwischen:
   a. Smart-Device und E-Banking Server
   b. ZTIC NextGen und Smart-Device
   ZTIC NextGen agiert dabei als Tunnel zur E-Banking Plattform.
   Durch Eingabe seiner Credentials, startet der Benutzer eine E-Banking Session.
2. Zum Erfassen einer Transaktion, ruft der Benutzer die entsprechende Transaktions-Eingabemaske auf. Darin werden Empfänger, Empfängerkontonummer, der zu überweisende Betrag und eine Referenznummer eingeben.
3. Die Transaktionsdaten werden an die Bank übermittelt.
4. Nach Empfang der Transaktionsdaten überprüft die Bank die Gültigkeit und die Vertrauenswürdigkeit der Transaktion.
5. Gilt die Transaktion als valid, wird eine Transaktionsbestätigung ausgelöst und an den Benutzer gesandt.
6. Die Transaktionsbestätigung wird nun allerdings nicht im Webbrowse, sondern über das integrierte Display des ZTIC NextGen.
   Der Benutzer kann nun die Angaben auf dem Display, mit denen auf dem Smart-Device vergleichen.
7. Über Tasten am ZTIC NextGen kann der Benutzer die Transaktion nun bestätigen oder allenfalls ablehnen.
8. Entsprechend wird eine Meldung an den E-Banking Server gesendet, welcher die Transaktion definitiv auslöst oder verwirft.

Da der Benutzer zwar weiterhin einen Webbrowser zum Erfassen der Transaktionen benötigt, hat ein Angreifer noch immer die Möglichkeit Transaktionen zu verändern oder einzuschleusen.

Die Bestätigung, sprich die Freigabe der Transaktion erfolgt allerdings auf dem ZTIC NextGen, welches einerseits einen dedizierten Kommunikationskanal mit dem E-Banking Server aufbaut und welches andererseits eine eigenständige Anzeige besitzt, auf welche der Angreifer nicht einflussnehmen kann.

Schlusswort

Zusammenfassung

Allgemeines

Electronic Banking ist keinesfalls ein junges Produkt. Bereits in den 80er Jahren kamen erste, elektronische Zahlungsmöglichkeiten auf. Heute nimmt, das uns bekannte E-Banking, einen immer höheren Stellenwert im alltäglichen, digitalen Leben ein. Kein Wunder: Das digitale Bankgeschäft ist in der Regel benutzerfreundlich, praktisch, nahezu immer sowie von fast überall her verfügbar und gilt grundsätzlich als sehr sicher.

Diese Arbeit zeigt jedoch ein differenzierteres Bild dieser Technik und der dahinter steckenden, vermeintlichen Sicherheit auf. E-Banking an und für sich ist sicher. Die E-Banking Server sind nahezu uneinnehmbar. Wie das Secure Platform Problem aufzeigte, liegt das Problem auf der Seite des Benutzers: Sowohl die Gutgläubigkeit und teils auch Fahrlässigkeit der Anwender, aber auch das Sicherheitskonzept auf den Endgeräten lassen zu wünschen übrig.

Einführung

Kaum ein Markt wächst derzeit so rasant wie derjenige der Smart-Devices: Egal ob Tablets oder Smartphones – die mobilen Endgeräte vollziehen aktuell einen Siegeszug, der seinesgleichen sucht.

Das in dieser Arbeit beschriebene Szenario zeigt einen komplett autonomen Angriff, welcher sowohl vor dem Benutzer, als auch vor der beteiligten Bank völlig im Verborgenen bleibt.

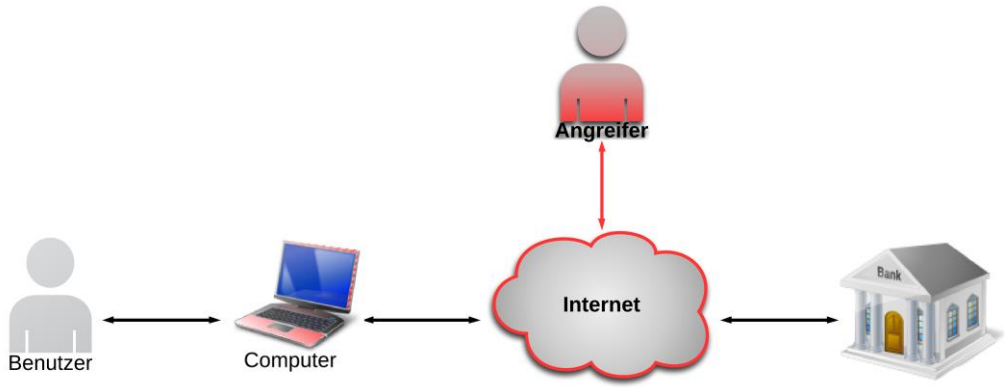

ABBILDUNG 84: KOMPROMITTIERTES SYSTEM

Die entwickelte Erst-Implementation (Proof-of-Concept) zeigt das Angriffspotential in seiner ganzen Tragweite auf. Das Szenario hinterfragt dabei die Sicherheit der eingesetzten Techniken, welche in zahlreichen, aktuellen E-Banking Lösungen zum Einsatz kommen.

