Haupert, Vincent
Other Persons: Müller, Tilo; Freiling, Felix; Herrmann, Dominik

Thesis

Sicherheit mobiler Bankgeschäfte zwischen Innovation und Regulierung

Sicherheit mobiler Bankgeschäfte zwischen Innovation und Regulierung

Der Technischen Fakultät der
Friedrich-Alexander-Universität Erlangen-Nürnberg
zur Erlangung des Doktorgrades

Doktor der Ingenieurwissenschaften

vorgelegt von

Vincent Haupert

Als Dissertation genehmigt von der Technischen Fakultät
der Friedrich-Alexander-Universität Erlangen-Nürnberg.

Tag der mündlichen Prüfung:	22. Juli 2019
Vorsitzender des Promotionsorgans:	Prof. Dr.-Ing. Reinhard Lerch
Gutachter:	Prof. Dr.-Ing. Felix Freiling
	Prof. Dr. rer. nat. Dominik Herrmann

Dissertation

Schriftsatz:	LuaLaTeX, KOMA-Script (scrbook), Vincent Haupert, Nürnberg
Druck:	Digitaldruck, Online-Druck GmbH & Co. KG, Krumbach
Bindung:	Fadenheftung, ebd.
Schriftart:	Libertinus (Serif & Sans), Inconsolata (Mono); 10pt
Block:	DIN A5, 100% Altpapier, 100 g/m^2, Blauer Engel
Auflage:	15 Exemplare

Diese Erstauflage ist über die Universitätsbibliothek Erlangen-Nürnberg abrufbar: https://nbn-resolving.de/urn:nbn:de:bvb:29-opus4-113211.

Wissen ist frei

Zur Unterstützung eines zügigen Leseflusses wird in dieser Dissertation verallgemeinernd das generische Maskulinum verwendet. Sämtliche Geschlechtsidentitäten sind bei diesen Formulierungen selbstverständlich gleichermaßen angesprochen und mitgemeint.

Kurzfassung

Seit seiner Einführung besticht das deutsche Onlinebanking mit der Sicherheit einer Zwei-Faktor-Authentifizierung. Obwohl für den Zugang zum Onlinebanking Benutzerkennung und Passwort genügen, müssen Transaktionen durch einen zusätzlichen Faktor bestätigt werden. Zu diesem Zweck fordert die Bank traditionell die Eingabe einer Transaktionsnummer, die der Kunde mithilfe seines Sicherungsverfahrens erhält. Das kontinuierliche Festhalten an dieser Trennung von Transaktionsauslösung und -bestätigung war dabei von einer anhaltenden Verbesserung der Sicherheitseigenschaften der Legitimierungsverfahren begleitet.

Dieser Trend droht sich mit der Verfügbarkeit von Smartphones und Tablets jedoch umzukehren und führt im Privatkundengeschäft der Banken zu deutlichen Nutzungs- und Marktverschiebungen. Denn alte wie neue Finanzdienstleister adaptieren eine Strategie, die möglichst alle Prozesse auf innovative Art und Weise durch das Mobilgerät des Kunden abbilden soll. Diese als Mobilebanking bezeichnete Entwicklung begründet nicht nur die Einführung von Banking-Apps und App-basierten Legitimierungsverfahren, sondern auch ein völlig neues Authentifizierungsparadigma, das es im Kontrast zum klassischen Onlinebanking erstmalig ermöglicht, alle Bankgeschäfte von ein und demselben mobilen Endgerät auszulösen und zu bestätigen.

Die Dissertation beschäftigt sich mit den Sicherheitsimplikationen, die sich durch das Aufkommen des Mobilebankings ergeben. Hierbei wird in der Arbeit zunächst festgestellt, dass sich mehr Angriffsmöglichkeiten durch Schadprogramme ergeben, als dies bisher der Fall war. Im Zentrum stehen zwei Angriffe: Erstens erlaubt die Softwareimplementierung dem Angreifer, einen Replikationsangriff durchzuführen, bei dem das App-basierte Legitimierungsverfahren in vollem Umfang auf ein unautorisiertes Gerät kopiert wird. Zweitens wird durch die fehlende Medientrennung zwischen Auslösung und Bestätigung die Echtzeitmanipulation einer nutzerinitiierten Transaktion möglich. Beide Angriffe fußen auf konzeptionellen Defiziten, die darauf zurückzuführen sind, dass die Mobilebanking-Verfahren ohne adäquate Hardwaremöglichkeiten zur Absicherung eingeführt wurden.

Aus diesem Grund versuchen die Banken ihre Apps durch kommerzielle Härtungsprodukte auf Softwareebene zu schützen. Weiterführende Untersuchungen zeigen solchen Lösungen jedoch klare Grenzen auf und machen deutlich, dass auch sie die konzeptionellen Defizite nicht ausgleichen können. Während den etablierten Banken damit zumindest attestiert werden kann, das strukturelle Sicherheitsrisiko zu kennen und adressieren zu wollen, reichen die Probleme bei neuen Marktteilnehmern weiter. Forschungen im Rahmen dieser Dissertation haben bei dem derzeit führenden deutschen Finanz-Start-up gravierende Sicherheitsmängel identifiziert, die ihre Ursache in einer mangelnden Priorität der IT-Sicherheit finden.

Die ermittelten Defizite sind auch in Bezug auf regulatorische Vorgaben zur Sicherheit mobiler Finanzlösungen relevant. Im Rahmen der Zahlungsdiensterichtlinie II hat die Europäische Union Vorgaben auf den Weg gebracht, die ab dem 14. September 2019 in Geltung treten. Die Dissertation beschäftigt sich in diesem Zusammenhang damit, welche Anforderungen an die Sicherheit digitaler Transaktionen allgemein zu stellen sind und konstatiert im Vergleich mit den rechtlichen Vorgaben weitgehende Kompatibilität. Weiterer Untersuchungsgegenstand ist die Richtlinienkonformität gängiger Sicherungsverfahren im Online- und Mobilebanking. Neben der Nichtkonformität listenbasierter Verfahren legt die Arbeit auch eine Unzulänglichkeit von Verfahren auf Basis des SMS-Telekommunikationsdienstes sowie unter der Verwendung von App-basierten Methoden nahe.

Obwohl die Richtlinie für eine Erhöhung des Sicherheitsniveaus bei Bankgeschäften sorgt, identifiziert die Abhandlung weitere Schwachstellen im Transaktionsprozess, die von der Regulierung nicht erfasst werden und ihre Ursache auch in menschlichen Faktoren finden. Die Arbeit setzt sich deshalb in einer Nutzerstudie mit der praktischen Sicherheit von Transaktionen beim Onlinebanking auseinander. Die Studie kommt zum Schluss, dass die Teilnehmer sich oft nicht im Klaren sind, welche Schritte für die Sicherheit essentiell sind, weshalb sie Transaktionsdaten gar nicht oder nur fehlerhaft prüfen. Die Banken sind dabei Teil des Problems, da sie dem Kunden mitunter irreführende Informationen zur Verfügung stellen.

Die Ergebnisse dieser Arbeit haben aufgrund ihrer angewandten Natur neben der Forschung auch Relevanz für die Öffentlichkeit sowie die Aufsichtsbehörden. Viele Beiträge der Dissertation wurden durch die Presse rezipiert, wodurch ein Sicherheitsbewusstsein für Bankgeschäfte in der Gesamtbevölkerung gefördert wird. Darüber hinaus stellt auch das Bundesamt für Sicherheit in der Informationstechnik die Bedeutung der Arbeit heraus, indem Teile davon Erwähnung im Bericht zur Lage der IT-Sicherheit in Deutschland für das Jahr 2018 fanden.

Abstract

Since its introduction, German online banking has been using a two-factor authentication scheme. Although a combination of a username and a password is sufficient for logging into personal online banking, transactions require a confirmation from an additional factor. To that end, in a second step, banks ask for a transaction authentication number, which the customer creates through an individual transaction confirmation method. The continuous separation between transaction issuing and confirmation was attended by a consistent further development of the security properties of the confirmation methods.

This trend seems to decline as smartphones and tablets become increasingly available and cause a significant shift in the usage patterns and the market. Consequently, both old and new financial institutions adopt a strategy that aims at implementing all processes in an innovative manner through the customer's mobile device. This development, called mobile banking, does not only mark the creation of banking apps and app-based confirmation methods but also establishes a new authentication paradigm. In contrast to the classic online banking, mobile banking allows for using the same device for transaction issuing and confirmation.

This dissertation deals with the security implications of the introduction of mobile banking. At first, the work identifies a broader attack surface for malware than has been the case so far. We highlight two attacks: First, the software implementation allows an adversary to perform a replication attack that copies the app-based confirmation method entirely to another, unauthorized device. Second, owing to the missing physical separation of issuing and confirmation, a real-time manipulation of a user-initiated transaction becomes possible. Both attacks depend on conceptual deficits that result from the introduction of mobile banking methods without proper hardware protection mechanisms.

As a result, banks attempt to secure their apps through commercial app-hardening products which employ software protection techniques. Additional investigations run rigs around the capabilities of these solutions, emphasizing that they cannot

eliminate the conceptual deficits. While this leads to the conclusion that the established banks acknowledge the structural security risks and are willing to address it, the issues of the new participants in the banking market are more profound. This research identified serious security holes at a leading financial startup due to a poor priority of information security.

Moreover, the conceptual and technical deficits are relevant for the regulatory requirements of the security of mobile financial solutions. As part of the Revised Payment Service Directive, the European Union has established statutory provisions that are slated to take effect from September 14, 2019. In the light of this regulation, the dissertation develops general requirements for the security of digital transactions and states broad compatibility with the regulatory terms. Another research topic is the compliance with the directive regarding popular transaction confirmation methods in online and mobile banking. Besides the non-compliance of list-based methods, the thesis also suggests an insufficiency of methods based on the SMS telecommunication service as well as for app-based procedures.

Although the Revised Payment Service Directive ensures increased security in the banking system, the dissertation identified additional weaknesses in the transaction process which are not covered by the regulation and result from human factors. Therefore, as part of a user study, the work deals with the practical security of online banking transactions. The study concludes that the participants are often not aware which steps are essential for security and therefore perform a faulty transaction verification, or omit the verification step altogether. The banks are part of the problem as they provide customers with misleading information.

Owing to the applied nature of this dissertation, the results are relevant not only for the research community but also for the public and regulatory authorities alike. Many contributions of this thesis were received by the press, resulting in an increased awareness of the security of banking among the general population. Additionally, the Federal Office for Information Security has highlighted the importance of our research by mentioning it in the 2018 report on the state of IT security in Germany.

Vorwort

Als ich mich im Sommer 2015 erstmals näher mit der Sicherheit App-basierter Legitimierungsverfahren im Mobilebanking beschäftigte, hatte ich die Untersuchungen auf dem Gebiet noch für einen kurzweiligen Exkurs gehalten. Immerhin galt meine Forschungsarbeit zu jener Zeit primär der hardwarenahen Sicherheit und hier insbesondere der x86-Architektur – eine Technologie, die weder damals noch heute im Verdacht stand, bei mobilen Endgeräten eine Rolle zu spielen. Am Lehrstuhl für IT-Sicherheitsinfrastrukturen waren Forschungsarbeiten zum Themenkomplex Online- und Mobilebanking ebenfalls ein Novum. Die breite Relevanz für die Gesamtbevölkerung und der öffentliche Zuspruch sorgten letztendlich dennoch dafür, dass es nicht bei einem Intermezzo blieb und sich der Schwerpunkt meiner Forschungsarbeit nachhaltig verlagerte. Obwohl das Themengebiet einen gewissen Exotenstatus in unserer Forschungsgruppe implizierte, war ich niemals allein: bei meinem Promotionsvorhaben dienten mir mehrere Personen als Wegbereiter und Wegbegleiter, förderten, forderten und unterstützten mich.

Mein Dank gebührt zunächst Felix Freiling, der mir die Möglichkeit zur Promotion an seinem Lehrstuhl eröffnet hat. Felix' Unterstützung endete jedoch nie an den Grenzen des Betreuers einer Promotion: er schreckte auch in schwierigen Situationen nicht davor zurück, seinen Mitarbeitern mit Wort und Tat beizustehen – Umstände, deren Eintreten mehrmals unmittelbar auf meine Arbeit und Person zurückzuführen waren. Seine ruhige und sachbezogene Herangehensweise sowie sein argumentativ geschicktes und präzises Auftreten habe ich stets als inspirierend empfunden.

Danke auch an Dominik Herrmann, der sich dazu bereit erklärt hat, das Zweitgutachten meiner Promotionsschrift zu übernehmen. Es ehrt mich, dass mit Felix und Dominik gleich zwei Gutachter bestellt wurden, die in den Jahren 2001 bzw. 2015 mit dem Dissertationspreis der Gesellschaft für Informatik prämiert wurden.

Eine besondere Rolle kommt Tilo Müller zu, dem ich in zweierlei Hinsicht meinen Dank aussprechen möchte. Zum einen stellte mir Tilo bereits im Frühjahr 2015 die Promotion in seiner Forschungsgruppe in Aussicht, nachdem ich bereits meine

Seminar- und Bachelorarbeit bei ihm anfertigen durfte. Dabei bot mir seine fachliche Betreuung stets ein angenehmes und ausgewogenes Maß an Orientierung und Freiheit. Zum anderen durfte ich mit Tilo auch abseits der Forschung viele Erfolge feiern und Herausforderungen meistern, die viel zu meiner beruflichen und persönlichen Entwicklung beigetragen haben.

Ferner gilt mein Dank Markus Golling und Dieter Wagner von der DATEV eG, die mir in unserem gemeinsamen Forschungsprojekt weitreichende Freiheiten eingeräumt haben. Neben der für das Forschungsprojekt vorgesehenen Gelder sorgten sie mit kreativen Lösungen außerdem für eine optimale technische und personelle Ausstattung. Bereits vor meiner Promotion haben sie mich im Rahmen meiner Ausbildung und der nachfolgenden Zeit als Werkstudent immer frei von Eigennützigkeit gefördert.

Ich möchte mich auch bei Teresa Hahn bedanken, die sich ohne Umwege dazu bereit erklärte, meine Arbeit sogar bei ausgesprochen knappen Fristen bis in die Morgenstunden zu redigieren – ein Freundschaftsdienst, der sich fern von Selbstverständlichkeit befindet und den ich sehr zu schätzen weiß.

Hervorheben möchte ich außerdem die Zusammenarbeit mit Dominik Maier, Stephan Gabert, Van Tuan Vo und Nicolas Schneider. Dominik ist ein langjähriger Wegbegleiter und Freund, der an mehreren Projekten unmittelbar beteiligt war. Sein Geschick, Forschungsergebnissen ein interessantes und kohärentes Thema zu geben, dürfte vor den Programmkomitees mehr als ein Mal den Unterschied gemacht haben. Stephan und Tuan unterstützten meine Arbeit als Hilfskräfte und produzierten mit ihrer jeweiligen Masterarbeit wertvolle Ergebnisse, die zum Teil auch in diese Dissertation Einzug genommen haben. Euch allen herzlichen Dank!

Nicht zuletzt möchte ich mich für das kollegiale Umfeld am Lehrstuhl bedanken, das mir nicht nur durch tiefgründige Fachgespräche, sondern auch durch eine Vielzahl schillernder Momente in Erinnerung bleiben wird. Ich freue mich auf den weiteren Austausch mit euch und illustre Abende im Montagsseminar.

München, 16. August 2019

Vincent Haupert

Inhaltsverzeichnis

1 Einleitung **1**
- 1.1 Motivation . . . 1
- 1.2 Forschungsfragen . . . 4
- 1.3 Beiträge . . . 5
- 1.4 Publikationen . . . 7
- 1.5 Inhaltsüberblick . . . 13

2 Vom Online- zum Mobilebanking **17**
- 2.1 Grundlagen . . . 17
- 2.2 Klassische Sicherungsverfahren . . . 20
- 2.3 App-basierte Sicherungsverfahren . . . 28
- 2.4 Diskussion . . . 31
- 2.5 Fazit . . . 34

3 Sicherheit beim Mobilebanking **35**
- 3.1 Angreifermodell . . . 35
- 3.2 Sicherheit mobiler Endgeräte . . . 36
- 3.3 Grundsätzliche Angriffe . . . 41
- 3.4 Diskussion . . . 55
- 3.5 Fazit . . . 57

4 Grenzen der App-Härtung **59**
- 4.1 Marktüberblick . . . 60
- 4.2 Fallstudie P Shield . . . 67
- 4.3 Diskussion . . . 81
- 4.4 Fazit . . . 85

5 Fintech-Sicherheit am Beispiel N26 **87**
- 5.1 Hintergrund und Forschungsstand . . . 88
- 5.2 Sicherheitsdefizite . . . 90
- 5.3 Angriffsszenarien . . . 103
- 5.4 Reaktion . . . 106
- 5.5 Fazit . . . 107

6 Bankgeschäfte unter der Zahlungsdiensterichtlinie II 109
6.1 Allgemeine Voraussetzungen 110
6.2 Regulatorische Voraussetzungen 113
6.3 Konformität etablierter Sicherungsverfahren 120
6.4 Ausblick auf potenzielle Angriffe 129
6.5 Fazit 134

7 Sorgfaltspflicht des Kunden in der Praxis 137
7.1 Forschungsstand 140
7.2 Methodologie 145
7.3 Resultate 151
7.4 Diskussion 157
7.5 Fazit 167

8 Schluss 169
8.1 Zusammenfassung 169
8.2 Ausblick 171

Abkürzungen 175

Literatur 177

1

Einleitung

Die Kunden akzeptieren Dinge, die wir nicht für möglich hielten.

— Alfred Richter, 1979 [Spi79]

Die Verbraucherbank ist heute ebenso in Vergessenheit geraten wie ihr ehemaliger Vorsitzender Alfred Richter. Die auf ihn zurückzuführenden technischen Errungenschaften im Privatkundengeschäft sind uns hingegen erhalten geblieben. Denn Richter gilt nicht nur als Vater des Geldautomats, sondern auch der digitalen Dienstleistung, die heute zum Standardrepertoire einer jeden Bank zählt: Onlinebanking [Wey99]. Damit ermöglicht die Bank ihren Kunden, sich eigenständig über das Internet im Onlineportal der Bank anzumelden, um beispielsweise den Kontostand einzusehen oder neue Aufträge wie Überweisungen zu zeichnen.

1.1 Motivation

Obwohl das damalige Onlinebanking äußerlich nicht mehr viel mit dem heutigen gemein hat, wurde es im Kern bereits am 12. November 1980 von der Verbraucherbank in Bonn vorgestellt und wenig später den eigenen Kunden angeboten [HB98]. Damit war die Verbraucherbank auf Richters Initiative die erste Bank weltweit, die ihren Kunden die Abwicklung von Bankgeschäften von zu Hause aus und sogar außerhalb der Öffnungszeiten anbot [Mar17]. Gerade deshalb bezeichnen sich die deutschen Kreditinstitute in Abgrenzung zu den heutigen Innovationen aus Übersee gerne als Innovations- und Digitalisierungsvorreiter [Kar15; DPA18]. Trotz der jährlich

steigenden Zahl an Onlinebanking-Nutzern ist auch noch Jahrzehnte später beinahe die Hälfte des Potenzials unerschlossen: 58% der Internetnutzer verwendeten 2017 Onlinebanking [DBB18]. Ob es jedoch die Innovationen der alteingesessenen Banken sein werden, die die bestehenden Onlinebanking-Kunden halten und die ausstehenden gewinnen, ist indes alles andere als gewiss. Denn auch im ebenso hochregulierten wie von Konkurrenz geprägten Finanzmarkt entstehen mit den Finanz-Start-ups (Fintechs) neue Unternehmen, die es sich zum Ziel machen, die etablierten Banken zu beerben [Vie14].

Den jungen Wettbewerbern im Privatkundengeschäft, die auch Challenger-Banken genannt werden, ist dabei gemein, dass sie sich als schlanke Direktbanken ohne Filialen mit reinen Onlinekonten ausrichten und das Smartphone des Kunden in den Mittelpunkt stellen. Im Gegensatz zu den etablierten Banken, die bedingt durch die Zinspolitik der Europäischen Zentralbank oftmals eine Kontoführungsgebühr erheben, sind die Konten bei den Fintech-Banken kostenfrei [AD16]. Noch gewichtiger als der Kostenvorteil ist der Fokus auf das Smartphone des Kunden: Alle Geschäftsvorfälle, für die bisher zum Teil noch ein Besuch in der Filiale notwendig wäre, soll der Kunde zu jederzeit allein über sein Smartphone tätigen können.

Die Zahlen sprechen für die selbstbewusste junge Konkurrenz. So hat beispielsweise die Berliner Fintech-Bank N26 in weniger als vier Jahren bereits über 2,3 Millionen Kunden für sich gewonnen und wurde im Januar 2019 mit 2,3 Milliarden Euro bewertet [KS19]. Neben der deutschen N26 schicken sich mit Revolut, Monzo oder der Atom Bank weitere Challenger-Banken aus dem Ausland der Europäischen Union (EU) an, den alteingesessenen Banken die Kunden streitig zu machen. Indes kämpfen die etablierten Geldhäuser mit einem eminenten Bedeutungsverlust: seit Jahren ist nicht nur die Anzahl der Filialen, sondern auch die der Kreditinstitute selbst rückläufig [Hin18]. Sogar in den obersten Riegen des deutschen Finanzmarkts manifestiert sich der Einfluss der Fintechs allmählich. Im Herbst 2018 verdrängte mit Wirecard ein Fintech der ersten Stunde die Commerzbank aus dem deutschen Leitindex DAX [Sch18]. Der Erfolg der neuen Marktteilnehmer setzt die alten zunehmend unter Druck und führt zu einem Strategiewechsel, der das Smartphone ebenfalls ins Zentrum rückt [Atz18]. Der Streit um das beste Benutzererlebnis bei mobilen Bankgeschäften bleibt jedoch nicht ohne Folgen für deren Sicherheit.

Schon bei der Konzeption des Onlinebankings bedachte Alfred Richter, dass solche entfernten Bankgeschäfte besonders zu sichern seien [FAZ98]. Deshalb werden Onlinebanking-Transaktionen seit jeher durch eine Zwei-Faktor-Authentifizierung (2FA) abgesichert. Dabei hat sich unter dem Eindruck gehäufter Schadensfälle die

Prämisse herausgebildet, dass an der Transaktionsauslösung und -bestätigung jeweils zwei unterschiedliche Geräte zu beteiligen sind [ENISA12]. Durch den Ansatz der Fintechs, alle Prozesse vollständig durch das Smartphone abzuwickeln, finden beide Schritte auf ein und demselben Gerät statt. Die Aufhebung der Gerätetrennung ist eine prägende Eigenschaft des Mobilebankings und hat weitreichenden Einfluss auf die Sicherheit mobiler Bankgeschäfte. Aus diesem Grund stellt der Paradigmenwechsel bei der Authentifizierung vom Online- zum Mobilebanking einen zentralen Untersuchungsgegenstand dieser Dissertation dar. Ferner geht die Arbeit der Frage nach, ob das Streben der Fintech-Banken nach schrankenfreiem Nutzererlebnis, mit dem sie die eher konservativ ausgerichteten, etablierten Institute vor sich hertreiben, auf Kosten der IT-Sicherheit geht.

Um den Veränderungen gerecht zu werden, hat sich auch die EU mit den neuen Voraussetzungen beschäftigt. Seit dem 13. Januar 2018 ist die Zahlungsdienstrichtlinie II (PSD2) in Kraft [ABl15a]. Die PSD2 hat auf der einen Seite das Ziel, aktuelle Gesetzeslücken zu schließen und das Vertrauen in den digitalen Zahlungsverkehr durch erhöhte Sicherheitsanforderungen zu fördern. Auf der anderen Seite soll die Richtlinie dazu beitragen, den Wettbewerb und Innovationen zu fördern. Für beide Ziele ist jedoch nicht die Richtlinie selbst maßgebend, sondern die Technischen Regulierungsstandards (RTS), die zum 14. September 2019 gelten [ABl18].

Für Bankgeschäfte im Online- wie Mobilebanking fordern die RTS nicht nur mindestens zwei unterschiedliche Authentifizierungselemente, sondern auch eine dynamische Verknüpfung von Zahlungsempfänger und -betrag an den obligatorischen Authentifizierungscode. Dabei war die Ausarbeitung der RTS nicht von weniger Kontroversen begleitet, als die nun notwendige Auslegung der Vorgaben und die daraus resultierenden Implikationen für die Banken. Die Dissertation beschäftigt sich deshalb mit den Anforderungen der RTS und ihren Auswirkungen auf bestehende Legitimierungsverfahren im Online- und Mobilebanking.

Obwohl die gesetzlichen Vorgaben zu einer Erhöhung der Sicherheit bei digitalen Bankgeschäften beitragen, können sie nicht alle Probleme vollumfänglich adressieren. Die Arbeit geht deshalb der Frage nach, welche Angriffsfläche zurückbleibt, wenn die RTS am 14. September 2019 ihre Wirkung entfalten. Ein besonderes Augenmerk liegt auf einer Nutzerstudie, die die praktische Sicherheit von Onlinebanking-Transaktionen erforscht. Untersuchungsgegenstand sind dabei Überweisungen, die mit Sicherungsverfahren durchgeführt werden, die vermutlich auch noch im Geltungsbereich der RTS Verwendung finden.

1.2 Forschungsfragen

Die Dissertation wird von fünf Forschungsfragen geleitet. Dabei beschäftigen sich die Forschungsfragen 1 bis 3 mit dem Paradigmenwechsel zum Mobilebanking und den konkreten Implementierungen von alten und neuen Marktteilnehmern. Die beiden Forschungsfragen 4 und 5 setzen sich hingegen mit rechtlich–regulatorischen Aspekten auseinander.

Forschungsfrage 1: Einordnung Mobilebanking.

- Wie unterscheidet sich Mobilebanking in Abgrenzung zum Onlinebanking?
- Welche neuen Formen von App-basierten Legitimierungsverfahren gibt es und wie lassen sie sich einsetzen?

Forschungsfrage 2: Angriffsfläche Mobilebanking.

- Welche konzeptionellen Angriffsmöglichkeiten ergeben sich gegen die neue Infrastruktur im Mobilebanking?
- Über welche Schutzmaßnahmen verfügen die App-basierten Lösungen und wie wirksam sind sie?

Forschungsfrage 3: Fintech-Sicherheit.

- Geht die Priorisierung der Fintech-Banken auf ein möglichst bequemes Benutzererlebnis zulasten der Sicherheit?

Forschungsfrage 4: Regulierung.

- Welche Anforderungen sind allgemein an sichere digitale Transaktionen zu stellen?
- Genügen die Vorgaben der RTS diesen allgemeinen Anforderungen?
- Entsprechen die gängigen Legitimierungsverfahren im Online- und Mobilebanking den regulatorischen Vorgaben?
- Welche Schwachstellen sind bei digitalen Bankgeschäften auch nach Inkrafttreten der RTS noch offen?

Forschungsfrage 5: Sorgfaltspflichten.

- Sind Nutzer des Onlinebankings in der Lage, eine manipulierte Überweisung durch korrekte Verifikation der Auftragsdaten in ihrem Sicherungsverfahren gemäß ihrer Sorgfaltspflichten zu erkennen?

1.3 Beiträge

Aufgrund ihrer angewandten und interdisziplinären Natur sind die Beiträge dieser Dissertation nicht nur für die Forschung, sondern auch für die Wirtschaft, die Rechtsprechung, die Bundesbehörden und die allgemeine Bevölkerung von Bedeutung. In Konsequenz wurden unsere Arbeiten nicht nur intensiv durch die Presse rezipiert, sondern erreichten auch die Aufmerksamkeit der Bundesanstalt für Finanzdienstleistungsaufsicht (BaFin) und des Bundesamts für Sicherheit in der Informationstechnik (BSI). Unsere wichtigsten Beiträge sind im Folgenden zusammengefasst.

Charakterisierung des Paradigmenwechsels zum Mobilebanking.
Wir beschäftigen uns als Erstes mit dem Phänomen Mobilebanking und grenzen es vom Onlinebanking ab. Im Onlinebanking forderten die Banken noch zwei physisch getrennte Authentifizierungsinstrumente zur Transaktionsauslösung und -bestätigung. Im Mobilebanking sind nun hingegen Transaktionen auf ein und demselben Gerät möglich. Die App-basierten Sicherungsverfahren lassen sich in drei Klassen einteilen und folgen entweder einer Zwei-Geräte-, Zwei-App, oder Ein-App-Authentifizierung.

Identifizierung konzeptioneller Schwächen im Mobilebanking.
Wir zeigen, dass App-basierte Sicherungsverfahren und die fehlende Gerätetrennung im Mobilebanking neue konzeptionelle Schwächen mit sich bringen. Zu diesem Zweck stellen wir einen Replikations- und einen Transaktionsmanipulationsangriff vor und zeigen deren Praxistauglichkeit anhand produktiver Systeme. Die Angriffe haben ihre Ursache in der oft noch fehlenden Hardwareunterstützung der Geräte und lassen sich deshalb erst mittel- bis langfristig verhindern.

Aufzeigen von Grenzen softwarebasierter App-Härtungsmaßnahmen.
Wir skizzieren den Markt an App-Härtungslösungen, die gerade bei Banken beliebt sind, um die konzeptionellen Defizite durch technische Maßnahmen auf Softwareebene zu adressieren. Wir zeigen, dass solchen Lösungen enge Grenzen gesetzt sind und präsentieren zwei Angriffe gegen die führende App-Härtungslösung für deutsche Banking-Apps und App-basierte Sicherungsverfahren. Unsere Angriffe sind vollautomatisch in der Lage, den Schutz der Härtungslösung entweder statisch zu entfernen oder dynamisch auszuschalten.

Offenlegung schwerer Sicherheitsmängel bei einem führenden Fintech.
Wir leisten die erste vollumfängliche Sicherheitsanalyse eines bedeutenden Fintechs und regen damit den Diskurs um die Sicherheit bei Finanz-Start-ups allgemein an.

Bei unserer Fallstudie handelt es sich nicht nur um das in Deutschland führende Fintech und die treibende Kraft im Mobilebanking, sondern auch um das EU-weit erste Fintech, das mit einer Vollbanklizenz operiert. Unsere Analyse findet bei der Fintech-Bank mehrere, zum Teil schwerwiegende Sicherheitsmängel, die auf Implementierungsfehler im Front- wie im Backend zurückzuführen sind. Alle Defizite haben wir vor dem Öffentlichmachen an die Bank gemeldet.

Bewertung der regulatorischen Anforderungen.
Wir stellen allgemeine Voraussetzungen an die Transaktionssicherheit auf und vergleichen diese mit den Anforderungen, die sich durch die RTS der PSD2 ergeben. Wir konstatieren, dass die regulatorischen Anforderungen an die starke Kundenauthentifizierung weitgehend kompatibel zu unseren allgemeinen Bedingungen sind.

Bewertung der Konformität etablierter Sicherungsverfahren.
Wir beurteilen die Konformität gängiger Sicherungsverfahren im Online- und Mobilebanking zu den regulatorischen Anforderungen der RTS der PSD2. Dabei stellen wir fest, dass listenbasierte Verfahren und solche auf Basis des Mobilfunks nicht konform sind. Mobilebanking-Verfahren können zwar mittel- bis langfristig die Voraussetzungen erfüllen, verfügen auf den allgemein verbreiteten Smartphones aber noch nicht über die notwendigen Hardwaremechanismen und Programmierschnittstellen.

Ausblick auf zukünftige Angriffsvektoren bei digitalen Bankgeschäften.
Wir geben eine Einschätzung dazu, wie sich Angriffe aufgrund der höheren regulatorischen Anforderungen entwickeln könnten. Hierzu skizzieren wir fünf Angriffsszenarien, die bis auf eine Ausnahme nicht die Transaktionsbestätigung, sondern die Transaktionsauslösung betreffen.

Studie zur praktischen Sicherheit von Onlinebanking-Transaktionen.
Wir liefern Indizien, dass der Onlinebanking-Kunde seine Sorgfaltspflichten nur unzureichend versteht oder vernachlässigt. In einer Studie mit 100 Onlinebanking-Kunden mussten die Probanden zwei Transaktionen durchführen. Die erste Transaktion lief regulär ab; bei der zweiten wurde der Zahlungsempfänger ausgetauscht und eine fehlerhafte Transaktionsverifikation provoziert. Die Probanden nutzten dabei dasselbe Sicherungsverfahren, das sie auch bei ihrer Hausbank einsetzten. 81 Teilnehmer erkannten die Manipulation nicht, da sie ihrer Sorgfaltspflicht zur Kontrolle der Auftragsdaten entweder gar nicht oder aufgrund unseres Angriffs nur unzureichend nachkamen.

1.4 Publikationen

In diesem Unterkapitel findet die erste Person Singular Verwendung, um die Eigenleistung des Autors dieser Dissertation an gebotener Stelle herauszuheben. Dass die übrigen Abschnitte und Kapitel in der ersten Person Plural formuliert sind, gebietet bereits der Umstand, dass alle Beiträge dieser Arbeit zu einem bestimmten Grad eine Leistung mehrerer Personen sind.

Akademische Veröffentlichungen. Die Dissertation baut auf acht Fachbeiträgen auf, die auf nationalen und internationalen Konferenzen vorgestellt wurden oder in anerkannten Fachzeitschriften erschienen sind:

[HM16] Vincent Haupert und Tilo Müller. „Auf dem Weg verTAN: Über die Sicherheit App-basierter TAN-Verfahren“. In: *Sicherheit 2016: Sicherheit, Schutz und Zuverlässigkeit, Beiträge der 8. Jahrestagung des Fachbereichs Sicherheit der Gesellschaft für Informatik e.V. (GI), 5.-7. April 2016, Bonn.* Hrsg. von Michael Meier, Delphine Reinhardt und Steffen Wendzel. Bd. 256. LNI. GI, 2016, S. 101–112.

[HHF17] Jochen Hoffmann, Vincent Haupert und Felix Freiling. „Anscheinsbeweis und Kundenhaftung beim Online-Banking“. In: *Zeitschrift für das gesamte Handelsrecht und Wirtschaftsrecht* 181 (2017), S. 780–816.

[HMM17] Vincent Haupert, Dominik Maier und Tilo Müller. „Paying the Price for Disruption: How a FinTech Allowed Account Takeover“. In: *Reversing and Offensive-oriented Trends Symposium*. ACM, 2017.

[HM18] Vincent Haupert und Tilo Müller. „On App-based Matrix Code Authentication in Online Banking“. In: *Proceedings of the 4th International Conference on Information Systems Security and Privacy, ICISSP 2018, Funchal, Madeira - Portugal, January 22-24, 2018.* Hrsg. von Paolo Mori, Steven Furnell und Olivier Camp. SciTePress, 2018, S. 149–160.

[HP18] Vincent Haupert und Gaston Pugliese. „Ich sehe was, das du nicht siehst: Die Realität von Mobilebanking zwischen allgemeinen und rechtlichen Anforderungen“. In: *Sicherheit 2018, Beiträge der 9. Jahrestagung des Fachbereichs Sicherheit der Gesellschaft für Informatik e.V. (GI), 25.-27.4.2018, Konstanz.* Hrsg. von Hanno Langweg, Michael Meier, Bernhard C. Witt und Delphine Reinhardt. Bd. P-281. LNI. Gesellschaft für Informatik e.V., 2018, S. 171–182.

[Hau+18] Vincent Haupert, Dominik Maier, Nicolas Schneider, Julian Kirsch und Tilo Müller. „Honey, I Shrunk Your App Security: The State of Android App Hardening“. In: *Detection of Intrusions and Malware, and Vulnerability Assessment - 15th International Conference, DIMVA 2018, Saclay, France, June 28-29, 2018, Proceedings*. Hrsg. von Cristiano Giuffrida, Sébastien Bardin und Gregory Blanc. Bd. 10885. Lecture Notes in Computer Science. Springer, 2018, S. 69–91.

[HG19a] Vincent Haupert und Stephan Gabert. „Short Paper: How to Attack PSD2 Internet Banking“. In: *Financial Cryptography and Data Security - 23rd International Conference, FC 2019, Frigate Bay, Saint Kitts and Nevis, February 18–22, 2019*. Hrsg. von Ian Goldberg und Tyler Moore. Bd. 11598. Lecture Notes in Computer Science. Springer, 2019.

[HG19b] Vincent Haupert und Stephan Gabert. „Where to Look for What You See Is What You Sign? User Confusion in Transaction Security“. In: *Computer Security - 24th European Symposium on Research in Computer Security, ESORICS 2019, Luxembourg, September 23-27, 2019, Proceedings, Part I*. Hrsg. von Kazue Sako, Steve Schneider und Peter Ryan. Bd. 11735. Lecture Notes in Computer Science. Springer, 2019.

Abgesehen von einer Ausnahme sind die genannten Publikationen auf meine Initiative hin entstanden. Dennoch stellen alle Beiträge eine Zusammenarbeit mit anderen Personen dar und sind zum Teil auf Forschungsarbeiten zurückzuführen, die bereits im Rahmen anderer Qualifizierungsarbeiten begonnen wurden. Im Folgenden wird deshalb offengelegt, in welcher Form und in welchem Umfang die Veröffentlichungen meine Leistung sind. An welcher Stelle die Beiträge in die einzelnen Kapitel der Dissertation einfließen, wird in Abschnitt 1.5 gezeigt:

- Die Publikation „Auf dem Weg verTAN: Über die Sicherheit App-basierter TAN-Verfahren“ [HM16] ist aus meinem Artikel „(Un-)Sicherheit App-basierter TAN-Verfahren im Onlinebanking“ entstanden, der dem Lehrstuhl für IT-Sicherheitsinfrastrukturen der Friedrich-Alexander-Universität Erlangen-Nürnberg zum Wintersemester 2015/2016 im Rahmen des IT-Sicherheitskonferenzseminars vorgelegt wurde. Sowohl die Implementierung als auch die schriftliche Ausarbeitung sind meine Eigenleistung. Dr.-Ing. Tilo Müller hat als Betreuer für die Arbeit fungiert. Mein Vortrag „(Un)Sicherheit von App-basierten TAN-Verfahren im Onlinebanking“ [Hau15] auf dem 32. Chaos Communication Congress ging der Publikation voraus.

- Gemeinsam mit Prof. Dr. Jochen Hoffmann und Prof. Dr.-Ing. Felix Freiling ist der Zeitschriftenartikel „Anscheinsbeweis und Kundenhaftung beim Online-Banking“ [HHF17] erschienen. Herr Hoffmann gab die Initiative zu dem Beitrag und hat den größten Anteil an den Inhalten, die weitgehend juristischer Natur sind. Insbesondere die Abschnitte III und IV (S. 793-804) sind jedoch wesentlich durch meine Person entstanden.

- Teile der Ergebnisse des Beitrags „Paying the Price for Disruption: How a FinTech Allowed Account Takeover“ [HMM17] sind auf Forschungsarbeiten zurückzuführen, die bereits in meiner Masterarbeit „On the Security of App-based Authentication Methods“ begonnen wurden. Die Arbeit wurde durch Dr.-Ing. Tilo Müller betreut und lag dem Lehrstuhl für IT-Sicherheitsinfrastrukturen der Friedrich-Alexander-Universität Erlangen-Nürnberg zum Oktober 2016 vor. Zusätzlich tritt auch Dominik Maier als Co-Autor auf. Herr Maier unterstützte mich nicht nur bei der praktischen Analyse, sondern auch bei der Abfassung der Publikation. Noch im Vorfeld der Veröffentlichung präsentierte ich die Ergebnisse in dem Vortrag „Shut Up and Take My Money! The Red Pill of N26 Security“ [Hau16b] auf dem 33. Chaos Communication Congress.

- Die Veröffentlichung „On App-based Matrix Code Authentication in Online Banking“ [HM18] basiert ebenfalls auf Resultaten, für die der Grundstein bereits in meiner Masterarbeit gelegt wurde.

- Die Publikation „Ich sehe was, das du nicht siehst: Die Realität von Mobilebanking zwischen allgemeinen und rechtlichen Anforderungen“ [HP18] ist in Co-Autorenschaft mit Gaston Pugliese entstanden. Herr Pugliese hat an Idee und Konzeption mitgewirkt, die Abfassung des Beitrags ist jedoch allein auf meine Person zurückzuführen.

- Der Artikel „Honey, I Shrunk Your App Security: The State of Android App Hardening“ [Hau+18] ist in Teilen auf Entwicklungsarbeiten zurückzuführen, die Nicolas Schneider im Rahmen seiner Masterarbeit „Automatic Replacement of an Android Application Security Solution with Malware“ durchgeführt hat. Herr Schneider führte seine Arbeit unter meiner Betreuung auf Basis einer vorangegangenen, allein von mir durchgeführten Machbarkeitsstudie durch. An der schriftlichen Ausarbeitung haben ferner Dominik Maier, Julian Kirsch und Dr.-Ing. Tilo Müller mitgewirkt. Noch vor der Publikation des

Artikels wurden die Ergebnisse im Vortrag „Die fabelhafte Welt des Mobilebankings“ [Hau17a] auf dem 34. Chaos Communication Congress durch mich präsentiert.

– An der Veröffentlichung „Short Paper: How to Attack PSD2 Internet Banking“ [HG19a] hat Stephan Gabert mitgewirkt. Während die Idee, die Inhalte und die Ausarbeitung ihren Ursprung in mir finden, hat Herr Gabert durch Prototypen für die Angriffe einen Beitrag zu der Publikation geleistet.

– Auch an dem Beitrag „Where to Look for What You See Is What You Sign? User Confusion in Transaction Security“ [HG19b] ist Stephan Gabert im Rahmen seiner von mir betreuten Masterarbeit „On the Practical Security of Two-Factor Authentication in Online Banking“ beteiligt. Herr Gabert hat die Studienplattform implementiert; die Durchführung und Auswertung erfolgten gemeinsam. Der Artikel wurde überwiegend von mir verfasst.

Nichtakademische Veröffentlichungen. Neben den akademischen Publikationen war es uns wichtig, unsere Forschung auch direkt an die Verbraucher, Banken, Fintechs und Behörden zu kommunizieren. Hervorzuheben sind meine Vorträge beim Chaos Communication Congress, die in allen Fällen die Grundlage für eine akademische Publikation waren. Die Einreichungen werden im Peer-Review-Verfahren beurteilt und unterliegen einem kompetitiven Wettbewerb:

[Hau15] Vincent Haupert. „(Un)Sicherheit von App-basierten TAN-Verfahren im Onlinebanking“. 32nd Chaos Communication Congress (32c3). Hamburg. 28. Dez. 2015. URL: `https://media.ccc.de/v/32c3-7360-un_sicherheit_von_app-basierten_tan-verfahren_im_onlinebanking`. Vortrag.

[Hau16b] Vincent Haupert. „Shut Up and Take My Money! The Red Pill of N26 Security“. 33rd Chaos Communication Congress (33c3). Hamburg. 27. Dez. 2016. URL: `https://media.ccc.de/v/33c3-7969-shut_up_and_take_my_money`. Vortrag.

[Hau17a] Vincent Haupert. „Die fabelhafte Welt des Mobilebankings“. 34th Chaos Communication Congress (34c3). Leipzig. 27. Dez. 2017. URL: `https://media.ccc.de/v/34c3-8805-die_fabelhafte_welt_des_mobilebankings`. Vortrag.

Darüber hinaus wurde ich zu den folgenden Gastbeiträgen, Interviews, Paneldiskussionen und Podcasts eingeladen:

1) Ingo Dachwitz und Vincent Haupert. „Sicherheitsmängel beim Bank-Startup N26: Eine Frage der Prioritäten“. In: *Netzpolitik.org* (27. Dez. 2016). URL: https://netzpolitik.org/2016/sicherheitsmaengel-beim-bank-startup-n26-eine-frage-der-prioritaeten. Interview.

2) Vincent Haupert. „Eine bedenkliche Abwärtsspirale“. In: *Spiegel Online* (27. Dez. 2016). URL: http://spon.de/aeTu9. Gastbeitrag.

3) André M. Bajorat, Vincent Haupert, Rafael Otero und Kilian Thalhammer. „Sicherheit bei Fintechs“. Paymentandbanking FinTech Podcast. 9. Jan. 2017. URL: https://soundcloud.com/paymentandbanking/fintech-podcast-83-sicherheit-bei-fintechs. Podcast.

4) Ingo Dachwitz und Vincent Haupert. „Sparkassen, Volksbanken, DKB und Co.: Interview über Sicherheitsprobleme beim mobilen Banking“. In: *Netzpolitik.org* (28. Dez. 2017). URL: https://netzpolitik.org/2017/sparkassen-volksbanken-dkb-und-co-interview-ueber-sicherheitsprobleme-beim-mobilen-banking. Interview.

5) Rudolf Linsenbarth und Vincent Haupert. „Erst Sparkassen, jetzt N26: Mobile Banking muss sicherer werden – Interview mit Vincent Haupert“. In: *IT-Finanzmagazin* (10. Jan. 2017). URL: https://www.it-finanzmagazin.de/erst-sparkassen-jetzt-n26-mobile-banking-muss-sicherer-werden-interview-mit-vincent-haupert-42893. Interview.

6) Hakan Tanriverdi und Vincent Haupert. „Banken müssen nicht alles anbieten, was technisch möglich ist“. In: *Süddeutsche.de* (24. Nov. 2017). URL: https://sz.de/1.3762995. Interview.

7) Dennis Kogel und Vincent Haupert. „Hacker erklärt die einzig sichere Methode für Online-Banking“. In: *Motherboard* (3. Mai 2018). URL: https://motherboard.vice.com/de/article/7xdge9/sicheres-online-banking-mit-chiptan-hacker-erklart. Interview.

8) Rafael Otero und Vincent Haupert. „PSD2 Security“. Paymentandbanking FinTech Podcast. 16. März 2018. URL: https://soundcloud.com/paymentandbanking/fintech-podcast-146-psd2-security. Podcast.

9) Rafael Otero, Vincent Haupert und Frank Rieger. „Cybercrime“. Payment Exchange 2018. Berlin. 26. Jan. 2018. Panel.

10) Thomas Rosenhain und Vincent Haupert. „Alles auf einem Gerät“. In: *SparkassenZeitung* 81.01-02 (12. Jan. 2018), S. 2. Interview.

Sonstige Auftritte. Neben den öffentlich verfügbaren und abrufbaren Beiträgen haben wir unsere Forschungsergebnisse auch im Rahmen von Veranstaltungen kommuniziert, die in geschlossener Gesellschaft stattfanden und nicht aufgezeichnet wurden. Die Vorträge und Paneldiskussionen fanden alle im Themenbereich der Banken- und Fintech-Sicherheit statt und erfolgten auf Einladung von Privatunternehmen, öffentlichen Stellen oder Interessenverbänden. Ich bin während meiner Zeit als Doktorand am Lehrstuhl für IT-Sicherheitsinfrastrukturen bei den folgenden nichtöffentlichen Veranstaltungen aufgetreten:

1) Eva Bahner, Kerstin Backofen, Markus Feck, Vincent Haupert und Olaf Jacobsen. „Bezahlen und überweisen im Internet“. Deutschlandfunk Marktplatz. Köln/Nürnberg. 13. Juli 2017. Radiodiskussion.

2) Markus Gürne, Vincent Haupert, Robert Herzig, Reinhold Pamler und Martin Schallbruch. „Sicherheit im Zahlungsverkehr“. American Express Insights Network. Frankfurt. 14. Sep. 2017. Panel.

3) Vincent Haupert. „Mobile First Meets Safety First?“ Reiner SCT Bankentag 2017. Bochum. 9. März 2017. Vortrag.

4) Vincent Haupert. „Sicherer Zugriff auf das Bankkonto“. Karten-Forum 2017 des Deutschen Genossenschafts-Verlags. Bad Homburg. 7. Nov. 2017. Vortrag.

5) Vincent Haupert. „Appsolut sicher?“ Elster Dialog. Starnberg. 22. Jan. 2018. Vortrag.

6) Vincent Haupert. „Fintech Security“. Mastercard Advisory Board. Berlin. 11. Okt. 2018. Vortrag.

7) Vincent Haupert. „Sicherheit beim Mobilebanking“. 40. Bankengespräch des Landeskriminalamts Baden-Württemberg. Stuttgart. 12. Apr. 2018. Vortrag.

8) Vincent Haupert. „Sicherheitsanforderungen im Digital Banking“. Bitkom Arbeitskreis Digitaler Zahlungsverkehr. Berlin. 5. Juli 2018. Vortrag.

9) Thomas Sauerlaender, Vincent Haupert, Matthias Hönisch und Christian Schollmeyer. „Zahlen Sie eigentlich schon mit Ihrem Smartphone?“ Kartensicherheit 2018. Berlin. 25. Sep. 2019. Panel.

1.5 Inhaltsüberblick

Die Arbeit beschäftigt sich in den nachfolgenden Kapiteln 2 bis 7 mit der Beantwortung der in Abschnitt 1.2 formulierten fünf Forschungsfragen (F1 bis F5). Die Inhalte lassen sich in zwei informelle Themengebiete zerlegen: Im ersten Teil befassen sich die Kapitel 2 bis 5 mit den Sicherheitsimplikationen der technischen Innovationen im Mobilebanking. Im zweiten Teil dominieren regulatorische und vertragsrechtliche Motive die Inhalte. Wie in Abschnitt 1.4 erwähnt, speisen sich die Forschungsbeiträge dieser Dissertation aus bereits existierenden Veröffentlichungen. Im Einzelfall basieren Kapitel auf einer alleinstehenden Publikation; überwiegend fließen jedoch unterschiedliche Aspekte einer Publikation in verschiedene Kapitel und Abschnitte ein. Welche Publikation an welcher Stelle innerhalb der Arbeit einbezogen wurde, zeigt Abbildung 1.1 im Detail. Die Kapitel gehen aber regelmäßig und zum Teil erheblich über die Inhalte der Publikationen hinaus, um Hintergründe zu verdeutlichen und Entwicklungen einzubeziehen, die sich erst nach dem Veröffentlichungszeitpunkt der einzelnen Forschungsbeiträge ergaben.

In Kapitel 2 klären wir zunächst grundlegende Abläufe und Begriffe im Online- und Mobilebanking. Im Anschluss skizzieren wir die klassischen Sicherungsverfahren, bevor wir die neuen App-basierten Sicherungsverfahren beschreiben und einordnen. Den Abschluss des Kapitels bildet eine Diskussion zur Gerätetrennung im Mobilebanking.

Welche Auswirkungen die Einführung App-basierter Sicherungsverfahren und die Aufhebung der Gerätetrennung im Mobilebanking haben, wird in Kapitel 3 ausgeführt. Hierfür stellen wir zuerst unser Angreifermodell vor und motivieren es durch die Sicherheit mobiler Endgeräte, ehe wir zwei grundsätzliche Angriffe vorstellen.

In Kapitel 4 sind softwarebasierte App-Härtungsmaßnahmen Untersuchungsgegenstand, wie sie von einer Vielzahl von Unternehmen angeboten werden. Die Produkte werben mit einem vollumfänglichen Schutz und sind nicht zuletzt deswegen insbesondere bei deutschen Banken beliebt. Anhand des in der deutschen Kreditwirschaft führenden Herstellers prüfen wir, ob die Lösung dem selbstgesteckten Anspruch gerecht wird.

Im folgenden Kapitel 5 gehen wir der Frage nach, wie es bei Fintechs um die Sicherheit bestellt ist. Zu diesem Zweck führen wir eine detaillierte Sicherheitsanalyse des in Deutschland führenden Fintechs N26 durch und decken sowohl im Front- als

auch im Backend zum Teil schwere Sicherheitslücken auf. Im Anschluss diskutieren wir, wie diese Sicherheitslücken in der Praxis zu erfolgreichen Angriffen hätten führen können.

Kapitel 6 setzt sich mit der Regulierung durch die Zahlungsdiensterichtlinie II auseinander, die das Sicherheitsniveau aller Fernzahlungsvorgänge erhöhen will. Wir entwickeln zuerst allgemeine Voraussetzungen, die für die digitalen Banktransaktionen gelten müssen. Im Anschluss stellen wir die Anforderungen der Technischen Regulierungsstandards an die starke Kundenauthentifizierung dar und vergleichen die regulatorischen mit unseren allgemeinen Voraussetzungen. Im nächsten Schritt liefern wir eine Einschätzung, ob die im Online- und Mobilebanking gängigen Sicherheitsanforderungen den regulatorischen Anforderungen entsprechen. Den Abschluss des Kapitels bildet eine Vorstellung weiterer Angriffspunkte, die durch die neue Regulatorik nicht erfasst werden und in Zukunft relevant werden könnten.

Kapitel 7 beschäftigt sich mit vertragsrechtlichen Nutzerverpflichtungen als Teil der Transaktionssicherheit. Ein zentraler Bestandteil ist, dass der Kunde die Richtigkeit der Zahlungsdaten in allen Authentifizierungsschritten kontrollieren und bei Unregelmäßigkeiten die Transaktion abbrechen muss. In einer Studie mit 100 Onlinebanking-Nutzern gehen wir der Frage nach, ob und wie die Kunden diesen Sorgfaltspflichten nachkommen.

Im letzten Kapitel 8 schließen wir die Dissertation mit einer Zusammenfassung unserer Ergebnisse und einem Ausblick auf weitere Forschungsfragen ab.

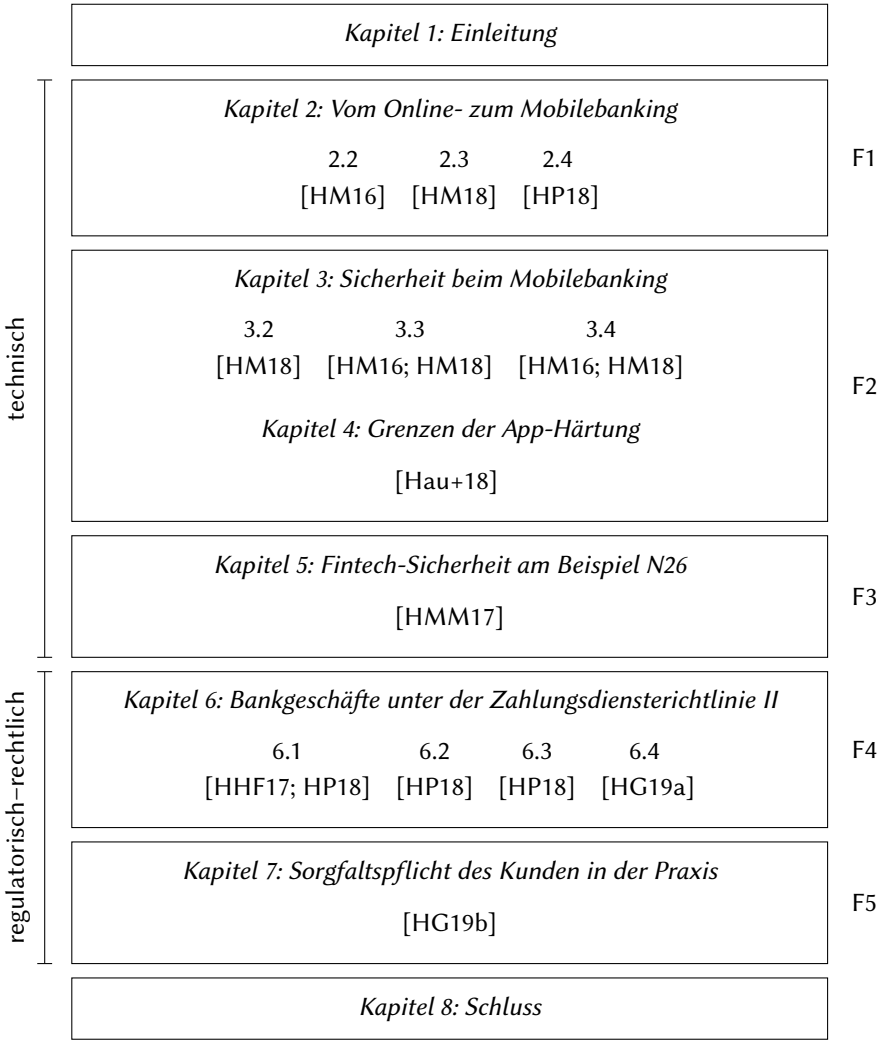


Abbildung 1.1: Aufbau der Dissertationsinhalte und Übersicht über die Verwendung der in Abschnitt 1.4 genannten akademischen Veröffentlichungen in den einzelnen Kapiteln und Abschnitten.

2

Vom Online- zum Mobilebanking

Im Bereich des Zahlungswesens hat sich eine besondere Kreativität in der Auslegung der Eigenschaft einer Zweifaktorauthentisierung entwickelt.

— Jens Bender und Dennis Kügler, BSI, 2016 [BK16]

In diesem Kapitel stellen wir die Entwicklung der Sicherungsverfahren dar, die mit den listenbasierten Verfahren im Onlinebanking begann und in den App-basierten Verfahren des Mobilebankings ihr vorläufiges Ende finden. Zu diesem Zweck klären wir zunächst die Grundlagen der Transaktionssicherheit sowie zentrale Begriffe. Hierbei ist insbesondere die Abgrenzung des Online- zum Mobilebanking hervorzuheben. Danach skizzieren wir eine Auswahl klassischer Sicherungsverfahren im Onlinebanking und bewerten sie hinsichtlich ihrer Sicherheitseigenschaften. Anschließend stellen wir die App-basierten Sicherungsverfahren vor, die zwar auch im Onlinebanking Verwendung finden, vor allem aber das Mobilebanking bedingen. Den Abschluss bildet die Diskussion zur Auflösung der Gerätetrennung im Mobilebanking.

2.1 Grundlagen

Zwei-Faktor-Authentifizierung. Transaktionen im Onlinebanking folgen seit seiner Einführung dem Prinzip einer Zwei-Faktor-Authentifizierung (2FA). Dabei handelt es sich jedoch nicht um eine Benutzer-, sondern um eine Transaktionsauthentifizierung in zwei Schritten. Demnach fordert auch nicht jeder Login in das Onlinekonto eine 2FA. Es ist vielmehr so, dass der erste Faktor dem Onlinebanking-

Kunden nur lesenden Zugriff auf sein Konto gewährt. In diesem Rechtekontext ist es ihm z. B. möglich, seinen Kontostand und gebuchte Transaktionen zu prüfen. Vielfach kann er auch seine Stammdaten einsehen. Jede Änderung, sei es durch das Transferieren von Guthaben oder Anpassung der Kundendaten, muss jedoch jeweils und für sich durch einen zweiten Faktor autorisiert werden. Die 2FA unterteilt Aktionen im Onlinebanking also in zwei Privilegienebenen.

Der erste Faktor ist damals wie heute eine Kombination aus Benutzername und Passwort. Der zweite Faktor tritt seit der Einführung des Onlinebankings in der Form eines Einmalpassworts auf, das im Kontext von digitalen Bankgeschäften Transaktionsnummer (TAN) genannt wird. Die Art und Weise, wie der Kunde eine TAN erhält, hängt dabei stark von dem eingesetzten Sicherungsverfahren ab, das wechselweise auch Legitimierungsverfahren oder TAN-Verfahren genannt wird. Nach Anwendung des Sicherungsverfahrens erhält der Kunde die zumeist sechsstellige, numerische TAN, die er dann im Onlinebanking eingeben muss. Ist die eingegebene TAN gültig, wird der Auftrag durch die Bank ausgeführt.

What-You-See-Is-What-You-Sign. In Abhängigkeit von dem eingesetzten Sicherungsverfahren werden die mit dem ersten Faktor authentifizierten, bei der Bank eingegangenen Auftragsdaten dem Kunden nochmals separat angezeigt. Im Rahmen der Vertragsbedingungen für die Nutzung des Onlinebankings verpflichtet die Bank den Kunden auf bestimmte Sorgfaltspflichten. Hierzu zählt auch, dass der Kunde zu überprüfen hat, ob die im Sicherungsverfahren dargestellten Auftragsdaten mit den gewünschten übereinstimmen. Ist dies nicht der Fall, muss der Kunde die Transaktion abbrechen. Dieses Prinzip ist als What-You-See-Is-What-You-Sign (WYSIWYS) bekannt [LP98].

PIN/TAN-Verfahren. Die geschilderte zweitstufige Absicherung des Onlinebankings wird von der Deutschen Kreditwirtschaft als PIN/TAN-Verfahren bezeichnet [DK14]. Der Begriff ist nach wie vor geläufig, obwohl bereits die Nomenklatur „Persönliche Identifikationsnummer“ ungenau ist: das Geheimnis darf in vielen Fällen nicht nur aus Zahlen bestehen, sondern muss auch Buchstaben und Sonderzeichen enthalten. Dementsprechend ist es treffender, von einem Passwort zu sprechen. Ähnliches gilt für die „TAN“: manche Sicherungsverfahren sind mittlerweile derart gestaltet, dass sie über einen elektronischen Rückkanal verfügen. In diesen Fällen ist oft keine für den Kunden sichtbare TAN mehr beteiligt. Dementsprechend reduziert sich die Tätigkeit des Kunden auf das Kontrollieren der angezeigten Transaktionsdaten im Sicherungsverfahren gemäß WYSIWYS.

Transaktionsauslösung und -bestätigung. Da sich die Dissertation vordergründig mit dem kritischsten Geschäftsvorfall – dem Tätigen von Transaktionen – auseinandersetzt, bemühen wir regelmäßig das Szenario, in dem sich ein Kunde nur zu dem Zweck in sein Banking-Portal einloggt, um eine Überweisung zu tätigen. Die beiden zugehörigen Schritte werden deshalb in die Transaktionsauslösung (der Kunde loggt sich mit dem ersten Faktor ein und füllt einen gültigen Überweisungsauftrag aus) und die Transaktionsbestätigung (der Kunde autorisiert die Transaktion mit dem zweiten Faktor) zerlegt.

Online- und Mobilebanking. Die Begriffe Onlinebanking und Mobilebanking sind jeweils für sich und in Beziehung zueinander unscharf definiert. Onlinebanking wird nicht selten als diffuser Sammelbegriff für jedwede Aktion verwendet, die im Kontext von digitalen Bankgeschäften durchgeführt wird.

Die Bezeichnung Mobilebanking nimmt hingegen Bezug darauf, dass Bankgeschäfte auch von unterwegs aus jederzeit getätigt werden können. Damit kann nicht gemeint sein, dass die am Onlinebanking beteiligte Infrastruktur mobil im Sinne von transportabel sein muss: Der Zugang zum Banking-Portal ist mit Benutzername/Passwort gesichert und somit ohnehin nicht an einen Ort gebunden. Das persönliche Sicherungsverfahren erfordert ebenfalls keine stationäre Installation und ist hinreichend kompakt, um es ohne Weiteres transportieren zu können. Es muss also gelten, dass die an Bankgeschäften beteiligte Infrastruktur in dem Sinn mobil ist, dass sie der Kunde auch abseits des Mobilebankings mit sich führt.

Außerdem muss sich im Mobilebanking ein vollwertiger Zugriff auf das Konto ergeben. Ein Authentifizierungssystem genügt also nicht den Ansprüchen des Mobilebankings, wenn es lediglich lesenden Zugriff mittels Benutzername/Passwort erlaubt. Der Kunde muss auch Transaktionen tätigen können. Damit er das Sicherungsverfahren ähnlich zu seinen Zugangsdaten beliebig zur Verfügung hat, muss es durch eine Infrastruktur realisiert sein, die der Kunde unabhängig von der Bank immer mit sich führt. Es kann sich dabei also nicht um ein physisches Medium handeln, das von der Bank dediziert für digitale Bankgeschäfte ausgegeben wird.

Nach aktueller Sachlage erfüllen nur mobile Endgerät wie Smartphones, Tablets oder Smartwatches die Voraussetzung, dass sie im Allgemeinen zu jederzeit mitgeführt werden. Sie verfügen darüber hinaus über eine ständige Internetverbindung und eine Erweiterbarkeit durch Apps Dritter. Wir definieren Online- und Mobilebanking wie folgt trennscharf: Die Transaktionsauslösung und -bestätigung erfolgen beim Onlinebanking physisch getrennt, während sie beim Mobilebanking auf ein und demselben mobilen Endgerät stattfinden.

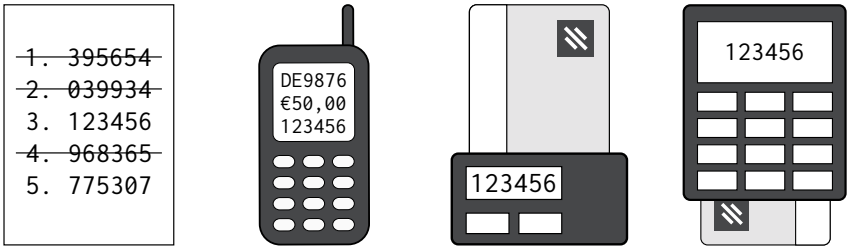
2.2 Klassische Sicherungsverfahren

In diesem Abschnitt beschreiben wir die Sicherungsverfahren, die aus dem klassischen Onlinebanking heraus entstanden sind. Die Verfahren setzen gemäß unserer obigen Definition voraus, dass die Transaktion auf einem Gerät ausgelöst wird und mithilfe eines anderen Mediums bestätigt wird. Für den weiteren Verlauf dieses Abschnitts nehmen wir an, dass der Kunde bereits eine Transaktion ausgelöst hat. Die Bestätigung erfolgt dann durch eines der im Folgenden beschriebenen Sicherungsverfahren. Eine Auswahl ist in Abbildung 2.1 schematisch dargestellt.

Das PIN/TAN-Verfahren ist älter als das Onlinebanking selbst [HB98]: Es wurde bereits 1976 von Alfred Richter erfunden und diente anfänglich dazu, die Konten der Mitarbeiter bei der Verbraucherbank vor den eigenen Kollegen zu schützen. Mit dem Onlinebanking wurde das Verfahren auch für die Endkunden adaptiert.

2.2.1 Listen

TAN-Listen sind das älteste Sicherungsverfahren im Onlinebanking, werden bei einigen Privatbanken aber auch heute noch eingesetzt. Der Kunde erhält dabei eine Papierliste, auf der eine Vielzahl zumeist sechsstelliger TANs aufgedruckt ist. Die Eingabe der TAN gibt die Transaktion frei und macht die verwendete TAN ungültig. Neigt sich die Zahl der noch verfügbaren TANs dem Ende zu, sendet die Bank dem Kunden automatisch eine neue Liste. Historisch betrachtet gibt es zwei Arten von TAN-Listen, die klassische und die indizierte.



(a) iTAN (b) smsTAN (c) smartTAN (d) chipTAN

Abbildung 2.1: Auswahl klassischer Sicherungsverfahren.

Klassische TAN-Liste

Diese Variante ist heute nicht mehr im Einsatz, war aber lange das vorherrschende Sicherungsverfahren. Bei der klassischen TAN-Liste waren die TANs ohne Ordnung abgedruckt, weshalb der Kunde eine beliebige TAN nutzen konnte, um jede Transaktion zu bestätigen.

Sicherheit. Mit der weiteren Verbreitung des Onlinebankings wurden Kunden zunehmend Ziel von Phishing-Angriffen, die nicht nur die Zugangsdaten des Kunden, sondern auch die eingesetzte TAN mitschnitten. Da die Bank jede beliebige zuvor noch nicht verwendete TAN zur Transaktionsbestätigung akzeptierte, konnte der Angreifer sie im Verbund mit den Zugangsdaten für eine Transaktion seiner Wahl verwenden.

iTAN

Um dem Phishing bei der klassischen TAN-Liste entgegenzuwirken, haben die Sparkassen [DSZ05; FW05] und die Postbank [BuM05] im Sommer 2005 damit begonnen, die indizierte TAN-Liste (iTAN) auszurollen (siehe auch Abbildung 2.1a). Im Gegensatz zur klassischen ist die iTAN-Liste durchnummeriert: bei der Transaktionsbestätigung fordert die Bank den Kunden auf, statt einer beliebigen eine ganz bestimmte TAN einzugeben. Da die Bank den Index der gültigen TAN per Zufall bestimmt, ist eine per Phishing gewonnene TAN weniger wertvoll.

Sicherheit. Obwohl das Verfahren die Schadensfälle durch Phishing zunächst eindämmen konnte [Sto05], ist das Verfahren nicht ohne Fehler. Kurz nachdem einige Banken die Einführung des iTAN-Verfahrens ankündigten, hat RedTeam Pentesting auf eine konzeptionelle Schwäche des Verfahrens hingewiesen [RTP05]. Zwar kann das Verfahren unterbinden, dass ein Angreifer eine durch Phishing erlangte TAN später für eine eigenständige Transaktion verwenden kann, kann aber einen Man-in-the-Middle- oder Man-in-the-Browser-Angriff nicht verhindern. Bei einem solchen Angriff ist entweder die Netzwerkkommunikation oder das transaktionsauslösende Gerät kompromittiert, sodass der Angreifer in der Lage ist, die bei der Bank eingehenden Transaktionsdaten beliebig zu verändern. Da der Angreifer auch die Daten kontrolliert, die das Opfer von der Bank empfängt, kann er zudem dafür sorgen, dass dieser die erwarteten Transaktionsdaten sieht. In Konsequenz hat das Opfer keine Möglichkeit, einen atypischen Verlauf festzustellen.

