Kontextbasierte Sicherheitsmaßnahmen für mobile Geräte in nicht vertrauenswürdigen Netzwerken

Von der Fakultät für Elektrotechnik und Informatik der
Gottfried Wilhelm Leibniz Universität Hannover
zur Erlangung des akademischen Grades eines

Doktor der Naturwissenschaften
(Dr. rer. nat.)

genehmigte Dissertation

von
M. Sc. Christian Szongott
geboren am 14. Juni 1981 in Hildesheim

2015

Referent: Prof. Dr. rer. nat. Matthew Smith
Koreferent: Prof. Dr.-Ing. Gabriele von Voigt

Tag der Promotion: 09.07.2015

Vorwort

Die vorliegende Arbeit entstand während meiner Tätigkeit in der Einrichtung Leibniz Universität IT Services (ehemals Regionales Rechenzentrum für Niedersachsen) und am Forschungszentrum L3S in Hannover.

Ich bedanke mich besonders bei Herrn Prof. Dr. Matthew Smith für seine Unterstützung bei der Erstellung dieser Arbeit und das entgegengebrachte Vertrauen. Ebenso danke ich Frau Prof. Dr. Gabriele von Voigt für ihre Unterstützung und dass sie immer ein offenes Ohr für mich hatte.

Auch meinen Kollegen der Distributed Computing & Security Group möchte ich für die vertrauensvolle Arbeitsatmosphäre und die zahlreichen, fruchtbaren Diskussionen danken. Weiterhin danke ich den Studierenden, die in Form von Projekten und Abschlussarbeiten einen wichtigen Teil zu dieser Arbeit beigetragen haben.

Bedanken möchte ich mich bei meiner Familie für den steten Zuspruch und bei meiner Frau Eileen für ihre Geduld und dafür, dass Sie mir in dieser Zeit stets den Rücken frei gehalten hat.

Hildesheim, im Juli 2015

Zusammenfassung

Die vorliegende Dissertation präsentiert neben einer Untersuchung von Gefahren durch sog. Evil Twin Access Points ein autarkes, mobiles Erkennungssystem, um Benutzer vor derartigen Gefahren zu beschützen.

Wurden in der Anfangszeit der mobilen Internetnutzung nur sehr kleine Datenmengen verschickt, so benötigen heutige Internetdienste und mobile Anwendungen nicht nur eine große Bandbreite, sondern auch ein zunehmend größeres Datenvolumen zur Erbringung ihrer Dienste. Auch die hohen Kosten für mobiles Datenvolumen sorgen dafür, dass die Nutzer derartiger Dienste öffentliche Hotspots in Anspruch nehmen. Während die Gefahren im Verlauf der Nutzung öffentlicher Hotspot hinlänglich bekannt und erforscht sind, ergeben sich durch Standardeinstellungen aktueller mobiler Betriebssysteme und durch die fehlende Authentifizierung von Access Points weitere Gefahren, die auch nach der Nutzung eines solchen Hotspots fortbestehen. 78 % aller Verbindungen zu Access Points werden ohne das Wissen des Benutzers aufgebaut und lassen somit keine Beurteilung der Verbindung zu.

In dieser Dissertation wird dargestellt, durch welche Mechanismen diese Gefahren entstehen und wie sich diese Unzulänglichkeiten durch Angreifer für die Verbreitung von Malware ausnutzen lassen. Ebenso wird ein autarkes, kontextbasiertes System zur Erkennung von Evil Twins Access Points vorgestellt, mit Hilfe dessen die o.g. Gefahren minimiert werden können. Ziel des Systems ist die Erkennung unbekannter Kontexte ausschließlich anhand von Parametern, die durch das mobile Gerät selbst ermittelt werden. Benutzer werden so in die Lage versetzt, der aktuellen Situation entsprechend zu reagieren und so den Verbindungsaufbau zu kontrollieren.

Zur Untersuchung des Gefahrenpotentials heutiger Infrastrukturen und zur Identifizierung geeigneter Kontextparameter für ein Erkennungssystem wurde eine Feldstudie durchgeführt. Die Ergebnisse haben sowohl zur Entwicklung und Konfiguration, als auch zur Evaluation des Erkennungssystems mit Echtweltdaten beigetragen. Das Erkennungssystem wurde in einem zweiten Schritt zur Durchführung weiterer Evaluationen und zur Demonstration einer möglichen Integration in die Benutzeroberfläche auf die Android-Plattform portiert.

Schlagwörter: Evil Twin, Hotspot, Kontextbasierte Erkennung, Mobile Malware, Ausbreitung

Abstract

This thesis presents an analysis of threats stemming from evil twin access points and proposes a self-contained, mobile detection system to guard users against them.

While in the beginning of mobile Internet usage only small amounts of data have been transmitted, todays Internet-based services and mobile applications do not only require higher bandwidths but also larger data volumes. High costs for mobile data volume cause users to connect to public Wi-Fi access points as well. Although the threats of these connections are well-known and adequately studied, new threats arise from default configurations of mobile operating systems and the lack of access point authentication which remain, even if there is no connection to the specific access point. 78 % of all connections to WiFi access points are being established without the users even noticing it, making it impossible for them to evaluate these connections.

This thesis describes which vulnerabilities lead to these threats and how attackers can exploit them in order to spread mobile malware. A context-based, self-contained detection system is proposed to minimize the risk of such attacks. It recognizes unknown contexts solely through parameters that are gathered by the device during the connection process. The system enables users to react appropriately and thereby control the connection establishment.

A field study has been conducted to analyze the risk potential in todays infrastructures and to find suitable context parameters for the detection system. The acquired data was used to develop and configure the detection system but also to evaluate the system on a real-world data basis. The system was ported to the Android platform to conduct further evaluations and to demonstrate the integration of the recognition system into the user interface.

Keywords: evil twin, hotspot, context-based recognition, mobile malware, spreading

Erklärung

Ich versichere, dass ich meine Dissertation

> Kontextbasierte Sicherheitsmaßnahmen für mobile Geräte in nicht vertrauenswürdigen Netzwerken

selbständig, ohne unerlaubte Hilfe angefertigt und mich dabei keiner anderen als der von mir ausdrücklich bezeichneten Quellen und Hilfen bedient habe. Die Dissertation wurde in der jetzigen oder einer ähnlichen Form noch bei keiner anderen Hochschule eingereicht und hat noch keinen sonstigen Prüfungszwecken gedient.

Christian Szongott
Hildesheim im April 2015

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Abkürzungen

AAA	Authentication / Authorization / Accounting
AES	Advanced Encryption Standard
AP	Access Point
API	Application Programming Interface
BSSID	Basic Service Set Identifier
CCMP	Counter Mode mit Cipher Block Chaining Message Authentication Code Protocol
CDF	Cumulative Distribution Function
DFU	Device Firmware Upgrade
DSL	Digital Subscriber Line
ESSID	Extended Service Set Identifier
FQDN	Fully Qualified Domain Name
IAT	Interpacket Arrival Time
LLB	Low Level Bootloader
LTE	Long Term Evolution
MANET	Mobile ad hoc Network
MMS	Multimedia Messaging Service
NFC	Near Field Communication
PKI	Public Key Infrastructure
POI	Point of Interest
RSN	Robust Secure Network
SSID	Service Set Identifier
TKIP	Temporary Key Integrity Protocol
TOFU	Trust On First Use
UAM	Universal Access Method
VPN	Virtual Private Network

WEP	Wired Equivalent Privacy
WISP	Wireless Internet Service Provider
WLAN	Wireless Local Area Network
WPA	Wi-Fi Protected Access

Kapitel 1

Einführung

Eine kurze Einführung in die Verbreitung und Ausgestaltung von Hotspot-Umgebungen und deren Nutzung wird im ersten Kapitel vorgestellt. Es werden ebenso Gefahren dargestellt, die sich aus der Nutzung solcher Infrastrukturen ergeben und wie sich hieraus die Motivation für diese Arbeit ableitet. Auch der wissenschaftliche Beitrag und eine Einordnung dieser Arbeit in das wissenschaftliche Umfeld sind Bestandteil der Einführung.

1.1 Einleitung und Motivation

Die Nutzung von internetbasierten Diensten ist aus dem heutigen Alltag nicht mehr wegzudenken. Nicht nur im Beruf werden sie für die Kommunikation und zur Verarbeitung verschiedenster Aufgaben eingesetzt. Auch im privaten Umfeld hat das Internet einen hohen Stellenwert eingenommen. Mehr als 81 % der Deutschen nutzen das Internet in ihrem Alltag. In der Altersgruppe bis 45 Jahre ist sogar eine Internetnutzung von mehr als 98 % zu beobachten. Betrachtet man die vergangenen Jahre so ist eine stetige Steigerung dieser Zahlen zu erkennen [60]. Früher war die Internetnutzung nur mit Hilfe von feststehenden Computern und später mit Laptops möglich. Die Art und der Ort der Nutzung haben sich in den letzten Jahren jedoch stark in Richtung einer mobilen Nutzung von Onlinediensten entwickelt [30]. Hierzu zählen längst nicht mehr nur die bereits durch Feature Phones bereitgestellten Möglichkeiten, wie das Surfen im Netz und die Kommunikation per Email. Durch mobile Apps[1] auf Smartphones und Tablets können verschiedenste internetbasierte Dienste auch mobil genutzt werden. Erst durch die Einführung von Smartphones mit ihren Touchscreens und durch die beiden nahezu zeitgleich eingeführten und zu ihrer Zeit revolutionären Betriebssysteme Android und iOS (damals noch iPhoneOS) hat sich die mobile Nutzung dieser Dienste in der breiten Bevölkerung etabliert.

Auch die hierfür erforderlichen Mobilfunknetze wurden durch die Mobilfunkanbieter sukzessive ausgebaut und erreichen heute durch LTE-Technik Übertragungsgeschwindigkeiten, die mit einem DSL-Festnetzanschluss konkurrieren können. Für die Mobilfunkanbieter steht neben der Finanzierung des weiteren Netzausbaus auch die Maximierung des operativen Gewinns im Vordergrund. Unter anderem aus diesen Gründen werden Datentarife derzeit zu verhältnismäßig hohen Preisen angeboten, vergleicht man sie mit Angeboten für herkömmliche Sprachtelefonie. Datentarife sind hierbei volumenbasiert. Die als Flatrate beworbenen Tarife sind nach Verbrauch des ausgewählten Kontingents auf eine sehr niedrige Bandbreite gedrosselt[2] . Bei aktivierter Drosselung können nur noch elementare Aufgaben, die eine Internetverbindung vorausset-

[1] Zusätzlich installierbare Programme für mobile Geräte wie Smartphones und Tablets

[2] In den meisten Fällen findet eine Drosselung auf 64 kbit/s statt

zen, durchgeführt werden. Dienste mit höheren Ansprüchen an die Bandbreite sind dann über das Mobilfunknetz nicht mehr möglich.
Da in der Vergangenheit und auch heute noch diese hohen Preise für die mobile Datennutzung erhoben wurden und werden, und weil davon auszugehen ist, dass sich an der angebotenen volumenbegrenzten Charakteristik der Datentarife in naher Zukunft nichts ändern wird, versuchen Endnutzer durch geeignete Maßnahmen die Nutzung des mobilen Datenverkehrs zu minimieren. Aus diesem Grund werden häufig zur Verfügung stehende öffentliche Hotspot-Infrastrukturen genutzt. Diese werden in den meisten Fällen von spezialisierten Hotspotanbietern und Mobilfunkanbietern betrieben und stellen über ein Funknetzwerk den Zugang zum Internet zur Verfügung. In vielen Fällen ist dieser Zugang kostenlos und kann über ein unverschlüsseltes WLAN direkt durch die Anwender in Anspruch genommen werden. In den folgenden Kapiteln werden die Begriffe WLAN und Funknetzwerk gleichbedeutend verwendet.
Bei der Nutzung dieser Hotspots besteht in vielen Fällen ein Sicherheitsrisiko, da durch die unverschlüsselte Bereitstellung des Netzwerks auch die Nutzdaten unverschlüsselt über das geteilte Medium Luft übertragen werden. Entsprechend hat ein Angreifer die Möglichkeit, diesen Datenverkehr mitzuschneiden. Diese triviale Art des Angriffs wurde sowohl im privaten als auch im wirtschaftlichen Umfeld durch die Einführung von Verschlüsselungsstandards für WLANs unterbunden. Öffentliche Hotspots basieren hingegen weiterhin auf offenen und somit unverschlüsselten Netzwerken und stellen somit weiterhin ein Sicherheitsrisiko dar. Der Grund für dieses Vorgehen der Anbieter liegt zum einen darin begründet, dass die Benutzbarkeit für den Endkunden eine sehr hohe Priorität beim Betrieb dieser Hotspot-Infrastrukturen hat. Die Betreiber vermeiden es daher, die Hotspots mit einer Verschlüsselung zu versehen und können auf diese Weise auf einen initialen Schlüsselaustausch – in welcher Form auch immer – verzichten. Es gibt bis heute keine praktikablen und durch Anbieter großflächig eingesetzten Ansätze und Maßnahmen, um dieses Problem zu lösen. Auch der Grundschutzkatalog des BSI beschreibt in Maßnahme M 4.293 (Sicherer Betrieb von Hotspots) [5] diese Situation, sieht aber keine zwingenden Maßnahmen zur Steigerung der Sicherheit vor:

> „ Jeder Betreiber eines Hotspots sollte mindestens ein geeignetes Verfahren zur Verschlüsselung der Funkstrecke anbieten, damit

> die Benutzer ihre Daten vor unbefugtem Mitlesen schützen können. Nicht alle Benutzer haben allerdings ein ausgeprägtes Interesse am Schutz ihrer Daten und Systeme. Es können auch die technischen Voraussetzungen für die Nutzung von angebotenen Verschlüsselungsverfahren fehlen. Daher sollte deren Nutzung optional sein. Die Benutzer sollten aber unbedingt auf die Möglichkeit und die Vorteile der verschlüsselten Übertragung hingewiesen werden. “

Aber nicht nur der triviale Angriff auf den unverschlüsselten Datenverkehr zwischen mobilen Geräten und den entsprechenden Access Points[3] stellt ein Sicherheitsproblem dar. Durch die Verwendung unverschlüsselter Netzwerke findet während der Verbindung zu einem solchen Netzwerk keinerlei Authentifizierung des Benutzers statt. Stattdessen findet die Authentifizierung und Autorisierung in den meisten Fällen erst in einem nachgelagerten Schritt mit Hilfe sog. Captive Portals statt. Das im Rahmen dieser Dissertation betrachtete Sicherheitsrisiko für den Endbenutzer ergibt sich hingegen aus der ebenfalls fehlenden Authentifizierung des Access Points gegenüber dem Gerät des Benutzers. Die hieraus resultierenden Probleme werden im Rahmen dieser Arbeit beschrieben und analysiert. Eine der größten Gefahren, die sich aus den beschriebenen Unzulänglichkeiten ergibt, ist die Möglichkeit eines Angreifers, einen sog. Evil Twin (bösartigen Zwilling) einzusetzen. Bei einem Evil Twin Angriff versucht der Angreifer die mobilen Endgeräte der Benutzer dazu zu bringen, sich mit einem von ihm betriebenen Access Point zu verbinden. Der Evil Twin täuscht hierbei einen für den Benutzer vertrauenswürdigen Access Point vor. Ein solcher Evil Twin kann durch den Angreifer an einem beliebigen Ort installiert und eingesetzt werden, da aktuelle Betriebssysteme den Ort eines Access Points weder speichern noch beim Verbindungsaufbau verifizieren. Ist die Verbindung zu einem Evil Twin Access Point erst einmal aufgebaut, so hat der Angreifer die Möglichkeit den Datenverkehr zwischen einem Benutzer und einem beliebigen Server im Internet zu belauschen, zu überwachen oder gar zu verändern. Dass diese Gefahr real ist, zeigt auch ein Beispiel aus dem Jahr 2013, bei dem EU-Politiker in Straßburg Opfer eines solchen Angriffs wurden [19]. Als erste Konsequenz hat das EU-Parlament seinerzeit den Betrieb des öffentlichen WLANs eingestellt [14].

[3] Zugangspunkt zu einem Netzwerk, beispielsweise ein WLAN-Router

Im Rahmen dieser Arbeit werden Untersuchungen dazu angestellt, wie groß die Gefahr durch Evil Twins im alltäglichen Leben eines Smartphone-Benutzers ist. Um die reale Gefahr in echten Infrastrukturen untersuchen zu können, müssen im Vorfeld Echtweltdaten gesammelt und aufbereitet werden. Neben der Analyse der Daten liegt ein weiteres Hauptaugenmerk dieser Arbeit auf der Erforschung und Entwicklung eines unabhängigen Erkennungssystems von Evil Twins, mit dem sich Benutzer vor dieser Art von Angriffen schützen können. Ein solches System ist bislang in keinem aktuellen Betriebssystem enthalten. Aus diesem Grund sollen Mechanismen und Lösungsansätze gefunden und erforscht werden, die den Benutzer auf möglichst intuitive Art und Weise vor der Verbindung mit den oben genannten Evil Twin Access Points bewahren. Bei der Entwicklung gilt es Anforderungen verschiedenster Bereiche zu berücksichtigen. Wichtigste Aufgabe eines solchen Systems ist die Steigerung der Sicherheit des Benutzers. Darüber hinaus müssen bei der Planung gegebene Rahmenbedingungen, wie beispielsweise eine begrenzte Sensorik, beachtet werden. Um das System unabhängig im Alltag einzusetzen muss es auf den mobilen Geräten selbst lauffähig sein, was weitere Anforderungen hinsichtlich der Energieeffizienz mit sich bringt. Eine weitere essentielle Anforderung ist die Benutzbarkeit des Systems. Nur so kann die Akzeptanz für ein Sicherheitssystem auf Seiten der Endanwender erreicht und die Sicherheit für jeden einzelnen gesteigert werden.
Da bislang keine Systeme zum Schutz vor derartigen Angriffen existieren, die mit Hilfe von Echtweltdaten evaluiert worden sind ist eine entsprechende Evaluation ebenfalls Bestandteil dieser Dissertation. Hierbei gilt es verschiedene Arten der Evaluation hinsichtlich der Sicherheit und der Privatsphäre der teilnehmenden Benutzer zu bewerten und schlussendlich ein geeignetes Verfahren für das Erkennungssystem auszuwählen.

1.2 Wissenschaftlicher Beitrag

In dieser Dissertation werden Angriffe auf Daten von Benutzern mobiler, internetfähiger Geräte durch Evil Twins erforscht. Es wird ein neuartiger Verbreitungsmechanismus vorgestellt und gezeigt, dass eine zugehörige Ausbreitung in heutigen Infrastrukturen epidemische Ausmaße annehmen kann. Durch die

Sammlung von Verbindungsdaten realer Benutzer konnte zum einen die Gefahr derartiger Angriffe für den einzelnen Benutzer nachgewiesen werden. Zum anderen wurde durch die gesammelten Daten der Grundstein für die Erforschung eines unabhängigen Erkennungssystems für Evil Twin Access Points gelegt.

- In dieser Arbeit wird der bereits bekannte und erforschte Angriffstyp des Evil Twin Access Point um eine mobile Komponente erweitert. Es wurde durch die Konzeption und durch eine prototypische Implementierung für iOS die Machbarkeit eines solchen, neuartigen Verbreitungsmechanismus gezeigt und analysiert. Basierend auf Standardeinstellungen moderner Betriebssysteme und aktueller Werkzeuge zur Entfernung von Nutzungsbeschränkungen konnte diese Art der Verbreitung implementiert werden.

- Im Rahmen eines angrenzenden Forschungsprojekts wurde ein Simulationsframework entwickelt mit Hilfe dessen sicherheitsrelevante Fragestellungen hinsichtlich mobiler Geräte untersucht werden können. Dieses Framework wurde dazu genutzt, die Ausbreitung der mobilen Malware in urbanen Gebieten zu simulieren. Es konnte im Rahmen dieser Arbeit gezeigt werden, dass sich eine derartige mobile Malware mit Hilfe der vorgestellten Mechanismen selbstständig weiterverbreiten kann und innerhalb kurzer Zeit eine großflächige Verbreitung erzielt werden kann.

- Um die bestehende Gefahr von Evil Twin Angriffen im Alltag zu untersuchen, wurde im Rahmen dieser Arbeit eine Studie durchgeführt. Mit Hilfe der gesammelten Daten und einer Befragung der Teilnehmer konnte nicht nur gezeigt werden, dass die Gefahr allgegenwärtig ist, sondern auch, dass bei einem Großteil der Benutzer kein Bewusstsein darüber besteht, dass sie einer derartigen Gefahr ausgesetzt sind.

- Ein weiterer Hauptbestandteil der Dissertation besteht in der Konzeption und Entwicklung eines Erkennungssystems für Evil Twin Angriffe. Es wurde ein autarkes System entwickelt, das mit Hilfe von Kontextparametern, die während des Verbindungsaufbaus erfasst werden, eine solche Erkennung durchführen kann. Zur Bestimmung dieser Kontextparameter wird ausschließlich die begrenzte Sensorik genutzt, die einem

mobilen Gerät (wie Smartphones oder Tablets) zur Verfügung steht. Mit Hilfe der im Rahmen der Feldstudie gesammelten Daten wurde das Erkennungssystem konfiguriert und an die Gegebenheiten der realen Welt angepasst.

- Zur Evaluation der Benutzbarkeit des Erkennungssystems wurden im Gegensatz zu verwandten Arbeiten reale Verbindungsdaten von Smartphonebenutzern verwendet. Durch Simulationen konnte gezeigt werden, dass bei den betrachteten Benutzern die Häufigkeit von Warnmeldungen durch die Anpassung der Parameter während der Konzeptions- und Implementierungsphase so weit minimiert werden konnte, dass ein alltäglicher Einsatz des Systems möglich erscheint. Um außerdem sicherzustellen, dass Angriffe zuverlässig erkannt werden, wurde das Erkennungssystem in eine Android-App portiert und evaluiert.

1.3 Aufbau der Arbeit

In diesem Kapitel wird neben einer Einleitung in das Thema der Dissertation und der Motivation für diese Arbeit der wissenschaftliche Beitrag erläutert.
Im zweiten Kapitel werden die für das Verständnis dieser Arbeit wesentlichen Grundlagen erläutert. Hierzu zählen neben Hotspot-Umgebungen, Captive Portals auch mögliche Angriffe auf derartige Infrastrukturen. Darüber hinaus wird ein Überblick über die beiden wegen ihrer Verbreitung relevanten mobilen Betriebssysteme gegeben und aufgezeigt, welche Möglichkeiten für Benutzer und entsprechend auch Angreifer existieren, um bestehende, sicherheitsrelevante Restriktionen der Systeme zu umgehen.
In Kapitel 3 werden verwandte, wissenschaftliche Arbeiten der angrenzenden Forschungsgebiete vorgestellt und in einen Kontext zu dieser Arbeit gesetzt. Ebenso wird beschrieben, welche weiterhin offenen Probleme mit den bestehenden Lösungen und Ansätzen bestehen.
Eine Sicherheitsanalyse von Hotspot-Umgebungen wird in Kapitel 4 vorgestellt. Es wird hierbei sowohl auf die Verschlüsselung öffentlicher Hotspots, als auch auf Captive Portals und Evil Twins als mögliche Angriffsvektoren für derartige Infrastrukturen eingegangen. Hierfür werden grundlegende Konzepte für solche Angriffe erläutert und ihre Durchführbarkeit diskutiert. Auch das

Verhalten moderner, mobiler Betriebssysteme wird beleuchtet und dargestellt, welche Faktoren und Funktionen die oben genannten Angriffe begünstigen.

In Kapitel 5 wird der Mobile Evil Twin-Angriff und die im Rahmen dieser Dissertation entstandene mobile Malware vorgestellt, die sich Eigenschaften moderner, mobiler Betriebssysteme zunutze macht, um sich mit Hilfe von Hotspots zu verbreiten. Um die Machbarkeit eines derartigen Angriffs unter Beweis zu stellen werden ebenfalls eine prototypische Implementierung und Messungen bezüglich des Zeit- Energieverhaltens vorgestellt.