Angriff

Das Szenario basiert auf der Tatsache, dass das Erfassen einer Transaktion und die Bestätigung der jeweiligen Transaktion mittels Transaktionscode (mTAN) nicht mehr über je einen dedizierten Kanal erfolgt, da die Kanäle durch den Einsatz eines Smart-Device, zusammenfallen. Konkret lassen sich Transaktionen manipulieren und vor dem E-Banking Benutzer verschleiern. Nebst den, vom Benutzer explizit ausgelösten und bestätigten Transaktionen, können beliebige Überweisungen autonom injiziert und zu Gunsten des Angreifers verarbeitet werden.

Der betroffene Benutzer wird den Angriff erst beim Verwenden eines alternativen, nicht befallenen Geräts oder mit dem nächsten, per Post zugestellten, physikalischen Bankauszug bemerken.

Ablauf im Überblick

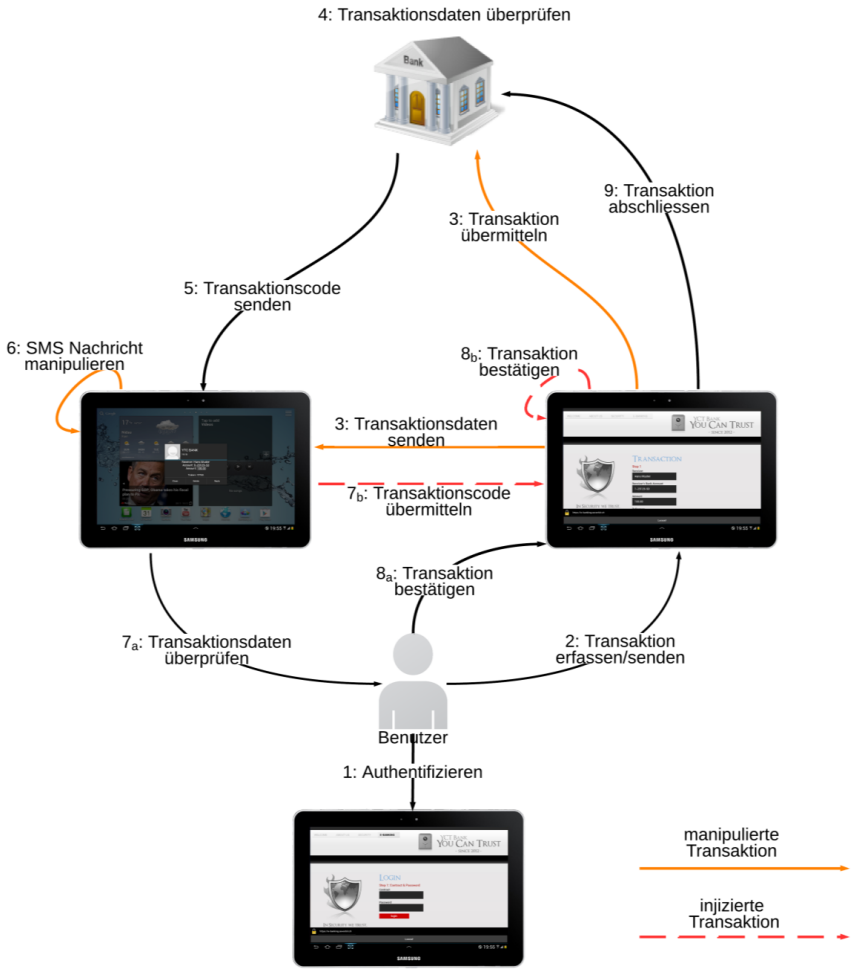

Abbildung 85: Angriff im Überblick

Fazit

Wie im Angriff gezeigt, ist die Sicherheit etablierter E-Banking Lösungen beim Einsatz von Smart-Devices nicht mehr gewährleistet. Die in dieser Arbeit aufgezeigten Angriffe lassen sich grundsätzlich auf alle Formen von E-Banking anwenden, bei denen kein dediziertes, hardwarebasiertes Security-Token zum Einsatz kommt.