Das zunächst theoretische Angriffsszenario sorgte später für hohe Schadensfälle, weshalb es bereits 2011 von der Postbank [ZGK11] und 2012 von den Sparkassen [Lip13] komplett durch andere Verfahren ersetzt wurde. Trotz seiner flagranten Sicherheitsprobleme ist das iTAN-Verfahren auch Anfang 2019 noch breit im Einsatz [Sei19].

iTAN++

Das Verfahren iTAN++ funktioniert analog zur iTAN, nur dass die Aufforderung neben dem Index der TAN auch noch die Transaktionsdetails mit der Bestätigungsnummer (BEN) als Wasserzeichen zeigt. Die TAN-Liste enthält dabei neben der TAN auch noch eine Zuordnung zur BEN. Bevor der Kunde die TAN mit dem entsprechenden Index überträgt, muss er die BEN im Hintergrund und die Transaktionsdetails im Vordergrund noch zusätzlich prüfen.

Sicherheit. Grundsätzlich sind gegen das Verfahren die gleichen Angriffe möglich wie gegen iTAN. Gegenüber Echtzeitmanipulationen bietet die iTAN++ in der Theorie ein leicht erhöhtes Maß an Sicherheit, weil die Auftragsdetails noch einmal in derselben Pixelgrafik dargestellt werden, die auch die BEN im Hintergrund zeigt. Wenn ein Angreifer die Auftragsdetails bei der Transaktionsauslösung nun unbemerkt im Hintergrund austauscht, muss er auch das Bild mit den Transaktionsdetails, dem Index der TAN auf der Liste und der BEN im Hintergrund nachstellen. Die Sicherheit dieses Vorgehens fußt dabei auf der Annahme, dass es hinreichend schwer ist, die BEN im Hintergrund automatisiert zu extrahieren und wiederum eigens maschinell ein Bild zu erstellen, das die gewünschten Transaktionsdetails zeigt. Dass solche Systeme dennoch automatisiert gebrochen werden können, hat die Forschung bereits vor Jahren gezeigt [Li+10]. In der Praxis darf zudem angezweifelt werden, dass der Kunde die bei der iTAN++ zusätzlich durchzuführenden Verifikationsschritte versteht. Auch ob der Nutzer eine Aufforderung, die nur den Index der TAN enthält, für ungewöhnlich halten würde, bleibt ungewiss.

2.2.2 Mobilfunk

Mobilfunkgeräte sind seit einiger Zeit weit verbreitet. Diese Geräte sind standardmäßig mit einer SIM-Karte bestückt, um mit ihnen Telefonieren und SMS senden sowie empfangen zu können.

smsTAN

Beim smsTAN-Verfahren erhält der Kunde nach Transaktionsauslösung eine SMS von seiner Bank an die hinterlegte Mobilfunknummer. Darin befindet sich nicht nur die TAN, die der Kunde manuell in den transaktionsauslösenden Kanal übertragen muss, sondern auch die wichtigsten Auftragsdaten (siehe Abbildung 2.1b). Das sind zumindest der vollständige Betrag und Teile der Internationalen Bankkontonummer (IBAN). Zusätzlich können auch weitere Metadaten, wie der Zeitpunkt der Transaktionsauslösung, enthalten sein. Bevor der Kunde die TAN überträgt, muss er sicherstellen, dass die Auftragsdaten mit den gewünschten Daten übereinstimmen.

Obwohl die Postbank einem Teil ihrer Kunden schon 2003 das smsTAN-Verfahren anbot, wurde es allen Kunden erst 2005 verfügbar gemacht. Bei den Genossenschaftsbanken und Sparkassen erfolgte die Einführung jeweils in den Jahren 2007 und 2008 [Sei08]. Das Verfahren funktioniert für alle Banken im Wesentlichen analog, trägt aber unterschiedliche Namen. Geläufig sind neben smsTAN auch die Bezeichnungen mobileTAN und mTAN. Zusätzlich wird die IBAN von den Instituten unterschiedlich maskiert. Ein übliches Vorgehen ist es, den Teil der IBAN darzustellen, der der Kontonummer entspricht und den Rest auszusparen.

Sicherheit. Das smsTAN-Verfahren bietet gerade im Vergleich zu listenbasierten Legitimierungsverfahren gute Sicherheitseigenschaften. Zum einen liefert es alle relevanten Transaktionsdaten zur Verifikation zusammen mit der TAN aus. Zum anderen muss das Gerät, das die TAN empfängt, gemäß Vorgabe der Deutschen Kreditwirtschaft verschieden zu dem transaktionsauslösenden Gerät sein [DK]. Da es sich hierbei vielmehr um eine vertragliche als um eine technische Einschränkung handelt, wird die Effektivität dieser Maßnahme in Abschnitt 2.4 diskutiert. Darüber hinaus wurde die Sicherheit des smsTAN-Verfahrens durch eine Reihe von Verwundbarkeiten erodiert, die im Folgenden skizziert werden. Insgesamt erachtet auch das Bundesamt für Sicherheit in der Informationstechnik (BSI) die smsTAN nicht mehr als empfehlenswert [BSI].

Im Jahr 2013 weist Mulliner u. a. darauf hin, dass eine Schadsoftware die in Symbian und Android angebotene Funktion zum Lesen eingehender SMS missbrauchen kann, um diese an den Angreifer weiterzuleiten [Mul+13]. Die Arbeit ist durch ihren Fokus auf moderne Smartphones besonders verwandt zu Kapitel 3, nimmt aber an, dass sich die Schadsoftware im regulären Rechtekontext bewegt. Unter iOS existiert zwar keine explizite Programmierschnittstelle, um eingehende SMS aus einer Drittanbieter-App auszulesen. Konoth, Veen und Bos zeigten 2016 aber, dass

sich die Continuity-Funktionalität von iOS und Mac OS X ausnutzen lässt, um die 2FA auf Basis von SMS auszuhebeln [KVB16]. Continuity sorgt ab iOS 8.1 und Mac OS X 10.10 dafür, dass die auf einem iPhone eingehenden SMS-Nachrichten auch auf den Mac synchronisiert werden. Infolge genügt eine Schadsoftware, die lediglich den Mac befällt, um Zugriff auf die Zugangsdaten und die per SMS versendeten Einmalpasswörter zu erhalten. Einen ähnlichen Angriff gegen iOS 12 und Mac OS X 10.14 schildern wir in Abschnitt 6.4.4.

Engel stellte 2014 mehrere Defizite in der internen Infrastruktur der Mobilfunkanbieter fest, die das SS7-Protokoll nutzten [Eng14]. Durch die in großen Teilen nicht vorhandene Rechte- und Plausibilitätsprüfung gelang es ihm nicht nur, beliebige Nutzer allein durch Kenntnis der Mobilfunknummer ausfindig zu machen, sondern auch deren Konfiguration zu verändern. Das schloss explizit die Möglichkeit ein, SMS-Nachrichten mitzuschneiden oder gar komplett umzuleiten. Zweieinhalb Jahre später wurde die Lücke von Kriminellen auch in der Praxis ausgenutzt, um TANs beim Onlinebanking abzufangen [TZ17].

Mehrere Arbeiten beschäftigen sich mit Man-in-the-Middle-Angriffen auf Basis eines IMSI-Catchers [Fox02; Dab+14; WSM10]. Ein IMSI-Catcher ist eine bösartige Basisstation, die sich zwei Umstände des GSM-Netzes (2G) zunutze macht: Erstens verbindet sich ein Mobilfunktelefon automatisch mit der Station, die das stärkste Signal aussendet. Zweitens muss sich nur das Mobilfunktelefon gegenüber dem Netzwerk authentifizieren, aber nicht umgekehrt. Infolgedessen kann ein Angreifer mittels eines IMSI-Catchers die Kommunikation – und damit auch die SMS-Nachrichten – seines Opfers im Klartext lesen. Der UMTS-Mobilfunkstandard (3G) führte zwar die gegenseitige Authentifizierung ein, kann aber auf GSM-Niveau zurückgestuft werden, weshalb weiter dieselben Angriffe möglich sind [MW04]. Diese Angriffe bleiben auch mit dem bis dato neusten Mobilfunkstandard LTE (4G) möglich [Sha+16; MO17].

2.2.3 Dedizierte Geräte

Dieser Abschnitt beschreibt Verfahren, die auf Geräte setzen, die explizit zur Freigabe von Transaktionen entworfen wurden. Als solche beschränkt sich ihr Funktionsumfang auch einzig und allein auf die Absicherung und Bestätigung von Transaktionen.

eTAN / smartTAN

Sowohl das eTAN-, als auch das smartTAN-Verfahren erzeugen mithilfe eines dedizierten Geräts, dem sog. TAN-Generator, eine zeitlich begrenzte TAN. Der Unterschied der Verfahren liegt darin, wie die Geräte personalisiert werden. Bei der eTAN sind die TAN-Generatoren bereits ab Werk mit einem individuellen Schlüssel ausgestattet, der für die Erstellung der TAN herangezogen wird. Das smartTAN-Verfahren funktioniert ähnlich, verwendet zur Gerätepersonalisierung jedoch keinen integrierten Schlüssel, sondern die persönliche Bankkarte des Kunden (siehe Abbildung 2.1c).

Sicherheit. Beide Verfahren sind unsicher, weil die resultierende TAN in keinem Bezug zu den Auftragsdaten steht. Das Sicherheitsniveau ist mit dem des iTAN-Verfahrens vergleichbar. Infolgedessen kann eine Schadsoftware auf dem transaktionsauslösenden Gerät die Transaktion im Hintergrund beliebig manipulieren; der Kunde hat keine Möglichkeit, den Betrug zu erkennen. Ein Vorteil des Verfahrens gegenüber der iTAN ist, dass es nicht repliziert werden kann.

eTAN+

Das eTAN+-Verfahren stellt gegenüber dem regulären eTAN-Verfahren die TAN in Bezug zu Teilen der Auftragsdaten. Zu diesem Zweck zeigt die Bank dem Kunden im Onlinebanking einen sechsstelligen Kontrollcode an, den der Kunde über die integrierte Tastatur in den eTAN+-Generator eingeben muss. Daraufhin wird dem Kunden die TAN angezeigt, die die Transaktion bestätigt. Der Betrag fließt in die Erstellung der TAN nicht ein.

Sicherheit. Wird das transaktionsauslösende Gerät mit einer Schadsoftware befallen, kann diese eine Transaktionsmanipulation im Hintergrund durchführen. Der Erfolg dieses Angriffs hängt maßgeblich davon ab, inwieweit dem Kunden die Rolle des Kontrollcodes bewusst ist: Hierbei handelt es sich um die letzten sechs Stellen der Ziel-IBAN. Statt den Kunden aufzufordern, die letzten sechs Stellen der IBAN seiner gewünschten Transaktion einzugeben, zeigt das Onlinebanking den Kontrollcode nochmals separat an. Insofern das Opfer den dargestellten Kontrollcode einfach überträgt, kann ein Angreifer eine Transaktion in beliebiger Höhe im Namen des Kunden veranlassen.

chipTAN

Das chipTAN-Verfahren folgt dem Secoder-Standard der Deutschen Kreditwirtschaft [DK14]. Hierfür wird ein Lesegerät mit Anzeige und Tastatur benötigt, in das der Kunde seine Girocard einführt. Das Lesegerät ist ein Klasse-3-Leser, der im Kontext von Transaktion im Onlinebanking auch TAN-Generator genannt wird. Bevor der TAN-Generator aus IBAN und Betrag eine TAN mithilfe der Bankkarte generiert, die nur für diesen Auftrag gültig ist, werden die Transaktionsdetails auf dem integrierten Display noch einmal angezeigt (siehe Abbildung 2.1d). Da die Matrixanzeige des TAN-Generators nur eine geringe Auflösung bietet, werden IBAN, Betrag und die TAN nicht gemeinsam, sondern nacheinander dargestellt und einzeln per Knopfdruck bestätigt.

Wie die Auftragsdaten der Transaktion übertragen werden, ist von der Bank und den Fähigkeiten des TAN-Generators abhängig. Die Generierung der TAN findet dabei immer gleich statt; lediglich der Übertragungsweg ist ein anderer:

- *Manuell*: Der Kunde muss die Auftragsdaten, also die IBAN (oder Teile davon) und den Betrag, erneut in den TAN-Generator eingeben. Zusätzlich zeigt das Onlinebanking dem Kunden noch einen Start-Code an, den er eingeben muss. Der Start-Code sorgt neben einem internen Zähler dafür, dass die resultierende TAN auch bei gleichen Auftragsdaten immer eine andere ist. Dadurch kann die TAN immer nur für diesen einen Auftrag verwendet werden.
- *Flickercode*: Statt die Daten manuell einzugeben, überträgt das chipTAN-Verfahren dieselben Informationen mit einer animierten Grafik – dem Flickercode – halbautomatisch. Diesen Flickercode zeigt das Onlinebanking nach Transaktionsauslösung im Onlinebanking an. Der TAN-Generator liest den Flickercode mithilfe von Dioden, die oben auf der Rückseite des TAN-Generators angebracht sind. Die resultierende TAN muss nach wie vor manuell im Onlinebanking eingegeben werden.
- *Matrixcode/QR-Code*: Eine Weiterentwicklung des Flickercodes ist chipTAN mit Matrix- bzw. QR-Code. Statt einer animierten Grafik wird dem Kunden ein Standbild angezeigt, das die Informationen kodiert.
- *USB/Bluetooth*: Manche TAN-Generatoren unterstützen auch eine bidirektionale Kommunikation mittels Bluetooth oder USB. Dadurch werden nicht nur die notwendigen Daten automatisch an den Leser gesendet, sondern es ist dem Verwender ebenfalls möglich, die TAN per Knopfdruck elektronisch zu

übertragen. Diese Funktionalität ist aus dem browsergestützten Onlinebanking jedoch regelmäßig nicht verfügbar. Finanzverwaltungssoftware auf dem PC oder Smartphone unterstützt die Funktionalität in der Regel.

Sicherheit. Das chipTAN-Verfahren ist unabhängig von dem gewählten Übertragungsweg ein sehr sicheres Verfahren. Zum einen ist der TAN-Generator eigens zur Verwendung des chipTAN-Verfahrens entwickelt worden und bietet eine vertrauenswürdige Anzeige, die eine sichere Verifikation der Auftragsdaten ermöglicht. Zum anderen ist der auf der Girocard aufgebrachte Chip, der auch für die Generierung der TAN zuständig ist, unter praktischen Gesichtspunkten nicht kopierbar. Insbesondere das Anfertigen einer Kopie aus der Ferne ist nicht denkbar.

Im Zuge der Einführung von chipTAN veröffentlichte RedTeam Pentesting einen Angriff gegen das Verfahren [RTP09]. Es nutzte die Möglichkeit zu Sammelüberweisungen, um einen Man-in-the-Middle-Angriff durchzuführen, der durch das chipTAN-Verfahren gerade verhindert werden sollte. In einer Sammelüberweisung können mehrere Transaktionen mit unterschiedlichen Zahlungsempfängern und -beträgen mit einer Bestätigung freigegeben werden. Das chipTAN-Verfahren zeigt in diesem Fall nur den Gesamtbetrag, nicht aber die einzelnen Positionen an. RedTeam Pentesting machte sich zunutze, dass einige Banken Sammelüberweisungen mit nur einer Position nicht in eine Einzelüberweisung umwandelten. Dadurch war es einem Angreifer möglich, den Empfänger der Transaktion zu verschleiern.

Bis dato ist gegen das HHD-Protokoll [DK18], das dem chipTAN-Verfahren zugrundeliegt, noch kein Angriff bekannt geworden. Lösungen auf Basis des verbreiteten Zahlungsprotokolls Europay International, MasterCard und VISA (EMV) waren hingegen angreifbar. Drimer, Murdoch und Anderson zeigten 2009, dass das CAP-Protokoll schon mit deutlichen Mängeln konzipiert wurde [DMA09]. CAP basiert auf dem EMV-Standard und wird z. B. im Vereinigten Königreich eingesetzt, um Onlinebanking-Transaktionen mittels eines Lesegeräts und der Chipkarte des Kunden abzusichern. Die Autoren kritisieren unter anderem, dass mit dem Leser die PIN validiert werden kann, die für Geldabhebungen oder Zahlungen am Point of Sale (PoS) notwendig ist. Diese Entscheidung führt dazu, dass Kriminelle im Fall eines Überfalls die Herausgabe der PIN durch das Opfer an Ort und Stelle validieren können. Ferner gelang es Murdoch u. a. 2010, eine gestohlene Chipkarte zur Zahlung am PoS zu verwenden, ohne die zugehörige PIN zu kennen. Grund war eine Sicherheitslücke im EMV-Protokoll, das keine kryptographischen Signaturen für die Antwortnachrichten der Chipkarte bei der PIN-Abfrage verlangte. Die in Deutschland verbreitete Girocard war von dem Angriff nicht betroffen.

2.3 App-basierte Sicherungsverfahren

Durch die Popularität von Smartphones haben die Banken ab 2013 damit begonnen, App-basierte TAN-Verfahren einzuführen, die zum Teil sehr unterschiedlich funktionieren [FS13; DSZ13]. Je nach Verfahren erlaubt das Authentifizierungsparadigma nur eine Nutzung im Onlinebanking (Zwei-Geräte-Authentifizierung) oder auch im Mobilebanking (Zwei- oder Ein-App-Authentifizierung). Die verschiedenen Authentifizierungsklassen App-basierter Sicherungsverfahren sind in Abbildung 2.2 dargestellt und werden weiter unten beschrieben.

Personalisierung. Alle App-basierten Verfahren müssen nach der Installation zunächst personalisiert werden. Oft ist an diesem Prozess ein Registrierungsbrief beteiligt, um mit der Bank ein gemeinsames Geheimnis auszutauschen, das dann auf dem mobilen Endgerät gespeichert wird. Mithilfe dieses Geheimnisses erzeugt das App-basierte Verfahren unter Bezugnahme eines konkreten Auftrags einen Authentifizierungscode, der die Transaktion bestätigt.

Online/offline. Manche App-basierte Verfahren empfangen die für Transaktionsverifikation benötigten Daten nicht wie beim smsTAN-Verfahren direkt über den Mobilfunk, sondern über ein IP-basiertes Protokoll. Dementsprechend ist für die Internetkommunikation nicht zwangsläufig eine SIM-Karte notwendig, kann aber für eine Datenverbindung über den Mobilfunkanbieter genutzt werden. Andere Verfahren arbeiten komplett offline und empfangen die Transaktionsdaten z. B. über einen Matrix- oder QR-Code.

TAN. Obwohl das PIN/TAN-Verfahren für die Transaktionsbestätigung stets eine für den Nutzer sichtbare TAN beinhaltet hat, erwächst aus der Zahlenfolge per se kein Sicherheitsgewinn, da sie semantisch wertlos ist. Sie ist vielmehr ein Artefakt derer Verfahren, die keinen digitalen Rückkanal bieten. Aus diesem Grund fordern App-basierte Onlineverfahren zunehmend nicht mehr, dass die TAN vom Nutzer manuell in den transaktionsauslösenden Kanal übertragen wird, bzw. verzichten komplett auf eine für den Nutzer sichtbare TAN.

Zugangsschutz. Um das App-basierte Sicherungsverfahren vor unbefugtem Zugriff zu schützen, sichern manche Anbieter die App noch zusätzlich mit einem Zugangsschutz ab. Im Regelfall erfolgt dies über ein zusätzliches Passwort, das beim Öffnen der App und nach einer bestimmten Inaktivitätszeit abgefragt wird. Einen entsprechenden Biometriesensor vorausgesetzt, ist es zum Teil auch möglich, die App mittels Gesichts- oder Fingerabdruckerkennung zu entsperren.

(a) Zwei-Geräte-Authentifizierung

(b) Zwei-App-Authentifizierung

(c) Ein-App-Authentifizierung

Abbildung 2.2: Klassen App-basierter Sicherungsverfahren.

Zwei-Geräte-Authentifizierung

Die ersten App-basierten Sicherungsverfahren waren noch von hoher Kontinuität geprägt, da sie für die Transaktionsauslösung und -bestätigung weiter zwei unterschiedliche Geräte vorsahen. Wir sprechen deshalb von App-basierten Sicherungsverfahren, die eine Zwei-Geräte-Authentifizierung (2GA) implementieren (Abbildung 2.2a).

Die Beteiligung von zwei Geräten erfolgt dabei entweder implizit oder explizit. Ein implizites Verfahren ist so konzipiert, dass zwei Geräte verwendet werden müssen. Nur ein Gerät zu verwenden ist technisch nicht möglich. Bei der expliziten 2GA ist das Verfahren so konzipiert, dass es technisch möglich wäre, auch nur ein Gerät für die Transaktionsauslösung und -bestätigung heranzuziehen. Stattdessen verpflichtet die Bank den Kunden über die Allgemeinen Geschäftsbedingungen (AGB) im Rahmen seiner Sorgfaltspflicht, zwei unterschiedliche Geräte zu verwenden.

Ein Beispiel für eine implizite 2GA sind Verfahren, die nach der Transaktionslösung eine Grafik (z. B. einen Matrix- oder QR-Code) anzeigen, die die Auftragsdaten kodiert. Der Nutzer muss zur Transaktionsbestätigung diese Grafik innerhalb des App-basierten Verfahrens mithilfe der Smartphone-Kamera abfotografieren. Durch diese Funktionsweise muss der Nutzer zwingend zwei Geräte verwenden.

Online-Verfahren implementieren eine explizite 2GA. Solche Verfahren arbeiten oft mit einer Push-Nachricht, die die Transaktionsdaten über das Internet bereitstellt oder signalisiert, dass eine neue Transaktion zur Bestätigung online zum Abruf

bereit steht. Abhängig vom Verfahren ist in der Nachricht auch die TAN enthalten; alternativ wird sie durch die App generiert. Bei Online-Verfahren ist es zum Teil möglich, die Transaktion direkt aus dem App-basierten Sicherungsverfahren heraus zu bestätigen. Der Medienbruch, in dem der Nutzer die TAN manuell in den transaktionsauslösenden Kanal übertragen muss, entfällt. Aus technischen Gesichtspunkten könnte der Nutzer beide Schritte auch nur über ein einziges Gerät abwickeln, weshalb eine vertragliche Bestimmung notwendig wird.

Das qr- und das photoTAN-Verfahren waren die ersten App-basierten Sicherungsverfahren [Wat13]. Beide realisierten eine implizite 2GA.

Zwei-App-Authentifizierung

Die Einführung von Verfahren, die eine Zwei-App-Authentifizierung (2AA) implementieren, markiert das Ende der Gerätetrennung und den Beginn des Mobilebankings. Die 2AA sieht es vor, dass nicht mehr zwei Geräte, sondern zwei Apps auf demselben mobilen Endgerät für Auf- und Freigabe von Transaktionen erforderlich sind (Abbildung 2.2b).

Die erste App ist die Banking-App, die eine für mobile Endgeräte optimierte, ansonsten jedoch weitgehend gleichwerte Funktionalität bietet, wie das browserbasierte Onlinebanking. In dieser App findet die Transaktionsauslösung statt. Die Transaktionsbestätigung erfolgt nun wie bei der 2GA im App-basierten Sicherungsverfahren. Es ist jedoch nicht mehr notwendig, das Verfahren auf einem getrennten Gerät zu betreiben.

Bei einer expliziten 2GA bedeutet das, dass die entsprechende vertragliche Bestimmung entfällt. Selbst implizite 2GA-Verfahren können heute zum Teil auf einem Gerät betrieben werden: statt die Transaktionsdaten in einer Grafik zu kodieren, überträgt die Banking-App die Daten über App-zu-App-Kommunikation direkt an das App-basierte Sicherungsverfahren. Das Fotografieren entfällt dementsprechend.

Mit dem pushTAN-Verfahren präsentierten die Sparkassen bereits 2013 einen 2AA-Piloten [DSZ13]. Das Verfahren wurde aber erst 2015 weitläufig verfügbar [Sch15]. Die Sparkassen betonen, dass die Kanaltrennung auch bei der 2AA in Form zweier logischer Kanäle aufrecht erhalten bleibt. Die Volksbanken und Raiffeisenbanken führten ihr 2AA-Verfahren erst 2016 ein, das dem der Sparkassen jedoch sehr ähnlich ist.

Ein-App-Authentifizierung

App-basierten Sicherungsverfahren, die eine Ein-App-Authentifizierung (1AA) realisieren, lösen nach der Gerätetrennung der 2GA nun auch die App-Trennung der 2AA auf. Bei der 1AA verschmilzt die Banking-App und das App-basierte Sicherungsverfahren zu einer einzigen App, die sowohl für die Transaktionsauslösung als auch für die -bestätigung herangezogen wird (Abbildung 2.2c).

Die Transaktionsbestätigung wird aus Nutzersicht auf einen zusätzlichen Bestätigungsdialog reduziert, der auf die Transaktionsauslösung folgt und nochmals die soeben eingegebenen Transaktionsdaten anzeigt. Eine TAN verwenden die 1AA-Verfahren nicht mehr. Im Hintergrund sind die Abläufe vergleichbar mit der Bestätigung im Rahmen einer 2AA.

Die Postbank bietet bereits seit März 2014 auf iOS-Geräten mit Touch ID die Möglichkeit, Überweisungen im 1AA-Ansatz per Fingerabdruck freizugeben. N26 war Anfang 2015 jedoch die erste Banking-App, die eine 1AA für alle iOS- und Android-Geräte implementierte [Hai15]. Mittlerweile bieten mehrere Banken ähnliche Verfahren, die N26 nachempfunden sind [DS16; God17; ING18].

2.4 Diskussion

Betrachtet man die Geschichte der Sicherungsverfahren, fällt auf, dass die Institute in der jüngeren Vergangenheit die Kontrolle über die Verfahren zunehmend aus der Hand gegeben haben. Während die Banken sich mit dem chipTAN-Verfahren noch angeschickt hatten, einen gemeinsamen, institutsübergreifenden Standard für sichere Transaktionen zu schaffen, rücken seit einiger Zeit Geräte in den Vordergrund, die die Geldhäuser wenig bis gar nicht kontrollieren. Dass diese Entwicklung zu Problemen führen kann, zeigt die Geräteevolution der für das smsTAN-Verfahren genutzten Mobilfunkgeräte.

Vom Spezial- zum Mehrzweckgerät. Als das smsTAN-Verfahren eingeführt wurde, waren die die SMS-empfangenden Geräte noch merklich in ihrer Funktionalität eingeschränkt, weshalb sie heute in der Vorherrschaft der Smartphones als Featurephones bezeichnet werden. Ihre primären Aufgaben waren das Telefonieren und das Senden und Empfangen von SMS. Beide Aufgabenfelder sind auf modernen Smartphones deutlich in den Hintergrund gerückt. Dennoch existiert die gleiche Funktionalität auch heute noch auf den mobilen Endgeräten und wird auch nach

wie vor von einer SIM-Karte abgewickelt. Hierbei hat die sie tragende Hardware eine Veränderung weg von den Spezialgeräten der Featurephones, hin zu den Mehrzweckgeräten moderner Smartphones vollzogen. Zu den wesentlichen Neuerungen von Smartphones zählen die quasi immer verfügbare Internetverbindung sowie ein Ökosystem aus verschiedenen Apps, die eine bestimmte Funktionalität bereitstellen.

Bei den Featurephones sorgte die fehlende Internetfähigkeit unter der Verwendung des smsTAN-Verfahrens noch implizit dafür, dass zwei unterschiedliche Geräte an der Transaktionsdurchführung beteiligt waren. Mit dem Aufkommen von Smartphones wurde diese implizite Gerätetrennung aufgehoben, weil es jetzt theoretisch möglich war, Transaktionen von ein und demselben Gerät durchzuführen. Es ist unmittelbar begreifbar, dass dieser Zustand zu einer Absenkung des Sicherheitsniveaus führte und man sich jetzt sogar wieder einem Bedrohungsszenario gegenübersah, das es mit der Einführung des smsTAN-Verfahrens gerade zu bekämpfen galt: nämlich die Infektion mit Schadsoftware. Im Vergleich zum iTAN-Verfahren hatte sich die Situation sogar noch verschlechtert: Während die reine Infektion des Nutzercomputers nur dazu genutzt werden konnte, Transaktionen transparent zu manipulieren, war durch die Verwendung des smsTAN-Verfahrens auf einem Smartphone sogar das eigenständige Ausführen beliebiger Transaktionen durch entsprechende Schadsoftware zu beliebigen Zeitpunkten denkbar.

Unwirksame DK-Richtlinie. Dieser Umstand ist auch der Deutschen Kreditwirtschaft (DK) nicht verborgen geblieben. Die DK ist ein Interessenverband eines Großteils der öffentlich-rechtlichen, genossenschaftlich und privat organisierten Banken. Auch die DK, damals noch als Zentraler Kreditausschuss bekannt, wollte dieser neuen Entwicklung Rechnung tragen und schrieb schon 2008 für die Verwendung des smsTAN-Verfahrens zwei Geräte vor [ZKA08]. Dies begründete sie damit [DK], dass die Sicherheit von Transaktionen maßgeblich davon abhängt, dass „man sich der Technik der Übertragung über zwei unterschiedliche Kanäle“ bedient. Demnach müsse Mobilebanking mit dem smsTAN-Verfahren „in den Kundenbedingungen für das Online-Banking explizit ausgeschlossen“ werden.

Diese Vorgabe führt dazu, dass den Kunden, die sich mit ihrem Smartphone über den Browser oder die Banking-App des Kreditinstituts anmelden, die Möglichkeit, Überweisungen zu tätigen, ausgeblendet wird, insofern sie das smsTAN-Verfahren bei einer Bank verwenden, die direkt oder indirekt in der DK organisiert ist. Obwohl diese Regelung zunächst sinnvoll und konsequent erscheint, stellt sich durch sie kein Sicherheitsgewinn, sondern nur ein Verlust an Benutzerfreundlichkeit ein. Dem Nutzer wird nämlich nicht untersagt, beispielsweise seinen Kontostand über den

mobilen Browser oder die Banking-App zu überprüfen. Um Zugang zu diesem zu erhalten, muss der Kunde jedoch seine Zugangsdaten eingeben. Sollte das Handy durch eine entsprechende Schadsoftware infiziert sein, kann diese die Zugangsdaten mitschneiden. Der Angreifer unterliegt jedoch nicht den gleichen Restriktionen wie der Nutzer: Oft erkennt der Bankenserver anhand des User-Agents des Browsers oder durch ein bestimmtes Protokoll der Banking-App, dass sich der Nutzer über ein mobiles Endgerät Zugang zu seinem Konto verschafft. Eine Schadsoftware könnte nun ohne Weiteres z. B. den User-Agent seiner HTTP(S)-Anfragen so aussehen lassen, dass der Bankenserver den Nutzer an einem Desktop-Computer vermutet. In Folge wäre die Vorgabe, zwei Geräte zu verwenden, erfüllt und eine Transaktionsauslösung mit Bestätigung über das smsTAN-Verfahren wird möglich. Da sich die Schadsoftware aber in Wirklichkeit auf dem Smartphone befindet, kommt die via SMS zugestellte TAN dennoch auf dem gleichen Endgerät an. Ein Angreifer kann somit beliebige Transaktionen tätigen, wenn es ihm gelingt, das Gerät zu kompromittieren.

SMS-App. Ein genauerer Blick auf das smsTAN-Verfahren offenbart heute große Ähnlichkeiten zu App-basierten Verfahren, besonders zu solchen, die über zwei eigenständige Apps realisiert sind. Dies ist darauf zurückzuführen, dass die Funktionalität, die die SMS darstellt, ebenfalls über eine App erfolgt. Die Unterschiede liegen vor allem bei den fehlenden Garantien bzgl. des Zustellwegs und bei der Tatsache, dass die SMS-empfangende App nicht von der zugehörigen Bank stammt. Obwohl es richtig ist, dass die App-basierten Verfahren den Schutzzielen der Authentizität, Vertraulichkeit und Integrität besser Rechnung tragen, bleibt die für das smsTAN-Verfahren aufgestellte Argumentation der DK, dass Transaktionsauslösung und -bestätigung nicht auf einem Gerät erfolgen dürfen, gültig.

Dass mit dieser Prämisse im Zuge von Mobilebanking und App-basierten Sicherungsverfahren zunehmend gebrochen wird, suggeriert, dass in der Finanzbranche großes, in jedem Fall aber größeres Vertrauen in die Sicherheit von Smartphones herrscht, als das bei stationären Computer und Notebooks der Fall ist. Bis vor Kurzem wäre es noch undenkbar gewesen, alle am Transaktionsprozess beteiligten Authentifizierungselemente über ein und dasselbe Gerät abzubilden, ohne dass dem ein entscheidender Sprung in Sachen Hardware- und Softwaresicherheit vorangegangen wäre. Nichtsdestotrotz sind die hierbei entstandenen und entstehenden Verfahren mittlerweile allesamt so ausgelegt, dass sie auch für das Nutzungsszenario des Mobilebankings geeignet sind. Ruft man sich den vorhergehenden Abschnitt und die Aussage der DK bzgl. der Verwendung des smsTAN-Verfahrens auf nur einem Gerät in Erinnerung, überrascht dieses Vorgehen.

2.5 Fazit

In diesem Kapitel haben wir Forschungsfrage 1 (Einordnung Mobilebanking) adressiert. Zu diesem Zweck haben wir die Begriffe Online- und Mobilebanking trennscharf definiert: Während das Authentifizierungsmedium im Onlinebanking immer ungleich zum transaktionsauslösenden Gerät ist, findet die Auslösung und Bestätigung einer Transaktion beim Mobilebanking auf ein und demselben Gerät statt. Um solche Mobilebanking-Transaktionen zu ermöglichen, benötigt es neue App-basierte Sicherungsverfahren. Die Verfahren lassen sich danach unterscheiden, ob sie nur im Online- oder auch im Mobilebanking verwendet werden können und implementieren entsprechend eine Zwei-Geräte- oder eine Ein- bzw. Zwei-App-Authentifizierung.

Obwohl auch viele der vorangegangenen Verfahren auf Basis von Listen, des Mobilfunks oder sogar dedizierter Geräte nicht ohne Schwächen waren und sind, haben sie die physische Trennung der Transaktionsauslösung von der Transaktionsbestätigung immer als inhärentes Sicherheitsmerkmal verstanden. Dieses Attribut wird für die Einführung des Mobilebankings bewusst aufgegeben.

Dass diese Entscheidung neue Angriffsvektoren für App-basierte Sicherungsverfahren allgemein, aber auch speziell für das Mobilebanking eröffnet, führt das nächste Kapitel aus.

3

Sicherheit beim Mobilebanking

Die Zwei-Faktor-Authentifizierung ist nicht unser Allheilsbringer.

— Bruce Schneier, 2005 [Sch05]

In diesem Kapitel stellen wir verschiedene Angriffe vor, die sich aus der Einführung App-basierter Sicherungsverfahren und des Mobilebankings ergeben. Zu Beginn stellen wir unser Angreifermodell vor, das durch die Sicherheitseigenschaften mobiler Endgeräte motiviert ist. Danach beschreiben wir zwei grundsätzliche Angriffe, die sich unmittelbar durch den Paradigmenwechsel im Mobilebanking ergeben. Neben der allgemeinen Angriffsbeschreibung auf Basis der konzeptionellen Schwächen präsentieren wir jeweils eine Fallstudie, die die Machbarkeit unserer Angriffe unterstreicht. Vor unserem Fazit diskutieren wir Maßnahmen, um die dargestellten Angriffe durch eine hardwarebasierte Gerätebindung und sichere Anzeige zu verhindern.

3.1 Angreifermodell

Ein Smartphone-Nutzer eröffnet ein Konto bei einer Bank und entschließt sich für die Nutzung von Mobilebanking. Zu diesem Zweck hat er das App-basierte Sicherungsverfahren seiner Bank bereits ordnungsgemäß auf seinem Smartphone eingerichtet. Insofern ein 2AA-Verfahren genutzt wird, ist auch die mit dem App-basierten Sicherungsverfahren korrespondierende Banking-App installiert und – falls nötig – ebenfalls eingerichtet. Für den Fall eines 1AA-Verfahrens entfällt dieser

Schritt, da das App-basierte Sicherungsverfahren auch die Funktionalität der Banking-App integriert. Die Einrichtung lief integer ab und der Kunde konnte bereits erfolgreich Transaktionen im Mobilebanking durchführen.

Zu einem späteren Zeitpunkt wird für das vom Kunden eingesetzte Smartphone eine Sicherheitslücke bekannt, die der Gerätehersteller durch eine Sicherheitsaktualisierung noch nicht behoben hat. Die Schwachstelle kann von einem unberechtigten Dritten ausgenutzt werden, um den Programmtext beliebiger Apps auf dem Smartphone zu verändern. Dadurch wird der Angreifer in die Lage versetzt, Programmabläufe von Apps zu manipulieren und zugehörige Nutzerdaten auszuleiten. Das schließt explizit die am Mobilebanking beteiligten Apps ein.

Obwohl eine Vielzahl von Infektionswegen denkbar ist, gehen wir im Rahmen dieses Angreifermodells davon aus, dass der Nutzer eine bösartige App aus den offiziellen Quellen bezieht und ausführt. Diese App nutzt die beschriebene Sicherheitslücke aus, ohne dass der Nutzer davon Kenntnis nimmt. Der Angreifer hat keinen physischen Zugriff auf das Gerät.

3.2 Sicherheit mobiler Endgeräte

In diesem Abschnitt motivieren wir das dargestellte Angreifermodell. Zu diesem Zweck zeigen wir, dass für die beiden vorherrschenden mobilen Systeme Android und iOS regelmäßig schwerwiegende Sicherheitslücken bekannt werden, die für eine Rechteausweitung im Sinne unseres Angreifermodells genutzt werden können.

Sandboxing. Unter iOS und Android sind die installierten Apps durch das sog. Sandboxing voneinander abgeschottet [Wan+14; BD19]. Jede App besitzt einen exklusiven Datenbereich, der für die App privat ist und von anderen Apps auf dem mobilen Endgerät nicht gelesen werden kann. Ferner operiert jede App in einem restriktiven Rechtekontext; mit anderen Apps kann nur über feste Programmierschnittstellen kommuniziert werden. Darüber hinaus benötigt der Zugriff auf bestimmte Ressourcen die explizite Genehmigung des Nutzers.

Sicherheitslücken. Das Sandboxing und Rechtesystem zeigt, dass Sicherheit bei Android und iOS bereits bei der Konzeption Berücksichtigung gefunden hat. Dennoch sind beide Systeme hochkomplexe Softwareprojekte. Demnach überrascht es nicht, dass für sie periodisch Sicherheitslücken bekannt werden. Eine der schwerwiegendsten Klassen an Verwundbarkeiten ist definitiv die Rechteausweitung. Solche

Plattform	2018									2019			Σ
	4	5	6	7	8	9	10	11	12	1	2	3	
Android	5	16	7	5	10	7	7	8	11	5	20	8	109
iOS	1	6	–	2	–	12	4	–	5	7	2	10	49

Tabelle 3.1: Anzahl der in den Monaten zwischen April 2018 und März 2019 entdeckten Schwachstellen, die unter Android bzw. iOS zu einer Rechteausweitung genutzt werden konnten.

Sicherheitslücken erlauben es, Befehle in einer höheren Privilegienebene auszuführen, als für den Kontext, in dem sie ausgenutzt wird, eigentlich vorgesehen ist. In Konsequenz könnte ein Angreifer z. B. aus der Sandbox ausbrechen oder gar Anweisungen mit Kernel-Rechten ausführen.

Dass solche Sicherheitslücken eher die Regel als die Ausnahme sind, zeigt Tabelle 3.1. In der Tabelle ist die Anzahl der Schwachstellen vermerkt, die im vergangenen Jahr bis zum März 2019 bekannt geworden sind und die eine Rechteausweitung ermöglichten. Für Android wurde das monatlich erscheinende Security Bulletin herangezogen, das behobene Sicherheitslücken dokumentiert [AOSP]. Seit Mitte 2017 ist darin auch der Typ der Schwachstelle verzeichnet, wobei „EoP“ für „Elevation of Privilege“, also Rechteausweitung steht. Gezählt wurden nur die EoP-Einträge, die nicht auf Treiber z. B. von Qualcomm oder NVIDIA zurückzuführen sind. Bei iOS erscheint lediglich mit der Veröffentlichung einer neuen iOS-Version eine Dokumentation der adressierten Sicherheitsprobleme [APC]. Dadurch sind in Tabelle 3.1 die Monate ausgespart, in denen keine neue iOS-Version erschienen ist. Gab es mehrere pro Monat, wurde die Anzahl der Sicherheitslücken, die eine Rechteausweitung zulassen, summiert.

Durch die deutlich höhere Anzahl an dokumentierten Schwachstellen bei Android als bei iOS sollte nicht darauf geschlossen werden, dass iOS sicherer implementiert ist als Android. Zum einen ist Android ein offenes System, bei dem der Quelltext verfügbar ist. Dadurch ist das Suchen von Sicherheitsproblemen weniger komplex. Apples iOS ist hingegen komplett geschlossen, wodurch die Schwachstellensuche zeitintensiver ist. Zum anderen dokumentiert die Tabelle nur die Schwachstellen, die eine Rechteausweitung zulassen, nicht aber, wie schwerwiegend die Lücken im Einzelnen sind. Die Kernaussage ist aber, dass sich die Hersteller beider Systeme immer wieder mit schweren Sicherheitslücken konfrontiert sehen.

Sicherheitsaktualisierungen. Entscheidender für die Sicherheit des Systems ist, wie schnell die Sicherheitsprobleme ausgeräumt und den Nutzern in Form von Softwareaktualisierungen zur Verfügung gestellt werden. An dieser Stelle gibt es deutliche Unterschiede zwischen Android und iOS. Zwar ist es richtig, dass Google Sicherheitslücken schnell schließt. Für das Verteilen von Aktualisierungen sind aber die einzelnen Hersteller verantwortlich [Gas+17; MN18; FLG17]. Die kommen ihrer Verantwortung zum Ausrollen von Sicherheitsaktualisierungen jedoch zwischen vorbildlich bis gar nicht nach. Dementsprechend bleiben viele Geräte für bekannte Sicherheitslücken lange verwundbar. Hierfür ursächlich ist auch, dass viele Hersteller noch eigens Modifikationen an Android vornehmen, um sich von der Konkurrenz abzuheben. Dadurch müssen Sicherheitsaktualisierungen seitens Google unter Umständen angepasst werden. Ein weiteres Problem ist, dass die Hersteller – ohne ihre Kunden in irgendeiner Form hiervon in Kenntnis zu setzen – schon oft nach ein bis zwei Jahren Aktualisierungen für ein Gerät komplett einstellen [TBR15; Fel+11]. Durch das von Android ins Leben gerufene *Project Treble* soll das Aktualisieren von hardwarespezifischen Komponenten herstellerübergreifend einfacher und damit schneller möglich werden. Die Änderungen der Hersteller an Android selbst sind davon jedoch nicht erfasst [MN18].

Die Situation gestaltet sich unter iOS fundamental anders. Apple ist nicht nur allein für die Entwicklung von iOS zuständig, sondern auch für alle Geräte, die mit dem Betriebssystem ausgeliefert werden. Zudem pflegt Apple Geräte deutlich länger als die Android-Konkurrenz. So erhält das 2013 mit iOS 7 erschiene iPhone 5S auch über fünf Jahre später noch alle Aktualisierungen im Rahmen von iOS 12. Wie Tabelle 3.1 zu entnehmen ist, veröffentlicht Apple kontinuierlich alle ein bis zwei Monate eine neue iOS-Version.

Schadsoftware. Unser Angreifermodell nimmt an, dass eine bösartige App auf das Smartphone gelangt. Dabei stellt sich die Frage, was Google und Apple dagegen unternehmen, dass Schadsoftware ihren Weg in den Play bzw. App Store findet. Auch hier unterscheiden sich die Ansätze. Während Apps bei Apple neben einer automatischen Analyse auch durch einen Mitarbeiter überprüft und kritisch hinterfragt werden, erfolgt die Begutachtung einer eingereichten App bei Google vollautomatisch [Che+16; Che+15].

Oberheide zeigte bereits 2010, dass der Google Play Store keinen ausreichenden Schutz vor Schadsoftware bietet [Obe10]. Er veröffentlichte eine scheinbar gutartige Android-App im Google Play Store, die beim Start eine Sicherheitslücke ausnutze, um sich privilegierten Zugriff auf das Gerät zu verschaffen. Im gleichen Jahr

unterstrichen auch Davi u. a. die Praxistauglichkeit einer Rechteausweitung auf Android [Dav+10]. Vier Jahre später stellten Maier, Müller und Protsenko zwar fest, dass sich die Prüfroutinen seitens Google beim Veröffentlichen von Apps im offiziellen Play Store verbessert haben [MMP14]. Sie ließen sich jedoch nach wie vor trivial umgehen: Innerhalb ihrer bösartigen App war ein gezippter Exploit enthalten, der erst beim Starten der App entpackt wurde. Diese Maßnahme reichte bereits, um Googles automatisierte Erkennung zu umgehen und die schadhafte App über den Play Store anzubieten. Laut Forschung von Poeplau u. a. ist es auch ohne Weiteres möglich, schadhafte Programmteile gar nicht erst mit der App auszuliefern, sondern den Schadcode nach Installation nachzuladen [Poe+14].

Für Android wirkt es sich außerdem nachteilig aus, dass Apps auch aus Drittquellen installiert werden können [MP15]. Dadurch existieren unter Android verschiedene inoffizielle App Stores, die nicht Googles Prüfroutinen durchlaufen. Darüber hinaus können Apps auch einzeln installiert werden, was in Bezug auf Phishing-Angriffe relevant ist. Um die Aktivität von Apps auch auf dem Gerät auf schadhaftes Verhalten zu prüfen, hat Google 2017 *Play Protect* eingeführt [KD18]. Die Effektivität von Play Protect ist jedoch noch Forschungsgegenstand. Unter iOS ist der App Store die einzige Bezugsquelle, der zudem wie beschrieben einen rigorosen Prüfprozess für die Aufnahme von Apps vorsieht.

Obwohl Apple durch strenge App-Begutachtungen und schnelle Aktualisierungen gegenüber Schadsoftware besser gefeit ist, werden auch unter iOS Geräte gegenüber Sicherheitslücken exponiert [Wan+13]. Nachdem bspw. iOS 12.1.4 erschienen ist, wurde bekannt, dass die damit behobenen kritischen Sicherheitslücken zur Rechteausweitung auch in freier Wildbahn ausgenutzt wurden [Cim]. Dennoch ergibt sich ingesamt das Bild, dass die Gefahrenlage unter Android ungleich höher ist, als unter iOS. Dafür verantwortlich ist neben der Fragmentierung und der zaghaften Update-Politik nicht zuletzt der hohe Marktanteil von Android, der ihn besonders attraktiv für Schadsoftware macht: Android hat global gesehen einen Marktanteil von 85% [Spe19].

Jailbreaking/Rooting. Unter iOS und Android verschafft sich ein kleiner Nutzerkreis bewusst erweiterten Zugriff auf das System [KK14a; CV18]. Dieser Vorgang wird unter iOS Jailbreaking und unter Android Rooting genannt. Jailbreaking/Rooting wird im Allgemeinen von erfahrenen Nutzern betrieben und erlaubt unter Umgehung des Sandboxing die Ausführung beliebigen Codes als Systemadministrator (Root-Benutzer). Dementsprechend können auf einem solchen Gerät Modifikationen des Systems oder auch anderer Apps betrieben werden.

Je nach Gerät und Hersteller ist es unter Android explizit möglich, das Betriebssystem zu modifizieren. Dadurch ist das Rooting auf diesen Geräten leicht zu bewerkstelligen [SCB15]. Apple sieht so eine Möglichkeit jedoch genauso wie einige Android-Hersteller nicht vor [KK14b]. Demnach benötigt es zum Jailbreaking bzw. Rooting zwangsläufig eine – in der Regel aber mehrere – Sicherheitslücken, mit denen Schutzmaßnahmen wie das Sandboxing deaktiviert und eigener Code platziert werden kann. Ein erfolgreicher Jailbreak bzw. Root-Exploit ist deshalb auch immer der Nachweis, dass die oben beschriebenen Sicherheitslücken erfolgreich ausgenutzt werden können, um die Systeme vollständig zu kompromittieren.

Gerade für Banking-Apps und App-basierte Sicherungsverfahren stellt Jailbreaking/Rooting eine Herausforderung dar, da dadurch die Sicherheitsgarantien des Betriebssystems aufgelöst werden. Deshalb ist es gängige Praxis, dass solche Apps diese bewusste Rechteausweitung erkennen [EBS15; KK14a]. Die Reaktion auf einen erkannten Root-Zugriff fällt je nach App unterschiedlich aus: Manche informieren nur das Backend, andere unterrichten den Nutzer bzgl. der Risiken oder verbieten ihm gar das Verwenden der App. Die Erkennung der Apps basiert im Regelfall auf der Präsenz von charakteristischen Dateisystemartefakten: Unter iOS wird mit einem Jailbreak z. B. standardmäßig der Cydia App Store installiert, während unter Android das SU-Programm platziert wird. In der Praxis führt der Ausschluss von Nutzern, die bewusst Rooting betreiben bzw. einen Jailbreak anwenden, dazu, dass die Jailbreaking/Rooting-Gemeinde wiederum Ansätze entwickelt, um der Erkennung zu entgehen. Das Ergebnis ist ein Katz-und-Maus-Spiel zwischen besseren Erkennungsroutinen auf der einen und besseren Umgehungsmethoden auf der anderen Seite [KK14a].

Obwohl die Detektierungsbemühungen der App-Hersteller nachvollziehbar sind, greifen sie für das Problem zu kurz. Es ist ungleich schwerer festzustellen, ob ein Gerät grundsätzlich verwundbar ist oder ob eine Sicherheitslücke auf dem Gerät ohne die bewusste Entscheidung des Nutzers bereits ausgenutzt wurde [Gas+17]. Genau so eine Erkennung wäre jedoch notwendig, um die Integrität der Ausführungsumgebung der App sicherzustellen. Dabei gilt es zu bedenken, dass es nicht im Interesse des Angreifers liegt, etwa den Cydia Store bzw. das SU-Programm zu installieren. Stattdessen ist es wahrscheinlicher, dass ein Angreifer durch das Ausnutzen einer Sicherheitslücke Daten stiehlt oder sich auf sonstige Art und Weise atypisch persistiert, sodass die üblichen Erkennungsroutinen ins Leere laufen. In Konsequenz können Apps im besten Fall erkennen, ob einer Nutzer willentlich Jailbreaking oder Rooting betrieben hat. Hat dies nicht stattgefunden, kann umgekehrt nicht auf ein sicheres und integeres System geschlossen werden.

3.3 Grundsätzliche Angriffe

Im Nachfolgenden beschreiben wir zwei Angriffe, die sich auf das in Abschnitt 3.1 dargestellte Angreifermodell beziehen. Zuerst stellen wir einen Replikationsangriff vor, der das App-basierte Sicherungsverfahren auf ein unautorisiertes Gerät kopiert. Der zweite Angriff bezieht sich auf die Echtzeitmanipulation einer kundeninitiierten Transaktion.

3.3.1 Replikation des Sicherungsverfahrens

Ziel dieses Angriffs ist es, dass ein Angreifer das App-basierte Sicherungsverfahren seines Opfers auf sein eigenes Gerät kopieren kann. Das Verfahren kann durch den Angreifer anschließend derart verwendet werden, dass er ausgelöste Transaktionen im Namen seines Opfers unabhängig bestätigen kann.

Dadurch soll gezeigt werden, dass App-basierte Sicherungsverfahren im Vergleich zu vorhergehenden und weit verbreiteten Verfahren wie sms- oder chipTAN ein relativer Rückschritt sind. Beide der genannten klassischen Sicherungsverfahren greifen auf ein Hardwareelement zurück, das sich unter praktischen Gesichtspunkten nicht kopiert werden kann.

Personalisierung

Alle App-basierten Sicherungsverfahren, ob 2GA, 2AA oder 1AA, sind in Software realisiert. Unter Android ist die zugrundeliegende Schnittstelle zum Betriebssystem in Java, unter iOS in Objective-C und Swift implementiert. Eben diese bieten die Basis, auf der alle Android- und iOS-Apps entwickelt sind. Die Apps werden zur Veröffentlichung dann jeweils in den Google Play Store bzw. Apple App Store geladen, wo sie der Allgemeinheit zur Verfügung stehen.

Wenn ein Anwender die App über den offiziellen Store bezieht, ist diese zunächst für einen jeden gleich. Eine Personalisierung der App erfolgt erst im Rahmen der Inbetriebnahme unter Zutun des Nutzers. Während es bei den reinen Banking-Apps für einen lesenden Zugriff auf das Konto genügt, sich mit Benutzername/Passwort einzuloggen, ist dieses Vorgehen für das App-basierte Sicherungsverfahren nicht ausreichend. Stattdessen muss die App durch eine Registrierung erst individualisiert und aktiviert werden. Obwohl die Schritte von Bank zu Bank variieren, ist es gerade

bei Neukunden üblich, dass die Einrichtung des App-basierten Sicherungsverfahrens einen separaten Registrierungsbrief benötigt. Der Brief trägt ein Geheimnis, das entweder direkt als kryptographischer Schlüssel fungiert, oder aber das Erzeugen und Setzen eines eigenen Schlüssels gegenüber der Bank erlaubt. Nach Abschluss der Registrierung ist das App-basierte Verfahren einsatzbereit und kann Transaktionen bestätigen.

Das Schlüsselmaterial des App-basierten Sicherungsverfahrens kann auf verschiedene Art und Weise an der Transaktionsbestätigung beteiligt werden. Es ist z. B. möglich, dass die Bank im Registrierungsprozess einen gemeinsamen Schlüssel mit der App austauscht. Dementsprechend könnte das Verfahren bspw. eine verschlüsselte TAN von der Bank empfangen, diese entschlüsseln und nach Verifikation der Transaktionsdaten durch den Kunden in Klartext an die Bank zurücksenden. Analog funktionieren auch Verfahren mit asymmetrischer Kryptographie. Es wäre aber auch denkbar, dass die App lediglich die Transaktionsdaten empfängt, dem Benutzer zur Verifikation anzeigt und zusammen mit einem Zeitstempel signiert zurücksendet.

Replikation

In jedem Fall muss jedoch Schlüsselmaterial auf dem Gerät gespeichert werden, das gleichzeitig die Personalisierung der App darstellt. Damit ein privilegierter Angreifer das Sicherungsverfahren auf sein eigenes Gerät replizieren kann, muss er diese Daten kopieren. Ein pragmatischer Ansatz ist es, schlicht das komplette private Datenverzeichnis der App zu kopieren. Abhängig von der konkreten Implementierung genügt es bereits, wenn der Angreifer die App in gleicher Version installiert (oder ebenfalls vom Opfergerät repliziert) und das persönliche Datenverzeichnis der App auf dem eigenen Gerät einspielt. Die notwendigen Schritte sind damit ähnlich zum Anfertigen und Wiederherstellen einer Sicherungskopie.

In der Praxis verhindern zwei Maßnahmen, dass das geschilderte Vorgehen auf Angreiferseite bereits zu einem funktionsfähigen Sicherungsverfahren führt: der Zugangsschutz und der Gerätefingerabdruck.

Zugangsschutz. Der Zugangsschutz wurde bereits in Abschnitt 2.3 angesprochen: Zum Teil verlangen die App-basierten Verfahren nach dem Start zunächst ein Passwort, das im Rahmen der Einrichtung festgelegt wurde. Dieses Passwort dient auch dazu, das in der App abgelegte Schlüsselmaterial wiederum selbst zu verschlüsseln.

Demnach kann es notwendig werden, dass der Angreifer nicht nur die App-Daten kopieren, sondern auch das Passwort für die App kennen muss.

Um an das Passwort zu gelangen, kann der Angreifer aktiv oder passiv vorgehen. Entweder er versucht den Nutzer – bspw. durch Phishing – aktiv dazu aufzufordern, ihm das Passwort für die App mitzuteilen, oder schneidet es passiv mit, sobald der Nutzer aus freien Stücken das App-basierte Sicherungsverfahren verwendet. Abhängig von der Komplexität und Ableitungsfunktion des Passworts ist auch ein Brute-Force-Angriff ein Vorgehen, das zur Offenlegung des Passworts führen kann.

Gerätefingerabdruck. Neben dem Schlüsselmaterial wird bei der Personalisierung der App auch ein sog. Gerätefingerabdruck erzeugt und abgelegt. Damit ist nicht der biometrische Fingerabdruck des Geräteinhabers gemeint. Stattdessen werden bei der Einrichtung des Verfahren verschiedene Umgebungswerte ausgelesen, die für das Gerät charakteristisch sind, und dann gespeichert. In regelmäßigen Abständen, z. B. beim Start der App, werden dieselben Umgebungswerte erneut gelesen, um sie dann mit den gespeicherten abzugleichen. Stellt die App eine Abweichung fest, reagiert sie mit dem Löschen des Schlüsselmaterials.

Um es zu vermeiden, in diesen Zustand zu laufen, muss der Angreifer dafür sorgen, dass die App nach wie vor die gleichen Werte aus seiner Ausführungsumgebung liest. Ein zielgerichteter und zuverlässiger Ansatz ist es, die durch die App gelesenen Umgebungsvariablen per statischer oder dynamischer Analyse zu ermitteln. Dieser Satz an Werten muss dann zusätzlich zur Kopie der personalisierten App-Daten an den Angreifer transferiert werden. Der muss nun dafür Sorge tragen, dass die App diese Werte bei jeder Ausführung statt der geräteeigenen liest. Hierfür könnte Hooking eingesetzt werden; alternativ kann die App auch so modifiziert werden, dass die Routinen zur Bildung des Geräteabdrucks die Werte als Konstanten lesen.

Ablauf. Die Replikation eines App-basierten Sicherungsverfahrens ist schematisch in Abbildung 3.1 dargestellt. Es umfasst nach erfolgreicher Einrichtung des Verfahrens durch den Nutzer die folgenden Schritte durch den Angreifer:

1) Kopie aller Daten, die die App im Personalisierungsprozess erzeugt und abgelegt hat.

2) Auslesen aller Umgebungswerte, die die App zur Bildung eines Gerätefingerabdrucks heranzieht.

3) Falls notwendig: Platzierung von Schadcode, der die Eingabe des für den Zugang zur App notwendigen Passworts mitschneidet.

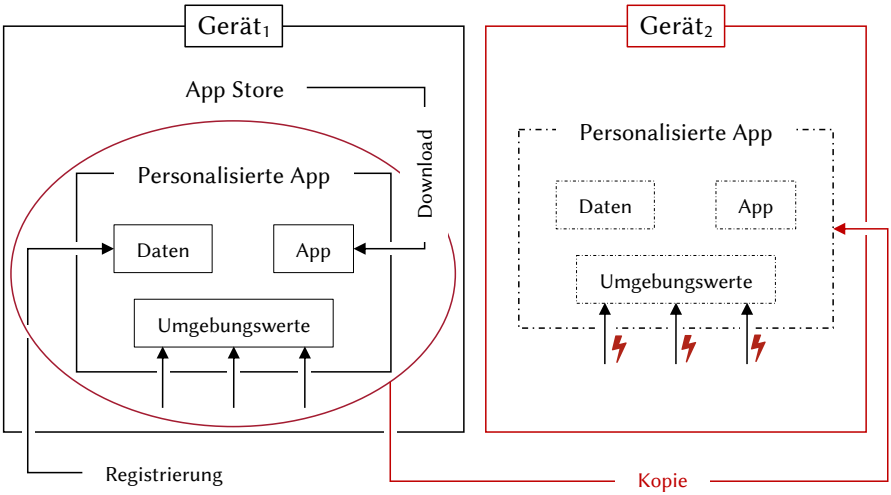


Abbildung 3.1: Schematische Darstellung des Replikationsangriffs. Die App wird inklusive der personalisierten Daten und etwaiger Umgebungsvariablen auf ein zweites Gerät kopiert.

Fallstudie photoTAN

Bereits 2014 präsentierten Dmitrienko u. a. Replikationsangriffe gegen verschiedene Apps zur 2FA [Dmi+14]. Sie bemühten ein mit unserem vergleichbares Angreifermodell. Neben populären Apps zur Benutzerauthentifizierung, wie dem Google Authenticator, führten sie ihren Replikationsangriff auch gegen eine Demo-Version der CrontoSign-App (v5.0.3) auf Android durch. Das Verfahren wird heute produktiv von mehreren Privatbanken unter dem Namen photoTAN als App-basiertes 2AA-Sicherungsverfahren eingesetzt. In ihrer Forschung gelang es Dmitrienko u. a., die CrontoSign-App durch eine triviale Eins-zu-Eins-Kopie auf ein anderes Gerät zu replizieren. Die Bildung und Verifizierung eines Gerätefingerabdrucks erwähnen sie nicht. Wir zeigen, dass sich das Verfahren auch in der Produktivversion nicht gegen einen Replikationsangriff schützen kann. Zu diesem Zweck führen wir unseren Angriff ebenfalls auf Android gegen die photoTAN-App von vier bekannten Privatbanken durch: Deutsche Bank (v2.1.7), Commerzbank (v7.1.7), Norisbank (v2.1.7) und Comdirect (v6.0.6).

Funktionsweise. Beim photoTAN-Verfahren stellt die Bank nach Transaktionsauslösung einen farbigen Matrixcode dar, der dann mit der photoTAN-App unter Zuhilfenahme der Smartphone-Kamera gelesen wird. Daraufhin zeigt die App dem Kunden die Transaktionsdaten zur Verifikation sowie die TAN zur Bestätigung an. Das photoTAN-Verfahren ist ein Produkt von One Span, das außerhalb des deutschsprachigen Raums als CrontoSign bekannt ist. Das Verfahren wird als White-Label-Produkt vertrieben, lässt hinsichtlich seines Verhaltens aber Konfigurationsmöglichkeiten durch die einsetzende Bank zu.

Registrierung. Für die Einrichtung des photoTAN-Verfahrens ist der Kunde auf einen Registrierungsbrief angewiesen, auf dem ein Matrixcode abgedruckt ist. Diese Grafik behält auch nach der Einrichtung eines Geräts seine Gültigkeit und kann dazu genutzt werden, mehrere Geräte gleichzeitig zur Transaktionsfreigabe zu verwenden. Die Registrierungsdaten werden in einer einzigen Datei im privaten Datenbereich der App abgelegt. Da keine der vier Banken die App zusätzlich durch ein Passwort sichert, ist die Datei unverschlüsselt.

Gerätefingerabdruck. Die Apps aller vier Banken erstellen einen Gerätefingerabdruck, der im Fall der Comdirect nur aus der `ANDROID_ID` und bei den restlichen Instituten noch zusätzlich aus der `IMEI` besteht. Die gelesenen Werte werden wieder in besagter Registrierungsdatei gespeichert und es ist notwendig, dass die App wiederkehrend eben diese Werte liest, damit das Verfahren funktioniert.

Wir haben durch statische Analyse ermittelt, dass die App die beiden Werte liest. Der Aufwand hielt sich jedoch in Grenzen, da es sich sowohl bei der `ANDROID_ID` als auch der `IMEI` um naheliegende Werte zur Geräteidentifikation handelt. Darüber hinaus sind sie für jedes Gerät einzigartig und zudem sehr stabil. Insbesondere die `IMEI` ändert sich nicht, die `ANDROID_ID` nur, wenn das Gerät zurückgesetzt wird.

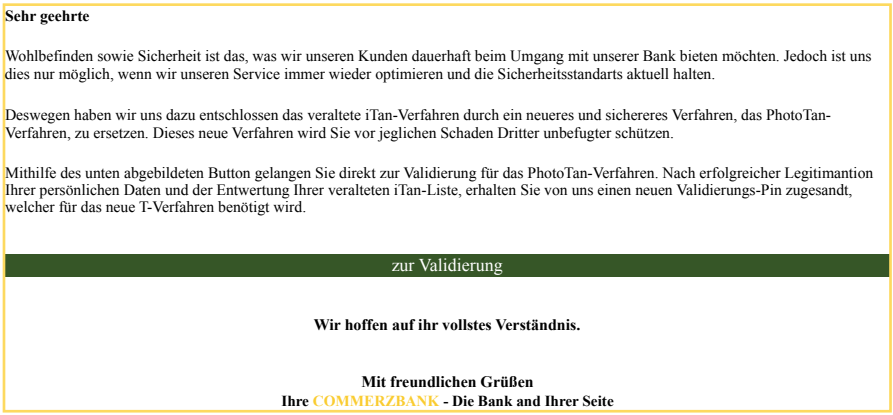
Repackaging. Damit die photoTAN-App auch auf unserem Angreifergerät immer die erwartete `ANDROID_ID` und ggfs. `IMEI` liest, können wir entweder unsere eigene Umgebung, oder aber die Apps statisch modifizieren. Letzteres Vorgehen wird Repackaging genannt und ist deshalb attraktiv, weil dadurch keine Anpassungen am System durchgeführt werden müssen. Der Angreifer könnte also ein reguläres Smartphone ohne Root-Zugriff verwenden und es lassen sich passende Versionen der App pro Opfer erstellen. Das Vorgehen war für die photoTAN-Apps der Deutschen Bank und Norisbank problemlos möglich: Die üblichen Aufrufe an die Klassen `Secure` (`ANDROID_ID`) und `TelephonyManager` (`IMEI`) wurden durch das Auslesen einer Konstante ersetzt. Die Transformationen wurden mithilfe von Dexlib2 [Gru] realisiert.

Eine andere Situation ergab sich für die photoTAN-Apps der Commerzbank und der Comdirect: beide Apps schützen sich durch Integritätsmaßnahmen vor Veränderungen. Jede Android-App ist zwangsläufig mit einem Zertifikat signiert, das jedoch nicht von einer vertrauenswürdigen Entität stammen muss. Im Fall der Commerzbank und der Comdirect nutzen die photoTAN-Apps diese Signatur, um Repackaging zu erkennen und unterbinden die Nutzung der App. Die Ansätze sind hierbei jedoch unterschiedlich.

Die Comdirect liefert die erwartete Signatur nicht als konstante mit der App aus, sondern bettet sie in die photoTAN-Grafik ein, die mit der App gescannt werden muss. Wenn die zur Laufzeit ermittelte App-Signatur nicht mit der enkodierten übereinstimmt, beendet sich die App sofort. Um diese Überprüfung zu umgehen, kann ähnlich vorgegangen werden, wie beim Anpassen der Werte für den Gerätefingerabdruck: Eine Transformation an den entsprechenden Aufruf der Klasse `PackageManager` wird durch die Rückgabe einer Konstanten, die der originalen Signatur entspricht, ersetzt. Die Commerzbank schützt ihre App mit einem kommerziellen Härtungsprodukt der Firma P, das die Signatur der App manuell statt durch einen Android-API-Aufruf liest. Die Routine ist in einer nativen Bibliothek implementiert und obfuskiert. Um die Routine zu umgehen, genügte es jedoch, Aufrufe der Libc-Funktion `open` mithilfe von `LD_PRELOAD` abzufangen und einen Dateideskriptor auf die originale statt unsere modifizerte App zurückzugeben. Andere Möglichkeiten den Schutz durch P zu deaktivieren, werden in Abschnitt 4.2 ausführlich dargestellt.

Das dargestellte Vorgehen führt dazu, dass die photoTAN-Apps von allen vier Banken von einem auf ein anderes Gerät kopiert werden können. Dort funktionieren sie regulär und können für die Bestätigung von Transaktionen eingesetzt werden. Unter dem folgenden Weblink findet sich eine exemplarische Videodemonstration des Angriffs für die Commerzbank: `https://www.cs1.tf.fau.de/appAuth`.

Alternativer Angriff. Dadurch, dass die Aktivierungsgrafik des photoTAN-Verfahrens auf dem Registrierungsbrief unbegrenzte Gültigkeit für beliebige Geräteaktivierungen besitzt, ergibt sich ein alternativer Angriff: Statt die aktivierte und funktionsfähige App-Instanz eines photoTAN-Kundens zu kopieren, könnte ein Angreifer es auch anstreben, die Aktivierungsgrafik zu erlangen. Hierfür sind wiederum verschiedene Szenarien denkbar. Ein Angreifer könnte z. B. alle Daten der App, die er sonst repliziert hätte, löschen. Infolgedessen ist der Kunde gezwungen, das Verfahren mithilfe des Registrierungsbriefs neu einzurichten. Der Schnittstellen-Aufruf an die Kamera des Smartphones, die der photoTAN-App das Bild liefert, könnte dann mitgeschnitten und an den Angreifer übertragen werden.



Sehr geehrte

Wohlbefinden sowie Sicherheit ist das, was wir unseren Kunden dauerhaft beim Umgang mit unserer Bank bieten möchten. Jedoch ist uns dies nur möglich, wenn wir unseren Service immer wieder optimieren und die Sicherheitsstandarts aktuell halten.