Die Evaluation der beschriebenen Malware und ihrer Ausbreitung ist Inhalt des sechsten Kapitels. Zunächst wird auf das Ziel dieser Evaluation eingegangen. Im nächsten Teil wird aufgezeigt, wieso die Notwendigkeit für ein Simulationsframework zur Untersuchung von Malware-Ausbreitungen besteht. Nachdem die Anforderungen an ein solches Framework dargestellt wurden, wird die Simulation und Evaluation der oben beschriebenen Malware erläutert und anhand der Ergebnisse dargestellt, dass sich die schnelle Ausbreitung in urbanen Gebieten zu einer ernstzunehmenden Gefahr entwickeln kann.

Kapitel 7 beschreibt die Konzeption eines kontextbasierten Schutzsystems für Evil Twin Access Points. Zunächst werden die Anforderungen an ein solches System dargestellt. Des Weiteren wird beschrieben, welche Kontextparameter für ein derartiges System in Frage kommen und wie diese mit modernen mobilen Geräten gesammelt werden können. Im letzten Teil dieses Kapitels werden verschiedene Angriffstypen definiert und beschrieben, wie diese durch das entstehende Erkennungssystem erkannt werden können. In Kapitel 8 wird die Implementierung des Erkennungssystems und die Entwicklung der Erkennungsstrategie beschrieben. Darüber hinaus wird die Portierung auf die Android-Plattform erläutert und die Darstellung des Erkennungssystems im Betriebssystem erörtert.

Die Evaluation des beschriebenen Erkennungssystem wird in Kapitel 9 präsentiert. Es wird beschrieben wie Simulationen zur Bestimmung der Genauigkeit des Systems eingesetzt wurden und welche Ergebnisse mit Hilfe des Systems erzielt werden können.

Kapitel 10 fasst noch einmal die wesentlichen Ergebnisse und Erkenntnisse dieser Arbeit zusammen, gibt einige Empfehlungen, wie die Sicherheit heutiger Hotspot-Infrastrukturen verbessert werden kann und gibt einen Ausblick für

die zukünftigen Möglichkeiten, die aus den Ergebnissen dieser Arbeit resultieren.

Kapitel 2

Grundlagen

In diesem Kapitel werden grundlegende Begriffe der verschiedenen thematischen Bereiche erläutert, die im Rahmen dieser Arbeit berührt werden und die zum weiteren Verständnis erforderlich sind. Zunächst wird auf das Thema Hotspot-Umgebungen eingegangen und beschrieben wie diese in heutigen Infrastrukturen aufgebaut sind. Im Anschluss wird aufgezeigt, mit welchen Techniken ein Angreifer versuchen kann, Zugriff auf die privaten Daten von Anwendern dieser Infrastrukturen zu erlangen. Des Weiteren wird ein Überblick über Malware und Viren für mobile Geräte gegeben und dargestellt, welche Techniken in mobiler Malware eingesetzt werden, um sowohl Schadsoftware zu installieren, als auch um eine effektive Verbreitung sicherzustellen.

2.1 Hotspot-Umgebungen

Um einen möglichst barrierefreien Zugang zum Internet anzubieten, stellen viele Mobilfunkanbieter und Firmen Funknetze zur Verfügung, an denen sich ihre Kunden anmelden können, um Zugriff auf das Internet oder andere Services zu erhalten. Die Verbreitung dieser öffentlichen Netzwerke und ihr rasantes Wachstum [1] zeigt das große Interesse sowohl auf Seiten der Anbieter, als auch auf Seiten der Kunden. Viele Mobilfunkanbieter betreiben ganze Netzwerkinfrastrukturen, die sie ebenfalls ihren Kunden zur Verfügung stellen. Ein Beispiel für eine solche Infrastruktur betreibt die British Telecom. Ihre Infrastruktur besteht aus einem weltweiten Netzwerke, welches sie Kunden wie z.B. Starbucks zur Verfügung stellen [59]. Viele Mobilfunkanbieter bieten im Rahmen ihrer Mobilfunkverträge die Nutzung von Hotspots an. Eine Ad-Hoc Nutzung der Infrastruktur für Nicht-Kunden ist in den meisten Fällen gegen Gebühr ebenfalls möglich.

In den meisten Fällen werden die genannten Hotspots mit öffentlichen Netzwerken realisiert, die ohne die Eingabe eines Kennworts genutzt werden können. Ein wesentlicher Nachteil dieser Lösung besteht in der unverschlüsselten Kommunikation zwischen den Benutzern und dem entsprechenden Access Point. Auf diese Weise werden Man-in-the-Middle-Angriffe (MITMA) sehr vereinfacht und sensible Daten der Nutzer können bei der Nutzung unverschlüsselter Dienste in die Hände von Angreifern gelangen.

Anbieter solcher Netzwerke wollen den Zugang kontrollieren und beschränken. In den meisten Fällen, in denen öffentliche und unverschlüsselte Netzwerke zum Einsatz kommen, sichern sich die Anbieter zumindest rechtlich gegen die missbräuchliche Nutzung ihrer Infrastruktur ab. Um dies barrierefrei zu ermöglichen werden sog. Captive Portals eingesetzt, die im folgenden Abschnitt näher beschrieben werden.

2.2 Captive Portals

Bei einem Captive Portal handelt es sich um eine Webseite, auf welche ein Benutzer eines WLANs umgeleitet wird, um den Zugang zu beschränken. Zum einen wird es eingesetzt, um den Benutzernamen und das Passwort zu erfragen

und anhand dieser Informationen und einer angebundenen Authentifizierung- und Autorisierungsinfrastruktur (AAA) den Zugriff zu erlauben oder zu verwehren. Zum anderen werden Captive Portals auch in Funknetzwerken eingesetzt, die nicht an AAA-Komponenten angebunden sind. Zweck des Captive Portals ist hierbei zumeist die Absicherung des Anbieters, indem an dieser Stelle den allgemeinen Geschäftsbedingungen oder ähnlichen Bestimmungen zugestimmt werden muss, bevor der Zugriff freigeschaltet wird.

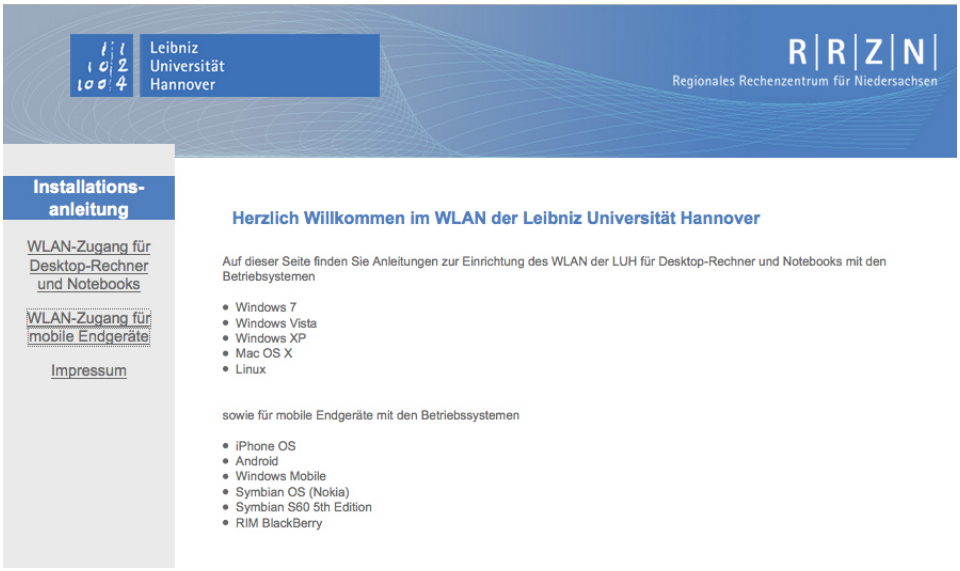


Abbildung 2.1: Captive Portal des WLANs der Universität Hannover.

In Abbildung 2.1 ist das Captive Portal der Universität Hannover dargestellt, welches einen informativen Charakter hat, um Studenten und Mitarbeitern Zugang zum Universitätsnetz zu gewähren.

Ablauf

Im Folgenden wird der Ablauf einer erfolgreichen Anmeldung an einem Captive Portal grafisch dargestellt und beschrieben. Neben dem Client, welcher sich mit dem Netzwerk verbinden möchte und dem Captive Portal selbst sind an diesem Vorgang auch AAA-Komponenten beteiligt, die über die Zugangsberechtigung entscheiden.
Im ersten Schritt sendet das Gerät des Benutzers einen normalen HTTP-Request für die gewünschte Internetseite aus. In einigen Betriebssystemen, wie beispielsweise iOS, geschieht dieser Request ohne das Zutun des Nutzers.

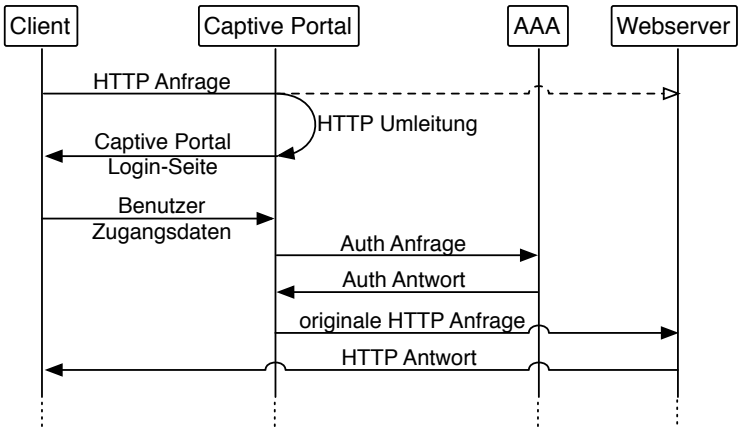


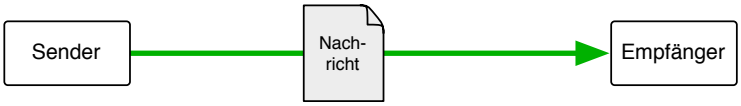
Abbildung 2.2: Ablauf einer erfolgreichen Anmeldung an einem Captive Portal

Hier wird automatisch die Seite des Captive Portals angezeigt, sobald sich das Gerät mit dem entsprechenden Funknetzwerk verbunden hat.
Das Captive Portal registriert den Zugriff eines nicht angemeldeten Benutzers auf das Netzwerk und leitet die ursprüngliche Anfrage auf eine entsprechende Anmeldeseite um. Nachdem der Benutzer seine Anmeldedaten eingegeben hat, werden diese zurück an das Captive Portal zur Verifizierung gesendet. In den meisten Infrastrukturen werden nun AAA-Komponenten in Anspruch genommen, da das Captive Portal selbst keine Berechtigungsentscheidung fällen kann. Somit wird die Anfrage an eine AAA-Infrastruktur weitergeleitet, welche das Ergebnis zurück an das Captive Portal übermittelt.
Bei einer negativen Entscheidung wird dem Benutzer eine entsprechende Fehlermeldung auf der Anmeldeseite angezeigt. Bei einer positiven Entscheidung wird der ursprünglich vom Captive Portal abgefangene und umgeleitete Request an den eigentlichen Empfänger gesendet und die Kommunikation mit allen durch das Netzwerk erreichbaren Servern kann beginnen. Eine detaillierte Erläuterung des Ablaufs findet sich in Abschnitt 4.2.

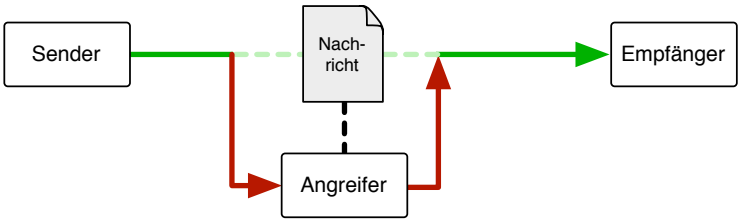
2.3 Man-in-the-Middle-Angriffe

Den Angriff auf eine Verbindung zwischen zwei Kommunikationsteilnehmern, bei dem Daten abgehört, verändert oder gelöscht werden nennt man Man-in-the-Middle-Angriff. Hierbei unterbricht der Angreifer die direkte Kommunikation zwischen den Kommunikationspartnern und gibt sich selbst auf beiden

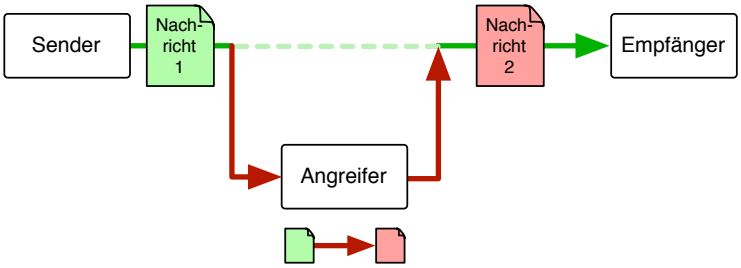
Seiten für den jeweils anderen Kommunikationspartner aus. Abbildung 2.3b zeigt wie ein Angreifer Nachrichten mit einem passiven Angriff belauscht und auf diese Weise an sensible Informationen gelangen kann. Ebenso ist in Abbildung 2.3c dargestellt, wie ein Angreifer die Daten auf ihrem Weg vom Sender zum Empfänger manipuliert.



(a) Nachricht wird ohne MITM-Angriff verschickt



(b) Passiver MITM-Angriff (Lauschen)



(c) Aktiver MITM-Angriff (Manipulation von Nachrichten)

Abbildung 2.3: Klassifizierung von Man-in-the-Middle Angriffen

Es gibt eine Vielzahl bekannter Angriffsmuster für Man-in-the-Middle-Angriffe. Diese werden nach den im Folgenden beschriebenen Kriterien unterschieden.

Passiv vs. Aktiv

Ein wesentliches Merkmal zur Unterscheidung von Man-in-the-Middle-Angriffen ist die Aktivität des Angreifers, also das reine Abhören einer Kommunikation im Gegensatz zur Veränderung oder Löschung von Teilen der Kommunikation. Schaltet sich ein Angreifer in die Kommunikation ein, um lediglich lauschend Informationen abzugreifen, so spricht man von einem passiven Angriff. Sollte der Angreifer Nachrichten zwischen den ursprünglichen Kommunikations-

partnern verändern, austauschen oder gar löschen, so spricht man von einem aktiven Angriff.

Bei einem passiven Angriff stellt die Offenlegung sensibler, geheimer Informationen die größte Gefahr dar. Bei einem aktiven Angriff hingegen können die Folgen einer manipulierten Kommunikation weitaus schwerwiegender sein. Während die Kommunikationspartner davon ausgehen, dass die ausgetauschten Informationen vertraulich und dementsprechend nicht für Dritte zugänglich sind, kann ein Angreifer Daten manipulieren, ohne dass beide Kommunikationspartner dies bemerken.

Beispiel Ein typisches Beispiel für einen Man-in-the-Middle-Angriff ist ein Bankkunde, der über den Browser eine Online-Überweisung durchführt. Bei einem erfolgreichen aktiven Angriff könnte ein Angreifer nicht nur den Empfänger der Überweisung manipulieren, sondern darüber hinaus auch den Betrag. Sollte der Angriff nicht umgehend durch den Kunden oder die Bank bemerkt werden, so wäre hätte dies finanzielle Konsequenzen zur Folge.

Geteiltes Medium vs. separates Medium (WLAN vs. LAN)

Ein anderes Merkmal zur Unterscheidung ist das Übertragungsmedium der Kommunikation. Die technischen Möglichkeiten für die Realisierung von Man-in-the-Middle-Angriffen variieren je nach Medium stark.

Ein Beispiel für die unterschiedlichen Komplexitäten derartiger Angriffe ist der physikalische Zugriff auf das Medium. Während die Daten in einem WLAN über das Medium Luft übertragen werden, muss bei einem kabelgebundenen Netzwerk zunächst durch einen Angreifer der Zugriff auf die vorhandene Infrastruktur gewährleistet werden. Ist dies sichergestellt, so sind einfache Netzwerkkomponenten wie Switches oder Router einsetzbar, um den anfallenden Netzwerkverkehr zu spiegeln und somit zu belauschen oder abzufangen und auf diese Weise einen aktiven Man-in-the-Middle-Angriff durchzuführen.

Für die Durchführung von Man-in-the-Middle-Angriffen stehen im Internet eine Vielzahl verschiedenster Werkzeuge frei zur Verfügung. Bekannte Werk-

zeuge wie Ettercap [1], Dsniff[2] und Burp Proxy[3] stellen jeweils eine Vielzahl von Angriffstechniken zur Verfügung, mit denen sich solche Angriffe realisieren lassen.
In einem WLAN wird ein geteiltes Medium zur Kommunikation verwendet. In den meisten Fällen sind derartige Netzwerke heutzutage verschlüsselt. Während das früher standardmäßig verwendete Verschlüsselungsprotokoll WEP seit 2005 als kryptografisch gebrochen gilt, hat sich heutzutage das Protokoll WPA bzw. seine Weiterentwicklung WPA2 durchgesetzt, die bis heute als sicher gilt. Weil die Verwendung von WPA bzw. WPA2 in vielen Fällen durch die Notwendigkeit eines vorangegangenen, sicheren Schlüsselaustauschs nicht praktikabel ist, wird wie bereits beschrieben in vielen Situationen auch heute noch auf eine Verschlüsselung verzichtet. Besonders verbreitet ist dieser Zustand in Hotspot-Infrastrukturen nahezu aller namhaften und großen Anbieter. In Verbindung mit WLANs lassen sich Man-in-the-Middle-Angriffe besonders einfach mit sog. Evil Twins realisieren. Die Funktionsweise eines solchen Evil Twins wird im nächsten Abschnitt beschrieben.

2.4 Evil Twins

Der Begriff des Evil Twins (bösartiger Zwilling) stammt aus der Literatur, wo er die physikalische Kopie eines Protagonisten darstellt, der gegenüber dem Protagonisten bösartige Absichten verfolgt. In Bezug auf ein IT-System bedeutet der Begriff des Evil Twins, dass eine Komponente einer IT-Infrastruktur vortäuscht, eine vertrauenswürdige andere Komponente zu sein. Dieses Verhalten ist auf verschiedenen Ebenen und in verschiedenen Dienstklassen möglich. Im Bereich der IT-Sicherheit und hier im Speziellen im Bereich der Netzwerke wird ein Access Point als Evil Twin bezeichnet, wenn er die SSID eines bekannten oder viel genutzten Hotspots oder Access Points verwendet, um Benutzer dazu zu bringen, sich mit ihm zu verbinden. Sowohl dem Benutzer als auch dem mobilen Gerät des Benutzers ist hierbei nicht bekannt, dass es sich bei der Verbindung nicht um den Access Point handelt. Ab dem Zeitpunkt der Verbindung wird ein Benutzer seinen gesamten Netzwerkverkehr über den Evil Twin

[1] `http://www.ettercap-project.org`
[2] `http://www.monkey.org/~dugsong/dsniff/`
[3] `http://portswigger.net/burp/proxy.html`

abwickeln, ohne dies zu bemerken. Das gibt dem Betreiber des Evil Twins eine Reihe von Möglichkeiten. So könnte er die Aktionen des Benutzer einerseits erfassen und ausspähen, andererseits sind auch Manipulationen der übertragenen Daten (sowohl ausgehende, als auch eingehende Daten) denkbar und potentiell möglich. Ausführlichere Erläuterungen in Bezug auf die Möglichkeiten und die zugrundeliegenden Techniken befinden sich in Kapitel 4.

Bedingungen für einen erfolgreichen Evil Twin Angriff

Im Folgenden sollen die Bedingungen dargestellt werden, die für einen erfolgreichen Angriff mit Hilfe eines Evil Twin Access Points gegeben sein müssen. Es werden die verschiedenen Bedingungen diskutiert und erläutert, wie diese durch einen Angreifer realisierbar sind.

Wahl der SSID Die Wahl der SSID, also eines geeigneten Namens für das zum Angriff zu nutzende Funknetzwerk hängt maßgeblich von den Zielen des Angriffs ab. Es muss an dieser Stelle differenziert werden zwischen einem allgemeinen, ungerichteten Angriff und einem gezielten Angriff auf bestimmte Personen. Soll eine bestimmte Person angegriffen werden, so bedient man sich der Person entsprechend einer SSID, die diese Person mit hoher Wahrscheinlichkeit auf ihrem mobilen Gerät konfiguriert hat. Hierbei kann es sich um ein Unternehmens-WLAN oder auch um ein privat genutztes Netzwerk handeln. Bei einem breit gefächerten Angriff auf keine spezielle Person wird eine SSID gewählt, die von möglichst vielen Personen bereits mindestens einmal genutzt wurde. Denkbar sind hier freie WLANs bekannter Unternehmen wie McDonald's oder Starbucks. Ebenso sind von Mobilfunkbetreibern betriebene Hotspots denkbar, die stets dieselbe SSID verwenden und entsprechend ebenso verbreitet sind.

Eine weitere Möglichkeit eine SSID zu finden, die sich für einen Angriff in unmittelbarer Umgebung eignet, ist es die SSIDs aus den sog. *Probe Requests* auszulesen, die Smartphones in regelmäßigen Abständen aussenden, wenn sie mit keinem WLAN verbunden sind. Weitere Details zu diesem Vorgehen finden sich in Abschnitt 5.1.

Örtliche Bedingungen Beim Betrieb eines Evil Twins muss durch den Angreifer dafür gesorgt werden, dass sich Geräte automatisiert mit dem bereitgestellten Access Point verbinden. Dies kann er durch zwei Vorgehen sicherstellen. Die einfachste Möglichkeit ist es, den Access Point an einem Ort zu betreiben, an dem sich kein originaler Access Point befindet.

Soll der Access Point hingegen in der Nähe des originalen und somit zu fälschenden betrieben werden, so muss der Angreifer dafür sorgen, dass die Signalstärke seines Access Points größer ist, als die des originalen. Die Geräte der Benutzer scannen alle Funknetze in der Umgebung und suchen bei mehreren vorhandenen Access Point denjenigen mit der größten Signalstärke aus. Auf diese Weise würden sich die entsprechenden Geräte mit dem Access Point des Angreifers verbinden und der Angriff könnte gestartet werden.

Fälschen der BSSID Jeder Access Point und jedes netzwerkfähige Gerät besitzt eine im Normalfall eindeutige ID, die sogenannte BSSID bzw. MAC-Adresse. Die BSSID ist eine 12-stellige hexadezimale Zahl und bietet entsprechend Platz für mehr als 281 Trillionen unterscheidbare Geräte. Anhand dieser BSSID könnte ein für einen Angriff genutzter, gefälschter Access Point enttarnt werden. Leider ist die Fälschung der BSSID mit vielen aktuellen Netzwerkkarten und in allen heutigen Betriebssystemen ohne weiteres möglich. Unter anderem hinter dem Begriff des MAC Spoofings verbergen sich Angriffe, die von der Fälschung der MAC-Adresse Gebrauch machen.

Sabotage des originären Access Points Soll ein Angriff in räumlicher Nähe eines originalen, zu fälschenden Access Points durchgeführt werden und ist es darüber hinaus nicht möglich eine größere Signalstärke als der originale Access Point zu erreichen, so muss ein Angreifer versuchen, den originalen Access Point vorübergehend außer Betrieb zu nehmen bzw. zu sabotieren. Dies kann er potentiell durch einfache Maßnahmen wie die Trennung vom Stromnetz erreichen. Ebenso sind auch komplexe Angriffe auf den Access Point bzw. seine Administrationsoberfläche denkbar, um den Betrieb einzustellen.

Bedrohungs- und Risikoanalyse

Wie bei der Risikoanalyse für allgemeine Angriffe auf IT-Systeme gilt es auch bei der speziellen Art von Evil Twin Angriffen die potentiellen Angreifer in festgelegte Kategorien einzugruppieren. Auf diese Weise kann das Risiko, dass von einem Angreifer ausgeht besser eingeschätzt und entsprechende Gegenmaßnahmen konzipiert werden. Oberhaitzinger [47] beschreibt hierfür eine Eingruppierung nach drei Kriterien, die in Abbildung 2.4 grafisch dargestellt sind.