Mögliche Lösung: ZTIC NextGen

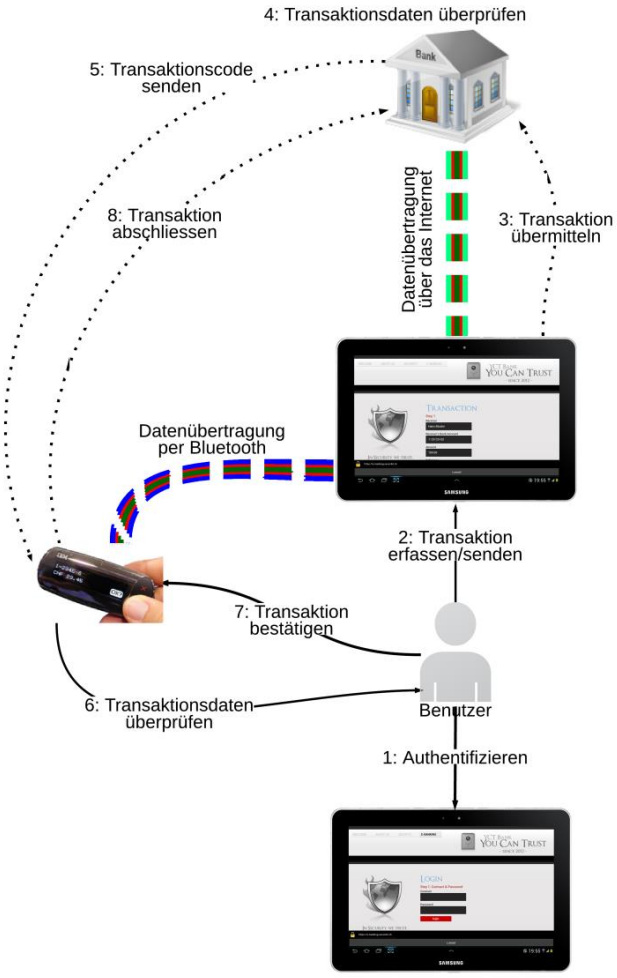

ABBILDUNG 86: PRINZIPIELLER ABLAUF ZTIC NEXTGEN

Die Umsetzung der konkreten Angriffe ist mit der Einführung von Smart-Devices um Grössenordnungen einfacher geworden und birgt somit ein sehr hohes Potential für die nahe Zukunft.

Feststellungen

Der Markt für mobiles E-Banking ist noch keinesfalls gesättigt, im Gegenteil: Die Anzahl Nutzer von Smart-Devices, aber auch die Nutzungsdauer steigt kontinuierlich: Rund 34% der Schweizer Bevölkerung nutzten im ersten Quartal Jahr 2011 ein Smartphone, ein Jahr später waren es bereits 43% der Schweizerinnen und Schweizer. Tendenz steigend: In einer Umfrage gaben 22% an, dass sie das mobile Internet häufiger verwenden werden. Immer mehr Tätigkeiten, welche früher den klassischen Systemen, wie Desktop-PC oder Notebooks, vorenthalten waren, werden nun auf mobilen Endgeräten ausgeführt. So auch das E–Banking. (Personalradar.ch, 2012)

Ein Angriff auf diese Smart-Platforms ist durchaus realistisch: Je mehr und je häufiger finanzkritische Applikationen, wie E-Banking, aber auch direkte Zahlungsmöglichkeiten auf den mobilen Endgeräten zum Einsatz kommen, desto interessanter und lukrativer wird die jeweilige Plattform für Kriminelle.

Zudem fehlt oder mangelt es den Nutzern von Smart-Devices häufig am Sicherheitsbewusstsein. Das Risiko, unbewusst eine Applikation zu installieren, welche im Hintergrund böswillige Aktivitäten durchführt, wird mit der schier endlosen Anzahl verfügbarer Applikationen je länger je grösser.

An dieser Stelle sollen nun die bereits Eingangs aufgeführten Fragen beantwortet werden:

- Wie gross ist der Aufwand für die Umsetzung eines solchen Angriffs?
- Wie viel Wissen wird für die Umsetzung der Komponenten auf Android benötigt?
- Wie viel Zeit wird für die Umsetzung des gesamten Szenarios benötigt?
- Welche Ressourcen und Mittel sind dafür nötig?
- Mit welchen Problemen wird man konfrontiert?