Deswegen haben wir uns dazu entschlossen das veraltete iTan-Verfahren durch ein neueres und sichereres Verfahren, das PhotoTan-Verfahren, zu ersetzen. Dieses neue Verfahren wird Sie vor jeglichen Schaden Dritter unbefugter schützen.

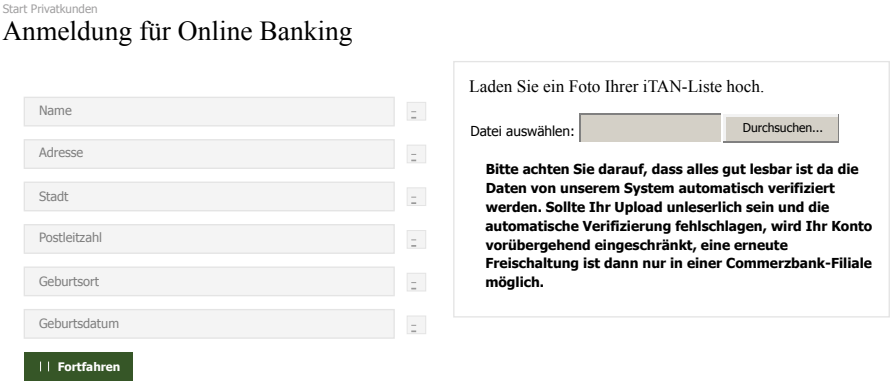
Mithilfe des unten abgebildeten Button gelangen Sie direkt zur Validierung für das PhotoTan-Verfahren. Nach erfolgreicher Legitimantion Ihrer persönlichen Daten und der Entwertung Ihrer veralteten iTan-Liste, erhalten Sie von uns einen neuen Validierungs-Pin zugesandt, welcher für das neue T-Verfahren benötigt wird.

zur Validierung

Wir hoffen auf ihr vollstes Verständnis.

Mit freundlichen Grüßen
Ihre COMMERZBANK - Die Bank and Ihrer Seite

(a) Phishing-E-Mail mit Aufforderung zum Login.



Start Privatkunden

Anmeldung für Online Banking

Name

Adresse

Stadt

Postleitzahl

Geburtsort

Geburtsdatum

Fortfahren

Laden Sie ein Foto Ihrer iTAN-Liste hoch.

Datei auswählen: Durchsuchen...

Bitte achten Sie darauf, dass alles gut lesbar ist da die Daten von unserem System automatisch verifiziert werden. Sollte Ihr Upload unleserlich sein und die automatische Verifizierung fehlschlagen, wird Ihr Konto vorübergehend eingeschränkt, eine erneute Freischaltung ist dann nur in einer Commerzbank-Filiale möglich.

(b) Formular zum Upload der iTAN-Liste.

Abbildung 3.2: Phishing für Commerzbank-Kunden (adaptiert von [CoBa19]).

Noch schädlicher wäre ein Phishing-Angriff, der nicht nur die Aktivierungsgrafik des photoTAN-Verfahrens mitschneidet, sondern auch die Zugangsdaten zum Banking-Portal. Dadurch ergäbe sich ein vollwertiger Zugriff auf das Bankkonto und es könnten beliebige Transaktionen getätigt werden.

Um ein solches Angriffsszenario zu konstruieren, kann ein bereits existierender Angriff, vor dem die Commerzbank im März 2019 warnte, trivial adaptiert werden [CoBa19]. Bedingt durch die regulatorischen Vorgaben der Zahlungsdiensterichtlinie II (PSD2) (näheres hierzu in Kapitel 6) migrieren die Privatbanken ihre Kunden von der iTAN zu anderen Sicherungsverfahren [Sei19]. Diesen Moment nutzen auch Kriminelle, um ihre Phishing-E-Mails plausibel erscheinen zu lassen. Darin fordern sie ihre Opfer auf, einem in der Nachricht genannten Link zu folgen, um eine dringende Aktion in ihrem Onlinebanking zu tätigen (siehe Abbildung 3.2a). Der Link führt jedoch nicht zum authentischen Internetauftritt der Bank, sondern zu einer – oft täuschend echten – Nachbaute. Gibt ein Besucher auf dieser Seite seine Zugangsdaten zum Onlinebanking ein, werden diese durch den Angreifer mitgeschnitten. Nach vermeintlichem Login fordert die Phishing-Seite ferner dazu auf, ein Foto der iTAN-Liste hochzuladen (Abbildung 3.2b). Wird dieser Anweisung Folge geleistet, besitzt der Angreifer auch Zugriff auf das Sicherungsverfahren und kann schlussendlich beliebige Transaktionen tätigen.

Durch die unbegrenzte Gültigkeit des Aktivierungsbriefs ergibt sich ein analoger Angriff für das photoTAN-Verfahren: Kriminelle versenden eine Phishing-E-Mail, die wie oben ausgeführt zum Login über einen genannten Link auffordert. Statt nach Eingabe von Benutzername/Passwort das Hochladen eines Fotos der iTAN-Liste zu erfragen, könnte die Webseite nun um ein Foto der Aktivierungsgrafik für das photoTAN-Verfahren bitten. Die Auswirkungen sind letztendlich dieselben: Mit den Zugangsdaten und der Aktivierungsgrafik kann der Angreifer ein beliebiges Gerät zur Nutzung des photoTAN-Verfahrens registrieren und im Anschluss beliebige Transaktionen durchführen.

Andere Sicherungsverfahren beugen diesem Angriff dadurch vor, dass jede Geräteaktivierung einen eigenständigen Registrierungsbrief benötigt. Bei Legitimierungsverfahren mit dedizierter Hardware, bei denen das Authentifizierungselement aus einer personalisierten Smartcard besteht, ist der Angriff grundsätzlich nicht möglich, da das Geheimnis unter praktischen Gesichtspunkten nicht extrahiert werden kann.

3.3.2 Echtzeitmanipulation von Transaktionen

Im Unterschied zum letzten Abschnitt geht es uns im Folgenden nicht darum, Authentifizierungselemente zu kopieren, sondern eine nutzerinitiierte Transaktion derart in Echtzeit zu manipulieren, dass der Empfänger und Betrag geändert werden, ohne dass dies dem Opfer auffallen könnte.

App-basierte Sicherungsverfahren werden nicht nur auf einem Mehrzweckgerät realisiert, sondern im Rahmen des Mobilebankings auch auf demselben Gerät betrieben, dass auch die Zugangsdaten entgegennimmt. Dadurch ergibt sich eine konzeptionelle Schwäche, die keine sichere Anzeige erlaubt. Infolgedessen ist es möglich, Transaktionen für den Nutzer transparent zu manipulieren.

Es ist zweitrangig, ob ein 2AA- oder 1AA-Verfahren für das Mobilebanking eingesetzt wird. Im Falle eines 2AA-Verfahrens müssen jedoch zwei statt nur einer App manipuliert werden; dementsprechend ist der 2AA-Angriff aufwendiger als der für 1AA. Aus diesem Grund beschreiben wir im Folgenden einen Angriff gegen ein 2AA-Mobilebanking-Verfahren, der Angriffe gegen 1AA aber explizit einschließt.

Vorgehen

Um einen erfolgreichen Angriff gegen das 2AA-Mobilebanking durchzuführen, müssen zwei Apps manipuliert werden: In der Banking-App müssen zuerst die Daten, die der Kunde für seine Überweisung eingibt, so geändert werden, dass sie dem Wunsch des Angreifers entsprechen. Im zweiten Schritt muss der Kunde diese Überweisung im App-basierten Sicherungsverfahren auf demselben Gerät bestätigen. Da die Bank die Auftragsdaten dort erneut zustellt, entsprechen diese dem vom Angreifer festgelegten Empfänger und Betrag. Dementsprechend muss der Angriff sicherstellen, dass genau diese Daten für die Bestätigung des Auftrags herangezogen werden, der Kunde im App-basierten Sicherungsverfahren aber gleichzeitig die Daten sieht, die er ursprünglich in der Banking-App eingegeben hat. Infolgedessen kann das Opfer auch im App-basierten Verfahren keine Abweichung von seinem Auftrag feststellen.

Im Einzelnen sind die in Abbildung 3.3 dargestellten Schritte notwendig:

1) Der Kunde füllt über die Banking-App einen Überweisungsauftrag aus. In dem Moment, als der Kunde den Auftrag an die Bank sendet, tauscht der Angreifer

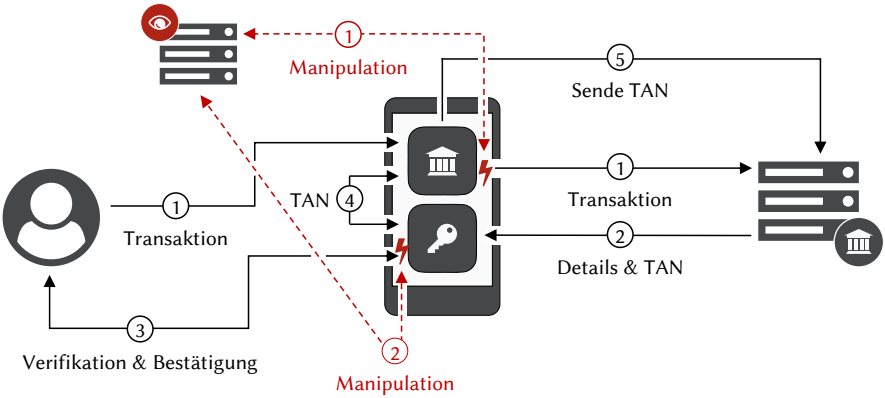


Abbildung 3.3: Schematische Darstellung des Transaktionsmanipulationsangriffs am Beispiel eines 2AA-Verfahrens im Mobilebanking.

Begünstigten und Betrag aus und sorgt somit dafür, dass der Überweisungswunsch des Angreifers bei der Bank eingeht. Die originalen Auftragsdaten hält der Angreifer für den Abruf im nächsten Schritt vor.

2) Die Bank empfängt den Auftrag und erzeugt eine entsprechende TAN. Zusammen mit den eigegangenen Überweisungsdetails schickt die Bank die TAN an das App-basierte TAN-Verfahren. Bevor die App dem Kunden die Transaktionsdetails darstellt, ändert die Schadsoftware den Begünstigten und den Betrag auf die Werte, die ursprünglich von dem Opfer in der Banking-App eingegeben wurden.

3) Als nächstes muss der Kunde die Auftragsdaten verifizieren. Da sowohl IBAN als auch Betrag dem Rechnungsbeleg entsprechen, bestätigt der Kunde die Transaktion.

4) Daraufhin überträgt das App-basierte Sicherungsverfahren die TAN – die eigentlich für den manipulierten Auftrag erstellt wurde – an die Banking-App.

5) Nachdem die TAN vom Nutzer innerhalb der Banking-App gesendet wurde, wird der Auftrag wirksam. Der Kunde hat also aktiv einen anderen Auftrag bestätigt, als er annahm.

Es ist wichtig, dass die App in Schritt 2) nur die angezeigten Daten ändert. Zum Teil ist es so, dass das App-basierte Sicherungsverfahren gar keinen Authentifizierungscode im Sinne einer TAN empfängt, sondern mithilfe seines Schlüsselmaterials eine Signatur bildet, die dann von der Bank verifiziert werden kann. Würden die Transaktionsdaten in dem Moment geändert werden, in dem sie auf dem Gerät eintreffen, würden andere Transaktionsdaten signiert werden, als von der Bank erwartet. Dementsprechend würde die Signaturprüfung fehlschlagen und die Bank würde den Auftrag zurückweisen.

Fallstudie Sparkasse pushTAN

Zur Demonstration der Praxistauglichkeit des Vorgehens stellen wir einen Angriff gegen das pushTAN-Verfahren der Sparkassen vor, den wir als Proof-of-Concept mit dem Hooking-Framework Xposed auf einem Android-Gerät mit Root-Zugriff durchgeführt haben. Unser Schadcode manipuliert die Daten, die die Banking-App („Sparkassen-App“, v2.7.1) an die Bank sendet, und die, die das App-basierte Sicherungsverfahren („pushTAN-App“, v 1.0.4) dem Nutzer darstellt. Hierfür wird der Angriff auf einem Gerät platziert, das der Nutzer zum Mobilebanking mit der Sparkassen- und der pushTAN-App nutzt.

Angriffspunkte der App Sparkasse. Die Sparkassen-App kann bei der Verwendung des pushTAN-Verfahrens zur Absetzung von Überweisungen genutzt werden und ist deshalb der Ansatzpunkt unseres Angriffs. In der Eingabemaske zur Erstellung einer Überweisung sind die folgenden Felder vom Nutzer auszufüllen: Begünstigter, IBAN, Geschäftskennzeichen, Betrag und Verwendungszweck.

Ziel dieses Angriffs ist es, diese Auftragsdaten zu manipulieren, bevor sie an den Server der Sparkasse gesendet werden, ohne diese Veränderungen zu irgendeinem Zeitpunkt für den Nutzer sichtbar zu machen. Durch statische Analyse wurde die Klasse `aoj` und dessen Methode `b` als zuständig für das Senden des Überweisungsauftrags identifiziert. Durch Instrumentation werden vor der Ausführung dieser Methode die Auftragsdaten des Angreifers von dessen Server abgerufen und mittels Reflection manipuliert. Nach der Ausführung der Methode werden die Auftragsdaten wieder auf das Original zurückgesetzt. Der Angreiferserver speichert außerdem dem eigentlichen Überweisungswunsch des Opfers, um diese später in der pushTAN-App abfragen zu können.

Im Folgenden erscheint ein weiteres Fenster, das nochmals die Auftragsdaten anzeigt und die TAN aus der pushTAN-App verlangt. Es war im Rahmen des Angriffs nicht nötig, diese Daten erneut anzupassen, da der Server offenbar keine Bestätigung/-Kopie der Auftragsdaten sendet oder diese von der Sparkassen-App schlichtweg nicht verwendet werden. Im nächsten Schritt muss zur pushTAN-App gewechselt werden.

Angriffspunkte der pushTAN-App. Die pushTAN-App zeigt im Wesentlichen nur die Auftragsdaten und die TAN zur Bestätigung an. Sie lässt sich allerdings nicht wie die Sparkassen-App mit Root betreiben und beendet sich im Gegensatz zu dieser mit einem entsprechenden Warnhinweis. Dieser Schutz musste zunächst deaktiviert werden. Der Hersteller Star-Finanz implementiert selbst keine eigene Überprüfung auf Root-Zugriff, sondern verwendet hierfür das externe und native Modul P Shield. Für das P Shield gibt es im Java-Code allerdings entsprechende Callbacks, die für den Warnhinweis und das Beenden verwendet werden. Das Callback für die Überprüfung auf Root-Rechte in der Methode `rootingStatus` der Klasse `aqh` wurde instrumentalisiert und abgebrochen.

Nach dem Deaktivieren der Root-Überprüfung lässt sich die App regulär verwenden und kann zum Abruf von TANs verwendet werden. Der Server der Sparkasse sendet allerdings die vom Angreifer manipulierten Auftragsdaten und zeigt diese in der pushTAN-App an. Die angezeigten Daten beschränken sich auf den Betrag und die (maskierte) IBAN. Beide Werte müssen auf die vom Nutzer ursprünglich eingegebenen Werte abgeändert werden. Hierfür stellt der Schadcode die zuvor in der Sparkassen-App gespeicherten Transaktionsdaten des Opfers über den Angreifer-Server wieder her. Zur Manipulation wird die Klasse `VisData` und dessen Methode `a` instrumentalisiert. Sie erzeugt aus den vom Sparkassen-Server erhaltenen Daten mehrere Key-Value-Paare zur Anzeige in der App. Dabei werden die beiden Paare Betrag und IBAN entsprechend der abgerufenen Originaldaten des Nutzers abgeändert. Dem Nutzer wird also das erwartete Ergebnis präsentiert, weshalb er die TAN schlussendlich in die Sparkassen-App überträgt und den Auftrag somit bestätigt und wirksam macht. Damit ist das Ziel des Angreifers erreicht: Das Opfer hat eine Transaktion bestätigt, die die Auftragsdetails des Angreifers und nicht die des Nutzers widerspiegelt. Dabei war es dem Opfer zu keiner Zeit möglich, ein atypisches Verhalten festzustellen.

Ein Video, das den beschriebenen Angriff veranschaulicht, ist unter dem folgenden Hyperlink verfügbar: `https://cs1.tf.fau.de/apptan`.

Verallgemeinerung des Angriffs

Der demonstrierte Angriff funktioniert nur für eine ganz bestimmte Version der Sparkassen- und der pushTAN-App. Das liegt darin begründet, dass das vom Obfuskator *ProGuard* eingesetzte Renaming für jeden Kompiliervorgang andere Namen für Klassen und Methoden vergibt. Obwohl sich die notwendigen Anpassungen noch stark in Grenzen halten, erfordern sie dennoch einen manuellen Aufwand. Größere Änderungen, insbesondere an der Sicherheitsarchitektur und den eingesetzten Sicherheitsmaßnahmen, können aber auch für substanziell größeren Aufwand auf der Angreiferseite sorgen. Zudem ist das dargestellte Vorgehen sehr individuell, wodurch eine Adaption auf ein anderes Mobilebanking-Verfahren erschwert wird. Aus Angreifersicht ist es offensichtlich attraktiver, wenn nicht nur ein, sondern mehrere Verfahren im Mobilebanking von dem Angriff erfasst sind.

Der Angriff kann jedoch verallgemeinert werden. Um die grafische Oberfläche einer App zu zeichnen, greifen auch die Banking-Apps und App-basierten Sicherungsverfahren auf Komponenten zurück, die durch Android- oder iOS bereitgestellt werden. Die Aufrufe an diese Klassen und Methoden stehen nicht im Hoheitsbereich des App-Herstellers und sind für ein bestimmtes API-Level immer gleich.

Für die Transaktionsauslösung werden normalerweise Eingabefelder und für die Bestätigung Anzeigefelder verwendet. Unter Android erbt das Eingabefeld `EditText` von dem Anzeigefeld `TextView`. Will ein Entwickler die Nutzereingabe aus dem `EditText` auslesen, verwendet er die Methode `getText`, will er den Inhalt des `TextViews` setzen, nutzt er die `setText`-Methode. Beide Methoden sind letztendlich in der `TextView`-Klasse implementiert. Um die Elemente der grafischen Oberfläche zu adressieren, vergibt der Entwickler für jeden Baustein eine ID. Der Name dieser ID kann sich zwar theoretisch mit einer neuen App-Version ändern, erfordert aber manuelles Eingreifen seitens des App-Herstellers und bleibt in der Regel konstant.

Eine Transaktionsmanipulation lässt sich allein über das Instrumentieren der `TextView`-Klasse und die dazugehörigen `getText`- und `setText`-Methoden realisieren. Voraussetzung ist lediglich, dass die Banking-App zur Eingabe und das App-basierte Sicherungsverfahren zur Anzeige auf ein Oberflächenelement setzt, dass von einem `TextView` erbt. In der Banking-App muss der Angreifer durch Analyse zuerst die IDs der Eingabefelder identifizieren, die für IBAN und Betrag stehen. Selbiges gilt für die IDs der Anzeigefelder, die dieselben Datenpunkte im App-basierten Sicherungsverfahren zur Verifikation darstellen.

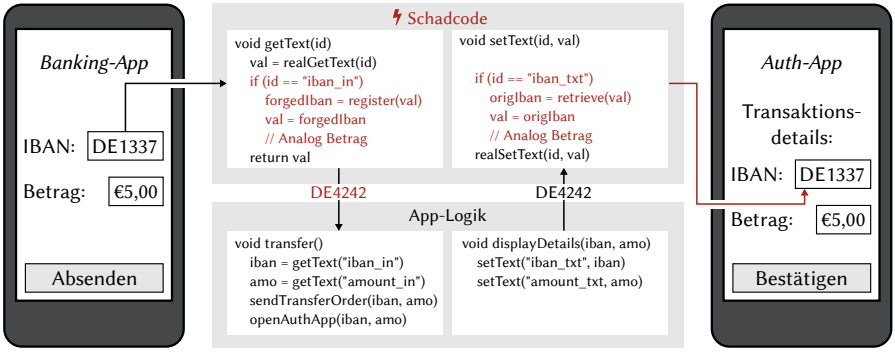


Abbildung 3.4: Generischer Ansatz zur Transaktionsmanipulation.

Abgesehen von diesem manuellen Schritt läuft der Angriff vollautomatisch ab. Neben der folgenden Beschreibung ist der Ablauf in Abbildung 3.4 visualisiert. Wenn der Nutzer in der Banking-App den Überweisungsauftrag ausgefüllt hat, löst er durch Betätigen des „Absenden"-Buttons die Methode `transfer` aus. Diese muss nun die Eingaben des Nutzers auslesen, indem es für die einzelnen Felder die `getText`-Methode ausführt. In dem Moment wird aber nicht der Bibliothekscode des Systems aktiv, sondern unser Schadcode. Dieser führt wiederum die echte `getText`-Methode aus und prüft, ob die ID des Eingabefelds der der IBAN gleicht. Ist das der Fall, sendet der Schadcode die originale IBAN an den Angreiferserver und liefert als Ergebnis der `getText`-Methode die IBAN, die dem Angreiferwunsch entspricht. Für den Betrag ist das Vorgehen analog. Letztendlich sendet die App den Auftrag an die Bank und fordert die Bestätigung im App-basierten Sicherungsverfahren.

Dort funktioniert der Ansatz ähnlich. Statt der `getText`- muss hier die `setText`-Methode durch den Schadcode instrumentiert werden. Wenn das Sicherungsverfahren die Transaktionsdetails empfangen hat, muss sie diese durch Aufrufen von `displayDetails` in der App anzeigen. Dies geschieht, indem sie die Methode `setText` für die Anzeigefelder der IBAN und des Betrags aufruft. Hierdurch wird wieder unser Schadcode aktiv und überprüft, ob eines der Felder die ID der IBAN oder des Betrags trägt. Ist das der Fall, dann wird der Wert so geändert, dass er dem Betrag bzw. der IBAN entspricht, die der Nutzer ursprünglich in der Banking-App eingegeben hat. Hierfür kontaktiert der Schadcode erneut den Angreiferserver. Als Resultat sieht das Opfer die Transaktionsdetails im App-basierten Sicherungsverfahren, die er erwartet und hat demnach keinen Anlass, die Überweisung nicht zu bestätigen.

3.4 Diskussion

In diesem Abschnitt diskutieren wir die Möglichkeiten, die mobile Endgeräte bieten müssen, um die konzeptionellen Defizite zu adressieren. Um einen Replikationsangriff zu verhindern, ist eine sichere Gerätebindung notwendig. Damit ein Nutzer im Mobilebanking zuverlässig die Integrität seines Auftrags überprüfen kann, bedarf es einer sicheren Anzeige. Inwiefern beide Voraussetzungen aktuell und perspektivisch verfügbar sind, stellen wir im Folgenden dar.

3.4.1 Gerätebindung

Wie unsere Fallstudie zeigt, wird unser dargestellter Replikationsangriff durchaus als eine Gefahr in der Praxis wahrgenommen. Aus diesem Grund versuchen die Anbieter App-basierter Sicherungsverfahren, eine Gerätebindung einzuführen.

Das Bilden eines Gerätefingerabdrucks ist jedoch keine adäquate Maßnahme, um dem dargestellten Angriff vorzubeugen. Der Schutz basiert auf der ungültigen Annahme, dass ein Angreifer die Werte, die in den Gerätefingerabdruck einfließen, nicht kennt und auch nicht auslesen kann. Die Informationen, die dabei herangezogen werden, sind jedoch öffentlich, weshalb sie für eine Authentifizierung ungeeignet sind [Bia+17]. Außerdem ist das Auslesen einzigartiger Geräteinformationen aus Sicht der Privatsphäre problematisch. Deshalb arbeiten Apple und Google daran, globale, vom Nutzer nicht zurücksetzbare Kennungen zu eliminieren. Studien zeigen zwar, dass sich eine Vielzahl, von Werten in Summe zu einem einzigartigen Fingerabdruck zusammenfassen lassen, die Zuverlässigkeit dieser Ansätze ist für ein Authentifizierungssystem aber zu gering [Wu+16; Kur+16].

Gleichwohl gibt es seit Android 6 [OSM19] und iOS 9 Hardwarefunktionen, mit denen sich eine sichere Gerätebindung herstellen lässt. In beiden Fällen erlaubt es der Android `KeyStore` bzw. die iOS `KeyChain`, ein asymmetrisches Schlüsselpaar in einer vertrauenswürdigen Ausführungsumgebung zu erzeugen, die vom regulären System isoliert betrieben wird. Apple realisiert diese Umgebung durch einen abgeschotteten Co-Prozessor (Secure Enclave), während Android zumeist die ARM TrustZone verwendet. Die konkrete Implementierung ist herstellerabhängig, muss jedoch zwangsläufig in Hardware erfolgen. In dieser Umgebung generierte Schlüssel sind zum einen so gestaltet, dass sie nur innerhalb der App verwendet werden können, die sie erzeugt haben. Zum anderen kann der private Schlüssel nicht exportiert werden, sondern verweilt zwangsläufig in der sicheren Ausführungsumgebung.

Nachdem auf den privaten Schlüssel nicht direkt zugegriffen werden kann, lässt er sich auch nicht kopieren [ST16; Teu+13].

Ein Angreifer könnte den vorgestellten Angriff nun so umgestalten, dass er die App nicht mehr kopiert, sondern so verändert, dass er aus der Ferne über das Opfer-Smartphone Zugriff auf den Schlüssel hat. Dieser Angriff wäre auch nach wie vor von unserem Angreifermodell gedeckt. Auf aktuellen Smartphones mit einem Fingerabdrucksensor (oder einer vergleichbaren biometrischen Erkennung) ist es jedoch möglich, den Schlüssel nur dann zugreifbar zu machen, wenn der Nutzer dies mit seinem Finger bestätigt [Bia+18].

Eine Lösung, die sowohl die hardwaregestützte Schlüsselverwaltung verwendet als auch die physische Präsenz des Nutzers über den Biometriesensor sicherstellt, kann eine robuste Gerätebindung herstellen. Dennoch zeigen unsere Stichproben, dass diese Funktionen noch zurückhaltend genutzt werden.

3.4.2 Sichere Anzeige

Der zurückliegende Abschnitt hat gezeigt, dass eine Echtzeittransaktionsmanipulation nicht nur praktisch durchgeführt, sondern auch weitreichend automatisiert werden kann. Ursächlich für die Machbarkeit des Angriff ist – und das zeigt der generische Angriff deutlich – das Fehlen einer sicheren Anzeige. Dass dieser Umstand nicht erkannt wurde, überrascht, weil es bereits beim iTAN-Verfahren zu vergleichbaren Schadensfällen gekommen ist. Dort ist man zwar mit dem Umstand konfrontiert, dass es gar keine erneute Anzeige der Transaktionsdetails durch das Sicherungsverfahren gibt, das Resultat ist jedoch dasselbe: Ist ein Gerät mit Schadsoftware befallen, fällt die ganze Sicherheit des Systems in sich zusammen. Während die sichere und vertrauenswürdige Anzeige beim chipTAN-Verfahren noch ein Schlüsselargument für dessen Einführung war, bewegt man sich mit der Einführung des Mobilebankings wieder ein gutes Stück zurück.

Obwohl es auf modernen Android- und iOS-Systemen mittlerweile Möglichkeiten gibt, eine sichere Gerätebindung umzusetzen und damit Kopierangriffe zu verhindern, existiert für mobile Endgeräte nach wie vor keine massentaugliche sichere Anzeige. Gerade unter Android konterkariert eine Reihe von Design-Entscheidungen eine vertrauenswürdige Anzeige [Fra+17; KBM18]. Neben akademischen Arbeiten [Sun+15; Yin+18; Zhe+16; Esk+19; Fil+11] hat die sichere Anzeige auch in

Android 9 Einzug genommen [Dan18]. Mithilfe dieser Funktion können beliebige Daten sicher angezeigt und bestätigt bzw. abgebrochen werden. Die sichere Interaktion mit dem Nutzer erfolgt über die Hardware-Knöpfe. Der Touchscreen kann genauso wenig verwendet werden, wie es Gestaltungsmöglichkeiten für den angezeigten Dialog gibt.

Damit ein App-basiertes Sicherungsverfahren (2AA) oder die Banking-App (1AA) von dem Verfahren Gebrauch machen können, muss mit der Registrierung ein asymmetrisches Schlüsselpaar erstellt werden. Hierbei zeigt ein entsprechender Schalter an, dass es ausschließlich für die „Protected Confirmation“ genannten Dialoge nutzbar ist. Der öffentliche Schlüssel muss an den Banken-Server übermittelt werden. Wenn dem Nutzer nun ein solcher geschützter Dialog angezeigt wird, signiert die vertrauenswürdige Umgebung die Transaktionsdaten mit dem privaten Schlüssel. Auf diese Weise kann sich der Bank-Server sicher sein, dass die signierten Daten auf dem Smartphone-Bildschirm angezeigt wurden. Rückschlüsse darüber, ob der Nutzer die Transaktionsdaten auch gewissenhaft auf Richtigkeit überprüft hat, ergibt sich dadurch aber natürlich nicht.

3.5 Fazit

In diesem Kapitel haben wir uns mit dem ersten Teil von Forschungsfrage 2 (Angriffsfläche Mobilebanking) auseinandergesetzt. Wir haben zwei konzeptionelle Angriffe identifiziert: Erstens ergibt sich durch die Softwareimplementierung der App-basierten Sicherungsverfahren die Möglichkeit zur Replikation der personalisierten App. Zusätzlich ist die Personalisierung des App-basierten Verfahrens eine strukturelle Schwachstelle. Zweitens lässt der Paradigmenwechsel im Mobilebanking eine Echtzeittransaktionsmanipulation zu, die sich daraus ergibt, dass nur noch ein einziges Gerät an der Transaktionsauslösung und -bestätigung beteiligt ist. Eine Fallstudie zum photo- bzw. pushTAN-Verfahren hat beiden Angriffen Nachdruck verliehen.

Dennoch können Hardwaremöglichkeiten in Zukunft sichere Mobilebanking-Verfahren ermöglichen. Eine effektive Gerätebindung ist unter der Verwendung von hardwaregebundener Kryptographie bereits weitläufig möglich. Eine sichere Anzeige sorgt jedoch erst mittel- bis langfristig für einen Schutz gegen Transaktionsmanipulation.

4

Grenzen der App-Härtung

Die Systemsicherheit sollte nicht auf der Geheimhaltung ihrer Implementierung oder Komponenten beruhen.

— NIST, 2008 [SJT08]

Mit den in Kapitel 3 dargestellten Angriffen haben wir gezeigt, dass die neuen Voraussetzungen im Mobilebanking einem Angreifer in die Hände spielen. Auf diese konzeptionellen Schwächen angesprochen, verweisen die Banken gerne auf die spezielle Härtung ihrer Apps. So merkt z. B. die Commerzbank zur Sicherheit ihres photoTAN-Verfahrens an, dass dieses „mit einem speziellen Schutz und einer entsprechenden Härtung versehen“ [CoBa] sei. Auch die Berliner Volksbank weist darauf hin, dass ihr App-basiertes Sicherungsverfahren *VR-SecureGo* ein „[h]ohes Sicherheitsniveau durch spezielle App-Härtung“ [VBB] biete. Unsere Analysen zeigen, dass insgesamt ein substanzieller Teil der Banken auf einen zusätzlichen Schutz setzt. Die Härtungsmaßnahmen sind dabei keine Eigenentwicklung, sondern kommen von einem kommerziellen Hersteller, der sich auf den Bereich der App-Härtung spezialisiert hat.

In den folgenden Abschnitten stellen wir dar, dass sich bereits ein relevanter Markt mit einer Vielzahl an Herstellern gebildet hat. Wir beschreiben die wichtigsten Funktionen und vergleichen eine Auswahl an Herstellern. Im Anschluss zeigen wir zwei praktikable Angriffe gegen den bedeutendsten Anbieter von App-Härtung auf dem deutschen Banking-Markt: P Shield (die authentische Firma wurde auf Wunsch des betroffenen Unternehmens pseudonymisiert).

4.1 Marktüberblick

Kommerzielle App-Härtungslösungen zielen darauf ab, eine App aus sich heraus zu schützen. Zu diesem Zweck implementieren die Lösungen neben statischen auch dynamische Schutzmaßnahmen. Deshalb spricht man im Kontext von App-Härtung auch von Runtime Application Self-Protection (RASP). Obwohl RASP-Produkte sowohl für Android als auch iOS existieren, konzentrieren wir uns im Folgenden auf die Lösungen für Android. Der Grund ist, dass der Bedarf an zusätzlichen Schutzmaßnahmen unter Android als höher zu bewerten ist, weil Android weltweit den höchsten Marktanteil hat und durch die hohe Fragmentierung sehr unterschiedliche Sicherheitsgarantien bietet.

Ein wichtiger Baustein im Schutzsystem der RASP-Anbieter sind statische Verschleierungsmaßnahmen (Obfuskierung). Sie führen Code-Transformationen ein, die es Menschen und automatisierten Werkzeugen erschweren sollen, ein Programm zu analysieren. Statische Obfuskierung ist insbesondere auch im Spielemarkt beliebt, um den integrierten Kopierschutz zu schützen. In diesem Geschäftsbereich ist es wichtig, dass insbesondere nicht unmittelbar nach Erscheinen des Spiels – wenn die Absatzzahlen besonders hoch sind – Raubkopien verfügbar sind. Die Zielgruppe der RASP-Anbieter ist aber nicht der Spielemarkt, sondern der Bereich der kritischen Infrastruktur: Apps aus den Domänen Gesundheit, öffentlicher Sektor und nicht zuletzt Finanzen. Für dieses Feld ist es nicht ausreichend, die Apps für eine limitierte Zeit allein unter Berücksichtigung monetärer Gesichtspunkte zu schützen. Aus diesem Grund setzten die RASP-Anbieter auch auf dynamische Sicherungsmaßnahmen und das automatisierte Einbetten von Sicherheits-Best-Practices.

Im Weiteren verwenden wir den Begriff *RASP-Anbieter* für die Entwickler der RASP-Lösung, *Kunde* für die App-Entwickler, die das Produkt des RASP-Anbieters einsetzen, und *Endnutzer* für den Verwender der Kunden-App.

4.1.1 Anbieter und Funktionen

Im Folgenden stellen wir RASP-Lösungen für Android vor und beschreiben die von ihnen beworbenen Schutzmaßnahmen. Unsere ursprüngliche Intention, durch manuelle Analyse verlässliche Aussagen über die Güte der einzelnen RASP-Produkte zu liefern, mussten wir verwerfen. Das hat primär zwei Gründe: Erstens sind die Kunden der Anbieter im Allgemeinen nicht bekannt, wodurch das Ausfindigmachen

geeigneter Apps zur Analyse erschwert wird. Zweitens sind die Produkte oft konfigurierbar. Der Grund dafür ist nicht zuletzt das Preismodell der Hersteller. Dadurch ergibt sich die Situation, dass selbst bei der Verfügbarkeit von mehreren Apps für eine RASP-Lösung Aussagen zur Güte schwer sind. Aus diesen Gründen stützt sich Tabelle 4.1 auf die Aussagen der Produktbeschreibungen der einzelnen Hersteller. Es handelt sich dabei um eine Auswahl aus dem Gartner *Market Guide for Application Shielding* [Gar17]. Wir weisen darauf hin, dass der Umfang des Informationsmaterials der Hersteller sehr unterschiedlich ist. Die Tabelle zeigt deshalb nur, ob eine Schutzfunktion durch den Anbieter genannt (●) oder nicht genannt wird (—).

In der Tabelle fällt auf, dass das P Shield die meisten Funktionen bereitstellt. Obwohl dieser Aussage mit einem gewissen Vorbehalt zu begegnen ist, kann P ohne Weiteres als wichtiger Marktteilnehmer bezeichnet werden. Gerade auf dem deutschen Finanzmarkt hat P Shield eine vorherrschende Stellung inne und schützt den größten Teil der relevanten Apps der Banken. Aus diesem Grund widmen wir uns der Angriffsfläche des P Shields in Abschnitt 4.2 ausführlich, zeigen aber bereits in der nachfolgenden Beschreibung, wie die Umsetzung der einzelnen Schutzmaßnahmen durch P geschieht.

Anti-Tampering. Wenn es einem Angreifer gelingt, den Programmcode zur Laufzeit zu modifizieren, erlangt er dieselben Rechte wie die Client-App. In diesem Kontext ist er z. B. in der Lage, die Serverkommunikation zu manipulieren oder Lizenz- und Sicherheitsüberprüfungen zu deaktivieren. Um sicherzustellen, dass ein Dritter den Programmcode der App nicht manipuliert hat, nutzen RASP-Anbieter Maßnahmen zum Integritätsschutz, die Anti-Tampering genannt werden. Wenn Code verändert werden kann, ist es auch möglich, andere Überprüfungen zu deaktivieren. Deshalb ist Anti-Tampering ein Eckpfeiler aller RASP-Lösungen.

Ein naheliegender Ansatz zur Umsetzung von Anti-Tampering ist das Überprüfen der Signatur des Android Package Kits (APK) beim Start der App. Falls die Signatur nicht dem erwarteten Entwicklerzertifikat entspricht, dann schließt die RASP-Lösung auf eine Modifikation durch einen unautorisierten Dritten. Fortschrittlichere Ansätze verteilen die Signaturprüfungen an verschiedenen Stellen der App oder nutzen Wasserzeichen, um ein Entfernen zu erschweren [RCL14].

P Shield überprüft die Integrität der `base.apk`, indem es das Entwicklerzertifikat beim Start der App aus einer verschlüsselten Konfigurationsdatei lädt und abgleicht. Zusätzlich zur Überprüfung der APK prüft es außerdem die Hashwerte von bestimmten Android-spezifischen Dateien – bspw. das `AndroidManifest.xml` oder die `classes.dex` – sowie die eigene Native Library `libshield.so`.

RASP-Produkt	Anti-Tampering	Anti-Hooking	Anti-Debugging	Anti-Emulator	Obfuskierung	Whitebox-Kryptographie	Gerätebindung	Root-Erkennung	Anti-Keylogger	Anti-Screenreader	Datenverschlüsselung	Sichere Kommunikation
Arxan	●	●	●	—	●	●	—	●	—	—	●	—
DNP HyperTech CrackProof	●	—	●	●	—	—	—	●	—	—	—	—
Entersekt Transakt	●	—	—	—	—	—	●	●	●	—	—	●
Gemalto Mobile Protector	—	●	●	—	●	—	●	●	●	—	●	●
GuardSquare DexGuard	●	●	●	●	●	●	—	●	—	—	—	●
Inside Secure Core	●	—	●	—	●	●	—	●	—	—	—	—
Intertrust WhiteCryption	●	—	●	—	●	●	●	●	—	—	—	—
PreEmptive DashO	●	—	●	●	●	—	●	●	—	—	—	—
P Shield	●	●	●	●	●	●	●	●	●	●	●	●
SecNeo AppShield	●	—	●	—	●	—	—	—	—	—	●	—

Tabelle 4.1: Übersicht über RASP-Anbieter und beworbene Schutzmaßnahmen.

Anti-Hooking. Selbst wenn es einem Angreifer nicht gelingt, den Code einer App zu verändern, kann er immer noch Anweisungen im Kontext der App ausführen, indem er bestimmte Aufrufe instrumentalisiert (Hooking). Beliebte Ziele sind Aufrufe an die Android-API oder an das Betriebssystem. Dadurch wird ein Angreifer befähigt, Methodenparameter und Rückgabewerte zu verändern oder existierende Methoden zu ersetzen. Unter Android existiert eine Vielzahl von Werkzeugen, die das Instrumentalisieren von Klassen und Methoden erlaubt. Dadurch, dass der Ansatz außerhalb der App und zum Teil in einer höheren Privilegienebene ansetzt, ist eine Erkennung schwierig. Die Erkennungsroutinen zum Anti-Hooking sind deshalb auf die charakteristischen Spuren der bekannten Hooking-Werkzeuge zugeschnitten. P Shield sucht z. B. nach Artefakten von *Xposed* und *Cydia Substrate*. Das beliebte und mächtige Werkzeug *Frida*, mit dem wir unsere dynamischen Analysearbeiten durchgeführt haben, bleibt hingegen unentdeckt.

Anti-Debugging. Ein Debugger kann ebenfalls eingesetzt werden, um den Kontrollfluss oder Funktionsrückgaben zu ändern, ohne den Code selbst zu manipulieren. Darüber hinaus liefert Debugging dem Angreifer Erkenntnisse zur Funktionsweise der Härtungskomponente und der App selbst. Deshalb versuchen die RASP-Lösungen Debugger zu erkennen und gegebenenfalls die App zu beenden. Unter Android kann sich ein Debugger an grundsätzlich zwei Stellen verbinden: Zum einen mit einen `ptrace`-basierten Native Debugger für C/C++ und zum anderen mit einem Java-Debugger, der das JDWP-Protokoll spricht.

Um einen Java-Debugger zu erkennen, überprüft eine triviale Erkennungsroutine lediglich den Rückgabewert von `isDebuggerConnected` auf `true`. P Shield wählt einen weniger offensichtlichen Ansatz und verhindert das Verbinden eines Debuggers durch das Manipulieren von Datenstrukturen: es überschreibt die Funktionspointer innerhalb der `JdwpState`-Datenstruktur mit `false`, die für den Austausch von Debugger-Paketen zuständig sind. Um gegen Native Debugger vorzugehen, erzeugt das P Shield mehrere Threads, die untereinander als `ptrace`-Debugger fungieren. Dadurch, dass `ptrace` nur einen und nicht mehrere gleichzeitige Debugger unterstützt, kann sich ein Dritter nicht mit der App verbinden, ohne die Schutzmaßnahme vorher deaktiviert zu haben.

Anti-Emulator. Wenn es einem Angreifer möglich ist, die App in einem Emulator, einer virtuellen Maschine oder einer Sandbox zu betreiben, kann er dadurch die Programmausführung beobachten und instrumentalisieren. Mithilfe des Android-Emulators ist es einfach, den Zustand des Systems zu inspizieren, es zu einem zuvor gespeicherten Zeitpunkt zurückzusetzen oder die Operationen der App zu analysieren. Um Emulatoren und vergleichbare Umgebungen zu erkennen, implementieren RASP-Produkte in der Regel Erkennungsroutinen, die bereits von Schadsoftware bekannt sind [MMP14; Pet+14; VC14].

Obfuskierung. Obfuskierung versucht den Code soweit wie möglich unkenntlich und schwer analysierbar zu machen, ohne dabei jedoch die Funktionalität zu ändern. Die Werkzeugkette zur Entwicklung von Android-Apps beinhaltet standardmäßig *ProGuard*, das durch Renaming von Klassen- und Methodennamen zur Übersetzungszeit eine grundlegende Obfuskierungstechnik implementiert [Goo]. Durch seine weite Verbreitung ist die Art und Weise wie *ProGuard* arbeitet gut erforscht und es existieren Lösungen, um den Vorgang umzukehren (Deobfuskierung) [Bic+16]. Neben Renaming gibt es noch eine Vielzahl weiterer Verschleierungsmaßnahmen, wie z. B. Control Flow Flattening, Opaque und Random Predicates oder auch Function Merging und Splitting [CN09].

Obwohl perfekt obfuskierter Programmcode laut Barak u. a. unmöglich ist, wenn er auf dem gleichen Gerät ausgeführt wird, das auch der Angreifer kontrolliert, kann es den Analyseaufwand deutlich erhöhen [Bar+01]. Gleichwohl stellen Krügel u. a. 2004 fest, dass Obfuskierung ein Wettrüsten ist, das üblicherweise den Deobfuskierer begünstigt [Krü+04]. An dieser Situation hat sich auch 2016 nichts geändert [Sch+16].

Eine weitere Kategorie der Verschleierungsmaßnahmen ist das sog. DEX-Packing. Die Technik ist bei Android-Schadsoftware beliebt und zielt darauf ab, sowohl die statische als auch die dynamische Analyse zu erschweren. DEX-Packing liefert die `classes.dex`, die den DEX-Bytecode der App beinhaltet, im Ganzen oder in Teilen verschlüsselt aus und entschlüsselt es erst zur Laufzeit. Die Beliebtheit der Technik ist bei Schadsoftware ungebrochen, weshalb sich die Forschung schon intensiv dem automatisierten Unpacking bzw. Analysetechniken gewidmet hat, die auch bei eingesetztem DEX-Packing noch effektiv funktionieren [Yan+15; ZLY15; Xue+17; Dua+18].

Der Kern der Schutzmaßnahmen des P Shields ist in der `libshield.so` implementiert und liegt verschlüsselt und obfuskiert vor. Der Java-Teil befindet sich im Paket `no.promon.shield` und ist abhängig von der einsetzenden App durch Renaming obfuskiert. Im Falle einer Obfuskierung werden die Namen aller Klassen, Methoden und Felder in eine zufällige, achtstellige Zeichenkette umbenannt. Die Verschleierung beschränkt sich jedoch auf P-eigenen Code; der Code der Client-App ist je nach App-Hersteller noch unabhängig durch *ProGuard* verschleiert.

Whitebox-Kryptographie. Ähnlich zur Obfuskierung von Programmcode zielt Whitebox-Kryptographie darauf ab, Geheimnise zu verschleiern [SW08]. Es versucht jedoch nicht, die App-Logik unkenntlich zu machen, sondern kryptographisches Schlüsselmaterial, das in der App statisch vorliegt [Cho+02]. Die Implementierungen versuchen eine Einwegfunktion anzubieten, die es dem Kunden der Härtungslösung einfach erlaubt, kryptographische Funktionen anzuwenden, es einem Angreifer aber gleichzeitig möglichst schwer macht, den zugrundeliegenden Schlüssel in Erfahrung zu bringen. Da der Schlüssel in irgendeiner Form letztendlich statisch vorliegen muss, kann er immer durch Reverse Engineering gewonnen werden [GMQ07]. Der durch den Analysten zu investierende Aufwand kann jedoch bedeutend höher sein. Das Reverse Engineering der Whitebox-Implementierung ist jedoch nicht immer nötig: Ein Angreifer kann die Whitebox auch als Blackbox betrachten und sie als Ganzes zum Ent- und Verschlüsseln nutzen.

Gerätebindung. Zwar kann eine Gerätebindung nicht verhindern, dass eine personalisierte App von einem Gerät auf ein anderes kopiert wird, sie kann aber versuchen, die Ausführung auf dem unautorisierten Gerät zu unterbinden. In Abschnitt 3.3.1 wurde bereits angesprochen, dass RASP-Produkte versuchen, eine Gerätebindung über das Bilden eines Gerätefingerabdrucks zu realisieren. Dabei werden beim initialen Start Werte aus der Umgebung gelesen und gespeichert. In periodischen Abständen liest das Härtungsprodukt die Umgebungsvariablen erneut und gleicht sie mit den erwarteten Werten ab. Stimmen diese nicht überein, reagiert die Lösung üblicherweise mit dem Beenden der App oder setzt die App sogar komplett zurück. Die Eignung eines Gerätefingerabdrucks für eine effektive Gerätebindung diskutieren wir ausführlich in Abschnitt 3.4.1. P Shield setzt zur Bildung des Gerätefingerabdrucks lediglich auf die `Build.SERIAL` und – insofern die entsprechende Berechtigung vorliegt – die IMEI.

Root-Erkennung. Wie in Abschnitt 3.2 angesprochen, gibt es einen kleinen Nutzerkreis, der sich durch Rooting bewusst ausgeweitete Rechte verschafft. Da Rooting das Sandboxing-Prinzip auflöst, bietet sich so eine erhöhte Angriffsfläche. Aus diesem Grund bieten RASP-Lösungen in der Regel Erkennungs- und Reaktionsroutinen an. Neben den in Abschnitt 3.2 genannten statischen Dateisystemartefakten implementiert das P Shield zusätzliche Maßnahmen, um auch dynamische Anzeichen eines gerooteten Geräts zu erkennen. Zu diesem Zweck überprüft es alle über das `proc`-Dateisystem gelisteten Prozesse darauf, ob sie zu einer Root-Management-App gehören (z. B. *SuperSU* oder *Magisk*). Die Suche erstreckt sich auch auf das Programmabbild unter `/proc/self/exe`. Die Überprüfungen lassen sich leicht umgehen, indem die betreffenden Programme umbenannt werden; dadurch sind manche Apps, die auf Root-Rechte angewiesen sind, jedoch auch nicht mehr verwendbar. In jedem Fall ist eine Root-Erkennung ungeeignet, um eine bösartige Rechteausweitung zu erkennen, geschweige denn zu verhindern [Che+18].

Anti-Keylogger. Die Möglichkeit, Drittanbietertastaturen unter Android zu installieren, birgt die Gefahr eines Keyloggers [CCK15]. Unter Android gibt es nicht wie unter iOS die Möglichkeit, für bestimmte Eingabefelder die vertrauenswürdige Systemtastatur zu erzwingen. Außerdem ist eine virtuelle Tastatur unter Android eine weitere reguläre App, deren Sicherheit die eigentliche Client-App nicht beeinflussen kann [Lin+14]. Deshalb liefern manche RASP-Anbieter eine eigene Tastatur aus, die für sensible Eingabefelder verwendet werden kann. Außerdem erlaubt es das P Shield, bestimmte Tastaturen anhand ihrer App-Signatur für vertrauenswürdig zu erklären. Verwendet der Nutzer eine vertrauenswürdige Tastatur, dann kann er sie auch für P-geschützte Eingabefelder verwenden.

Anti-Screenreader. Wie von Fratantonio u. a. 2017 sowie Kalysch, Bove und Müller 2018 gezeigt, kann Schadsoftware die Accessibility-API Androids missbrauchen, um Informationen mitzuschneiden. Anti-Screenreader-Methoden versuchen diesen Angriffen entgegenzuwirken. Das P Shield iteriert alle installierten Apps, die die Acccessibility-API verwenden, und prüft ob deren Name und Signatur auf der intern geführten Liste der vertrauenswürdigen Apps steht. Ist dem nicht der Fall, beendet P – abhängig von der Kundenkonfiguration – die App. Eine weitere Möglichkeit, angezeigte Daten mitzuschneiden, ist das Anfertigen von Screenshots. Wenn entsprechend konfiguriert, setzt das P Shield zum Start der App für bestimmte Fenster das `FLAG_SECURE`, das solche Aufnahmen verhindert.

Datenverschlüsselung. RASP-Produkte bieten die Möglichkeit, die Daten der App zusätzlich zu verschlüsseln. Dadurch soll verhindert werden, dass ein höher privilegierter Angreifer die privaten Daten der geschützten App lesen kann. Somit sollen sensitive Daten, wie bspw. Transaktionshistorien oder kryptographisches Schlüsselmaterial, vor unbefugtem Zugriff geschützt werden. Das Hauptproblem der RASP-Lösungen ist die sichere Ablage des für die Chiffrierung genutzten Schlüssels. Insofern der Schlüssel nicht zur Laufzeit von einer Benutzereingabe abgeleitet wird, bleibt der Härtungslösung nur noch eine Verschleierung. Das P Shield bietet seinen Kunden die `SecureStorage`-Klasse, die eine Datenverschlüsselung auf Basis von Ps Whitebox-Kryptographie erlaubt. Die Fallstudie zu P liefert hierzu später detaillierte Einblicke.

Sichere Kommunikation. Maßnahmen zur sicheren Kommunikation reduzieren nicht nur das Risiko zu Man-in-the-Middle-Angriffen auf Netzwerkebene, sondern auch gegenüber lokalen Angreifern. Das P Shield hält hierfür zwei Funktionen bereit. Erstens bietet es eine gehärtete HTTPS-Netzwerk-Klasse für Java. Entscheidet sich der Kunde für einen Wechsel von der `HttpUrlConnection` zur `PromonHttpUrlConnection`, werden alle Netzwerkanfragen und -antworten durch die P Native Library geleitet. Diese implementiert Zertifikatspinning entsprechend der Kundenkonfiguration und baut somit nur Verbindungen zu vertrauenswürdigen Endpunkten auf. Neben dem Zertifikatspinning ist es auch möglich, ein TLS-Client-Zertifikat zu verwenden, das ebenfalls über die Native Library realisiert ist. Dadurch kann der Server feststellen, ob er aus der App heraus kontaktiert wird oder über eine Drittanbieter-Anwendung. Zweitens implementiert das P Shield auch ein eigenes Protokoll auf Basis der `DeviceManagement`-Klasse, die im Kern ebenfalls über die `libshield.so` realisiert ist. Die API erlaubt es, eine App bei einem Server zu registrieren und z. B. Transaktionen zu tätigen. Das Protokoll ist primär auf eine Verwendung im Banking-Umfeld ausgelegt.

4.1.2 Angreifermodell

Die RASP-Anbieter haben primär zwei Schutzziele: Das geistige Eigentum des App-Herstellers und die Nutzerdaten innerhalb der App. Die Annahmen der Härtungsindustrie sind zwar spezifischer, aber im weiteren Sinne gleichwertig zu dem allgemeinen Angreifermodell, das wir in Abschnitt 3.1 bemüht haben.

Geistiges Eigentum. Das Ziel des Angreifers ist es, das geistige Eigentum oder andere Geheimnise der Client-App – z. B. hartkodierte Zugangsdaten [Ege+13] – in Erfahrung zu bringen. Der Angreifer hat hierfür vollen Zugriff auf das eigene Gerät und versucht mittels statischer und dynamischer Analyse Erkenntnisse über die App zu gewinnen. Seine Absichten können vielfältig sein, schließen aber die folgenden Szenarien ein: Umgehung der Lizenzüberprüfung zum Raubkopieren der App, Kopie der Geschäftslogik, oder das Öffentlichmachen von Zugangsdaten. Der Angreifer könnte auch beabsichtigen, eine eigene Anwendung zu schreiben, die die API der originalen App verwendet, ohne eine entsprechende Vereinbarung mit dem Hersteller der App zu haben.

Nutzerdaten. Das Ziel dieses Angreifers ist es, über einen entfernten Kanal Nutzerdaten zu sammeln oder Transaktionen im Namen eines echten Nutzers zu tätigen. Ein solcher Angreifer bemüht Social Engineering [Kro+15b], Drive-by-Downloads [CKV10], App Piggybacking [Li+17a] oder jede andere Methode, die es ihm erlaubt, Code auf dem Gerät des Opfers auszuführen. Dieser Angreifer ist omnipotent, da er Zugriff zu Exploits hat, die eine Rechteausweitung erlauben. Er ist damit in der Lage, das Gerät vollständig zu kompromittieren, das die Ziel-App ausführt. In diesem Kontext kann er Code innerhalb der App ausführen, im Namen des Nutzers mit dem Backend kommunizieren, oder auch die komplette personalisierte App auf ein eigenes Gerät kopieren. Zusätzlich ist er in der Lage Man-in-the-Middle-Angriffe durchzuführen und somit sensible Informationen zu gewinnen [Olt+15].

4.2 Fallstudie P Shield

In diesem Abschnitt präsentieren wir eine Fallstudie zur populären App-Härtungslösung P Shield. P ist ein norwegisches Unternehmen, das sich der Sicherheit von mobilen Apps und PC-Programmen verschrieben hat. Das Unternehmen ist ein Global Player: Ihr Produkt, P Shield, hat weltweit rund 100 Geschäftskunden und schützt so ungefähr 100 Millionen Endanwender [Tan17]. Wer die Kunden Ps sind, ist

jedoch nicht öffentlich bekannt. Aus diesem Grund haben wir den Google Play Store nach Apps durchsucht, die Ps Härtungslösung einsetzen. Wie bereits angedeutet, sind wichtige Funktionen der App in die Native Library `libshield.so` ausgelagert. Die Bibliothek folgt einem charakteristischen Namensschema und ist deshalb leicht innerhalb einer Android-App ausfindig zu machen. Auf diesen Weg haben wir in über 150 000 kostenfreien Android-Apps 31 Apps identifiziert, die P Shield zum Dezember 2018 einsetzten.

Obwohl P sein Produkt auch für andere Bereiche der kritischen Infrastruktur bewirbt – z. B. für Automobilhersteller – stammen alle 31 Apps aus der Finanz-Kategorie des Google Play Stores. 20 Apps kommen aus Deutschland, jeweils zwei aus Norwegen und Finnland und jeweils eine aus den Niederlanden, Schweden, Großbritannien, den USA, Mexiko, Brasilien und Hong Kong. Die große Popularität am deutschen Markt ist auffällig: P Shield schützt den überwiegenden Anteil der Apps deutscher Banken. Zum 23. April 2018 nutzten vier der 10 beliebtesten Finanz-Apps in Deutschland Ps Lösung.

Die große Beliebtheit des P Shields suggeriert, dass die Lösung des norwegischen Herstellers besonders hochwertig ist. Auch wird die Lösung durchweg von allen drei Säulen des deutschen Bankensystems eingesetzt; wichtige Banking-Apps und App-basierte Sicherungsverfahren der öffentlich-rechtlichen, genossenschaftlichen und privatrechtlichen Institute setzen P Shield ein. Aus diesem Grund widmen wir uns im Folgenden ausführlich dem Produkt. Wir geben detaillierte Einblicke in die Funktionsweise des P Shields und zeigen in zwei Angriffen auf, wie sich die Sicherungsmaßnahmen vollständig aushebeln lassen. Hierfür haben wir das Werkzeug *Nomorp* entwickelt, das vollautomatisch arbeitet: *sNomorp* bricht das Ziel zum Schutz des geistigen Eigentums, *dNomorp* das zum Schutz der Nutzerdaten.

Wir hatten keinen Zugriff auf Programme, die Kunden Ps einsetzen, um ihre App mit P Shield zu härten. Darüber hinaus hatten wir auch keinerlei Kenntnis von Insiderinformationen oder gar den Quelltext des P Shields vorliegen. Ein Gegenspieler konnte und kann also ohne Einschränkung dieselben Techniken und Werkzeuge einsetzen. Für unsere Analyse verwendeten wir sowohl manuelles statisches Reverse Engineering, als auch dynamische Analysetechniken, die hauptsächlich auf einer angepassten Android-Laufzeitumgebung (ART) und dem beliebten Instrumentalisierungswerkzeug *Frida* basieren.

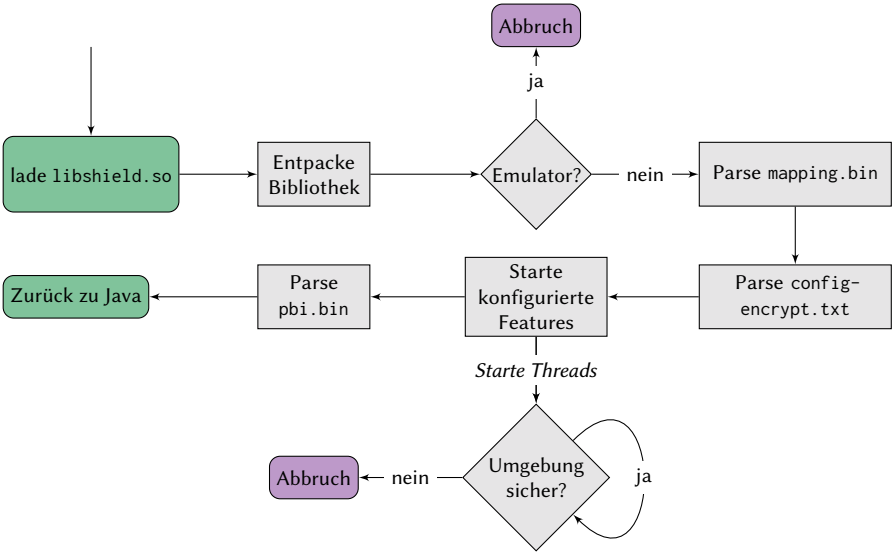


Abbildung 4.1: Lebenszyklus der P Shield Native Library `libshield.so`.

4.2.1 Integration und Initialisierung

Die Integration des P Shields wird von P als sehr einfach beschrieben [Pro]. Demnach muss ein Kunde nur eine Konfigurationsdatei anpassen, die bestimmt, welche Sicherheitsfunktionen aktiviert werden und wie auf Anomalien zu reagieren ist. Anschließend härtet Ps Integrationswerkzeug die App entsprechend der Konfiguration. Es ist dabei nicht notwendig, das P Shield aktiv in den Entwicklungsprozess mithilfe eines Software Development Kit einzubinden; der Kunde liefert stattdessen das APK als Eingabe für das P Integrationswerkzeug. Die Ausgabe ist eine gehärtete APK, die der Kunde nun sofort veröffentlichen kann.

Nach Installation und Start gestaltet sich der Lebenszyklus einer P-gehärteten App auf dem Kundengerät wie in Abbildung 4.1 dargestellt. Die App lädt zuerst die Native Library `libshield.so`. Zu diesem Zweck hat das Integrationswerkzeug initialisierenden Code zu der `onCreate`-Methode der Main Activity, wie sie sich aus der `AndroidManifest.xml` ergibt, hinzugefügt. Nachdem die `libshield.so` geladen wurde, überprüft sie zunächst, ob die App in einem Emulator ausgeführt wird und

beendet den Prozess gegebenenfalls. Die Native Library benötigt insgesamt drei Konfigurationsdateien, die während des Integrationsprozesses erstellt und schließlich verschlüsselt in dem `assets`-Ordner der APK abgelegt wurden. Im Folgenden wird jede der Dateien entschlüsselt und geparst, also in entsprechende Datenstrukturen innerhalb der `libshield.so` überführt. Der Inhalt und Zweck der einzelnen Konfigurationsdateien wird im Folgenden noch erläutert. Die konfigurierten Funktionen werden durch eine Reihe von Threads realisiert, die die Umgebung periodisch prüfen und bei Unregelmäßigkeiten entsprechend der Konfiguration mit einer Terminierung der App reagieren. Für diesen Fall kann der Kunde ebenfalls noch eine URL in der Konfiguration angeben, die kurz vor Beendigung des Prozesses im Webbrowser geöffnet wird. Dort kann der Hersteller der App den Endkunden über das Verhalten aufklären.

4.2.2 Interna der `libshield.so`

Die `libshield.so` ist das Herzstück des P Shields. Es nutzt eine Reihe von kryptographischen Primitiven, um die Native Library selbst, aber auch die Konfigurationsdateien sowie die Verschlüsselung durch die `SecureStorage`-Klasse zu schützen. Abbildung 4.2 zeigt eine Übersicht und stellt die Entschlüsselung der Datei `pbi.bin` exemplarisch dar. Die anderen von P verarbeiteten Dateien werden jedoch auf ähnliche Art und Weise behandelt.

Programmabbild. Die Verschlüsselung schützt mehrere Programmbereiche der `libshield.so`. Nach dem Laden des Programmabbilds in den Speicher sowie dem Auflösen aller Abhängigkeiten führt der dynamische Lader die in der `.init_array` spezifizierten Konstruktoren aus. Einer der Konstruktoren führt eine verschleierte Version des RC4-Verschlüsselungsverfahrens aus, um die `.rodata`-, `.text`- und `.ncd`-Sektionen zu entschlüsseln. Jede Sektion verwendet einen anderen Schlüssel, die mit einer neuen Version des P Shields wechseln.

Abhängigkeiten. Wie bereits erwähnt, erzeugt das Integrationswerkzeug drei unterschiedliche Konfigurationsdateien, die verschlüsselt vorliegen. Die Dateien bestehen dabei nicht nur aus dem Ciphertext: Die ersten 8 Bytes sind ein Initialisierungsvektor *iv*, während die letzten 64 Bytes eine Signatur *h* darstellen. Dazwischen befinden sich die verschlüsselten Daten *dat*. Die Initialisierungsroutine der `libshield.so` führt nun für jede der Dateien die folgenden Schritte aus:

1) Ein SHA512-Hash h' wird von *iv* und *dat* gebildet.

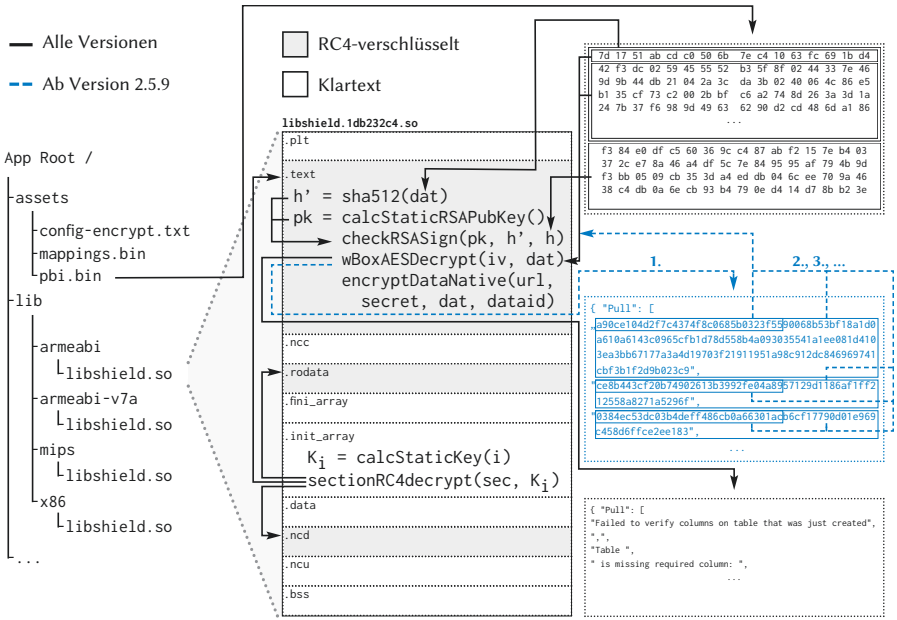


Abbildung 4.2: Kryptographische Funktionen zum Schutz der `libshield.so`.

2) Ein statischer, DER-enkodierter, öffentlicher RSA-Schlüssel *pk* wird extrahiert. Dabei wird die gleiche Kombination von arithmetischen Operationen und Schleifen verwendet, die alle anderen konstanten Strings verschleiert.

3) Eine Signaturvalidierung findet statt: nur falls $RSA_D(h', pk)$ und h inhaltsgleich sind, wird fortgefahren.

4) Die Daten *dat* werden mithilfe der `libshield.so` AES-Whitebox und des Initialisierungsvektors *iv* entschlüsselt. Die Whitebox fußt auf OpenSSL und einer modifizierten Implementierung der Blockchiffre, die den geheimen symmetrischen Schlüssel lediglich in einer expandierten Fassung verwendet. Es kommt AES-128 im CBC-Modus zum Einsatz.

5) Das entschlüsselte Ergebnis wird abschließend mithilfe der `inflate`-Funktion der `zlib`-Bibliothek dekomprimiert.

Dateiverschlüsselung. Die Kunden Ps können auch explizit von der `SecureStorage`-Klasse Gebrauch machen, die eine Verschlüsselung beliebiger Dateien erlaubt. Intern verwendet die Java-Klasse die native Funktion `encryptDataNative`. Neben den zu verschlüsselnden Daten (`dat`) verlangt die Funktion auch die URL eines P-kompatiblen HTTPS-Servers als Parameter (`url`) sowie zwei Byte-Strings. Ein Byte-String wird zufällig zur Laufzeit gewählt (`secret`), während der andere zur Identifikation der gespeicherten Daten dient (`dataid`).

`libshield.so` bildet zuerst eine HMAC von `dataid` und verwendet `secret` als HMAC-Schlüssel. Das Ergebnis wird wiederum in `dataid` abgelegt. Der Prozess wird insgesamt 4096 Mal wiederholt, bevor eine 16 Bytes lange Gerätekennung berechnet wird. Hierbei kommt wieder die Technik zum Einsatz, die zuvor für den öffentlichen RSA-Schlüssel beschrieben wurde.

Zusammen mit einer `protocol`- und `msg`-Variable sowie der Version des P Shields dienen die Gerätekennung und die `dataid` als HTTP-Form-Daten (`application/x-www-form-urlencoded`), die `libshield.so` über HTTPS (mit einem optionalem TLS-Client-Zertifikat) and `url` verschickt. Der Endpunkt antwortet mit einem Schlüssel, der spezifisch für alle Parameter ist. Dieser Schlüssel wird dann genutzt, um die in `dat` angegebenen Daten mit AES-256 im CBC-Modus zu chiffrieren.

4.2.3 Statisches Nomorp

Wir präsentieren nun *sNomorp*, ein Werkzeug, um das P Shield automatisiert aus einer beliebigen App zu entfernen. Die Native Library `libshield.so` implementiert das Gros der App-Härtungsmaßnahmen. Wenn sie entfernt wird, kommt fast keine der angebotenen Sicherungsmaßnahmen mehr zum Tragen. Dieser Umstand ist P bewusst, weshalb sie versuchen, eine engmaschige Verknüpfung zwischen der `libshield.so` und dem Java-Code der Client-App zu installieren. Die Verknüpfung beruht auf zwei Mechanismen: Externalisierung von Strings und Konstanten.