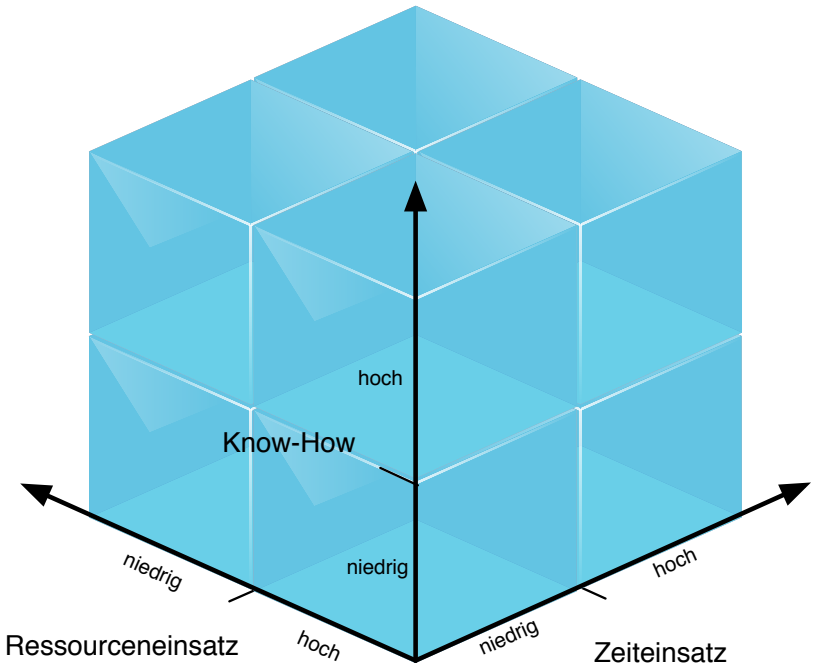


Abbildung 2.4: Klassifizierung von Angreifern (in Anlehnung an Oberhaitzinger [47])

Zeiteinsatz Die aufgewendete Zeit, sowohl für die Vorbereitung als auch für die Durchführung eines Angriffs haben in vielen Fällen einen direkten Einfluss auf den Erfolg eines Angriffs und somit auf das mit ihm verbundene Risiko für den Benutzer.

Know-how Auch das Know-how eines Angreifers hat direkte Auswirkungen auf das Risiko eines Angriffs. Das Know-how entscheidet in vielen Fällen darüber, ob es sich bei Angriffen um großflächige, allgemeine Angriffe durch Personen mit geringem Kenntnisstand oder um komplizierte und zielgerichtete

Angriffe durch Experten handelt. Zielgerichtete Angriffe sind hierbei in jedem Fall mit einem höheren Risiko zu bewerten.

Ressourceneinsatz Der Ressourceneinsatz hat einen eher indirekten Einfluss auf das Risiko eines Angriffs. Stehen einem Angreifer viele Ressourcen zur Verfügung, so beeinflusst dies auch die beiden zuvor genannten Faktoren. Sowohl die aufgewendete Zeit für Vorbereitungen und Einarbeitungen, als auch der zusätzlich Aufbau von Know-how können mit Hilfe eines großen Ressourceneinsatzes begünstigt werden. Zu diesen Faktoren kommt hinzu, dass bei Angriffen auch der Einsatz hochwertiger Hardware für den Angreifer notwendig werden kann. In diesem Fall kann das Risiko eines Angriffs auch direkt durch den Einsatz monetärer Ressourcen gesteigert werden.
Eine andere Betrachtung des Risikos entsteht aus dem Zusammenhang zwischen Schwachstellen, Bedrohungen und Risiken. Eckert [7] stellt die Beziehung zwischen diesen Entitäten wie in Abbildung 2.5 dargestellt vor.

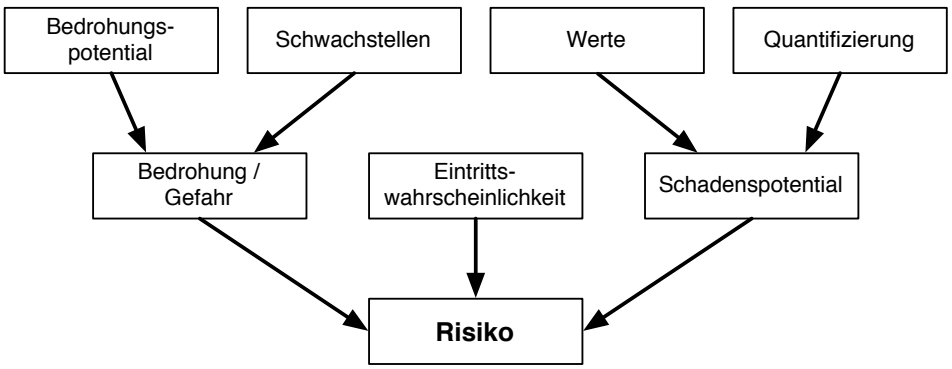


Abbildung 2.5: Zusammenhang zwischen Schwachstellen, Bedrohungen und Risiken nach Eckert [7]

Das Risiko was aus einem Angriff resultiert setzt sich aus den drei Faktoren der Bedrohung, der Eintrittswahrscheinlichkeit eines Angriffs und dem Schadenspotential zusammen. Die Bedrohung seinerseits ergibt sich aus dem Bedrohungspotential, was für eine spezifische Bedrohung existiert und aus Schwachstellen, die zur Durchführung eines Angriffs genutzt werden. Diese Schwachstellen müssen zum einen in der anzugreifenden Software enthalten und zum anderen durch den Angreifer ausnutzbar sein. Das Schadenspotential selbst ergibt sich zum einen aus den Werten, die durch einen Angriff gefährdet sind.

Dies können monetäre, aber auch ideelle und geistige Werte sein. Zum anderen spielt die Anzahl möglicher Angriffsziele eine entscheidende Rolle. Alle Faktoren vereinen sich schlussendlich im Gesamtrisiko.

2.5 Mobile Malware

Im Juni 2004 wurde mit Cabir [61, 16] die erste Malware für Smartphones entdeckt. Cabir war ein Wurm für das zu dieser Zeit für Smartphones beliebte mobile Betriebssystem SymbianOS. Cabir hatte als mobile Malware erstmals die Eigenschaft, sich über die Bluetooth-Schnittstelle der Smartphones von einem Gerät auf ein anderes übertragen zu können. Auf diese Weise konnte sich der Wurm ausbreiten. Es handelte sich bei Cabir um eine Proof-of-concept-Implementierung mit der lediglich die Machbarkeit eines derartigen Angriffs unter Beweis gestellt werden sollte. Er enthielt keinerlei Schadcode. Der Sicherheitsforscher und Chief Research Officer der Firma F-Secure Miko Hyppönen hat bereits im Jahr 2006 in einem Journal-Artikel [43] beschrieben, dass die auf Desktop- und Serverrechnern bekannten Malwarestrukturen in Zukunft auch auf Smartphones und anderen mobilen Geräten Einzug halten werden. Er beschreibt bereits zu dieser frühen Zeit, dass dieser Bereich eine rasante Entwicklung durchläuft. Von der ersten bekannten Malware aus 2004 bis zum Zeitpunkt der Veröffentlichung seines Beitrags im November 2006 waren bereits mehr als 320 aktive mobile Malwares durch verschiedene Systeme der Antivirenhersteller und von Sicherheitsforschern identifiziert und klassifiziert worden. Auch die stark zunehmende Anzahl an Smartphones stellt einen wesentlichen Faktor für die schnelle Verbreitung mobiler Malware dar. Je mehr potentiell angreifbare Geräte im Umlauf sind, desto größer ist der Anreiz für Angreifer entsprechende Malwares zu entwickeln und sie in Umlauf zu bringen.

2.6 Administrativer Zugriff auf mobilen Geräten

Bevor man sich Gedanken über die Gefahren macht, die ein administrativer Zugriff auf mobile Geräte bedeutet, muss zunächst geklärt werden, aus welchen

Gründen Benutzer an dieser Form der Befreiung interessiert sind. In erster Linie wird ein Jailbreak, also die Erlangung von administrativen Rechten auf den Geräten, heutzutage von Benutzern dazu verwendet, auf den Geräten neben Apps aus den jeweiligen App Stores auch Apps aus anderen Quellen zu installieren und zu betreiben. Dies ist in allen bisherigen iOS-Versionen nicht gestattet und wird von Apple unterbunden. In der Anfangszeit der Smartphones waren neu erworbene Smartphones zumeist per Netlock an den Provider gebunden, bei dem das Gerät gekauft wurde. Die Nutzung des Smartphones in anderen Handy-Netzen war nicht möglich. Ein Jailbreak brachte zur dieser Zeit ebenfalls die Möglichkeit mit sich, den Netlock mit Hilfe eines Programms aufzuheben. Von diesem Zeitpunkt an konnte das Smartphone mit allen SIM-Karten und in allen Netzen verwendet werden.
Benutzer mit den oben beschriebenen Interessen stoßen den Prozess bewusst und beabsichtigt an. Eine Malware hingegen, die sich ebenso den administrativen Zugang zunutze macht, installiert in den meisten Fällen Software, die dem Benutzer verborgen bleibt. Damit eine Malware systemnahe Befehle ausführen oder Änderungen an Systemeigenschaften vornehmen kann, benötigt auch sie administrativen Zugang zum Betriebssystem. Wie auch bei Desktop- und Server-Betriebssystemen liegt den mobilen Betriebssystemen ebenfalls ein Benutzer-basiertes Zugriffsmodell zu Grunde. iOS basiert auf Unix, Android auf Linux. Beide besitzen entsprechend mindestens ein Benutzerkonto, welches über Administrationsprivilegien verfügt und somit den vollen Zugriff auf das System hat. Diese Benutzerkonten sind auf normalem Wege nicht zu erreichen. Durch Mechanismen wie Sandboxing sind Apps nicht in der Lage auf Ressourcen des Systems oder anderer Apps zuzugreifen. An dieser Stelle sei auf Abschnitt 5.3 verwiesen, in dem eine detailliertere Beschreibung dieses Mechanismus zu finden ist.
Unter iOS wird für den Ausbruch aus dieser Sandbox und die Erlangung der Administrationsrechte der Begriff *Jailbreak* verwendet, wohingegen sich unter Android der Begriff *Rooting* durchgesetzt hat.

2.6.1 Jailbreak unter iOS

Das erste iPhone wurde von Apple im Januar 2007 vorgestellt und konnte ab dem 29. Juni 2007 zunächst ausschließlich in den USA gekauft werden. Bereits

11 Tage nach dem Verkaufsstart am 10. Juli 2007 wurde der erste nachgewiesene Jailbreak dokumentiert. Dieser erlaubte zunächst lediglich das Hinzufügen beliebiger Klingeltöne. Durch diese Möglichkeit konnte seinerzeit der Zugriff auf das Dateisystem des iPhones nachgewiesen werden, in dem der Klingelton abgelegt werden musste, um ihn mit Hilfe der nativen Systemeinstellungs-App auswählen zu können. Auch für die nachfolgenden Versionen sind in kürzester entsprechende Jailbreaks verfügbar gewesen.

In Anhang A.1 befindet sich eine Auflistung aller bis zur Abgabe dieser Dissertation verfügbaren iOS-Versionen zusammen mit einem Vermerk, ob für die entsprechende Version ein Jailbreak öffentlich verfügbar ist oder war. Darüber hinaus befindet sich in Anhang A.2 eine Tabelle mit allen bisher für Jailbreaks eingesetzten Werkzeuge in Abhängigkeit von der eingesetzten iOS-Version.

Es gilt stets zu bedenken, dass auch für Versionen, für die kein Jailbreak Werkzeug öffentlich zugänglich war, Exploits für den administrativen Zugriff vorhanden waren und sind. Aus verschiedenen Gründen wurden diese zurückgehalten. Gerade in der Anfangszeit der Jailbreak-Szene mussten zunächst für den Endanwender einfach benutzbare Jailbreak-Lösungen entwickelt werden. Ebenso muss damit gerechnet werden, dass eine Vielzahl weiterer Exploits im Verborgenen gehandelt und eingesetzt werden. An dieser Situation hat sich bis heute nichts geändert.

Für die Realisierung der verschiedenen Jailbreak-Lösungen wurden eine Vielzahl von Exploits verwendet. In der Regel wurden sogar mehrere Exploits benötigt, um den Jailbreak einer Version durchzuführen. Diese verschiedenen Exploits haben ebenso verschiedene technische Voraussetzungen, auf die im Folgenden kurz eingegangen werden soll.

Arten des Jailbreaks Zunächst gilt es zu unterscheiden ob es sich um einen sog. *tethered* (engl. „angebunden“) oder einen *untethered* Jailbreak handelt. Der Unterschied liegt darin, dass ein *tethered* Jailbreak die Anbindung des Smartphones an einen Computer bei jedem Bootvorgang des Telefons erfordert. Dies zieht einige Nachteile nach sich. So wäre ein Einschalten, Neustart oder auch ein Neustart nach dem Absturz des Telefons nicht möglich, ohne Zugriff auf einen Computer mit dem entsprechenden Jailbreak Werkzeug zu haben. Speziell bei mobilen Geräten wie Smartphones ist dies oftmals nicht der Fall, was das

Gerät zunächst gänzlich funktionsunfähig macht. Bei einem *untethered* Jailbreak hingegen wird eine solche Verbindung zu einem Computer beim Booten des Smartphones nicht benötigt. Alle für das Entsperren beim Neustart benötigten Daten befinden sich hierbei auf dem Smartphone und können direkt genutzt werden. Die dritte Variante ist der *semi-tethered* Jailbreak. Hierbei startet das zuvor gejailbreakte Gerät zwar normal. Es lassen sich nun aber ausschließlich Grundfunktionen des Gerätes verwenden. Insbesondere lassen sich keine Systemerweiterungen oder zusätzlich aus dem Cydia Store[4] geladene Apps verwendet werden. Erst nach erneutem Anschließen des Geräts an einen Computer mit der entsprechenden Jailbreak-Software lassen sich diese Funktionen wieder freischalten. Der Nachteil eines tethered Jailbreaks wird hierbei also durch eine eingeschränkte Nutzungsmöglichkeit aufgehoben.

Bootvorgang iOS Um weitere Unterscheidungen einordnen zu können wird im Folgenden kurz auf den Bootvorgang eines iOS Geräts eingegangen. Der zeitliche Ablauf beim Booten eines iOS-Geräts kann folgendermaßen dargestellt werden:

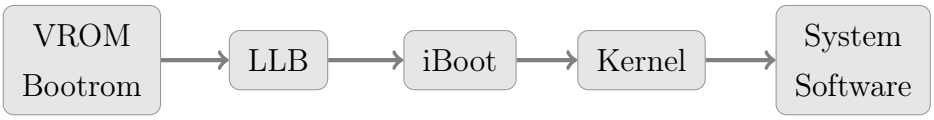


Abbildung 2.6: Zeitlicher Ablauf des iOS Bootvorgangs

Nach dem Einschalten des Gerätes wird zunächst auf das sog. Bootrom zugegriffen. Hier liegt der erste Code, der nach dem Start des Gerätes ausgeführt wird. Schon zu diesem frühen Zeitpunkt des Bootprozesses befindet sich die Möglichkeit in den sog. *Device Firmware Upgrade*-Mode (DFU-Mode) zu wechseln. Im DFU-Mode kann auf das Gerät eine neue Firmware aufgespielt werden. Sollte der DFU-Mode nicht gestartet werden, so wird der *Low Level Bootloader* (LLB) gestartet. Neben hardwarenahen Konfigurationen sorgt der LLB für den korrekten Start des Stage 2-Bootloaders *iBoot*. Seit iOS 2.0 wird an dieser Stelle zusätzlich die Signatur von iBoot vor dem Start validiert. Sollte eine nicht von Apple signierte Version von iBoot gefunden werden, so bricht der

[4]Ein Marktplatz für kostenlose und kostenpflichtige Apps, die nur für gejailbreakte verfügbar sind.

Startvorgang ab. iBoot selbst beinhaltet neben dem *Recovery Mode* die Routinen zum Starten des Unix Darwin Kernels und entsprechend das Laden und Ausführen der Firmware und der enthaltenen Systemdateien. Der Recovery Mode seinerseits bietet die Möglichkeit eine neue Version des Betriebssystems einzuspielen.

Userland exploit vs. Bootrom exploit Für Jailbreaks wurde und wird in allen in Abbildung 2.6 dargestellten Modulen, die beim Starten eines iOS Geräts geladen werden, nach Exploits gesucht. Dabei sind Schwachstellen im Bootrom aus mehreren Gründen schwerer zu finden als beispielsweise solche in der Systemsoftware. Zum einen ist ein Bootrom im Vergleich zur Systemsoftware sehr kompakt und systemnah programmiert. Zum anderen gibt es für die einzelnen iOS-Geräte jeweils angepasste Bootroms. Die Nachteile von Bootrom-Exploits ergeben sich entsprechend dadurch, dass gerätespezifische Lücken ausfindig gemacht und ausgenutzt werden müssen. Ein enormer Vorteil dieser Schwachstellen ist, dass sie nicht wie Schwachstellen in der Systemsoftware durch ein einfaches Software-Update seitens Apple behoben bzw. geschlossen werden können. Die im Bootrom enthaltenen Schwachstellen können nicht behoben werden, da nur ein Hardwareaustausch dies ermöglichen würde. Userland exploits, welche entsprechend leichter gefunden werden können sind zumeist geräteübergreifend ausnutzbar. Sie haben aber im Gegensatz zu Bootrom- Exploits den Nachteil, dass ein Software-Update die Ausnutzung der verwendeten Exploits unmöglich macht und entsprechend neue Schwachstellen gefunden werden müssen.

Möglichkeiten und Gefahren Wie bereits angedeutet bietet der administrative Zugriff auf ein Gerät viele Möglichkeiten der Anpassung und Erweiterung seiner Funktionalität. So erlaubt ein Jailbreak unter iOS nicht nur die Installation von Apps, die nicht aus dem Apple-eigenen AppStore stammen, sondern auch die Anpassung und Erweiterung systemnaher Funktionen und Dienste. So können beispielsweise modifizierte Varianten der Bedienoberfläche und Erweiterungen bereits vorhandener Funktionalitäten installiert werden. Während des Jailbreaks wird auf den Geräten ein Paketsystem installiert, welches die Installation weiterer Komponenten steuert. Hierbei handelt es sich um

ein Standard Paketsystem, welches bereits aus der Linux-Distribution Debian bekannt ist.
Apps und Erweiterungsmodule sind aber nicht das einzige was mit Hilfe des Paketsystems installiert und betrieben werden kann. Es ist ebenso möglich, Dienste und Server auf einem Gerät zu installieren, welche fortan auf dem Gerät laufen und ihre Funktionalität bereitstellen. Dies stellt zur Zeit die größte Gefahr für den unerfahrenen Benutzer dar. Sie führen einen Jailbreak durch, installieren im Anschluss daran Dienste und aktivieren diese ohne deren genaue Bedeutung und Funktionsweise zu verstehen. Ein Beispiel für einen solchen Dienst ist der OpenSSH Server, der ebenfalls einfach über das Paketsystem installiert werden kann. Die große Gefahr, die bei der Installation lauert, ist das vorhandene Administrationskonto mit seinem Standardpasswort, welches auf alles iOS-Geräten identisch ist. Das hat zur Folge, das ein Benutzer, der diesen Dienst installiert und keine weitere Konfiguration vornimmt, von jedem Angreifer, der per Netzwerk auf sein Telefon zugreifen kann, angreifbar ist.

2.6.2 Rooting unter Android

Der Begriff des Jailbreakens ist in der Android-Welt nicht üblich. Hier wird ein Smartphone *gerootet*, um sich den vollen Zugriff auf alle Teile des Systems zu verschaffen. Während für iOS nur wenige Geräte- und Softwarekombinationen betrachtet werden müssen, sieht das bei Android anders aus. Mitte 2012 hat das Unternehmen Staircase Zahlen zur Fragmentierung von Android veröffentlicht, die sie im Rahmen einer Erhebung mit einem ihrer verbreiteten Softwareprodukte sammeln konnten [48]. Hierbei konnten bereits knapp 4.000 unterschiedliche Smartphone-Modelle auf Android-Basis beobachtet werden. Obwohl einige dieser beobachteten Geräte auf selbsterstellten Custom-ROMs[5] basieren, zeigt die Statistik trotzdem deutlich die zunehmende Fragmentierung im Android-Lager. Durch die Vielzahl verschiedener Modelle sind ebenfalls sehr viel mehr Software-Versionen des Android-Betriebssystems aktiv in Verwendung, als es bei iOS der Fall ist. Da wie auch unter iOS die verwendete Hardware in Verbindung mit der Softwareversion ausschlaggebend für die

[5]Ein Custom-ROM ist eine selbst kompilierte und installierte Version eines Betriebssystems. Es unterscheidet sich nicht nur in der Funktionalität sondern auch in den Herstellerbezeichnungen und -versionen von offiziell verfügbaren Versionen.

Möglichkeit des Rootings ist, ergeben sich an dieser Stelle sehr viele Kombinationsmöglichkeiten. Im Rahmen von Untersuchungen von Android Malware haben Felt et al. in einer ihrer Arbeiten [17] bereits gezeigt, dass die Möglichkeit des Rootings bei den von ihnen betrachteten Geräten im Schnitt bei 88,3% gegeben war. Betrachtet wurde hierbei der Lebenszyklus des Geräts. Bei einigen betrachteten Geräten lag die Verfügbarkeit eines entsprechenden Root-Exploits bei mehr als 90% der Lebensdauer.
So unterschiedlich die beiden Hardware- und Softwarewelten von Android und iOS auch sind, die öffentliche Verfügbarkeit von Root-Exploits ist auf beiden Plattformen gegeben. Die zur Verfügung stehenden Werkzeuge nutzen in beiden Welten vorhandene Sicherheitslücken in den Betriebssystemen aus, um den Administrationszugriff auf den Geräten zu ermöglichen. Die verwendeten Lücken können durch motivierte Angreifer durchaus auch für weniger ehrenhafte Zwecke, wie beispielsweise Spionage oder das Ausspähen privater Informationen zweckentfremdet werden. Auf beiden Plattformen haben in der Vergangenheit nahezu durchgängig die hierfür notwendigen Lücken bestanden und wurden innerhalb sehr kurzer Zeit ausgenutzt. Ein nachlassender Trend hinsichtlich dieser Entwicklung ist nicht zu beobachten. Auch für die heute aktuellen Versionen von Android und iOS sind Jailbreaks bzw. Werkzeuge für ein Rooting verfügbar. Es ist entsprechend davon auszugehen, dass sich diese Situation auch in Zukunft nicht ändern wird.

Kapitel 3

Aktueller Stand der Technik und offene Probleme

In diesem Kapitel wird zunächst der aktuelle Stand der Technik im Bereich mobiler Malware, ihrer Simulation und dem Schutz vor Evil Twins beschrieben und dargestellt welche offenen Probleme mit den heute zur Verfügung stehenden Systemen und Lösungsansätzen bestehen.

3.1 Verwandte Arbeiten

Im Rahmen dieser Dissertation werden sicherheitskritische Probleme mit Evil Twin Access Points in Funknetzwerken dargestellt. Ebenso werden Lösungen zur Absicherung gegen diese Art von Angriffen erforscht und evaluiert. Des Weiteren entsteht die prototypische Implementierung eines Schutzsystems. Entsprechend werden in dieser Arbeit viele Themengebiete berührt, für die relevante, wissenschaftliche Vorarbeiten existieren. Diese werden im Folgenden gruppiert nach Themengebieten beleuchtet und in einen Kontext zu dieser Arbeit gesetzt.