Um es bereits vorweg zu nehmen: Der Aufwand hält beziehungsweise hielt sich stark in Grenzen. Angesichts der geringen Vorkenntnisse in der Programmierung und Entwicklung auf Android, sowie der Entwicklungszeit von rund 2,5 Monaten, darf sich das Proof-of-Concept sicherlich sehen lassen.

Dank dem Szenario, welches mit dem Projekt 2 schrittweise aufgearbeitet wurde, konnte relativ rasch ein theoretischer Angriff, sowie eine erste Implementierung realisiert werden. Die Umsetzung der Teilapplikationen erforderte einiges an Geschick, Wissen, Zeit und Recherche. Gerade weil der Aufbau und die Implementierung der Funktionen zu einem grossen Teil mehr oder weniger autonom erfolgte, konnte die Entwicklung stetig vorangetrieben werden. Neue Erkenntnisse, welche in der Theorie erarbeitet wurden, fanden rasch eine praktische Umsetzung.

Die Zahl der nötigen Ressourcen und Mittel hielt sich in Grenzen. Der Faktor Zeit war dabei wohl das knappste Gut. Die Teilprodukte hätten somit problemlos noch weiter ergänzt und mit zusätzlichen Funktionen und Optimierungen ausgestattet werden können. Während der Entwicklung traten jedoch einige Probleme auf, welche jedoch allesamt mit vertiefter Recherche und zahlreichen Testimplementierungen gelöst werden konnten.

Zukunft & Take-Home-Message

Ein wachsendes Business

Zahlreiche aktuelle Fälle von Malware, welche es sowohl auf die klassischen, wie auch auf die mobilen E-Banking-Lösungen abgesehen haben, zeigen, dass dieses kriminelle Business floriert. Noch in keinem Jahr, gab es so viele Meldungen über Betrugsversuche im E-Banking Bereich, wie im Jahr 2012. (ComputerBase, 2012)

Die Vorarbeiten zu dieser Thesis im Frühjahr letzten Jahres, haben diesen Trend aufgegriffen und den Weg zu diesem Proof-of-Concept etabliert.

Geringer Handlungsbedarf

Viele Banken sehen jedoch die Gefahr der mobilen Lösungen nicht, zu wenig oder versuchen sich mit Klauseln abzusichern.

Die Zahlen der manipulierten Transaktionen sind oft veraltet. Angesichts des rasanten Wachstums des Smart-Device Markts, dürften die Infektionsrate und somit auch der Reputationsschaden, welchen die Banken durch erfolgreiche Attacken erleiden, immer grösseren Ausmasses sein.

Alleine durch den Trojaner Zeus in the Mobile (ZitMO), auch bekannt als Eurograbber, wurden anscheinend über 30‘000 Bankkunden um ihr Geld betrogen: Total bis zu 36 Millionen Euro wurden in relativ kurzer Zeit gestohlen. Ein infizierter Desktop-PC sowie ein befallenes Android Smart-Device oder ein Blackberry nutzten einen ähnlichen Ansatz, wie in dieser Thesis vorgestellt.

Rechenschaft

Dass etwas mit seinem Smart-Device nicht stimmen kann, wird der Benutzer in der Regel es erst dann feststellen, wenn die nächste Rechnung des Telekomanbieters auf dem Tisch liegt. Das Abstreiten der angefallenen Kosten, dürfte sich unter Umständen schwierig erweisen. Der Besitzer des Smart-Devices könnte wohl in solchen Fällen zur Rechenschaft gezogen werden. Man darf dabei aber sicherlich auch auf die Kulanz des Netzanbieters hoffen.

In Zwischenzeit ist dieses Risiko scheinbar einigen Banken bekannt und verbieten explizit die Nutzung ihrer Web-Plattform auf den Smart-Devices, da dadurch der alternative Kanal keiner mehr ist. Dies steht oftmals im Kleingedruckten. Akzeptiert der Benutzer diese AGBs und missachtet die Nutzungsbedingungen, haftet er für den entstandenen Schaden. (Netzwelt.de, 2012)

Doch auch die sicherste Methode, wie ein beispielsweise ein hardwarebasiertes Security-Token, ist nicht mehr sicher, wenn die eigentliche Rechnung gefälscht wurde. Dies wäre zum Beispiel durch Abfangen der Briefpost oder bei einer vorgetäuschten oder manipulierten E-Rechnung möglich, bedarf zwar eines gezielten Angriffs auf ein bekanntes Opfer, hat jedoch mit ziemlicher Sicherheit den grössten Erfolg zu verbuchen.