String-Externalisierung

Die String-Externalisierung ist in Abbildung 4.3 veranschaulicht und funktioniert wie folgt. Wenn das Integrationswerkzeug die App-Härtung in die Client-App einpflegt, macht sie alle Java-Strings im Kunden-Code ausfindig. Für jeden String wird ein Eintrag in einem Wörterbuch angelegt; ein Index, der linear wächst, fungiert

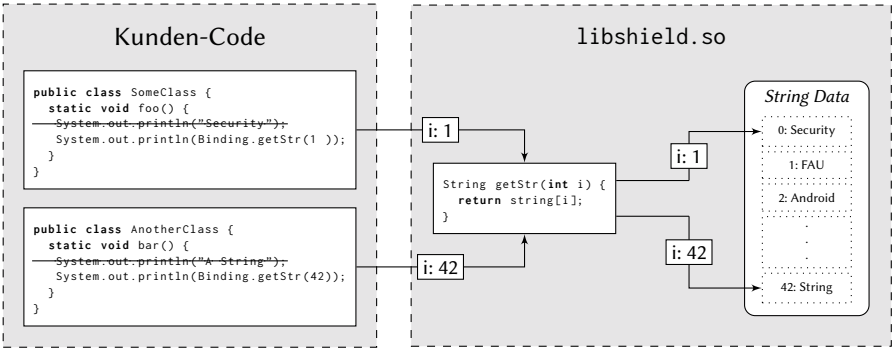


Abbildung 4.3: String-Externalisierung des P Shields.

als Schlüssel, der Java-String selbst als Wert. Nachdem das Index-String-Paar in das Wörterbuch eingetragen wurde, wird die String-Anweisung entfernt und durch einen Aufruf ersetzt, der in die native `libshield.so` zeigt. Die Funktion benötigt einen Integer als Argument, der einem Index in dem Wörterbuch entspricht und liefert den entsprechenden String zurück. Aufgrund ihrer Semantik haben wir die Funktion `getStr` genannt. Die String-Externalisierung sorgt also dafür, dass z. B. der Aufruf `System.out.println("Security")` zu `System.out.println(Binding.getStr(0))` wird, wobei `Binding` als Brücke zwischen Java und nativem Code fungiert.

Konstanten-Externalisierung

Neben den Java-Strings externalisiert das P Shield auch die Java-Konstanten der Client-App. Das funktioniert in Anlehnung an Abbildung 4.4 wie folgt:

1) Das P Integrationswerkzeug ersetzt jedes Feld einer Klasse, das als `static` oder `final` deklariert wurde, mit einem zufälligen Wert des gleichen Typs. Die ersetzten Werte werden in einem zweifach geschachtelten Wörterbuch gespeichert: In der ersten Ebene fungieren die vollständigen Klassennamen (inklusive Paketname) als Schlüssel und die Werte sind wiederum selbst ein Wörterbuch, das den Feldnamen der Klasse dem ursprünglichen Feldwert zuordnet. Um diese Werte wiederherzustellen, bevor das erste Mal auf die betreffende Klasse zugegriffen wird, installiert das P Shield einen statischen

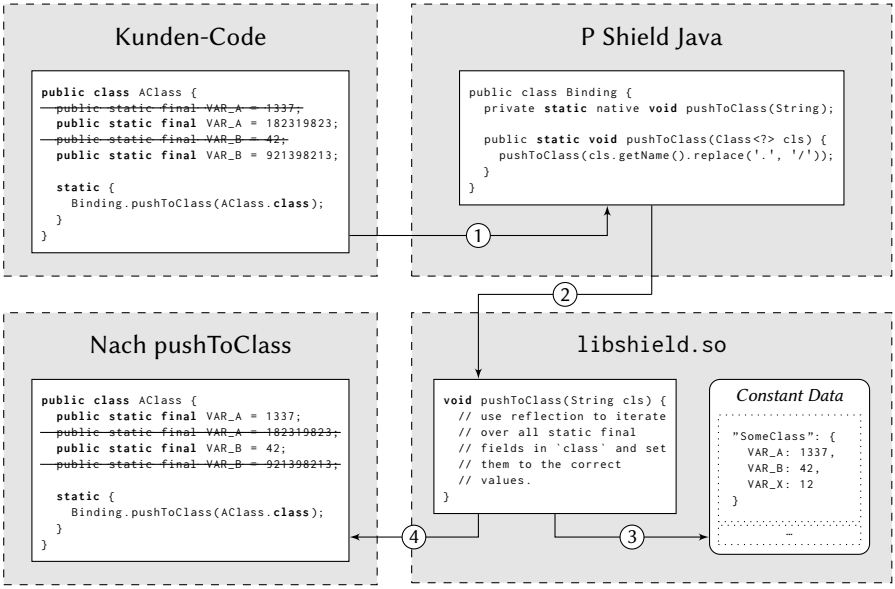


Abbildung 4.4: Konstanten-Externalisierung des P Shields.

Konstruktor in der betreffenden Klasse. In dem Konstruktor findet sich ein Aufruf an den Java-Brückencode des P Shields, der das `Class`-Objekt der Klasse als Argument trägt. Die Android-Laufzeitumgebung garantiert, dass der statische Konstruktor automatisiert ausgeführt wird, bevor der erste dynamische Zugriff auf die Klasse erfolgt.

2) Der Java Wrapper-Code des P Shields ersetzt die Punkte in der Rückgabe von `Class.getName()` mit Schrägstrichen und ruft die zugehörige native Funktion in der `libshield.so` auf.

3) Die `libshield.so` liest das zur Klasse gehörende Wörterbuch aus.

4) Im letzten Schritt ersetzt der native Code die Zufallswerte in der Java-Klasse der Client-App durch die ursprünglichen Werte. Wir nennen die Funktion aufgrund ihrer Funktionsweise `pushToClass`. Es ist erwähnenswert, dass es die Java VM nicht erlaubt, Felder zur Laufzeit zu ändern, die als `final` deklariert

wurden. Das Java Native Interface (JNI), von dem die `pushToClass`-Funktion Gebrauch macht, hat diese Beschränkung jedoch nicht.

Umschreiben der Client-App

Nachdem wir nun wissen, wie die Verknüpfung zwischen dem Java-Code der Client-App und dem P Shield funktioniert, müssen wir die App so modifizieren, dass die Strings und Konstanten wieder Bestandteil der App sind. Hierfür benötigen wir neben den entsprechenden Wörterbüchern aber einen verlässlichen Ansatzpunkt für die Transformationen. Bei der Analyse verschiedener Apps ist uns aufgefallen, dass der Name des Pakets, in dem P seine Java-Klassen ablegt, von App zu App variiert. Das Gleiche gilt für die Namen der Klassen und Methoden. Dieser Umstand würde Heuristiken erfordern, um die Aufrufe zu `getStr` und `pushToClass` zu identifizieren. Die weitere Analyse hat jedoch gezeigt, dass sich nur der Java-Code des P Shields unterscheidet, nicht aber der der `libshield.so`; mehrere Apps nutzen eine `libshield.so` mit identischer Hashsumme.

Diese Information lässt unmittelbar zwei Schlussfolgerungen zu. Erstens liefert P die `libshield.so` bereits kompiliert an seine Kunden aus und es findet auch beim Kunden vor Ort keinerlei Modifikation mehr an der Native Library statt. Die fehlende Individualisierung der `libshield.so` erklärt auch die Existenz der zuvor beschriebenen Wörterbücher für `getStr` und `pushToClass`. Zweitens lässt sich daraus schließen, dass noch mindestens ein weiteres Wörterbuch existieren muss, das das Renaming des Java-Codes des P Shields abbildet. So ein Mapping muss existieren, weil die JNI-Methoden über `RegisterNatives` verfügbar gemacht werden. Dadurch ist es notwendig, dass der vollständige Name der als `native` deklarierten Methoden innerhalb der `libshield.so` bekannt ist.

Durch dynamische Analyse mittels *Frida* waren wir in der Lage, alle erforderlichen Wörterbücher und die Konfiguration im Klartext zu extrahieren. Hierfür war es ausreichend, `malloc`, `free` und `memset` zu instrumentalisieren. Bei jeder Ausführung von `malloc` wurde der zurückgegebene Pointer an eine interne Liste angefügt, die bei einem Aufruf von `free` komplett traversiert wurde. Wir waren überrascht, dass dieser triviale Ansatz zum Erfolg führte; wir gingen davon aus, dass die `libshield.so` eine eigene dynamische Speicherverwaltung implementiert oder eine bestehende statisch gelinkt hat. Dass sich dadurch eine Angriffsfläche bietet, scheint P sogar bewusst zu sein, da einige Speicherbereiche mittels `memset` gelöscht werden, bevor sie `free` freigibt. Um sicherzustellen, dass wir keinen Speicherbereich verpassen, traversieren

wir unsere Pointer-Liste auch bei jedem Aufruf von `memset`. Aus Performanzgründen haben wir eine eigene Native Library `libnomorp.so` erstellt, die die *Frida*-Hooks in C statt in JavaScript implementiert.

Um zu erreichen, dass unsere Native Library `libnomorp.so` als Erstes noch vor `libshield.so` geladen wird, verwenden wir *dexlib2* [Gru]. Mithilfe des Werkzeugs lässt sich DEX-Bytecode zuverlässig modifizieren. Wenn die App nun startet, wird unser Code noch vor dem des P Shields aktiv und erlaubt uns so die Mappings und die Konfigurationsdatei zu erlangen. Die Datei `config-encrypt.txt` beinhaltet die durch den Kunden definierte Konfigurationsdatei im CSV-Format. Die Wörterbücher liegen beide als JSON-Datei verschlüsselt vor: In der Datei `mapping.bin` ist das Renaming-Mapping enthalten, während die Datei `pbi.bin` die Zuordnungen für die String- und Konstanten-Externalisierung enthält.

Nachdem die notwendigen Mappings nun vorliegen, kann die App mit *dexlib2* so umgeschrieben werden, dass alle Strings und Konstanten wieder an den ursprünglichen Stellen zu finden sind. Neben den genannten nimmt *sNomorp* noch kleinere Modifikationen an den Apps vor, um das P Shield komplett zu entfernen. Dazu zählt unter anderem das Umschreiben von Aufrufen der `PromonHttpUrlConnection` zur regulären `HttpUrlConnection` oder das statische Einbetten eines eventuell in der `config-encrypt.txt` definierten TLS-Client-Zertifikats.

Bewertung

Mithilfe von *sNomorp* kann ein Angreifer eine Version der App erstellen, die einfacher statisch zu analysieren ist. Der Prozess ist komplett automatisiert und erfordert nicht länger als fünf Minuten vom Start des Downloads der App bis zur Ausgabe der umgeschriebenen App. Obwohl es nicht im Fokus lag, ist der Großteil der Apps nach Anwendung von *sNomorp* vollständig lauffähig. Dadurch wird die korrekte Arbeitsweise unseres Ansatzes unterstrichen.

Während unserer groß angelegten Analyse ist uns jedoch aufgefallen, dass neun Apps eine andere Version des P Shields verwenden, die das Gerät aus der `libshield.so` heraus mit dem Backend registriert. Darüber hinaus sind auch komplette Geschäftsprozesse, wie z. B. das Ausführen einer Transaktion innerhalb einer Banking-App, auf diese Weise realisiert. Dadurch führt das Entfernen der `libshield.so` auch dazu, dass große Teile der HTTP-Kommunikation nicht mehr funktionsfähig

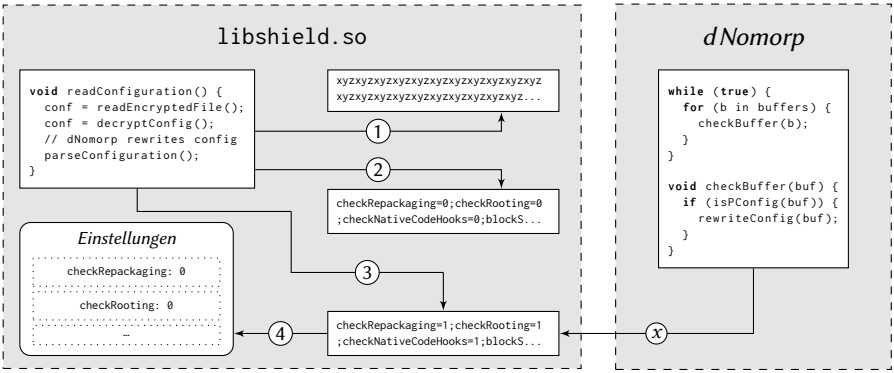


Abbildung 4.5: Umschreiben der Konfiguration des P Shields zur Laufzeit.

sind. Aus diesem Grund ist *sNomorp* für diese Apps weniger hilfreich. Die dynamische Analyse der Kommunikation wird aber mit dem Einsatz von *dNomorp* möglich, das im Folgenden beschrieben wird.

4.2.4 Dynamisches Nomorp

Während *sNomorp* gezeigt hat, wie sich das P Shield statisch von einer App entfernen lässt, beschäftigt sich *dNomorp* mit einem dynamischen Ansatz. Anstatt die `libshield.so` zu entfernen, behalten wir das P Shield vollkommen bei, deaktivieren aber alle Sicherungsfunktionen zur Laufzeit.

Um den Kontrollfluss der App dynamisch zu beeinflussen, benötigen wir erneut unsere eigene Native Library `libnomorp.so`. Statt wie zuvor die Mappings und die Konfigurationsdatei zu extrahieren, wird letztere zur Laufzeit umgeschrieben, sobald sie sich entschlüsselt im Speicher befindet. In der Konfigurationsdatei haben die meisten Schutzfunktionen drei Einträge: zwei binäre Schalter, die anzeigen, ob eine Funktion aktiviert ist und ob ein Befund zum Absturz führen soll; der dritte Wert ist optional und kann eine Prozent-enkodierte URL beinhalten, die bei einem bewusst herbeigeführten Absturz die angegebene URL im Webbrowser öffnet. Für die Repackaging-Erkennung könnten die Einträge z. B. wie folgt aussehen: `checkRepackaging=1`, `exitOnRepackaging=0` und `exitOnRepackagingURL=https%3A%2F%2Ffau.de`

Ziel von *dNomorp* ist es, alle Schutzmaßnahmen dadurch zu deaktivieren, dass alle Schalter von `1` auf `0` gesetzt werden. Dieser Schritt muss vor dem Parsen der Klartext-Konfigurationsdatei und nach dem Entschlüsseln der `config-encrypt.txt` erfolgen. Der reguläre Vorgang findet innerhalb der `libshield.so` in vier Schritten statt und ist in Abbildung 4.5 dargestellt:

1) Die verschlüsselte Konfigurationsdatei `config-encrypt.txt` wird in einen `malloc`-allokierten Puffer eingelesen.
2) Mithilfe der P AES-Whitebox wird die Datei entschlüsselt. Das Resultat wird in einen weiteren `malloc`-Buffer geschrieben.
3) Die Konfigurationsdatei wird im Klartext gelesen.
4) Die Daten werden in interne Datenstrukturen überführt.

Gleich nachdem die Konfigurationsdatei in 2) entschlüsselt wurde, wird sie mittels `memcopy` in einen anderen Puffer kopiert, bevor die `libshield.so` anfängt, den entschlüsselten Inhalt in Schritt 3) zu lesen. Aus diesem Grund haben wir den Aufruf von `memcopy` instrumentalisiert, um so Zugriff auf den Puffer zu haben, bevor er in den neuen kopiert wird. Dadurch können wir die Konfigurationsdatei im Klartext beliebig modifizieren (Schritt *x* in Abb. 4.5). Um den richtigen Moment zu erkennen, durchsuchen wir den Puffer nach bekannten Schlüsselwerten, wie `checkDebugger` und `checkRooting`.

Bewertung

Im Gegensatz zu unserem statischen Angriff ist das dynamische Umschreiben der Konfigurationsdatei unmittelbar eingängig. Wir machen uns den Umstand zunutze, dass das P Shield ein kommerzielles Produkt ist, das möglichst ohne Anpassungen für alle Kunden angeboten werden kann. Statt alle Funktionen einzeln zu deaktivieren, verwenden wir bei *dNomorp* einfach die dafür vorgesehene Konfigurationsdatei.

Dieser Ansatz funktioniert zuverlässig für alle 31 Apps, die wir als Kunden des P Shields identifiziert haben. Der Nachteil des Vorgehens ist, dass sich daraus keinerlei Hilfestellung für die statische Analyse ergeben. Ein zentraler Vorteil ist hingegen, dass alle Apps nach wie vor voll funktionsfähig sind, weil weiterhin Zugriff auf Interna der `libshield.so`, wie z. B. die Whitebox-Implementierung, existiert. Da alle Schutzfunktionen des P Shields erfolgreich ausgeschalten wurden, kann ein Angreifer die App beliebig verändern (z. B. Schadcode hinzufügen) oder sie

dynamisch analysieren (z. B. den HTTPS-Verkehr über einen Man-in-the-Middle-Angriff).

4.2.5 Bekanntmachung und Reaktion

Die Bekanntmachung unserer Forschungsergebnisse zur Sicherheit der Lösung P Shield sorgte für eine breite Reaktion in der Öffentlichkeit, durch eine Bundesbehörde und durch P selbst. Das Unternehmen reagierte nicht nur mit technischen, sondern auch rechtlichen Maßnahmen.

Disclosure. Die Kommunikation unserer Angriffe gegenüber P und der einsetzenden Banken erfolgte in enger Zusammenarbeit mit Hakan Tanriverdi, einem Journalisten der Süddeutschen Zeitung. Er stellte den Kontakt mit P her und bat sowohl das Unternehmen als auch die betroffenen Institute der Deutschen Kreditwirtschaft um Stellungnahme. Letztendlich publizierte Herr Tanriverdi am 24. November 2017 einen Artikel im Wirtschaftsteil der Süddeutschen Zeitung, der die Angriffsfläche abstrakt ohne Nennung von Details darstellt [Tan17]. Wir haben uns bewusst dazu entschieden, *Nomorp* nicht zu veröffentlichen, um Kriminellen nicht in die Hände zu spielen. Technische Eckpfeiler zu unserem Vorgehen wurden erst auf dem 34. Chaos Communication Congress (34c3) vorgestellt. Dadurch wurde P die Gelegenheit gegeben, auf unsere Angriffsvektoren zu reagieren, die auch die Deutsche Kreditwirtschaft in ihrer Stellungnahme ankündigte [DK17]. Die Umgesetzten technischen Neuerungen sind unten ausgeführt.

BSI Lagebericht IT-Sicherheit 2018. Sowohl der Artikel in der Süddeutschen Zeitung als auch der Vortrag auf dem 34c3 wurden von einer breiten Berichterstattung begleitet. Letztendlich griff auch das Bundesamt für Sicherheit in der Informationstechnik (BSI) unsere Ergebnisse in seinem Jahresbericht zur Lage der IT-Sicherheit in Deutschland 2018 auf [BSI18, S. 19]. Als Reaktion hat das BSI ein Projekt gestartet, das die Sicherheit von Online- und Mobilebanking-Anwendungen bewerten und Handlungsempfehlungen aussprechen soll. Das BSI folgt auch unserer Empfehlung, bei digitalen Bankgeschäften auf eine physische Trennung von Transaktionsauslösung und -bestätigung zu achten. Eine software-basierte App-Härtung ist demnach unzureichend.

Rechtsstreit. Im Juni 2018 präsentierten wir unsere Forschung im Rahmen der 15. Jahrestagung der akademischen Konferenz „Detection of Intrusions and Malware, and Vulnerability Assessment“. Der dazugehörige Beitrag „Honey, I Shrunk

Your App Security: The State of Android App Hardening“ wurde durch Springer veröffentlicht [Hau+18]. Zur gleichen Zeit kündigte auch die Technische Universität München eine Publikation an, die den Schlüssel der Whitebox-Implementierung des P Shields automatisiert extrahiert. P versuchte letztendlich mit Rechtsmitteln gegen die Forschung an ihrem Produkt vorzugehen und eröffnete im Juli einen Rechtsstreit, der acht Wissenschaftler von vier Universitäten vor das Landgericht Nürnberg-Fürth zwang [Erm18]. Der Rechtsstreit wurde zwar in weiten Teilen zu unseren Gunsten in einem Vergleich beigelegt, strahlt in seiner Bedeutung aber über den Fall hinaus und stellt ganz generell die Frage nach der Rechtssicherheit bei Sicherheitsforschungen. Darüber hinaus kann das rechtliche Vorgehen Ps durchaus als Teilerfolg für die Klägerin gewertet werden, da die Forscher der Technischen Universität München ihre Publikation noch vor der Gerichtsverhandlung zurückzogen. Der Vorfall wurde auch in einem Vortrag auf dem 35. Chaos Communication Congress durch Dominik Maier und Fabian Franzen thematisiert, die zwei der acht Antragsgegner waren [MF18].

P Shield ab Version 2.5.9. Als Reaktion auf unsere Angriffe hat P einige Gegenmaßnahmen innerhalb der `libshield.so` ergriffen. Nach bestem Wissen wurden die Änderungen in Version 2.5.9 eingeführt und beziehen sich alle auf die Art und Weise, wie P die Konfigurationsdatei `config-encrypt.txt` sowie die Push- und Pull-Wörterbücher in der `pbi.bin` einbezieht.

Als erstes verwendet P Shield nun zwei Verschlüsselungsschichten der AES-Whitebox für beide Dateien, um es zu verhindern, dass der Ort der Konfigurationsdatei durch leicht identifizierbare Strings offengelegt werden kann. Zu diesem Zweck wurden alle Schlüssel durch eine Kombination von 16 + 4 Bytes Kennungen ersetzt. Zweitens wurden die Konfigurationsparameter, die angeben, ob bestimmte Laufzeitüberprüfungen wie `checkRooting` durchgeführt werden sollen, entfernt und sind nun direkt in die `libshield.so` einkompiliert.

Insgesamt zielen diese Modifikationen darauf ab, dass *Nomorp* in seiner beschriebenen Arbeitsweise nicht mehr lauffähig ist. Weder *sNomorp* noch *dNomorp* lassen sich für aktuelle Versionen des P Shields anwenden und es ist offen, welche Anpassungen notwendig wären, damit unsere Angriffe wieder funktionieren. Um den Aufwand für einen Angreifer signifikant in die Höhe zu treiben, schlagen wir eine Reihe substanzieller Verbesserungen vor, die wir im Folgenden diskutieren.

4.3 Diskussion

Unsere Fallstudie zum P Shield hat Lösungen zur App-Härtung Grenzen aufgezeigt. Dennoch stellt unsere Analyse die Sinnhaftigkeit von Schutz- und Härtungsmaßnahmen nicht generell infrage. Es ist wichtig und richtig insbesondere kritische Apps zusätzlich zu schützen. Im Folgenden wollen wir Schutztechniken vorstellen, die RASP-Hersteller zur Verbesserung ihrer Produkte umsetzen sollten. Im Grundsatz bleibt es jedoch dabei, dass der Angreifer auch nach der Umsetzung der vorgestellten Maßnahmen im Vorteil bleibt. Deshalb sollten App-Entwickler zuerst den Einsatz einer unabhängigen 2FA in Erwägung ziehen. Gleichzeitig sollte eine vertrauenswürdige Serverkomponente in der Sicherheitsarchitektur eine möglichst große Rolle spielen.

Die Schutzziele der RASP-Anbieter implizieren, dass eine Analyse für einen Gegenspieler möglichst zeitaufwendig und teuer sein sollte. Außerdem sollten insbesondere automatisierbare Angriffe gegen die Apps der Kunden vermieden werden. Steigt der Wert der zu schützenden Inhalte, dann steigt auch die Motivation diese Schutzmaßnahmen zu brechen. Demnach kann der Schutz für eine App ausreichend sein, während er für eine andere App unzulänglich ist. Ähnlich verhält es sich mit der Anzahl der Endnutzer, die die App der RASP-Kunden einsetzt: Je mehr Endnutzer es gibt, desto wertvoller ist ein generalisierbarer Angriff. Daraus erwächst nicht nur die Anforderung, dass die RASP-Anbieter ihre Lösung kontinuierlich verbessern, sondern auch die, ihre Lösung möglichst diversifiziert einzupflegen. Die Art und Weise, wie der Schutz platziert wird, sollte sich von Kunde zu Kunde und von App zu App unterscheiden. Außerdem sollte der Programmcode der Schutzmaßnahmen des RASP-Anbieters möglichst stark mit dem Programmcode der Kunden-App verwoben werden.

Vermeidung zentraler Konfigurationen. *Nomorp* war gerade deshalb so erfolgreich, weil es zentrale Konfigurations- und Mapping-Dateien gab. RASP-Lösungen sollten einen Ansatz vermeiden, der eine kundenspezifische Konfiguration erst zur Laufzeit vornimmt. Die Konfiguration sollte vor der Anwendung durch den Kunden stattfinden und nur die konfigurierten Maßnahmen direkt in den Kunden-Code einarbeiten. Diese Schutztechniken dürfen sich nicht an einer zentralen Stelle deaktivieren oder entfernen lassen. Die Wörterbücher für die String- und Konstanten-Externalisierung sowie für das Renaming sollten nicht unabhängig abgelegt, sondern direkt einkompiliert werden. Um es zu vermeiden, dass ein Angreifer die Mappings für Strings und Konstanten durch dynamisches Iterieren erstellt, könnte die RASP-

Lösung die nächsten Java-Anweisungen in den Native Code einbetten, statt direkt den Wert zurückzugeben [PKM15]. Zu diesem Zweck könnte für jede Konstante eine native Funktion erzeugt werden, die zusätzliche Parameter empfängt und diese an die eingebetteten Funktionsaufrufe, die die betreffende Funktion verwenden, weitergibt.

Anti-Tampering. Maßnahmen zum Schutz des RASP- und des Kunden-Codes sind bereits großflächig im Einsatz. Idealerweise wird die Integrität verschiedener Code-Blöcke mehrmals an unterschiedlichen Stellen durch verschiedene Techniken sichergestellt. Die Code-Blöcke sollten ihre Integritätsmaßnahmen verketten, damit es schwieriger wird, den Programmcode zu verändern.

Anti-Hooking & Anti-Debugging. Die Ansätze des Anti-Hookings und -Debuggings sind verwandt. Ein Angreifer kann die Aufgaben eines Debuggers, wie das Lesen von Speicherbereichen, durch Hooks realisieren. Umgekehrt ist das Einsetzen von Hooks trivial, wenn die App Verbindungen durch einen Debugger erlaubt. Anti-Hooking-Maßnahmen für bestimmte Ansätze und Werkzeuge, wie z. B. `LD_PRELOAD`, *Xposed*, *Cydia Substrate* oder *Frida*, existieren bereits in den RASP-Produkten. Die Erkennung dieser Werkzeuge lässt sich durch Umbenennen jedoch leicht umgehen. Aus diesem Grund sollten RASP-Anbieter Maßnahmen implementieren, die nicht spezifisch die Ansätze und Werkzeuge erkennen, sondern die zugrundeliegenden Techniken. Diese Erkennungsroutinen sollten so eng wie möglich mit dem eigentlichen Kunden-Code verwoben sein und wiederkehrend zu zufälligen Zeitpunkten ausgelöst werden. Sollte die Integrität der App gebrochen werden, dann sollte die App keine Callbacks einsetzen, da diese über Instrumentalisierung einfach abgefangen werden. Stattdessen sollte sich die App beenden. Idealerweise erfolgt die Terminierung durch einen Crash, der noch dazu nicht auf die Stelle zurückzuführen ist, die tatsächlich für die bewusste Beendigung verantwortlich ist. Auf diese Weise wird die Analyse und die Isolation der entscheidenden Code-Stelle durch den Gegenspieler erschwert.

Anti-Emulator. Solange es möglich ist, die Kunden-App in einem Emulator zu betreiben, bleibt Hooking und Debugging möglich. Ein perfekter Emulator ist schwer umsetzbar, weshalb Emulator-Fingerprinting eingesetzt werden sollte, um die Analyse zu erschweren. Zu diesem Zweck können mehrere verschiedene Prüfungsroutinen an zufälligen Stellen im RASP-, aber auch im Kunden-Code platziert werden [MMP14]. Forschung zum automatisierten Entpacken von Schadsoftware auf Basis komplett emulierter Systeme legt nahe, dass Anti-Emulator-Techniken effektiv sind [Dua+18].

Gerätebindung. Um das Replizieren einer Kunden-App zu verhindern, sollte soweit wie möglich auf Hardwaremöglichkeiten zurückgegriffen werden. Inwiefern eine sichere Gerätebindung auf mobilen Endgeräten möglich ist, wird in Abschnitt 3.4.1 diskutiert.

Code Obfuscation. Die RASP-Lösung sollte nicht nur den eigenen Code, sondern vor allem auch den der Kunden-App verschleiern. Je umfangreicher der RASP- und Kunden-Code verwoben und obfuskiert sind, desto aufwendiger ist das Reverse Engineering. Das bedeutet, dass das Integrationswerkzeug des RASP-Anbieters sogar den Code von Drittanbieter-Bibliotheken einbeziehen sollte, um eine große Einheit zu erzeugen. Dadurch wird es einem Gegenspieler zusätzlich erschwert, relevante Stellen zu identifizieren. Das Einweben kann auf Ebene von nativen Bibliotheken, dem DEX-Bytecode oder zwischen einer nativen Bibliothek und Java (über JNI) erfolgen. Moderne Android-Packer-Techniken, wie DEX-Packing oder VM-basierte Obfuskierung, sollten in Betracht gezogen werden [ZLY15; Yan+15]. Darüber hinaus sollte die Ausgestaltung der Obfuskierung zur Kompilierzeit zufällig bestimmt werden. Dadurch lässt sich erreichen, dass jede neue Version unter Umständen deutlich anders aufgebaut und obfuskiert ist. In Konsequenz wird der Einsatz von Werkzeugen wie *Nomorp*, die ein automatisiertes Entfernen erlauben, signifikant erschwert. Manche der angesprochenen Schritte können es technisch notwendig machen, dass der RASP-Anbieter seine Lösung im Quelltext oder einer Zwischensprache ausliefert. Es ist also notwendig, dass der RASP-Anbieter das geistige Eigentum seiner Kunden höher wertet als das eigene.

Root-Erkennung. Um Reverse Engineering zu verlangsamen, kann eine Root-Erkennung helfen, da viele Analyse-Werkzeuge Root-Rechte voraussetzen. Eine Root-Erkennung kann jedoch nicht dazu beitragen, eine Rechteausweitung durch einen Angreifer zu erkennen. Die App-Entwickler müssen sich entscheiden, ob sie den Nutzerkreis, der seine Geräte bewusst mit Root-Zugriff ausstattet, von der Verwendung der App ausschließen wollen. Abhängig vom Einsatzzweck kann es auch ausreichend sein, den Nutzer lediglich über die Risiken zu unterrichten. In jeden Fall sollte sich die Root-Erkennung nicht nur darauf beschränken, charakteristische Dateien der Root-Lösungen zu erkennen; diese Maßnahmen sind leicht zu umgehen [Kel+19]. Stattdessen scheinen die beiden folgenden Ansätze besser geeignet:

1) *Google SafetyNet* bietet eine API, die es Apps erlaubt, die Integrität des Geräts zu überprüfen [Ngu+17]. Die Prüfroutinen werden von Google entwickelt

und regelmäßig dynamisch aktualisiert. Zusätzlich besitzt der Code von SafetyNet höhere Privilegien als eine reguläre App und kann infolgedessen auch Schutzmaßnahmen umsetzen, auf die eine RASP-Lösung keinen Zugriff hätte. Das aus der Verwendung der *SafetyNet*-API entstehende Resultat sollte nicht auf dem Gerät, sondern auf einem vertrauenswürdigen Server geprüft werden. Zu diesem Zweck wird das Ergebnis von Google signiert. Daraus ergibt sich, dass die Maßnahme auch eine Unterstützung im Backend der RASP- bzw. App-Anbieter voraussetzt.

2) Angriffe, die eine Rechteausweitung durchführen, können weder erkannt noch verhindert werden. Der einzige Weg, um die Wahrscheinlichkeit zu reduzieren, dass Endnutzer Opfer einer solchen Attacke werden, ist das Forcieren einer sinnvollen Mindestversion. Diese Version sollte direkt über das Angebot im App Store bestimmt werden. Um alte Versionen bei einem Update auszuschließen, sollten Änderungen an der API durchgeführt werden, statt einen leicht zu fälschenden Vergleich des Versionsstrings durchzuführen.

Anti-Keylogger & Anti-Screen Reader. Eine App-eigene Tastatur sowie das Zulassen vertrauenswürdiger Tastaturen sind sinnvolle Vorgehen. Es bleibt einem Angreifer jedoch möglich, die Anzeige der App zu überlagern [Fra+17; KBM18].

Secure Communication. RASP-Lösungen können TLS-Verbindungen automatisch mit Zertifikats-Pinning versehen. Ein Client-Zertifikat erhöht die Komplexität zusätzlich. Wenn Client- und Server-Zertifikat obfuskiert und an sich wechselnden Positionen vorliegt, steigt der Aufwand, den ein Angreifer für Man-in-the-Middle-Angriffe investieren muss. Dadurch wird es auch erschwert, Reverse Engineering der API-Kommunikation zu betreiben.

Datenverschlüsselung. Das verschlüsselte Ablegen von Daten macht es für einen Angreifer substanziell schwerer, an diese Informationen zu gelangen. Auf neueren Geräten ist es möglich, das Schlüsselmaterial sicher innerhalb eines speziell gesicherten Hardwarebereichs abzulegen. Manche Geräte ab Android 7 (Nougat) erlauben es über Key Attestation sogar, die Erstellung in Hardware auf Serverseite sicherzustellen. Dennoch hat ein Angreifer, der bereits vor der Erstellung des Schlüsselmaterials auf dem Gerät ist, die Möglichkeit, die entsprechenden API-Aufrufe zu instrumentalisieren und somit eigenes Schlüsselmaterial zu erstellen. Auf älteren Geräten kann der Einsatz von Whitebox-Kryptographie sinnvoll sein. Um zu verhindern, dass ein Angreifer die Whitebox-Implementierung als Ganzes extrahiert und verwendet, sollte der Code der Whitebox möglichst verwoben sein. Auch die

Einsprungspunkte der Whitebox sollten möglichst verschleiert und vielfältig sein. Die Whitebox sowie die sie schützenden Maßnahmen sollten regelmäßig wechseln. Eine vorherige Version sollte durch den Server nur für einen begrenzten Zeitraum akzeptiert werden.

4.4 Fazit

Insbesondere bei den Apps deutscher Banken ist der Einsatz softwarebasierte Härtungsmaßnahmen durch Drittanbieter beliebt. Im Rahmen von Forschungsfrage 2 (Angriffsfläche Mobilebanking) haben wir uns in diesem Kapitel mit den Schutzmaßnahmen beschäftigt, die solche Lösungen offerieren. Zwar implementieren die verschiedenen Hersteller eine Vielzahl an Sicherungs- und Härtungsmaßnahmen, sind in ihren Möglichkeiten aber letztendlich limitiert. Das Prinzip des Selbstschutzes der App impliziert bereits, dass auch der Programmcode der App-Härtungslösung im gleichen Rechtekontext ausgeführt wird, wie die App selbst. Hierdurch ergibt sich ein grundsätzlicher Unterschied zu Virenscannern auf PC-Systemen, die mit Administratorrechten betrieben werden und somit einen umfangreicheren Schutz bieten können. Zusätzlich sind die Maßnahmen zumeist vollständig in Software implementiert, weshalb sich systembedingt ein geringeres Schutzniveau vor Schadsoftware ergibt. Der Hauptbeitrag, den App-Härtungslösungen zum Schutz einer App bieten können, ergibt sich durch einen höheren Individualaufwand, den ein Angreifer leisten muss.

Unsere Fallstudie zum P Shield hat jedoch gezeigt, dass ein Angreifer nicht zwangsläufig für jede App, die mit derselben Lösung geschützt ist, erneut hohe Kosten hat. Das liegt darin begründet, dass P seine Lösung uniform für alle Apps anwendet. Eine Individualisierung findet nur in sehr beschränktem Umfang statt. Gepaart mit dem Umstand, dass P eine vorherrschende Stellung am Markt der deutschen Banking-Apps und App-basierten Sicherungsverfahren hat, ergibt sich für einen automatisierten Angriff ein attraktives Ziel. Wir haben mit *Nomorp* gezeigt, dass der durch das P Shield umgesetzte Schutz in 31 Apps auf gleiche Art und Weise deaktiviert oder sogar umgekehrt werden kann.

Es bleibt festzuhalten, dass App-Härtung ein sinnvoller Baustein im Sicherungskonzept einer Bank sein kann. Den versprochenen vollumfänglichen Schutz können die Produkte jedoch nicht leisten. Dadurch ergibt sich insbesondere im Kontext des Online- und Mobilebankings kein Schutzniveau, das mit dem einer unabhängigen

2FA zu vergleichen wäre, wie sie z. B. das chipTAN-Verfahren bietet. Obwohl für die Zukunft auch auf Smartphones und Tablets eine adäquate Absicherung von Transaktionen insbesondere im Mobilebanking möglich scheint, spielt die Bedeutung der softwarebasierten App-Härtung für ein Gelingen eine untergeordnete Rolle. Zentral sind hierfür die in Abschnitt 3.4 dargestellten Hardwarefunktionen zur Gewährleistung einer sicheren Gerätebindung und Anzeige.

5

Fintech-Sicherheit am Beispiel N26

Was ist ein Einbruch in eine Bank gegen die Gründung einer Bank?

— Die Dreigroschenoper, 1928

Während sich die beiden vorangehenden Kapitel mit der Sicherheit im Mobilebanking im Allgemeinen und den App-basierten Sicherungsverfahren etablierter Geldhäuser im Speziellen beschäftigt haben, steht im Folgenden die Sicherheit bei einem neuen Marktteilnehmer im Vordergrund: N26. Das Start-up aus Berlin gilt als Aushängeschild der deutschen Fintech-Szene und zählt in diesem Bereich zu den erfolgreichsten Unternehmen. N26 ist eine Direktbank, die ihren Kunden seit Anfang 2015 ein Girokonto bietet, das komplett vom Smartphone aus verwaltet werden kann. Transaktionen werden über eine 1AA freigegeben. Das junge Unternehmen ist mit seiner Banklizenz europaweit aktiv und zählt bei einer Unternehmensbewertung von 2,3 Milliarden Euro zum Januar 2019 bereits 2,3 Millionen Kunden [Nes19]; jeden Tag kamen im März 2019 rund 10 000 Neukunden hinzu [SE19]. Für das Jahr 2019 kündigte das wertvollste deutsche Fintech zudem seinen Markteintritt in den USA und Brasilien an.

Wir zeigen durch das Aufdecken ernstzunehmender Sicherheitslücken, dass der schnelle Aufstieg auch eine Schattenseite hat. Nachfolgend stellen wir einzelne Angriffsvektoren im Front- und Backend vor, die in Kombination zu einer Kontoübernahme im großen Stil hätten genutzt werden können. Um die Kunden zu schützen, haben wir unsere Funde vor einer Veröffentlichung an N26 gemeldet.

5.1 Hintergrund und Forschungsstand

Im Folgenden definieren wir den Begriff Fintech und beschreiben den Untersuchungsbedarf am Stand der Forschung.

Definition. Das Wort Fintech steht für „Financial Technology“ und ist ein Sammelbegriff für technologiegetriebene Innovationen im Finanzsektor. Da die Ideen und deren Umsetzungen zumeist durch Start-ups erfolgen, werden diese Unternehmen heutzutage selbst als Fintechs bezeichnet. Die Geschäftsfelder der Fintechs umfassen unter anderem die Bereiche Finanzierung, Vermögensanlage und Zahlungsverkehr [DH18a]. Zudem sind das Smartphone des Kunden und die Kommunikation über das Internet zentral [Dor+17]. Der Bereich gilt als relativ neu und gewinnt erst nach 2010 an Bedeutung: die Börsen-Zeitung [Hip14], das Handelsblatt [Slo14], die Frankfurter Allgemeine Zeitung [Brü14] und die Süddeutsche Zeitung [Hof14] erwähnen Fintechs erstmals im Jahr 2014.

Vertrauen in Fintechs. Obwohl die Begriffe Fintech und Bedrohung des Öfteren gemeinsam genannt werden, ist damit in den meisten Fällen das „mögliche Bedrohungspotenzial aus Sicht tradierter Finanzdienstleister“ [RES18, S. 1] gemeint. Arbeiten, die sich mit den Sicherheits- und Datenschutzimplikationen der konkreten Implementierung der Fintechs auseinandersetzen, sind aufgrund des jungen Marktes aber noch die Ausnahme. Nach unserem Kenntnisstand ist unsere Sicherheitsanalyse zu N26 die erste, die sich systematisch und fundiert mit der Sicherheit eines Fintechs oder eines verwandten Start-ups auseinandersetzt. Dass es hierzu jedoch deutlichen Klärungsbedarf gibt, zeigen Nutzerbefragungen mit deutschen Bankkunden. Eine repräsentative Umfrage der Gesellschaft für Konsumforschung im Auftrag des Bundesverband deutscher Banken (BDB) kommt 2017 zu dem Ergebnis, dass die etablierten Institute gegenüber den Fintechs einen erheblichen Vertrauensvorschuss genießen: „Während sechs von zehn Deutschen (61%) davon überzeugt sind, dass Kundendaten bei Banken vor dem missbräuchlichen Zugriff Dritter gut oder sehr gut geschützt sind, glauben das von den jungen FinTech-Unternehmen nur 17%“ [BG17]. Lediglich das Vertrauen gegenüber Finanzdienstleistungen der etablierten US-Technologiekonzerne wie Google, Amazon oder Facebook ist mit 10% noch geringer. Vergleichbare Schlussfolgerungen zieht Pricewaterhouse Coopers im gleichen Jahr nach einer eigenen repräsentativen Befragung. Zwar seien die Produkte der Fintechs durchaus attraktiv, die Kunden wünschten sich diese aufgrund von Sicherheitsbedenken aber lieber von der eigenen Bank in Kooperation mit einem Fintech [Kra17].

Datenschutz. Zwei wissenschaftliche Gutachten von Dorfleitner und Hornuf setzten sich 2018 mit der Fragestellung auseinander, ob und in welchem Umfang Fintechs Daten ihrer Nutzer durch ihren Internetauftritt erfassen [DH18b] und inwiefern deren Verhalten konform zu den Anforderungen der Datenschutz-Grundverordnung (DSGVO) ist [DH18a]. Zu diesem Zweck werteten sie die Datenschutzerklärungen von 505 deutschen Fintechs aus. Sie stellten dabei fest, dass 130 Fintechs keine oder eine zum Erhebungszeitpunkt zumindest nicht verfügbare Datenschutzerklärung hatten. Durch die DSGVO sind zwar mit 80% viele Datenschutzerklärungen angepasst worden; sie hätten sich aus Nutzersicht aber kaum gebessert und teils sogar verschlechtert. So nahmen die Datenschutzerklärungen im Umfang deutlich zu, ließen aber weiter eine transparente und vollständige Offenlegung der durch Dritte verarbeiteten personenbezogenen Daten vermissen. Eine bedeutende Einschränkung der Arbeit ist jedoch, dass sich die Analysen der Datenschutzerklärung nur auf die Webseite beschränken und die vielfach zentraleren Apps der Fintechs außen vor lassen.

Mit dem Bereich der Fintechs verwandt sind Unternehmen aus dem Bereich „Insurance Technology“, sogenannte Insurtechs. Wie die Fintechs haben sie den Anspruch, durch den Einsatz moderner Technologien Innovationen zu schaffen, die aber nicht in den Bereich der Finanzen, sondern in den der Versicherungen fallen. In diesem Bereich sind Anwendungen, die die Gesundheitsdaten ihrer Nutzer verarbeiten, besonders kritisch. Aus diesem Grund ist es leicht nachvollziehbar, dass bei der Konzeption und Implementierung besondere Vorsicht gewahrt werden sollte. Der Informatiker und Datenschutzexperte Mike Kuketz identifizierte 2018 kurz nach Erscheinen der Insurtech-App Vivy Mängel beim konkreten Datensendeverhalten [Kuk18]. Demnach macht die App – die über 10 Millionen Versicherten die Verwaltung ihrer Gesundheitsdaten erlaubt – reichen Gebrauch von Telemetriediensten (sog. Trackern). Dabei würden auch schon Daten an Dritte übergeben, bevor der Nutzer die Gelegenheit hat, diesem Verhalten zuzustimmen oder es abzulehnen.

Sicherheit. Wenig später beschäftigt sich auch Martin Tschirsich mit dieser und weiteren Lösungen zur elektronischen Gesundheitsakte [Tsc18]. In seinen Analysen entdeckte er mehrere zum Teil schwerwiegende Sicherheitslücken in den Implementierungen der Apps und Portale. Die Probleme sieht er bei Start-ups auch darin begründet, dass diese oft kein Personal haben, das explizit mit Sicherheit beauftragt ist. Er stellt außerdem fest, dass Sicherheit als Wettbewerbsnachteil wahrgenommen wird.

5.2 Sicherheitsdefizite

Obwohl die Entwicklung intuitiver und einfach verwendbarer Lösungen fraglos erstrebenswert ist, ist das Vertrauen der Kunden in die Bank das höchste Gut. In der Vergangenheit kam bei Banken ein eher konservativer Ansatz zum Tragen, der die Sicherheit an erste Stelle setzte. Daraus ergab sich ein stetiger Zuwachs in Sachen Transaktionssicherheit, der zum Teil jedoch auf Kosten der Benutzerfreundlichkeit ging. Kapitel 3 hat gezeigt, dass heute auch das Gros der alteingesessenen Kreditinstitute App-basierte Sicherungsverfahren und Mobilebanking anbietet. N26 war jedoch das Unternehmen, das als Erstes eine 1AA für alle Android- und iOS-Geräte anbot. Der Erfolg des Fintechs setzt die etablierten Institute unter Druck und sorgt für eine Abwärtsspirale konzeptionell wichtiger Sicherheitseigenschaften. Aus diesem Grund ist es umso wichtiger, dass die Sicherheit der Implementierung rigoros überprüft wird, damit sich neben der konzeptionellen zumindest keine technische Angriffsfläche bietet.

Die folgenden Unterabschnitte belegen jedoch, dass weder das Backend, noch die Android- oder iOS-App von N26 als sicher beschrieben werden konnten. Stattdessen legt die Anzahl und Schwere der identifizierten Probleme nahe, dass Sicherheit zu keiner Zeit die Priorität in dem erfolgsverwöhnten Fintech war.

Die Beschreibung der Architektur von N26 sowie die der Sicherheitsmängel stellen den Stand vom Dezember 2016 dar. Es ist davon auszugehen, dass seitdem Änderungen eingeführt wurden, die bereits zum Druckzeitpunkt dieser Dissertation keine Gültigkeit mehr besaßen.

5.2.1 Architektur

Ehe wir die einzelnen Schwachstellen präsentieren, geben wir einen Überblick über die Sicherheitsarchitektur von N26. Im Gegensatz zu anderen Systemen im Online- und Mobilebanking setzt N26 auf mehr als zwei Authentifizierungselemente, die aber in der Regel alle über die N26-App nachgewiesen werden. Alle Faktoren gehören entweder zur Kategorie Wissen oder Besitz. Darüber hinaus kann der Kunde eigenständig verschiedene Limits setzen und sich über Echtzeitbenachrichtigungen über Kontoaktivitäten informieren lassen.

Wissenselemente

Es gibt zwei zentrale Wissenselemente in der Sicherheitsarchitektur von N26: Ein Passwort für den Login und eine PIN zum Überweisen und Geldabheben.

Zugangsdaten. Der Login in das N26-Konto benötigt die E-Mail-Adresse des Kontoinhabers sowie das dazugehörige Passwort. Beides wird bereits bei der Kontoeröffnung, die vollständig online abläuft, festgelegt. Das Passwort muss mindestens die folgenden Kriterien erfüllen: sieben Zeichen, eine Zahl, ein Sonderzeichen und ein Großbuchstabe [N2617]. Bei unseren Tests war es jedoch nicht notwendig, dass das Passwort ein Sonderzeichen enthält; sieben Zeichen mit einem Großbuchstaben und einer Zahl waren ausreichend.

Transfer-PIN. Bei der Transfer-PIN handelt es sich um eine vierstellige Zahl, die der Nutzer während des Aktivierungsprozesses angibt. Die Transfer-PIN wird vor allem für Bargeldabhebungen am Automaten und für das Auslösen von Transaktionen benötigt. Der Kunde kann die Transfer-PIN zu jedem Zeitpunkt innerhalb der App ändern. Hierfür ist es nicht notwendig, die alte PIN zu kennen; stattdessen ist die Eingabe der Mastercard-ID ausreichend, die weiter unten beschrieben ist. Bei dreimaliger Falscheingabe der PIN wird die Karte gesperrt.

Besitzelemente

Die folgenden Elemente lassen sich im weiteren Sinn der Kategorie Besitz zuordnen. Mit Besitz sind hier Objekte gemeint, die sich in der Regel im alleinigen Zugriff des Kontoinhabers befinden, ohne dabei zwangsläufig Konformität zu regulatorischen Auflagen zu erreichen, wie sie in Kapitel 6 beschrieben werden.

Mastercard-ID. Diese zehnstellige Zahl ist auf der N26-Mastercard unterhalb des Karteninhabernamens aufgedruckt (siehe Abbildung 5.1). Es handelt sich dabei jedoch nicht um die Kreditkartennummer. Die Mastercard-ID ist eine wichtige Authentifizierungskennung im N26-Sicherheitssystem. Der Kunde benötigt sie für die initiale Verknüpfung, um die Transfer-PIN zu ändern und um ein Gerät wieder zu entknüpfen. Der Entknüpfungsprozess wird noch ausführlich in Abschnitt 5.2.3 dargestellt.

Abbildung 5.1: Die N26-Mastercard. Die zehnstellige Nummer unterhalb des Namens des Karteninhabers ist die Mastercard-ID.

Verknüpftes Gerät. Durch die Personalisierung fungiert das Smartphone des Kunden als Besitzelement. Basis hierfür ist ein RSA-Schlüsselpaar. Um ein Gerät erstmalig zu verknüpfen, muss der Kunde zum einen seine Mastercard-ID und zum anderen ein Einmalpasswort, das ihm per SMS zugestellt wird, eingeben. Sind beide Informationen korrekt, hinterlegt das N26-Backend den öffentlichen RSA-Schlüssel, den die App erstellt und sendet. Diesen Schlüssel verwendet das Backend in Zukunft für eine Challenge-Response-Authentifizierung, die unter anderem zur Bestätigung von Überweisungen zum Einsatz kommt. Das verknüpfte Smartphone ist jedoch nicht für alle Änderungen am Konto notwendig.

SIM-Karte. Wie bereits angesprochen, erhält der Kunde im Rahmen des Verknüpfungsprozesses einen Code per SMS. Zu diesem Zweck muss er eine Mobilfunknummer angeben. Die Nummer muss nicht zwangsläufig der SIM-Karte zugeordnet sein, die üblicherweise in einem Smartphone steckt. Ein Zugriff auf die Nummer wird nur im Rahmen der Entknüpfung erneut benötigt. Darüber hinaus spielt die SIM-Karte keine Rolle.

Sonstige Sicherheitsmerkmale

Abgesehen von Wissens- und Besitzelementen bietet ein N26-Konto noch weitere Sicherheitsfunktionen. Sie haben gemeinsam, dass sie nicht für die Authentifizierung genutzt werden, sondern dem Kunden das Setzen persönlicher Limits oder das schnelle Reagieren bei Anomalien erlauben. Alle Einstellungen benötigen keinen Zugriff auf das verknüpfte Smartphone.

Geschützte Daten. Die meisten Daten und sogar Authentifizierungselemente kann der Kunde eigenhändig innerhalb der App ändern. Bestimmte Daten sind jedoch geschützt und können nur durch den N26-Support geändert werden. Ein wichtiger Datenpunkt, der nur durch das N26-Personal geändert werden kann, ist z. B. die E-Mail-Adresse des Kunden. Nur über den Zugriff zum E-Mail-Konto kann z. B. das Passwort zurückgesetzt werden. Die Lieferadresse, an die z. B. eine neue Karte geschickt wird, kann hingegen zu jederzeit geändert werden.

Karteneinsatz und -limits. Der Kunde kann selbstständig entscheiden, wie die N26-Mastercard genutzt werden kann. Hierfür bietet die App Schalter, die entscheiden, ob die Karte für Onlinezahlungen, Bargeldabhebungen oder Zahlungen im Ausland verwendet werden darf. Er kann die Verwendung der Karte auch komplett sperren und jederzeit wieder entsperren. Zusätzlich kann der Kunde auch Limits für Abhebungen und Überweisungen setzen. Da diese Einstellungen ohne das verknüpfte Gerät angepasst werden können, schützen sie den Kontoinhaber nur, wenn lediglich die Kreditkartendaten oder die Karte selbst in fremde Hände gelangen.

Push-Nachrichten. Bei vielen Ereignissen erhält der Kunde Echtzeitnachrichten via Push auf sein Smartphone. Hierzu zählen erfolgreiche und erfolglose Zahlungen, ein- und ausgehende Überweisungen sowie Lastschriftabbuchungen. Diese Nachrichten können einem Kunden helfen, auf ungewöhnliche Aktivitäten schneller zu reagieren.

5.2.2 Frontend

Im Folgenden beschreiben wir die Schwachstellen, die wir in den N26-Frontends identifiziert haben. Die App für Android und iOS ist die wichtigste Schnittstelle zum Kunden und erlaubt das Tätigen sämtlicher Aktionen. Darüber hinaus bietet N26 auch ein browserbasiertes Onlinebanking. Diese Version ist jedoch im Vergleich zur App im Funktionsumfang eingeschränkt. Insbesondere können Aktionen, die das verknüpfte Smartphone benötigen, nur ausgelöst werden; für die Bestätigung muss zwingend das verknüpfte Gerät bemüht werden. Letztendlich sprechen alle drei Frontends mit der gleichen API. Die Android, wie auch die iOS-App verwenden keinerlei Schutzmaßnahmen, wie wir sie in Abschnitt 4.3 empfohlen haben. Dadurch war es für uns ein Einfaches, die Kommunikation zu analysieren und Funktionen per Reverse Engineering zu verstehen.

Auf diesem Weg haben wir zwei Schwachstellen identifiziert: Eine kann für eine vom eingesetzten Frontend unabhängige Transaktionsmanipulation genutzt werden, während es die andere unter Android erlaubt, beliebige Webinhalte zu injizieren.

Transaktionsmanipulation

Ähnlich zu Abschnitt 3.3.2, ist es das Ziel dieses Angriffs, eine nutzerinitiierte Transaktion in Echtzeit zu manipulieren. Der Angriff ist dabei so gestaltet, dass das Opfer die Manipulation nicht erkennen kann. Technisch läuft eine reguläre Überweisung wie folgt ab:

1) Der Kunde loggt sich mit E-Mail/Passwort ein und füllt eine Überweisung mit den üblichen Zahlungsdetails Empfängername, IBAN, Betrag und Verwendungszweck aus.
2) Als Nächstes gibt der Nutzer seine Transfer-PIN ein, die zusammen mit dem Überweisungsauftrag an das Backend versandt wird.
3) Insofern die eingegebene PIN richtig ist, ist die Transaktionsauslösung abgeschlossen und das Backend sendet eine Push-Nachricht an das verknüpfte Smartphone. Die Push-Nachricht beinhaltet eine verschlüsselte TAN, die mit dem privaten Schlüssel des verknüpften Geräts entschlüsselt werden kann. Daten abseits der verschlüsselten TAN sind in der Push-Nachricht nicht enthalten.
4) Wenn der Kunde die Push-Nachricht öffnet, werden ihm die Transaktionsdetails zur Bestätigung erneut angezeigt. Gibt der Nutzer sein Einverständnis, dann wird die TAN entschlüsselt und im Klartext an den Server gesendet.
5) Ist die TAN korrekt, wird die Transaktion ausgeführt.

Falls es einem Angreifer gelingt, einen Man-in-the-Middle-Angriff durchzuführen, dann kann er die Transaktion für den Nutzer transparent manipulieren. Der Angriff wird durch zwei Defizite begünstigt. Erstens erfolgt keinerlei Zertifikats-Pinning, obwohl die N26-App jedwede Kommunikation konsequent via TLS verschlüsselt überträgt. Zweitens enthält die Push-Nachricht einzig und allein die verschlüsselte TAN, nicht aber die Transaktionsdetails, die beim N26-Backend eingegangen sind. Dadurch muss die App im Bestätigungsdialog auf die Daten zurückgreifen, die das Opfer initial eingegeben hat, statt die Details von einem zweiten Kommunikationskanal verschlüsselt zu empfangen und anzuzeigen. Alternativ hätte die N26-App

die Transaktionsdaten auch signieren und zusammen mit der entschlüsselten TAN an das Backend schicken können. Ein solches Vorgehen erfolgt ebenfalls nicht.

Ein Angreifer kann sich diese Schwachstellen zunutze machen, um Transaktionen in Echtzeit zu manipulieren. Hierfür muss er zum einen in der Lage sein, den Netzwerkverkehr umzuleiten und zum anderen benötigt er ein gültiges Zertifikat des N26-API-Endpunkts, dem das Frontend des Opfers vertraut. Es sind verschiedene Ansätze denkbar, um beides zu erreichen:

- Gelingt es einem Angreifer, den DNS-Eintrag des N26-API-Endpunkts unter seine Kontrolle zu bringen, kann er die Transaktionen aller Kunden beliebig manipulieren. Durch die Kontrolle des DNS-Eintrags kann er die gesamte Kommunikation über seinen Server umleiten. Damit die N26-App das Zertifikat des Angreifer-Servers akzeptiert, benötigt er noch ein Zertifikat von einer Zertifizierungsstelle, dem sowohl Android als auch iOS standardmäßig trauen. Ein solches Zertifikat kann sich der Angreifer bei Kontrolle des DNS-Eintrags z. B. per Let's Encrypt kostenlos ausstellen lassen [Aer+17]. Das Szenario ist weit entfernt von reiner Theorie: Am 22. Oktober 2016 wurde die komplette Onlinebanking-Kommunikation einer wichtigen brasilianischen Bank durch einen übernommenen DNS-Eintrag umgeleitet [Gre17].
- Ein Nutzer könnte über Phishing dazu angewiesen werden, einen VPN-Anbieter zu installieren [Ikr+16]. Alternativ könnte ein N26-Kunde auch dazu verleitet werden, dass er einen systemweiten Proxy einträgt, der auf den Angreifer-Server zeigt. Zusätzlich trägt das Opfer eine Zertifizierungsstelle ein, die der Angreifer kontrolliert. Es ist auch denkbar, dass eine als vertrauenswürdig geltende Zertifizierungsstelle fälschlicherweise ein Zertifikat für die N26-Domain ausstellt [Ama+17]. In Summe kann der Angreifer wieder alle Kommunikation in Klartext lesen und manipulieren.
- Ein Angreifer kann die Transaktion über einen Trojaner manipulieren, wenn es ihm gelingt, das transaktionsauslösende Gerät mit Schadsoftware zu infizieren. Selbstverständlich könnte auch eine privilegierte Schadsoftware, wie wir sie in Abschnitt 3.3.2 angenommen haben, den Angriff durchführen.

WebView-Popup-Injektion

Während des Reverse Engineerings der Android-App sind wir auf mehrere Deeplinks gestoßen, die im Manifest der App registriert sind. Die Forschung hat bereits ge-

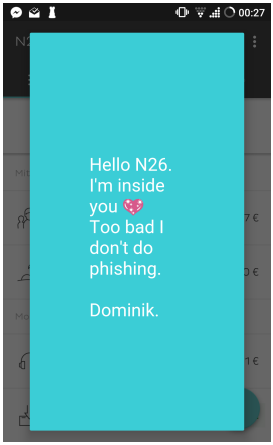


Abbildung 5.2: Injizierte Webseite. Ein externer Deeplink startet die N26-App und stellt innerhalb eines Popups eine beliebige Webseite dar.

zeigt, dass Deeplinks meist einen negativen Einfluss auf die Sicherheit einer App haben [Liu+17]. In diesem Fall sind wir auf einen Deeplink gestoßen, der es uns erlaubt, einen Popup in die N26 Android-App zu injizieren: `number26://main/?tutorial=https://cs1.tf.fau.de`. Durch diesen Link kann ein Angreifer die N26-App starten und jede Webseite, die durch `tutorial=` angegeben ist, innerhalb der App als Popup anzeigen lassen. Das Ausführen von JavaScript-Code ist ebenfalls möglich. Hierdurch erhöht sich die Angriffsfläche abhängig von der Android-Version erneut [Li+17b].

Der Popup kann ein wichtiger Baustein in einem erfolgreichen Phishing-Angriff sein: Da sich der Nutzer zuerst wie üblich in die N26-App einloggen muss, bevor der Popup dargestellt wird, wirkt der dargestellte Inhalt besonders authentisch. Im Gegensatz zu bekannten Phishing-Angriffen, die in der Regel eine bösartige App mit entsprechenden Berechtigungen benötigt [CQM14], braucht dieser Angriff keine zusätzliche Software oder Berechtigungen. Stattdessen könnte der verwundbare Deeplink sogar von einer bösartigen Webseite ausgelöst werden. Die Größe und Gestaltung des Popups hängt vollständig von der angreiferkontrollierten Ziel-URL ab. Abbildung 5.2 zeigt einen Beispiel-Popup, der über den genannten Deeplink innerhalb der N26-App angezeigt wurde. Der Parameter `tutorial` deutet darauf hin, dass der Deeplink existiert, um Anleitungen auf eine generische Art und Wei-

Abbildung 5.3: Kundenidentifikation. Die N26-App lädt das Adressbuch des Kunden hoch und antwortet mit den Einträgen von N26-Kunden.

se darzustellen. Die Sicherheitsimplikation dieser Entscheidung wäre vermutlich augenscheinlich geworden, hätte Sicherheit einen höheren Stellenwert genossen.

5.2.3 Backend

Im Gegensatz zum vorhergehenden Abschnitt stellen wir nun Sicherheitslücken vor, die unabhängig von den eingesetzten Frontends ihren Ursprung in der Implementierung des Backends finden. Neben den Erkenntnissen, die wir bereits durch das Reverse Engineering der Frontends gewonnen haben, waren vor allem Mitschnitte der HTTPS-Kommunikation zum Aufspüren der Defizite hilfreich. Im Folgenden zeigen wir eine Reihe von teilweise schwerwiegenden Sicherheitslücken im Backend von N26.

Informationsleck

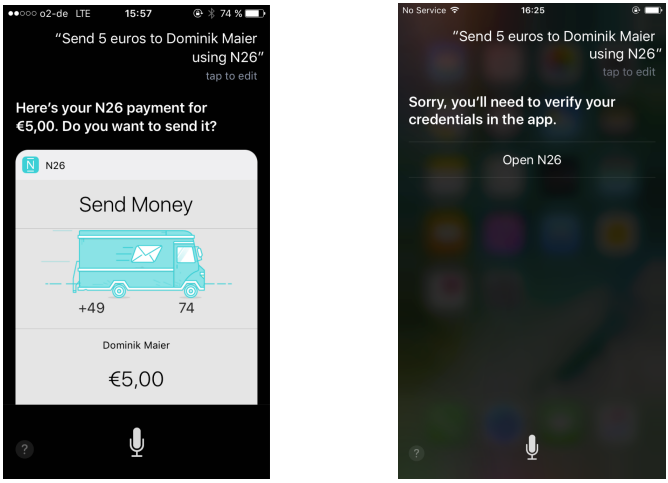
N26 bietet Peer-to-Peer-Zahlungen an, die das Unternehmen *MoneyBeams* nennt. Im Unterschied zu regulären SEPA-Überweisungen werden die Beträge sofort versendet und gutgeschrieben. Diese Funktion ist jedoch nur verfügbar, insofern sowohl der Sender als auch der Empfänger zu einem N26-Konto gehören. Um seinen Kunden innerhalb der App die Kontakte anzuzeigen, die ebenfalls ein N26-Konto besitzen, schickt die App die E-Mail-Adressen und Telefonnummern aller Kontakte in Klartext an das N26-Backend. Der Server antwortet daraufhin mit den E-Mail-Adressen und Telefonnummern, die er einem N26-Kunden zuordnen kann. Dieser Prozess ist ebenfalls in Abbildung 5.3 dargestellt.

Die Daten in dem Adressbuch werden vor dem Upload nicht pseudonymisiert, geschweige denn, dass ein Zero-Knowledge-Ansatz Verwendung findet [FFS88]. Fernab der offensichtlichen Datenschutzproblematik kann diese Schnittstelle von einem Angreifer genutzt werden, um E-Mail-Adressen und Telefonnummern auf Zuordnung zu einem N26-Konto zu prüfen. Das ist besonders nützlich, um gezielte Phishing-Angriffe durchzuführen [Cap+14]. Da ein Adressbuch unter Umständen sehr viele Einträge beinhaltet, können über die Schnittstelle mehr als 1 000 Kandidaten auf einmal getestet werden. Technisch ist die Funktion als HTTP-Post-Anfrage realisiert, die die Kontaktdaten als JSON-Liste an das Backend übermittelt. Das Backend antwortet wiederum mit einer Untermenge der empfangenen Listeneinträge. Es werden also zumindest keine zusätzlichen Informationen über den Kunden an den Anfragenden übermittelt. Abgesehen von der an sich fragwürdigen Existenz einer solchen Funktion ist das fehlende Anfrage-Limit ein Problem: In unseren Tests war es problemlos möglich, mehrere Millionen E-Mail-Adressen und Telefonnummern zu testen. Selbst ein Brute-Force-Angriff wäre möglich gewesen.

Unbestätigte Siri-Transaktionen

Mit dem Erscheinen von iOS 10 können Drittanbieter-Anwendungen erstmals Siri zur Interaktion mit dem Benutzer einsetzen. N26 adaptierte die neue Funktion sofort und erlaubt seinen Kunden, Transaktionen über Siri zu diktieren. Das Limit für solche Transaktionen liegt bei €25 pro Transaktion und bei €200 pro Tag. Nachdem Siri den Befehl des Sprechers verarbeitet hat, werden die Transaktionsdetails erneut angezeigt und nach Bestätigung des Nutzers versendet. Es ist nicht notwendig, dass der Kunde mit der N26-App interagiert. Wie Abbildung 5.4 zeigt, sieht N26 nur vor, dass solche Transaktionen mit dem verknüpften iPhone durchgeführt werden können. Es liegt deshalb nahe, dass N26 bei Siri-Transaktionen ähnlich vorgeht, wie bei regulären Überweisungen innerhalb der App.

Es stellt sich jedoch heraus, dass dem nicht so ist. Stattdessen hat N26 einen neuen API-Endpunkt für Siri-Transaktionen geschaffen, der keine Bestätigung durch das verknüpfte Gerät vorsieht. Es handelt sich bei der Forderung nach dem verknüpften Gerät also lediglich um eine clientseitige Restriktion. Ein solches Vorgehen entspricht in keiner Weise den Best Practices. Obwohl sich auf diesem Weg maximal €200 pro Tag erbeuten lassen, steht das Geld sofort zum Abheben an einem Geldautomaten zur Verfügung, insofern der Empfänger auch ein N26-Konto besitzt. Gelingt es die Zugangsdaten mehrerer N26-Konten zu erlangen, könnten mittels Siri-



(a) Verknüpftes iPhone. (b) Unverknüpftes iPhone.

Abbildung 5.4: Reaktion der N26-App auf die Spracheingabe einer Siri-Transaktion auf einem verknüpften und einem unverknüpften iPhone.

Transaktionen hohe Geldbeträge auf einem Konto vereint und dort sofort abgehoben werden.

Freizügige Risikoanalyse

Bei jeder Sicherheitslücke stellt sich die Frage, ob es in der Praxis aufgrund von Risikoanalysen im Backend nicht deutlich schwieriger ist, sie auszunutzen. Gerade N26 brüstet sich damit, dass sie über intelligente Algorithmen verfügen, die Betrug identifizieren und verhindern. Hierfür bezieht N26 sogar Gelder aus den Europäischen Fonds für regionale Entwicklung. Wir haben die Effektivität der Betrugsprävention getestet. Zu diesem Zweck haben wir über 2 000 Siri-Transaktionen in einer Zeitspanne von 30 Minuten durchgeführt. Jede Transaktion wurde vom N26-Backend sofort angenommen. Es gab ebenfalls kein Anfrage-Limit. Dennoch waren wir davon überzeugt, dass N26 kurzfristig auf uns zukommen würde, um Nachforschungen zu dieser offensichtlich ungewöhnlichen Kontoaktivität anzustellen.

Eine Anfrage erhielten wir jedoch erst drei Wochen später per E-Mail durch den N26-Support. In der Nachricht forderte man uns auf, eine Erklärung für die „ungewöhnlich hohe Anzahl“ zu liefern. Sollten wir dem nicht nachkommen, kann eine Kündigung des Kontos die Folge sein. Die Reaktion ist zunächst nachvollziehbar, da das Verhalten eine klare Verletzung der AGB darstellt. Es überraschte uns jedoch, dass N26 die Kommunikation mit dem *Empfänger* und nicht dem Sender der Zahlungen suchte.