3.1.1 Mobile Malware

Im Bereich mobiler Malware gibt es zahlreiche wissenschaftliche Arbeiten. Die im Folgenden aufgeführten sind in erster Linie Arbeiten, welche die Entwicklung dieser verhältnismäßig neuen Art der Malware beschreiben und untersuchen. Ebenso werden Arbeiten beleuchtet, welche die Gefahren mobiler Malware erforschen und Ansätze für Gegenmaßnahmen präsentieren.
Alegre-Sanahuja et al. beschreiben ein agentenbasiertes Modell [32] mit dem sie den Anteil an Android-Smartphones bestimmen können, die mit einer Malware infiziert sind. Mit Hilfe des von Ihnen entwickelten Modells kommen sie auf eine Ansteckungsrate von rund 0,3% pro Monat. Diese Zahl soll über den betrachteten Zeitraum von zwei Jahren nahezu konstant gewesen sein. Mit Hilfe eines kumulativen Ansatzes auf den gesammelten Daten kommen sie auf einen Gesamtinfektionsanteil von mehr als 13% in der Population der von ihnen betrachteten Geräte. Die Ergebnisse zeigen, dass sich die Verbreitung von Malware auf Smartphones zu einem zunehmend besorgniserregenden Problem entwickelt.
Schmidt et al. stellen in einer ihrer Arbeiten [56] beispielhaft eine Malware für die Android-Plattform vor, die in der zu seiner Zeit aktuellen Android-Version beliebigen Code ausführen konnte. Mit Hilfe dieser Konzeptimplementierung wollen die Autoren darauf aufmerksam machen, dass Smartphones im Allgemeinen und Android-Smartphones im Speziellen lohnenswerte und mögliche Ziele für Angreifer darstellen. Des Weiteren stellen Schmidt et al. eine Statistik vor, die beschreibt, über welche Wege Malwares ihren Weg auf die Geräte

der Benutzer finden. Über 76% der untersuchten Malwares wurden durch die Installation einer App auf dem Telefon installiert. Denkbar sind an dieser Stelle Trojaner-Apps, die vorgeben eine andere Funktion zu erfüllen. Besonders beliebte Apps sind hierbei Spiele, die sehr beliebt sind und eine schnelle Verbreitung aufweisen. Die weiteren relevanten Einfallstore für Malware waren zum Zeitpunkt ihrer Arbeit die Bluetooth-Schnittstelle und MMS.

Allgemeine Untersuchungen zum Stand mobiler Malware, der zu Grunde liegenden Motivation und Möglichkeiten zur Eindämmung wurden von Porter Felt et al. in ihrer Arbeit [17] durchgeführt. Die Motivation hinter mobiler Malware unterliegt einer ähnlichen Entwicklung wie die herkömmlicher Malware für Desktop-Computer. Während es in der Anfangsphase vermehrt um humoristische Konzeptimplementierungen wie Eeki.A [50] und Smspacem [62] handelt, ändert sich die Motivation schnell dahingehend, dass monetäre Aspekte im Fokus der Malware-Entwickler liegen. Das Versenden von Premium-SMS oder der Diebstahl von Zugangsdaten für die verschiedensten Dienste gehören hierbei zu den prominentesten Beispielen. Sie haben ebenfalls das Vorkommen verschiedener Malwares auf den unterschiedlichen Plattformen untersucht und hierbei festgestellt, dass der von Apple durchgeführte Review-Prozess[1] für Apps ein sehr gutes Ergebnis erzielt und keine der untersuchten Malwares den Review-Prozess mit positivem Resultat durchlaufen hat. Porter Felt et al. haben ebenfalls herausgefunden, dass die Suche nach Root-Exploits[2] in vielen Fällen nicht nur durch Angreifer im ursprünglichen Sinn vorangetrieben wird, sondern oftmals durch Online-Communities, die sich das Ziel gesetzt haben, die Software auf den Geräten nach eigenen Vorstellungen zu designen und anzupassen. Eine für diese Dissertation ebenfalls wichtige Erkenntnis ist, dass zu 74 - 100% der Gesamtlebenszeit eines Smartphones Root-Exploits existieren und veröffentlicht sind, die sowohl von den beschriebenen Communities, aber auch von Angreifern ausgenutzt werden können.

[1] Um eine App in Apples AppStore einzustellen, müssen Entwickler ihre Apps für ein Review einreichen. Die genaue interne Verfahrensweise gibt Apple nicht bekannt. Die Vergangenheit hat gezeigt, dass an dieser Stelle nicht nur automatisierte Tests durchlaufen werden, sondern zusätzlich auch menschliche Gutachter zum Einsatz kommen.

[2] Fehler in Software oder Betriebssystemen, die einem Angreifer dazu verhelfen administrative Rechte auf dem jeweiligen Gerät zu erlangen

Ho et al. haben in ihrer Arbeit [27] den Ansatz von Bose und Schmidt erweitert. Während Bose et al. [3] in ihrer Arbeit ein Erkennungssystem für Malware vorschlagen, welches auf Basis von Verhaltensmustern der Apps arbeitet, verfolgen Schmidt et al. in ihrer Arbeit den Ansatz [55] Systemeigenschaften und spezielle Systemereignisse zu beobachten. Dies wird durch eine Client-Software auf dem mobilen Gerät selbst durchgeführt. Die so gesammelten Daten werden im Anschluss an einen zentralen Dienst gesendet, der die eigentliche Angriffserkennung durchführt. Je nach Ergebnis der Prüfung wird eine entsprechende Meldung an das mobile Gerät zurückgesandt. Ho et al. erweitern diese Ansätze in einem herstellerunabhängigen Modell und schlagen vor, dass vor dem Senden von Informationen durch ein Erkennungssystem geprüft werden muss, ob es sich bei der durchzuführenden Aktion um eine vom Benutzer angestoßene handelt. Sollte dies nicht der Fall sein, so gehen Ho et al. davon aus, dass mit hoher Wahrscheinlichkeit eine Malware potentiell sensible und private Informationen verschickt. Mit ihrem Ansatz kann die Verbreitung von Malware nicht gestoppt werden. Der Anspruch den Ho et al. verfolgen ist es, die Ausbreitung einer Malware zu verlangsamen und sie auf diese Weise eindämmen zu können.

Bei der Ausbreitung mobiler Malware können auch zunächst unbetroffene Nutzer durch eine Malware ausspioniert werden. Husted et al. [28] zeigen in ihrer Arbeit, dass eine mobile Malware so weit verbreitet sein kann, dass mit ihrer Hilfe auch Unbeteiligte – hier mobile Endgeräte, die nicht mit der Malware infiziert sind – zu Opfern werden können. Sie nutzen das Simulationsframework UDel und erstellen auf diese Weise Pfade für die einzelnen simulierten Agenten. Ziel der Untersuchung ist es herauszufinden, wie gut nicht infizierte Geräte durch infizierte verfolgt werden können, in dem sie in ihrer Nähe aufgespürt werden. Die infizierten Geräte spannen hierfür eine Art Malware-Netzwerk (auch Malnet genannt) auf. Husted et al. haben mit Hilfe verschiedener Simulationen herausgefunden, dass schon ein kleiner prozentualer Anteil infizierter Geräte ausreichend ist, um eine nahezu lückenlose Überwachung der nicht infizierten Geräte erreichen zu können. Diese Beobachtung ist besonders unter dem Gesichtspunkt der gezielten Überwachung einzelner Personen interessant. Zur Überwachung dieser Person könnten für eine möglichst unauffällige Verfahrensweise nur die Geräte nahestehender Personen infiziert und überwacht werden, nicht jedoch das Gerät der Zielperson selbst.

3.1.2 Simulation mobiler Malware

In der Erforschung mobiler Malware sind bereits verschiedenste Modelle zur Simulation von Angriffsvektoren, als auch deren Gegenmaßnahmen zum Einsatz gekommen und in wissenschaftlichen Arbeiten beschrieben worden [34, 74, 12, 13, 52, 46]. Viele dieser Modelle stützen sich auf mathematische Modelle, die versuchen die Eigenschaften der Ausbreitung von realen, natürlichen Viren auf die Ausbreitung mobiler Malware zwischen verschiedenen netzwerkfähigen Geräten abzubilden. Als nachteilig für die Simulation von mobiler Malware mit Hilfe mathematischer Modelle hat sich die fehlende Berücksichtigung räumlicher Parameter erwiesen. In mathematischen Modellen für netzwerkbasierte Simulationen wurden bislang nur zeitliche Verläufe von Ausbreitungen berücksichtigt. Die räumliche Ausbreitung hingegen wurde in den oben genannten Modellen nicht berücksichtigt. Für die Simulation heutiger Anwendungen und Netzinfrastrukturen im Bereich mobiler Systeme und Malware sind diese Ausbreitungscharakteristika hingegen unerlässlich.

Ein anderer Ansatz sind sog. agentenbasierte Simulationen. Hierbei werden reale Eigenschaften von Objekten und Personen auf Objekte innerhalb der Simulation übertragen, um auf diese Weise eine möglichst reale Verhaltensweise der Objekte miteinander zu bewirken. In vielen wissenschaftlichen Arbeiten [18, 57] wurden auch in diesem Bereich viele verschiedene Ansätze und Erweiterungen beschrieben und evaluiert. Im Vergleich zu den oben genannten mathematischen Modellen wurden in agentenbasierten Simulationen bereits räumliche Ausbreitungen bei der Simulation berücksichtigt. Hierfür wurden zum Zeitpunkt der Veröffentlichung des im Rahmen dieser Arbeit entwickelten Mobile Security & Privacy Simulator [24] aber nur rudimentäre Parameter, wie eine homogene Benutzergruppe, ausschließlich zufällige Bewegungen im Raum bzw. auf einer Ebene berücksichtigt. Ebenso wird in Simulationen von Infektionsszenarien stets von einer unmittelbaren Infektion ausgegangen. Dies trifft auf die allermeisten Angriffsszenarien im Bereich mobiler Malware nicht zu und galt somit zusätzlich als Ausschlusskriterium für die entsprechenden Simulationsframeworks.

Eine Ausnahme stellt der Siafu Simulator [39] dar. In Siafu können Straßenzüge und Städte als Basis für ein Bewegungsmodell dienen. Ebenso kann das Verhalten der einzelnen Agenten programmatisch gesteuert werden und erlaubt

auch bereits rudimentäre Interaktionen mit dem zugrundeliegenden Weltmodell. Der größte Nachteil dieses Simulationsframeworks ist, dass das grundlegende Weltmodell in Form von Rastergrafiken erzeugt und eingebunden werden muss. Aus diesem Grund ist das Einbinden echten Kartenmaterials ohne weiteres nicht möglich. Weitere Nachteile dieser Lösung sind die schlechte Skalierbarkeit und die begrenzten Möglichkeiten, die Bewegungen der Agenten zu modellieren. Die Simulation von mehreren Tausend Agenten, wie man sie in Szenarien im Rahmen dieser Dissertation betrachten muss, lässt die derzeitige Implementierung des Frameworks nicht zu. Auch die nur über Umwege oder eine Reimplementierung mögliche Einbindung besonderer Orte, an denen sich Agenten aufhalten können, erlauben die Nutzung des Siafu Frameworks für die im Rahmen dieser Arbeit durchzuführenden Simulationen nicht.

Mascetti et al. [40] haben in ihrer Arbeit gezeigt, dass die Bewegungsmuster und das Verhalten der Agenten einen erheblichen Einfluss auf der Verbreitung mobiler Malware haben. Sie haben die einfachen, auf Zufallsentscheidungen basierenden Bewegungsmodelle, wie sie in aktuellen Infektionsmodellen zur Analyse von mobiler Malware eingesetzt werden, mit einem kartenbasierten Bewegungsmodell verglichen, in dem sich Agenten ausschließlich entlang realer Straßen bewegen können. Hierbei haben sie deutlich messbare Unterschiede zwischen den Bewegungsmodellen feststellen können.

In ihrer Arbeit nutzen Wang et al.[70] Bewegungsdaten von Mobilfunkanbietern, also die Aufzeichnung der Positionen von Benutzern. Diese werden durch die Mobilfunkanbieter durch die Funkzellen ermittelt, in denen das Handy eines Benutzers angemeldet war. Datenpunkte werden hierbei immer dann aufgezeichnet, wenn ein Benutzer eine SMS oder einen Anruf erhält oder absetzt. Auf diese Weise kann die ungefähre, also auf mehrere hundert Meter genaue, Position eines Benutzers ermittelt werden. Wang et al. nutzen die durch die Mobilfunkanbieter gewonnenen Positionsdaten um möglichst realistische Bewegungsmodelle für ihre Untersuchung zur Ausbreitung mobiler Viren zu nutzen. Dieser Ansatz ist für Untersuchungen im Rahmen dieser Arbeit nicht geeignet. Bei der Ausbreitung mobiler Viren, die nicht auf eine zugrundeliegende Netzwerkinfrastruktur setzen, sind die Übertragungsentfernungen gering. Um realistische Bewegungsmuster für die Simulation derartiger Ausbreitungen zu erhalten, sind wesentlich genauere Positionsdaten erforderlich. Hierbei sind

Abstände von wenigen Metern und Zeitintervalle von wenigen Sekunden zu betrachten. Diese Genauigkeit kann durch die Aufzeichnungen von Mobilfunkprovidern nicht geliefert werden, weswegen derartige Quellen für Simulationen im Rahmen dieser Arbeit ausscheiden.
Neben mathematischen Modellen und agentenbasierten Systemen erscheint die Verwendung von MANET Simulatoren zunächst sinnvoll. MANETs sind Netzwerke bestehend aus miteinander kommunizierenden mobilen Geräten, die sich spontan miteinander verbinden ohne die Verwendung weiterer bestehender Infrastruktur. Bei Geräten die sich in Reichweite zueinander befinden, findet eine P2P-Kommunikation statt. Mit Hilfe von MANET Simulationsframeworks wie UDel können realitätsnah urbane Funknetzwerke simuliert werden und erlaubten Kim et al. [35] in ihrer Arbeit die Entwicklung eines Mobilitätsmodells, welches viele Eigenschaften umfasst, die eine realitätsnahe Bewegung auszeichnen. Hierzu zählen neben einem dreidimensionalen Modell und entsprechend modellierten, mehrstöckigen Gebäuden auch die Simulation von Aktivitäten basierend auf Forschungen zur Zeitnutzung und des Zeitmanagements. Auch das dynamische Gruppieren von Agenten und Geschwindigkeitsänderungen und -anpassungen sind Teil dieses Frameworks. Das Ergebnis des beschriebenen Modells sind Bewegungsdaten mobiler Geräte / Benutzer, die in weiteren Simulationen genutzt werden können. Für Simulationen im Rahmen dieser Arbeit sind diese Ergebnisse leider nur bedingt brauchbar, da nicht nur die Interaktion zwischen Agenten wichtig für diese Arbeit ist, sondern auch ein dynamisches, von Interaktionen abhängiges, nicht vorherbestimmtes Handeln der Agenten. Dies ist mit Hilfe von zuvor berechneten und simulierten Bewegungsdaten nicht möglich.

3.1.3 Evil Twins und Schutzmechanismen

In diesem Bereich sind vergleichsweise wenig verwandte wissenschaftliche Arbeiten zu finden. Die Gefahren, die aus derartigen Angriffen entstehen und in anderen wissenschaftlichen Arbeiten vorgestellte Schutzmaßnahmen werden im Folgenden dargestellt und diskutiert.
Bauer et al. haben in einer frühen Arbeit [2] gezeigt, wie Benutzer, die Verwendung von gebräuchlichen und häufig vorkommenden SSIDs machen, von einem Angreifer dazu gebracht werden können, sich mit einem Evil Twin Access

Point zu verbinden. Sie schlagen eine Erkennungsstrategie vor, die auf Basis der SSIDs von umgebenden Funknetzwerken arbeitet. In ihrer Arbeit haben sie den von ihnen vorgeschlagenen Ansatz nicht mit Hilfe von echten Nutzerdaten evaluiert, was den Einsatz des Systems im Alltag in Frage stellt. Im Rahmen dieser Dissertation wird darüber hinaus zu einem späteren Zeitpunkt erläutert, warum die Erkennung alleinig mit Hilfe der SSIDs umliegender Funknetzwerke nicht ausreichend für eine zuverlässige Erkennung ist. Die Arbeit von Gonzales et al. [20] erweitert den Ansatz um eine zusätzliche Verifikation der Signalstärken der umliegenden Access Points. Auch hier wurden keinerlei Evaluationen mit Hilfe realer Daten durchgeführt. Auch diese Erweiterung zur Erkennung von Evil Twin Access Points musste im Rahmen dieser Arbeit als unzulänglich bewertet werden. In Experimenten im Rahmen dieser Dissertation wurde festgestellt, dass die Signalstärke zu sehr schwankt, um sie vielversprechend als zuverlässige Variable für einen Erkennungsalgorithmus einzusetzen.

Roth et al. schlagen in einer ihrer Arbeiten [54] ein Authentifizierungsverfahren für Access Points vor. In ihrem Ansatz nutzen sie Lichtsequenzen, welche auf Bildschirmen nahe des Access Points angezeigt werden, um die Authentizität des zu verbindenden Access Points unter Beweis zu stellen. Kindberg et al. [36] greifen auf ein ähnliches Konzept zurück und nutzen ebenfalls Anzeigen, die in der Nähe der Access Point montiert werden müssen. Sie verwenden zur Authentifizierung eine Adaption des Interlock Protokolls [53] von Rivest und Shamir, um einen Sitzungsschlüssel auszuhandeln. Die beiden vorgestellten Lösungen haben im Alltag einige Nachteile. Zunächst einmal müssen die Access Points oder zumindest die nahe an ihnen angebrachten Anzeigen gut für jeden Benutzer sichtbar sein. Dies wird in vielen Fällen nicht möglich sein, was diese Art von Lösungen disqualifiziert. Ein weiterer Nachteil dieses Lösungsansatzes ist, dass für alle zu sichernden Access Points zusätzliche Hardware benötigt wird, die sowohl beschafft, als auch eingerichtet und überwacht werden muss. Gerade für Betreiber großer Hotspot-Infrastrukturen ergeben sich an dieser Stelle immense Kosten. Ein weiterer Nachteil dieser Lösungen liegt in der notwendigen Interaktion des Benutzers. Der Benutzer muss bei jeder Verbindung mit einem entsprechend ausgestatteten Access Points mit seinem mobilen Gerät interagieren, was nicht zur Akzeptanz eines solchen Systems beitragen dürfte. In einer Feldstudie im Rahmen dieser Dissertation konnte gezeigt werden,

dass sich Mobiltelefone automatisch mehrfach innerhalb kurzer Zeitintervalle in Funknetzwerke einbuchen. Dies geschieht in vielen Fällen ohne jegliche Nutzerinteraktion. Beim Einsatz des beschriebenen Systems müsste in diesen Fällen entweder komplett auf die Verwendung des Funknetzwerks verzichtet werden, oder aber die Benutzer müssen für jede einzelne Verbindung mit dem Endgerät interagieren. Beide Varianten sind für den Benutzer nicht zumutbar.
Betrachtet man Systeme, die nicht der initialen Authentifizierung der Access Points dienen, sondern vielmehr dem Erkennen von bösartigen Evil Twins, so findet man ebenfalls verwandte und relevante wissenschaftliche Arbeiten. So verfolgen Song et al. in ihrer Arbeit [58] zwei unterschiedliche Ansätze zur Erkennung. Der erste Ansatz nutzt die sog. Interpacket Arrival Times (IAT), also die Zeit, die zwischen zwei ankommenden Paketen vergeht. Song et al. gehen davon aus, dass bereits in der Vergangenheit Informationen gesammelt und Statistiken zur Verbindung mit einem Access Point gesammelt wurden. Diese werden durch das von ihnen vorgeschlagene Erkennungssystem mit aktuell gemessenen Werten verglichen. Der vorher durchzuführende Lernprozess ist notwendig, da stark schwankende Faktoren wie die Signalstärke und die Sättigung eines Funkkanals einen direkten Einfluss auf die IAT haben. In einem weiteren Ansatz betrachten Song et al. ebenfalls die IATs zwischen dem Client und einem entfernten Server. Die Tatsache, dass ein Angriff durch das System erst dann erkannt werden kann, wenn bereits Daten ausgetauscht werden, macht den Gebrauch des Gesamtsystems unsicher, da nicht mit Sicherheit ausgeschlossen werden kann, dass während der initialen Erkennungsphase bereits sensible Daten ausgetauscht worden sind. Der Austausch zum Teil sensibler Informationen findet mit Smartphones und Tablets hingegen nahezu durchgängig statt. Die Integration des benötigten Netzwerk Sniffers[3] ist nur durch die Hersteller der Betriebssysteme möglich. Diese müssen entweder die Systeme eigenständig entwickeln oder Entwicklern entsprechende Schnittstellen zur Verfügung stellen. Die benötigte Energie zum dauerhaften Betrieb eines solchen Systems auf einem mobilen Gerät spricht ebenfalls gegen diese Art von Erkennungssystem für den mobilen Bereich.
Das von Mónica et al. vorgeschlagene System zur Erkennung von Evil Twin Access Points [44] nutzt für seine Dienste einen Netzwerkscanner. Zur Erken-

[3]Software zum Aufzeichnen und Analysieren von Datenverkehr

nung sendet das System ein markiertes Datenpaket aus. Gleichzeitig scannt das System durchgängig alle Funkkanäle und sucht nach weiteren Netzwerken in denen das markierte Paket verschickt oder ein Echopaket erhalten wird. Sollte ein solches weiteres Netzwerk existieren so gehen Mónica et al. davon aus, dass der zusätzliche Hop einen bösartigen Access Point darstellt, der von einem Angreifer zwischengeschaltet wurde. Für den Fall, dass der Angreifer ein verschlüsseltes Netzwerk zwischen dem originalen Access Point und dem Evil Twin eingerichtet hat, nutzen Mónica et al. die Paketlänge als Erkennungszeichen und senden entsprechend viele Paketsequenzen aus, um einen Angriff auf diese Weise erkennen zu können. Alle hier beschriebenen Möglichkeiten eignen sich jedoch nicht für den Einsatz im mobilen Bereich. Die Energiekosten für einen durchgängig laufenden Netzwerkscanner zusammen mit einer Echtzeitanalyse würden den Akku so sehr belasten, dass der Betrieb des Systems nicht mehr zweckmäßig erscheint. Darüber hinaus dauert die Erkennung mehr als 20 Sekunden für jede neue Verbindung mit einem Access Point, was im Alltag mit vielen wechselnden Access Points zu langsam für einen sinnvollen Einsatz des Systems erscheint. Das im Rahmen dieser Dissertation entstandene Erkennungssystem ist während des Betriebs nicht nur deutlich energieeffizienter, sondern ist auch in der Lage, bösartige Access Points in deutlich kürzerer Zeit auf dem mobilen Endgerät zu erkennen.

In einer Arbeit von Chen et al. [6] wird ein VPN-artiges Gateway[4] vorgestellt, mit dem sich Benutzer auch in unsicheren Netzwerkumgebungen sicher verbinden können. Sie stellen hiermit eine benutzerorientierte Lösung vor. Im Gegensatz zum Erkennungssystem, welches im Rahmen dieser Dissertation entstanden ist, wird auch für diese Lösung zusätzliche Netzwerkinfrastruktur benötigt, die der Benutzer im Vorfeld einrichten muss. Dies senkt die Akzeptanz eines solchen Systems massiv und wird die Verbreitung und Nutzung erheblich einschränken.

Einen anderen Weg schlagen Conti et al. ein [9]. Sie schlagen zum Schutz sensibler Daten ein System zur Verifizierung von SSL-Verbindungen auf App-Basis vor. Sie schlagen ein System vor, welches bei SSL-Verbindungen von Android-Apps im Hintergrund prüft, ob ein Man-in-the-middle-Angriff vorliegt. Hierfür werden Daten an einen zum System zugehörigen zentralen Server

[4]Komponenten zur Verbindung von zwei oder mehreren Netzwerken

gesendet. Dieser wertet ebenso wie das auf dem Smartphone befindliche System Daten der SSL-Verbindung aus, die im Anschluss miteinander verglichen werden. Das System führt effektiv einen gewünschten MITM-Angriff auf die SSL-Verbindung aus und prüft, ob die betreffende App darauf reagiert. Auf diese Weise können Apps erkannt werden, die anfällig für MITM-Angriffe sind und vor ihnen gewarnt werden. Dieses System stellt lediglich den Schutz bereits vorhandener SSL-Verbindungen sicher. Unverschlüsselte Verbindungen, wie sie bei einem Evil Twin weiterhin auftreten, können hiermit nicht verhindert oder analysiert werden. Ebenso kann nicht das Vorhandensein eines Evil Twins erkannt werden, der lediglich verschlüsselte und unverschlüsselte Verbindungen aufzeichnet, um sie zu einem späteren Zeitpunkt zu analysieren.