Anhang

Literaturverzeichnis

1. Bundesamt für Statistik - Internetnutzung. [Online] http://www.bfs.admin.ch/bfs/portal/de/index/themen/16/04/key/approche_globale.indicator.30106.301.html?open=302#302.

2. **Schläper, Michael.** *Institue of Information Security, ETH Zurich.* [Online] 10 2012. http://www.infsec.ethz.ch.

3. Wikipedia - Authentifizierung. [Online] http://de.wikipedia.org/wiki/Authentifizierung.

4. 123RF. [Online] http://de.123rf.com/photo_10025457_online-banking-konzept-karte-holding-hande.html.

5. **UBS AG.** UBS Access Key. [Online] http://www.ubs.com/ch/de/online_services/access_key.html.

6. **aspsms.com.** Network Infrastructure. [Online]

7. gdata. [Online] http://www.gdata.ch/securitylabs/mobile/sicherheitsbewusstsein.html.

8. **ZTIC.** Zone Trusted Information Channel. [Online] http://www.zurich.ibm.com/ztic/.

9. **UBS AG.** Sicherheit. *Bestätigung des Begünstigten.* [Online] http://www.ubs.com/ch/de/online_services/faq/security.html.

10. **Cronto.** [Online] http://www.cronto.com.

11. **bank, die.** Zeitschrift für Bankpolitik und Praxis. [Online] 06 2009. http://www.die-bank.de/banking/testbetrieb-fur-photo-tan.

12. **Kerckhoffs, Auguste.** Wikipeida. [Online] http://de.wikipedia.org/wiki/Kerckhoffs'_Prinzip.

13. **heise.** Online-Banking-Trojaner hat es auf chipTAN-Nutzer abgesehen. *heise Security.* [Online] 09 2012. http://www.heise.de/security/meldung/Online-Banking-Trojaner-hat-es-auf-chipTAN-Nutzer-abgesehen-1701184.html.

14. **Bluetooth SIC.** [Online] http://www.bluetooth.com, http://www.bluetooth.org.

15. **Institut für Internet-Sicherheit.** Bluetooh Pairing. [Online] http://www.internet-sicherheit.de/service/glossar/eintrag/eintrag-detail/bluetooth-pairing/.

16. Personalradar.ch. [Online] Mai 2012. http://www.personalradar.ch/wp-content/uploads/2012/09/Unser-mobiler-Planet-Schweiz-der-mobile-Nutzer-Mai-2012.pdf.

17. ComputerBase. [Online] Januar 2012. http://www.computerbase.de/news/2012-01/steigende-anzahl-von-malware-auf-smartphones/.

18. Netzwelt.de. [Online] 18. 01 2012. http://www.netzwelt.de/news/90356-hohes-risiko-verbraucherschuetzer-warnen-online-banking-smartphone.html.

19. **Lucidchart.** [Online] www.lucidchart.com.

Abbildungsverzeichnis

Bemerkung

Zahlreiche Abbildungen wurden mit der Online-Applikation Lucidchart (Lucidchart) erstellt. Darin enthaltene Piktogramme könnten gegen das Copyright des jeweiligen Publishers verstossen. Wir distanzieren uns davon, da die Piktogramme über eine, in Lucidchart integrierte Suchfunktion bereitgestellt wurden.

C

Cache
: Ein Puffer-Speicher innerhalb der Applikation · 78

H

Handler
: Eine Softwarekomponente, welche spezifische Aufgaben übernimmt/verrichtet · 75

I

Intent
: Intents stellen einen Nachrichtendienst dar, mit dem es möglich ist, Activities und Services zu starten und Broadcast-Receiver über Ereignisse zu informieren · 72

Interfaces
: Schnittstelle · 67

M

mTAN
: Mobile Transaction Number · 66

P

Phishing-Attacke
: Unter Phishing werden Versuche verstanden, über gefälschte WWW-Adressen, E-Mail oder Kurznachrichten an Daten eines Internet-Benutzers zu gelangen und damit Identitätsdiebstahl zu begehen um mit den erhaltenen Daten beispielsweise Kontoplünderung zu begehen und den entsprechenden Personen zu schaden · 17

S

Smart-Device
: Ein elektronisches, kabelloses und mobiles Gerät mit eingebettetem Prozessor, Speicher und Netzwerkverbindung, welches meist über das Display bedient wird. Nebst üblichen Telefonfunktionen weisen solche Geräte eine gewisse Intelligenz auf, um selbstständig Tätigkeiten zu verrichten · 8