Neben der offensichtlich manuellen Bearbeitung der Fälle, legt das Vorgehen durch N26 nahe, dass für das Aktivwerden keine Sicherheitsmotive ausschlaggebend waren. Stattdessen gehen wir davon aus, dass N26 eine geschäftliche Nutzung des Kontos annahm. Darüber hinaus kann unser Vorgehen in keiner Art und Weise als unauffällig bezeichnet werden. In Summe bezweifeln wir, dass die Risikoanalyse des N26-Backends gegenüber einem echten Angriff mit höheren Beträgen angemessen reagiert hätte.

Ungeschützte Entknüpfung

Selbst wenn ein Angreifer in der Lage ist, die Zugangsdaten sowie die Transfer-PIN zu erlangen, kann er mit diesen Informationen allein keine SEPA-Überweisungen durchführen, da er keinen Zugriff auf den privaten Schlüssel des verknüpften Geräts hat. Das Ziel des folgenden Angriffs ist es deshalb, Kontrolle über das verknüpfte Smartphone zu erlangen. Im Gegensatz zu dem Vorgehen in Abschnitt 3.3.1 soll die App des Opfers jedoch nicht kopiert, sondern die Verknüpfung mit dem Smartphone aufgelöst werden. Im Anschluss kann der Angreifer dann sein eigenes Gerät verknüpfen. Der Verknüpfungsprozess ist unkompliziert und erfordert neben den Zugangsdaten nur die Angabe einer Telefonnummer. Die Bestätigung der Nummer erfolgt über eine vierstelliges, numerisches Einmalpasswort, das per SMS zugestellt wird. Der Entknüpfungsprozess ist hingegen durch mehrere Schritte und Faktoren gesichert. Um ein Smartphone regulär zu entknüpfen, muss ein Kunde die folgenden Aktionen durchführen:

1) Starten des Entknüpfens über die App oder das Webinterface. Daraufhin sendet N26 einen Link, der ein Token als Parameter trägt, an die E-Mail-Adresse des Kunden. Diesen Link muss der Nutzer öffnen und den dargestellten Instruktionen folgen.

2) Eingabe der Transfer-PIN.

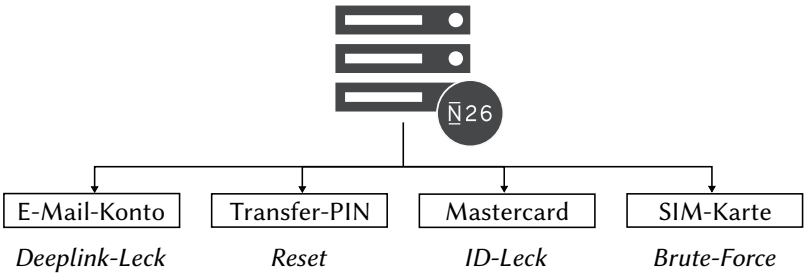


Abbildung 5.5: Smartphone-Entknüpfung. Die beteiligten Authentifizierungselemente zum Auflösen der Smartphone-Verknüpfung und ihre Schwachstellen.

3) Eingabe der Mastercard-ID.

4) N26 sendet an die im Verknüpfungsprozess angegebene Telefonnummer eine fünfstellige Nummer, die zum Abschluss eingegeben werden muss.

Der Entknüpfungsprozess erfordert Zugriff auf vier Ressourcen: 1) Das E-Mail-Konto, 2) die Transfer-PIN, 3) die N26-Mastercard und 4) die SIM-Karte. Obwohl es positiv zu bewerten ist, dass N26 den Entknüpfungsprozess als besonders sicherheitskritisch identifiziert hat, lässt sich jeder der genannten Schritte nur mit Kenntnis der Zugangsdaten umgehen. Das Vorgehen ist in Abbildung 5.5 zusammengefasst und verläuft wie folgt:

1) Der Start des Entknüpfungsprozesses führt zu einer HTTP-GET-Anfrage, die im E-Mail-Versand des beschriebenen Links mündet. Statt jedoch die Anfrage nur über einen Statuscode zu bestätigen, übermittelt N26 in der HTTP-Antwort ebenfalls den Link, der an die E-Mail-Adresse des Kontoinhabers versendet wurde. In Konsequenz ist kein Zugriff auf das E-Mail-Konto notwendig.

2) Die Transfer-PIN ist dem Angreifer nicht bekannt, kann aber allein durch Kenntnis der Mastercard-ID zu einer beliebigen PIN geändert werden. Wie die Mastercard-ID gewonnen werden kann, ist im nächsten Punkt beschrieben. Grundsätzlich lässt sich die Transfer-PIN in dem Prozess als überflüssig bezeichnen, da sie ohnehin zurücksetzbar ist. Es ergibt sich durch die Forderung keine zusätzliche Sicherheit.

3) Die Abfrage der Mastercard-ID würde eigentlich erfordern, dass der Angreifer die Karte physisch in seinem Besitz hat. Die Mastercard steht auf keine Art und Weise in Bezug zur Kreditkartennummer. Ein Zugriff auf die Karte ist jedoch nicht notwendig, da die Mastercard-ID Teil einer jeden Transaktion ist, die mit der Mastercard getätigt wird. Dabei ist es unerheblich, ob es sich um eine Abhebung am Geldautomaten, eine Zahlung an der Verkaufsstelle oder eine Onlinezahlung handelt. Obwohl dieser Wert im Frontend nicht sichtbar dargestellt wird, fungiert die Mastercard-ID des Kunden als Prefix für eine Transaktionskennung. Damit ein Angreifer die Mastercard-ID eines Opfers in Erfahrung bringen kann, muss dieses die Karte nur ein einziges Mal eingesetzt haben. Wie im vorangehenden Punkt angesprochen, kann mit der Kenntnis der Mastercard-ID auch die Transfer-PIN auf eine beliebige Zahlenkombination geändert werden.

4) Im letzten Schritt muss der Angreifer regulär Zugriff auf die SMS erlangen, die N26 an die hinterlegte Mobilfunknummer sendet. Obwohl in der Vergangenheit bereits Angriffe beobachtet wurden, die das Mitschneiden von SMS-Nachrichten erlaubten (siehe Abschnitt 2.2.2), ist in diesem Fall ein primitiveres Vorgehen möglich. Da der Endpunkt, der das Einmalpasswort in der SMS überprüft, kein Anfragelimit implementiert und beliebig viele Versuche zulässt, kann das Token per Brute-Force-Angriff erraten werden. Auf diese Weise waren wir in der Lage, 160 Kandidaten pro Sekunde zu testen. Da das Einmalpasswort lediglich aus fünf Ziffern besteht, hat ein Angreifer die Kombination im Durchschnitt in etwa fünf Minuten erraten.

Diese Defizite zusammengenommen kann der Angreifer das Smartphone entknüpfen, ohne Zugriff auf das E-Mail-Konto zu haben, die Transfer-PIN zu kennen und ohne die Mastercard oder die SIM-Karte zu besitzen. Im Anschluss kann der Angreifer trivial ein eigenes Gerät verknüpfen.

Der komplette Entknüpfungsprozess dauert im schlechtesten Fall maximal 10 Minuten. Dennoch sollte der Angreifer einen Zeitpunkt wählen, an dem das verknüpfte Mobilgerät nicht die Aufmerksamkeit des Opfers hat, da es insgesamt drei E-Mails und eine SMS erhält. Die erste E-Mail enthält den Link, die zweite informiert über das Zurücksetzen der Transfer-PIN und die dritte über das erfolgreiche Entknüpfen. Die SMS beinhaltet das Einmalpasswort für den abschließenden Schritt. Ein Opfer hätte keine Möglichkeit mehr, selbst über die App zu reagieren. Auch um eine schnelle Reaktion durch den N26-Support auszuschließen, böte sich aus Angreifersicht eine Durchführung außerhalb der Geschäftszeiten an.

5.3 Angriffsszenarien

Im letzten Abschnitt haben wir die Sicherheitsdefizite im Front- und Backend beschrieben. Im Folgenden zeigen wir verschiedene Angriffsszenarien auf, die sich die verschiedenen Defizite zunutze machen, um zuerst die Zugangsdaten zu erlangen und anschließend volle Kontrolle über das Konto zu erhalten.

Wir nehmen an, dass die Opfer der Angriffe zumindest einmalig ihre Mastercard in irgendeiner Form eingesetzt haben. Das ist notwendig, damit die Mastercard-ID über die Zugangsdaten verfügbar ist. Es ist jedoch wahrscheinlich, dass der Kunde die Mastercard bereits im Einsatz hatte, da sie eine zentrale Rolle im N26-Kontomodell einnimmt.

5.3.1 Ermittlung der Zugangsdaten

Um vollen Zugriff auf ein N26-Konto zu erhalten, muss ein Angreifer zuerst die Zugangsdaten ermitteln. Es ergeben sich vielfältige Möglichkeiten, um an die Kombination aus E-Mail-Adresse und Passwort zu gelangen.

Kundenidentifikation. Zu allererst muss ein Angreifer potenzielle Opfer identifizieren, die ein Konto bei N26 führen. Da N26 eine E-Mail-Adresse als Benutzerkennung verwendet, kommen die Adressen aus Datenbanklecks hierfür in Frage [Hun19]. Um die Zugehörigkeit einer E-Mail-Adresse zu einem N26-Konto zu überprüfen, kann das vorgestellte Informationsleck genutzt werden. Auf diese Weise kann eine Vielzahl an Kandidaten schnell und zuverlässig überprüft werden.

E-Mail-Konto. Es ist ein Schwachpunkt des N26-Sicherheitskonzepts, dass das Passwort lediglich über den Zugriff auf das E-Mail-Konto zurückgesetzt werden kann. An sich realisiert N26 eine solide Passwortrichtlinie und es kommt ein Anfrage-Limit beim Login zum Einsatz. In dieser Kombination sind Brute-Force-Angriffe praktisch ausgeschlossen. Durch die Zurücksetzbarkeit des Passworts allein durch das E-Mail-Konto fällt die N26-Passwortrichtlinie jedoch faktisch auf die des E-Mail-Anbieters zurück. Auf dessen Richtlinie hat N26 keinen Einfluss; es sind selbst Passwörter wie `1234` denkbar. Aus Angreifersicht kann es also auch lohnend sein, sich nicht direkt auf das Passwort zum N26-Konto, sondern auf das des E-Mail-Kontos zu konzentrieren.

Passwortwiederverwendung. Passwörter und PINs werden aufgrund der pro Person großen Anzahl an Onlinekonten häufig wiederverwendet [Das+14; Gol+18; Lya+18; Wan+16]. Demnach ist es nicht unwahrscheinlich, dass ein N26-Kunde dasselbe Passwort auch für einen anderen Dienst nutzt. Die Wahrscheinlichkeit erhöht sich sogar noch, da die Passwortrichtlinie von N26 leicht unter den üblichen Standards liegt. In Konsequenz kann der Kunde ein bereits existierendes Passwort verwenden und muss sich nicht erst ein neues ausdenken.

Phishing. Die größte Gefahr ergibt sich jedoch durch Phishing-Angriffe, die damals wie heute sehr erfolgreich sind [She+10; Aon+18]. Phishing erfolgt oft per E-Mail an einen breiten Adressatenkreis. Die Phishing-E-Mails fordern den Empfänger – oftmals unter Erzeugung eines Drangs – dazu auf, dem dargestellten Link zu folgen, um sich in ihr Konto einzuloggen. Der Link führt jedoch nicht zum authentischen Internetauftritt des Anbieters, sondern zu einer Nachbaute. Gibt der Kunde dort seine Daten ein, schneidet sie der Angreifer mit.

Im Falle von N26 ist sogar ein Spear-Phishing-Angriff möglich [Cap+14; BGL17], da sich über das Informationsleck die Zielgruppe einwandfrei identifizieren lässt. Darüber hinaus ist es für einen N26-Kunden nicht unüblich, dass ihn offizielle E-Mails der Bank dazu auffordern, einem eingebetteten Link zu folgen.

Ein Angreifer kann die Authentizität des Phishing-Angriffs sogar noch steigern, indem er auf die Lücke zur Injektion einer Webview zurückgreift. Der Link in der Mail würde also nicht eine Webseite im Browser, sondern direkt die N26-App öffnen. Da der Angreifer den dargestellten Inhalt kontrolliert, kann er dem Nutzer dort weitere Anweisungen geben, profitiert aber trotzdem von der vertrauenswürdigen Umgebung der N26-App.

5.3.2 Großflächiger Angriff

Im Folgenden stellen wir einen Angriff vor, der im großen Stil N26-Kunden identifiziert, ihre Zugangsdaten in Erfahrung bringt, ein eigenes Gerät verknüpft und schlussendlich das komplette Guthaben inklusive Kredit entwendet.

Im Jahr 2016 wurden durch ein Datenbankleck die E-Mail-Adressen und gehashten Passwörter von 65 Millionen Dropbox-Nutzern öffentlich bekannt [Gib16]. Mittels der MoneyBeam-Schnittstelle konnten wir alle E-Mail-Adressen evaluieren und haben auf diesen Weg letztendlich 33 000 N26-Nutzer identifiziert. Diese Anzahl stellte zum Evaluationszeitpunkt über 10% aller Kunden dar. Ein Krimineller hätte

diese Information nutzen können, um gegen all diese E-Mail-Adressen einen Spear-Phishing-Angriff zu starten. Der Ursprung der E-Mail-Adressen liefert sogar einen validen Grund für eine Kontaktaufnahme. So könnte ein Angreifer behaupten, dass sein Opfer dringend das N26-Passwort ändern sollte, weil er unter dieser E-Mail-Adresse auch ein Dropbox-Konto führt. Um die Aufforderung besonders plausibel zu gestalten, öffnet der Link die Phishing-Webseite über die WebView-Popup-Injektionslücke direkt in der N26-App.

Nachdem der Angreifer die Zugangsdaten kennt, kann er – wie von uns gezeigt – bereits Siri-Transaktionen durchführen. Um jedoch die Begrenzung von €200 pro Tag zu überwinden, wird das verknüpfte Gerät benötigt. Wie von uns zuvor dargestellt, kann ein Angreifer aufgrund der vielfältigen Defizite automatisiert das Opfergerät entknüpfen und daraufhin einen eigenen RSA-Schlüssel registrieren. Im Anschluss kann er über das komplette Guthaben des Opfers verfügen. Es ist mithilfe des verknüpften Geräts sogar möglich, über weitere Einlagen zu verfügen: Jeder Kunde kann einen Kredit beantragen, der sofort einen bestimmten Überziehungsrahmen gewährt. Dabei sind mindestens €50 garantiert und maximal bis zu €2000 möglich. Nachdem sich das Transaktionsüberwachung als sehr freizügig herausgestellt hat, ist es nicht unwahrscheinlich, dass ein solcher Angriff unerkannt geblieben, zumindest aber nicht verhindert worden wäre.

5.3.3 Imitationsangriff

Während sich das letzte Angriffsszenario auf einen Angriff bezog, der sehr gut für eine große Kundengruppe funktioniert, ist es mit Kenntnis der Zugangsdaten auch möglich, bestimmte Kunden durch einen Imitationsangriff individuell anzugreifen. Wie wir bereits in Abschnitt 5.2.1 dargestellt haben, gibt es bestimmte Daten, die der Kunde nicht eingeständig ändern kann. Hiervon ist insbesondere die E-Mail-Adresse des Kunden betroffen, die nicht nur für den Login, sondern auch für das Entknüpfen vonnöten ist. Letzteres gilt auch für die im Verknüpfungsprozess hinterlegte Telefonnummer. Es gibt jedoch durchaus plausible Gründe, warum ein Kunde keinen Zugriff mehr auf die hinterlegte Telefonnummer haben könnte. Eine Ursache wäre ein Anbieterwechsel, der zu einer neuen Rufnummer führt. In solchen Fällen muss sich der Kunde an den N26-Support wenden. Um den Kunden zu authentifizieren, fragt er die folgenden Daten ab:

1) Nachdem der Kundendienst den Namen des Kunden aufgenommen hat, fragt er ihn nach der Mastercard-ID.

2) Als Nächstes muss der Kunde sein Geburtsdatum angeben.

3) Im letzten Schritt fragt das Servicepersonal nach dem aktuellen Kontostand. Das ungefähre Saldo ist ausreichend.

Alle drei Informationen sind allein über die Zugangsdaten abrufbar. Die Mastercard-ID fungiert wie beschrieben als Prefix für alle Mastercard-Transaktionen. Das Geburtsdatum wird zwar nicht in den Frontends dargestellt, ist aber Bestandteil einer Serverantwort. Der aktuelle Kontostand kann mit einfachem Zugriff auf das Konto problemlos ermittelt werden.

Da der Angreifer in der Lage ist, alle Antworten korrekt zu beantworten, geht der N26-Support von einem authentischen Kunden aus. Infolgedessen kann der Angreifer die Änderung beliebiger Informationen inklusive E-Mail-Adresse und Mobilfunkrufnummer veranlassen. Obwohl sich diese Art nur für gezielte und nicht für großflächige Angriffe eignet, hat sie den Vorteil, dass keinerlei Benachrichtigung des Opfers erfolgt. Der Angriff kann also insbesondere bei Opfern mit hohen Guthaben lukrativ sein.

5.4 Reaktion

Aufgrund der Schwere der identifizierten Defizite haben wir uns dazu entschlossen, die Mängel vorab an N26 zu melden. Da es uns unklar war, wie N26 von einer rechtlichen Perspektive auf unsere Funde reagieren würde, baten wir den Chaos Computer Club um Kontaktaufnahme. Die Befürchtungen erwiesen sich als unbegründet und N26 zeigte sich freundlich und dankbar. Am 25. September 2016 übermittelten wir einen ausführlichen Bericht an N26, der nicht nur die einzelnen Sicherheitsprobleme beschrieb, sondern auch konkrete Lösungsvorschläge enthielt. Wir räumten dem Unternehmen drei Monate Zeit ein, um die Probleme zu adressieren, bis wir die Ergebnisse unserer Arbeit erstmals am 27. Dezember 2016 auf dem 33. Chaos Communication Congress in Hamburg präsentierten [Hau16]. Soweit wir wissen, hat N26 alle gemeldeten Punkte adressiert. N26 reagierte zusätzlich mit der Einrichtung eines Bug-Bounty-Programms. Weiteres Material wie Videodemonstrationen finden sich unter dem folgenden Weblink: `https://www.cs1.tf.fau.de/n26`.

5.5 Fazit

In diesem Kapitel haben wir verschiedene Angriffe gegen das erfolgreiche und schnell wachsende Fintech N26 gezeigt. Zusammengenommen hätten die identifizierten Defizite einem Angreifer substanzielle Hilfestellung gegeben, um an die Einlagen der N26-Kunden zu gelangen. Die Antwort auf unsere Forschungsfrage 3 (Fintech-Sicherheit), ob sich der Ansatz der Fintechs negativ auf die Sicherheit auswirkt, fällt vor diesem Hintergrund eindeutig aus: Nachdem es sich bei N26 um ein außergewöhnlich gut finanziertes Start-up handelt, das dazu noch mit einer Vollbanklizenz operiert, darf bezweifelt werden, dass die IT-Sicherheit bei einer Mehrheit der Fintechs eine große Rolle spielt.

Durch seine nicht-funktionale Natur scheint es zunächst nachvollziehbar, die IT-Sicherheit geringer zu priorisieren, da sie dem eigentlichen Produkt keinen unmittelbaren Mehrwert verschafft. Ein Schutz der Endnutzer ist jedoch unabdingbar, um nicht nur einen Reputationsschaden für das betroffene Unternehmen selbst, sondern auch für die Fintech- und Bankenbranche allgemein zu vermeiden. Das Vertrauen der Kunden als konstante Gegebenheit zu begreifen, ist keine Strategie, die nachhaltig funktioniert. Deshalb ist es umso wichtiger, die Sicherheit konsequent von Anfang an in alle Prozesse mit einzubeziehen. Wir hoffen deshalb mit unserer Forschung ein Umdenken in Gang zu setzen, das zu einer höheren Priorisierung der Sicherheit führt. Unsere Arbeit kann auch dazu beitragen, mehr Gelder und Personal für diesen kritischen Bereich gegenüber Investoren zu rechtfertigen.

Ob es durch unseren Beitrag bei N26 tatsächlich zu einem nachhaltigen Strategiewechsel in Sachen IT-Sicherheit kam, muss zumindest bezweifelt werden. Im Oktober 2018 geriet N26 in das Visier der Bundesanstalt für Finanzdienstleistungsaufsicht (BaFin), da der Verdacht im Raum stand, N26 gehe mit seinem Identifizierungsverfahren bei der Kontoeröffnung nicht entschieden genug gegen Geldwäsche vor [BL18]. Demnach sei es ohne große Mühen möglich, ein Konto unter falschem Namen zu eröffnen, da N26 das günstigere aber weniger sichere Fotoidentverfahren dem der Videoidentifizierung vorzieht. Dem Vorwurf wird dadurch Nachdruck verschafft, dass N26 bei Kriminellen tatsächlich große Beliebheit zu genießen scheint [Sch19b; WW19]. Zudem wurde N26 im März 2019 Medienberichten zufolge Ziel großflächiger Phishing-Angriffe, die zu hohen Schadenssummen führten [Sch19a; Jau19]. Wenig später wurde bekannt, dass die BaFin im Rahmen einer Sonderprüfung „unter anderem Missstände bei der Personalausstattung sowie beim Management von ausgelagerten Aufgaben und bei der Technik“ [KOS19] festgestellt hat. Würden

die Mängel nicht schnellstmöglich adressiert, könne die BaFin mit einer Einlagendeckelung reagieren und die Wachstumsperspektive des Fintechs so empfindlich bremsen. Außerdem hätten neben den Kunden auch andere Banken über die dürftige Erreichbarkeit von N26 bei akuten Problemen geklagt.

Die von uns dargestellte Gefahr hat sich demnach als real erwiesen, auch wenn sich das Schadenspotenzial durch unsere Forschung deutlich verringert hat. Obwohl N26 die Sicherheitsprobleme anfänglich als rein theoretisch abtat, zeigte sich der Gründer und CEO Valentin Stalf 2018 gegenüber dem BSI dann doch erleichtert darüber, dass die Sicherheitslücken nicht in der Praxis ausgenutzt wurden und sah schwerwiegende Folgen für seine Bank als abgewendet [Gut18]. Diese Aussage gewinnt nun vor den Entwicklungen im Frühjahr 2019 erneut an Relevanz und unterstreicht damit schlussendlich die Wichtigkeit unseres Forschungsbeitrags.

6

Bankgeschäfte unter der Zahlungsdiensterichtlinie II

Das Schlimmste ist überstanden.

— Richard Fuld, CEO Lehman Brothers, 2008 [Rie08]

Das Kapitel beschäftigt sich mit den technischen Rahmenbedingungen, die allgemein und regulatorisch für die Sicherungsverfahren im Online- und Mobilebanking gelten müssen, damit sie sicher verwendet werden können. Die sichere Anwendung des Verfahrens wird dem Kunden wiederum durch die Bank beschrieben und als Sorgfaltspflicht auferlegt. Die nachfolgenden Betrachtungen geschehen unter der Annahme, dass der Kunde die Verfahren tatsächlich nach Vorgabe seiner Bank anwendet. Diese Hypothese setzt hingegen voraus, dass die durch den Nutzer auszuführenden Authentifizierungsschritte an sich zumutbar sind und klar kommuniziert werden. Inwiefern der Kunde den Auflagen seiner Bank in der Praxis nachkommt und nachkommen kann, ist Gegenstand von Kapitel 7.

Im Folgenden beschreiben wir zunächst abstrakte Anforderungen, die allgemein an sichere Transaktionen zu stellen sind. Im nächsten Unterkapitel werden dann die Vorgaben zur starken Kundenauthentifizierung der Technischen Regulierungsstandards (RTS) der Zahlungsdiensterichtlinie II vorgestellt, ehe wir sie mit unseren allgemeinen Vorgaben vergleichen. Anschließend bewerten wir die gängigen Sicherungsverfahren im Online- und Mobilebanking hinsichtlich ihrer Konformität zu den regulatorischen Vorgaben. Der letzte Abschnitt beschäftigt sich mit der Frage, welche Angriffsfläche im Transaktionsprozess auch im Geltungsbereich der neuen Regulierung noch zurückbleibt.

6.1 Allgemeine Voraussetzungen

In diesem Abschnitt sollen allgemeine Anforderungen an die Sicherungsverfahren formuliert werden. Zum einen sind abstrakte Kriterien zur grundsätzlichen Bewertung der Sicherheit eines Legitimierungsverfahrens notwendig, zum anderen haben sie auch für das Haftungsrecht Relevanz.

Praktische Unüberwindbarkeit. Seit dem Grundsatzurteil des Bundesgerichtshofs (BGH) vom 26. Januar 2016 gilt der Anscheinsbeweis bei der Verwendung eines Sicherungsverfahrens, das zum Transaktionszeitpunkt den Ansprüchen der „praktischen Unüberwindbarkeit" genügt [BGH16]. Der Anscheinsbeweis stellt ein Mittel zur vereinfachten Beweisführung dar, bei dem der Beweis eines bestimmten Tatbestands entfällt; stattdessen gilt der Beweis aufgrund des ersten Anscheins als erbracht. Es wird also durch das Vorliegen eines bestimmten Umstands auf die Ursache geschlossen, ohne dass hierfür ein Beweis erbracht werden muss. Eine Beweislastumkehr, in der der durch den Anscheinsbeweis Benachteiligte das Gegenteil beweisen muss, ergibt sich jedoch nicht: Der Anscheinsbeweis kann bereits dadurch erschüttert werden, dass konkrete Umstände vorgetragen werden, die einen atypischen Geschehensverlauf nahelegen. Gelingt die Erschütterung, liegt die volle Beweislast bei der durch den Anscheinsbeweis begünstigten Partei.

Allgemein anerkannt ist der Anscheinsbeweis bei Kartenzahlungen, die durch den Kartenchip und die PIN-Eingabe authentifiziert werden. Solche Zahlungen liegen regelmäßig am PoS oder bei Geldautomatenverfügungen vor. Da für Chip-und-PIN-Zahlungen weder allein der Besitz der Karte noch das Wissen um die PIN für eine erfolgreiche Zahlung genügt, wird entweder eine selbstständige Verfügung durch den Kunden oder aber eine Pflichtverletzung angenommen, die sich durch die gemeinsame Verwahrung von Zahlungskarte und PIN ergibt.

Ein derart typischer Ereignisverlauf ergibt sich im Online- und Mobilebanking schon aufgrund der unterschiedlich ausgeprägten Sicherungsverfahren nicht. Zur Feststellung der praktischen Unüberwindbarkeit eines Verfahrens ist es deshalb zweckdienlicher, abstrakte Anforderungen zu formulieren. Dabei ist es nicht Ziel, jeden Angriffsvektor kategorisch auszuschließen; stattdessen sollen Auflagen entwickelt werden, die zum einen realistisch sind, zum anderen aber auch ein Sicherheitsniveau erreichen, das einen Angriff im Massenzahlungsverkehr theoretisch erscheinen lässt. Sind diese Bedingungen an das Sicherungsverfahren erfüllt, kann von praktischer Unüberwindbarkeit gesprochen und der Anscheinsbeweis anerkannt werden.

Allgemeines Sicherheitsziel. Obwohl die Vielzahl der sehr verschieden funktionierenden Legitimierungsverfahren die Formulierung allgemeiner Sicherheitsanforderungen erschwert, sind die Sicherheitsziele bei digitalen Bankgeschäften gut verstanden. Wie im klassischen papiergebundenen Zahlungsverkehr existieren im Online- und Mobilebanking zwei Akteure: die Bank und der Kunde. In beiden Fällen ist das wesentliche Sicherheitsziel auch dasselbe: Eine Transaktion wird durch die Bank nur dann ausgeführt, wenn eine Autorisierung (also eine persönliche Willenserklärung) eines Kontobevollmächtigten für die Transaktion vorliegt und diese Willenserklärung rechtlich bindend bei der Bank abgegeben wurde.

Im klassischen Zahlungsverkehr darf eine Transaktion nur dann ausgeführt werden, wenn ein entsprechender Auftrag unterschrieben bei der Bank abgegeben wurde. Zur Beurteilung der Sicherheit eines Legitimierungsverfahrens muss dieses allgemeine Sicherheitsziel erst in technische Bedingungen überführt werden.

Technische Voraussetzungen. Die Vorteile des Onlinebankings ergeben sich durch die Abwicklung von Zahlungsvorgängen mithilfe einer Rechenanlage, die autonom Finanztransaktionen ausführt. Um mit diesem Computer zu interagieren, benötigt der Kunde jedoch gleichsam mindestens einen Computer, der seinen Auftrag entgegennimmt und der dafür sorgt, dass dieser an den Computer der Bank übermittelt wird. In der Praxis spielen auf Seiten des Kunden dabei oft mehrere Computer eine Rolle: etwa die persönliche Bankkarte des Nutzers, ein spezielles Gerät, in das diese eingesteckt wird, oder der Computer, der den Auftrag entgegennimmt. Für die Sicherheitsbetrachtung wesentlich sind die Computer, die zusammenwirken müssen, damit der Kunde die Daten seiner digitalen Transaktion prüfen und anschließend seine Autorisierung erklären kann. Mit diesem Blick kann man das oben formulierte allgemeine Sicherheitsziel in Form von zwei konkreten Anforderungen technisch präzisieren:

1) Eine digitale Transaktion wird nur genau dann durchgeführt, wenn sie vom Nutzer willentlich ausgeführt wurde.
2) Eine vom Nutzer willentlich ausgeführte digitale Transaktion ist manipulationsfrei.

Hierbei fordert 1), dass ein Zugriff auf die am Transaktionsprozess beteiligten Authentifizierungselemente nicht, bzw. nur in solch einer Art und Weise durch unautorisierte Dritte möglich ist, dass keine digitalen Transaktionen durchgeführt werden können. Weiter fordert 2), dass auch dann, wenn ein unautorisierter Dritter keinen direkten Zugriff auf die Authentifizierungselemente im Sinne von 1) erlangen

kann, es ihm nicht möglich ist, eine Transaktion für den Nutzer transparent zu manipulieren. Aus dieser Anforderung folgt auch, dass zumindest eines der am digitalen Transaktionsprozess beteiligten Geräte über einen *Trusted Path* verfügt. Hiermit ist ein geschützter, vertrauenswürdiger Kanal gemeint, der zumindest Authentizität bezüglich der zum Zwecke der Durchführung der digitalen Transaktion ein- und ausgegebenen Daten garantiert, sodass eine sichere Verifikation und Bestätigung durch den Nutzer möglich ist [Zho+12; WW17].

Praktische Sicherheit. Nachdem wir also die Sicherheitsziele des Onlinebankings präzisiert haben, wenden wir uns nun den Umständen zu, die gelten müssen, damit man aus technischer Sicht von Sicherheit sprechen kann. Diese Umstände werden als einschränkende Annahmen über die Möglichkeiten des Angreifers beschrieben. Dies können nur Grundannahmen sein, die unter Beachtung des Standes der Technik auf absehbare Zeit gerechtfertigt sein werden. Die folgenden drei Annahmen sind ausreichend, um mit hinreichender Präzision Aussagen über die Sicherheit von Online- und Mobilebanking-Verfahren zu machen:

- Die eingesetzten kryptographischen Verfahren zur Gewährleistung der Authentizität sind nicht gebrochen.
- Geheimnisse können aus speziellen Geräten (wie beispielsweise einer Smartcard) trotz physischen Zugriffs nicht ausgelesen werden.
- In einer Software, die einen bestimmten Umfang nicht überschreitet, sind keine Schwachstellen vorhanden.

Es gibt viele Beispiele in der Praxis, die zeigen, dass die beschriebenen Annahmen durch Angreifer verletzt werden können. Dies ist insbesondere dann der Fall, wenn der Angreifer (in der Regel als Innentäter) die Chance hat, die initiale Einrichtung eines Verfahrens (etwa die Produktion von Geräten oder die Zustellung von Listen oder Geräten an den Kunden) zu manipulieren. Unter Beachtung der heutigen technischen und organisatorischen Möglichkeiten können Systeme jedoch so ausgestaltet werden, dass die Überwindung dieser Annahmen für den Angreifer in jedem Einzelfall einen hohen Aufwand bedeutet, so dass entsprechende Angriffe nicht ökonomisch im Massenzahlungsverkehr eingesetzt werden können. Es liegt in der Verantwortung der Bank, ständig die Verletzung dieser Grundannahmen zu prüfen und entsprechende Gegenmaßnahmen (wie der Austausch der verwendeten Hardware oder Kryptographie) zu ergreifen, sollten sie verletzt sein. Wir setzen diese drei Annahmen darum als gegeben voraus.

6.2 Regulatorische Voraussetzungen

Die PSD2 [ABl15a] ist eine EU-Richtlinie, die seit dem 13. Januar 2018 gilt und die Zahlungsdiensterichtlinie I (PSD1) [ABl15b], die den Europäischen Zahlungsraum SEPA geschaffen hat, ersetzt und ausbaut [BaFin17]. Die PSD2 ist eine umfangreiche Regulierung, die den EU-Binnenmarkt in Bezug auf elektronische Zahlungen weiterentwickeln soll. Eine besondere Bedeutung kommt internetbasierten Zahlungsverfahren und mobilen Endgeräten zu, die sich seit der PSD1 zwar stark verbreitet haben, regulatorisch aber nur unzureichend erfasst wurden. Die PSD2 schafft hier Regeln für alle Zahlungsdienste.

Neben dem Schließen regulatorischer Lücken hat die PSD2 auch das Ziel, den Wettbewerb zu fördern. Aus diesem Grund müssen die Banken Schnittstellen bereitstellen, die von Nichtbanken – oft handelt es sich hierbei um Fintechs – genutzt werden können. Voraussetzung ist eine Zulassung als Kontoinformations- bzw. Zahlungsauslösedienst. Ferner wird der Kunde in Haftungsfragen besser gestellt, indem die Selbstbeteiligung bei nichtautorisierten Zahlungen aufgrund einfacher Fahrlässigkeit auf maximal 50 € reduziert wird. Außerdem sollen die Kunden durch die Anwendung einer starken Kundenauthentifizierung (SCA) besser vor Betrug geschützt werden, um die Haftungsfrage erst gar nicht zu stellen.

Die Anforderungen an die SCA wirken sich gleichermaßen auf bereits existierende und zukünftige Sicherungsverfahren aus und sind deshalb für unsere Bewertung und den anschließenden Vergleich mit den allgemeinen Voraussetzungen zentral. Die Eigenschaften der SCA werden jedoch nicht in der Richtlinie selbst geregelt, sondern in einer an die Europäische Bankenaufsichtsbehörde (EBA) delegierten Verordnung. Die Ausgestaltung der sog. RTS [ABl18] war dabei äußerst kontrovers: Im Rahmen ihres Mandats hatte die EBA auf ihr Diskussionspapier 224 Rückmeldungen zu verkünden und damit so viele wie nie zuvor für ein Regulierungsvorhaben der Behörde [EBA17]. Die RTS sind am 13. März 2018 mit der Veröffentlichung im Amtsblatt der EU in Kraft getreten, gelten aber erst nach einer Übergangsfrist von 18 Monaten und damit zum 14. September 2019.

Im Folgenden beschreiben wir die Anforderungen an die SCA. Es ist dabei nicht das Ziel, die Bestimmungen vollumfänglich wiederzugeben. Stattdessen begrenzen wir uns auf die Aspekte, die für die Ausgestaltung der Sicherungsverfahren im Online- und Mobilebanking im Mittelpunkt stehen. Zum Schluss des Abschnitts vergleichen wir die regulatorischen mit den allgemeinen Anforderungen.

6.2.1 Mehrfaktorauthentifizierung

Obwohl der SCA ein zentraler Stellenwert beizumessen ist, wird sie im öffentlichen Diskurs gerne auf eine 2FA reduziert. Das ist umso bemerkenswerter, da sich jener Begriff weder in der Richtlinie noch in der Verordnung wiederfindet. Was die Regulierung fordert, ist eine Authentifizierung, die sich mindestens auf zwei unterschiedliche [EBA18a] Authentifizierungselemente aus den Kategorien Wissen (Art. 6), Besitz (Art. 7) und Inhärenz (Art. 8) stützt.

Wissen. Ein Geheimnis, das nur dem Zahler bekannt ist, gilt als Wissenselement. Das prototypische Beispiel für ein solches Element ist ein Passwort von hinreichender Komplexität. Die Zahlungsnetzwerke hätten hier auch gerne die Kreditkartennummer samt CVC2 als Wissenselement verstanden gewusst; eine Auffassung, der die EBA jüngst eine Absage erteilt hat [EBA18a].

Besitz. Es handelt sich um ein Besitzelement, insofern ein Medium zum Einsatz kommt, das sich nicht nur im Besitz des Kunden befindet, sondern auch Kopierversuchen widersteht. Allgemein als Besitzelement akzeptiert gelten in Hardware realisierte, dedizierte Programmbausteine, die kryptographisch sichere Operationen ausführen und somit den Besitznachweis erbringen. Ein gängiges Besitzelement, das diesem Prinzip folgt, ist beispielsweise eine Bankkarte, die ihre Authentifizierung über den eingebetteten Chip nach dem EMV-Standard abwickelt. Ob auch in Software realisierte Lösungen – wie sie insbesondere für mobilen Endgeräte zu finden sind – den Ansprüchen der starken Kundenauthentifizierung genügen, ist indes unklar und wird später noch Diskussionsgegenstand sein.

Inhärenz. Obwohl die Begriffe der Biometrie und Inhärenz oft synonym verwendet werden, meinen sie dennoch nicht dasselbe: Inhärenz ist grundsätzlich eine Eigenschaft, die der zu authentifizierenden Person untrennbar anhaftet. Dass Inhärenz mit Biometrie gleichgesetzt wird, ist durch die vorherrschende Inhärenzauthentifizierung auf Basis einer Erkennung des Fingerabdrucks oder des Gesichts zu erklären. Obwohl hinsichtlich ihrer Eignung umstritten, sind auch konforme Inhärenzverfahren denkbar, die sich z. B. auf Verhaltensweisen der zu authentifizierenden Person stützen. Im Gegensatz zu den Authentifizierungselementen der Kategorien Wissen und Besitz basieren Inhärenzverfahren immer auf der Anwendung einer Heuristik, die keine exakte Genauigkeit liefern kann. Insofern schreiben die RTS auch vor, dass ein konformes Inhärenzelement immer von einer vernachlässigbar geringen Wahrscheinlichkeit gekennzeichnet sein muss, die die fälschliche Authentifizierung eines illegitimen Nutzers zulässt.

6.2.2 Unabhängigkeit

Die RTS fordern neben der Anwendung unterschiedlicher Authentifizierungselemente, dass diese voneinander unabhängig sind (Art. 9). Das bedeutet, dass die einzelnen Faktoren so ausgestaltet sein müssen, dass die Kompromittierung eines der Elemente nicht auch automatisch die Kompromittierung der anderen Elemente nach sich zieht. Dieser Aspekt ist dann von besonderer Bedeutung, wenn eines oder gar mehrere Elemente über ein Mehrzweckgerät – wie es z. B. PCs und Smartphones sind – verwendet werden. Um den dadurch entstehenden Risiken entgegenzuwirken, sieht die Verordnung verschiedene Maßnahmen vor: Zum einen ist die „Nutzung getrennter sicherer Ausführungsumgebungen“ (Absatz 3 Satz b) vorgeschrieben. Zum anderen müssen die Zahlungsdienstleister Maßnahmen ergreifen, die Manipulationen durch Dritte oder auch durch den Kunden selbst erkennen und negative Folgen für die Zuverlässigkeit der Verfahren abwenden.

Der letzte Punkt zielt auf bewusste Veränderungen am System ab, wie sie sich z. B. durch das Rooting/Jailbreaking ergeben. Welche Schritte die Zahlungsdienste ergreifen müssen, damit auch ungewollte Manipulationen durch Dritte mit hinreichender Zuverlässigkeit erkannt werden, bleibt ungeklärt. Wie in Abschnitt 3.2 ausgeführt, ist eine Erkennung schwierig bis unmöglich. Wie eine angemessene Reaktion auf eine ermittelte Manipulation auszusehen hat, ist ebenfalls diffus. Ähnlich verhält es sich mit der Forderung nach zwei getrennten sicheren Ausführungsumgebungen. Weitgehend gesichert scheint nur, dass Verfahren, die die Authentifizierungselemente über ein jeweils separates Endgerät realisieren, die Bedingung erfüllen. Was unter der Verwendung durch Mehrzweckgerät zu verstehen ist, lässt jedoch viel Interpretationsspielraum offen. Es ist recht eindeutig, dass Besitz- und Inhärenzelemente eine technische Infrastruktur benötigen, die durchaus auch durch ein Mehrzweckgerät realisiert werden kann; anschauliche Beispiele finden sich für mobile Endgeräte. Demnach wären entsprechende Maßnahmen zur Eindämmung von System- und Gerätemanipulationen gefordert. Wie es sich hingegen mit Wissenselementen verhält, ist weniger klar; die Eingabe und damit auch die Verarbeitung erfolgt in aller Regel durch ein Mehrzweckgerät. Dem Wortlaut nach wären also auch hier entsprechende Gegenmaßnahmen zu ergreifen. Wie dies technisch z. B. aus einem Webbrowser heraus möglich sein soll, ist völlig offen und erscheint auf den nach dem Stand der Technik gängigen Systemen gar unmöglich. Wir bemühen deshalb eine pragmatischere Lesart, die die Anwendung solcher Maßnahmen nur dann fordert, wenn mehrere Authentifizierungselemente durch dasselbe Mehrzweckgerät realisiert werden.

6.2.3 Authentifizierungscode

Die Anwendung der Authentifizierungselemente muss zur Generierung eines Authentifizierungscodes führen, der vom Zahlungsdienstleister nur einmalig akzeptiert wird und nicht gefälscht werden kann. Dabei muss der Authentifizierungscode derart gestaltet sein, dass sich aus ihm weder erkennen lässt, wie sich weitere Authentifizierungscodes erstellen lassen, noch darf er Rückschlüsse auf Zahlungsdetails oder den Zahler selbst zulassen. Weitere Schutzmaßnahmen müssen sicherstellen, dass der Authentifizierungscode nach höchstens fünf Minuten seine Gültigkeit verliert und maximal fünf Fehlversuche zulässig sind.

Dieses Konzept ist im deutschen Onlinebanking hinlänglich bekannt: dort werden Zahlungen bereits seit jeher durch eine TAN bestätigt, die durch das Sicherungsverfahren erzeugt oder empfangen werden. Es ist noch weitgehend gängige Praxis, dass der Authentifizierungscode in Form einer sichtbaren TAN, die sogar auch noch oft eigens durch den Zahler übertragen wird, auftritt. Es ist jedoch weder vorgeschrieben, dass der Authentifizierungscode für den Kunden sichtbar ist, noch, dass dieser zwangsläufig durch ein Einmalpasswort realisiert wird. So entsprechen auch Verfahren den Anforderungen, die einen Authentifizierungscode im Hintergrund am Transaktionsprozess beteiligen und dabei z. B. auf kryptographische Signaturen zurückgreifen. Auch aus Sicht der Sicherheit ist das Fehlen einer sichtbaren TAN unbedenklich (vgl. Abschnitt 2.3) und kann sogar einen positiven Effekt haben. Denn die Notwendigkeit zur manuellen Übertragung der TAN erwuchs nur aus dem damals aus technisch-ökonomischen Gesichtspunkten noch nicht überbrückbaren Medienbruch, sorgt aber dafür, dass die Kunden die TAN – und nicht etwa die Kontrolle der Transaktionsdetails im Sicherungsverfahren – als wesentlich erachten.

6.2.4 Dynamische Verknüpfung

Der ausgestellte Authentifizierungscode muss derart mit dem Zahlungsempfänger und -betrag dynamisch verknüpft sein (Art. 5), dass dieser nur genau für diesen Auftrag gültig ist. Eine Änderung des Zahlungsempfängers oder -betrags muss zu einer Invalidierung des Authentifizierungscodes führen. Darüber hinaus müssen die Vertraulichkeit, die Authentizität und die Integrität aller Daten gewahrt werden, die dem Kunden zum Zwecke der Auftragsauslösung angezeigt werden. Hiervon sind insbesondere der Zahlungsempfänger und -betrag betroffen, da beide Informationen dem Zahler zwingend anzuzeigen sind.

Eine Verknüpfung des Authentifizierungscodes an den Zahlungsempfänger und -betrag ist bei den meisten Sicherungsverfahren bereits heute gewährleistet. Neu ist hingegen die Anforderung, dass Vertraulichkeit, Authentizität und Integrität ebenfalls garantiert sein müssen. Um dieser Anforderung gerecht zu werden, muss das Legitimierungsverfahren eine sichere Anzeige bieten, die insbesondere für Mehrzweckgeräte eine Herausforderung darstellt.

6.2.5 Ausnahmen

Obwohl unsere Betrachtung grundsätzlich von einem Fall ausgeht, der eine SCA fordert, formulieren die RTS auch eine Reihe von Ausnahmen, die es erlauben, hiervon abzusehen. Die Ausnahmen sollen insgesamt und jeweils für sich einen Kompromiss zwischen Anwenderfreundlichkeit und geringem Betrugs- und Schadensrisiko darstellen.

Lesender Zugriff. Das bestehende Paradigma, das die Authentifizierung mit einem Element nur lesenden Zugriff erlaubt und Änderungen grundsätzlich ein weiteres fordern, bleibt durch die RTS nicht unangetastet: initial und spätestens nach 90 Tagen ist auch zum Abruf des Kontostands und von Umsätzen eine SCA notwendig. In der Praxis bedeutet das, dass der Kunde in Zukunft auch beim Login in regelmäßigen Abständen sein Sicherungsverfahren anwenden muss. Es steht der kontoführenden Bank jedoch frei, ob es die Ausnahmeregel beansprucht und 90 Tage auf eine SCA verzichtet oder diese bei jedem Zugriff verlangt.

Dieser Punkt ist für Kontoinformationsdienste, die eine sog. Multibanking-Funktionalität bieten, von herausragender Bedeutung. Dabei handelt es sich um einen Dienst, der die Konten verschiedener Kreditinstitute im Auftrag des Kunden abruft und aggregiert. Die RTS schränken diesen Dienst deutlich ein, da in regelmäßigen Abständen eine SCA für jedes Institut erfolgen muss. Aus Konsistenzgründen ist es sogar sinnvoll, wenn die kontoführenden Institute immer eine SCA fordern. Dadurch ergäbe sich aus Kundensicht ein einheitliches und erwartbares Verhalten der Bank. Auf der anderen Seite führt der Verzicht auf die Ausnahme dazu, dass Multibanking-Anwendungen vollständig ad absurdum geführt werden, da potenziell jeder Abruf pro Konto eine SCA nötig machen würde, die noch dazu sehr unterschiedlich ablaufen kann. Aus regulatorischer Sicht möglich, aus praktischen Gesichtspunkten jedoch unwahrscheinlich, ist die Auslagerung der SCA an den Kontoinformationsdienst.

Vertrauenswürdige Empfänger. Eine Ausnahme sieht vor, dass auf eine SCA verzichtet werden kann, insofern der Zahlungsempfänger zuvor durch Anwendung einer SCA auf eine Liste vertrauenswürdiger Empfänger gesetzt wurde. Bei Transaktionen auf ein anderes Konto, das jedoch beim gleichen Zahlungsdienstleister geführt wird, kann grundsätzlich auf eine SCA verzichtet werden.

Die Regelung scheint sinnvoll und auch aus Sicherheitssicht ausbalanciert. Problematisch ist die Regelung eher aus Wettbewerbssicht, weil sie große Händler im E-Commerce begünstigt, aber kleinere Händler benachteiligt. Es gibt von Seiten der Banken auch keine Pflicht, eine Schnittstelle anzubieten, die das Markieren des Empfängers als vertrauenswürdig vorsieht. Stattdessen muss der Kunde den entsprechenden Prozess voraussichtlich selbstständig bei der Bank anstoßen. Daher ist es wahrscheinlicher, dass der Kunde einen Händler auf die Liste der vertrauenswürdigen Empfänger setzt, bei dem er regelmäßig einkauft.

Daueraufträge. Bei wiederkehrenden Zahlungsaufträgen sind die Vorgaben wie erwartet: Daueraufträge mit einem bestimmten Empfänger und Betrag müssen durch den Kunden initial mit einer SCA freigegeben werden, finden dann aber autonom zu den gewählten Zeitpunkten statt.

Kleinbetragszahlungen. Die RTS sehen Ausnahmen für Transaktionen mit geringen Beträgen vor. Demnach können Zahlungsdienstleister bei Transaktionen mit einem Zahlungsbetrag von bis zu 30 € von der Anwendung einer SCA absehen. Diese Regelung kann jedoch nicht beliebig oft beansprucht werden, sondern ist von einer Grenze gedeckelt; danach wird wieder eine SCA notwendig. Hierbei kann der Zahlungsdienstleister zwischen zwei Schwellen wählen: maximal fünf aufeinanderfolgende oder beliebig viele Transaktionen bis zu einem akkumulierten Höchstbetrag von 100 € können ohne SCA durchgeführt werden. Die Bedingung, dass jede Überweisung für sich die Grenze von 30 € nicht überschreiten darf, bleibt hiervon unberührt.

Ob sich aus dieser Regelung ein Sicherheitsproblem ergibt, hängt wesentlich davon ab, ob und wie die Banken von der 90-Tage-Ausnahme Gebrauch machen. Wird pauschal 90 Tage auf eine SCA bei jedwedem Login verzichtet, kann ein Angreifer mit nur einem Authentifizierungselement bis zu 150 € pro Kunde erbeuten. Bei einem groß angelegten Phishing-Angriff kann sich daraus für Kriminelle eine durchaus attraktive Gesamtsumme ergeben.

Transaktionsrisikoanalyse. Abgesehen von den bisherigen, recht spezifischen Ausnahmeregelungen kann ein Zahlungsdienstleister auch dann von einer SCA

absehen, wenn er ermittelt hat, dass der Zahlungsauftrag mit einem geringen Risiko verbunden ist. Um auf Basis der Transaktionsrisikoanalyse von der SCA abzusehen, muss eine Vielzahl an Parametern berücksichtigt werden. Hierunter fällt insbesondere das bisherige Zahlungsverhalten. Von der Ausnahme kann nur bei Zahlungen bis 500 € Gebrauch gemacht werden. Darüber hinaus muss der Zahlungsdienstleister bestimmte Referenzbetrugsraten einhalten, die für Überweisungen besonders streng gewählt werden. Welchen Einfluss diese Ausnahmeregelung auf die Anwendungspraxis der SCA haben wird, bleibt abzuwarten.

6.2.6 Vergleich zu den allgemeinen Anforderungen

Im Unterschied zu unseren allgemeinen Anforderungen sind die soeben geschilderten rechtlichen Voraussetzungen deutlich spezifischer. Auch fordern die RTS explizit eine Mehr-, jedoch mindestens eine Zwei-Faktor-Authentifizierung. Obwohl diese Vorgabe aus Sicht der IT-Sicherheit zweifelsohne sinnvoll ist, folgt sie aus unseren Anforderungen nicht zwangsläufig, insofern beiden allgemeinen Bedingungen erfüllt sind.

Dennoch lassen sich die Vorgaben durch die RTS jeweils unseren allgemeinen Voraussetzungen an die Sicherheit zuordnen. So entsprechen die Anforderungen an die Authentifizierungselemente, den Authentifizierungscode und die Unabhängigkeit weitgehend unserer Vorgabe, dass die Elemente durch einen unautorisierten Dritten nicht zugreifbar sein dürfen. Die Anforderung an die dynamische Verknüpfung der Transaktion geht sogar noch weiter als unsere Forderung nach Manipulationsfreiheit. Während wir nur einen Trusted Path voraussetzen, damit dem Zahler eine lückenlose Transaktionsverifikation möglich ist, fordert die Regulierung neben Authentizität auch Integrität und Vertraulichkeit. In unserer Anforderung kommt die Integrität nicht vor, da diese vom Nutzer zu prüfen ist. Einzige Voraussetzung ist, dass ein Nutzer auch genau das bestätigt bzw. nicht bestätigt, was ihm dargestellt wird. Obwohl Vertraulichkeit zweifelsfrei ein erstrebenswertes Gut ist, das in der Praxis stets gewahrt werden sollte, spielt sie für die technische Transaktionssicherheit eine untergeordnete Rolle.

6.3 Konformität etablierter Sicherungsverfahren

Im Folgenden werden die gängigen Sicherungsverfahren im Online- und Mobilebanking hinsichtlich ihrer Konformität zu den RTS bewertet.

6.3.1 Gegenstand

Laut einer repräsentativen Umfrage im Auftrag der Norisbank waren das sms- (36%), chip- (31%), und iTAN-Verfahren (25%) Ende 2016 die geläufigsten Verfahren [NB16]. Noch am wenigsten verbreitet waren App-basierte Sicherungsverfahren (8%). Die Verteilung ist unter den Instituten jedoch recht verschieden. Sowohl die Sparkassen als auch die Genossenschaftsbanken haben die TAN-Listen bereits vollständig abgeschafft.

In einer von uns gestellten Anfrage vom 6. April 2018 an die DK nannte uns der Bundesverband der Deutschen Volks- und Raiffeisenbanken (BVR) die folgende Verteilung unter seinen knapp 18 Millionen Mitgliedern zum Stichtag 31. Dezember 2016: 49% chipTAN, 41% smsTAN und 5% App-basiertes Sicherungsverfahren. Die restlichen 5% entfallen auf Kunden, die den HBCI/FinTS-Standard nutzen.

Auch der Deutscher Sparkassen- und Giroverband (DSGV) erteilte uns auf die gleiche Anfrage Auskunft, lieferte aber aktuellere Zahlen mit Stand vom 23. April 2018. Demnach entfielen unter den rund 50 Millionen Kunden 51% auf das chip- und 31% auf das smsTAN-Verfahren. Das App-basierte Sicherungsverfahren der Sparkassen machte bereits 17% aus. Nachdem die Kundengruppen der Genossenschaftsbanken und Sparkassen durchaus vergleichbar sind, zeigt sich, dass sich insbesondere die App-basierten Verfahren im Aufwind befinden.

Der BDB konnte uns keine Zahlen zur Verteilung der Sicherungsverfahren bei den privatrechtlich organisierten Instituten nennen. Die einzelne Privatbanken setzen zum Teil auf spezielle Sicherungsverfahren, die in ihrem Gesamtanteil aber nur eine geringfügige Bedeutung spielen. Sie werden deshalb im Folgenden nicht gesondert betrachtet. Stattdessen liegt der Fokus auf den klassischen TAN-Verfahren iTAN, smsTAN und chipTAN sowie auf den vergleichsweise neuen App-basierten Legitimierungsverfahren.

6.3.2 Kriterien

Die RTS wollen zwar spezifische Vorgaben für eine konforme Lösung für die SCA beschreiben, zielen aber gleichzeitig darauf ab, möglichst allgemeingültig und technologieneutral zu bleiben. Dieser Balanceakt scheint vor dem Hintergrund sinnvoll, dass die Anforderungen der RTS auch für Lösungen anwendbar sein sollen, die potenziell deutlich anders funktionieren, als das zum Zeitpunkt des Entwurfs durch die EBA der Fall war. Dennoch ist es für unsere Bewertung sinnvoll, die allgemeinen Anforderungen in für die Verfahren spezifischere Kriterien zu überführen. Dabei sollen die Kriterien hinreichende Bedingungen sein, um Konformität zu erreichen.

Nichtkopierbarkeit. Den Verfahren ist gemein, dass sie alle ein Authentifizierungselement der Kategorie Besitz darstellen wollen. Gemäß den RTS muss es sich dabei um ein Element handeln, das ausschließlich im Besitz des Kunden ist. Darüber hinaus muss ein Besitzfaktor so gestaltet sein, dass er sich unter praktischen Gesichtspunkten nicht kopieren lässt. Demnach muss ein Besitzelement das Kriterium der Nichtkopierbarkeit erfüllen.

Auftragsbindung. Zwischen den Zahlungsdaten und dem Authentifizierungscode muss eine dynamische Verknüpfung bestehen. Das bedeutet, dass der Authentifizierungscode spezifisch für einen bestimmten Auftrag entsteht. Ein bestimmter Authentifizierungscode kann so nur einen bestimmten Auftrag freigeben. Das Sicherungsverfahren muss also eine Auftragsbindung sicherstellen.

Authentifizierungscode. Die Anwendung des Sicherungsverfahrens muss einen Authentifizierungscode nach sich ziehen. Es ist dabei unerheblich, ob das Legitimierungsverfahren den Code abruft, oder ihn eigens erstellt. In jedem Fall müssen die Vertraulichkeit, Integrität und Authentizität des Codes gewährleistet sein.

Übertragungshoheit. Die Anforderung an die dynamische Verknüpfung fordert ebenfalls, dass die Vertraulichkeit, Integrität und Authentizität des Zahlungsempfängers und -betrags gewahrt bleiben. Das bedeutet, dass der Zahlungsdienstleister den Übertragungskanal derart kontrollieren muss, dass die Zahlungsdaten von Dritten weder gelesen noch verändert werden können. Faktisch wird hierfür eine authentifizierte Kryptographie auf all jenen Geräten notwendig, die sich nicht unter der Kontrolle des Zahlungsdienstleisters oder des Kunden befinden.

Sichere Anzeige. Im Rahmen der Anforderung zur dynamischen Verknüpfung muss der Prozess der SCA dem Kunden zumindest den Zahlungsbetrag und -empfänger anzeigen. Hierfür sind Maßnahmen zu ergreifen, die die Vertraulichkeit,

	Onlinebanking				Mobilebanking	
	iTAN	smsTAN	chipTAN	2GA	2AA	1AA
Nichtkopierbarkeit	○	●	●	●	●	●
Auftragsbindung	○	●	●	●	●	●
Auth'code	●	●	●	●	●	●
Übertragungshoheit	●	○	●	●	●	●
Sichere Anzeige	○	○	●	●	○	○

Tabelle 6.1: Eigenschaften der gängigen Sicherungsverfahren.

Integrität und Authentizität dieser Zahlungsdaten gewährleisten. Hieraus lässt sich ableiten, dass das Sicherungsverfahren eine sichere Anzeige bieten muss. Eine solche Anforderung ist evident, da ein Kunde den Auftrag sonst nicht zuverlässig verifizieren kann.

Ein gangbarer Weg, um diese Anforderung zu konkretisieren, scheint die Forderung nach einem zweiten Anzeigekanal zu sein, der unabhängig von den anderen Authentifizierungselementen betrieben wird. Hierbei kann es sich um ein weiteres Gerät mit Display handeln, das idealerweise nur dem Zweck der Transaktionsbestätigung dient. Ein separates Gerät ist jedoch nicht zwangsläufig erforderlich: Es ist auch denkbar, dass derselbe Bildschirm von einer weiteren, vertrauenswürdigen und vollständig separaten Ausführungsumgebung angesteuert wird. Eine solche Absicherung muss aber in jedem Fall durch Hardwaremechanismen realisiert sein, sodass sich eine zuverlässige Isolation ergibt.

Zusammengefasst bleiben aus praktischen Gesichtspunkten zwei Möglichkeiten, um Konformität im Sinne einer sicheren Anzeige zu erreichen: Entweder erfolgen die Transaktionsauslösung und -bestätigung über zwei getrennte Geräte oder das Gerät bietet eine Möglichkeit zur Transaktionsbestätigung, die dem Kunden die Auftragsdaten genau so anzeigt, wie sie letztendlich bei der Bank eingehen.

6.3.3 Bewertung

Im Nachfolgenden prüfen wir, ob die gängigen Sicherungsverfahren im Online- und Mobilebanking unseren Kriterien entsprechen und somit Konformität zu den RTS erreichen. Die Ergebnisse dieser Bewertung sind in Tabelle 6.1 zusammengefasst.

iTAN

Die iTAN-Liste erfüllt die Voraussetzungen an ein Besitzelement nicht. Dadurch, dass sie auf Papier gedruckt und den Kunden zugestellt wird, lässt sie sich trivial kopieren. Auch der Versuch, die Liste einer anderen Kategorie zuzuordnen, scheitert: Um ein Inhärenzelement kann es sich augenscheinlich nicht handeln. Die These, bei der iTAN handele es sich um ein Wissenselement, schlägt ebenso fehl. Zunächst scheint es kohärent, einen Vergleich z. B. mit der Karten-PIN herzustellen, die ebenfalls per Post zugestellt wird. Im Unterschied zur TAN-Liste ist der Kontoinhaber aber dazu angehalten, die Karten-PIN umgehend zu memorieren und das Trägermedium restlos zu vernichten. Erst dadurch entsteht ein Geheimnis, das – neben der Bank – nur dem Kunden bekannt ist. Die gleiche Intention gibt es zu keiner Zeit für die TAN-Liste. Es kann sich also auch nicht um ein Wissenselement handeln.

Ebenso existiert keine gültige Auftragsbindung. Die Argumentation, die TAN würde über ihren Index mit einem konkreten Auftrag verknüpft, ist zwar korrekt, erreicht aber keine Konformität im Sinne der Anforderung. Die TANs liegen bereits zum Druckzeitpunkt statisch vor und sind somit kein Resultat eines konkreten Auftrags. Infolgedessen kann ein Gegenspieler die TANs auf Vorrat sammeln. Eine gültige Auftragsbindung ergibt sich dadurch nicht. Eine Anzeigemöglichkeit der Transaktionsdaten fehlt offensichtlich vollständig.

Konformität erreicht die iTAN-Liste hingegen zu den Voraussetzungen des Authentifizierungscodes und der Hoheit über den Übertragungsweg. Insgesamt ist das iTAN-Verfahren jedoch aus mehreren Gründen als nicht konform zu erachten. An dieser Auffassung rütteln auch die betroffenen Banken nicht, weshalb die Abschaffung der Liste Anfang 2019 in vollem Gange ist [Sei19].

smsTAN

Die smsTAN erfüllt die Voraussetzung an die Nichtkopierbarkeit. Grund ist, dass es sich bei der SIM-Karte um eine Smartcard handelt, die sich de facto nicht kopieren lässt. An diesem Umstand ändern auch die Schadensfälle nichts, bei denen es Kriminellen gelungen war, eine SIM-Karten-Dublette im Namen des Opfers zu bestellen (vgl. Abschnitt 2.2.2). Es war also nicht gelungen, die SIM-Karte eines Opfers zu replizieren, sondern nur, eine neue anzufordern. Eine Verletzung der Nichtkopierbarkeit liegt dadurch nicht vor. Auch eine Auftragsbindung bietet das Verfahren, da die TAN in der SMS nur für die beigefügten Auftragsdaten gilt.

Eine sichere Anzeige ist bei der smsTAN nicht durchgehend gegeben. Wie in Abschnitt 2.4 dargestellt, verbieten die Institute über ihre AGB, dass der Kunde die SMS auf demselben Endgerät empfängt, das auch den Auftrag ausgelöst hat. Dabei handelt es sich jedoch nicht um eine funktionale, sondern um eine rechtliche Einschränkung des Verfahrens. Um ein konformes Verfahren zu erreichen, müsste die Bank also sicherstellen, dass der Kunde niemals dasselbe Endgerät für die Transaktionsauslösung und -bestätigung heranzieht. Es gibt zwar durchaus Kunden, die ein Featurephone zum Empfang der SMS verwenden, aber diese Kundengruppe stellt in der Vorherrschaft der Smartphones mittlerweile eine Minderheit dar. Insgesamt kann das smsTAN-Verfahren für bestimmte Kunden eine sichere Anzeige bieten, muss dies aber nicht zwangsläufig. In jedem Fall hat die Bank hierauf keinen Einfluss, weshalb in toto nicht von einer sicheren Anzeige gesprochen werden kann.

Selbst wenn man das Vorhandensein einer sicheren Anzeige anerkennen würde, ergäbe sich dadurch noch kein konformes Verfahren. Es ist unstrittig, dass die Banken über die Zustellung der SMS keine Kontrolle in dem Sinn haben, dass Vertraulichkeit, Integrität und Authentizität gewahrt sind. Im Gegenteil: es lassen sich mehrere konkrete Schwachstellen benennen, die zeigen, dass keines der geforderten Sicherheitsziele erfüllt wird [Sah+17]. Eine SMS ist maximal bis zur Basisstation des Mobilfunkanbieters verschlüsselt. Im Anschluss liegt die Nachricht im kompletten Netzwerk unverschlüsselt vor. Von Wahrung der Vertraulichkeit der enthaltenen Informationen – also zumindest Zahlungsempfänger, -betrag und TAN – kann vor diesem Hintergrund nicht gesprochen werden. Die Mobilfunkinfrastruktur sieht auch keinerlei Authentizitäts- oder Integritätsgarantien bzgl. der enthaltenen Informationen und des Absenders vor. Ein Gegenspieler, der die Nachricht lesen kann, kann sie auch beliebig manipulieren.

Insbesondere die fehlende Vertraulichkeit ist beim smsTAN-Verfahren fatal, da die zur Transaktionsbestätigung notwendige TAN Teil der Nachricht ist und nicht etwa erst auf dem empfangenden Endgerät erzeugt wird. Ist ein Angreifer in der Lage, den Inhalt der SMS zu lesen, kann er beliebige Transaktionen freigeben. Abschnitt 2.2.2 hat bereits gezeigt, dass es in diesem Bezug auch schon zu konkreten Schadensfällen gekommen ist. Insgesamt kann die smsTAN also nicht als ein konformes Sicherungsverfahren erachtet werden.

Die DK nimmt indes eine andere Position ein und betonte Ende 2018 gegenüber der Stiftung Warentest, dass das smsTAN-Verfahren die regulatorischen Anforderungen erfülle und insbesondere ein gültiges Besitzelement sei [Bac18]. Die Eignung der SIM-Karte als Besitzelement wird von uns jedoch nicht infrage gestellt. Auch die

EBA stellt ein gültiges Besitzelement im Sinne der Anforderung fest, merkt jedoch gleichzeitig an, dass sich allein dadurch keine Konformität ergibt [EBA18b] und weist auf die in Artikel 22(1) geforderte Vertraulichkeit und Integrität hin. Eine belastbare Einschätzung der EBA ergibt sich dadurch zwar noch nicht, zeigt aber, dass die Einschätzung der DK zu kurz greift.

chipTAN

Das chipTAN-Verfahren bietet Konformität zu all unseren Kriterien. Das Sicherungsverfahren verwendet ein dediziertes Lesegerät zusammen mit der persönlichen Bankkarte des Kunden. Dabei enthält die Karte eine Smartcard mit einem Geheimnis, mit dessen Hilfe eine Signatur über die Zahlungsdaten gebildet und in Form einer TAN ausgegeben wird. Ähnlich zur SIM-Karte beim smsTAN-Verfahren ist dieser Chip unter praktischen Gesichtspunkten nicht kopierbar.

Die TAN entsteht beim chipTAN-Verfahren nur auf Basis der Kontonummer des Empfängers, des Betrags und eines Start-Codes, der als Nonce fungiert. Es herrscht also volle Auftragsbindung. Eine digitale Übertragung von Zahlungsdaten ist beim chipTAN-Verfahren nicht immer notwendig: Start-Code, Kontonummer und Betrag können auch manuell in das Lesegerät eingegeben werden. Es gibt daneben auch eine Reihe halb- und vollautomatischer Übertragungswege vom Endgerät des Nutzers zum Lesegerät. Diese sind nicht notwendigerweise verschlüsselt, finden aber vollständig in der Sphäre des Kunden statt. Dementsprechend ist auch die Hoheit über den Übertragungsweg gewahrt. Dadurch, dass das Lesegerät für das chipTAN-Verfahren dediziert für die Bestätigung von Transaktionen entworfen wurde, bietet es eine vertrauenswürdige und sichere Anzeige.

App-basierte Verfahren

Besonders kontrovers diskutiert wurden die Anforderungen an App-basierte Sicherungsverfahren, haben die Institute in den vergangenen Jahren doch beträchtliche Ressourcen in deren Entwicklung und Vermarktung investiert. In Abschnitt 2.3 haben wir die verschiedenen Klassen App-basierter Verfahren – 2GA, 2AA und 1AA – vorgestellt, die regulatorisch zum Teil unterschiedlich zu bewerten sind.

Abschnitt 3.4.1 hat deutlich gemacht, dass sich eine zuverlässige Gerätebindung auf gängigen Smartphones grundsätzlich bewerkstelligen lässt, wenn dabei auf

entsprechende Hardwaremöglichkeiten zur An- und Ablage kryptographischen Schlüsselmaterials zurückgegriffen wird. Umgekehrt haben wir auch gezeigt, dass das Bilden eines Gerätefingerabdrucks keine adäquate Möglichkeit ist, um Nichtkopierbarkeit herzustellen. Letztendlich lässt sich aber konstatieren, dass es grundsätzlich möglich ist, App-basierte Verfahren aller Klassen so zu gestalten, dass sie auf den meisten gängigen mobilen Endgeräten Konformität zur Nichtkopierbarkeit erreichen. Dasselbe Schlüsselmaterial kann nicht nur dazu verwendet werden, einen Authentifizierungscode zu erzeugen, der aus den konkreten Auftragsdaten entsteht, sondern auch dafür, um einen symmetrischen Sitzungsschlüssel auszutauschen, mit dem die Kommunikation abgesichert wird. Demnach wäre auch die von uns geforderte Übertragungshoheit erfüllt.