3.2 Offene Probleme

Im Folgenden sollen offene Probleme existierender Lösungen dargestellt werden, die es mit Hilfe dieser Dissertation zu lösen gilt. Zunächst soll hierbei auf allgemeine ungelöste Probleme und bislang nicht untersuchte Situationen eingegangen werden. Im zweiten Teil wird beschrieben, welche offenen Probleme in Bezug auf existierende Sicherheitssysteme bestehen.

Bislang existieren keine Untersuchungen, welche die Gefahr von Evil Twins für die Benutzer mobiler Geräte in ihrem Alltag quantifizieren. Insbesondere gibt es keine Analysen, die aufzeigen, wie groß und verbreitet die Gefahr für mobile Geräte mit ihren Standardeinstellungen ist. Durch die Erweiterung eines Evil Twin Angriffs um die Mobilität dieser Geräte ergeben sich entsprechend neue Ausbreitungsverläufe, die es ebenfalls zu untersuchen gilt. Für diese Art der Untersuchungen sind Simulationen erforderlich, wie sie bislang ebenfalls nicht existieren. Vorhandene Ausbreitungsmodelle müssen um zusätzliche relevante Eigenschaften erweitert werden, um das Ausbreitungsverhalten mobiler Malware adäquat bewerten zu können.

Ziel bei der Erforschung neuer Sicherheitssysteme ist es, dem Benutzer neue Methoden und Funktionen an die Hand geben zu können, mit denen er sich besser gegen digitale Gefahren schützen kann. Für die in dieser Arbeit vorgestellten Gefahren mobiler Geräte durch Evil Twin Access Points gibt es bislang kein Verfahren, dass es einem Benutzer erlaubt, sich ohne die Verwendung zu-

sätzlicher Hard- und Software vor diesen Gefahren zu schützen. Insbesondere bei der hier untersuchten Gefahr von Evil Twin Access Points bieten heutige mobile Betriebssysteme keine Möglichkeit die Authentizität von offenen, unverschlüsselten Access Points hinreichend zu prüfen. Aus diesem Grund ist es dem Benutzer unmöglich, bereits bekannte und sichere Access Points von bösartigen, durch einen Angreifer installierten zu unterscheiden. Existierende Systeme benötigen ihrerseits zusätzliche Anpassungen der Infrastruktur oder weitere Systeme für ihren Betrieb. Dies ist hinsichtlich der Kosten nicht zielführend. Aber auch der Betrieb von Sicherheitssystemen auf mobilen Geräten zieht neue Anforderungen nach sich, die ebenfalls im Rahmen dieser Dissertation untersucht werden. Die Evaluation derartiger Sicherheitssysteme stellt eine weitere Herausforderung dar, die es im Rahmen dieser Arbeit zu lösen gilt. Während die beschriebenen, existierenden Systeme in vielen Fällen nicht hinreichend evaluiert wurden, soll das im Rahmen dieser Arbeit entstandene System mit Hilfe von Echtweltdaten hinsichtlich seiner Erkennungsleistung und Benutzerfreundlichkeit untersucht werden.

Kapitel 4

Sicherheitsanalyse von Hotspot-Umgebungen

4.1 Verschlüsselung öffentlicher Hotspots

Im Jahr 1997 wurde der erste Standard 802.11 für kabellose Funknetzwerke (WLAN) verabschiedet [29]. Er enthielt neben den zahlreichen Grundeigenschaften heutiger WLAN-Infrastrukturen unter anderem auch den Sicherheitsstandard *Wired Equivalent Privacy* (WEP). Durch die Nutzung von WEP in WLAN-Umgebungen sollte die gleiche Privatsphäre im Funkverkehr sichergestellt werden, wie man sie bereits aus kabelgebundenen Umgebungen kannte. Zur Verschlüsselung wurde ein aus heutiger Sicht zu simpler Ansatz mit dem RC4-Algorithmus gewählt. Während schon einige Jahre zuvor erste Versuche den WEP-Standard zu knacken erfolgreich verliefen, ist die Sicherheit spätestens mit der Veröffentlichung eines Papers von Darmstädter Forschern im Jahr 2007 und einer entsprechenden prototypischen Implementierung eines Werkzeugs gefallen. In ihrer Arbeit [67] beschreiben Tews et al. wie sie das zur Verschlüsselung verwendete Kennwort in weniger als 60 Sekunden mit Hilfe eines aktiven Angriffs auf den entsprechenden Router berechnen können. Spätestens zu diesem Zeitpunkt war die Sicherheit aller durch WEP verschlüsselten Netzwerke gefährdet.

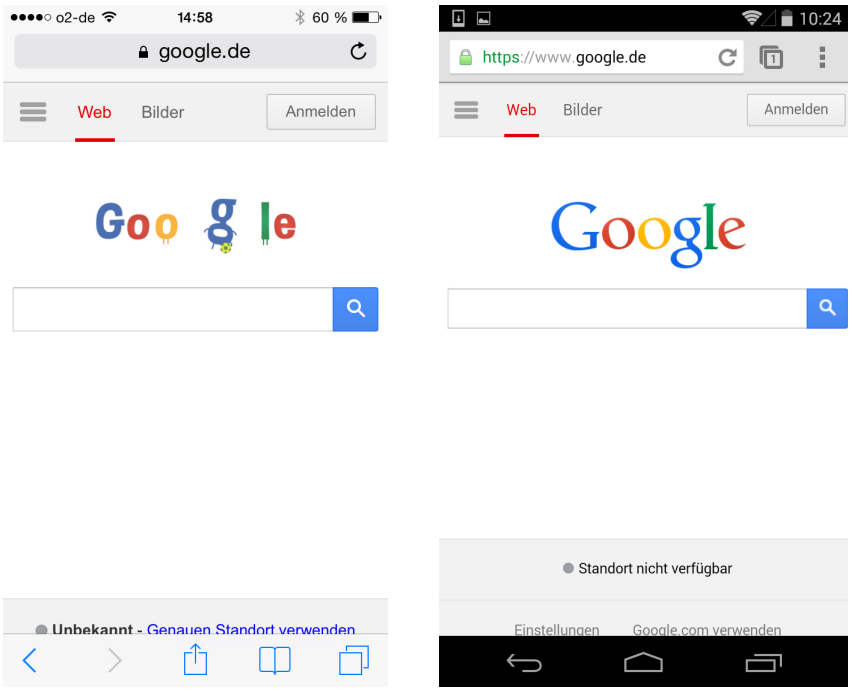
In der Zwischenzeit hatte sich bereits mehr und mehr der Standard WPA durchgesetzt. WPA setzt auf WEP auf und verbessert es durch gezielt hinzugefügte Sicherheitsvorkehrungen. Hierzu zählen unter anderem dynamisch

erzeugte Schlüssel auf Basis des *Temporary Key Integrity Protocol* (TKIP) und Per-Packet Key Mixing, bei dem nicht ein und derselbe Schlüssel für den gesamten zu verschlüsselnden Datenverkehr verwendet, sondern ein neuer Schlüssel für jedes zu versendende Datenpaket erzeugt wird. Diese Aspekte sind unter dem Namen des Robust Security Networks (RSN) im Standard definiert [29]. Ebenso war es mit WPA nun auch möglich RADIUS-Server zur Authentifizierung von Benutzern einzubinden und den Zugang zum WLAN nur bei erfolgreicher Authentifizierung zu gewähren. Doch auch dieser Standard galt als zunehmend unsicher, weswegen mit der Entwicklung und Implementierung des Standards WPA2 begonnen wurde. WPA2 basierte nicht mehr auf WEP, sondern auf dem *Advanced Encryption Standard* (AES).

Da heutige WLAN-Router im privaten und auch im industriellen Umfeld in den allermeisten Fällen mit WPA2 vorkonfiguriert sind, sind sie ohne weiteres Zutun zunächst einmal gegen das Mitlesen des Netzwerkverkehrs durch Dritte gesichert. Trotzdem sind öffentliche Hotspots auch heutzutage meist komplett unverschlüsselt und bergen entsprechende Gefahren für die Anwender.

Ein Grund für die fehlende Verschlüsselung ist sicherlich die hierfür notwendige zusätzliche Infrastruktur. Eine Verschlüsselungslösung mit Hilfe eines zuvor verteilten Schlüssels, welcher für alle Teilnehmer identisch ist, ist für diese Szenarien naturgemäß ungeeignet. Stattdessen muss durch die entsprechenden Anbieter eine AAA-Infrastruktur bereitgestellt werden, die jeden einzelnen Nutzer authentifizieren und autorisieren kann. Dies bedeutet erhebliche Mehrkosten auf Anbieterseite. Ebenso müssen alle potentiellen Benutzer der Hotspots im Besitz ihrer gültigen Zugangsdaten sein. An dieser Stelle kommt der zweite, ebenfalls wichtige Faktor der Benutzbarkeit des Gesamtsystems zum Tragen. Die Verwendung unverschlüsselter Netzwerke bietet dem Kunden ein maximales Maß an Einfachheit und Barrierefreiheit. Diese Barrierefreiheit entsteht in hierbei allerdings auf Kosten der Sicherheit. Offensichtlich ist der einfache Zugang zu den Hotspots in den Augen der Anbieter ein wichtigerer Faktor, als die Sicherheit. Während die Anmeldung durch Benutzername und Kennwort in Captive Portals verschlüsselt durchgeführt wird, empfehlen Anbieter großer Hotspot-Infrastrukturen, wie zum Beispiel Mobilfunkprovider für die weitere sichere Nutzung des Netzzugangs auf VPN-basierte Lösungen zu setzen oder sich nur auf SSL-geschützten Seiten zu bewegen. Hiermit geben die

Anbieter die unzulängliche Sicherheit dieses Teils ihrer Infrastruktur implizit zu und verweisen lediglich auf Hilfsmittel und Ausweichlösungen. Durch dieses Verfahren entstehen für den Endbenutzer Schwierigkeiten und Gefahren, die er selbst nicht sieht und einschätzen kann. So wird ein durchschnittlicher Endnutzer durch fehlende Kenntnis nicht in der Lage sein, sich einen VPN-Endpunkt einzurichten oder zu mieten. Ebenso ist die Einrichtung auf mobilen Geräten, wie auch auf Laptops in vielen Fällen für Laien zu kompliziert gestaltet.



(a) (b)

Abbildung 4.1: Darstellung SSL-geschützter Verbindungen unter iOS 7.1.1 (a) und in Android 4.4.3 (b)

Auch der Hinweis, ausschließlich SSL-gesicherte Verbindungen zu nutzen, kann durch den Endbenutzer nicht umgesetzt werden. In den standardmäßig zum System gehörenden und auch in den beliebtesten sonstig verfügbaren Browsern wird der Schutzstatus der aktuellen Verbindung in der Statusleiste angezeigt. Abbildung 4.1 zeigt die jeweiligen Systembrowser von iOS und Android. Beide zeigen die Absicherung der Verbindung durch Schloss-Symbole an, Android stellt die Sicherheit zusätzlich farblich dar.

Deutlich schwieriger wird es für den Benutzer bei der Nutzung von beliebigen anderen Apps auf seinem Smartphone. Ob diese Apps über SSL-gesicherte Verbindungen verschlüsselt kommunizieren oder ganz und gar unverschlüsselt, ist für den Benutzer an keiner Stelle ersichtlich. Entsprechend hat der Benutzer keine Möglichkeit für ihn gefährliche Apps in derartigen Umgebungen zu erkennen und von der Nutzung abzusehen.

4.2 Captive Portals in Hotspotumgebungen

Um den Zugang zu Hotspots für alle Geräte zu ermöglichen, greifen Anbieter in den meisten Fällen auf die *Universal Access Method* (UAM) zurück. Hierbei erhält jedes Gerät, welches sich in das unverschlüsselte Netzwerk einbucht, eine IP-Adresse und ein Standardgateway per DHCP zugewiesen. Bei einer beliebigen HTTP-Anfrage wird diese zunächst auf eine Login-Seite des Anbieters, das sog. Captive Portal umgeleitet. Im Folgenden wird die Funktionsweise eines Captive Portals und die mit ihm einhergehende Bedrohung für Hotspot-Benutzer dargestellt und erläutert.
Wie bereits in Abschnitt 2.2 beschrieben, handelt es sich bei einem Captive Portal um ein System, das in einem lokalen Netzwerk die Eingabe von Berechtigungsnachweisen ermöglicht und mit Hilfe von nachgelagerten Systemen zur Authentifizierung und Autorisierung den Zugang zu einem Netzwerk erlauben, beschränken oder gänzlich verwehren kann. Der genaue Ablauf der Nutzung und die Funktionsweise der beteiligten Komponenten sollen in folgendem Ablauf verdeutlicht werden.

1. Der erste Schritt besteht aus der Assoziierung eines Benutzers mit dem zum Captive Portal gehörenden Access Point. In den meisten Fällen sind die Hotspot-Netzwerk unverschlüsselt und es werden keine Zugangsdaten für die Verbindung mit dem Funknetzwerk selbst benötigt. Der Verbindungsaufbau ist hierbei prinzipiell jedem Gerät erlaubt.

2. Im zweiten Schritt wird dem mobilen Gerät des Benutzers per DHCP eine IP-Adresse und ein Standard-Gateway zugeteilt.

3. Als nächstes muss der Benutzer auf die Login-Seite des Captive Portals navigieren werden. Da die Adresse einem neuen Benutzer nicht bekannt

sein kann, wird jede Anfrage an einen beliebigen Host zunächst auf die besagte Login-Seite umgeleitet. Um dem Benutzer von Laptops und mobilen Geräten den Zugang zu diesen Netzwerken so schnell wie möglich zu erlauben, senden viele Betriebssysteme automatisch nach der Verbindung mit einem Funknetzwerk eine Anfrage an einen bekannten Server, um eine überprüfbare Antwort zu erhalten. Auf diese Weise prüfen die Betriebssysteme, ob bereits eine Verbindung zum Internet besteht, die von weiteren Systemdiensten genutzt werden kann (beispielsweise zum Abruf von E-Mails oder andere anstehende Synchronisierungen). Im Fall eines Captive Portals wird diese Anfrage auf die Login-Seite umgeleitet. Die von den Betriebssystemen erwartete Antwort wird entsprechend nicht zurückgeliefert, was auf eine eingeschränkte oder nicht vorhandene Verbindung zum Internet schließen lässt. In diesem Fall zeigen die Betriebssysteme die zurückgelieferte Antwort dem Benutzer. Auf diese Weise gelangt der Benutzer in vielen Fällen ohne die manuelle Eingabe einer URL oder die Nutzung einer App zur besagten Login-Seite des Captive Portals.

4. Auf dieser Seite gibt der Benutzer je nach Art des Captive Portals entweder seine seine Zugangsdaten ein oder er bestätigt mit den allgemeinen Geschäftsbedingungen des Anbieters einverstanden zu sein. Die eingegeben Daten werden zurück an das Captive Portal übermittelt.

5. Sollten im vorangegangenen Schritt Zugangsdaten eingegeben worden sein, so müssen sie in diesem weiteren Schritt verifiziert werden. Hierfür stellt das Captive Portal eine Anfrage an einen AAA-Server. Dieser überprüft die Angaben und liefert die Entscheidung zum Captive Portal zurück.

6. Im Falle einer positiven Entscheidung oder bei der reinen Bestätigung von allgemeinen Geschäftsbedingungen wird die Sitzung gestartet. Das Gerät des Benutzers erhält Zugang zum Internet und die Anfrage aus Schritt 3 wird an den entsprechenden Server weitergeleitet. Bis zur Abmeldung, zur Trennung vom Funknetzwerk oder bis zum Ende eines gebuchten Zeitkontingents bleibt die Sitzung erhalten.

Die Gefahr beim oben beschriebenen Verfahren besteht zum einen darin, dass die gesamte Kontrolle über die im Captive Portal angezeigten Internetseiten beim Wireless Internet Service Provider (WISP) liegen. Zum anderen ist die Authentifizierung des Access Points gegenüber dem Benutzerendgerät bis heute nicht möglich.
Zunächst soll hier auf die Login-Seite des Captive Portal eingegangen werden. Bei der Verbindung zu einem Access Point kann durch den Benutzer nicht sichergestellt werden, dass es sich beim verbundenen Access Point tatsächlich um den gewünschten handelt. Verbindet sich ein Benutzer mit einem ihm unbekannten Hotspot und ist auf diesem ein Captive Portal aktiviert bzw. nachgeschaltet, so wird automatisch (insbesondere ohne jegliche Nutzerinteraktion) die entsprechende Internetseite angezeigt. Da diese Seite vollkommen in der Hand des Access Point-Betreibers liegt, sind an dieser Stelle verschiedene Angriffe denkbar.

Phishing Die als Captive Portal angezeigte Seite könnte eine täuschend echt nachempfundene Login-Seite eines bekannten Anbieters sein. Ein argloser Benutzer würde sich in vielen Fällen durch eine der Originalseite nachempfundenen Darstellung blenden lassen und seine Zugangsdaten für den Zugriff auf das Internet eingeben. Diese können von einem Angreifer mitgeschnitten und für betrügerische Zwecke eingesetzt werden.

Einschleusen von Malware Über die angezeigte Seite des Captive Portals könnte Malware verteilt werden, die sich beim Besuch der Seite automatisch auf dem entsprechenden Endgerät installiert oder weitere Aktionen durchführt. Auf die Gefahren, die eine solche Lücke mit sich bringt wird eingehend im folgenden Abschnitt 4.3 eingegangen.

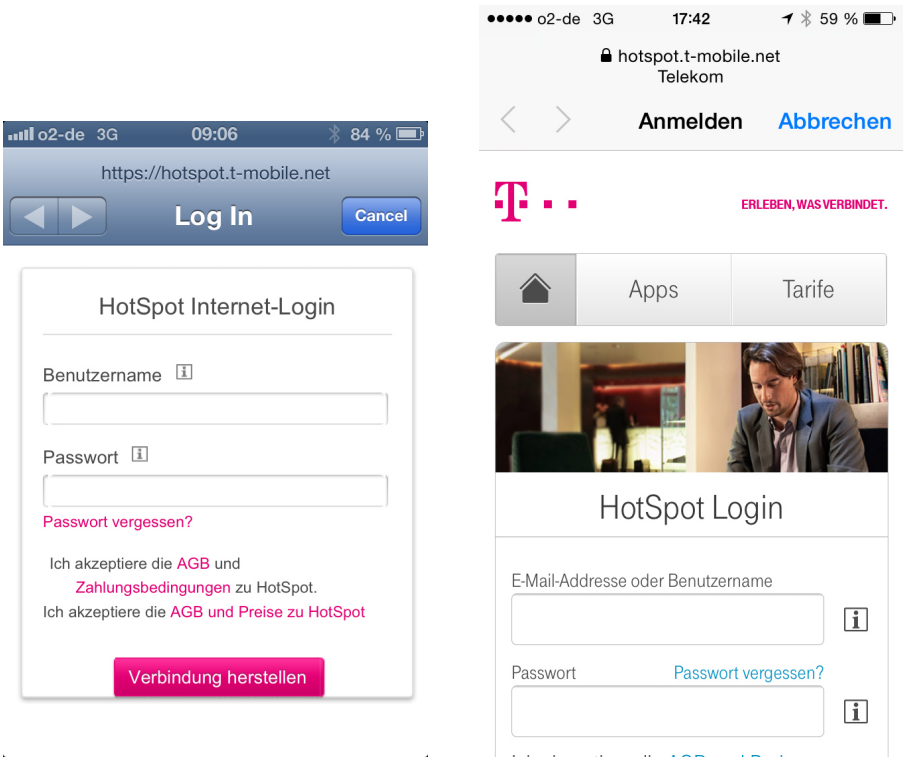
Manipulation von Daten Beim Abruf von Internetseiten, die nicht zusätzlich durch eine Verschlüsselung gesichert sind, kann ein Angreifer sowohl die gesendeten (möglicherweise sensiblen) Daten, als auch die zurückgelieferten Internetseiten manipulieren. Insbesondere können Eingaben des Benutzers, die an einen Server im Internet gesendet werden durch einen Angreifer gelöscht, verändert oder vervielfältigt werden.

Als weitere Gefahr wurde bereits einleitend die fehlende Möglichkeit der Authentifizierung eines Access Points gegenüber des Benutzergeräts genannt. Heut-

zutage besteht keine zweckmäßige Möglichkeit für einen Access Point seine Echtheit gegenüber dem Benutzer bzw. gegenüber des Endgeräts des Benutzers unter Beweis zu stellen. Viele Anbieter behelfen sich mit einer SSL-verschlüsselten Verbindung zum Captive Portal. Doch auch dieses Vorgehen ist nicht praktikabel für eine sichere Authentifizierung von Access Points. Zum einen können arglose Benutzer durch die vermeintlich sichere Verbindung in die Irre geführt werden. Die Browserkomponente, die zur Anzeige des Captive Portals genutzt wird, verifiziert lediglich die Gültigkeit des vom Webserver bzw. Captive Portal ausgelieferten Zertifikats und zeigt das entsprechende Ergebnis an. Bedauerlicherweise besteht keinerlei logische oder technische Verknüpfung zwischen der SSID des Access Points (anhand derer die Verbindung zum Access Point aufgebaut wurde) und dem Hostnamen bzw. dem Fully Qualified Domain Name (FQDN) des Webservers, der die Captive Portal Seite ausliefert. Zusammenfassend kann festgehalten werden, dass eine hinreichende Authentifizierung von Access Points auf diesem Wege mit den technischen Mitteln, die von den Anbietern eingesetzt werden, nicht sichergestellt werden kann.

Vom technischen Standpunkt gesehen, wäre eine ausreichende Absicherung der Hotspots durchaus denkbar. Das in Privat- und Geschäftsnetzwerken heutzutage standardmäßig eingesetzte Verschlüsselungsverfahren WPA2 bietet die Möglichkeit, sensible Informationen der Benutzer eines Hotspots zu schützen. Die Anbieter von Hotspots verfolgen in den meisten Fällen finanzielle Interessen. Sowohl die Akquise neuer Kunden, als auch die Sicherung und Steigerung der Zufriedenheit von Bestandskunden sind die langfristigen Ziele dieser Unternehmen. Genau diese Ziele widersprechen der technisch sicheren Möglichkeit, das gesamte Hotspot-Netzwerk über WPA2 abzusichern.

Für Bestandskunden wäre die Absicherung durchaus denkbar. Die entsprechenden Kunden würden Zugangsdaten erhalten, die sie bei einer ersten Verbindung mit einem an das Netzwerk angeschlossenen Hotspot eingeben müssten. Die Zugangsdaten könnten mit Hilfe eines RADIUS-Servers überprüft werden, um so dem Kunden Zugang zum Netzwerk zu gewähren oder diesen zu verwehren, wenn keine korrekten Zugangsdaten angegeben wurden. Wäre das vorhandene Hotspot-Netzwerk mit WPA2 gesichert, so würde für potentielle Neukunden keine Möglichkeit bestehen, sich ein Zugangsticket für das Internet über das Hotspot-Netzwerk selbst zu kaufen. Ebenso wäre die Eingabe von Ticketcodes



(a) (b)

Abbildung 4.2: Darstellung eines Captive Portals der Deutschen Telekom (links iOS 6 / rechts iOS 8)

auf der Captive Portal-Seite für Gelegenheitsnutzer nicht mehr möglich. Die einzige Möglichkeiten die genannten Nachteile dieser Lösung zu kompensieren besteht in der Bereitstellung eines zweiten unverschlüsselten Netzwerks. Auf diese Weise wären zwar Bestandskunden mit fest vorgegebenen Zugangsdaten weitestgehend gegen Angriffe gesichert. Für alle anderen Kunden würde sich durch die Bereitstellung zusätzlicher Netzwerke keine Änderung der Bedrohungslage ergeben. Die Kosten für die Bereitstellung und den Betrieb weiterer Netzwerke, wie auch der verhältnismäßig geringe Nutzen bewegt die Anbieter dazu, sich gegen diese Lösung zu entscheiden. Sie setzen weiterhin auf gänzlich unverschlüsselte Funknetzwerke und geben ihren Kunden die Empfehlung, sich nur auf verschlüsselten Seiten zu bewegen, solange sie das entsprechende Hotspot-Netzwerk nutzen.