Eine ganz andere Situation stellt sich aber für die Anforderung nach einer sicheren Anzeige ein. Nachdem wir eine Anzeige auch dann als sicher deklariert haben, wenn die Bestätigung der Daten auf einem zum transaktionsauslösenden verschiedenen Gerät erfolgt, ist auch das 2GA-Verfahren als konform zu unserem Kriterium zu betrachten. Es wäre jedoch vermessen, ein äquivalentes Sicherheitsniveau zu konstatieren, wie es z. B. das dedizierte Lesegerät des chipTAN-Verfahrens bietet. Dennoch ist es evident, dass es im Allgemeinen mehr Aufwand bedarf, zwei Geräte statt nur ein einziges zu kompromittieren.

Von eben diesem Umstand profitieren die Mobilebanking-Verfahren der 2AA und 1AA hingegen nicht. Da beide Verfahren auf ein und demselben Mobilgerät realisiert werden, müsste es für das System eine Hardwaremöglichkeit zur Darstellung und Bestätigung des Zahlungsempfängers und -betrags geben, wie wir sie zuvor in Abschnitt 3.4.2 skizziert haben. Eine solche Funktion bietet sich zum Stand April 2019 jedoch nur für eine erlesene Auswahl an Android-Geräten, die bereits im Auslieferungszustand die bis dato neuste Version 9 (Pie) bieten. Vor diesem Hintergrund kann nicht davon gesprochen werden, dass zumindest Android auch für die 1AA und 2AA eine sichere Anzeige auf einem Gerät bietet. Davon abgesehen ist zur gleichen Zeit eine vergleichbare Lösung für iOS weder verfügbar noch angekündigt. In letzter Konsequenz kann den Mobilebanking-Verfahren noch keine Möglichkeit zur sicheren Anzeige der Transaktionsdaten attestiert werden.

An dieser Tatsache ändern auch die in Kapitel 4 als schwer haltbar identifizierten Versprechen der Härtungsindustrie nichts. Gleichwohl legt Artikel 9(2) nahe, dass App-basierte Verfahren unabhängig von ihrer Authentifizierungsklasse speziell zu sichern seien. Es ist ein zentrales Verkaufsargument der Anbieter softwarebasierter App-Härtungslösungen, dass sie Geräte- und Systemmanipulationen erkennen (Satz

b) und auch dann einen sicheren Betrieb gewährleisten, insofern sich dadurch Sicherheitsimplikationen ergeben (Satz c). Wie im entsprechenden Kapitel dargelegt, kann es durchaus sinnvoll sein, eine App neben dem Einhalten von Best Practices zusätzlich zu härten. Dieser Softwareschutz kann jedoch nicht dazu führen, dass ein aufgrund von Strukturproblemen nicht erfülltes Kriterium nach Anwendung der Lösung konform zu einer Anforderung wird. Die Härtungsmaßnahmen können also nur einen komplementären Schutz bieten.

Obwohl die 1AA und 2AA gleichermaßen keine sichere Anzeige leisten können, ist das Sicherheitsniveau bei der 1AA als geringer einzustufen als bei der 2AA, da Transaktionsauslösung und -bestätigung nicht durch zwei separate Apps realisiert sind und deshalb auch nicht von der Isolation durch das Sandboxing der mobilen Betriebssysteme profitieren. Durch das Fehlen dieser Schutzmaßnahme reicht es bei einer 1AA unter Umständen, nur eine Sicherheitslücke in der betreffenden App auszunutzen, um das System vollständig zu kompromittieren. Bei einer 2AA wäre hierfür entweder eine Rechteausweitung, wie wir sie in Kapitel 3 angenommen haben, oder eine entsprechend schwerwiegende Sicherheitslücke in beiden Apps notwendig. Diese Argumentation scheinen auch die RTS anzuerkennen, indem sie die „Nutzung getrennter sicherer Ausführungsumgebungen durch die im Mehrzweckgerät installierte Software“ (Artikel 9(3)a) vorschreiben. Dieser Vorgabe ist mit der Implementierung von zwei Apps entsprochen. Dennoch wäre es auch denkbar, dass sich durch Sandboxing auf App-Ebene zwei getrennte Ausführungsumgebungen erzeugen lassen, die den regulatorischen Ansprüchen genügen. Zumindest unter Android hat die Forschung schon entsprechende Ansätze gezeigt [Bac+15]. Inwiefern sich ein vergleichbarer Isolationsmechanismus innerhalb einer iOS-App erreichen lässt, ist ein noch offener Forschungsgegenstand.

Ein weiterer Diskussionspunkt in Bezug auf App-basierte Verfahren ist der zunehmende Verzicht auf den Wissensfaktor, der stattdessen durch ein Inhärenzelement abgebildet wird. Gängig sind im Jahr 2019 vor allem noch Biometrielösungen, die auf die Erkennung des Gesichts oder Fingerabdrucks des Geräteinhabers abzielen. Hier stellen sich zwei technische Herausforderungen: Erstens verlassen die Sensordaten schon aus Gründen der Privatsphäre das Gerät nicht [Bia+18]. Infolgedessen müssen Entwickler die entsprechenden Programmierschnittstellen der Plattform nutzen, führen selbst also keine biometrischen Erkennungsroutinen durch. Die Güte des Hardwaresensors – z. B. Auflösung beim Erfassen des Fingerabdrucks – und der Implementierung – also der eigentliche Algorithmus, der entscheidet, ob der gelesene mit dem hinterlegten Fingerabdruck übereinstimmt – sind aber insbesondere im heterogenen Android-Markt sehr unterschiedlich. Die RTS würdigen diesen Umstand

in Artikel 8(1) und schreiben deshalb vor, dass eine „sehr geringe Wahrscheinlichkeit besteht, dass ein Unbefugter als Zahler authentifiziert wird". Ein quantitativ messbarer Wert ergibt sich dadurch aber nicht. In einer Arbeitskreissitzung des Bitkom am 7. Dezember 2018 deutete die BaFin in einem Vortrag an, dass sie die Vorgabe einer Falschpositivschwelle plant, die eine ähnlich geringe Wahrscheinlichkeit der fälschlichen Authentifizierung bietet wie das zufällige Erraten eines Wissenselements. Dieser Ansatz überrascht schon vor dem Hintergrund, dass es für Wissenselemente keine entsprechende Bemessungsgrundlage gibt, die so einen Vergleich zulassen würde.

Zweitens besteht auf den Mobilgeräten oft die Möglichkeit, mehrere Merkmale zu hinterlegen. Dementsprechend können z. B. mehrere Fingerabdrücke erfasst werden. Das System sieht jedoch keine Unterscheidungsmöglichkeit vor, die dazwischen differenzieren könnte, ob es sich um einen weiteren Finger des Geräteeigentümers oder einer weiteren Person handelt. Wird der Finger eines anderen Menschen hinterlegt, führt dies nicht nur dazu, dass dieser das Gerät entsperren kann, sondern auch, dass alle Zugangssperren passiert werden können, die auf der biometrischen Authentifizierung aufbauen. In unserem Fall wäre es also möglich, Transaktionen bei einer 2AA- oder 1AA-Lösung freizugeben, insofern diese ausschließlich auf die Elemente Inhärenz und Besitz zurückgreifen. So eine Situation ist auch aus Sicht der RTS problematisch, da Artikel 24(1) vorsieht, „dass nur der Zahlungsdienstnutzer den personalisierten Sicherheitsmerkmalen zugeordnet" ist. Nach Einrichtung der biometrischen Authentifizierung in der App ist es zumindest möglich, Veränderungen zu erkennen (`evaluatedPolicyDomainState` unter iOS bzw. `setInvalidatedByBiometricEnrollment` ab Android 7). Daraus ergibt sich jedoch nicht, ob Merkmale derselben Person hinzugefügt oder entfernt wurden.

Insgesamt genügt nur das 2GA-Verfahren all unseren Kriterien. Die beiden Mobilebanking-Implementierungen im 2AA- und 1AA-Verfahren zeigen zwar auch eine große Kompatibilität zu unseren Anforderungen, können aber noch nicht in der Breite auf die Möglichkeit einer sicheren Anzeige und Bestätigung der Transaktionsdaten zurückgreifen. Dadurch bleibt es auch in Zukunft möglich, Transaktionen für den Nutzer transparent in Echtzeit zu manipulieren.

Selbst sechs Monate vor der Umsetzungspflicht der RTS herrscht weitgehende Unklarheit darüber, welche Anforderungen die EBA und die nationalen Aufsichtsbehörden an die App-basierten Sicherungsverfahren stellen. Es ist nur gesichert, dass die EBA keine Notwendigkeit darin sieht, die Authentifizierungselemente über verschiedene Geräte zu realisieren [EBA17].

6.4 Ausblick auf potenzielle Angriffe

Der letzte Abschnitt hat gezeigt, dass sich durch die RTS höhere Anforderungen an die Sicherungsverfahren und damit an die Transaktionsbestätigung ergeben. Es kann also durchaus davon gesprochen werden, dass die RTS dazu beitragen, die Sicherheit zu erhöhen und Betrug zu erschweren. Gerade der Wegfall der bis vor kurzem noch sehr verbreiteten iTAN-Liste wird dazu führen, dass ein beliebter Angriffsvektor der Kriminellen in Zukunft nicht mehr existiert. Mit Blick in die Vergangenheit ist jedoch auch davon auszugehen, dass dadurch nicht das Ende der kriminellen Machenschaften bei Bankgeschäften markiert wird. Stattdessen werden sich die Angreifer neue Schwachstellen suchen.

Im Folgenden stellen wir verschiedene Defizite im Transaktionsprozess vor, die auch dann noch zum Tragen kommen, wenn eine SCA zur Anwendung kommt. Der Fokus der Angriffe liegt zum Teil auf Bereichen, die außerhalb des Einflusses der Banken liegen und nimmt auch Bezug auf menschliche Schwachpunkte. Neben dem eigentlichen Angriffsvektor geben wir ebenfalls Hinweise, wie sich die Defizite adressieren oder zumindest eindämmen lassen.

Insofern der entsprechende Abschnitt nichts anderes darlegt, bemühen die Angriffe alle das folgende Szenario: Ein Kunde hat ein Produkt in einem Onlineshop auf Rechnung bestellt und will diese nach Erhalt der Ware per Online-Überweisung begleichen. Zur Transaktionsauslösung nutzt er abhängig vom konkreten Angriff entweder einen PC oder ein Smartphone. Abgesehen von einer Ausnahme sind die Angriffe unabhängig vom eingesetzten Sicherungsverfahren. Es ist das Ziel des Angreifers, den Empfänger der Transaktion zu manipulieren; der Rechnungsbetrag bleibt hingegen unverändert. Dieses Vorgehen zielt darauf ab, dass der Kunde in der Regel sehr genau weiß, wie hoch der Betrag seiner Rechnung ist, die IBAN des Empfängers aber nicht kennt. Infolgedessen würde ihm ein veränderter Rechnungsbetrag eher auffallen als ein abweichender Empfänger. Die manipulierte IBAN zeigt dabei auf ein Konto, das in derselben Nation geführt wird. Solche Transaktionen sind durch die Betrugserkennungssysteme der Banken besonders schwer zu erkennen [Car+18]. Gelingt ein so durchgeführter Angriff, bleibt der Betrug vermutlich längere Zeit unentdeckt und offenbart sich erst mit der Mahnung durch den rechtmäßigen Gläubiger. Auf diese Weise können auch ohne Einflussnahme auf den Betrag beträchtliche Summen unter die Kontrolle des Angreifer gebracht werden.

6.4.1 Manipulation der Zwischenablage

Die Zwischenablage ist sowohl bei den Desktop- als auch den Mobilsystemen ein geteilter Zwischenspeicher, den jedes Programm auf dem System nicht nur lesen, sondern auch schreiben kann. Diese Eigenschaft kann sich Schadsoftware zunutze machen, um sensible Daten auszuspähen [Fah+13] oder auch zu manipulieren [ZD14].

Angriff. Die IBAN ist ISO-standardisiert und folgt einem wohldefinierten Schema. Demnach kann eine IBAN aus bis zu 34 alphanumerischen Zeichen bestehen. Aufgrund dieser Länge liegt die Verwendung der Zwischenablage nahe, um die IBAN in einer digitalen Rechnung zu kopieren und anschließend in das entsprechende Überweisungsformular einzufügen. Neben dem kontoführenden Institut und der Kontonummer kodiert die IBAN auch zwei Prüfziffern. Dieser Umstand spielt einem Angreifer, der mit einer Schadsoftware auf dem Opfergerät die Zwischenablage überwacht, in die Hände: wann immer sich der Inhalt der Zwischenablage ändert, kann der Angreifer zuverlässig erkennen, ob es sich um eine IBAN handelt. Ist dies der Fall, kann er die IBAN sofort durch eine beliebige andere austauschen. Fügt der Kunde die IBAN ein, fällt ihm die Veränderung vermutlich nicht auf; es war schließlich die ursprüngliche Intention des Kunden, die IBAN zu kopieren, damit er sie sich nicht ganz oder zumindest in Teilen merken muss. Außerdem beugt das Kopieren und Einfügen Fehlern bei der manuellen Übertragung vor. Demnach könnte ein Kunde die Nutzung der Zwischenablage auch für besonders zuverlässig und sicher halten. Obwohl uns kein Fall bekannt ist, in dem der beschriebene Angriff bereits stattgefunden hat, wurde im Februar 2019 im Google Play Store eine Schadsoftware entdeckt, die einen analogen Angriff für Kryptowährungen durchführt: Kopiert der Nutzer eine Bitcoin- oder Ethereum-Wallet-Adresse, tauscht die Schadsoftware diese Adresse im Hintergrund aus [Ste19]. Fällt dem Opfer die Manipulation der Zwischenablage nach dem Einfügen nicht auf, schreibt er ungewollt dem Angreifer den gewünschten Betrag gut.

Verteidigung. Um solchen Angriffen vorzubeugen, sollten die App-Anbieter es verbieten, Daten über die Zwischenablage in sensible Formelemente einzufügen. Sowohl im Browser als auch für mobile Android- und iOS-Apps existieren entsprechende Möglichkeiten.

6.4.2 Rechnungsmanipulation

Es ist heutzutage üblich, dass Onlineshops ihre Rechnungen digital zustellen. In vielen Fällen wird dafür eine PDF-Rechnung oder sogar ein Link, der zu einer Webseite mit den Zahlungsdetails führt, an die E-Mail-Adresse des Kunden zugestellt. Eine Papierrechnung entfällt dann oft vollständig.

Angriff. Anstatt die Transaktionsauslösung zu manipulieren, kann eine Schadsoftware auch direkt die Rechnungsdaten manipulieren. Durch das wohldefinierte Format lassen sich auch in PDF- oder HTML-Rechnungen IBANs gut automatisiert erkennen und austauschen. Insbesondere die Manipulation einer Rechnungswebseite ist trivial und lässt sich ohne Weiteres z. B. durch eine bösartige Browser-Erweiterung realisieren [Kap+14]. Da die Rechnung in der Regel an die E-Mail-Adresse des Kunden zugestellt wird, ist es nicht zwangsläufig notwendig, das Endgerät des Opfers mit einer Schadsoftware zu infizieren: Gelingt es einem Angreifer, Kontrolle über das Postfach zu erlangen, kann er eingehende E-Mails abfangen und entsprechend modifizierte zustellen. Ein auf diese Weise durchgeführter Angriff ist besonders effektiv, da das Oper auch bei korrekter Anwendung des Sicherungsverfahrens und Kontrolle der Transaktionsdaten mit der Rechnung kein auffälliges Verhalten feststellen kann.

Verteidigung. Diese Art des Angriffs kann nicht durch die Bank, sondern nur durch den Onlineshop eingedämmt werden. Demnach könnten die Shop-Betreiber die Rechnung ausschließlich per Post zusammen mit der Ware zustellen. Diese Option ergibt sich aber nur beim Rechnungskauf, nicht aber beim Kauf auf Vorkasse: Hier geht der Kunde in Vorleistung und zahlt dabei per Banküberweisung. Dieses Vorgehen sollte der Kunde jedoch vermeiden, da es mit Giropay oder Paydirekt andere Lösungen gibt, mit denen direkt vom Girokonto des Kunden bezahlt werden kann. Der Vorteil dieser Ansätze ist, dass der Onlineshop die Zahlungsdaten direkt an den Zahlungsdienstleister übermittelt. Eine Manipulation dieser Daten auf Kundenseite ist deshalb ausgeschlossen. Es ist möglich, dass solche Zahlungsangebote durch die in der PSD2 neu hinzugekommenen Zahlungsauslösedienste zunehmen werden. Das Signieren von PDF-Rechnungen bringt – abgesehen von den fehlerhaften Verifikationsroutinen in einer Vielzahl von PDF-Anzeige-Programmen [Sel18] – voraussichtlich nicht den gewünschten Effekt, da bezweifelt werden muss, dass einem Kunden die fehlende Signatur in einer manipulierten Rechnung auffallen würde.

6.4.3 Überweisungsvorlagen

Um es einem Kunden zu erleichtern, in unregelmäßigen Abständen auftretende Zahlungen an denselben Empfänger durchzuführen, bieten viele Banken explizite und implizite Überweisungsvorlagen an. Eine explizite Überweisungsvorlage muss ein Kunde eigenständig anlegen. Hierfür sieht das Banking-Portal des Kunden eine entsprechende Aktion vor; zusätzlich wird dem Nutzer aber auch vor Absenden eines Überweisungsauftrags die Möglichkeit geboten, diesen als Vorlage zu speichern. Will der Kunde nun eine Transaktion an eine Überweisungsvorlage schicken, kann er den entsprechenden Kontakt aus einer Liste auswählen. Der Empfängername und die IBAN werden daraufhin automatisch eingefügt. Implizite Vorlagen funktionieren ähnlich, ein Kunde muss diese aber nicht explizit anlegen. Stattdessen bietet das Überweisungsformular schon beim Tippen des Empfängernamens entsprechende Vorschläge, die sich aus der Transaktionshistorie des Kunden speisen. Wählt ein Kunde einen Vorschlag an, wird wieder der Empfängername vervollständigt und die IBAN eingefügt.

Angriff. Überweisungsvorlagen sind eine rein clientseitige Komfortfunktion. Das bedeutet, dass sie lediglich beim Ausfüllen des Überweisungsformulars helfen, sich jedoch die Abläufe im Vergleich zu einer Überweisung ohne Verwendung einer Vorlage vonseiten des Servers nicht ändern. In Konsequenz kann ein Angreifer die IBAN, die durch Auswahl der Überweisungsvorlage in das entsprechende Feld eingefügt wird, beliebig verändern. Es ist unwahrscheinlich, dass der Kunde in solchen Fällen überhaupt einen Originalbeleg verfügbar hat, um die IBAN während der Transaktionsbestätigung zu verifizieren.

Verteidigung. Überweisungsvorlagen sind nur sehr eingeschränkt mit dem Prinzip WYSIWYS in Einklang zu bringen. Deshalb ist es schwierig, eine Lösung anzubieten, die auf der einen Seite den Komfort von Überweisungsvorlagen bietet, auf der anderen Seite den Kunden aber auch dazu anhält, die Transaktionsdetails ordentlich zu vergleichen. Eine Möglichkeit, die beiden Ziele auszubalancieren, könnte das Maskieren eines Teils der Empfänger-IBAN sein, die der Kunde dann manuell eingeben muss. Auf diese Weise kann der Kunde auf das Eintippen eines Großteils der Daten verzichten, fördert aber das Vorhandensein einer weiteren Verifikationsquelle.

6.4.4 SMS-Autovervollständigung

Bereits 2016 haben Konoth, Veen und Bos gezeigt, dass sich die Continuity genannte SMS-Synchronisierung von iOS zu macOS missbrauchen lässt, um beliebige smsTAN-Transaktionen nur durch Kompromittierung des transaktionsauslösenden Geräts tätigen zu können [KVB16]. Mit der Einführung von iOS 12 und macOS 10.14 im September 2019 ist diese Integration noch enger geworden: Wenn ein Nutzer eine Seite besucht, die nach einem SMS-Einmalpasswort fragt, dann bietet Safari das automatische Einfügen des Codes an, insofern das Feld entsprechend deklariert wurde. Diese Funktionalität ist analog für Apps ab iOS 12 möglich.

Angriff. Das automatische Einsetzen eines Einmalpassworts ist aus Sicherheitssicht nur unbedenklich, wenn mit dem Code keine zusätzlichen Daten authentifiziert werden. Eine solche Situation ergibt sich bei der Benutzer-, nicht aber bei der Transaktionsauthentifizierung. Beim smsTAN-Verfahren ist nicht die in der SMS enthaltene TAN das wesentliche Sicherheitsmerkmal, sondern die ebenfalls enthaltenen Transaktionsdaten, die mit der Übermittlung der TAN durch den Kunden bestätigt werden. Wird dem Nutzer nun angeboten, die TAN ohne Umwege direkt einzufügen, dann wird er dazu angehalten, keinerlei Verifikation der Transaktionsdaten vorzunehmen. Ein Angreifer könnte sich diese Funktion zunutze machen, um eine Kontrolle der Auftragsdaten weniger wahrscheinlich zu machen. Zwar zeigt der entsprechende Vorschlag zur Vervollständigung, den Betrag an, nicht aber den Zahlungsempfänger. Wird wie in unserem Szenario nur die IBAN des Begünstigten ausgetauscht, lässt sich die Manipulation ohne Öffnen der SMS nicht erkennen.

Verteidigung. Aus Sicht der Bank ist es schwierig, diesen Angriffsvektor zu verhindern. In unseren Tests mit der iOS 12 Beta genügte es noch, auf das Schlüsselwort „code“ zu verzichten. Zusätzlich wurden nur sechsstellige, numerische Einmalpasswörter zur Vervollständigung erkannt. In der Version 12.2 wurde das Einsetzen der TAN aber auch ohne das Vorhandensein des Worts „code“ angeboten. Dadurch wird deutlich, dass Apple die Erkennungsroutine jederzeit ändern kann. Der Bank bleiben letztendlich nur eine kontinuierliche Überwachung und Anpassung der Formatierung der SMS, um es zu verhindern, dass iOS die Autovervollständigung der TAN auslöst. Im Übrigen stellt die mangelnde Kontrolle der Bank über das Vorgehen von Apple ein weiteres Indiz für die Unzulänglichkeit der smsTAN zu den Anforderungen der RTS dar und zeigt, dass die Sicherheit der Legitimierungsverfahren in zunehmende Abhängigkeit von Dritten gerät.

6.4.5 Transaktionsmanipulation

Ein Angreifer könnte sein Opfer auch dazu bewegen, eine fehlerhafte Transaktionsverifikation durchzuführen: Statt während der Transaktionsbestätigung die im Sicherungsverfahren dargestellten Auftragsdaten mit denen der Originalrechnung zu vergleichen, könnte der Angreifer Anweisungen anzeigen, die sein Opfer dazu auffordern, die im transaktionsauslösenden Kanal angezeigten Details für den Vergleich heranzuziehen. In vielen Fällen muss der Angreifer diese Auftragsdetails nicht selbst in die Bestätigungsseite, in der der Kunde zur Eingabe einer TAN aufgefordert wird, injizieren, da viele Banken dort ohnehin die Auftragsdetails erneut anzeigen. Dieses Angriffsszenario wird in Kapitel 7 in einer Nutzerstudie mit 100 Teilnehmern ausführlich evaluiert und stellt sich als durchaus effektiv heraus.

6.5 Fazit

In diesem Kapitel haben wir uns im Rahmen von Forschungsfrage 4 (Regulierung) mit den regulatorischen Vorgaben der PSD2 und ihren Auswirkungen beschäftigt. Hierfür haben wir zuerst zwei abstrakte, allgemeine Sicherheitsvoraussetzungen für digitale Transaktionen im Online- und Mobilebanking formuliert. Demnach müssen Transaktionen durch den Nutzer willentlich ausgeführt worden und zudem frei von Manipulationen sein. Obwohl viele offene Fragen und ein beträchtlicher Interpretationsspielraum zurückbleiben, fördern die regulatorischen Vorgaben diese beiden Sicherheitsziele in einem solchen Umfang, dass man von einer weitreichenden Kompatibilität sprechen kann.

Aus der Regulierung ergeben sich für einen Teil der Bankkunden auch unmittelbare Auswirkungen, da diese ihr bisheriges Legitimierungsverfahren wechseln müssen. So können die Nutzer der listenbasierten iTAN dieses Verfahren ab September 2019 nicht mehr verwenden. Obwohl die Kreditwirtschaft den Eindruck erwecken will, dass die smsTAN den Vorgaben entspräche, bietet sie insbesondere keine Vertraulichkeit der über den Mobilfunk übertragenen Transaktionsdaten und damit auch keine Konformität. Der Regulator lässt ebenfalls verlässliche und exakte Aussagen vermissen und nährt den Verdacht, dass es am ihm politischen Willen fehlt, ein so verbreitetes Sicherungsverfahren wie die smsTAN vom Markt zu verbannen. Auch an der Konformität App-basierter Verfahren, die auf ein und demselben mobilen Endgerät betrieben werden, nährt unsere Analyse begründete Zweifel. Dennoch scheint es auch hier wahrscheinlicher, dass sich der Regulator eher nach dem Status

Quo im Markt richtet, als die Implikationen der RTS für die App-Verfahren komplett durchzudeklinieren. Wäre dies der Fall, müssten die 2AA- und 1AA-Verfahren nach dem Stand der Technik konsequenterweise für nicht konform erklärt werden.

Obwohl sich durch die regulatorischen Vorgaben in Bezug auf die Sicherheit der Legitimierungsverfahren eine klare Verbesserung einstellt, handelt es sich nur um einen Baustein von vielen, die die Transaktionssicherheit ausmachen. Anhand von weiteren Defiziten im Transaktionsprozess haben wir gezeigt, dass sich auch im Geltungsbereich der RTS genügend Angriffspotenzial für Kriminelle bietet. Eine der skizzierten Angriffsmöglichkeiten beleuchtet das folgende Kapitel 7 mit einer Nutzerstudie ausführlich.

7

Sorgfaltspflicht des Kunden in der Praxis

Das Problem ist nicht der Nutzer.

— Bruce Schneier, 2016 [Sch16]

Um die Sicherheit im Online- und Mobilebanking in der Praxis zu gewährleisten, stehen sowohl die Bank als auch der Kunde in der Verantwortung. Zum einen muss die Bank dem Kunden ein Sicherungsverfahren zur Verfügung stellen, das es dem Kunden erlaubt, Transaktionen technisch sicher durchzuführen. Diesem Aspekt hat sich die Dissertation in den zurückliegenden Kapiteln von verschiedener Seite genähert und es kann zumindest konstatiert werden, dass Verfahren existieren, die eine technisch sichere Abwicklung erlauben. Zum anderen lässt sich Sicherheit bei der Verwendung des WYSIWYS-Schemas nicht ausschließlich technisch umsetzen, da zumindest die Richtigkeit der Transaktionsdaten durch den Nutzer zu prüfen ist. Die Bank kann Vertraulichkeit, Integrität und Authentizität nur in dem Sinne gewährleisten, als dass sich der Nutzer sicher sein kann, dass die im Legitimierungsverfahren angezeigten Daten auch tatsächlich denen entsprechen, die bei ihr eingegangen sind. Ob es sich dabei auch um die vom Nutzer gewünschten Auftragsdaten handelt, liegt hingegen im Verantwortungsbereich des Kunden.

Sorgfaltspflichten. Eben diese Kontrolle der Auftragsdaten erlegen die Banken dem Kunden im Rahmen seiner Sorgfaltspflichten auf. Hierfür findet sich im Allgemeinen ein entsprechender Passus in den Bedingungen zur Nutzung des Onlinebankings. So schreibt z. B.die Sparkasse Nürnberg in ihren „Bedingungen für das Online-Banking“ in der Fassung vom 13. Januar 2018 Folgendes vor:

> *Soweit die Sparkasse dem Teilnehmer Daten aus seinem Online-Banking-Auftrag (z. B. Betrag, Kontonummer des Zahlungsempfängers, Wertpapierkennnummer) im Kundensystem oder über ein anderes Gerät des Teilnehmers (z. B. Mobiltelefon, Chipkartenlesegerät mit Display) zur Bestätigung anzeigt, ist der Teilnehmer verpflichtet, vor der Bestätigung die Übereinstimmung der angezeigten Daten mit den für die Transaktion vorgesehenen Daten zu prüfen. Bei Feststellung von Abweichungen ist die Transaktion abzubrechen.*

In den Bedingungen anderer Banken finden sich sehr ähnliche und zum Teil wortgleiche Formulierungen, weshalb von einer Vorlage der DK auszugehen ist. Dass solche Vorgaben gemacht werden, ist nachvollziehbar, da das allgemein akzeptierte Authentifizierungsparadigma bei Bankgeschäften im Online- und Mobilebanking wie beschrieben auf einer nachgelagerten Zweitverifikation der Transaktionsdetails im Sicherungsverfahren beruht. Stimmen die dort dargestellten Auftragsdaten nicht mit den gewünschten überein, so ist der Auftrag abzubrechen.

Auftragsdaten auf der Bestätigungsseite. Es überrascht jedoch, dass viele Banken den Kunden auf der Bestätigungsseite, die den Kunden unter Anwendung seines Sicherungsverfahrens zur Eingabe einer TAN auffordert, die Transaktionsdetails erneut anzeigen. Dieses Vorgehen ist deshalb irreführend und kontraproduktiv, da dem Kunden auf diese Weise suggeriert wird, die dort dargestellten Informationen seien vertrauenswürdig. Viel mehr noch: die Darstellung der Auftragsdaten auf der Bestätigungsseite könnte die Kunden dazu verleiten, eine fehlerhafte Transaktionsverifikation durchzuführen. Für eine korrekte und sichere Transaktionsbestätigung muss der Kunde die Zahlungsdaten, die ihm sein Sicherungsverfahren anzeigt, mit denen vergleichen, die er zur Eingabe der Transaktion herangezogen hat. Dies könnte z. B. eine Papierrechnung eines Onlineshops sein. Durch die Anzeige der Transaktionsdetails wird ein Kunde dazu verleitet oder sogar dazu erzogen, die Korrektheit der Daten im Sicherungsverfahren auf Basis der am Bildschirm des transaktionsauslösenden Kanals zu überprüfen.

Unsere Stichproben zeigen, dass die meisten Banken die Auftragsdaten auf die Eingabeseite der TAN spiegeln. Die Volksbanken und Raiffeisenbanken gehen durchweg so vor, Commerzbank und Comdirect ebenso. Besonders bemerkenswert ist das Vorgehen der Sparkassen, die das Verhalten vom eingesetzten Sicherungsverfahren abhängig machen. Wie Abbildung 7.1a am Beispiel der Sparkasse Nürnberg zeigt, werden bei der Verwendung des chipTAN-Verfahrens keine Transaktionsdaten

(a) chipTAN

(b) pushTAN

Abbildung 7.1: Sparkasse Nürnberg (17. April 2019): Die Anzeige der Auftragsdaten der gleichen Überweisung hängt vom eingesetzten Sicherungsverfahren ab.

angezeigt. Dieses Vorgehen wäre zur Förderung einer ordentlichen Transaktionsverifikation aus den oben geschilderten Gründen auch zu erwarten, wird jedoch nicht konsequent verfolgt: verwendet der Nutzer das pushTAN-Verfahren (Abbildung 7.1b), so werden ihm im Bestätigungsschritt die vollen Auftragsdaten angezeigt. Dasselbe trifft auch für das smsTAN-Verfahren zu.

Unser Angriff. Das Vorgehen der Banken motiviert unseren Angriff, der darauf abzielt, dass das Opfer seine Transaktionsverifikation auf Grundlage der auf der Bestätigungsseite angezeigten Daten vornimmt, statt den Originalbeleg heranzuziehen. Zu diesem Zweck tauscht unser Angriff die IBAN des Zahlungsempfängers nach Eingabe durch den Nutzer aus; der Betrag bleibt unangetastet. Führt der Nutzer in Übereinstimmung mit seinen Sorgfaltspflichten eine korrekte Transaktionsverifikation mit der Rechnung durch, fällt ihm die Abweichung auf. Zieht er jedoch stattdessen die dargestellten Daten am PC-Bildschirm zurate und verzichtet auf die Konsultation der Rechnung, stimmen alle Daten überein. Insofern der Zahler nicht anderweitig, z. B. indem er sich an die Eingabe der authentischen Original-IBAN erinnert, Verdacht schöpft, wird er zu dem Schluss gelangen, dass die Transaktion integer und deshalb durch die Eingabe der TAN zu bestätigen ist.

Um die Effektivität dieses Angriffs zu überprüfen, haben wir eine Nutzerstudie mit 100 Onlinebanking-Nutzern durchgeführt. Dabei kamen durchweg TAN-basierte 2GA-Verfahren zum Einsatz: die Transaktion wurde über einen PC ausgelöst und dann mit einer durch Anwendung des sms-, chip- oder pushTAN-Verfahrens erhaltenen TAN bestätigt.

Gliederung. Bevor wir die Details unserer Methodologie und des Angriffs darlegen, stellen wir zunächst wichtige vorhergehende Arbeiten vor und ordnen unsere Studie vergleichend zu anderen im Bereich der Transaktionsauthentifizierung ein. Im Anschluss präsentieren wir unsere Resultate, die wir nachfolgend diskutieren, bevor wir dieses Kapitel mit einem Fazit schließen.

7.1 Forschungsstand

Im Folgenden legen wir den Forschungsstand zu Nutzerstudien im Bereich der Benutzer- und Transaktionsauthentifizierung dar. Der Fokus liegt auf akademischen Arbeiten, die sich speziell mit der Authentifizierung bei digitalen Bankgeschäften befassen. Den Abschluss des Abschnitts bildet ein Vergleich zu zwei Arbeiten zur Transaktionssicherheit, die mit unserer Studie eng verwandt sind.

7.1.1 Benutzerauthentifizierung

Krol u. a. führten 2015 eine Längsschnittstudie mit 21 Teilnehmern durch, um die Benutzbarkeit verschiedener Zwei-Faktor-Verfahren für den Login in das Onlinebanking-Portal zu erforschen [Kro+15a]. Die verwendeten Ansätze griffen u. a. auf Lösungen mit dedizierter Hardware sowie Spezial- und Mobilgeräten zurück, deren Funktionsweise in etwa der chip- und smsTAN sowie den App-basierten Verfahren entspricht. Im Rahmen der Studie führten die Teilnehmer elf Tage lang Protokoll über ihre Erfahrungen bei der Authentifizierung. Die Teilnehmer konnten die Verfahren aufgrund ihrer Erfahrung zwar zuverlässig verwenden, waren aber insbesondere mit Hardware-Lösungen und Ansätzen mit Einmalpasswörtern unzufrieden. Darüber hinaus sahen sie sich oft mit einer inkonsequenten Bezeichnung der Authentifizierungselemente konfrontiert. Der zusätzliche Aufwand beim Login führte bei manchen Teilnehmern zu weniger Zugriffen auf die Dienstleistungen der Bank.

2018 evaluierten Das, Dingman und Camp die Benutzbarkeit des Yubikey Hardware-Tokens mit einem Think-Aloud-Protokoll [DDC18]. Sie stellten eine Vielzahl an praktischen Problemen bei der Einrichtung und Verwendung fest. Die Teilnehmer wussten teils nicht, woher sie Informationen zum Yubikey erhalten können und welches Modell sie verwenden. Daneben war vielen der Nutzen des Hardwareschlüssels nicht klar. Im gleichen Jahr beschäftigten sich auch Reynolds u. a. in einer Labor- und einer Längsschnittstudie ausführlich mit der Benutzbarkeit des Yubikey [Rey+18]. Die Laborstudie beobachtete 31 Teilnehmer und identifizierte ähnliche Probleme wie Das, Dingman und Camp bei der Einrichtung und Erstverwendung: Nutzer sperrten sich zum Teil aus oder schlossen das Setup erfolglos ab. Die vierwöchige Langzeitstudie mit 25 Probanden zeigte jedoch, dass Teilnehmer nach erfolgreicher Einrichtung sehr zufrieden mit der Lösung waren.

Schechter u. a. befassten sich 2007 mit der Nützlichkeit von Sicherheitsindikatoren in Webbrowsern [Sch+07]. Zu diesem Zweck fand eine Laborstudie mit 67 Teilnehmern statt, die verschiedene Tätigkeiten im Rahmen ihres persönlichen Onlinebankings durchführen mussten. Für jede Aufgabe mussten sie sich von ihrem Onlinebanking-Portal abmelden und waren mit zunehmend alarmierenderen Sicherheitswarnungen konfrontiert. Eine Vielzahl der Teilnehmer meldete sich trotz aller Indikatoren und Warnungen mit den persönlichen Logindaten an, woraus die Autoren eine durchschlagende Ineffektivität dieser Maßnahmen ableiteten.

7.1.2 Transaktionsauthentifizierung

Weir u. a. zeigten 2009 eine Nutzerstudie mit 50 Teilnehmern, die die Sicherheit, den Komfort und die Benutzbarkeit von drei Hardwarelösungen zur Transaktionsauthentifizierung bewerten sollten [Wei+09]. Die Autoren stellten insgesamt eine starke Korrelation zwischen der Präferenz der Nutzer und den Verfahren fest, die die Teilnehmer als am komfortabelsten und benutzbarsten bewerteten. Das wahrgenommene Sicherheitsniveau spielte eine nur untergeordnete Rolle.

Im Jahr 2008 berichteten Zomai u. a. von einer Studie mit 92 Probanden [Zom+08]. Die Autoren wollten herausfinden, ob die Teilnehmer ihr smsTAN-Sicherungsverfahren korrekt zur Transaktionsverifikation verwenden. Die Studie fand jedoch nicht tatsächlich unter Verwendung von smsTAN statt, sondern wurde über E-Mails simuliert. Die Teilnehmer verwendeten eine eigens entwickelte Onlinebanking-Plattform, um zehn Transaktionen hintereinander durchzuführen und wussten, dass die Transaktionssicherheit im Fokus steht. Die Probanden wurden per E-Mail angeworben, waren zumeist Studenten und führten die Studie autonom an einem Computer ihrer Wahl zu einem selbstdefinierten Zeitpunkt durch. Jeweils acht Transaktionen liefen integer ab, während zwei eine manipulierte Kontonummer beinhalteten. Die erste Manipulation betraf zunächst nur eine Ziffer der achtstelligen Kontonummer; die zweite änderte fünf Stellen. Der erste Angriff wurde von 61%, der zweite von 21% nicht bemerkt. Aufgrund von Abbrüchen wurden jedoch nur 75 Teilnehmer vom ersten und 53 vom zweiten Angriff erfasst.

Hartl und Schmuntzsch beschrieben 2016 ebenfalls eine Laborstudie mit 25 Teilnehmern, in der Onlinebanking-Transaktionen im Hintergrund angegriffen wurden [HS16]. Dabei kamen unter der Verwendung einer eigens erstellten Studienplattform das sms- und das chipTAN-Verfahren zum Einsatz. Im Unterschied zu Zomai u. a. wurden die Verfahren von den Probanden auch tatsächlich angewendet. Die Teilnehmer waren im Mittel seit sieben Jahre mit dem Onlinebanking vertraut, kannten die anzuwendenden Sicherungsverfahren aber unter Umständen nicht. Die Probanden mussten drei Bankgeschäfte durchführen, die in zufälliger Reihenfolge abliefen und alle für sich angegriffen wurden. Einer der Angriffe manipulierte im Hintergrund den Zahlungsempfänger und -betrag; in welchem Umfang ist genauso unklar wie die Verteilung auf das sms- bzw. chipTAN-Verfahren. In Summe wurde der Angriff in 71% der Fälle nicht erkannt, während 50% die IBAN und 39% den Betrag nicht auf Korrektheit überprüften.

	Zomai u. a.	Hartl u. a.	Unsere Studie
Veröffentlichung	2008	2016	2019
Teilnehmer	53	25	100
⌀ Dauer	n. a.	60 min	9,5 min
Methodologie			
Studienform	Entfernt	Labor	Labor
Täuschung	○	n. a.	●
Rekrutierung	Universität	Universität	Unternehmen
Beobachtung	○	●	○
Transaktionen	10	3	2
Sicherungsverfahren			
sms-/chip-/pushTAN	●/○/○	●/●/○	●/●/●
Persönliches Verfahren	○	○	●
Modus Operandi			
Betrag	○	●	○
Änderung Kontonr.	5/8 (62,5%)	n. a.	16/22 (72,7%)
Änderung Anzeige	○	○	●
Angriffe	2	3	1
Opferrate	21%	71%	82%

Tabelle 7.1: Vergleich zu vorherigen Arbeiten zur Transaktionssicherheit.

7.1.3 Vergleich zu unserer Studie

Unsere Studie führt ähnlich zu Zomai u. a. und Hartl und Schmuntzsch eine verdeckte Transaktionsmanipulation durch. Beide Beiträge waren uns vor unserer Arbeit bekannt, weshalb sie in der Studienkonzeption berücksichtigt wurden. Tabelle 7.1 fasst die Gemeinsamkeiten und Unterschiede zusammen, die im Folgenden dargestellt und bewertet werden.

Studienpopulation. Unsere Studie beruft sich auf doppelt so viele Studienteilnehmer wie bei Zomai u. a. und sogar vierfach so viele wie bei Hartl und Schmuntzsch. Eine größere Teilnehmerzahl ist für die ökologische Validität wichtig. Darüber hinaus haben wir unsere Teilnehmer aus einem IT-Unternehmen rekrutiert, wodurch sich ein besonders technikaffiner Gesellschaftsausschnitt ergibt, der aus Gesichtspunkten der Sicherheit einen Idealfall darstellt: fällt ein technikaffiner Proband dem Angriff zum Opfer, trifft dies im Regelfall auch für Nichttechniker zu.

Beeinflussung. Wir haben unsere Teilnehmer in dem Glauben gelassen, dass sie eine Studie zur Benutzerfreundlichkeit von Sicherungsverfahren im Onlinebanking durchführen. Dies steht im Gegensatz zum Ansatz von Zomai u. a., die ihre Probanden bzgl. des Sicherheitsfokuses explizit unterrichtet haben. Beide Vorgehen haben ihre Vor- und Nachteile; für uns war es jedoch wesentlich, die Probanden mit dem Blick auf die Sicherheit nicht zu beeinflussen. Aus diesem Grund haben unsere Teilnehmer die Studie auch einzeln in einem abgetrennten Raum ohne jede Beobachtung durchgeführt. Es ist unklar, welche Informationen Hartl und Schmuntzsch ihren Teilnehmern vermittelten und ob sich dadurch eine Beeinflussung der Teilnehmer ergeben hat. Die Probanden wurden durch die Studienführung jedoch intensiv beobachtet, was bereits die Anwendung der Think-Aloud-Methode zum Ausdruck bringt.

Hauptaufgabe. Unsere Studienteilnehmer hatten eine klare Aufgabe, die aus dem Tätigen von zwei Transaktionen bestand. Die Tätigkeit war den Teilnehmern leicht zu vermitteln und konnte von diesen autonom durchgeführt werden. Im Unterschied dazu mussten die Probanden bei Hartl und Schmuntzsch und Zomai u. a. je drei bzw. zehn Transaktionen durchführen. Unsere Studiensituation genießt dabei mehr Nähe zur Realität, da es plausibel ist, dass ein Onlinebanking-Nutzer zwei Transaktionen in einer Sitzung durchführt. Die umfangreichen Aufgaben bei Hartl und Schmuntzsch benötigten im Schnitt je Proband eine Stunde Zeit; unsere Teilnehmer brauchten im Mittel keine zehn Minuten zur Durchführung der kompletten Studie; Zomai u. a. machen keine Angaben zur Studiendauer.

Sicherungsverfahren. Bei Zomai u. a. kommt mit der smsTAN nur ein einziges Verfahren zum Einsatz, das dazu noch durch E-Mails statt durch SMS simuliert wird. Von Erfahrung mit dem Verfahren kann vor diesem Hintergrund kaum gesprochen werden. Hartl und Schmuntzsch greifen zwar tatsächlich auf das chip- und smsTAN-Verfahren zurück, die Probanden verwenden es aber nicht immer auch selbst. Deshalb fand in der Arbeit von Hartl und Schmuntzsch zu Beginn der Studie auch eine Einführung in die Verfahren statt. Zudem sind die Häufigkeiten, mit denen jeweils das sms- und das chipTAN-Verfahren angegriffen wurden, nicht angegeben. Unsere Studie bietet nicht nur das sms- und das chipTAN-Verfahren, sondern auch App-basierte Sicherungsverfahren. Angegriffen wurde immer die Transaktion, in der der Teilnehmer das Sicherungsverfahren verwendete, das er seiner Angabe nach auch mit seiner Hausbank einsetzt. Dadurch ist nicht nur gesichert, dass der Proband mit dem Verfahren vertraut ist, sondern es wird auch das Ziel gefördert, dass die Studienteilnahme möglichst realitätsnah verläuft.

Angriffsmethodik. Unsere Studie greift nur eine der beiden Transaktionen an, während bei Zomai u. a. zwei und bei Hartl und Schmuntzsch sogar alle drei angegriffen werden. An einem Teilnehmer mehr als einen unabhängigen Angriff zu evaluieren, ist problematisch, da das Verhalten in den nachfolgenden Transaktionen dadurch beeinflusst werden kann. Alle drei Studien manipulieren den Überweisungsempfänger, wobei Zomai u. a. in ihrem realitätsnäheren Angriff fünf von acht Stellen der Kontonummer ändern, während unsere ausgetauschte IBAN an 16 von 22 Stellen zu der Original-IBAN verschieden ist. Hartl und Schmuntzsch machten hierzu keine Angabe. Die Studie von Zomai u. a. zeigt jedoch, dass die Ähnlichkeit der Kontonummern durchaus einen Einfluss darauf hat, ob der Angriff erkannt wird oder nicht. Wir wählen bewusst eine Angreifer-IBAN, die sich drastisch unterscheidet und bei einem Vergleich sofort auffallen würde, da wir mit unserem Angriff erreichen wollen, dass es erst gar nicht zu einem Vergleich kommt, in dem die Abweichung augenscheinlich werden könnte. Zu diesem Zweck stellen wir auf der Bestätigungsseite die Transaktionsdetails erneut dar. Die anderen beiden Arbeiten nehmen eine derartige Modifikation der Anzeige nicht vor. Wir verzichten genauso wie Zomai u. a. auf die Manipulation des Betrags, da sich der Kunde in einer realistischen Situation im Allgemeinen über die Höhe des Zahlungsbetrags sehr genau bewusst ist. Hartl und Schmuntzsch manipulierten neben der Kontonummer auch den Betrag in unbekannter Höhe.

7.2 Methodologie

Im Rahmen unserer Studie mussten die Teilnehmer zwei Rechnungen per Überweisung in zwei Transaktionen – Transaktion I und Transaktion II – durchführen. In Transaktion II kam immer das persönliche Sicherungsverfahren zum Einsatz, das die Probanden auch im Privaten bei ihrer Hausbank zum Tätigen von Überweisungen einsetzten. Diese Transaktion wurde angegriffen, indem im Verborgenen die IBAN des Begünstigten durch eine andere ausgetauscht und zusätzlich die Auftragsdaten auf der Bestätigungsseite, die die Eingabe einer TAN fordert, angezeigt wurden.

Für unsere Studie haben wir eine eigene Onlinebanking-Plattform implementiert, bei welcher der Auftritt der Sparkassen als Vorbild diente. Das System unterstützt die gängigsten Sicherungsverfahren sms-, chip- und pushTAN. Bei letzterem handelt es sich um ein App-basiertes Verfahren, das die TAN als Push-Nachricht auf das Smartphone erhält – ein gängiger Ansatz.

7.2.1 Hypothesen

Auf Basis der vorangegangen Arbeiten und einer Pilotstudie mit 17 Teilnehmern haben wir die folgenden Hypothesen formuliert:

H1 Teilnehmer, die Transaktion I richtig mit der Rechnung verifiziert haben, führen in Transaktion II eine fehlerhafte Verifikation mit den auf dem PC-Bildschirm dargestellten Auftragsdaten durch.

Es herrscht eine Beziehung zwischen dem Nichterkennen des Angriffs und dem Umstand, dass der Proband

H2 ein bestimmtes TAN-Verfahren einsetzt.
H3 seit einer bestimmten Zeit Onlinebanking nutzt.
H4 mit einer bestimmten Anzahl an Sicherungsverfahren vertraut ist.
H5 einen technischen Hintergrund hat.

7.2.2 Studienumgebung

Wir führten unsere Studie in Kooperation mit einem mittelständischen IT-Unternehmen durch, dessen hauptsächliches Geschäft in den Bereich der Softwareentwicklung fällt. Die betreffende Abteilung beschäftigte in etwa 240 Mitarbeiter und ist für die Entwicklung der Kommunikations- und Sicherheitskomponenten der Produkte zuständig.

Ethische Gesichtspunkte. Bevor wir unsere Studie durchführten, haben wir unser Konzept der Abteilungsleitung, den Verantwortlichen für Datensicherheit und Datenschutz sowie dem Betriebsrat vorgelegt. Aus dieser Konsultation sind einige Auflagen entstanden; demnach durfte die Teilnahme während der Arbeitszeit aber völlig freiwillig und ohne das Erfassen persönlicher Daten erfolgen. Der letzte Punkt machte es notwendig, dass wir in bestimmten Fällen – z. B. beim Alter – Bereiche statt konkrete Werte abfragen.

Am Ende unserer Studie haben wir für alle Teilnehmer gemeinsam eine Nachbesprechung abgehalten, die über den Studienverlauf aufklärte. Selbstverständlich haben wir auch ausführlich dargestellt, wie die Sicherungsverfahren korrekt angewendet werden. Die Teilnehmer hatten die Gelegenheit, Fragen zu stellen sowie Bemerkungen, Lob und Kritik zu äußern. Die meisten Teilnehmer zeigten sich jedoch dankbar

und viele empfanden die Studie als wichtigen Hinweisgeber, um in Zukunft Transaktionen sicher durchzuführen. Für den Fall, dass ein Teilnehmer unseren Angriff unmittelbar während der Studie entdeckte, haben wir die Nachbesprechung sofort abgehalten. Nach unserer Kenntnis hat weder die Studienteilnahme selbst noch die Nachbesprechung bei einem der Probanden zu einem nachwirkend negativen Effekt geführt, den es vollständig zu vermeiden galt.

Rekrutierung. Die Teilnahme erfolgte freiwillig durch die Antwort auf eine E-Mail, die die Studie ankündigte. Um das Priming der Probanden in Richtung der IT-Sicherheit zu vermeiden, wurde die Evaluation der Sicherungsverfahren als Zielsetzung angegeben. Um den Aufwand für die Teilnahme gering zu halten, haben wir die Studiendurchführung jeweils in Besprechungsräumen in unmittelbarer Nähe der Probanden abgehalten. Das hatte außerdem den Effekt, dass sich die Teilnehmer in einer vertrauten Umgebung aufhielten.

Teilnehmergeräte. Alle Teilnehmer mussten einen Google Chrome auf einem PC mit Windows 7 – das unternehmensweite Standard-Betriebssystem – bedienen. Für den SMS-Empfang und für das App-basierte Sicherungsverfahren kam ein LG Nexus 5X mit Android 8.1 zum Einsatz. Am PC waren darüber hinaus auch die für das Unternehmen üblichen Peripheriegeräte angeschlossen. Um auch bei den Lesegeräten für das chipTAN-Verfahren möglichst große Vertrautheit zu dem Exemplar zu erzeugen, das die Probanden privat einsetzen, konnten sie aus drei verschiedenen Generatoren wählen, die bei den meisten Banken in der Region zum Einsatz kommen.

7.2.3 Studienablauf

Einweisung. Gleich zu Beginn baten wir den Teilnehmer, uns nach Möglichkeit nicht mit Fragen zu unterbrechen, um allen Teilnehmern den gleichen Grad an Informationen zur Verfügung zu stellen. Als Nächstes erfassten wir das Alter, das Geschlecht und den Beruf des Teilnehmers und stellten den grundlegenden Verlauf der Studie vor, in dessen Kern zwei Transaktionen mit zwei unterschiedlichen Sicherungsverfahren zu tätigen waren. Daneben haben wir auch sichergestellt, dass der Nutzer versteht, wie das Smartphone zu verwenden ist. Bevor wir im Anschluss den Raum verließen, haben wir den Probanden dazu angehalten, sich in die Situation zu versetzen, in der er auch zuhause zwei Rechnungen per Überweisung tätigen würde und dasselbe Verhalten zu adaptieren. Von hier an musste der Proband die Studie bis zum Schluss selbstständig durchführen. Wir haben den Probanden jedoch

darauf hingewiesen, dass er die Studie im Fall eines technischen Problems über einen dedizierten Knopf jederzeit unterbrechen kann. Damit haben wir den Fall antizipiert, dass ein Teilnehmer die Manipulation erkennt, jedoch nicht weiß, wie er weiter verfahren soll.

Im Folgenden stellen wir den weiteren Studienverlauf aus Teilnehmersicht dar, nachdem wir den Raum bereits verlassen haben.

Fragebogen I. Noch vor den beiden Transaktionen musste der Teilnehmer einen in die Studienplattform integrierten Fragebogen ausfüllen. Dieser beschränkte sich auf ein Minimum und hatte hauptsächlich das Ziel, das bei der Hausbank eingesetzte Sicherungsverfahren und die Onlinebanking-Erfahrung abzufragen.

Transaktion I. In diesem Schritt mussten die Teilnehmer eine erste Transaktion mit einem Sicherungsverfahren durchführen, das nicht ihrem bei der Hausbank gewählten entsprach. Der Ablauf war frei von Manipulationen und zielte lediglich darauf ab, den Teilnehmer mit der Studienplattform vertraut zu machen. In Fragebogen I wurde auch die Erfahrung mit weiteren Legitimierungsverfahren abgefragt; wir nutzten diese Information, um in Transaktion I ein Verfahren zuzuweisen, das der Proband bereits kannte. Waren keine Erfahrungen mit anderen Verfahren vorhanden, so hat die Studienplattform eines zugewiesen.

Um die Transaktion durchzuführen, musste der Teilnehmer unter Bezugnahme auf die entsprechende Rechnung ein Formular ausfüllen. Nachdem der Auftrag abgesendet wurde, erfolgte eine Weiterleitung zur Bestätigungsseite. Dort wurde der Teilnehmer aufgefordert, eine TAN mithilfe des Sicherungsverfahrens abzurufen. Diese Seite lieferte auch Instruktionen dazu, wie das Sicherungsverfahren zu verwenden ist. Die Beschreibung war an die der Sparkassen angelehnt.

Transaktion II. Direkt nach Transaktion I musste der Teilnehmer die zweite Transaktion mithilfe des auch persönlich genutzten Sicherungsverfahrens unter Verwendung einer neuen Rechnung tätigen. Abgesehen davon, lief die Transaktionsauslösung vollständig analog ab. Beim Senden der Transaktion fand jedoch unser Angriff statt, der die IBAN des Begünstigten austauschte und auf der nachfolgenden Bestätigungsseite die Auftragsdaten – mit der betrügerischen IBAN – anzeigte. Die Details des Angriffs sind in Abschnitt 7.2.4 dargestellt. Falls ein Proband den Angriff bemerkte, zählte er als Nicht-Opfer. Hat ein Teilnehmer die Transaktion jedoch nicht abgebrochen und hat auch das nachgelagerte Gespräch keinen Zweifel daran genährt, dass er den Angriff trotz alledem erkannt hat, wurde er als Opfer unseres Angriffs vermerkt.

Fragebogen II. Der zweite Fragebogen hatte hauptsächlich zum Ziel, abzufragen, ob und wie die Probanden Transaktion II verifiziert haben. Es wurden jedoch auch weitere Datenpunkte, wie z. B. die Benutzerfreundlichkeit der beiden Sicherungsverfahren, abgefragt.

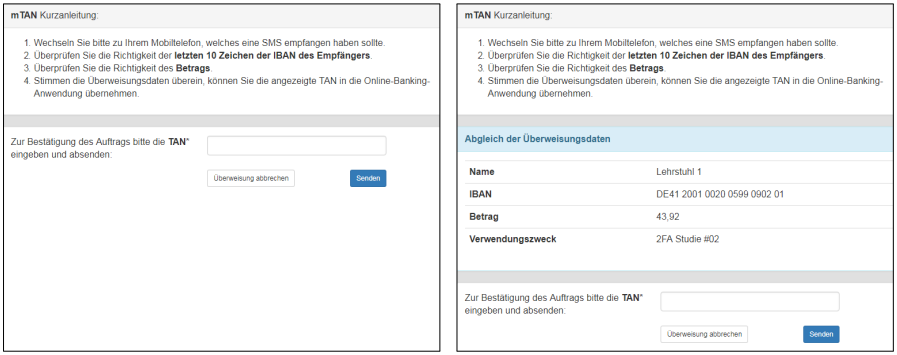
Nachbesprechung. Nachdem der Teilnehmer die Beantwortung von Fragebogen II beendetet hatte, betraten wir den Raum erneut. Daraufhin fragten wir den Probanden, ob er uns noch etwas mitteilen möchte, das der Fragebogen nicht erfassen konnte. Dadurch wollten wir Teilnehmer identifizieren, die zwar eine Abweichung in Transaktion II festgestellt, die Transaktion aber dennoch nicht abgebrochen haben. Ist ein Teilnehmer jedoch Opfer unseres Angriffs geworden und und legte im Folgegespräch nicht nahe, dass er eine Manipulation erkannt haben könnte, wurde er ohne Weiteres entlassen. Eine Aufklärung über den Angriff fand somit erst in der gemeinsamen Nachbesprechung mit allen Probanden statt. Wir haben eine einzelne Unterrichtung unterlassen, um zu vermeiden, dass sich der versteckte Angriff im Gespräch unter Kollegen offenbart. Insofern ein Proband den Angriff erkannt hat, haben wir ihn dazu angehalten, diese Information für sich zu behalten. Wir haben keinen Grund anzunehmen, dass ein Proband durch einen vorhergehenden Teilnehmer beeinflusst wurde.

7.2.4 Angriff

Unser Angriff erfolgte über eine eigens erstellte Erweiterung für den Chrome-Browser, die wir vorab auf dem Studien-PC installiert hatten. Die Forschung zeigt jedoch, dass bösartige Browser-Erweiterungen durchaus auch in der Praxis eine Gefahr darstellen [RL12; Liu+12; Sha+14; Kap+14; Xin+15]. Sobald der Teilnehmer das Überweisungsformular abgeschickt hatte, erfolgte unser Angriff, der zum einen die Empfänger-IBAN austauschte und zum anderen die modifizierten Auftragsdaten auf der Bestätigungsseite anzeigte.

Manipulation des Begünstigten. Die Browser-Erweiterung prüfte zunächst, ob es sich um eine valide IBAN handelte; war dies der Fall, wurde sie durch die Angreifer-IBAN ersetzt und die Transaktion anschließend an den Server weitergeleitet. Die IBAN, die in der Rechnung für Transaktion II enthalten war, unterschied sich mit 16 von 22 Stellen (72,73%) deutlich von der eingeschleusten:

Original-IBAN	`DE62 3702 0500 0000 1020 30`
Angreifer-IBAN	`DE41 2001 0020 0599 0902 01`



(a) Transaktion I: nicht angegriffen (b) Transaktion II: angegriffen

Abbildung 7.2: Das unterschiedliche Aussehen der Bestätigungsseiten während Transaktion I und II.

Der Betrag blieb unangetastet, da er dem Zahler in der Regel sehr genau bekannt ist und sich zudem schnell erfassen lässt. Das Gegenteil trifft auf die IBAN des Begünstigten zu, die gemäß ISO-Standard aus bis zu 34 Stellen besteht und somit nicht nur schwer zu erfassen, sondern auch zu memorieren ist. Sowohl die Original- als auch die Angreifer-IBAN zeigen auf Konten, die bei einer deutschen Bank geführt werden. Auch diese Entscheidung geschah bewusst, da eine IBAN, die nicht mit „DE“ beginnt, unseren deutschen Probanden eher ins Auge gefallen wäre. Außerdem sind nationale IBANs bei Betrügern nicht zuletzt deshalb beliebt [WW19], weil sie von den Betrugserkennungssystemen der Banken mit einem geringeren Risiko bewertet werden als grenzüberschreitende Transaktionen [Car+18].

Auftragsdaten auf der Bestätigungsseite. Im Anschluss wurde der Teilnehmer auf die Bestätigungsseite weitergeleitet, die ihn aufforderte, sein Sicherungsverfahren anzuwenden und nach Verifikation der Auftragsdaten die TAN in das vorgesehene Feld einzutragen. In Transaktion I, die regulär ohne Manipulationen ablief, stellte die Bestätigungsseite (vgl. Abbildung 7.2a) nur Instruktionen zur Anwendung des Verfahrens und ein Eingabefeld für die TAN dar. In Transaktion II zeigte die Bestätigungsseite (vgl. Abbildung 7.2b) hingegen die vollen Auftragsdetails an, wie sie bei unserem Server eingegangen waren. Wie zu Beginn des Kapitels erwähnt, ist dieses Vorgehen kein Anzeichen für einen Angriff, da der Großteil der Banken die Auftragsdetails ebenfalls auf der Bestätigungsseite anzeigt.

7.3 Resultate

Insgesamt sind 82 der 100 Teilnehmer unserem Angriff zum Opfer gefallen, haben also weder während der Studie Transaktion II abgebrochen, noch auf irgendeine Art und Weise eine Ungereimtheit im nachgelagerten Gespräch mit uns angemerkt. Im Nachfolgenden stellen wir die Stichprobe unserer Studie vor und testen die in Abschnitt 7.2.1 formulierten Hypothesen.

Als statistischer Signifikanztest kam der Exakte Fisher-Test (EFT) mit Mid-p-Korrektur zum Einsatz, um die Opfer-Variable auf Unabhängigkeit zu anderen binären Variablen zu testen. Dieser nichtparametrische Test wird als exakte Alternative zum Chi-Quadrat-Test verwendet, da er auch bei kleinen Stichproben akkurate Ergebnisse liefert [LFL09]. Der Mid-p-Ansatz korrigiert das leicht zu konservative Verhalten des traditionellen Exakten Fisher-Tests [HY01]. Für den Fall einer ordinalskalierten abhängigen Variable wurde der ebenfalls nichtparametrische Mann-Whitney-U-Test (MWUT) eingesetzt. Die Hypothesentests wurden stets beidseitig angewendet. Um die Falscherkennungsrate durch das multiple Testen derselben Stichprobe zu kontrollieren, haben wir die p-Werte mit der Benjamini-Krieger-Yekutieli-Prozedur korrigiert [BKY06]. Die korrigierten p-Werte wurden daraufhin mit dem etablierten Signifikanzniveau $\alpha = 0.05$ verglichen und die Nullhypothese wurde nur dann abgelehnt, wenn $p < \alpha$ war.

7.3.1 Studienpopulation

In Summe haben wir 100 Teilnehmer, die im März 2018 an unserer Studie teilnahmen, in unsere Evaluation einbezogen. Dabei setzte knapp die Hälfte ($N = 48$) der Probanden smsTAN- und rund ein Drittel ($N = 31$) das chipTAN-Verfahren ein; 21 gaben an, ein App-basiertes Sicherungsverfahren zu nutzen. Diese Zahlen entsprechen fast exakt der Verteilung bei den Sparkassen zum 23. April 2018 (vgl. Abschnitt 6.3.1).

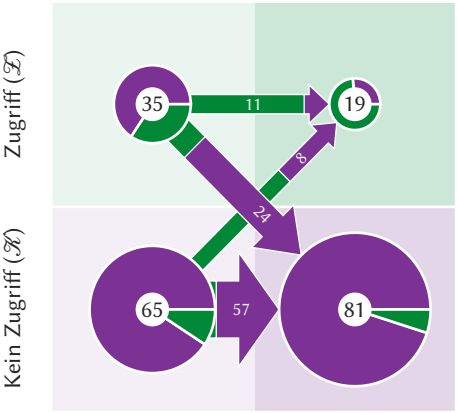
Die Fragebogen I und II sammelten weitere demographische Daten zu den Probanden, die in Tabelle 7.2 dargestellt sind. Aufgrund der Studiendurchführung und -rekrutierung innerhalb eines Unternehmens waren alle Teilnehmer Angestellte des mittelständischen Softwarehauses, wobei 65 männlichen und 35 weiblichen Geschlechts waren. Das Alter umfasste eine breite Spannweite: im Schnitt war unser Teilnehmer 31–40 Jahre alt; der jüngste und der älteste Proband waren 18–25 bzw. 61–70. Die Teilnehmer waren technisch meist sehr versiert, da sie angaben, dass ihre

	SMS		Chip		App		Alle	
	O	N	O	N	O	N	O	N
Geschlecht								
Männl.	90	30	75	20	60	15	78	65
Weibl.	94	18	82	11	83	6	89	35
Alter								
18–25	93	14	83	6	75	4	88	24
26–30	100	2	75	4	67	6	75	12
31–40	85	13	57	7	75	4	75	24
41–50	89	9	83	6	67	3	83	18
51–60	100	8	86	7	50	4	84	19
61–70	100	2	100	1			100	3
Beruf								
Berater	100	2					100	2
Leitung	100	2	100	2			100	4
IT	88	33	72	25	62	16	77	74
Service	100	3	100	1	100	1	100	5
Andere	100	8	100	3	75	4	93	15
Erfahrung in Jahren								
1	100	4					100	4
2	100	4	100	2	100	5	100	11
3	100	2	0	2			50	4
4	67	3	100	1			75	4
5	100	3	100	3			100	6
6	100	4	100	1	0	2	71	7
7								
8	80	5	100	2	100	3	90	10
9			50	2			50	2
10	100	2	100	3			100	5
+	90	21	71	14	58	12	77	47
Transaktionen im Monat								
0	100	1					100	1
1	100	5	100	2	50	2	89	9
2	89	9	50	4	33	3	69	16
3	86	7	88	8	83	6	86	21
4	100	4	60	5	100	1	80	10
5	90	10	75	8	75	4	82	22
6	100	2			0	1	67	3
7	100	3	100	2			100	5
8	100	1			100	1	100	2
9								
10	67	3	100	1			75	4
11–15	100	2			100	1	100	3
+	100	1	100	1	50	2	75	4
Total	92	48	77	31	67	21	82	100

Tabelle 7.2: Opferrate (O, %) und Häufigkeit (N) in Abhängigkeit zur Demographie und zum Sicherungsverfahren der Teilnehmer in Transaktion II.

Tätigkeit im Unternehmen unmittelbar mit der Informationstechnologie (IT) zusammenhing. Aufgrund der Art des Unternehmens und der konkreten Abteilung ist auch bei den übrigen Teilnehmern davon auszugehen, dass sie ein überdurchschnittliches technisches Verständnis hatten.

In Bezug auf die Onlinebanking-Erfahrung lassen sich die Teilnehmer grob in zwei Gruppen einteilen, die entweder bis zu zehn Jahre ($N = 53$) oder bereits mehr als zehn Jahre ($N = 47$) Onlinebanking verwendeten. Nach eigenen Angaben führten die Probanden im Schnitt vier Überweisungen pro Monat durch. Zehn gaben an, dass sie auch an ihrem Arbeitsplatz Überweisungen tätigen. Insbesondere bei diesen Teilnehmern kann nicht nur von einem vertrauten, sondern sogar von einem üblichen Umfeld zum Tätigen von Überweisungen gesprochen werden. Insgesamt können die Teilnehmer als mit dem Onlinebanking sehr erfahren eingestuft werden.



	Trans. I		Trans. II	
	$\mathcal{Z}$	$\mathcal{K}$	$\mathcal{Z}$	$\mathcal{K}$
O	23	59	5	77
¬O	12	6	14	4
$\sum$	35	65	19	81

	$\mathcal{Z}\to\mathcal{Z}$	$\mathcal{Z}\to\mathcal{K}$	$\mathcal{K}\to\mathcal{K}$	$\mathcal{K}\to\mathcal{Z}$
O	2	21	56	3
¬O	9	3	1	5
$\sum$	11	24	57	8

O: Opfer ¬O: Nicht-Opfer

Abbildung 7.3: Rechnungszugriffe: Die Kreise beziffern die Teilnehmer, die in den Transaktionen I und II (nicht) auf die Rechnung zugegriffen haben. Die Pfeile zeigen den Teilnehmerfluss von Transaktion I zu II.