4.3 Evil Twins

Der Begriff des Evil Twins stammt ursprünglich aus der Literatur, wo der Begriff für eine Person verwandt wurde, die äußerlich dem Original entsprechen konnte, aber gegensätzliche moralische Grundsätze hat. Der Begriff des Evil Twin hat sich wegen seiner Bedeutungsnähe zu den oben angesprochenen Funknetzwerken auch im Bereich der Funknetzwerke durchgesetzt, wo er für einen Access Point / Hotspot steht, der mit der gleichen SSID wie ein anderer betrieben wird. Im Gegensatz zum originalen Hotspot wird der Evil Twin ausschließlich zum Zweck des Ausspionierens oder des Angriffs auf Benutzer des vermeintlich echten Hotspots eingesetzt. Im folgenden Abschnitt wird dargestellt, welche Informationen einem Angreifer vorliegen müssen und wie ein derartiger Angriff mit Hilfe aktueller Technik durchgeführt werden kann. Eine Erweiterung des Evil Twin-Angriffs um eine mobile Komponenten wird im zweiten Abschnitt beschrieben.

4.3.1 Allgemeines Konzept

Während derartige Angriffe vor einiger Zeit nur mit komplizierten Software-Werkzeugen durchführbar waren, sind mittlerweile aufeinander abgestimmte Hard- und Softwarekomponenten verfügbar, die diese Angriffe auch für Laien ermöglichen. Ein Beispiel für ein solches Produkt ist der *WiFi Pineapple*

Mark V [23] der Firma Hak5 LLC. Dieser bietet mit Hilfe einer übersichtlichen Oberfläche die Möglichkeit, verschiedenste Angriffe durchzuführen. Die Kombination aus Hard- und Software ist dank seiner Einfachheit durchaus auch durch Laien bedienbar.
Im Folgenden sollen die in Abschnitt 2.4 vorgestellten Bedingungen, die für einen Angriff relevanten sind, dahingehend untersucht werden, welche Hürden diese Anforderungen für einen Angreifer mit heutiger Technik darstellen.

Wahl der SSID

Zunächst entscheidend ist die Wahl des für den speziellen Angriff richtigen Netzwerknamens. Abhängig davon, ob es sich um einen allgemeinen oder einen gezielten Angriff auf eine Einzelperson handelt, müssen die entsprechenden Namen gewählt werden. Bei allgemeinen Angriffen eignen sich hierzu insbesondere die Netzwerknamen großer, bekannter Hotspots oder anderer weitverbreiteter und beliebter Netzwerke. Für zielgerichtete Angriffe auf Einzelpersonen oder kleine Gruppen gibt es weitere Möglichkeiten, an geeignete SSIDs zu gelangen. Hat ein Angreifer vor einen Evil Twin-Angriff zu starten, so muss er zunächst geeignete SSIDs finden, mit denen sich die mobilen Endgeräte der Zielpersonen automatisch verbinden werden. Eine Möglichkeit für einen Angreifer, um an diese SSIDs in der aktuellen Umgebung zu gelangen ist es, den aktuellen WLAN-Verkehr kurzzeitig mitzuschneiden und die entsprechenden SSIDs aus *Probe Request*- und *Probe Response*-Paketen zu extrahieren. Probe Requests werden kontinuierlich von mobilen Geräten ausgesendet, um bekannte Access Points und WLAN-Router möglichst schnell in der Umgebung zu finden. Die Access Points beantworten die an sie gerichteten Probe Requests mit entsprechenden Probe Responses. Dass das kontinuierliche Aussenden der Probe Requests ein Problem bzgl. der Privatsphäre und der Sicherheit darstellen kann beschreibt auch die Electronic Frontier Foundation in einem Artikel vom Juli 2014 [51].
In Abbildung 4.3 sind zwei verschiedene Arten von Probe Requests dargestellt. In Abbildung 4.3A sendet ein Client Probe Requests mit einem leeren SSID-Feld aus. Auf diese Weise versucht das Gerät alle in der Umgebung befindlichen Stationen zu erreichen. In Abbildung 4.3B hingegen sendet der Client die gesuchte SSID im Rahmen des Probe Requests mit. Der Client will sich ent-

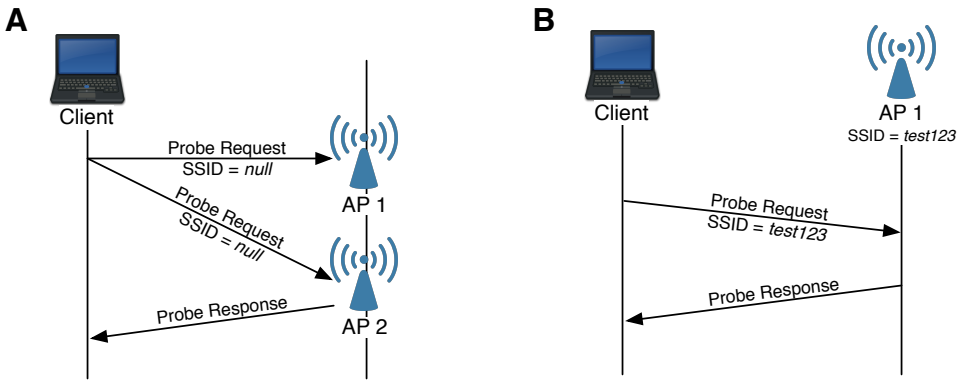


Abbildung 4.3: Probe Requests im Verbindungsprozess mit WLAN Access Points. Dargestellt ist der Ablauf der Kommunikation zwischen einem Client und einem Access Point. In Teil A ohne eine im Probe Request enthaltene SSID. In Teil B ist eine SSID enthalten.

sprechend mit einem spezifischen Netzwerk verbinden und sucht hierfür einen Access Point mit dem entsprechenden Namen.
Bei allen im Rahmen dieser Kommunikation verschickten Nachrichten und Datenpakete handelt es sich um Broadcast-Nachrichten im geteilten Medium. Somit sind diese Daten für alle Stationen empfangbar, die sich in der Nähe des Senders befinden. Die ausgesendeten Probe Requests mit SSIDs kann ein Angreifer gezielt dazu nutzen, einen Access Point zu imitieren, welcher aktuell von einem in der Nähe befindlichen Gerät aktiv gesucht wird. Darüber hinaus kann ein Angreifer auch dem Problem begegnen, dass sich eines der gesuchten Netzwerke tatsächlich in der Umgebung befindet. Hierfür muss er lediglich zusätzlich die von Access Points ausgesendeten Probe Responses und Beacon Frames auswerten und auf die Verwendung der hier enthaltenen SSIDs verzichten.
Es gibt entsprechend für beide Arten von Angriffen, seien es allgemeine oder auch zielgerichtete, einfache Möglichkeiten an die entsprechenden Netzwerknamen zu gelangen.

Örtliche Bedingungen

Bei der Wahl des Ortes, an dem ein Angriff durchgeführt werden soll, gilt es für den Angreifer zu beachten, dass der Betrieb seines Evil Twin Access Points in räumlicher Nähe zum originalen Hotspot mit Risiken verbunden ist. So können

entsprechend installierte Evil Twin Access Points beispielsweise durch den Betreiber des originalen Access Points entdeckt und entsprechende Maßnahmen eingeleitet werden. Die Ausführung derartiger Angriffe lässt sich deutlich einfacher an Orten durchführen, die weit genug vom Original-Access Point entfernt sind. Mobile Betriebssysteme prüfen bei der Verbindung zu einem Funknetzwerk die eigene Position nicht, wodurch derartige Angriffe möglich werden.

Fälschen der BSSID

Um die eigene Identität zu schützen oder originale Access Points besser zu imitieren muss ein Angreifer in der Lage sein, die BSSID, also die MAC-Adresse seiner WLAN-Schnittstelle zu fälschen. Dies stellt heutzutage kein Problem mehr dar. Viele Netzwerkkarten und -chips erlauben es dem Benutzer, die MAC-Adresse direkt aus entsprechenden Software-Werkzeugen auf einen beliebigen Wert zu setzen. Auf diese Weise kann ein Angreifer zum einen seine eigene Identität verbergen und gleichzeitig zu fälschende Access Points detailgetreu nachbilden.

4.3.2 Mobile Evil Twins

Eine Erweiterung des Evil Twin-Angriffs ist der Mobile Evil Twin-Angriff, welcher im Rahmen dieser Dissertation entwickelt und prototypisch implementiert wurde. Die Grundidee hierbei ist es, den stationären Charakter eines Evil Twin Angriffs mit herkömmlicher Technik hin zu einem mobilen Angriffsvektor für Smartphones und Tablets zu erweitern. Auf diese Weise sind Angriffe an jedem beliebigen Ort denkbar. Zusätzlich zu diesem Vorteil wird im folgenden Kapitel 5 eine Malware für iOS vorgestellt, welche die hier genannten Unzulänglichkeiten dazu ausnutzt, sich selbstständig auf weitere Geräte mit iOS-Betriebssystem zu verbreiten.

4.4 Standardverhalten mobiler Geräte

Bevor im folgenden Kapitel der detaillierte Ablauf der Infektion der Malware beschrieben wird, soll an dieser Stelle noch eine wesentliche Eigenschaft moderner Smartphones beschrieben und diskutiert werden. Hierbei handelt es sich

um die Eigenschaft, sich automatisch mit bekannten Funknetzwerken wieder zu verbinden. Verbindet sich ein Benutzer unter Android oder auch iOS mit einem Funknetzwerk – sei es ein öffentlicher Hotspot oder ein privater WLAN-Router – so werden die entsprechenden Zugangsdaten vom Gerät eingefordert. Diese werden für zukünftige Anmeldungen sicher vom Betriebssystem gespeichert. Kommt der Benutzer ein weiteres Mal in den Einzugsbereich des Funknetzwerks, so wird sich das Smartphone völlig automatisch ohne jegliche Nutzerinteraktion erneut mit dem bekannten Funknetzwerk verbinden. Während es in früheren Versionen von iPhone OS / iOS und Android eine Einstellungsmöglichkeit gab, die dieses Verhalten unterdrückte und entsprechend keine automatische Verbindung herstellte, sind diese Optionen in aktuellen Versionen der Betriebssysteme nicht mehr enthalten. Es hat sich aufgrund seiner Benutzerfreundlichkeit als Standardverhalten durchgesetzt und ist aus diesen Gründen nicht mehr zu deaktivieren.

Leider öffnet die offensichtlich so benutzerfreundliche Lösung ein Tor für potentielle Angreifer. Das Problem bei einer Wiederverbindung zu einem bekannten Funknetzwerk ist der Mangel an Informationen, der durch das System herangezogen werden kann, um ein bekanntes von einem vermeintlich bekannten unterscheiden zu können. Die einzige Information die für diese Entscheidung herangezogen wird ist der Name des Netzwerks (SSID) und im Falle eines verschlüsselten WLANs der Typ der Verschlüsselung. Gerade in Bezug auf Hotspot-Umgebungen, in denen zum Großteil unverschlüsselte Funknetzwerke zum Einsatz kommen, ist dieses Verhalten äußerst problematisch und bietet einem Angreifer die Möglichkeit sehr leicht Man-in-the-middle-Angriffe mit Hilfe von Evil Twins durchzuführen. Das grundlegende Problem besteht also im Mangel an Kontextinformationen, die das mobile Endgerät für die Auswertung und Bestimmung der Echtheit eines Funknetzwerks heranziehen kann.

4.5 Zusammenfassung

In diesem Kapitel wurde die Sicherheit heutiger Hotspot-Umgebungen beleuchtet und bewertet. Im ersten Teil wurde dargestellt, wie heutige Hotspots funktionieren und wo die Gefahren für die Benutzer lauern. Ein wichtiger Aspekt für die Sicherheit dieser Infrastrukturen ist die fehlende Verschlüs-

selung, die bis heute auch von großen Hotspot-Anbietern aus Gründen der Benutzerfreundlichkeit nicht angeboten wird. Des Weiteren wurde die Funktionsweise der Universal Access Method (UAM) beleuchtet und dargestellt wie diese in Verbindung mit Captive Portals in heutigen Hotspot-Umgebungen zu einem Sicherheitsrisiko werden können.
Auch die Gefahr durch Evil Twin-Angriffe wurde beschrieben und es wurde gezeigt, dass diese heutzutage mit einfachen Mitteln und zum Teil bereits durch Laien durchgeführt werden können. Ebenso wurden von Smartphones und Tablets bekannte Funktionen, wie zum Beispiel das automatische Wiederverbinden mit bekannten WLAN-Netzwerken beschrieben und erläutert, welche Probleme dieses Verhalten nach sich ziehen kann. Die Erweiterung des Evil Twin-Angriffs um eine mobile Komponenten in Kombination mit dem beschriebenen Standardverhalten heutiger mobiler Betriebssysteme hat die Entwicklung einer prototypischen mobilen Malware ermöglicht, wie sie im folgenden Kapitel beschrieben wird.

Kapitel 5

Der Mobile Evil Twin Angriff

In diesem Kapitel wird der Mobile Evil Twin Angriff beschrieben, der im Rahmen dieser Dissertation geplant und prototypisch entwickelt worden ist. Zunächst wird beschrieben, auf welche Weise ein Angreifer an die für den Angriff benötigten, verbreiteten SSIDs gelangt. Im Anschluss daran folgt die Beschreibung der Implementierung des Mobile Evil Twin-Konzepts. Es wird gezeigt, dass durch die Anpassung eines bestehenden Jailbreaks die automatisierte Infektion neuer Geräte durch ein Wirtsgerät ohne das Zutun des Benutzers möglich ist. Auch das neu infizierte Gerät verwandelt sich nach erfolgter Installation der Malware-Software in ein Wirtsgerät und kann seinerseits neue mobile Geräte infizieren. Teile des Inhalts dieses Kapitels basieren auf einer wissenschaftlichen Veröffentlichung [64] im Rahmen der *International Conference on Cryptology and Network Security.*

Ort	Start	Dauer	Total	Mgmt	PrReq	PrResp	SSIDs
Bahnhof	12:01	72:42	1.369.281	329.540	27.663	48.158	673
Extrablatt	13:30	68:54	688.176	197.064	12.232	64.566	293
Kröpcke	14:47	61:36	1.089.132	617.276	45.252	146.702	1.845

Tabelle 5.1: Ergebnisse der Untersuchung von WLAN-Management Informationen in öffentlichen Bereichen

5.1 Suche geeigneter SSIDs

Wie bereits in Abschnitt 4.3.1 beschrieben besteht für einen Angreifer, die Möglichkeit SSIDs von nahen mobilen Geräten aufzuzeichnen, um sie in einem späteren Schritt für einen Evil Twin-Angriff zu nutzen. Um nachzuweisen, dass diese Art der Angriffsvorbereitung in heutigen öffentlichen Umgebungen denkbar ist und um einschätzen zu können, wie groß die Gefahr für Anwender ist, wurde im Rahmen einer Feldstudie an verschiedenen Orten ermittelt, wie viele der entsprechenden WLAN-Management-Daten frei empfangbar sind und aufgezeichnet werden können. Hierfür wurde am Mittwoch, den 08.10.2014, zur Mittagszeit an drei verschiedenen belebten Orten in Hannover ein Laptop in Betrieb genommen und der umgebende Netzwerkverkehr aufgezeichnet. Als Laptop kam hierbei ein Apple MacBook Pro 15" zum Einsatz. Zum Aufzeichnen des Netzwerkverkehrs wurde das freie Netzwerkanalyseprogramm Wireshark [73] in seiner derzeit aktuellen Version 1.12.1 verwendet. In Tabelle 5.1 sind neben den exakten Uhrzeiten der verschiedenen Messungen auch die folgenden Zahlen zu finden. *Total* beschreibt die Gesamtzahl aller aufgezeichneten Pakete. *Mgmt* ist die Gesamtzahl an WLAN-Management-Paketen. Die Spalten *PrReq* und *PrResp* stellen die Anzahl der empfangenen Probe Request und Probe Responses dar. In der Spalte *SSIDs* ist die Anzahl einmaliger SSIDs enthalten, die für die hier angestellten Betrachtungen maßgeblich ist.

Als Standort für die erste Messung M1 wurde der vordere Bereich des Hauptbahnhofs Hannover gewählt. Die Messung begann an diesem Ort um 12:01 Uhr und dauerte knapp 73 Minuten. In dieser Zeit wurden insgesamt 27.663 Probe Requests aufgezeichnet. Insgesamt konnten hierbei 673 verschiedene SSIDs empfangen und gespeichert werden. Als zweiten Ort wurde das stark frequen-

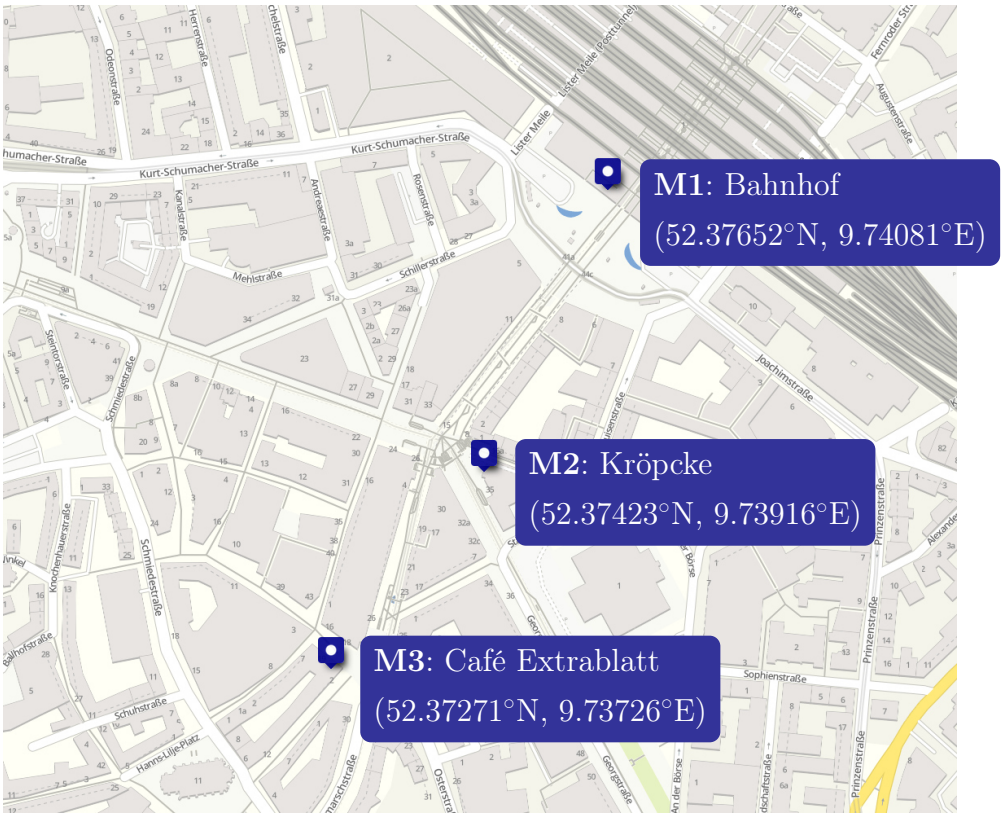


Abbildung 5.1: Karte Hannovers mit markierten Orten, an denen die Messungen durchgeführt wurden.

tierte Café Extrablatt in der Innenstadt von Hannover ausgewählt. Hier konnten bei Messung M2 in knapp 69 Minuten 12.232 Probe Requests erfasst und daraus 293 verschiedene SSIDs extrahiert werden. Als dritten und letzten Messpunkt M3 wurde der sehr belebte und oftmals als Treffpunkt genutzte freie Platz Kröpcke gewählt, der sich in der Fußgängerzone von Hannovers Innenstadt befindet. An diesem Ort konnten 45.252 Probe Requests mitgeschnitten werden. An diesem Ort konnten insgesamt 1.845 verschiedene SSIDs aus den vorhandenen Probe Requests extrahiert werden.

Die am Kröpcke deutlich höhere Erfolgsquote im Vergleich zu beiden anderen Messungen ist darauf zurückzuführen, dass dieser Ort einerseits sehr belebt ist. Andererseits halten sich viele Menschen für einen längeren Zeitpunkt in der unmittelbaren Umgebung auf, treffen Bekannte oder besuchen das direkt neben dem Platz befindliche Café. Durch die Kombination aus einem hoch frequentierten Ort mit einer zum Teil langen Verweildauer ergeben sich die entsprechenden Zahlen. Zieht man als Vergleich den Bahnhof heran, so sind hier zwar mehr Menschen unterwegs. Diese haben aber im Allgemeinen eine deutlich geringere Verweildauer. Aus diesen Gründen ist die Erfolgsquote an diesem Ort geringer. Da mobile Geräte in unregelmäßigen Abständen Probe Requests mit SSIDs aussenden ist die Verweildauer ein wichtiger Faktor bei den obigen Betrachtungen.

Zusammenfassend kann festgehalten werden, dass an allen drei Orten innerhalb kürzester Zeit mehr SSIDs gesammelt werden konnten, als für einen realen Angriff benötigt werden. Diese Art der Sammlung potentiell angreifbarer SSIDs wirkt entsprechend äußerst vielversprechend für einen Angreifer. Diese zu verhindern würde auf Seiten der mobilen Betriebssysteme mit einer Minderung der Benutzbarkeit einhergehen. Ein zum Schutz notwendiges Verzichten auf Probe Requests ist von daher nicht zu erwarten, was das Schutzbedürfnis des Benutzers an anderer Stelle noch wichtiger erscheinen lässt.

Die im Rahmen dieser Studie gesammelten Daten können aufgrund der Vielzahl personenbezogener Daten nicht öffentlich zur Verfügung gestellt werden. Sie befinden sich in der Abgabeversion dieser Dissertation im beigefügten elektronischen Verzeichnis im Ordner *SSID_Studie*.

5.2 Implementierungssystem

Das Betriebssystem, auf dem die im Folgenden beschriebene Malware läuft und sich verbreitet ist iOS in Version 4.3.3. Der Grund für die Wahl dieser Version war zum einen, dass dies die derzeitig aktuelle Version war, als mit der Entwicklung und Forschung im Bereich mobiler Evil Twins im Rahmen dieser Arbeit begonnen wurde. Zum anderen bot sich diese Version an, weil für sie ein öffentlicher Jailbreak verfügbar war, der in dieser Arbeit modifiziert, erweitert und im Anschluss für die Infektionsroutinen genutzt wurde.

5.3 Ablauf einer Infektion

Um eine möglichst große Verbreitung von Malware innerhalb kürzester Zeit zu ermöglichen, müssen unter anderem die folgenden Anforderungen erfüllt werden:

- Große Anzahl potentieller Opfergeräte
- Automatische Infektion neuer Geräte
- Schnelle Übertragungswege für alle benötigten Daten
- Nutzung aktiver Übertragungsmechanismen, die möglichst vielen mobilen Geräten zur Verfügung stehen

In den folgenden Abschnitten wird auf die einzelnen Anforderungen eingegangen und beschrieben, wie die im Rahmen dieser Dissertation erarbeitete Malware-Lösung diese Anforderungen zu erfüllen versucht. Zunächst wird im Folgenden der Ablauf der initialen Infektion eines Opfergeräts dargestellt. Da weitere Infektionen durch bereits infizierte Geräte auf die gleiche Art und Weise erfolgen ist der folgende Ablauf nicht nur die Beschreibung des initialen Infekts, sondern auch aller weiteren. In Abbildung 5.2 ist schematisch der gesamte Ablauf dargestellt. Dieser teilt sich in vier Einzelschritte auf, die nachfolgend beschrieben werden. Während Schritt 2 & 3 Bestandteil des bestehenden Jailbreaks sind und im Rahmen dieser Arbeit nicht verändert wurden, sind die Schritte 1 & 4 die für diese Dissertation maßgeblichen. Diese Schritte stellen den eigentlichen Entwurf der mobilen Malware dar, der im Rahmen dieser Dissertation geplant, entwickelt und untersucht wurde.