7.3.2 Transaktionsverifikation (H1)

Ein zentraler Schritt der Transaktionssicherheit ist die korrekte und genaue Kontrolle der Auftragsdaten während der Transaktionsbestätigung: stimmt der durch das Sicherungsverfahren angezeigte Zahlungsbetrag oder -empfänger nicht mit dem Rechnungsbeleg überein, dann muss die Transaktion abgebrochen werden. Auf keinen Fall dürfen die Auftragsdaten im Sicherungsverfahren mit Daten verglichen werden, die das transaktionsauslösende Gerät anzeigt, da dieses kompromittiert sein könnte. Generell ist das Legitimierungsverfahren das einzige Gerät, das im Bestätigungsschritt als vertrauenswürdig betrachtet werden kann.

Um zu überprüfen, ob und wie die Probanden ihre Transaktionen verifizieren, wurden uns alle Zugriffe auf die PDF-Rechnung elektronisch signalisiert. Fand während der Transaktionsbestätigung ein Rechnungszugriff statt, so gehen wir von einem zumindest partiellen Rechnungsvergleich aus. Rückschlüsse darauf, welche Daten die Teilnehmer im Detail überprüften, lassen sich daraus aber nicht ziehen: Es ist denkbar, dass ein Teilnehmer auf die Rechnung nur zugreift, um die Korrektheit des Betrags, nicht aber der IBAN zu prüfen; der Angriff würde dann jedoch nicht

auffallen. Umgekehrt führt ein ausbleibender Rechnungszugriff nicht zwangsläufig dazu, dass der Teilnehmer die Manipulation nicht erkennt: er könnte sich z. B. auch an die Original-IBAN erinnern und deshalb den eklatanten Unterschied zur Angreifer-IBAN bemerken.

Wie viele Probanden während der Bestätigung von Transaktion I und II jeweils auf die Rechnung zugegriffen bzw. nicht zugegriffen haben, ist in Abbildung 7.3 aufgeführt. Daraus ergibt sich, dass rund zwei Drittel ($N = 68$) der Teilnehmer bei beiden Transaktionen das gleiche Verhalten adaptierten, wobei der Großteil ($N = 57$) weder zur Bestätigung von Transaktion I noch II auf die Rechnung zugegriffen hat. Entsprechend hoch ist auch die Opferrate von 98,25% ($N = 56$). Die beste Gruppe konsultierte in beiden Transaktionen die Rechnung ($N = 11$), wodurch mit 18,18% ($N = 2$) nur vergleichsweise wenige den Angriff nicht bemerkten. Die restlichen Teilnehmer ($N = 32$) verhielten sich in Transaktion I anders als in Transaktion II: acht Probanden griffen auf die Rechnung nur in Transaktion II zu, fielen in 37,50% ($N = 3$) der Fälle dem Angriff aber trotzdem zum Opfer.

Von besonderen Interesse zur Messung der Effektivität unseres Angriffs waren die verbleibenden 24 Probanden, die zwar in Transaktion I noch auf die Rechnung zugegriffen haben, in Transaktion II aber nicht mehr (H1). Von diesen Teilnehmern sind 87,50% ($N = 21$) Opfer unseres Angriffs geworden, während nur drei die Transaktion trotz des fehlenden Rechnungsvergleichs abbrachen. Von den Opfern gaben in Fragebogen II nur zwei an, dass sie die Verifikation von Transaktion II komplett unterlassen haben, während die übrigen vermerkten, dass sie eine Verifikation durchgeführt haben. Das Gespräch mit den drei Probanden, die den Angriff erkannten, ergab, dass ihnen der Unterschied in der Empfänger-IBAN sofort aufgefallen ist, weshalb sie die Transaktion ohne zusätzliche Rechnungsverifikation abbrachen. Das Ergebnis des Hypothesentests war statistisch signifikant ($p < 0{,}001, OR = 0{,}032$, EFT, 2×2), wodurch sich ein starkes Indiz für die Effektivität unseres Angriffs ergibt.

7.3.3 Eingesetztes Sicherungsverfahren (H2)

Wie in Abbildung 7.4 visualisiert, sind 92% der 48 Teilnehmer, die das smsTAN-Verfahren bei ihrer Hausbank verwenden, 77% der 31 chipTAN-Nutzer und 67% der 21 Anwender des pushTAN-Verfahrens Opfer unseres Angriffs geworden. Die Statistik legt eine Abhängigkeit zwischen dem Abschneiden der Teilnehmer und dem eingesetzten Sicherungsverfahren nahe (H2), die durch ein statistisch signifikantes Resultat bestätigt wurde ($p = 0{,}033$, EFT, 2×3). Dabei schnitten die Verwender des

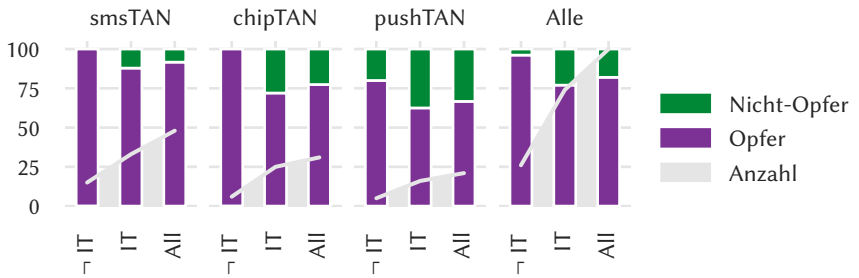


Abbildung 7.4: Abschneiden in Transaktion II in Abhängigkeit vom persönlichen Sicherungsverfahren.

smsTAN-Verfahrens im Vergleich signifikant schlechter ($p = 0{,}024, OR = 4{,}053$, EFT, 2×2) und die Nutzer des pushTAN-Verfahrens signifikant besser ($p = 0{,}033, OR = 0{,}324$, EFT, 2×2) ab, als die übrigen Probanden mit anderen Verfahren. Einzig für das chipTAN-Verfahren ließ sich kein signifikanter Unterschied feststellen ($p = 0{,}221, OR = 0{,}650$, EFT, 2×2).

Am Ende der Studie sollten die Teilnehmer in Fragebogen II auch die Benutzerfreundlichkeit ihres in Transaktion II verwendeten Sicherungsverfahrens angeben (Likert-Skala, $L = [1; 5]$, $1 \coloneqq$ trifft voll zu, $5 \coloneqq$ trifft überhaupt nicht zu). Insgesamt stimmte ein Großteil zu ($N = 32$) oder sehr zu ($N = 65$), dass ihr Verfahren benutzerfreundlich ist. Nur drei Teilnehmer bewerteten ihr Verfahren neutral und niemand lehnte die Aussage ab, dass es einfach anzuwenden sei. Das pushTAN-Verfahren wurde nach der smsTAN am besten bewertet, während das chipTAN-Verfahren am schlechtesten abschnitt. Es ist auch erwähnenswert, dass die Opfer ihr Verfahren im Schnitt besser bewerteten ($L = 1{,}31$) als die Nicht-Opfer ($L = 1{,}67$).

7.3.4 Erfahrungen im Onlinebanking (H3 & H4)

Erfahrung im Onlinebanking kann aufgrund verschiedener Parameter festgestellt werden. Eine naheliegende Annahme ist es, von der Dauer der Onlinebanking-Verwendung auf eine gewisse Erfahrung im Onlinebanking zu schließen. Wir sind davon ausgegangen, dass Probanden mit steigender Nutzungsdauer des Onlinebankings unseren Angriff eher erkennen (H3). Ein Blick in Abbildung 7.5 zeigt bereits, dass sich diese Vermutung nicht erhärtet, da sich das Abschneiden in Bezug auf

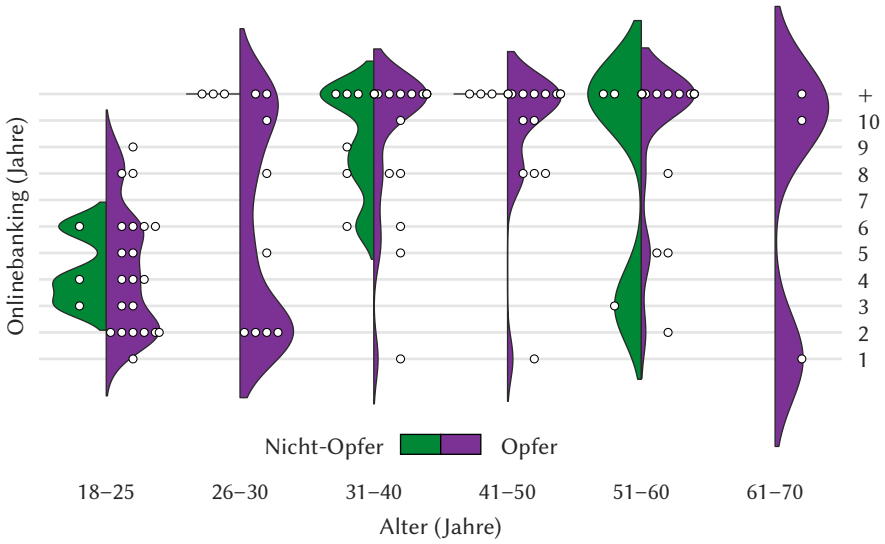


Abbildung 7.5: Abschneiden nach Alter und Onlinebanking-Erfahrung.

die Verwendungsdauer des Onlinebankings recht breit verteilt. Eine bemerkenswerte Ausnahme sind die 15 Probanden mit nur maximal zwei Jahren an Erfahrung im Onlinebanking, die allesamt Opfer des Angriffs wurden. Diese Gruppe ist dazu besonders jung. Obwohl unter den 52 Probanden mit zehn oder mehr Jahren Onlinebanking-Erfahrung nur 78,85% Opfer wurden, ergab sich nahezu kein Unterschied zum Mittelfeld mit drei bis neun Jahren Erfahrung (Opferrate von 78,79%). Insgesamt konnten wir keinen statistisch signifikanten Zusammenhang zwischen dem Abschneiden der Probanden und deren Jahren an Onlinebanking-Erfahrung feststellen ($p = 0{,}118, U = 887{,}500$, MWUT).

Ein anderer Ansatz schließt von der Kenntnis mehrerer Sicherungsverfahren auf Erfahrung im Onlinebanking (H4). Die Vermutung fußt darauf, dass sich ein Onlinebanking-Kunde beim Wechsel der Legitimierungsverfahren erst mit diesem vertraut machen muss und in diesem Zug Sicherheitshinweise zum Onlinebanking allgemein, aber auch im Speziellen zur Sicherheit bei der Transaktionsfreigabe konsultiert. Tatsächlich schnitten die Probanden besser ab, wenn sie bereits Erfahrungen mit mehreren Sicherungsverfahren gesammelt hatten. Insgesamt waren die Teilnehmer mit maximal vier verschiedenen Verfahren vertraut. Die 32 Probanden, die kein

anderes Verfahren als das in Transaktion II angewandte kannten, wurden mit einer Wahrscheinlichkeit von 96,88% Opfer des Angriffs. 68 Teilnehmer hatten zumindest schon ein weiteres Verfahren genutzt und schnitten mit einer Opferrate von 75% bereits besser als der Gesamtdurchschnitt ab. Hatte der Teilnehmer mindestens drei ($N = 19$) oder vier ($N = 5$) Legitimierungsverfahren angewendet, wurden nur noch 57.89% bzw. 60% Opfer der Transaktionsmanipulation. Wir konstatieren, dass es signifikant wahrscheinlicher war, dass ein Proband den Angriff erkennt, wenn er bereits mit mehreren Sicherungsverfahren vertraut war ($p = 0{,}002, U = 1085{,}500$, MWUT).

7.3.5 Technischer Beruf (H5)

Wir haben angenommen, dass Teilnehmer mit einem technischen Beruf besser abschneiden, als Probanden mit einem nicht-technischen Aufgabenbereich (H5). Dem liegt zugrunde, dass ein technisches Verständnis dazu beiträgt, die im Hintergrund stattfindenden Abläufe besser zu verstehen und somit auch Gefahren für die IT-Sicherheit adäquater abzuschätzen. Wie Abbildung 7.4 zeigt, schnitten die 74 Probanden mit einem IT-Beruf deutlich besser ab, als die 26 ohne ausgewiesene IT-Erfahrung: Erstere wurden nur in 77.03% der Fälle Opfer, während 96.15% der letzteren den Angriff nicht bemerkten. Damit hat nur ein Proband ohne technischen Hintergrund den Angriff erkannt. Der Hypothesentest kommt zum Schluss, dass der Unterschied zwischen den Technikern und Nicht-Technikern statistisch signifikant ist ($p = 0.033, OR = 0.134$, EFT, 2×2).

7.4 Diskussion

Insgesamt war unser Angriff bei 82 der 100 Studienteilnehmer erfolgreich. Ursächlich hierfür war die fehlende Verifikation der im Sicherungsverfahren angezeigten Auftragsdaten mit der Originalrechnung. Die Kontrolle der Transaktionsdetails blieb entweder grundsätzlich aus oder erfolgte mit den nicht vertrauenswürdigen Daten auf der Bestätigungsseite. Im Folgenden beleuchten wir, welche Rolle der Nutzer, das Sicherungsverfahren und die Banken dabei spielen. Im Anschluss schlagen wir einige Maßnahmen vor, die dazu beitragen sollen, den Erfolg des durchgeführten sowie weiterer Angriffe zu verhindern oder zumindest einzudämmen. Das Kapitel endet mit der Vorstellung der Limitierungen, die mit dem Konzept und der Durchführung unserer Studie verbunden sind.

7.4.1 Rolle des Nutzers

Arglosigkeit. Der häufigste Grund, warum die Teilnehmer unserer Studie den Angriff nicht bemerkten, war die fehlende Verifikation der Auftragsdaten im Bestätigungsschritt beider Transaktionen. Dieses Verhalten legt nahe, dass es vielen Onlinebanking-Nutzern nicht bewusst ist, dass die Kontrolle der Überweisungsdetails mithilfe des Sicherungsverfahrens und des Originalbelegs ein essentieller Schritt für die Sicherheit der Transaktion ist [DTH06; Sch+07; WZ17; MO07]. Die Nachgespräche mit den Probanden unterstützen diesen Eindruck: drei Teilnehmer gaben an, dass sie die Daten im Sicherungsverfahren nie kontrollieren. Ihnen sei erst durch Fragebogen II, der nach der Verifikation fragte, bewusst geworden, dass dieses Verhalten ein Sicherheitsproblem darstellt. Ein weiterer Teilnehmer stellte hingegen die Frage, ob sich durch eine fehlende Kontrolle eine Bedrohung einstelle.

Onlinebanking-Nutzer halten es zudem für unwahrscheinlich, selbst Opfer von Betrug zu werden und sehen die Bank in der Verantwortung, sollte es zu Schadensfällen kommen [DS14]. Es sei die Aufgabe der Bank, adäquate Maßnahmen zu ergreifen, um Angriffe zu verhindern [RCJ14]. Die Studie von Hartl und Schmuntzsch berichtete von einem Teilnehmer, der seine Aufgabe darin sah, die Transaktionsdaten korrekt einzugeben; alles Weitere sei technisch durch die Bank zu lösen [HS16]. Der Kunde ist im Rahmen seiner Sorgfaltspflichten jedoch zur Verifikation der Auftragsdaten verpflichtet. Findet mutwillig keine Überprüfung statt, wird die Bank ihm grobe Fahrlässigkeit attestieren und für den vollen Schaden haftbar machen [Mur+16].

Intuition. Wenn eine Verifikation durch das Opfer stattfand, erfolgte diese oft mit den unauthentischen Daten auf der Bestätigungsseite. Aus Kundensicht ist dieses Vorgehen aus mehreren Gründen intuitiv und plausibel. Erstens wird dem Nutzer diese Seite direkt nach der Transaktionsauslösung angezeigt, weshalb er die dargestellten Informationen bereits unwillkürlich erfasst. Zusätzlich muss der Nutzer zur Bestätigung zumindest insofern mit der Seite interagieren, dass er die TAN aus dem Sicherungsverfahren eingeben muss. Im Falle des chipTAN-Verfahrens muss dort zusätzlich der Flickercode eingelesen werden. Hinzu kommt nicht zuletzt, dass der Großteil der Banken auf der Bestätigungsseite tatsächlich die Auftragsdaten anzeigt. Dieses Vorgehen der Geldhäuser ist deshalb schädlich, da es dem Kunden suggeriert, die dort angezeigten Daten wären grundsätzlich vertrauenswürdig. Der Zweck des Sicherungsverfahren ist es aber gerade, dass diesen Daten nicht vertraut werden muss und infolgedessen auch nicht vertraut werden darf.

Erfahrung. Bei Teilnehmern mit einem technischen Hintergrund war es im Vergleich zu Nicht-Technikern signifikant wahrscheinlicher, dass sie den Angriff entdecken. Dieses Ergebnis ist wenig überraschend, da ein technisch versierter Nutzer die eigentlichen Abläufe im Hintergrund besser verstehen und nachvollziehen kann. In Konsequenz ist er auch eher in der Lage, Risiken abzuschätzen und informierte Sicherheitsentscheidungen zu treffen [Ona+12]. Die Ergebnisse zeigen auch, dass junge und weibliche Personen besonders angreifbar sind. Dieser Befund deckt sich mit Verhaltensstudien, die die Anfälligkeit für Phishing-Angriffe untersuchten [Jag+07; She+10; BPC11]. Die größere Verwundbarkeit von Frauen führen wir wie Sheng u. a. nicht darauf zurück, dass das weibliche Geschlecht per se angreifbarer ist, sondern auf die geringere Affinität zu technischen Berufen, die sich auch in unserer Studienpopulation niederschlägt [She+10].

Wir fanden keine statistisch signifikante Beziehung zwischen der Nutzungsdauer des Onlinebankings und dem Erkennen des Angriffs. Zwar hatten sehr junge Teilnehmer unserer Studie wie zu erwarten auch weniger lang das Onlinebanking genutzt und wurden besonders oft Opfer unseres Angriffs. Unsere Ergebnisse deuten aber nicht darauf hin, dass die Wahrscheinlichkeit, unserem Angriff zu erliegen, mit einer steigenden Nutzungsdauer des Onlinebankings abnimmt. Wie angedeutet scheint für die besonders junge Gruppe das Alter ausschlaggebend.

Wir konnten jedoch eine Beziehung zwischen der Anzahl der Sicherungsverfahren, die ein Proband bereits eingesetzt hat, und der Opferrate ermitteln. Demnach schnitten die Teilnehmer umso besser ab, je mehr Verfahren sie kannten. Wir erklären uns diesen Zusammenhang damit, dass sich der Nutzer bei einem neuen Sicherungsverfahren erst mit dessen Anwendung vertraut machen muss. In diesem Zug werden auch für die Sicherheit wichtige Informationen konsultiert, die die Wichtigkeit der Kontrolle der Auftragsdaten unterstreichen. Daraus resultiert eine zunehmende Sensibilisierung des Nutzers. Unserer Einschätzung könnte mit dem Argument widersprochen werden, dass das bessere Abschneiden der mit mehreren Verfahren vertrauten Probanden lediglich ein Artefakt des Studienkonzepts sei: Ein Nutzer, der nur ein Sicherungsverfahren kannte, musste in Transaktion I ein Verfahren nutzen, mit dem er nicht vertraut war. Dadurch könnte sich eine stressreiche Situation eingestellt haben, die dann auf Transaktion II überschlug. Für diese Argumentation fand sich keine Unterstützung in unseren Daten, da wir keine signifikant verschiedene Opferrate feststellen konnten. Zudem schlossen die Opfer Transaktion I im Schnitt schneller ab als die Teilnehmer, die den Angriff erkannt hatten. Es ist deshalb eher davon auszugehen, dass die Opfer elementare Sicherheitsschritte schlicht nicht oder fehlerhaft ausführten.

7.4.2 Rolle des Sicherungsverfahrens

Unsere Ergebnisse deuten darauf hin, dass das Sicherungsverfahren einen Einfluss darauf hat, ob ein Nutzer unseren Angriff erkennt. Dabei schnitt das smsTAN-Verfahren signifikant schlechter und das pushTAN-Verfahren signifikant besser ab; für das chipTAN-Verfahren ergab sich kein signifikanter Unterschied. Diese Beobachtung bleibt auch gültig, wenn man nur die technikaffinen Teilnehmer betrachtet, ist also nicht darauf zurückzuführen, dass das IT-Personal besonders häufig das pushTAN-Verfahren verwendete und die smsTAN mied.

Um das Sicherungsverfahren anzuwenden, müssen die Nutzer unterschiedliche Aktionen durchführen. Zusätzlich unterscheidet sich die Art und Weise, wie die Empfängerkontonummer und der Betrag im Sicherungsverfahren dargestellt werden, zum Teil deutlich. Die Verfahren lassen sich nach Struktur und Vollständigkeit der Anzeige sowie der Dauer und den benötigten Interaktionen kategorisieren.

Strukturiertheit. Das Sicherungsverfahren sollte die IBAN und den Zahlungsbetrag strukturiert anzeigen, so dass sie sich einfach erfassen lassen. App-basierte Sicherungsverfahren stellen die Auftragsdaten im Allgemeinen tabellarisch formatiert dar, wodurch sich eine übersichtliche Darstellung ergibt. Das chipTAN-Verfahren zeigt die Daten zwar ebenfalls strukturiert, abhängig vom Gerät aber erst nacheinander an. Keinerlei Struktur hat hingegen die smsTAN, die alle Auftragsdaten in einer einzigen Zeile ausgibt. Die fehlende Formatierung erschwert eine Überprüfung entsprechend.

Vollständigkeit. Es ist wichtig, dass das Verfahren alle notwendigen Informationen auf einmal darstellt, damit der Nutzer alle Daten sofort einsehen kann. Dies ist beim chipTAN-Verfahren regelmäßig nicht der Fall, da es nur über einen kleinen Punktmatrixdisplay verfügt, der nicht in der Lage ist, alle Daten auf einmal auszugeben. Daraus ergibt sich, dass sich der Nutzer die Transaktionsdaten durch Betätigen eines Knopfes nacheinander anzeigen lassen muss. Das sms- und das pushTAN-Verfahren zeigen hingegen alle Auftragsdaten vollständig an.

Dauer. Das Ziel des Nutzers ist der Geldtransfer auf ein anderes Konto und nicht die Anwendung des Sicherungsverfahrens. Deshalb sollte möglichst wenig Zeit verstreichen, bis der Nutzer die Auftragsdaten zur Verifikation und die TAN zur Bestätigung vorliegen hat. Beides ist für das sms- und das pushTAN-Verfahren gegeben. Der beim chipTAN-Verfahren noch übliche Flickercode benötigt substanziell mehr Zeit.

Interaktionen. Auch die Interaktionen des Nutzers mit dem Sicherungsverfahren beschränken sich idealerweise auf ein Minimum. Für das Lesen des Inhalts einer SMS ist zum Teil keine weitere Aktion notwendig; oft kann sie auf Smartphones bereits gelesen werden, ohne das Gerät zu entsperren. Beim pushTAN-Verfahren muss zumindest die entsprechende Push-Nachricht abgerufen werden. Das Vorgehen hängt jedoch stark vom jeweiligen Kreditinstitut ab und erfordert zum Teil eine separate PIN zum Öffnen der App. Zur Anwendung des chipTAN-Verfahrens muss der Nutzer zahlreiche Schritte durchführen: Einführen der Bankkarte in das Lesegerät, Scannen des Flickercodes am Bildschirm, Bestätigung der IBAN und des Betrags durch Drücken eines entsprechenden Knopfs.

Bewertung. Im Durchschnitt benötigte die Anwendung des sms- und des pushTAN-Verfahrens genauso viel Zeit; Opfer benötigten insgesamt weniger Zeit. Wir gehen deshalb davon aus, dass die strukturierte Darstellung der Transaktionsdetails besser geeignet ist, um die Aufmerksamkeit des Nutzers zu gewinnen. Diese Einschätzung wird von einer anderen Forschungsarbeit geteilt, die der Schrift und der Farbe einen wichtigen Effekt beimisst [ES13].

Vor diesem Hintergrund nehmen wir an, dass die ansprechende Organisation beim pushTAN-Verfahren dazu beiträgt, die Auftragsdaten sofort auch ohne bewusste Anstrengung zu erfassen. Das Gegenteil trifft auf die smsTAN zu, die alle Informationen in einer Zeile darstellt, die mit der TAN endet. Wir vermuten deshalb, dass die Darstellung der Auftragsdaten einem mühelosen Erfassen der IBAN und des Betrags entgegenwirkt, weshalb der Nutzer statt einer Verifikation sofort die TAN überträgt. Obwohl das chipTAN-Verfahren in Bezug auf seine Benutzerfreundlichkeit von unseren Teilnehmern am schlechtesten bewertet wurde, haben seine Verwender eine unterdurchschnittliche Opferrate, die aber nicht an die des pushTAN-Verfahrens heranreicht. Obwohl das chipTAN-Verfahren eine strukturierte Anzeige bietet, konnte es unseren Probanden nicht alle Informationen auf einmal vollständig anzeigen. Da der Nutzer zum Bestätigen seiner Transaktion zuallererst die TAN benötigt, überspringt er die Anzeige der IBAN und des Betrags unter Umständen [Her09]. Für den Fall, dass dem Nutzer erst bei der Darstellung der TAN im Display des Lesegeräts bewusst wird, dass er keine Verifikation vorgenommen hat, hat er keine Möglichkeit mehr, zurück zu gehen. Er müsste das Sicherungsverfahren erneut komplett anwenden, um eine ordentliche Verifikation durchzuführen. Eben dieser Schritt ist bei der Verwendung des chipTAN-Verfahrens aber mit besonders viel Aufwand verbunden.

7.4.3 Rolle der Banken

Wir haben der DK die Problematik ihres Vorgehens dargelegt und sie um eine Stellungnahme gebeten. Die drei großen Dachverbände kamen unserer Anfrage nach: der DSGV antwortete für die Sparkassen, der BVR für die Genossenschaftsbanken und der BDB für die Privatbanken.

DSGV. Der Dachverband der Sparkassen äußerte sich am 23. April 2018 wie folgt:

> *Es gibt eine grundsätzliche Sicherheitsempfehlung an die Kunden, die besagt, dass der Nutzer wenn möglich gegen seinen Originalbeleg und nicht gegen die Anzeige im Online-Banking prüfen soll. Da viele der heutigen Anwendungsszenarien wie Foto-Überweisung, PDF-Scan usw. einen einfachen Vergleich nicht zulassen, werden bei den meisten Verfahren die Auftragsdaten zusätzlich auf der Legitimierungsseite angezeigt.*

Die Begründung für die Anzeige der Transaktionsdaten legt nahe, dass der Kunde in manchen Fällen keine Möglichkeit hat, die Auftragsdaten im Sicherungsverfahren mit dem Rechnungsbeleg zu vergleichen. Die Argumentation ist aber in keiner Weise nachvollziehbar: Zwar gibt es Verfahren wie die Foto-Überweisung, die die IBAN und den Betrag einer Rechnung automatisch einlesen, diese Verfahren greifen jedoch letztendlich auch auf den Originalbeleg des Rechnungsstellers zurück. Inwiefern ein PDF-Scan eine Verifikation der Auftragsdaten im Sicherungsverfahren verhindert, ist unklar und scheint ausgeschlossen. Auch die Antwort vom 29. Mai 2018 auf unsere Nachfrage, die um eine Präzisierung der getätigten Aussagen bat, brachte keine Klarheit. Der DSGV wies dort aber zumindest darauf hin, dass „der Kunde die im Rahmen des Sicherheitsverfahrens visualisierten Transaktionsdaten gegen den Originalbeleg (z.B. Rechnung) prüfen soll - also nicht gegen auf dem PC- oder Smartphone-Bildschirm angezeigte Daten".

Obwohl auch der DSGV letztendlich anerkannte, dass das Vorgehen eine fehlerhafte Transaktionsverifikation fördert, legte er sich auf keine weiteren Schritte fest. Demnach hing bei den Sparkassen die Anzeige von Auftragsdetails auf der Bestätigungsseite auch im April 2019 weiter vom eingesetzten Sicherungsverfahren ab: für den Fall des sms- und pushTAN-Verfahrens werden die Daten angezeigt, bei der Verwendung des chipTAN-Verfahrens hingegen nicht. Auch für dieses inkonsistente Verhalten legte der DSGV keine schlüssige Begründung vor.

BVR. In seiner Stellungnahme vom 6. April 2018 erkennt der BVR die Problematik zwar an, weist jedoch indirekt auf die Sorgfaltspflicht des Kunden hin:

> *Auf der Webseite, auf der eine Transaktion mit einer TAN abschließend freigegeben werden muss, werden sämtliche Transaktionsdaten angezeigt. Der Kunde ist jedoch angewiesen, die Kontrolle der gewünschten Transaktion anhand der im SmartTAN-Leser bzw. mobileTAN-Gerät angezeigten Daten vorzunehmen.*

In einer Folgeantwort vom 22. Mai 2018 stellte der BVR weiter klar, dass auf der Bestätigungsseite „viele Details der vorbereiteten Transaktion, die für den Kunden interessant sein könnten", angezeigt werden. Diese Informationen würden den Vergleich mit dem Original-Beleg, den die Bestätigungsseite neben der Anzeige der Auftragsdaten ebenfalls explizit einfordere, jedoch nicht ersetzen.

Im persönlichen Gespräch bezeichnete der BVR die Anzeige der Auftragsdaten auf der Bestätigungsseite letztendlich als Abstimmungsproblem zwischen den Fachabteilungen. Demnach hätten die für Sicherheit der Legitimierungsverfahren Zuständigen nicht eng genug mit den Verantwortlichen zusammengearbeitet, die die Bestätigungsseite schließlich umsetzten. Es wurde uns jedoch kommuniziert, dass die VR-Banken in Zukunft keine Transaktionsdetails mehr auf der Legitimierungsseite darstellen wollen. Einen Termin nannte der BVR jedoch nicht; im April 2019 war noch keine Veränderung festzustellen.

BDB. Der für die Privatbanken zuständige BDB konnte keine gemeinsamen Aussagen für die in ihm organisierten Banken machen, da ihm die Information, ob Transaktionsdetails auf der Bestätigungsseite angezeigt werden, nicht vorlag. Der Verband geht „aber davon aus, dass alle Mitgliedsinstitute Transaktionsdetails anzeigen". Einen Handlungsbedarf leitete der BDB nicht ab.

7.4.4 Maßnahmen

Durch die sehr hohe Erfolgsquote unseres Angriffs entsteht auf mehreren Seiten Handlungsbedarf. Auf Basis unserer bisherigen Analyse stellen wir im Folgenden Maßnahmen vor, mit welchen die Robustheit der Verfahren gegenüber Angriffen erhöht werden soll. Hierbei geht es nicht um Ansätze, die eine rein technisch sichere Abwicklung von Transaktionen ermöglicht; diesem Aspekt wird bereits durch die PSD2 Rechnung getragen. Es geht stattdessen um Methoden, die eine sichere Verwendung durch den Nutzer fördern, damit die Auftragsverifikation keine

rein vertragsrechtliche Klausel bleibt, die vor allem vor den Gerichten eine Rolle spielt.

TAN-Freiheit. Ein Legitimierungsverfahren sollte so gestaltet sein, dass es keinerlei Einmalpasswort mehr enthält. Stattdessen sollte es einen auf „Bestätigen" und „Abbrechen" reduzierten Dialog implementieren. Die TAN wirkt auf den Nutzer wie das zentrale Sicherheitsmerkmal, ist aber nur das Beiwerk zu den eigentlich wichtigen Auftragsdaten. Ursächlich hierfür ist auch, dass die früheren Sicherungsverfahren gar keine Auftragsdetails anzeigten; ihre einzige Funktion war der Erhalt einer TAN. Außerdem werden Lösungen ohne Einmalpasswort vom Nutzer besser angenommen [Kro+15a].

Durchdachte Anwendung und Darstellung. Das Sicherungsverfahren kann vom Nutzer idealerweise auf intuitive Art und Weise bedient werden. Dementsprechend benötigt seine Anwendung so wenige Schritte wie möglich, fördert aber zugleich die Transaktionsverifikation durch eine übersichtliche Darstellung, die alle notwendigen Auftragsdaten auf einmal anzeigt.

Einheitliche Darstellung der IBAN. Sowohl die Banken als auch der Rechnungssteller sollten die IBAN gemäß der DIN 5008 von links aus in Viererblöcken trennen. Zum Teil erfolgt keinerlei Sperrung oder es werden erst nach der Länderkennung Viererblöcke gebildet. Bei einer unformatierten und zusätzlich inkonsistenten Darstellung zwischen Sicherungsverfahren der Bank und Beleg des Rechnungsstellers wird ein Vergleich deutlich erschwert.

Das Trennen der IBAN erleichtert zwar das Erfassen und Vergleichen, kann aber nicht den Umstand adressieren, dass die IBAN bis zu 34 Zeichen lang ist. Es ist deshalb erstrebenswert, dem Nutzer einen partiellen Vergleich zu empfehlen, der eine schnelle und zugleich ausgewogen sichere Verifikation ermöglicht: Die IBAN enthält nach der Länderkennung zwei Prüfziffern, über die ein Kunde die Integrität schnell und zuverlässig prüfen kann. Oft kennt der Kunde diese Prüfziffern jedoch nicht, weshalb er aktiv aufgeklärt werden sollte. Hierfür ist es aber auch notwendig, dass die Sicherungsverfahren die vollständige IBAN anzeigen; viele Institute zeigen im Legitimierungsverfahren nur die letzten zehn Stellen an, wodurch eine Kontrolle der Prüfziffern unmöglich wird.

Verzicht auf Überweisungen. Aus Nutzersicht ist es bei Bestellungen in Onlineshops oft möglich, mehrere Bezahlverfahren zu wählen. Der Kauf auf Rechnung ist aus Kundensicht attraktiv, weil er die Ware erst nach Erhalt zahlen muss. Eine Möglichkeit, Angriffe auf Überweisungen zu verhindern, ist offensichtlich der Verzicht

auf eine Bezahlung per Überweisung. Ein Kunde könnte eigenständig entscheiden, über ein alternatives Bezahlverfahren in Vorleistung zu gehen. Das hat den Vorteil, dass die Zahlungsdaten vom Onlineshop direkt an den Zahlungsdienstleister weitergeleitet werden. Eine Manipulation der Auftragsdaten auf Kundenseite ist deshalb ausgeschlossen. Es wäre jedoch auch ein Rechnungskauf denkbar, bei dem der Kunde mit einem anderen Zahlungsmittel als einer Banküberweisung zahlt. Hierfür ist jedoch auch die Unterstützung durch den Händler gefragt.

What-You-Enter-Is-What-You-Sign. Das Prinzip von WYSIWYS muss mittelbis langfristig zur Disposition gestellt werden. Grundsätzlich ist an die Online- und Mobilebanking-Verfahren der Anspruch zu stellen, dass sie nicht erst durch das vom Nutzer dediziert zur Anwendung gebrachte Sicherungsverfahren die Integrität der Zahlungsdaten ergibt. Stattdessen sollten für die Auftragsdaten vom Zeitpunkt der Eingabe an Vertraulichkeit, Integrität und Authentizität gelten [KVE14]. Für die massentaugliche Umsetzung solcher Ansätze fehlt es aktuell noch an den entsprechenden technischen Umsetzungen. Abschnitt 3.4.2 hat aber gezeigt, dass Android mittlerweile eine durch Hardwaremaßnahmen abgesicherte Anzeige besitzt. Eine sichere Eingabemöglichkeit, die den üblichen Nutzungsparadigmen eines Smartphone entspricht, bleibt hingegen eine Zukunftsaufgabe.

7.4.5 Einschränkungen

Obwohl wir große Sorgfalt bei der Konzeption und Durchführung walten ließen, weisen manche Studienaspekte Einschränkungen auf. Wir wollen mit diesen Limitierungen offen umgehen, um eine adäquate Einschätzung unserer Ergebnisse zu erlauben und Anreize für nachfolgende Studien zu geben.

Ökologische Validität. Unsere Teilnehmer haben bei der Teilnahme an unserer Studie unter Umständen ein weniger umsichtiges Verhalten adaptiert, als sie es üblicherweise bei der Verwendung des persönlichen Onlinebanking-Kontos an den Tag legen. Die in Abschnitt 7.1.1 erwähnte Studie von Schechter u. a. [Sch+07] wertete auch die Sicherheit des Handelns von drei verschiedenen Gruppen aus: Während sich Gruppe 1 und 2 nur in eine Rolle versetzten, nutzte Gruppe 3 den persönlichen Onlinebanking-Zugang. Gruppe 1 war im Unterschied zu Gruppe 2 angehalten, der Sicherheit besondere Beachtung beizumessen. Im Vergleich von Gruppe 3 mit der Vereinigung von Gruppe 1 und 2 stellten die Autoren fest, dass sich die Teilnehmer von Gruppe 3 signifikant sicherer verhielten. Der unabhängige

Vergleich von Gruppe 1, die in etwa unserer Studie entspricht, mit Gruppe 3 zeigte jedoch keinen signifikanten Unterschied in Bezug auf ein sicheres Verhalten.

Wir erkennen an, dass es für die ökologische Validität unserer Ergebnisse förderlich gewesen wäre, wenn unsere Teilnehmer die Studie mit ihrem persönlichen Onlinebanking durchgeführt hätten. Daraus ergibt sich jedoch ein schwer zu kontrollierendes Bedrohungsszenario, da wir im Unterschied zu Schechter u. a. nicht nur den Loginprozess, sondern auch eine tatsächliche Transaktion angegriffen hätten. Die gemeinsame Risiko-Nutzen-Analyse mit dem kooperierenden Unternehmen hat bzgl. eines solchen Vorgehens große ethische Vorbehalte erzeugt, weshalb ein solches Studiendesign letztendlich verworfen wurde.

Aus diesem Grund stand nicht die Fragestellung im Vordergrund, ob Onlinebanking-Nutzer grundsätzlich ihre Transaktionen verifizieren, sondern ob sie die in Transaktion II dargestellten Auftragsdaten auf der Bestätigungsseite für einen fehlerhaften Vergleich mit dem Sicherungsverfahren heranziehen würden, statt die Transaktion korrekt mithilfe der Rechnung zu verifizieren. Für diesen Fall war es insbesondere von Interesse, wie sich die Nutzer in der Transaktion II verhielten, die in Transaktion I noch eine Verifikation mit der Rechnung durchgeführt hatten. Bei dieser Gruppe ist anzunehmen, dass sie der Studiendurchführung eine angemessene Ernsthaftigkeit beigemessen haben, die auch bei der Nutzung ihres eigenen Onlinebankings zu erwarten wäre.

Studienpopulation. Alle unsere Teilnehmer waren Angestellte eines IT-Unternehmens, weshalb unsere Studienpopulation nicht repräsentativ für die Gesamtbevölkerung ist. Durch Fokus auf Softwareentwicklung übten die meisten Probanden eine Tätigkeit aus, die unmittelbar mit der IT verbunden war. Darüber hinaus war der Großteil der Teilnehmer männlich. Obwohl eine ausgewogenere Studienpopulation der Generalisierbarkeit unserer Ergebnisse zuträglich gewesen wäre, betrachten wir die Technikerfahrung unserer Stichprobe als ein Bestfallszenario: Wenn sogar technikaffine Nutzer nicht in der Lage sind, Transaktionen sicher zu verifizieren, dann gelingt dies dem allgemeinen Onlinebanking-Verwender mit einem weniger tiefgreifenden Technikverständnis vermutlich auch nicht.

Rechnungsformat. Der Beitrag fokussiert sich auf die Effektivität unseres Angriffs. Es soll jedoch nicht unerwähnt bleiben, dass eine Hälfte der Teilnehmer über ein dediziertes Tablet auf die Rechnung zugriff, während die andere Hälfte hierfür ein PDF auf demselben PC öffnete, mit dem sie auch die Studie durchführte. Die Opferraten beider Gruppen lagen mit 80% für das Tablet und 84% für das PDF jedoch sehr nah beieinander, weshalb nicht von einem messbaren Effekt auszugehen ist.

7.5 Fazit

Das zurückliegende Kapitel hat sich mit Forschungsfrage 5 (Sorgfaltspflichten) beschäftigt und ging der Frage nach, ob Onlinebanking-Nutzer eine manipulierte Transaktion entdecken, indem sie die Auftragsdaten im Sicherungsverfahren korrekt mit dem Originalbeleg vergleichen. Zu diesem Zweck haben wir eine Nutzerstudie mit 100 Mitarbeitern eines IT-Unternehmens durchgeführt, die zwei Transaktionen durchführen mussten. Während die erste Überweisung regulär ablief, wurde bei der zweiten die IBAN des Begünstigten im Hintergrund ausgetauscht. Zusätzlich wurden bei der Transaktion die betrügerischen Auftragsdaten auf der TAN-Eingabeseite dargestellt. Obwohl die Teilnehmer in der zweiten Transaktion dasselbe Sicherungsverfahren verwendeten, mit dem sie auch bei ihrer Hausbank Aufträge freigeben, und aufgrund ihres Arbeitsumfelds als besonders technikaffin bezeichnet werden konnten, hat weniger als ein Fünftel den Angriff erkannt.

Unsere Ergebnisse zeigen, dass fast die Hälfte der Teilnehmer in beiden Transaktionen keine Verifikation der Auftragsdaten im Sicherungsverfahren vorgenommen hat. Bei den Probanden, die in der ersten Transaktion noch einen ordentlichen Vergleich mit den Rechnungsdaten durchführten, sorgte die Anzeige der Auftragsdaten auf der Bestätigungsseite für eine fehlerhafte Transaktionsverifikation: Statt die Daten im Sicherungsverfahren mit der Rechnung zu vergleichen, hat ein Großteil die durch den PC-Bildschirm dargestellten Auftragsdaten auf der Bestätigungsseite herangezogen. Da sich dort kein Unterschied ergab, bestätigten sie die manipulierte Transaktion.

Für die Erkennung des Angriffs spielte es neben einem technischen Hintergrund und der Erfahrung mit mehreren Sicherungsverfahren auch eine Rolle, welches Verfahren konkret zum Einsatz kam: Bei Nutzern des pushTAN-Verfahrens war es besonders wahrscheinlich, während es beim smsTAN-Verfahren besonders unwahrscheinlich war, den Angriff zu erkennen. Das chipTAN-Verfahren schnitt leicht besser als der Durchschnitt ab, zeigte aber keinen signifikanten Unterschied zu den anderen beiden Verfahren.

Der von uns implementierte und evaluierte Angriff bezieht sich auf das Verhalten, das Banken bei der Durchführung von Transaktionen tatsächlich an den Tag legen. Bei der überwiegenden Mehrheit der Banken ist es üblich, die Transaktionsdaten auf die Legitimierungsseite zu spiegeln. Dieses Vorgehen ist kontraproduktiv und suggeriert dem Kunden, dass der transaktionsauslösende Kanal als vertrauenswürdig zu betrachten ist. Es widerspricht auch den Sorgfaltspflichten, die die Banken

den Kunden auferlegen und die ihn auffordern, die Transaktion mit dem Originalbeleg zu verifizieren. Obwohl die Banken das Problem durchaus anerkennen, haben nur die Genossenschaftsbanken signalisiert, in Zukunft auf die Anzeige von Transaktionsdetails auf der Bestätigungsseite zu verzichten.

Zwar attestierte Kapitel 6 der Regulierung durch die PSD2 sinnvolle technische Vorgaben, die aber nur sehr eingeschränkt dazu beitragen, dass auch sicher benutzbare Verfahren entstehen. Das wird bereits anhand der schädlichen und wenig durchdachten Handhabung der Bestätigungsseiten deutlich. Wir haben deshalb Maßnahmen vorgeschlagen, um eine sichere Durchführung von Transaktionen aus Nutzersicht sowohl auf kurze wie auch auf lange Sicht zu fördern, statt wie die Regulatorik durch die PSD2 sichere Transaktion rein funktional zu ermöglichen.

8

Schluss

Banking ist notwendig, Banken sind es nicht.

— Bill Gates, 1994

In diesem letzten Kapitel fassen wir die Ergebnisse der Dissertation zunächst zusammen, ehe wir die Arbeit mit einem Ausblick auf weitere Forschungsfelder abschließen.

8.1 Zusammenfassung

Im Folgenden fassen wir die Kernaussagen der Arbeit in Form von sieben Thesen zusammen und erläutern sie kurz. Die Thesen folgen im Wesentlichen dem Aufbau der Dissertation und sind entsprechend in den Kapiteln begründet.

Der Paradigmenwechsel vom Online- zum Mobilebanking führt zu App-basierten Legitimierungsverfahren und löst die Gerätetrennung auf.
Die App-basierten Verfahren lassen sich in drei Kategorien einteilen, die immer auch im Online-, verfahrensbedingt aber nicht zwangsläufig im Mobilebanking verwendet werden können: 1) Das Verfahren fordert zwingend ein zweites Gerät, 2) findet auf demselben Gerät, aber in zwei unterschiedlichen Apps statt, oder 3) nutzt eine App auf einem Gerät für den kompletten Vorgang. Während 1) noch in Kontinuität zu dem bisher üblichen Medienbruch zwischen Transaktionsauslösung und -bestätigung steht, brechen 2) und 3) mit dem Konzept.

Mobilebanking und App-basierte Legitimierungsverfahren führen zu neuen konzeptionellen Angriffsmöglichkeiten.
Dadurch, dass die App-basierten Verfahren in Software implementiert sind, können sie systembedingt auf ein anderes Gerät kopiert werden. Zwar existieren mittlerweile Möglichkeiten zur effektiven Gerätebedingung, die notwendigen Hardware- und Softwarevoraussetzungen bringen jedoch bei weitem noch nicht alle Geräte mit sich und werden selbst dann teilweise nicht genutzt. Ein weiterer Angriffsvektor bezieht sich auf die Echtzeitmanipulation von Transaktionsdaten im Mobilebanking, die unabhängig zum Replikationsangriff möglich ist. Ursächlich ist das Fehlen einer sicheren Anzeige, was dazu führt, dass das WYSIWYS-Prinzip ausgehebelt wird. Mit der Verfügbarkeit und allgemeinen Verbreitung der für eine sichere Anzeige auf mobilen Endgeräten notwendigen Hardware- und Softwaremaßnahmen ist erst mittel- bis langfristig zu rechnen.

Härtungsmaßnahmen Dritter können die konzeptionellen Schwächen des Mobilebankings systembedingt nur unzureichend adressieren.
Gerade die etablierten Banken sind sich der erhöhten Angriffsfläche bei App-basierten Mobilebanking-Verfahren bewusst. Aus diesem Grund nutzen sie softwarebasierte Härtungsmaßnahmen kommerzieller Drittanbieter, die die App aus demselben Rechtekontext heraus schützen soll. Das daraus resultierende Schutzniveau ist systembedingt sehr eingeschränkt und kann umgangen, deaktiviert oder sogar umgekehrt werden. Die Dominanz eines einzigen Herstellers auf dem deutschen Mobilebanking-Markt führt außerdem dazu, dass sich der Schutz der wichtigsten Banking-Apps und App-basierten Legitimierungsverfahren mit demselben Angriff vollständig ausschalten lassen.

Die einseitige Fokussierung auf das Benutzererlebnis des Kunden geht zulasten der IT-Sicherheit von Fintech-Apps.
Fintechs setzen durch innovative und einfach zu bedienende App-basierte Lösungen die etablierten Finanzinstitute unter Druck. Damit leisten sie auch einen wesentlichen Beitrag zum Verfall zentraler, konzeptioneller Sicherheitseigenschaften wie der Gerätetrennung bei Bankgeschäften. Der einseitige Fokus auf möglichst ansprechende Lösungen führt jedoch sogar bei finanziell sehr gut ausgestatteten Fintechs zu einer strukturellen Vernachlässigung der Sicherheit.

Die regulatorischen Vorschriften erhöhen die Sicherheit im Online- und Mobilebanking, bieten aber einen hohen Interpretationsspielraum.
Mit der starken Kundeauthentifizierung führen die RTS der PSD2 sinnvolle regulatorische Vorgaben ein, um die allgemeinen Sicherheitsziele bei digitalen Bankge-

schäften zu fördern. Mit dem Inkrafttreten der RTS am 14. September 2019 werden die weitverbreiteten Verfahren iTAN und smsTAN nicht mehr konform zu den Anforderungen sein. Selbiges gilt für App-basierte Sicherungsverfahren, die neben der Banking-App auf demselben Gerät betrieben werden. Dennoch ist nicht zu erwarten, dass die europäische oder die nationale Bankenaufsichtsbehörde die Vorgaben rigoros umsetzt, weshalb die smsTAN und die App-basierten Verfahren vermutlich in ihrer bestehenden Form weiter Einsatz finden werden.

Das Angriffspotenzial bleibt auch im Geltungsbereich der RTS hoch.
Die regulatorischen Vorschriften erfassen hauptsächlich technische Aspekte zur Absicherung der Transaktionsbestätigung. Gerade aufseiten der Transaktionsauslösung bleibt das Angriffspotenzial jedoch hoch und wird durch die erhöhten Sicherungsstandards bei den Legitimierungsverfahren für Kriminelle attraktiver werden. Angreifbar sind insbesondere die Zwischenablage, digitale Rechnungen, Überweisungsvorlagen, die SMS-Autovervollständigung unter iOS und die Verifikation durch den Nutzer.

Selbst technisch hochsichere Verfahren führen in der Praxis nicht zu sicheren Transaktionen.
Obwohl insbesondere Verfahren mit dedizierter Hardware zur Absicherung von Transaktionen über herausragende Sicherheitseigenschaften verfügen, werden sie von den Kunden oft nicht richtig verwendet. Ein wesentlicher Grund besteht darin, dass sich der Verwender oft nicht im Klaren darüber ist, welche Elemente im Transaktionsprozess vertrauenswürdig sind. Die Bank fördert dieses Verhalten durch das Darstellen irreführender Informationen. Die Industrie und Wissenschaft sind deshalb gefordert, Lösungen zu entwickeln, die nicht nur in der Theorie sichere Bankgeschäfte ermöglichen, sondern auch eine richtige Verwendung durch den Nutzer in der Praxis fördern.

8.2 Ausblick

Obwohl die Dissertation einen wertvollen Beitrag zur Sicherheit von Bankgeschäften liefert, steht der Umbruch in der Branche der mobilen Finanzdienstleistungen noch am Anfang. Es ergeben sich deshalb zahlreiche weitere Untersuchungsgegenstände, von denen wir im Folgenden einige motivieren möchten.

Implementierung. Die Arbeit hat die verschiedenen Ausprägungen App-basierter Verfahren kategorisiert und grundsätzliche Angriffsmöglichkeiten dargelegt. Daneben haben wir aber auch Ansätze vorgestellt, mit denen sich die Angriffe durch die Verwendung von Hardwaremaßnahmen wirkungsvoll verhindern lassen. Es ist deshalb essentiell, dass die Banken und Fintechs diese Funktionen möglichst vollständig ausschöpfen. Weitere Forschungsarbeiten sollten sich deshalb damit auseinandersetzen, ob Hardwaremaßnahmen genutzt werden und ob sie auch korrekt implementiert wurden [Bia+18; Ege+13].

Konformität. Die Ergebnisse einer solchen Untersuchung wären auch unmittelbar für die Aufsichtsbehörden von Bedeutung, da der Regulator selbst keine technischen Überprüfungen durchführt. Stattdessen müssen die Institute den Behörden auf Anfrage glaubhaft machen, dass sie den regulatorischen Ansprüchen genügen. Deshalb ist davon auszugehen, dass die Finanzaufsicht einen Bericht zur technischen Umsetzung dankbar aufgreifen würde. Ein weiterer Forschungsgegenstand in Bezug auf die Eignung zur starken Kundenauthentifizierung sind Inhärenzelemente, die nicht auf biometrischen Merkmalen fußen, sondern eine Authentifizierung anhand des Benutzerverhaltens durchführen.

Registrierung und Identifizierung. Die Dissertation hat auch deutlich gemacht, dass der Registrierungsprozess von App-basierten Verfahren eine Schwachstelle darstellen kann. Die Forschung sollte sich deshalb mit den Sicherheitsmerkmalen auseinandersetzen, die die Institute für die Einrichtung und Personalisierung ihrer Apps einsetzen. Vor diesem Hintergrund stellt sich auch die Frage nach der Sicherheit von Identifizierungsdienstleistern, die den Nutzer online ausweisen. Gerade Fintechs setzen an dieser Stelle auf Anbieter, die zum Teil nur ein Foto des Personalausweises und der zu identifizierenden Person verlangen [BL18].

PSD2 APIs. Die Arbeit hat sich nur mit den regulatorischen Vorgaben durch die PSD2 beschäftigt, die für die Sicherheit beim Tätigen von Transaktionen im Online- und Mobilebanking zentral sind. Mit dem Inkrafttreten der RTS sind die Banken aber nicht nur im Bereich der starken Kundenauthentifizierung gefordert, sondern müssen sich auch gegenüber Dritten öffnen. Zu diesem Zweck sieht die Regulatorik vor, dass die Banken Programmierschnittstellen (APIs) anbieten, die dann von Kontoinformationsdiensten und Zahlungsauslösediensten genutzt werden können [Sch19c]. Diese Dienste müssen sich bei den Aufsichtsbehörden zwar lizensieren, leiten aber dennoch einen weiteren Paradigmenwechsel ein: Bisher galt die Vorgabe, dass der Nutzer die Zugangsdaten zu seinem Onlinebanking lediglich bei seiner Bank eingibt. Durch die Implementierung der Drittdienste wird damit gebrochen, da die PSD2

diesen Diensten das Abfragen und Verarbeiten der Zugangsdaten explizit gestattet [Alb18]. Vonseiten der IT-Sicherheit stellt sich vor allem die Frage, ob diese Schnittstellen sicher konzipiert und implementiert werden [FKS16; FKS17; Rup+19].

Fintech-Sicherheit. Außerdem sorgt die Zwangsöffnung der Banken dafür, dass die hochsensiblen Zahlungsdaten der Kunden zukünftig bei verschiedenen Anbietern – zumeist Fintechs – vorgehalten werden. Die Dissertation hat gezeigt, dass bei einem vollständig regulierten Unternehmen mitnichten davon ausgegangen werden kann, dass es auch besonders sicher ist. Die Forschung kann an dieser Stelle nicht nur einen wichtigen Beitrag dazu leisten, dass Schäden von den Nutzern abgewendet und diese sensibilisiert werden, sondern auch dazu beitragen, dass gerade Fintechs der Sicherheit grundsätzlich mehr Bedeutung beimessen.

Es sollte auch der Argumentation entgegengetreten werden, die den Rückbau von Sicherheitseigenschaften mit dem Verweis auf die Haftungsübernahme durch das Institut zu rechtfertigen versucht. Diese Betrachtung greift zu kurz und vernachlässigt den sozio-ökonomischen Schaden, der auch dann zurückbleibt, wenn finanzielle Verluste ausgeglichen werden. Sicherheit bedeutet nicht, dass man sich sicher sein kann, sein Geld zurückzuerhalten, sondern es erst gar nicht zu verlieren.

Abkürzungen

1AA	Ein-App-Authentifizierung
2AA	Zwei-App-Authentifizierung
2FA	Zwei-Faktor-Authentifizierung
2GA	Zwei-Geräte-Authentifizierung
AGB	Allgemeinen Geschäftsbedingungen
APK	Android Package Kit
BaFin	Bundesanstalt für Finanzdienstleistungsaufsicht
BDB	Bundesverband deutscher Banken
BSI	Bundesamt für Sicherheit in der Informationstechnik
BVR	Bundesverband der Deutschen Volks- und Raiffeisenbanken
DK	Deutsche Kreditwirtschaft, ehemals Zentraler Kreditausschuss
DSGV	Deutscher Sparkassen- und Giroverband
DSGVO	Datenschutz-Grundverordnung
EBA	Europäische Bankenaufsichtsbehörde
EMV	Europay International, MasterCard und VISA
EU	Europäische Union
Fintech	Finanz-Start-up
IBAN	Internationale Bankkontonummer
PIN	Persönliche Identifikationsnummer
PoS	Point of Sale
PSD1	Zahlungsdiensterichtlinie I
PSD2	Zahlungsdiensterichtlinie II
RASP	Runtime Application Self-Protection
RTS	Technischen Regulierungsstandards
SIM	Subscriber Identity Module
SMS	Kurznachrichtendienst (Short Message Service)
TAN	Transaktionsnummer
WYSIWYS	What-You-See-Is-What-You-Sign

Literatur

[ABl15a] „Richtlinie (EU) 2015/2366 des Europäischen Parlaments und des Rates vom 25. November 2015 über Zahlungsdienste im Binnenmarkt, zur Änderung der Richtlinien 2002/65/EG, 2009/110/EG und 2013/36/EU und der Verordnung (EU) Nr. 1093/2010 sowie zur Aufhebung der Richtlinie 2007/64/EG“. In: *Amtsblatt der Europäischen Union* (23. Dez. 2015), S. 35–127. URL: https://eur-lex.europa.eu/legal-content/DE/TXT/PDF/?uri=CELEX:32015L2366 (siehe S. 3, 113).

[ABl15b] „Richtlinie 2007/64/EG des Europäischen Parlaments und des Rates vom 13. November 2007 über Zahlungsdienste im Binnenmarkt, zur Änderung der Richtlinien 97/7/EG, 2002/65/EG,2005/60/EG und 2006/48/EG sowie zur Aufhebung der Richtlinie 97/5/EG“. In: *Amtsblatt der Europäischen Union* (23. Dez. 2015), S. 1–36. URL: https://eur-lex.europa.eu/legal-content/DE/TXT/PDF/?uri=CELEX:32007L0064 (siehe S. 113).

[ABl18] „Delegierte Verordnung (EU) 2018/389 der Kommission vom 27. November 2017 zur Ergänzung der Richtlinie (EU) 2015/2366 des Europäischen Parlaments und des Rates durch technische Regulierungsstandards für eine starke Kundenauthentifizierung und für sichere offene Standards für die Kommunikation“. In: *Amtsblatt der Europäischen Union* (13. März 2018), S. 23–43. URL: https://eur-lex.europa.eu/legal-content/DE/TXT/PDF/?uri=CELEX:32018R0389 (siehe S. 3, 113).

[AD16] Elisabeth Atzler und Frank M. Drost. „Gratis gibt’s nicht mehr“. In: *Handelsblatt* 228 (24. Nov. 2016) (siehe S. 2).

[Aer+17] Maarten Aertsen, Maciej Korczynski, Giovane C. M. Moura, Samaneh Tajalizadehkhoob und Jan van den Berg. „No domain left behind: is Let’s Encrypt democratizing encryption?“ In: *Proceedings of the Applied Networking Research Workshop, Prague, Czech Republic, July 15, 2017*. ACM, 2017, S. 48–54. DOI: 10.1145/3106328.3106338 (siehe S. 95).

[Alb18] Andreas Albert. „Warum Sie öfter nach der PIN gefragt werden“. In: *Spiegel Online* (29. Juli 2018). URL: https://spon.de/afhwr (besucht am 24. 04. 2019) (siehe S. 173).

[Ama+17] Johanna Amann, Oliver Gasser, Quirin Scheitle, Lexi Brent, Georg Carle und Ralph Holz. „Mission accomplished?: HTTPS security after diginotar“. In: *Proceedings of the 2017 Internet Measurement Conference, IMC 2017, London, United Kingdom, November 1-3, 2017*. Hrsg. von Steve Uhlig und Olaf Maennel. ACM, 2017, S. 325–340. DOI: 10.1145/3131365.3131401 (siehe S. 95).

[Aon+18] Simone Aonzo, Alessio Merlo, Giulio Tavella und Yanick Fratantonio. „Phishing Attacks on Modern Android“. In: *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, CCS 2018, Toronto, ON, Canada, October 15-19, 2018*. Hrsg. von David Lie, Mohammad Mannan, Michael Backes und XiaoFeng Wang. ACM, 2018, S. 1788–1801. DOI: 10.1145/3243734.3243778 (siehe S. 104).

[AOSP] Android Open Source Project. *Android Security Bulletins*. URL: https://source.android.com/security/bulletin (besucht am 20. 03. 2019) (siehe S. 37).

[APC] Apple. *Apple security updates*. URL: https://support.apple.com/en-us/HT201222 (besucht am 20. 03. 2019) (siehe S. 37).

[Atz18] Elisabeth Atzler. „Der Kampf um die Kunden der Zukunft“. In: *Handelsblatt* 162 (23. Aug. 2018) (siehe S. 2).

[Bac+15] Michael Backes, Sven Bugiel, Christian Hammer, Oliver Schranz und Philipp von Styp-Rekowsky. „Boxify: Full-fledged App Sandboxing for Stock Android“. In: *24th USENIX Security Symposium, USENIX Security 15, Washington, D.C., USA, August 12-14, 2015*. Hrsg. von Jaeyeon Jung und Thorsten Holz. USENIX Association, 2015, S. 691–706. URL: https://www.usenix.org/conference/usenixsecurity15/technical-sessions/presentation/backes (siehe S. 127).

[Bac18] Kerstin Backofen. „Onlinebanking: Keine Bange vor der Onlinebank“. In: *Finanztest* 11 (1. Nov. 2018), S. 12–17 (siehe S. 124).

[BaFin17] Bundesanstalt für Finanzdienstleistungsaufsicht. *Jahresbericht 2016*. 9. Mai 2017. URL: https://www.bafin.de/SharedDocs/Downloads/DE/Jahresbericht/dl_jb_2016.pdf?__blob=publicationFile (besucht am 15. 04. 2019) (siehe S. 113).

[Bar+01] Boaz Barak, Oded Goldreich, Russell Impagliazzo, Steven Rudich, Amit Sahai, Salil P. Vadhan und Ke Yang. „On the (Im)possibility of Obfuscating Programs“. In: *Advances in Cryptology - CRYPTO 2001, 21st Annual International Cryptology Conference, Santa Barbara, California, USA, August 19-23, 2001, Proceedings*. 2001, S. 1–18 (siehe S. 64).

[BD19] Parnika Bhat und Kamlesh Dutta. „A Survey on Various Threats and Current State of Security in Android Platform“. In: *ACM Comput. Surv.* 1 (2019), 21:1–21:35. URL: https://dl.acm.org/citation.cfm?id=3301285 (siehe S. 36).

[BG17] Bundesverband Deutscher Banken und Gesellschaft für Konsumforschung. *Peters: Banken punkten bei Datensicherheit*. 6. Apr. 2017. URL: https://bankenverband.de/newsroom/presse-infos/peters-banken-punkten-bei-datensicherheit/ (besucht am 07. 03. 2019) (siehe S. 88).

[BGH16] Bundesgerichtshof. *Urteil des XI. Zivilsenats vom 26.1.2016*. Aktenzeichen XI ZR 91/14. 26. Jan. 2016. URL: http://oj.is/881954 (besucht am 15. 04. 2019) (siehe S. 110).

[BGL17] Zinaida Benenson, Freya Gassmann und Robert Landwirth. „Unpacking Spear Phishing Susceptibility“. In: *Financial Cryptography and Data Security - FC 2017 International Workshops, WAHC, BITCOIN, VOTING, WTSC, and TA, Sliema, Malta, April 7, 2017, Revised Selected Papers*. Hrsg. von Michael Brenner, Kurt Rohloff, Joseph Bonneau, Andrew Miller, Peter Y. A. Ryan, Vanessa Teague, Andrea Bracciali, Massimiliano Sala, Federico Pintore und Markus Jakobsson. Bd. 10323. Lecture Notes in Computer Science. Springer, 2017, S. 610–627. DOI: 10.1007/978-3-319-70278-0_39 (siehe S. 104).

[Bia+17] Antonio Bianchi, Eric Gustafson, Yanick Fratantonio, Christopher Kruegel und Giovanni Vigna. „Exploitation and Mitigation of Authentication Schemes Based on Device-Public Information“. In: *Proceedings of the 33rd Annual Computer Security Applications Conference, Orlando, FL, USA, December 4-8, 2017*. ACM, 2017, S. 16–27. DOI: 10.1145/3134600.3134615 (siehe S. 55).

[Bia+18] Antonio Bianchi, Yanick Fratantonio, Aravind Machiry, Christopher Kruegel, Giovanni Vigna, Simon Pak Ho Chung und Wenke Lee. „Broken Fingers: On the Usage of the Fingerprint API in Android“. In: *25th Annual Network and Distributed System Security Symposium, NDSS*

2018, San Diego, California, USA, February 18-21, 2018. The Internet Society, 2018. URL: http://wp.internetsociety.org/ndss/wp-content/uploads/sites/25/2018/02/ndss2018_03B-1_Bianchi_paper.pdf (siehe S. 56, 127, 172).

[Bic+16] Benjamin Bichsel, Veselin Raychev, Petar Tsankov und Martin T. Vechev. „Statistical Deobfuscation of Android Applications". In: *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, October 24-28, 2016*. 2016, S. 343–355 (siehe S. 63).

[BK16] Jens Bender und Dennis Kügler. „Was ist starke Authentisierung?" In: *Datenschutz und Datensicherheit* 4 (2016), S. 212–216. DOI: 10.1007/s11623-016-0580-3 (siehe S. 17).

[BKY06] Yoav Benjamini, Abba M. Krieger und Daniel Yekutieli. „Adaptive linear step-up procedures that control the false discovery rate". In: *Biometrika* 3 (2006), S. 491–507. DOI: 10.1093/biomet/93.3.491 (siehe S. 151).

[BL18] Melanie Bergermann und Saskia Littmann. „Bullshit Banking". In: *WirtschaftsWoche* 42 (12. Okt. 2018), S. 52 (siehe S. 107, 172).

[BPC11] Mark Blythe, Helen Petrie und John A. Clark. „F for fake: four studies on how we fall for phish". In: *Proceedings of the International Conference on Human Factors in Computing Systems, CHI 2011, Vancouver, BC, Canada, May 7-12, 2011*. Hrsg. von Desney S. Tan, Saleema Amershi, Bo Begole, Wendy A. Kellogg und Manas Tungare. ACM, 2011, S. 3469–3478. DOI: 10.1145/1978942.1979459 (siehe S. 159).

[Brü14] Volker Brühl. „Die Digitalisierung revolutioniert das Bankgeschäft". In: *Frankfurter Allgemeine Zeitung* 228 (1. Okt. 2014), S. 25 (siehe S. 88).

[BSI] Bundesamt für Sicherheit in der Informationstechnik. *Sicherheit im Online-Banking: Das mTAN-Verfahren*. URL: https://www.bsi-fuer-buerger.de/BSIFB/DE/DigitaleGesellschaft/OnlineBanking/SoFunktioniertDasOnlineBanking/Sicherheit/PIN-TAN-Schutzverfahren.html?nn=6596940&cms_pos=3 (besucht am 16. 03. 2019) (siehe S. 23).

[BSI18] Bundesamt für Sicherheit in der Informationstechnik. *Die Lage der IT-Sicherheit in Deutschland 2018*. 10. Okt. 2018. URL: https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/Lageberichte/Lagebericht2018.pdf (besucht am 27. 03. 2019) (siehe S. 79).

[BuM05] „Postbank: Mit PhishingSicherheit in die Presse“. In: *Bank und Markt* 9 (1. Sep. 2005), S. 7. ISSN: 1433-5204 (siehe S. 21).

[Cap+14] Deanna D. Caputo, Shari Lawrence Pfleeger, Jesse D. Freeman und M. Eric Johnson. „Going Spear Phishing: Exploring Embedded Training and Awareness“. In: *IEEE Security & Privacy* 1 (2014), S. 28–38. DOI: 10.1109/MSP.2013.106 (siehe S. 98, 104).

[Car+18] Michele Carminati, Alessandro Baggio, Federico Maggi, Umberto Spagnolini und Stefano Zanero. „FraudBuster: Temporal Analysis and Detection of Advanced Financial Frauds“. In: *Detection of Intrusions and Malware, and Vulnerability Assessment - 15th International Conference, DIMVA 2018, Saclay, France, June 28-29, 2018, Proceedings*. Hrsg. von Cristiano Giuffrida, Sébastien Bardin und Gregory Blanc. Bd. 10885. Lecture Notes in Computer Science. Springer, 2018, S. 211–233. DOI: 10.1007/978-3-319-93411-2_10 (siehe S. 129, 150).

[CCK15] Junsung Cho, Geumhwan Cho und Hyoungshick Kim. „Keyboard or keylogger?: A security analysis of third-party keyboards on Android“. In: *13th Annual Conference on Privacy, Security and Trust, PST 2015, Izmir, Turkey, July 21-23, 2015*. Hrsg. von Ali A. Ghorbani, Vicenç Torra, Hüseyin Hisil, Ali Miri, Ahmet Koltuksuz, Jie Zhang, Murat Sensoy, Joaquín García-Alfaro und Ibrahim Zincir. IEEE Computer Society, 2015, S. 173–176. DOI: 10.1109/PST.2015.7232970 (siehe S. 65).

[Che+15] Kai Chen, Peng Wang, Yeonjoon Lee, XiaoFeng Wang, Nan Zhang, Heqing Huang, Wei Zou und Peng Liu. „Finding Unknown Malice in 10 Seconds: Mass Vetting for New Threats at the Google-Play Scale“. In: *24th USENIX Security Symposium, USENIX Security 15, Washington, D.C., USA, August 12-14, 2015*. Hrsg. von Jaeyeon Jung und Thorsten Holz. USENIX Association, 2015, S. 659–674. URL: https://www.usenix.org/conference/usenixsecurity15/technical-sessions/presentation/chen-kai (siehe S. 38).