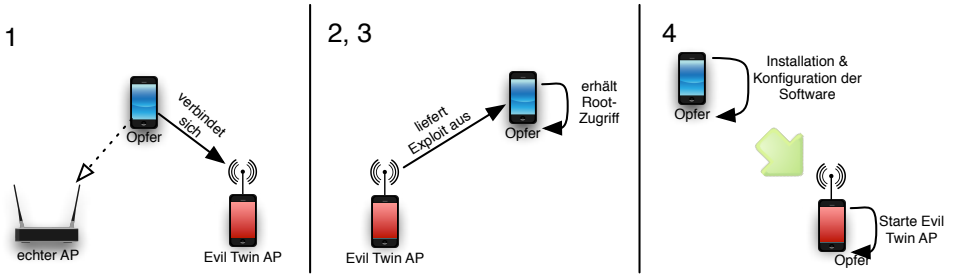


Abbildung 5.2: Ablauf einer Infektion durch die Malware

Schritt 1: Masquerading & initiale Infektion Im ersten Schritt wird das in Abschnitt 4.4 beschriebene Feature zur Wiederverbindung bekannter Netze ausgenutzt, um ein Opfergerät in der Umgebung zu finden. Hierfür bedient sich das initiale Infektionsgerät einer SSID, die auf möglichst vielen Geräten als bekanntes Netz gespeichert ist. Hierfür bieten sich besonders die SSIDs großer Hotspot-Anbieter, Mobilfunkanbieter oder ähnlich weit verbreitete SSIDs an. Alle mobilen Geräte, die sich bereits einmal mit einem Hotspot des Anbieters verbunden haben, werden sich mit dem vermeintlich zum Anbieter gehörenden Hotspot verbinden. Sämtliche Kommunikation mit Servern im Internet wird von diesem Zeitpunkt an über das Wirtsgerät geleitet. Ab hier ist nicht nur das Mitlesen, sondern auch das Manipulieren von Daten möglich, die unverschlüsselt verschickt oder empfangen werden. Wie in einem realen Szenario mit einem Captive Portal als Schnittstelle (siehe Abschnitt 4.2) für die Authentifizierung und Autorisierung werden auch in diesem Fall alle Anfragen des gefangenen Benutzers an einen lokalen Webserver umgeleitet. Das heißt, dass keine der Anfragen ins Internet geroutet werden. Der lokale Webserver befindet sich ebenfalls auf dem Wirtsgerät und verarbeitet alle eingehenden Anfragen. Weitere Details zu diesem Schritt der Infektion sind in Abschnitt 5.4 zu finden. Dieser Schritt ist essentiell für den Gesamtablauf des Evil Twin-Angriffs und ist zusammen mit den in Schritt 4 beschriebenen Maßnahmen im Rahmen Dissertation erarbeitet und entwickelt worden.

Schritt 2: Application Exploit Ist das Gerät des arglosen Benutzers erst einmal mit dem Evil Twin verbunden, so kann die zweite Phase des Angriffs beginnen. Unter iOS wird bei der Verbindung mit einem Hotspot

mit Captive Portal ein Popup-Fenster des MobileSafari (dem Systembrowser) angezeigt. Warum an dieser Stelle ein Popup-Fenster erscheint und bei der Anmeldung in Nicht-Hotspotnetzwerken nicht, wird in Abschnitt 5.4.1 beschrieben. Als Application Exploit wurde eine seinerzeit noch nicht geschlossene Sicherheitslücke einer Komponente des Browsers verwendet. Genauer handelte es sich um eine Sicherheitslücke in der WebKit Engine [71]. Die WebKit Engine ist eine Bibliothek zur Darstellung von Internetseiten, die in den Browsern vieler mobiler Betriebssysteme, wie unter anderem auch Android und iOS, eingesetzt wird. Die Tatsache, dass es sich bei der WebKit Engine um eine Open-Source-Software handelt, die von einer Vielzahl von Firmen entwickelt und vorangetrieben wird, bringt auch einen entscheidenden Nachteil mit sich. Durch die freie Verfügbarkeit des Quellcodes werden immer wieder Sicherheitslücken entdeckt und veröffentlicht. Diese werden zwar sukzessive geschlossen bleiben aber auf den verschiedenen Plattformen und Browsern so lange bestehen, bis der Anbieter ein entsprechendes Update verteilt und der Benutzer dieses installiert. Durch die stetige Weiterentwicklung im Bereich der Rendering Engines und das ständige Bestreben mit neuen Technologien mehr Geschwindigkeit und Effizienz zu erreichen ist davon auszugehen, dass auch in Zukunft viele dieser Schwachstellen nicht nur entstehen, sondern auch gefunden und veröffentlicht werden. Einige der in der Webkit Engine gefundenen Sicherheitslücken erlauben es einem Angreifer, beliebigen Programmcode mit den Rechten des Webbrowsers auf den Geräten auszuführen. Diese Art von Sicherheitslücke, die auch unter dem Begriff *Arbitrary Code Execution* bekannt ist, wird auch in der beschriebenen Malware verwendet. Im Fall des hier verwendeten Jailbreaks resultiert die Möglichkeit beliebigen Code auszuführen aus einem Pufferüberlauf des Stacks.

Die prototypische Implementierung der Malware verwendet als Basis einen Jailbreak der unter dem Namen *Star* bekannt geworden ist. Dieser Jailbreak kann ohne technische Kenntnisse über eine Webseite installiert und ausgeführt werden. Unter der URL `http://www.jailbreakme.com/` konnten Benutzer nach mehrmaligem Bestätigen von Warnungen den Start des Jailbreaks anstoßen.

Schritt 3: Kernel Exploit Durch die Ausnutzung der Schwachstelle in WebKit ist die Ausführung beliebigen Codes mit den Rechten des MobileSafari innerhalb der Sandbox der App möglich. Für die Durchführung des eigentlichen Jailbreaks und die Installation von nicht durch Apple autorisierter Software auf dem Gerät benötigt man Administrationsrechte auf dem Gerät. Mit Hilfe des sogenannten Sandboxings ist es auf den Geräten laufenden Apps nicht möglich auf Daten zuzugreifen, die sich außerhalb ihrer Sandbox befinden. Insbesondere ist der Zugriff auf das Systemdateien nicht möglich.

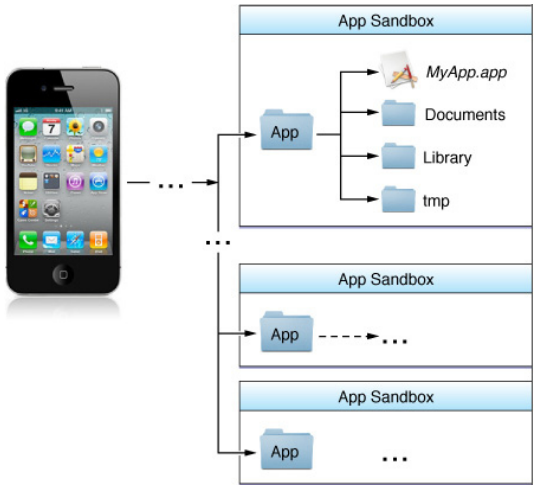


Abbildung 5.3: Sandboxing unter iOS: Jede App hat ausschließlich Zugriff auf ihre eigenen Daten im Dateisystem [31]

Hierfür wird eine weitere, in dieser Version von iOS vorhandene, Schwachstelle genutzt. Durch einen weiteren Pufferüberlauf im privaten IOSurface Framework von iOS werden die benötigten Privilegien auf Betriebssystemebene erlangt und die Installation weiterer Software auf dem Gerät kann erfolgen.

Die Erfahrung zeigt, dass diese Art von Schwachstellen deutlich schwieriger zu finden sind, als jene auf Anwendungsebene. Dies liegt maßgeblich in der Tatsache begründet, dass es sich bei den entsprechenden Systemen zumeist um proprietäre Software handelt, die nicht öffentlich einsehbar ist (Closed-Source-Software). Trotz der Schwierigkeiten beim Auffinden solcher Schwachstellen werden jedes Jahr zahlreiche solcher Schwachstellen durch Sicherheitsforscher, Hacker und andere Gruppie-

rungen gefunden und ausgenutzt. In vielen Fällen wird der Fund einer solchen Schwachstelle nicht öffentlich bekannt gegeben. Stattdessen wird versucht, sich mit dem gefundenen Einfallstor für eine Software bzw. ein Betriebssystem einen finanziellen Vorteil zu verschaffen.

Schritt 4: Übergang in das Evil Twin Malware-Netzwerk Durch den Kernel-Exploit kann im nun folgenden vierten Schritt beliebige Software auf dem Gerät installiert, gestartet und verwaltet werden. Im Rahmen das Jailbreaks wird an dieser Stelle Cydia[1] installiert, ein Paketmanager mit grafischer iOS-Oberfläche zur Installation, Deinstallation und Modifikation von Paketen auf dem iOS-Gerät.

Die im Rahmen dieser Arbeit entstehende Malware hat ausschließlich das Ziel, sich möglichst effizient und schnell zu verbreiten. Da an dieser Stelle nicht nur der Zugriff auf das gesamte Dateisystem möglich ist, sondern auch der Zugriff auf jegliche Kommunikationsschnittstellen wurde bei der Auswahl der Schnittstelle zur Verbreitung auf die WLAN-Schnittstelle gesetzt, die in allen aktuellen Smartphones vorhanden ist. Ebenso sind alle heutigen Smartphones in der Lage, persönliche, mobile Hotspots über ihre WLAN-Schnittstellen bereitzustellen.

Diese Möglichkeit macht sich die im Rahmen dieser Arbeit entwickelte Malware, zunutze. Neben weiteren Paketen wird in diesem Schritt eine Software installiert, die es der Malware erlaubt, beliebige Hotspots zur Verfügung zu stellen. Auf diese Weise kann, wie in Schritt 1 beschrieben, ein neuer Hotspot mit einer viel genutzten SSID erstellt werden. Dieser stellt seinerseits nicht nur den Hotspot zur Verfügung sondern verfolgt den gleichen Ablauf und stellt somit selbst einen Mobile Evil Twin mit allen technischen Voraussetzungen dar.

Durch diese vier Schritte kann sich die Malware zunächst von einem initialen Infektionsgerät verbreiten. Die Malware selbst verbreitet sich hierbei über den modifizierten Jailbreak, der unter anderem auch die Softwarepakete enthält, die sowohl für die Infektion, als auch für die Bereitstellung eines neuen Mobile Evil Twins benötigt werden. Die genauen technischen Details werden im folgenden Abschnitt erklärt und diskutiert.

[1] `https://cydia.saurik.com/`

5.4 Implementierung

In diesem Abschnitt wird zunächst darauf eingegangen, mit Hilfe welcher in iOS vorhandenen Schwachstellen eine automatisierte Infektion neuer Geräte realisierbar ist. In einem weiteren Schritt wird darauf eingegangen, welche Software zur Infektion weiterer Geräte benötigt wird und wie die Komponenten zusammenarbeiten.

5.4.1 Automatisierte Infektion unter iOS

Zunächst soll an dieser Stelle auf die vollautomatische Infektion neuer Geräte eingegangen werden. Für eine effiziente und schnelle Verbreitung einer Malware muss diese in der Lage sein, sich ohne Nutzerinteraktion verteilen zu können. Hierfür wird das bereits beschriebene Feature von iOS verwendet, sich standardmäßig automatisch mit bereits bekannten Netzwerken erneut zu verbinden. Hierbei muss sich keineswegs auf lediglich eine SSID festgelegt werden. Um einen möglichst großen Kreis an mobilen Geräten zu erreichen, sind durchaus auch wechselnde SSIDs denkbar. So wäre auch die Hinterlegung einer Datenbank mit den am häufigsten verwendeten SSIDs denkbar, welche der Hotspot abwechselnd verwendet, wenn eine bestimmte Zeit lang unter der vorherig verwendeten SSID keine neuen Verbindungen mehr aufgebaut wurden.
Jedes Mal wenn sich ein iOS-Gerät mit einem Access Point verbindet testet es seine Internetkonnektivität. Dies geschieht, sobald die Assoziation mit einem WLAN erfolgreich abgeschlossen ist und das Gerät per DHCP oder auf anderem Wege eine IP-Adresse erhalten hat. iOS sendet nun einen HTTP Request an die fest definierte URL `http://www.apple.com/library/test/success.html` und wertet die Antwort aus. Die Antwort des Apple-Servers auf diese Anfragen ist im Folgenden dargestellt:

```
1 <HTML>
2   <HEAD>
3     <TITLE>Success</TITLE>
4   </HEAD>
5   <BODY>Success</BODY>
6 </HTML>
```

Listing 5.1: Antwort des Apple-Servers auf eine erfolgreiche Prüfung der Internetkonnektivität von iOS / Mac OS

> **Anmerkung** Seit iOS 7 verwendet Apple nicht mehr ausschließlich die oben genannte URL. Stattdessen wird eine Verbindung mit einer zufällig aus einer Liste ausgewählten URL aufgebaut. Große Teile dieser Liste sind allerdings bereits im Internet verfügbar. Ebenso sind Workarounds veröffentlicht worden, in denen mit Hilfe des User Agent Strings die entsprechenden Anfragen eindeutig identifiziert werden konnten, um entsprechend reagieren zu können. Diese Änderungen am Verhalten von iOS können den beschriebenen Angriff entsprechend nicht verhindern.

Das Betriebssystem wertet nun die Antwort aus. Es muss sich einerseits um ein HTML-Dokument handeln, andererseits darf sich im validen HTML-Code im `BODY`-Tag ausschließlich die Zeichenkette *Success* befinden. Sollte eine der Bedingungen nicht erfüllt sein, so deutet iOS dies als Hinweis darauf, dass keine oder eine eingeschränkte Internetkonnektivität besteht. Sollte dies der Fall sein, so wird auch bei bestehender Verbindung zu einem Access Point keine weitere Kommunikation über diese Schnittstelle mit dem Internet ausgetauscht werden. Erst wenn die Antwort des Apple Servers die gegebenen Bedingungen erfüllt erkennt iOS die Verbindung an und leitet von nun an den gesamten Datenverkehr über diese Schnittstelle.

Apple versucht mit dieser Methode nicht, die Sicherheit der Verbindung zu verifizieren. Der eigentliche Grund für das o.g. Vorgehen liegt in einer möglichst benutzerfreundlichen Nutzung von Captive Portals begründet. Durch die automatisch im Hintergrund stattfindende Abfrage des Apple-Servers und der entsprechenden Antwort kann iOS entscheiden, ob es sich bei der Verbindung zum Access Point um einen Access Point mit einem Captive Portal handelt oder ob bereits eine Verbindung zum Internet besteht. Der Vorteil an diesem Vorgehen ist, dass es bei erfolgreicher Auswertung der Antwort zu keinerlei Beeinträchtigungen für den Benutzer kommt. Der beschriebene Prozess bleibt dem Benutzer also bei positiver Auswertung gänzlich verborgen.

Sollte die Auswertung hingegen negativ ausfallen, so kann das System davon ausgehen, dass sich zwischen dem iOS-Gerät und dem Apple-Server noch mindestens ein Gerät befindet, welches die Antwort auf die ursprüngliche Anfrage übernimmt oder die ursprüngliche Antwort verändert. In diesem Fall geht iOS vom Vorhandensein eines Captive Portals aus. Dem Benutzer wird daraufhin

ein Popup-Fenster des MobileSafari mit der vom Server erhaltenen Antwort angezeigt. In der Regel wird dies eine Webseite mit einem Anmeldeformular sein, wie sie in Abbildung 4.2 beispielhaft für Hotspots der Deutschen Telekom dargestellt sind. In iOS beschleunigt dieses Verfahren den Zugriff auf Hotspot-Netzwerke durch den initialen Test der Internetkonnektivität. Dem Benutzer wird direkt nach dem beschriebenen Assoziationsprozess die Webseite zur Anmeldung angezeigt, ohne dass der Benutzer selbst aktiv eine Webseite mit dem mobilen Browser ansurfen oder eine App nutzen muss.

Genau dieses Vorgehen birgt eine ernstzunehmende Gefahr. Obwohl iOS durch die fehlerhafte oder fehlende Antwort des Apple-Servers bereits erkannt hat, dass keine Internetkonnektivität besteht und die Antwort nicht der vordefinierten Form entspricht, wird die Antwort der Anfrage ohne Umwege in Form des Popup-Fensters präsentiert.

Es findet an dieser Stelle keinerlei Prüfung des angezeigten Inhalts statt. Die größten Gefahren bei einer solchen, ungeprüften Webseite sind Phishing und die Installation von Malware. Speziell der Phishing-Ansatz ist an dieser Stelle besonders leicht durchführbar. Die Benutzer sind es ohnehin gewohnt und entsprechend darauf konditioniert genau an dieser Stelle ihre Zugangsdaten für den Hotspot-Zugriff einzugeben. Bedauerlicherweise wird an dieser Stelle keinerlei Sorge dafür getragen, dass es sich bei der angezeigten Webseite um eine echte handelt oder um den Nachbau eines Angreifers. Es wird an dieser Stelle nicht einmal eine SSL-geschützte Verbindung vorausgesetzt, was eine Phishing-Attacke überaus vereinfacht.

Während Phishing das Abgreifen von Zugangsdaten unbedarfter Benutzer zum Ziel hat, hat die hier beschriebene *Mobile Evil Twin Attacke* ausschließlich das Ziel, sich in kürzester Zeit möglichst großflächig zu verbreiten. Aus diesem Grund und aufgrund der für diese Malware nicht erwünschten Nutzerinteraktion, wird an dieser Stelle kein Phishing eingesetzt. Stattdessen wird an dieser Stelle eine Internetseite ausgeliefert, die eine präparierte PDF-Datei enthält. Durch die beschriebenen Schwachstellen in der Rendering-Engine wird bei der Anzeige der PDF-Datei der Jailbreak-Prozess gestartet. Durch das automatische Erscheinen dieses Fensters und der direkten Ausnutzung einer Schwachstelle in diesem Fenster kann eine hohe Ausbreitung erreicht werden.

Da die reale Ausbreitung einer Malware im Rahmen dieser Forschungsarbeiten nicht erwünscht ist, wurden relevante Parameter der Ausbreitung gemessen und analysiert, um mögliche Ausbreitungen mit Hilfe eines Simulators untersuchen zu können. Nähere Informationen zum Simulator und zu den durchgeführten Simulation befinden sich im folgenden Kapitel 6.

5.4.2 Implementierung der Infektion

In diesem Abschnitt wird der Infektionsprozess detailliert beschrieben. Ebenso werden alle an der Infektion beteiligten Systeme und Komponenten erläutert und ihre Konfigurationen dargestellt. Die zur Implementierung, Konfiguration und zum Testen genutzten Geräte waren ein iPhone 4, ein iPhone 3GS, ein iPad 1 und ein iPad 2.
Als initialen Infektionshost wurde das iPhone 4 mit der Softwareversion iOS 4.3.3 genutzt. Der initiale Infektionshost wurde dafür zunächst mit einem Jailbreak versehen. Die für die Malware benötigte Software und entsprechende Konfigurationen wurden per Hand auf dem Gerät installiert. Alle anderen Geräte dienten zunächst als Opfergeräte. Auf ihnen war die gleiche Softwareversion iOS 4.3.3 installiert. Darüber hinaus waren keine weiteren Anpassungen an den Geräten selbst oder an der auf ihnen installierten Software notwendig. Dies ist ein wichtiger Aspekt, da bei einer realen Ausbreitung einer solchen mobilen Malware ebenfalls unpräparierte Geräte als Opfergeräte dienen müssen. Insbesondere musste keines der Opfergeräte mit einem Jailbreak versehen sein. Dieser wird im Laufe des Infektionsprozesses durch die Mobile Evil Twin Malware automatisiert durchgeführt. Der Infektionsprozess wurde auf allen o.g. Plattformen verifiziert und getestet.
Zum Betrieb eines Evil Twin Access Points wird sowohl auf dem initialen Infektionsgerät, als auch auf allen weiteren angesteckten Geräten eine Software für die Bereitstellung eines Hotspots benötigt. Die Wahl einer geeigneten SSID ist, wie in Abschnitt 2.4 beschrieben entscheidend für einen Evil Twin. Die installierte Hotspot-Software wurde in dieser prototypischen Implementierung dahingehend konfiguriert einen echten WLAN Hotspot der Deutschen Telekom nachzuahmen. Die entsprechende SSID lautet *Telekom*.
Im Rahmen der Erstellung eines Hotspots sollte zunächst die mit iOS 4 eingeführte Betriebssystem-eigene Funktion des *Personal Hotspots* zum Einsatz

kommen. Diese dient dem Zweck die eigene Mobilfunkdatenverbindung anderen WLAN-fähigen Geräten in der direkten Umgebung (wie beispielsweise Laptops) zur Verfügung zu stellen. Die Nutzung der Betriebssystem-eigenen Funktion hat den Vorteil, dass keine weitere Software für den Betrieb des Hotspots benötigt und entsprechend bei einer Infektion übertragen werden muss. Dieser Ansatz hat sich im Laufe der Entwicklung der Mobile Evil Twin Malware als ungeeignet herausgestellt und musste aus mehreren Gründen verworfen werden:

- Apple hat sich bei der Implementierung des *Personal Hotspots* dazu entschlossen, nur sichere Verbindungen über den Hotspot zuzulassen. Insbesondere ist die Bereitstellung eines unverschlüsselten WLANs, mit welchem man sich ohne die Eingabe eines Kennworts verbinden kann, nicht möglich. Gerade die Verwendung viel verwendeter, öffentlicher und unverschlüsselter WLANs macht den Mobile Evil Twin Angriff hingegen so gefährlich.

- Ein weiteres Problem mit der System-eigenen Lösung ergab sich aus den fehlenden Möglichkeiten, die Hotspot-Funktion aus einem Skript heraus von der Konsole zu starten, zu steuern oder zu beenden. Entsprechende Werkzeuge für den Einsatz in Skripten waren und sind nicht vorhanden.

Aus diesen Gründen konnte die iOS-eigene Funktion für die prototypische Implementierung nicht zum Einsatz kommen. Stattdessen fiel die Wahl auf die Software MyWi 4[2]. Sie bietet neben den Features des iOS eigenen Personal Hotspots entscheidende zusätzliche Möglichkeiten, wie den für uns unverzichtbaren Betrieb unverschlüsselter Funknetzwerke und die Konfiguration über plist-Dateien. Ein integrierter optionaler DHCP-Server regelt die Adressvergabe in den bereitgestellten Netzwerken. Es handelt sich bei MyWi um eine Software für iOS-Geräte, die mit einem Jailbreak versehen sind. Aber auch diese Softwarekomponente ermöglicht es von Hause aus nicht, über ein Skript angesteuert zu werden. Mit Hilfe des Reverse Engineering Werkzeugs und Disassemblers IDA Pro [3] war es möglich einen undokumentierte Parameter des

[2] `http://intelliborn.com/mywi.html`

[3] `https://www.hex-rays.com/products/ida/index.shtml`

Steuerungsprogramms zu finden, der genau diese Funktion ermöglicht. Mit Hilfe der Parameter ist sowohl das Starten und Stoppen des Hotspots, als auch die Anpassung der Konfiguration zur Laufzeit über ein Skript möglich.

In Bezug auf die Größe der zu übertragenden Malware wandelt sich der o.g. Vorteil der Betriebssystem-eigenen Lösung in einen Nachteil. Die Verwendung von Software Dritter setzt die zusätzliche Übertragung der Software vom Wirts- zum Opfergerät voraus. MyWi 4 hat als Paket eine Größe von 1,8 MB. Dies ist ein erheblicher Anstieg des Transfervolumens bei einer Infektion. Es ist hierbei zu bedenken, dass diese Daten bei jeder Infektion eines weiteren Opfergeräts übertragen werden müssen. Neben der erhöhten Belastung für den Akku ist ebenfalls die zusätzlich benötigte Zeit für die Übertragung der Daten zu berücksichtigen. Die Ergebnisse der Untersuchung genau dieser Fragestellungen finden sich in Abschnitt 5.5.