[Che+16] Kai Chen, Xueqiang Wang, Yi Chen, Peng Wang, Yeonjoon Lee, XiaoFeng Wang, Bin Ma, Aohui Wang, Yingjun Zhang und Wei Zou. „Following Devil’s Footprints: Cross-Platform Analysis of Potentially Harmful Libraries on Android and iOS“. In: *IEEE Symposium on Security and Privacy, SP 2016, San Jose, CA, USA, May 22-26, 2016*. IEEE Computer Society, 2016, S. 357–376. DOI: 10.1109/SP.2016.29 (siehe S. 38).

[Che+18] Sen Chen, Ting Su, Lingling Fan, Guozhu Meng, Minhui Xue, Yang Liu und Lihua Xu. „Are mobile banking apps secure? what can be improved?“ In: *Proceedings of the 2018 ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, ESEC/SIGSOFT FSE 2018, Lake Buena Vista, FL, USA, November 04-09, 2018*. Hrsg. von Gary T. Leavens, Alessandro Garcia und Corina S. Pasareanu. ACM, 2018, S. 797–802. DOI: 10.1145/3236024.3275523 (siehe S. 65).

[Cho+02] Stanley Chow, Philip A. Eisen, Harold Johnson und Paul C. van Oorschot. „White-Box Cryptography and an AES Implementation“. In: *Selected Areas in Cryptography, 9th Annual International Workshop, SAC 2002, St. John's, Newfoundland, Canada, August 15-16, 2002. Revised Papers*. Hrsg. von Kaisa Nyberg und Howard M. Heys. Bd. 2595. Lecture Notes in Computer Science. Springer, 2002, S. 250–270 (siehe S. 64).

[Cim] Catalin Cimpanu. *Google warns about two iOS zero-days 'exploited in the wild'*. URL: https://zd.net/2SfMe0e (besucht am 20. 03. 2019) (siehe S. 39).

[CKV10] Marco Cova, Christopher Krügel und Giovanni Vigna. „Detection and analysis of drive-by-download attacks and malicious JavaScript code“. In: *Proceedings of the 19th International Conference on World Wide Web, WWW 2010, Raleigh, North Carolina, USA, April 26-30, 2010*. Hrsg. von Michael Rappa, Paul Jones, Juliana Freire und Soumen Chakrabarti. ACM, 2010, S. 281–290. DOI: 10.1145/1772690.1772720 (siehe S. 67).

[CN09] Christian Collberg und Jasvir Nagra. *Surreptitious Software: Obfuscation, Watermarking, and Tamperproofing for Software Protection*. 1. Aufl. Addison-Wesley Professional, 2009. ISBN: 9780321549259 (siehe S. 63).

[CoBa] Commerzbank. *Fragen & Antworten zu den Mobile Banking Apps*. URL: https://www.commerzbank.de/portal/de/privatkunden/service-und-hilfe/ihre-wege-zu-uns/mobile-banking-apps/faq-s/faqs.html (besucht am 23. 02. 2018) (siehe S. 59).

[CoBa19] Commerzbank. *Aktueller Warnhinweis: Warnung vor Phishing mit Aufforderung zum Upload der TAN Liste*. 2019. URL: https://www.commerzbank.de/portal/de/privatkunden/service-und-hilfe/sicherheit/ihr-online-banking/aktuelle-warnhinweise/tan-upload/tanupload.html (besucht am 22. 03. 2019) (siehe S. 47, 48).

[CQM14] Qi Alfred Chen, Zhiyun Qian und Zhuoqing Morley Mao. „Peeking into Your App without Actually Seeing It: UI State Inference and Novel Android Attacks“. In: *Proceedings of the 23rd USENIX Security Symposium, San Diego, CA, USA, August 20-22, 2014*. Hrsg. von Kevin Fu und Jaeyeon Jung. USENIX Association, 2014, S. 1037–1052. URL: https://www.usenix.org/conference/usenixsecurity14/technical-sessions/presentation/chen (siehe S. 96).

[CV18] Luca Casati und Andrea Visconti. „The Dangers of Rooting: Data Leakage Detection in Android Applications“. In: *Mobile Information Systems* (2018), 6020461:1–6020461:9. DOI: 10.1155/2018/6020461 (siehe S. 39).

[Dab+14] Adrian Dabrowski, Nicola Pianta, Thomas Klepp, Martin Mulazzani und Edgar R. Weippl. „IMSI-catch me if you can: IMSI-catcher-catchers“. In: *Proceedings of the 30th Annual Computer Security Applications Conference, ACSAC 2014, New Orleans, LA, USA, December 8-12, 2014*. Hrsg. von Charles N. Payne Jr., Adam Hahn, Kevin R. B. Butler und Micah Sherr. ACM, 2014, S. 246–255. DOI: 10.1145/2664243.2664272 (siehe S. 24).

[Dan18] Janis Danisevskis. *Android Protected Confirmation: Taking transaction security to the next level*. 19. Okt. 2018. URL: https://android-developers.googleblog.com/2018/10/android-protected-confirmation.html (besucht am 23. 03. 2019) (siehe S. 57).

[Das+14] Anupam Das, Joseph Bonneau, Matthew Caesar, Nikita Borisov und XiaoFeng Wang. „The Tangled Web of Password Reuse“. In: *21st Annual Network and Distributed System Security Symposium, NDSS 2014, San Diego, California, USA, February 23-26, 2014*. The Internet Society, 2014. URL: https://www.ndss-symposium.org/ndss2014/tangled-web-password-reuse (siehe S. 104).

[Dav+10] Lucas Davi, Alexandra Dmitrienko, Ahmad-Reza Sadeghi und Marcel Winandy. „Privilege Escalation Attacks on Android“. In: *Information Security - 13th International Conference, ISC 2010, Boca Raton, FL, USA, October 25-28, 2010, Revised Selected Papers*. Hrsg. von Mike Burmester, Gene Tsudik, Spyros S. Magliveras und Ivana Ilic. Bd. 6531. Lecture Notes in Computer Science. Springer, 2010, S. 346–360. DOI: 10.1007/978-3-642-18178-8_30 (siehe S. 39).

[DBB18] Deutsche Bundesbank. *Zahlungsverhalten in Deutschland 2017*. 13. Feb. 2018. URL: https://www.bundesbank.de/resource/blob/737876/40094ed787ec5b0dd1f968dcd7eda7e9/mL/zahlungsverhalten-in-deutschland-2017-praesentation-data.pdf (siehe S. 2).

[DDC18] Sanchari Das, Andrew Dingman und L Jean Camp. „Why Johnny Doesn't Use Two Factor: A Two-Phase Usability Study of the FIDO U2F Security Key". In: *Financial Cryptography and Data Security - 22nd International Conference, FC 2018, Curaçao, February 26 - March 2, 2018, Revised Selected Papers*. 2018 (siehe S. 141).

[DH18a] Gregor Dorfleitner und Lars Hornuf. *Analyse der Datenschutzerklärungen deutscher Fintech-Unternehmen nach Einfürhung der DS-GVO*. Techn. Ber. Abida, 13. Dez. 2018, S. 48. URL: http://www.abida.de/sites/default/files/ABIDA_Folgegutachten_Fintech_DSGVO.pdf (siehe S. 88, 89).

[DH18b] Gregor Dorfleitner und Lars Hornuf. *Neue digitale Akteure und Ihre Rolle in der Finanzwirtschaft*. Techn. Ber. Abida, 26. März 2018, S. 100. URL: http://www.abida.de/sites/default/files/Gutachten_ABIDA_Neue_Digitale_Akteure_Finanzwirtschaft.pdf (siehe S. 89).

[DK] Die Deutsche Kreditwirtschaft. *mobileTAN*. URL: https://die-dk.de/zahlungsverkehr/electronic-banking/mobiletan (besucht am 14. 12. 2017) (siehe S. 23, 32).

[DK14] Die Deutsche Kreditwirtschaft. *DK-Kompendium Online-Banking-Sicherheit*. Feb. 2014. URL: https://www.hbci-zka.de/dokumente/diverse/DK%20Kompendium%20Online-Banking-Sicherheit%20V1.2%20final%20version.pdf (besucht am 16. 03. 2019) (siehe S. 18, 26).

[DK17] Die Deutsche Kreditwirtschaft. *Berichterstattung zur Sicherheit von Banking-Apps*. 23. Nov. 2017. URL: https://die-dk.de/themen/pressemitteilungen/berichterstattung-zur-sicherheit-von-banking-apps (besucht am 28. 03. 2018) (siehe S. 79).

[DK18] Die Deutsche Kreditwirtschaft. *ZKA-TAN-Generator: Belegungsrichtlinien für das chipTAN-Verfahren*. 16. Apr. 2018. URL: https://www.hbci-zka.de/dokumente/spezifikation_deutsch/hhd/Belegungsrichtlinien%20TANve1.5%20FV%20vom%202018-04-16.pdf (besucht am 16. 03. 2019) (siehe S. 27).

[DMA09] Saar Drimer, Steven J. Murdoch und Ross J. Anderson. „Optimised to Fail: Card Readers for Online Banking“. In: *Financial Cryptography and Data Security, 13th International Conference, FC 2009, Accra Beach, Barbados, February 23-26, 2009. Revised Selected Papers*. Hrsg. von Roger Dingledine und Philippe Golle. Bd. 5628. Lecture Notes in Computer Science. Springer, 2009, S. 184–200. DOI: 10.1007/978-3-642-03549-4_11 (siehe S. 27).

[Dmi+14] Alexandra Dmitrienko, Christopher Liebchen, Christian Rossow und Ahmad-Reza Sadeghi. „On the (In)Security of Mobile Two-Factor Authentication“. In: *Financial Cryptography and Data Security - 18th International Conference, FC 2014, Christ Church, Barbados, March 3-7, 2014, Revised Selected Papers*. 2014, S. 365–383. DOI: 10.1007/978-3-662-45472-5_24 (siehe S. 44).

[Dor+17] Gregor Dorfleitner, Lars Hornuf, Matthias Schmitt und Martina Weber. *FinTech in Germany*. 1st. Springer Publishing, 2017. ISBN: 9783319546650 (siehe S. 88).

[DPA18] Deutsche Presse Agentur. *Bitkom-Studie: Für Bankhäuser bahnt sich ein Gezeitenwechsel an*. 7. Mai 2018. URL: https://www.handelsblatt.com/21251870.html (besucht am 01. 03. 2019) (siehe S. 1).

[DS14] Nicola Davinson und Elizabeth Sillence. „Using the health belief model to explore users' perceptions of 'being safe and secure' in the world of technology mediated financial transactions“. In: *Int. J. Hum.-Comput. Stud.* 2 (2014), S. 154–168. DOI: 10.1016/j.ijhcs.2013.10.003 (siehe S. 158).

[DS16] Frank M. Drost und Katharina Schneider. „Telefónica greift Banken und Start-ups an“. In: *Handelsblatt* 142 (26. Juli 2016) (siehe S. 31).

[DSZ05] „1822direkt: Mehr Sicherheit beim Online-Banking“. In: *Sparkassen-Zeitung* 24 (17. Juni 2005), S. 20. ISSN: 0012-0766 (siehe S. 21).

[DSZ13] „Mehr Sicherheit“. In: *SparkassenZeitung* 8 (22. Feb. 2013), S. 13. ISSN: 0012-0766 (siehe S. 28, 30).

[DTH06] Rachna Dhamija, J. D. Tygar und Marti A. Hearst. „Why phishing works“. In: *Proceedings of the 2006 Conference on Human Factors in Computing Systems, CHI 2006, Montréal, Québec, Canada, April 22-27, 2006*. Hrsg. von Rebecca E. Grinter, Tom Rodden, Paul M. Aoki, Edward Cutrell, Robin Jeffries und Gary M. Olson. ACM, 2006, S. 581–590. DOI: 10.1145/1124772.1124861 (siehe S. 158).

[Dua+18] Yue Duan, Mu Zhang, Abhishek Vasisht Bhaskar, Heng Yin, Xiaorui Pan, Tongxin Li, Xueqiang Wang und XiaoFeng Wang. „Things You May Not Know About Android (Un)Packers: A Systematic Study based on Whole-System Emulation". In: *25th Annual Network and Distributed System Security Symposium, NDSS, 2018, San Diego, California, USA, February 18 - 21, 2018*. 2018 (siehe S. 64, 82).

[EBA17] Europäische Bankenaufsichtsbehörde. *Final Report Draft Regulatory Technical Standards on Strong Customer Authentication and common and secure communication under Article 98 of Directive 2015/2366 (PSD2)*. 23. Feb. 2017. URL: https://www.eba.europa.eu/documents/10180/1761863/Final+draft+RTS+on+SCA+and+CSC+under+PSD2+%28EBA-RTS-2017-02%29.pdf (siehe S. 113, 128).

[EBA18a] Europäische Bankenaufsichtsbehörde. *Opinion of the European Banking Authority on the implementation of the RTS on SCA and CSC*. 13. Juni 2018. URL: https://eba.europa.eu/documents/10180/2137845/Opinion+on+the+implementation+of+the+RTS+on+SCA+and+CSC+%28EBA-2018-Op-04%29.pdf (siehe S. 114).

[EBA18b] Europäische Bankenaufsichtsbehörde. *Qualification of SMS OTP as an authentication factor*. 5. Okt. 2018. URL: https://eba.europa.eu/single-rule-book-qa/-/qna/view/publicId/2018_4039 (besucht am 16. 04. 2019) (siehe S. 125).

[EBS15] Nathan S. Evans, Azzedine Benameur und Yun Shen. „All your Root Checks are Belong to Us: The Sad State of Root Detection". In: *Proceedings of the 13th ACM International Symposium on Mobility Management and Wireless Access, MobiWac 2015, Cancun, Mexico, November 2-6, 2015*. Hrsg. von Mirela Sechi Moretti Annoni Notare, Ángel Cuevas Rumín und Miguel López-Guerrero. ACM, 2015, S. 81–88. DOI: 10.1145/2810362.2810364 (siehe S. 40).

[Ege+13] Manuel Egele, David Brumley, Yanick Fratantonio und Christopher Kruegel. „An empirical study of cryptographic misuse in android applications". In: *2013 ACM SIGSAC Conference on Computer and Communications Security, CCS'13, Berlin, Germany, November 4-8, 2013*. Hrsg. von Ahmad-Reza Sadeghi, Virgil D. Gligor und Moti Yung. ACM, 2013, S. 73–84. DOI: 10.1145/2508859.2516693 (siehe S. 67, 172).

[Eng14] Tobias Engel. *SS7: Locate. Track. Manipulate.* 31st Chaos Communication Congress (31c3): a new dawn. Chaos Computer Club e.V., 27. Dez. 2014. Vortrag (siehe S. 24).

[ENISA12] Europäische Agentur für Netz- und Informationssicherheit. *Flash note: EU cyber security agency ENISA; "High Roller"online bank robberies reveal security gaps.* Englisch. 5. Juli 2012. URL: https://www.enisa.europa.eu/news/enisa-news/copy_of_eu-cyber-security-agency-enisa-201chigh-roller201d-online-bank-robberies-reveal-security-gaps (besucht am 05. 03. 2019) (siehe S. 3).

[Erm18] Monika Ermert. *Offenlegung von Softwarelücken: Rechtsstreit endet mit Vergleich.* 6. Sep. 2018. URL: https://heise.de/-4156393 (besucht am 28. 03. 2018) (siehe S. 80).

[ES13] Serge Egelman und Stuart E. Schechter. „The Importance of Being Earnest [In Security Warnings]". In: *Financial Cryptography and Data Security - 17th International Conference, FC 2013, Okinawa, Japan, April 1-5, 2013, Revised Selected Papers.* Hrsg. von Ahmad-Reza Sadeghi. Bd. 7859. Lecture Notes in Computer Science. Springer, 2013, S. 52–59. DOI: 10.1007/978-3-642-39884-1_5 (siehe S. 161).

[Esk+19] Saba Eskandarian, Jonathan Cogan, Sawyer Birnbaum, Peh Chang Wei Brandon, Dillon Franke, Forest Fraser, Gaspar Garcia Jr., Eric Gong, Hung T. Nguyen, Taresh K. Sethi, Vishal Subbiah, Michael Backes, Giancarlo Pellegrino und Dan Boneh. „Fidelius: Protecting User Secrets from Compromised Browsers". In: *2019 IEEE Symposium on Security and Privacy, SP 2019, SAN FRANCISCO, CA, USA, May 20-22, 2019.* IEEE Computer Society, 2019 (siehe S. 56).

[Fah+13] Sascha Fahl, Marian Harbach, Marten Oltrogge, Thomas Muders und Matthew Smith. „Hey, You, Get Off of My Clipboard - On How Usability Trumps Security in Android Password Managers". In: *Financial Cryptography and Data Security - 17th International Conference, FC 2013, Okinawa, Japan, April 1-5, 2013, Revised Selected Papers.* 2013, S. 144–161. DOI: 10.1007/978-3-642-39884-1_12 (siehe S. 130).

[FAZ98] „Wer?" In: *Frankfurter Allgemeine Zeitung* 98 (28. Apr. 1998), T2 (siehe S. 2).

[Fel+11] Adrienne Porter Felt, Matthew Finifter, Erika Chin, Steve Hanna und David A. Wagner. „A survey of mobile malware in the wild". In: *SPSM'11, Proceedings of the 1st ACM Workshop Security and Privacy in Smartphones and Mobile Devices, Co-located with CCS 2011, October 17, 2011, Chicago, IL, USA*. Hrsg. von Xuxian Jiang, Amiya Bhattacharya, Partha Dasgupta und William Enck. ACM, 2011, S. 3–14 (siehe S. 38).

[FFS88] Uriel Feige, Amos Fiat und Adi Shamir. „Zero-Knowledge Proofs of Identity". In: *J. Cryptology* 2 (1988), S. 77–94. DOI: 10.1007/BF02351717 (siehe S. 98).

[Fil+11] Atanas Filyanov, Jonathan M. McCune, Ahmad-Reza Sadeghi und Marcel Winandy. „Uni-directional trusted path: Transaction confirmation on just one device". In: *Proceedings of the 2011 IEEE/IFIP International Conference on Dependable Systems and Networks, DSN 2011, Hong Kong, China, June 27-30 2011*. IEEE Compute Society, 2011, S. 1–12. DOI: 10.1109/DSN.2011.5958202 (siehe S. 56).

[FKS16] Daniel Fett, Ralf Küsters und Guido Schmitz. „A Comprehensive Formal Security Analysis of OAuth 2.0". In: *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, October 24-28, 2016*. Hrsg. von Edgar R. Weippl, Stefan Katzenbeisser, Christopher Kruegel, Andrew C. Myers und Shai Halevi. ACM, 2016, S. 1204–1215. DOI: 10.1145/2976749.2978385 (siehe S. 173).

[FKS17] Daniel Fett, Ralf Küsters und Guido Schmitz. „The Web SSO Standard OpenID Connect: In-depth Formal Security Analysis and Security Guidelines". In: *30th IEEE Computer Security Foundations Symposium, CSF 2017, Santa Barbara, CA, USA, August 21-25, 2017*. IEEE Computer Society, 2017, S. 189–202. DOI: 10.1109/CSF.2017.20 (siehe S. 173).

[FLG17] Sadegh Farhang, Aron Laszka und Jens Grossklags. „An Economic Study of the Effect of Android Platform Fragmentation on Security Updates". In: *CoRR* (2017). arXiv: 1712.08222. URL: http://arxiv.org/abs/1712.08222 (siehe S. 38).

[Fox02] Dirk Fox. „Der IMSI-Catcher". In: *Datenschutz und Datensicherheit* 4 (2002) (siehe S. 24).

[Fra+17] Yanick Fratantonio, Chenxiong Qian, Simon P. Chung und Wenke Lee. „Cloak and Dagger: From Two Permissions to Complete Control of the UI Feedback Loop". In: *2017 IEEE Symposium on Security and Privacy, SP 2017, San Jose, CA, USA, May 22-26, 2017*. 2017, S. 1041–1057. DOI: 10.1109/SP.2017.39 (siehe S. 56, 66, 84).

[FS13] Wolfram Funk und Thomas Schickling. „Banking Modern - So geht es richtig". In: *Focus Money* 10 (27. Feb. 2013), S. 66–69 (siehe S. 28).

[FW05] Manfred Funk und Annette Weihrauch. „Sparkasse Siegen bietet als eine der ersten Sparkassen effektiven Schutz gegen Phishing-Attacken". In: *SparkassenZeitung* 38 (23. Sep. 2005), S. 14. ISSN: 0012-0766 (siehe S. 21).

[Gar17] Gartner, Inc. *Market Guide for Application Shielding*. 22. Juni 2017. URL: https://www.gartner.com/doc/3747622/market-guide-application-shielding (besucht am 23. 03. 2019) (siehe S. 61).

[Gas+17] Ioannis Gasparis, Zhiyun Qian, Chengyu Song und Srikanth V. Krishnamurthy. „Detecting Android Root Exploits by Learning from Root Providers". In: *26th USENIX Security Symposium, USENIX Security 2017, Vancouver, BC, Canada, August 16-18, 2017*. Hrsg. von Engin Kirda und Thomas Ristenpart. USENIX Association, 2017, S. 1129–1144. URL: https://www.usenix.org/conference/usenixsecurity17/technical-sessions/presentation/gasparis (siehe S. 38, 40).

[Gib16] Samuel Gibbs. „Dropbox hack leads to leaking of 68m user passwords on the internet". In: *The Guardian* (31. Aug. 2016). URL: https://www.theguardian.com/technology/2016/aug/31/dropbox-hack-passwords-68m-data-breach (besucht am 24. 08. 2017) (siehe S. 104).

[GMQ07] Louis Goubin, Jean-Michel Masereel und Michaël Quisquater. „Cryptanalysis of White Box DES Implementations". In: *Selected Areas in Cryptography, 14th International Workshop, SAC 2007, Ottawa, Canada, August 16-17, 2007, Revised Selected Papers*. Hrsg. von Carlisle M. Adams, Ali Miri und Michael J. Wiener. Bd. 4876. Lecture Notes in Computer Science. Springer, 2007, S. 278–295 (siehe S. 64).

[God17] Björn Godenrath. „1822direkt startet Smartphone-Konto". In: *Börsen-Zeitung* 121 (28. Juni 2017), S. 2 (siehe S. 31).

[Gol+18] Maximilian Golla, Miranda Wei, Juliette Hainline, Lydia Filipe, Markus Dürmuth, Elissa M. Redmiles und Blase Ur. „"What was that site doing with my Facebook password?": Designing Password-Reuse Notifications". In: *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, CCS 2018, Toronto, ON, Canada, October 15-19, 2018*. Hrsg. von David Lie, Mohammad Mannan, Michael Backes und XiaoFeng Wang. ACM, 2018, S. 1549–1566. DOI: 10.1145/3243734.3243767 (siehe S. 104).

[Goo] Google Developers. *Shrink Your Code and Resources*. URL: https://developer.android.com/studio/build/shrink-code.html (besucht am 23. 03. 2019) (siehe S. 63).

[Gre17] Andy Greenberg. *How Hackers Hijacked a Bank's Entire Online Operation*. 4. Apr. 2017. URL: https://www.wired.com/2017/04/hackers-hijacked-banks-entire-online-operation (besucht am 31. 03. 2019) (siehe S. 95).

[Gru] Ben Gruver. *dexlib2*. URL: https://github.com/JesusFreke/smali (besucht am 23. 03. 2019) (siehe S. 45, 76).

[Gut18] Joachim Gutmann. „Im Visier". In: *BSI-Magazin* 2 (10. Okt. 2018), S. 17–19 (siehe S. 108).

[Hai15] Manfred Haider. „1500 Kunden binnen zwei Wochen". In: *Wirtschafts-Blatt* 4785 (12. Feb. 2015), S. 13 (siehe S. 31).

[Hau+18] Vincent Haupert, Dominik Maier, Nicolas Schneider, Julian Kirsch und Tilo Müller. „Honey, I Shrunk Your App Security: The State of Android App Hardening". In: *Detection of Intrusions and Malware, and Vulnerability Assessment - 15th International Conference, DIMVA 2018, Saclay, France, June 28-29, 2018, Proceedings*. Hrsg. von Cristiano Giuffrida, Sébastien Bardin und Gregory Blanc. Bd. 10885. Lecture Notes in Computer Science. Springer, 2018, S. 69–91. DOI: 10.1007/978-3-319-93411-2_4 (siehe S. 79, 80).

[Hau16] Vincent Haupert. „Shut Up and Take My Money! The Red Pill of N26 Security". 33rd Chaos Communication Congress (33c3). Hamburg. 27. Dez. 2016. URL: https://media.ccc.de/v/33c3-7969-shut_up_and_take_my_money. Vortrag (siehe S. 106).

[HB98] „Onlinebanking". In: *Handelsblatt* 183 (23. Sep. 1998) (siehe S. 1, 20).

[Her09] Cormac Herley. „So long, and no thanks for the externalities: the rational rejection of security advice by users“. In: *Proceedings of the 2009 Workshop on New Security Paradigms, Oxford, United Kingdom, September 8-11, 2009*. Hrsg. von Anil Somayaji und Richard Ford. ACM, 2009, S. 133–144. DOI: 10.1145/1719030.1719050 (siehe S. 161).

[Hin18] Markus Hinterberger. „Gekommen, um nicht zu bleiben“. In: *Börse Online* 8 (22. Feb. 2018), S. 76–77 (siehe S. 2).

[Hip14] Andreas Hippin. „Fintech mischt die Finanzbranche auf“. In: *Börsen-Zeitung* 33 (18. Feb. 2014), S. 8 (siehe S. 88).

[Hof14] Norbert Hofmann. „Kontakt auf allen Kanälen“. In: *Süddeutsche Zeitung* 231 (8. Okt. 2014), S. 26 (siehe S. 88).

[HS16] Verena M. I. A. Hartl und Ulrike Schmuntzsch. „Fraud Protection for Online Banking - A User-Centered Approach on Detecting Typical Double-Dealings Due to Social Engineering and Inobservance Whilst Operating with Personal Login Credentials“. In: *Human Aspects of Information Security, Privacy, and Trust - 4th International Conference, HAS 2016, Held as Part of HCI International 2016, Toronto, ON, Canada, July 17-22, 2016, Proceedings*. Hrsg. von Theo Tryfonas. Bd. 9750. Lecture Notes in Computer Science. Springer, 2016, S. 37–47. DOI: 10.1007/978-3-319-39381-0_4 (siehe S. 142–145, 158).

[Hun19] Troy Hunt. *Pwned websites*. 2019. URL: https://haveibeenpwned.com/PwnedWebsites (besucht am 01. 04. 2019) (siehe S. 103).

[HY01] J. T. Gene Hwang und Ming-Chung Yang. „An optimality theory for mid p-values In 2 x 2 contingency tables“. In: *Statistica Sinica* 3 (Juli 2001), S. 807–826. ISSN: 10170405, 19968507. URL: http://www.jstor.org/stable/24306848 (siehe S. 151).

[Ikr+16] Muhammad Ikram, Narseo Vallina-Rodriguez, Suranga Seneviratne, Mohamed Ali Kâafar und Vern Paxson. „An Analysis of the Privacy and Security Risks of Android VPN Permission-enabled Apps“. In: *Proceedings of the 2016 ACM on Internet Measurement Conference, IMC 2016, Santa Monica, CA, USA, November 14-16, 2016*. Hrsg. von Phillipa Gill, John S. Heidemann, John W. Byers und Ramesh Govindan. ACM, 2016, S. 349–364. URL: http://dl.acm.org/citation.cfm?id=2987471 (siehe S. 95).

[ING18] ING-DiBa. *SmartSecure wurde durch Banking to go App abgelöst*. 30. Aug. 2018. URL: https://www.ing-diba.de/ueber-uns/wissenswert/smartsecure-app (besucht am 17. 03. 2019) (siehe S. 31).

[Jag+07] Tom N. Jagatic, Nathaniel A. Johnson, Markus Jakobsson und Filippo Menczer. „Social phishing". In: *Commun. ACM* 10 (2007), S. 94–100. DOI: 10.1145/1290958.1290968 (siehe S. 159).

[Jau19] Henning Jauernig. „Probleme bei Online-Bank N26 häufen sich". In: *Spiegel Online* (28. März 2019). URL: http://spon.de/afrZe (besucht am 01. 04. 2019) (siehe S. 107).

[Kap+14] Alexandros Kapravelos, Chris Grier, Neha Chachra, Christopher Kruegel, Giovanni Vigna und Vern Paxson. „Hulk: Eliciting Malicious Behavior in Browser Extensions". In: *Proceedings of the 23rd USENIX Security Symposium, San Diego, CA, USA, August 20-22, 2014*. Hrsg. von Kevin Fu und Jaeyeon Jung. USENIX Association, 2014, S. 641–654. URL: https://www.usenix.org/conference/usenixsecurity14/technical-sessions/presentation/kapravelos (siehe S. 131, 149).

[Kar15] Ibrahim Karasu. „Innovationen sind für private Banken nichts Neues". In: *Börsen-Zeitung* 120 (27. Juni 2015), B4 (siehe S. 1).

[KBM18] Anatoli Kalysch, Davide Bove und Tilo Müller. „How Android's UI Security is Undermined by Accessibility". In: *Proceedings of the 2Nd Reversing and Offensive-oriented Trends Symposium*. ROOTS'18. Vienna, Austria: ACM, 2018, 2:1–2:10. ISBN: 978-1-4503-6171-2. DOI: 10.1145/3289595.3289597 (siehe S. 56, 66, 84).

[KD18] Abdullah Talha Kabakus und Ibrahim Alper Dogru. „An in-depth analysis of Android malware using hybrid techniques". In: *Digital Investigation* (2018), S. 25–33. ISSN: 1742-2876. DOI: 10.1016/j.diin.2018.01.001 (siehe S. 39).

[Kel+19] Ansgar Kellner, Horlboge Micha, Konrad Rieck und Christian Wressnegger. „False Sense of Security: A Study on the Effectivity of Jailbreak Detection in Banking Apps". In: *2019 IEEE European Symposium on Security and Privacy, EuroS&P 2019, Stockholm, Sweden, June 17-19, 2019*. IEEE, 2019 (siehe S. 83).

[KK14a] Andreas Kurtz und Tobias Klein. „Katz und Maus". In: *c't* 20 (5. Sep. 2014), S. 144–148 (siehe S. 39, 40).

[KK14b] Andreas Kurtz und Tobias Klein. „Schilde runter“. In: *c't* 20 (5. Sep. 2014), S. 142–143 (siehe S. 40).

[KOS19] Andreas Kröner, Yasmin Osman und Katharina Schneider. „Finanzaufsicht rügt N26“. In: *Handelsblatt* 70 (9. Apr. 2019), S. 1 (siehe S. 107).

[Kra17] Eva-Susanne Krah. *Kunden vertrauen Banken mehr als Fintechs*. 4. Sep. 2017. URL: https://www.springerprofessional.de/fintechs/bank-it/kunden-vertrauen-banken-mehr-als-fintechs/14971380 (besucht am 07. 03. 2019) (siehe S. 88).

[Kro+15a] Kat Krol, Eleni Philippou, Emiliano De Cristofaro und Martina Angela Sasse. „They brought in the horrible key ring thing!”Analysing the Usability of Two-Factor Authentication in UK Online Banking“. In: *Proceedings of the NDSS Workshop on Usable Security, USEC 2015, San Diego, California, USA, February 8-11, 2015*. 2015. DOI: 10.14722/usec.2015.23001 (siehe S. 141, 164).

[Kro+15b] Katharina Krombholz, Heidelinde Hobel, Markus Huber und Edgar R. Weippl. „Advanced social engineering attacks“. In: *J. Inf. Sec. Appl.* (2015), S. 113–122. DOI: 10.1016/j.jisa.2014.09.005 (siehe S. 67).

[Krü+04] Christopher Krügel, William K. Robertson, Fredrik Valeur und Giovanni Vigna. „Static Disassembly of Obfuscated Binaries“. In: *Proceedings of the 13th USENIX Security Symposium, August 9-13, 2004, San Diego, CA, USA*. 2004, S. 255–270 (siehe S. 64).

[KS19] Peter Köhler und Katharina Schneider. „Geld für die Expansion“. In: *Handelsblatt* 7 (10. Jan. 2019) (siehe S. 2).

[Kuk18] Mike Kuketz. *Gesundheits-App Vivy: Datenschutz-Bruchlandung*. 18. Sep. 2018. URL: https://www.kuketz-blog.de/gesundheits-app-vivy-datenschutz-bruchlandung (besucht am 08. 03. 2019) (siehe S. 89).

[Kur+16] Andreas Kurtz, Hugo Gascon, Tobias Becker, Konrad Rieck und Felix C. Freiling. „Fingerprinting Mobile Devices Using Personalized Configurations“. In: *PoPETs* 1 (2016), S. 4–19 (siehe S. 55).

[KVB16] Radhesh Krishnan Konoth, Victor van der Veen und Herbert Bos. „How Anywhere Computing Just Killed Your Phone-Based Two-Factor Authentication“. In: *Financial Cryptography and Data Security - 20th International Conference, FC 2016, Christ Church, Barbados, February 22-26, 2016, Revised Selected Papers*. Hrsg. von Jens Grossklags und Bart

Preneel. Bd. 9603. Lecture Notes in Computer Science. Springer, 2016, S. 405–421. DOI: 10.1007/978-3-662-54970-4_24 (siehe S. 23, 24, 133).

[KVE14] Sven Kiljan, Harald P. E. Vranken und Marko C. J. D. van Eekelen. „What You Enter Is What You Sign: Input Integrity in an Online Banking Environment". In: *2014 Workshop on Socio-Technical Aspects in Security and Trust, STAST 2014, Vienna, Austria, July 18, 2014*. Hrsg. von Giampaolo Bella und Gabriele Lenzini. IEEE Computer Society, 2014, S. 40–47. DOI: 10.1109/STAST.2014.14 (siehe S. 165).

[LFL09] Stian Lydersen, Morten W. Fagerland und Petter Laake. „Recommended tests for association in 2 x 2 tables". In: *Statistics in Medicine* 7 (März 2009), S. 1159–1175. DOI: 10.1002/sim.3531 (siehe S. 151).

[Li+10] Shujun Li, S. Amier Haider Shah, M. Asad Usman Khan, Syed Ali Khayam, Ahmad-Reza Sadeghi und Roland Schmitz. „Breaking e-banking CAPTCHAs". In: *Twenty-Sixth Annual Computer Security Applications Conference, ACSAC 2010, Austin, Texas, USA, 6-10 December 2010*. Hrsg. von Carrie Gates, Michael Franz und John P. McDermott. ACM, 2010, S. 171–180. DOI: 10.1145/1920261.1920288 (siehe S. 22).

[Li+17a] Li Li, Daoyuan Li, Tegawendé F. Bissyandé, Jacques Klein, Yves Le Traon, David Lo und Lorenzo Cavallaro. „Understanding Android App Piggybacking: A Systematic Study of Malicious Code Grafting". In: *IEEE Trans. Information Forensics and Security* 6 (2017), S. 1269–1284. DOI: 10.1109/TIFS.2017.2656460 (siehe S. 67).

[Li+17b] Tongxin Li, Xueqiang Wang, Mingming Zha, Kai Chen, XiaoFeng Wang, Luyi Xing, Xiaolong Bai, Nan Zhang und Xinhui Han. „Unleashing the Walking Dead: Understanding Cross-App Remote Infections on Mobile WebViews". In: *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017, Dallas, TX, USA, October 30 - November 03, 2017*. Hrsg. von Bhavani M. Thuraisingham, David Evans, Tal Malkin und Dongyan Xu. ACM, 2017, S. 829–844. DOI: 10.1145/3133956.3134021 (siehe S. 96).

[Lin+14] Chia-Chi Lin, Hongyang Li, Xiao-yong Zhou und XiaoFeng Wang. „Screenmilker: How to Milk Your Android Screen for Secrets". In: *21st Annual Network and Distributed System Security Symposium, NDSS 2014, San Diego, California, USA, February 23-26, 2014*. The Internet Society, 2014. URL: https://www.ndss-symposium.org/ndss2014/screenmilker-how-milk-your-android-screen-secrets (siehe S. 65).

[Lip13] Gregory Lipinski. „Dreiste Masche“. In: *SparkassenZeitung* 44 (31. Okt. 2013), S. 1. ISSN: 0012-0766 (siehe S. 22).

[Liu+12] Lei Liu, Xinwen Zhang, Guanhua Yan und Songqing Chen. „Chrome Extensions: Threat Analysis and Countermeasures“. In: *19th Annual Network and Distributed System Security Symposium, NDSS 2012, San Diego, California, USA, February 5-8, 2012*. The Internet Society, 2012. URL: https://www.ndss-symposium.org/ndss2012/chrome-extensions-threat-analysis-and-countermeasures (siehe S. 149).

[Liu+17] Fang Liu, Chun Wang, Andres Pico, Danfeng Yao und Gang Wang. „Measuring the Insecurity of Mobile Deep Links of Android“. In: *26th USENIX Security Symposium, USENIX Security 2017, Vancouver, BC, Canada, August 16-18, 2017*. Hrsg. von Engin Kirda und Thomas Ristenpart. USENIX Association, 2017, S. 953–969. URL: https://www.usenix.org/conference/usenixsecurity17/technical-sessions/presentation/liu (siehe S. 96).

[LP98] Peter Landrock und Torben P. Pedersen. „WYSIWYS? - What you see is what you sign?“ In: *Inf. Sec. Techn. Report* 2 (1998), S. 55–61. DOI: 10.1016/S0167-4048(98)80005-8 (siehe S. 18).

[Lya+18] Sanam Ghorbani Lyastani, Michael Schilling, Sascha Fahl, Michael Backes und Sven Bugiel. „Better managed than memorized? Studying the Impact of Managers on Password Strength and Reuse“. In: *27th USENIX Security Symposium, USENIX Security 2018, Baltimore, MD, USA, August 15-17, 2018*. Hrsg. von William Enck und Adrienne Porter Felt. USENIX Association, 2018, S. 203–220. URL: https://www.usenix.org/conference/usenixsecurity18/presentation/lyastani (siehe S. 104).

[Mar17] Bernard Marks. „Geschichte des Online-Banking“. In: *Eichsfelder Tageblatt* 269 (18. Nov. 2017), S. 25 (siehe S. 1).

[MF18] Dominik Maier und Fabian Franzen. „Mehr schlecht als Recht: Grauzone Sicherheitsforschung“. 35th Chaos Communication Congress (35c3). Leipzig. 29. Dez. 2018. URL: https://media.ccc.de/v/35c3-9898-mehr_schlecht_als_recht_grauzone_sicherheitsforschung. Vortrag (siehe S. 80).

[MMP14] Dominik Maier, Tilo Müller und Mykola Protsenko. „Divide-and-Conquer: Why Android Malware Cannot Be Stopped“. In: *Ninth International Conference on Availability, Reliability and Security, ARES 2014,*

Fribourg, Switzerland, September 8-12, 2014. IEEE Computer Society, 2014, S. 30–39 (siehe S. 39, 63, 82).

[MN18] Mehran Mahmoudi und Sarah Nadi. „The Android update problem: an empirical study". In: *Proceedings of the 15th International Conference on Mining Software Repositories, MSR 2018, Gothenburg, Sweden, May 28-29, 2018*. Hrsg. von Andy Zaidman, Yasutaka Kamei und Emily Hill. ACM, 2018, S. 220–230. DOI: 10.1145/3196398.3196434 (siehe S. 38).

[MO07] Mohammad Mannan und Paul C. van Oorschot. „Security and usability: the gap in real-world online banking". In: *Proceedings of the 2007 Workshop on New Security Paradigms, White Mountain Hotel and Resort, New Hampshire, USA - September 18-21, 2007*. Hrsg. von Konstantin Beznosov und Angelos D. Keromytis. ACM, 2007, S. 1–14. DOI: 10.1145/1600176.1600178 (siehe S. 158).

[MO17] Stig Fr. Mjølsnes und Ruxandra F. Olimid. „Easy 4G/LTE IMSI Catchers for Non-Programmers". In: *Computer Network Security - 7th International Conference on Mathematical Methods, Models, and Architectures for Computer Network Security, MMM-ACNS 2017, Warsaw, Poland, August 28-30, 2017, Proceedings*. Hrsg. von Jacek Rak, John Bay, Igor V. Kotenko, Leonard J. Popyack, Victor A. Skormin und Krzysztof Szczypiorski. Bd. 10446. Lecture Notes in Computer Science. Springer, 2017, S. 235–246. DOI: 10.1007/978-3-319-65127-9_19 (siehe S. 24).

[MP15] Ibtisam Mohamed und Dhiren Patel. „Android vs iOS Security: A Comparative Study". In: *2015 12th International Conference on Information Technology - New Generations*. Apr. 2015, S. 725–730. DOI: 10.1109/ITNG.2015.123 (siehe S. 39).

[Mul+13] Collin Mulliner, Ravishankar Borgaonkar, Patrick Stewin und Jean-Pierre Seifert. „SMS-Based One-Time Passwords: Attacks and Defense - (Short Paper)". In: *Detection of Intrusions and Malware, and Vulnerability Assessment - 10th International Conference, DIMVA 2013, Berlin, Germany, July 18-19, 2013. Proceedings*. Hrsg. von Konrad Rieck, Patrick Stewin und Jean-Pierre Seifert. Bd. 7967. Lecture Notes in Computer Science. Springer, 2013, S. 150–159. DOI: 10.1007/978-3-642-39235-1_9 (siehe S. 23).

[Mur+10] Steven J. Murdoch, Saar Drimer, Ross J. Anderson und Mike Bond. „Chip and PIN is Broken". In: *31st IEEE Symposium on Security and Privacy, S&P 2010, 16-19 May 2010, Berleley/Oakland, California, USA*.

IEEE Computer Society, 2010, S. 433–446. DOI: 10.1109/SP.2010.33 (siehe S. 27).

[Mur+16] Steven J. Murdoch, Ingolf Becker, Ruba Abu-Salma, Ross J. Anderson, Nicholas Bohm, Alice Hutchings, M. Angela Sasse und Gianluca Stringhini. „Are Payment Card Contracts Unfair? (Short Paper)“. In: *Financial Cryptography and Data Security - 20th International Conference, FC 2016, Christ Church, Barbados, February 22-26, 2016, Revised Selected Papers*. Hrsg. von Jens Grossklags und Bart Preneel. Bd. 9603. Lecture Notes in Computer Science. Springer, 2016, S. 600–608. DOI: 10.1007/978-3-662-54970-4_35 (siehe S. 158).

[MW04] Ulrike Meyer und Susanne Wetzel. „A man-in-the-middle attack on UMTS“. In: *Proceedings of the 2004 ACM Workshop on Wireless Security, Philadelphia, PA, USA, October 1, 2004*. Hrsg. von Markus Jakobsson und Adrian Perrig. ACM, 2004, S. 90–97. DOI: 10.1145/1023646.1023662 (siehe S. 24).

[N2617] N26 GmbH. *Password For Your N26 Account*. 24. Aug. 2017. URL: https://support.n26.com/read/000001288?locale=en (siehe S. 91).

[NB16] norisbank GmbH. *norisbank-Umfrage zum Thema Online-Banking*. 17. Nov. 2016. URL: https://www.norisbank.de/ueberuns/presseinformation-norisbank-umfrage-online-banking-ein-viertel-der-deutschen-nutzt-veraltetes-tan-verfahren.html (besucht am 13. 12. 2017) (siehe S. 120).

[Nes19] Frank Nestler. „Der Internetbanker“. In: *Frankfurter Allgemeine Zeitung* 12 (15. Jan. 2019), S. 19 (siehe S. 87).

[Ngu+17] Long Nguyen-Vu, Ngoc-Tu Chau, Seongeun Kang und Souhwan Jung. „Android Rooting: An Arms Race between Evasion and Detection“. In: *Security and Communication Networks* (2017), 4121765:1–4121765:13. DOI: 10.1155/2017/4121765 (siehe S. 83).

[Obe10] Jon Oberheide. *Android Hax*. SummerCon 2010. Juni 2010. URL: https://jon.oberheide.org/files/summercon10-androidhax-jonoberheide.pdf. Vortrag (siehe S. 38).

[Olt+15] Marten Oltrogge, Yasemin Acar, Sergej Dechand, Matthew Smith und Sascha Fahl. „To Pin or Not to Pin-Helping App Developers Bullet Proof Their TLS Connections“. In: *24th USENIX Security Symposium, USENIX Security 15, Washington, D.C., USA, August 12-14, 2015*. Hrsg. von

Jaeyeon Jung und Thorsten Holz. USENIX Association, 2015, S. 239–254. URL: https://www.usenix.org/conference/usenixsecurity15/technical-sessions/presentation/oltrogge (siehe S. 67).

[Ona+12] Kaan Onarlioglu, Utku Ozan Yilmaz, Engin Kirda und Davide Balzarotti. „Insights into User Behavior in Dealing with Internet Attacks". In: *19th Annual Network and Distributed System Security Symposium, NDSS 2012, San Diego, California, USA, February 5-8, 2012*. The Internet Society, 2012. URL: https://www.ndss-symposium.org/ndss2012/insights-user-behavior-dealing-internet-attacks (siehe S. 159).

[OSM19] Tim Ohlendorf, Wolfgang Studier und Marian Margraf. „Digitale Identitäten auf dem Smartphone". In: *Datenschutz und Datensicherheit* 1 (2019), S. 17–22. DOI: 10.1007/s11623-019-1054-1 (siehe S. 55).

[Pet+14] Thanasis Petsas, Giannis Voyatzis, Elias Athanasopoulos, Michalis Polychronakis und Sotiris Ioannidis. „Rage against the virtual machine: hindering dynamic analysis of Android malware". In: *Proceedings of the Seventh European Workshop on System Security, EuroSec 2014, April 13, 2014, Amsterdam, The Netherlands*. Hrsg. von Davide Balzarotti und Juan Caballero. ACM, 2014, 5:1–5:6 (siehe S. 63).

[PKM15] Mykola Protsenko, Sebastien Kreuter und Tilo Müller. „Dynamic Self-Protection and Tamperproofing for Android Apps Using Native Code". In: *10th International Conference on Availability, Reliability and Security, ARES 2015, Toulouse, France, August 24-27, 2015*. 2015, S. 129–138 (siehe S. 82).

[Poe+14] Sebastian Poeplau, Yanick Fratantonio, Antonio Bianchi, Christopher Kruegel und Giovanni Vigna. „Execute This! Analyzing Unsafe and Malicious Dynamic Code Loading in Android Applications". In: *21st Annual Network and Distributed System Security Symposium, NDSS 2014, San Diego, California, USA, February 23-26, 2014*. The Internet Society, 2014. URL: https://www.ndss-symposium.org/ndss2014/execute-analyzing-unsafe-and-malicious-dynamic-code-loading-android-applications (siehe S. 39).

[Pro] Promon AS. *SHIELD: Application Protection and Security for Mobile Apps*. URL: https://promon.co/products/mobile-app-security (besucht am 23. 02. 2018) (siehe S. 69).

[RCJ14] Heather Rosoff, Jinshu Cui und Richard S. John. „Behavioral Experiments Exploring Victims' Response to Cyber-based Financial Fraud and Identity Theft Scenario Simulations". In: *Tenth Symposium on Usable Privacy and Security, SOUPS 2014, Menlo Park, CA, USA, July 9-11, 2014*. Hrsg. von Lorrie Faith Cranor, Lujo Bauer und Robert Biddle. USENIX Association, 2014, S. 175–186 (siehe S. 158).

[RCL14] Chuangang Ren, Kai Chen und Peng Liu. „Droidmarking: resilient software watermarking for impeding android application repackaging". In: *ACM/IEEE International Conference on Automated Software Engineering, ASE '14, Vasteras, Sweden - September 15 - 19, 2014*. Hrsg. von Ivica Crnkovic, Marsha Chechik und Paul Grünbacher. ACM, 2014, S. 635–646 (siehe S. 61).

[RES18] Sebastian Reinig, Katharina Ebner und Stefan Smolnik. „FinTechs - Eine Analyse des Marktes und seines Bedrohungspotenzials für etablierte Finanzdienstleister". In: *HMD - Praxis Wirtschaftsinform.* 6 (2018), S. 1311–1325 (siehe S. 88).

[Rey+18] Joshua Reynolds, Trevor Smith, Ken Reese, Luke Dickinson, Scott Ruoti und Kent E. Seamons. „A Tale of Two Studies: The Best and Worst of YubiKey Usability". In: *2018 IEEE Symposium on Security and Privacy, SP 2018, Proceedings, 21-23 May 2018, San Francisco, California, USA*. IEEE Computer Society, 2018, S. 872–888. DOI: 10.1109/SP.2018.00067 (siehe S. 141).

[Rie08] Torsten Riecke. „Mit dem Rücken zur Wand". In: *Handelsblatt* 109 (9. Juni 2008) (siehe S. 109).

[RL12] Sampsa Rauti und Ville Leppänen. „Browser extension-based man-in-the-browser attacks against Ajax applications with countermeasures". In: *2012 Conference on Computer Systems and Technologies, CompSysTech'12, Ruse, Bulgaria, June 22-23, 2012*. Hrsg. von Boris Rachev und Angel Smrikarov. ACM, 2012, S. 251–258. DOI: 10.1145/2383276.2383314 (siehe S. 149).

[RTP05] RedTeam Pentesting. *Advisory: New banking security system iTAN not as secure as claimed*. 25. Aug. 2005. URL: https://www.redteam-pentesting.de/advisories/rt-sa-2005-014.txt (besucht am 16. 03. 2019) (siehe S. 21).

[RTP09] RedTeam Pentesting. *Man-in-the-Middle-Angriffe auf das chipTAN comfort-Verfahren im Online-Banking*. 23. Nov. 2009. URL: https://www.redteam-pentesting.de/publications/2009-11-23-MitM-chipTAN-comfort_RedTeam-Pentesting.pdf (besucht am 16. 03. 2019) (siehe S. 27).

[Rup+19] David Rupprecht, Katharina Kohls, Thorsten Holz und Christina Pöpper. „An Extensive Formal Security Analysis of the OpenID Financial-grade API". In: *2019 IEEE Symposium on Security and Privacy, SP 2019, San Francisco, CA, USA, MAY 20-22, 2019*. IEEE Computer Society, Mai 2019 (siehe S. 173).

[Sah+17] Merve Sahin, Aurélien Francillon, Payas Gupta und Mustaque Ahamad. „SoK: Fraud in Telephony Networks". In: *2017 IEEE European Symposium on Security and Privacy, EuroS&P 2017, Paris, France, April 26-28, 2017*. IEEE, 2017, S. 235–250. DOI: 10.1109/EuroSP.2017.40 (siehe S. 124).

[SCB15] San-Tsai Sun, Andrea Cuadros und Konstantin Beznosov. „Android Rooting: Methods, Detection, and Evasion". In: *Proceedings of the 5th Annual ACM CCS Workshop on Security and Privacy in Smartphones and Mobile Devices, SPSM 2015, Denver, Colorado, USA, October 12, 2015*. Hrsg. von David Lie und Glenn Wurster. ACM, 2015, S. 3–14. DOI: 10.1145/2808117.2808126 (siehe S. 40).

[Sch+07] Stuart E. Schechter, Rachna Dhamija, Andy Ozment und Ian Fischer. „The Emperor's New Security Indicators". In: *2007 IEEE Symposium on Security and Privacy (S&P 2007), 20-23 May 2007, Oakland, California, USA*. IEEE Computer Society, 2007, S. 51–65. DOI: 10.1109/SP.2007.35 (siehe S. 141, 158, 165, 166).

[Sch+16] Sebastian Schrittwieser, Stefan Katzenbeisser, Johannes Kinder, Georg Merzdovnik und Edgar R. Weippl. „Protecting Software through Obfuscation: Can It Keep Pace with Progress in Code Analysis?" In: *ACM Comput. Surv.* 1 (2016), 4:1–4:37 (siehe S. 64).

[Sch05] Bruce Schneier. „Two-factor authentication: too little, too late". In: *Commun. ACM* 4 (2005), S. 136. DOI: 10.1145/1053291.1053327 (siehe S. 35).

[Sch15] Marco Schöning. „Boom beim Mobile Banking überrascht nicht". In: *Börsen-Zeitung* 79 (25. Apr. 2015), B3–B4 (siehe S. 30).

[Sch16] Bruce Schneier. „Stop Trying to Fix the User“. In: *IEEE Security & Privacy* 5 (2016), S. 96. DOI: 10.1109/MSP.2016.101 (siehe S. 137).

[Sch18] Meike Schreiber. „Spät aufgewacht“. In: *Süddeutsche Zeitung* 224 (28. Sep. 2018), S. 20 (siehe S. 2).

[Sch19a] Caspar Tobias Schlenk. „Einem N26-Kunden werden 80.000 Euro gestohlen – und die Bank ist überfordert“. In: *Gründerszene* (28. März 2019). URL: https://www.gruenderszene.de/fintech/n26-axel-seitz-phishing (besucht am 01. 04. 2019) (siehe S. 107).

[Sch19b] Katharina Schneider. „Bafin warnt vor Betrugsmasche“. In: *Handelsblatt* 63 (29. März 2019), S. 37 (siehe S. 107).

[Sch19c] Katharina Schneider. „Eine Revolution im Verborgenen“. In: *Handelsblatt* 10 (15. Jan. 2019), S. 30 (siehe S. 172).

[SE19] Patrick Stähler und Jan Evers. „Eine Bank, die cool sein will“. In: *brand eins* 4 (2019) (siehe S. 87).

[Sei08] Karsten Seibel. „Das Handy macht Onlinebanking sicherer“. In: *Die Welt* 126 (31. Mai 2008), S. 17 (siehe S. 23).

[Sei19] Karsten Seibel. „Das Buch Tan“. In: *Welt am Sonntag* 8 (24. Feb. 2019), S. 39 (siehe S. 22, 48, 123).

[Sel18] Karsten Meyer zu Selhausen. „Security of PDF Signatures“. Magisterarb. Ruhr-Universität Bochum, 5. Nov. 2018. URL: https://www.pdf-insecurity.org/download/DIGITALVERSION_KMeyerZuSelhausen_SecurityOfPDFSignatures_2018-11-25.pdf (besucht am 17. 04. 2019) (siehe S. 131).

[Sha+14] Hossain Shahriar, Komminist Weldemariam, Mohammad Zulkernine und Thibaud Lutellier. „Effective detection of vulnerable and malicious browser extensions“. In: *Computers & Security* (2014), S. 66–84. DOI: 10.1016/j.cose.2014.06.005 (siehe S. 149).

[Sha+16] Altaf Shaik, Jean-Pierre Seifert, Ravishankar Borgaonkar, N. Asokan und Valtteri Niemi. „Practical Attacks Against Privacy and Availability in 4G/LTE Mobile Communication Systems“. In: *23rd Annual Network and Distributed System Security Symposium, NDSS 2016, San Diego, California, USA, February 21-24, 2016*. The Internet Society, 2016. URL: http://wp.internetsociety.org/ndss/wp-content/uploads/sites/25/2017/09/practical-attacks-against-privacy-availability-4g-lte-mobile-communication-systems.pdf (siehe S. 24).

[She+10] Steve Sheng, Mandy B. Holbrook, Ponnurangam Kumaraguru, Lorrie Faith Cranor und Julie S. Downs. „Who falls for phish?: a demographic analysis of phishing susceptibility and effectiveness of interventions". In: *Proceedings of the 28th International Conference on Human Factors in Computing Systems, CHI 2010, Atlanta, Georgia, USA, April 10-15, 2010*. Hrsg. von Elizabeth D. Mynatt, Don Schoner, Geraldine Fitzpatrick, Scott E. Hudson, W. Keith Edwards und Tom Rodden. ACM, 2010, S. 373–382. DOI: 10.1145/1753326.1753383 (siehe S. 104, 159).

[SJT08] Karen Scarfone, Wayne Jansen und Miles Tracy. „Guide to General Server Security". In: *Recommendations of the National Institute of Standards and Technology* (Juli 2008). DOI: 10.6028/NIST.SP.800-123 (siehe S. 59).

[Slo14] Katharina Slodczyk. „Mit Albert Einsteins Hilfe". In: *Handelsblatt* 63 (31. März 2014), S. 32 (siehe S. 88).

[Spe19] Michael Spehr. „Android hat fertig". In: *Frankfurter Allgemeine Zeitung* 54 (5. März 2019), T4 (siehe S. 39).

[Spi79] „Stählerne Kassierer". In: *Der Spiegel* 11 (12. März 1979), S. 93 (siehe S. 1).

[ST16] Mohamed Sabt und Jacques Traoré. „Breaking into the KeyStore: A Practical Forgery Attack Against Android KeyStore". In: *Computer Security - ESORICS 2016 - 21st European Symposium on Research in Computer Security, Heraklion, Greece, September 26-30, 2016, Proceedings, Part II*. Hrsg. von Ioannis G. Askoxylakis, Sotiris Ioannidis, Sokratis K. Katsikas und Catherine A. Meadows. Bd. 9879. Lecture Notes in Computer Science. Springer, 2016, S. 531–548. DOI: 10.1007/978-3-319-45741-3_27 (siehe S. 56).

[Ste19] Lukas Stefanko. *First clipper malware discovered on Google Play*. 8. Feb. 2019. URL: https://www.welivesecurity.com/2019/02/08/first-clipper-malware-google-play (besucht am 16. 04. 2019) (siehe S. 130).

[Sto05] Matthias Stoffel. „Phishing-Abwehr: Auf dem Weg zur indizierten TAN". In: *SparkassenZeitung* 38 (23. Sep. 2005), S. 15. ISSN: 0012-0766 (siehe S. 21).

[Sun+15] He Sun, Kun Sun, Yuewu Wang und Jiwu Jing. „TrustOTP: Transforming Smartphones into Secure One-Time Password Tokens“. In: *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, Denver, CO, USA, October 12-6, 2015*. 2015, S. 976–988. DOI: 10.1145/2810103.2813692 (siehe S. 56).

[SW08] Amitabh Saxena und Brecht Wyseur. „On White-box Cryptography and Obfuscation“. In: *CoRR* (2008). arXiv: 0805.4648. URL: http://arxiv.org/abs/0805.4648 (siehe S. 64).

[Tan17] Hakan Tanriverdi. „Überweisung vom Hacker“. In: *Süddeutsche Zeitung* 270 (24. Nov. 2017) (siehe S. 67, 79).

[TBR15] Daniel R. Thomas, Alastair R. Beresford und Andrew C. Rice. „Security Metrics for the Android Ecosystem“. In: *Proceedings of the 5th Annual ACM CCS Workshop on Security and Privacy in Smartphones and Mobile Devices, SPSM 2015, Denver, Colorado, USA, October 12, 2015*. Hrsg. von David Lie und Glenn Wurster. ACM, 2015, S. 87–98. DOI: 10.1145/2808117.2808118 (siehe S. 38).

[Teu+13] Peter Teufl, Thomas Zefferer, Christof Stromberger und Christoph Hechenblaikner. „iOS Encryption Systems - Deploying iOS Devices in Security-critical Environments“. In: *SECRYPT 2013 - Proceedings of the 10th International Conference on Security and Cryptography, Reykjavík, Iceland, 29-31 July, 2013*. Hrsg. von Pierangela Samarati. SciTePress, 2013, S. 170–182. URL: http://ieeexplore.ieee.org/document/7223165/ (siehe S. 56).

[Tsc18] Martin Tschirsich. *All Your Gesundheitsakten Are Belong To Us*. 35th Chaos Communication Congress (35c3): refreshing memories. Chaos Computer Club e.V., 27. Dez. 2018. URL: https://media.ccc.de/v/35c3-9992-all_your_gesundheitsakten_are_belong_to_us. Vortrag (siehe S. 89).

[TZ17] Hakan Tanriverdi und Markus Zydra. „SMS von gestern Nacht“. In: *Süddeutsche Zeitung* 101 (3. Mai 2017) (siehe S. 24).

[VBB] Berliner Volksbank. *Das neue Push-TAN Verfahren - VR-SecureGo*. URL: https://www.berliner-volksbank.de/banking/push-tan.html (besucht am 23. 02. 2018) (siehe S. 59).

[VC14] Timothy Vidas und Nicolas Christin. „Evading android runtime analysis via sandbox detection“. In: *9th ACM Symposium on Information, Computer and Communications Security, ASIA CCS '14, Kyoto, Japan - June 03 - 06, 2014*. Hrsg. von Shiho Moriai, Trent Jaeger und Kouichi Sakurai. ACM, 2014, S. 447–458 (siehe S. 63).

[Vie14] Susanne Vieser. „Banken unter Druck“. In: *Internet World Business* 14 (7. Juli 2014) (siehe S. 2).

[Wan+13] Tielei Wang, Kangjie Lu, Long Lu, Simon P. Chung und Wenke Lee. „Jekyll on iOS: When Benign Apps Become Evil“. In: *Proceedings of the 22th USENIX Security Symposium, Washington, DC, USA, August 14-16, 2013*. Hrsg. von Samuel T. King. USENIX Association, 2013, S. 559–572. URL: https://www.usenix.org/conference/usenixsecurity13/technical-sessions/presentation/wang_tielei (siehe S. 39).

[Wan+14] Tielei Wang, Yeongjin Jang, Yizheng Chen, Simon P. Chung, Billy Lau und Wenke Lee. „On the Feasibility of Large-Scale Infections of iOS Devices“. In: *Proceedings of the 23rd USENIX Security Symposium, San Diego, CA, USA, August 20-22, 2014*. Hrsg. von Kevin Fu und Jaeyeon Jung. USENIX Association, 2014, S. 79–93. URL: https://www.usenix.org/conference/usenixsecurity14/technical-sessions/presentation/wang_tielei (siehe S. 36).

[Wan+16] Ding Wang, Zijian Zhang, Ping Wang, Jeff Yan und Xinyi Huang. „Targeted Online Password Guessing: An Underestimated Threat“. In: *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, October 24-28, 2016*. Hrsg. von Edgar R. Weippl, Stefan Katzenbeisser, Christopher Kruegel, Andrew C. Myers und Shai Halevi. ACM, 2016, S. 1242–1254. DOI: 10.1145/2976749.2978339 (siehe S. 104).

[Wat13] Brigitte Watermann. „Banken rüsten weiter auf“. In: *Börse Online* 23 (29. Mai 2013), S. 54–55 (siehe S. 30).

[Wei+09] Catherine S. Weir, Gary Douglas, Martin Carruthers und Mervyn A. Jack. „User perceptions of security, convenience and usability for ebanking authentication tokens“. In: *Computers & Security* 1-2 (2009), S. 47–62. DOI: 10.1016/j.cose.2008.09.008 (siehe S. 142).

[Wey99] Ulrike Weyse. „Vater von Geldautomat und Homebanking“. In: *Sächsische Zeitung* (10. Juli 1999), S. 9 (siehe S. 1).

[WSM10] Dennis Wehrle, Dirk von Suchodoletz und Konrad Meier. „GSM - Zwischen neuer Freiheit und Massenüberwachung. Gefährdung der fragilen Sicherheit in Mobilfunknetzen durch Ortung und IMSI-Catcher“. In: *Praxis der Informationsverarbeitung und Kommunikation* 3 (2010), S. 227–238. DOI: 10.1515/piko.2010.039 (siehe S. 24).

[Wu+16] Wenjia Wu, Jianan Wu, Yanhao Wang, Zhen Ling und Ming Yang. „Efficient Fingerprinting-Based Android Device Identification With Zero-Permission Identifiers“. In: *IEEE Access* (2016), S. 8073–8083. DOI: 10.1109/ACCESS.2016.2626395 (siehe S. 55).

[WW17] Samuel Weiser und Mario Werner. „SGXIO: Generic Trusted I/O Path for Intel SGX“. In: *Proceedings of the Seventh ACM on Conference on Data and Application Security and Privacy, CODASPY 2017, Scottsdale, AZ, USA, March 22-24, 2017*. 2017, S. 261–268. DOI: 10.1145/3029806.3029822 (siehe S. 112).

[WW19] Jan Willmroth und Nils Wischmeyer. „Komfort über Sicherheit“. In: *Süddeutsche Zeitung* 91 (17. Apr. 2019), S. 19 (siehe S. 107, 150).

[WZ17] Bryan Watson und Jun Zheng. „On the User Awareness of Mobile Security Recommendations“. In: *Proceedings of the 2017 ACM Southeast Regional Conference, Kennesaw, GA, USA, April 13-15, 2017*. ACM, 2017, S. 120–127. DOI: 10.1145/3077286.3077563 (siehe S. 158).

[Xin+15] Xinyu Xing, Wei Meng, Byoungyoung Lee, Udi Weinsberg, Anmol Sheth, Roberto Perdisci und Wenke Lee. „Understanding Malvertising Through Ad-Injecting Browser Extensions“. In: *Proceedings of the 24th International Conference on World Wide Web, WWW 2015, Florence, Italy, May 18-22, 2015*. Hrsg. von Aldo Gangemi, Stefano Leonardi und Alessandro Panconesi. ACM, 2015, S. 1286–1295. DOI: 10.1145/2736277.2741630 (siehe S. 149).

[Xue+17] Lei Xue, Xiapu Luo, Le Yu, Shuai Wang und Dinghao Wu. „Adaptive Unpacking of Android Apps“. In: *2017 IEEE/ACM 39th International Conference on Software Engineering (ICSE)*. 2017, S. 358–369 (siehe S. 64).

[Yan+15] Wenbo Yang, Yuanyuan Zhang, Juanru Li, Junliang Shu, Bodong Li, Wenjun Hu und Dawu Gu. „AppSpear: Bytecode Decrypting and DEX Reassembling for Packed Android Malware“. In: *Research in Attacks, Intrusions, and Defenses - 18th International Symposium, RAID 2015,*

Kyoto, Japan, November 2-4, 2015, Proceedings. 2015, S. 359–381 (siehe S. 64, 83).

[Yin+18] Kailiang Ying, Amit Ahlawat, Bilal Alsharifi, Yuexin Jiang, Priyank Thavai und Wenliang Du. „TruZ-Droid: Integrating TrustZone with Mobile Operating System“. In: *Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services, MobiSys 2018, Munich, Germany, June 10-15, 2018*. Hrsg. von Jörg Ott, Falko Dressler, Stefan Saroiu und Prabal Dutta. ACM, 2018, S. 14–27. DOI: 10.1145/3210240.3210338 (siehe S. 56).

[ZD14] Xiao Zhang und Wenliang Du. „Attacks on Android Clipboard“. In: *Detection of Intrusions and Malware, and Vulnerability Assessment - 11th International Conference, DIMVA 2014, Egham, UK, July 10-11, 2014. Proceedings*. 2014, S. 72–91. DOI: 10.1007/978-3-319-08509-8_5 (siehe S. 130).

[ZGK11] „Postbank - Neues Sicherheitsverfahren“. In: *Zeitschrift für das gesamte Kreditwesen* 2 (2. Mai 2011), S. 30. ISSN: 0341-4019 (siehe S. 22).

[Zhe+16] Xianyi Zheng, Lulu Yang, Jiangang Ma, Gang Shi und Dan Meng. „TrustPAY: Trusted mobile payment on security enhanced ARM TrustZone platforms“. In: *IEEE Symposium on Computers and Communication, ISCC 2016, Messina, Italy, June 27-30, 2016*. 2016, S. 456–462. DOI: 10.1109/ISCC.2016.7543781 (siehe S. 56).

[Zho+12] Zongwei Zhou, Virgil D. Gligor, James Newsome und Jonathan M. McCune. „Building Verifiable Trusted Path on Commodity x86 Computers“. In: *IEEE Symposium on Security and Privacy, SP 2012, 21-23 May 2012, San Francisco, California, USA*. 2012, S. 616–630. DOI: 10.1109/SP.2012.42 (siehe S. 112).

[ZKA08] Zentraler Kreditausschauss. *Mindestsicherheitsanforderungen an die mobile TAN*. 14. Apr. 2008. URL: https://die-dk.de/media/files/Mindestsicherheitsanforderungen_mobileTAN_V1_20110621.pdf (siehe S. 32).

[ZLY15] Yueqian Zhang, Xiapu Luo und Haoyang Yin. „DexHunter: Toward Extracting Hidden Code from Packed Android Applications“. In: *Computer Security - ESORICS 2015 - 20th European Symposium on Research in Computer Security, Vienna, Austria, September 21-25, 2015, Proceedings, Part II*. 2015, S. 293–311 (siehe S. 64, 83).

[Zom+08] Mohammed Al Zomai, Bander AlFayyadh, Audun Jøsang und Adrian McCullagh. „An Experimental Investigation of the Usability of Transaction Authorization in Online Bank Security Systems“. In: *Sixth Australasian Information Security Conference, AISC 2008, Wollongong, NSW, Australia, January 2008*. Hrsg. von Ljiljana Brankovic und Mirka Miller. Bd. 81. CRPIT. Australian Computer Society, 2008, S. 65–73. URL: http://crpit.com/abstracts/CRPITV81AlZomai.html (siehe S. 142–145).