Obwohl die Verwendung der Drittanbieter-Software MyWi die o.g. Nachteile mit sich bringt wurde sie für die Entwicklung des Prototypen genutzt. Es geht bei der Entwicklung des Prototypen in erster Linie um die Überprüfung und den Beweis der Machbarkeit der konstruierten Malware. Ziel der prototypischen Implementierung ist es insbesondere nicht, die Malware in Bezug auf ihre Verbreitungsgeschwindigkeit weiter zu optimieren. Aus diesem Grund können die Nachteile, die sich aus der Lösung ergeben, hingenommen werden. Zu beachten ist hierbei aber, dass die veränderte Größe der Malware sich auf die Verbreitung der Malware auswirken kann. Deswegen müssen die geänderten Rahmenbedingungen bei der Simulation (siehe Kapitel 6) Beachtung finden.

Neben der Hotspot-Software selbst wird zur Auslieferung der Malware ein Webserver benötigt. Für den Einsatz auf einem Smartphone bietet sich die Software *lighttpd* an. Der Webserver ist ebenfalls als Paket über Cydia verfügbar. Er hat eine kleine Größe und verlängert die Ladezeit des fertigen Malware-Pakets dadurch nur unerheblich. Der Server lauscht auf dem für HTTP-Verkehr standardmäßigen Port 80. Auf diesem Port wird ebenfalls die oben beschriebene Anfrage des iOS-Betriebssystems gesendet, die bei nicht erfolgreicher Antwort das Popup-Browserfenster öffnet. Der Webserver stellt nun die präparierte Webseite bereit, welche den Exploit ausnutzt und die entsprechende Payload an das Opfer ausliefert.

Sobald ein Opfergerät sich mit dem vermeintlich bekannten Hotspot verbunden hat beginnt die Arbeit der Malware. Der in MyWi integrierte DHCP-Server vergibt dem Opfer eine IP-Adresse. Sobald die Assoziation abgeschlossen ist, steht der lokale Webserver bereit, um die Malware an das Opfer auszuliefern. Damit Anfragen an beliebige Hosts auf den lokalen Webserver umgeleitet werden, wurde im Vorfeld zusätzlich die in Listing 5.2 dargestellten Regeln für den systemeigenen Paketfilter *pf* hinzugefügt.

```
1  nat on pdp_ip0 inet
2    from 192.168.40.0/24 to any
3    -> (pdp_ip0:0) static-port
4  no nat on ap0 inet
5    from 192.168.40.1 to 192.168.40.0/24
6
7  rdr on ap0 proto tcp
8    from 192.168.40.0/24 to any port 80
9    -> 127.0.0.1 port 80
10
11 pass on pdp_ip0
12   from any to any flags S/SA keep state
13 pass on ap0 all flags any
14   xkeep state (source-track global) rtable 4
```

Listing 5.2: PF Firewallregeln, welche die Umleitung jeglicher Anfragen auf Port 80 auf den lokalen Webserver umleiten (siehe Zeile 7-9)

Direkt nach der erfolgreichen Assoziation des Opfers mit dem Evil Twin Hotspot wird wie beschrieben ein Apple-Server kontaktiert, um die Internetkonnektivität zu testen. Diese Anfrage wird auf den lokalen Webserver umgeleitet und durch ein Skript zunächst negativ beantwortet. Die Antwort ist in Listing 5.3 dargestellt.

```
1 <HTML>
2   <HEAD>
3     <TITLE>No Success</TITLE>
4   </HEAD>
5   <BODY>No Success, thus we will see the popup window</BODY>
6 </HTML>
```

Listing 5.3: Antwort des Evil Twin Webservers auf die erste Anfrage nach einer Assoziation

Sobald iOS eine nicht positive Antwort durch ein Captive Portal erhält, wird das Popup-Fenster geöffnet. Des Weiteren wird die URL `http://www.apple.com` aufgerufen und das Ergebnis im Popup-Fenster angezeigt. Auch diese Antwort wird durch den lokalen Webserver ausgeliefert. An dieser Stelle beginnt der eigentliche Infektionsprozess der Mobile Evil Twin Malware. Die nun ausgelieferte Webseite enthält ein verstecktes iframe-Element, in welchem eine PDF-Datei eingebunden ist. Die Webseite zusammen mit dem iframe-Element werden durch den MobileSafari Browser gerendert. Hier ist der Jailbreak-Exploit für die iOS-eigene CoreGraphics Bibliothek enthalten. Der Star-Jailbreak, der seinerzeit durch die Webseite `http://jailbreakme.com` ausgeliefert wurde verschafft sich zunächst Administrationsprivilegien und nutzt diese im Anschluss zum Download und zur Installation des eigentlichen Jailbreaks. Die hierbei im System installierte Software sorgt nicht nur dafür, dass der Benutzer zukünftig Administrationsprivilegien in Anspruch nehmen kann, sie sorgt ebenfalls dafür, dass bei einem erneuten Start des Smartphones die entsprechenden Routinen des Jailbreaks erneut durchgeführt werden (siehe *Arten des Jailbreaks* in Abschnitt 2.6.1).

Durch die Verwendung eines bestehenden Jailbreaks hat sich bei der Implementierung des Prototypen die Schwierigkeit ergeben, dass die für den Jailbreak notwendigen weiteren Daten innerhalb des Popup-Fenster nicht vom entsprechenden Server heruntergeladen werden konnten. Dies liegt in der Tatsache begründet, dass iOS die vom lokalen Webserver gelieferte negative Antwort als eine fehlende Verbindung zum Internet interpretiert. Aus diesem Grund schlägt der durch den initialen Schritt des Jailbreaks durchgeführte Download der weiteren Jailbreak-Daten fehl. Dies lässt sich durch den erneuten Aufruf einer beliebigen URL beheben. Das auf dem Webserver befindliche Skript, welches den von Apple betriebenen Server simuliert kann bei einer erneuten Anfrage, mit einem positiven Ergebnis antworten und ermöglicht auf diese Weise den Download weiterer Daten durch den Jailbreak-Prozess. In Bezug auf die Malware stellt diese Problematik kein konzeptionelles Problem dar. Bei der Entwicklung einer *echten* Malware, könnte ein zweiter Verbindungsversuch zu einem der Jailbreak-Server mit einer kleinen zeitlichen Verzögerung gestartet werden. Es soll mit Hilfe des Prototypen lediglich die Machbarkeit eines derartigen Angriffs gezeigt werden. Da es sich beim Jailbreak selbst nicht um eine

quelloffene Software handelt, wurde auf die Anpassung des Jailbreak-Codes an dieser Stelle verzichtet. Die Problematik und der Verzicht auf eine Anpassung an dieser Stelle schränkt die prinzipielle Funktionsweise und somit die folgenden Betrachtungen nicht ein.
Die während des Jailbreak-Prozesses nachzuladenden Daten lassen sich in drei Bereiche gliedern:

Initiales Dateisystem: Das initiale Dateisystem beinhaltet neben der benötigten Verzeichnisstruktur ebenso das `/bin`-Verzeichnis, in welchem weitere Softwarewerkzeuge für den weiteren Ablauf des Jailbreaks und den späteren Betrieb des Linux-Paketsystems benötigt werden. Eines der Werkzeuge, welches im Folgenden auch für die Entwicklung der Malware benötigt wird ist der für das Debian-Paketsystem entwickelte *Debian Package Manager* `dpkg` [4]. Mit seiner Hilfe können im laufenden System Softwarepakete installiert, verwaltet und entfernt werden.

Softwarepaket: Das ebenfalls durch den Jailbreak installierte Softwarepaket umfasst neben weiteren Werkzeugen und Programmen auch den für iOS entwickelten grafischen Paketmanager Cydia, der es dem Endanwender ermöglicht auf einfache Art und Weise das Paketsystem auf seinem iOS-Gerät zu verwalten.

Dynamische Bibliothek: Die dritte und letzte Komponente ist eine dynamische Bibliothek, welche die Installation der o.g. Komponenten startet und den Installationsprozess steuert und überwacht.

Der Jailbreak-Prozess bezieht die o.g. Dateien über eine unverschlüsselte HTTP-Verbindung mit dem Host `http://www.jailbreakme.com`. Anfragen an diesen Host werden ebenfalls über den lokalen Paketfilter `pf` an den Webserver des Wirtsgerätes umgeleitet. Hier liegen die für die Malware angepassten Pakete und Dateien bereit zur Installation auf dem Opfergerät.
Um die für den Betrieb der Mobile Evil Twin Malware benötigten Daten und Softwarepakete zusätzlich zu den Jailbreak-eigenen zu installieren musste ein Weg gefunden werden, diese Daten bereits während des Jailbreak-Prozesses

[4] `https://alioth.debian.org/projects/dpkg`

mit zu installieren. Prinzipiell gibt es zwei Möglichkeiten die benötigten Anwendungen, Skripte und Konfigurationsdateien in den Jailbreak zu integrieren. Als einfache und naheliegende Möglichkeit besteht die Möglichkeit sie in das initiale Dateisystem einzupflegen, so dass sie beim Mounten des Dateisystems bereits vorhanden sind und entsprechend direkt nach dem Jailbreak ausgeführt und genutzt werden können. Diese Möglichkeit war nicht realisierbar, weil die Integrität des Dateisystems vor der Kopie durch den Jailbreak-Prozess geprüft wird. Eine Änderung des Dateisystems hat den sofortigen Abbruch des Jailbreaks zur Folge. Die benötigten Komponenten müssen entsprechend auf einem anderen Weg installiert werden. Die zweite Möglichkeit, die letztendlich zum Erfolg führte, war es, die Daten in das Debian-Paket zu integrieren. Für dieses Paket wird keine Integritätsprüfung durchgeführt.
Das zu modifizierende Debian Software-Paket besteht aus den folgenden drei Dateien die zusammengefasst das `deb`-Paket ergeben [15]:

- `debian-binary`
- `control.tar.gz`
- `data.tar.gz`

Die Datei `debian-binary` ist eine Textdatei. Sie enthält die Versionsnummer des verwendeten Paketformats. In diesem Fall die Version 2.0. Die Datei `control.tar.gz` ist ein Archiv, das verschiedene Skripte zur Installation, Deinstallation und Verwaltung des Pakets enthält. Unter anderem sind hier die Skripte `preinst`, `postinst`, `prerm` und `postrm` enthalten, die vor und nach der Installation oder der Entfernung des Softwarepakets automatisiert durch das Paketsystem ausgeführt werden. Diese Skripte werden in der Malware genutzt um die in der Datei `data.tar.gz` enthaltenen Programme und Konfigurationen an die entsprechenden Stellen im Dateisystem zu kopieren und weitere Startskripte für die Malware auszuführen. Die Datei `data.tar.gz` enthält die eigentlichen Programm- und Konfigurationsdateien. Sie sind in der nach der Installation gewünschten Verzeichnisstruktur abgelegt und werden durch das Paketmanagement an die entsprechenden Stellen des Dateisystems kopiert.
Die Anwendungen, Konfigurationsdateien und Skripte zur Ausführung der Malware befinden sich im `data.tar.gz`-Archiv im `/tmp`-Verzeichnis. Das Skript `postinst` wird direkt im Anschluss an die Installation automatisiert ausge-

führt. Es wird durch die Malware benutzt um die weitere Installation der Dateien im Dateisystem zu starten. Einen direkten Aufruf von `dpkg` aus diesem Skript ist nicht möglich, da das Paketsystem während der Installation eines Pakets für andere Aufrufe gesperrt ist. Aus diesem Grund wird im `postinst`-Skript mit Hilfe des Befehls `launchctl` ein Daemon beim `launchd`-Dienst registriert und gestartet, der seinerseits nach Beendigung der ersten Installation, die Installation der weiteren Pakete übernimmt. Der `launchd`-Dienst hat die Aufgabe für das Betriebssystem wichtige Dienste zu überwachen und sie nach einem unerwarteten Abbruch erneut zu starten. Die für die Installation der Malware relevanten Zeilen des Skripts befinden sich im folgenden Listing 5.4.

```
1 launchctl submit -l evilTwinInstall -- /tmp/evilTwinInstall.sh
2 launchctl start evilTwinInstall
```

Listing 5.4: `postinst`-Skript zur Registrierung des Installationsdaemons

Der Daemon `evilTwinInstall` wird zunächst gegenüber dem Dienst `launchd` registriert und anschließend gestartet. Das Daemonskript selbst ist in Anhang B zu finden. Zunächst werden alle Softwarepakete installiert, die Abhängigkeiten der benötigten Softwarepakete darstellen. Im zweiten Schritt werden die eigentlich benötigten Pakete `lighttpd` und alle für die Hotspot-Software MyWi gebrauchten Pakete installiert. In einem dritten Schritt werden die durch den Webserver auszuliefernden Dateien an die korrekte Stelle im Dateisystem kopiert. Ebenso werden alle erforderlichen Konfigurationen an die vorgesehenen Stellen kopiert. Im vierten Schritt wird sowohl der Webserver, als auch die Hotspot-Software gestartet. Im fünften Schritt werden die in den bereits kopierten Konfigurationsdateien enthaltenen zusätzlichen Regeln für den Paketfilter `pf` aktiviert. Im sechsten und letzten Schritt wird eine leere Datei im `/tmp`-Verzeichnis angelegt, die dafür sorgt, dass sich der Installationsdaemon beim nächsten Start automatisch vom `launchd`-Dienst abmeldet und sich beendet.

Nachdem das Installationsskript erfolgreich beendet wurde, steht nun ein weiterer Mobile Evil Twin bereit, der seinerseits neue Geräte infizieren kann. Da im Rahmen des Jailbreak-Prozesses weitere Softwarepakete über das Paketmanagement-System installiert werden können, ist an dieser Stelle auch die

Installation, Konfiguration und der Start von Malware denkbar. Auf diese Weise ließen sich vielfältige Angriffe auf das Gerät selbst, aber auch passive und aktive Angriffe von Geräten in der Nähe realisieren. Auch das Ausspionieren von Daten und der gesamten Kommunikation des Benutzers sind an dieser Stelle möglich. In der für diese Arbeit angefertigten, prototypischen Malware wurde auf die Implementierung derartiger Funktionen verzichtet, da der Fokus ausschließlich auf dem Verbreitungsmechanismus liegt und Funktionen dieser Art bereits in vielen anderen Arbeiten gezeigt worden sind.
Im Folgenden ist noch einmal der Gesamtablauf einer Infektion dargestellt. Abbildung 5.4 zeigt die einzelnen Schritte des Ablaufs und die daran beteiligten Komponenten.

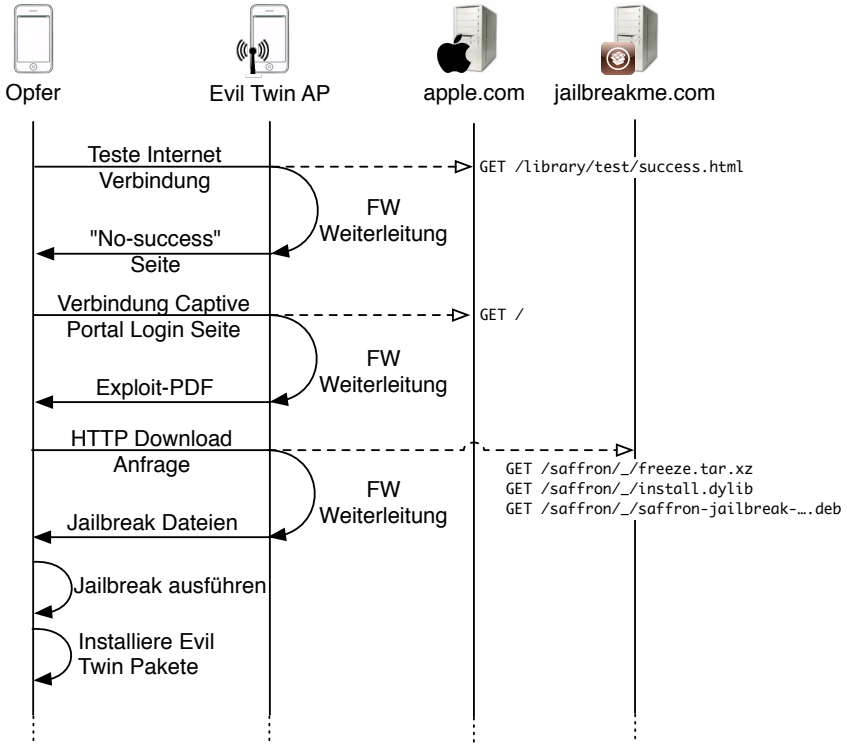


Abbildung 5.4: Schematische Darstellung des gesamten Infektionsprozesses

Die entstandene prototypische Malware eignet sich nicht für einen ernstzunehmenden Angriff unter realen Bedingungen. Dadurch, dass sich die Entwicklung auf die reine Machbarkeit eines derartigen Angriffs konzentriert hat, wurden drei Probleme für eine reale Implementierung nicht bearbeitet. Zunächst wird bei der Infektion mit Hilfe des Popup-Fensters eine kleine Verzögerung benötigt, um nach der Umleitung eine lokale Internetseite im Popup-Fenster anzei-

gen zu können und auf diese Weise die Infektion in Gang zu setzen. Eine solche Verzögerung ließe sich mit einfachen Mitteln implementieren. Des Weiteren benötigt die verwendete Hotspot-Software MyWi einen Neustart des Geräts um in Betrieb genommen werden zu können. Auch hier sind Lösungen denkbar, indem die Software durch eine andere ausgetauscht wird oder dem Benutzer ein anderer Grund für einen sofortigen Neustart des Geräts plausibel gemacht wird. Ebenso ist eine Lösung denkbar, die erst automatisiert beim nächsten Neustart des Gerätes in Aktion tritt. Das dritte und letzte Problem der Implementierung besteht darin, dass der durchgeführte Jailbreak nicht verschleiert wird. Das bedeutet, dass sich nach erfolgtem Jailbreak deutliche Anzeichen dafür in der Benutzeroberfläche wiederfinden. So wird beispielsweise die Paketmanagementsoftware Cydia zusammen mit dem Jailbreak installiert und befindet sich nach Abschluss des Jailbreaks als App auf dem Telefon. Alle beschriebenen bestehenden Probleme wären durch einen Angreifer mit entsprechenden finanziellen Mitteln oder dem nötigen Know-how zu lösen. Sie schränken die Machbarkeit derartiger Angriffe nicht ein.

5.5 Infektionsdauer und Abschätzung des Energiebedarfs

Um die für die Simulation benötigten Parameter zu ermitteln wurden verschiedene Messungen mit Hilfe der entsprechenden Geräte durchgeführt. Zum einen wurde untersucht, welche Übertragungsraten zwischen zwei iOS-Geräten in verschiedenen Entfernungen erreicht werden können. Zum anderen wurde der Energieverbrauch eines iPhones in verschiedenen Situationen gemessen. So wurden Messungen im Ruhezustand und vergleichende Messungen mit einem betriebenen mobilen Hotspot durchgeführt.
Zunächst soll an dieser Stelle auf die Messungen der Übertragungsraten eingegangen werden. Die Wichtigkeit dieser Erhebung ergibt sich aus dem direkten Zusammenhang zu einem der Parameter für die anstehende Simulation der Ausbreitung. Die Übertragungsrate und entsprechend die Dauer, die für die Übertragung der Malware benötigt wird, wird in verschiedenen Abständen gemessen. Das Ergebnis beschreibt die Zeit, für die sich ein infizierendes Gerät mindestens in der Nähe eines Opfergeräts aufhalten muss, um eine Infekti-

on erfolgreich durchzuführen. Es wird an dieser Stelle nur die Übertragung des Malware-Pakets simuliert. Die Übertragung der initialen Webseite mitsamt des enthaltenen bösartigen PDF-Dokuments wird wegen der geringen Größe von gerade einmal 17 kB nicht betrachtet. Im Vergleich hierzu ist das benötigte Softwarepaket, welches zur Generierung eines neuen Evil Twins benötigt wird um ein Vielfaches größer und nimmt entsprechend mehr Zeit in Anspruch. Die Messungen wurden mit zwei iPhones unter freiem Himmel durchgeführt. Die iPhones wurden während der Tests jeweils in der Hosentasche getragen, um Abweichungen zu vermeiden, die aufgrund von Berührungen der Antennen entstehen könnten. Des Weiteren befanden sich keinerlei Hindernisse zwischen den Geräten. Eines der Geräte eröffnete für die Messungen einen mobilen Hotspot. Hierfür wurde dieselbe Hotspot-Software MyWi eingesetzt, welche in Abschnitt 5.4.2 beschrieben wurde. In jeweils 5 Durchgängen wurde ein 10 MB großes Softwarepaket von einem Gerät zum anderen übertragen. Die Entfernungen zwischen den Geräten wurde zwischen den Messungen sukzessive von 1m bis auf 25m erhöht.

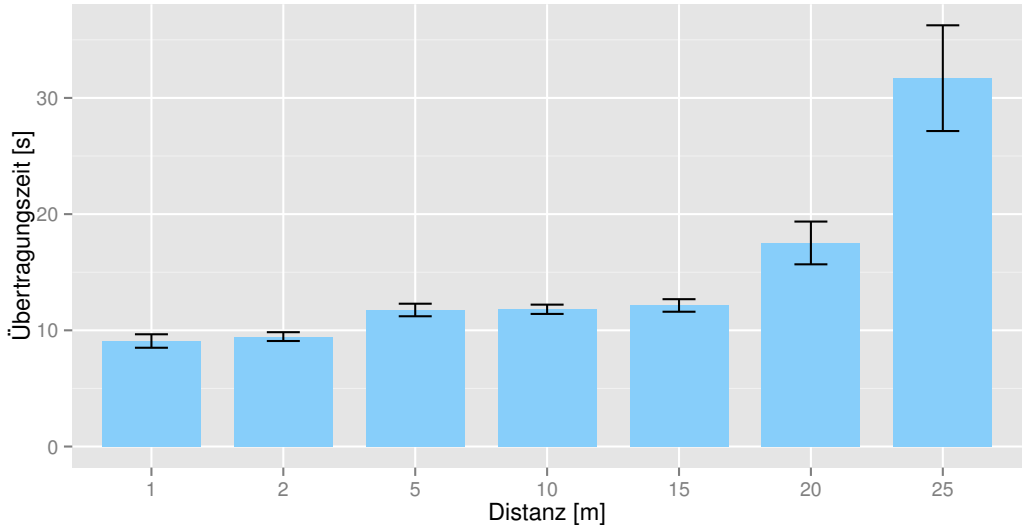


Abbildung 5.5: Übertragungszeiten zwischen zwei mobilen Geräten über verschiedene Distanzen

In Abbildung 5.5 sind die bei der Messung erhobenen Zeiten dargestellt. Es wird pro Distanz die durchschnittliche Zeit für die Übertragung des Softwarepakets zusammen mit der Standardabweichung aufgetragen. Man kann er-

kennen, dass wie erwartet die Dauer für die Übertragung mit zunehmender Entfernung steigt. Ebenso bei einer Entfernung zwischen 20m und 25m ein massiver Anstieg der Dauer beobachtbar. Hingegen bleibt die benötigte Zeit für Entfernungen bis 15 nahezu konstant. Für die folgenden Simulationen wird von einer Übertragungsdauer von konservativen 12 Sekunden ausgegangen. Diese Zeit wird auf Entfernungen bis zu 15 Metern erreicht.

Neben der Betrachtung der Übertragungsraten zwischen mobilen Geräten, wurde ebenfalls der Energieverbrauch eines im Hintergrund arbeitenden Hotspots untersucht. Hierfür wurde ein iPhone dahingehend präpariert, dass es kontinuierlich den aktuellen Ladezustand des Akkus protokolliert. Die Messungen wurden jeweils über einen Zeitraum von 6 Stunden durchgeführt. Begonnen wurde jede Messreihe mit einem vollständig geladenen Akku. In mehreren Durchläufen wurden nun verschiedene Situationen simuliert. Eine dieser Situationen soll das Smartphone ohne einen vorhandenen Evil Twin darstellen. Es wird also das Entladeverhalten des Akkus beobachtet, ohne dass das Geräte benutzt wird und ohne das Vorhandensein einer Malware. Im Rahmen der zweiten Messreihe wird die o.g. Hotspot-Software gestartet und läuft während der gesamten Messung im Hintergrund. Es finden jedoch weder Verbindungen noch Datenübertragungen mit dem Hotspot statt. Für die dritte Messreihe wurde der Aufbau dahingehend verändert, dass ein weiterer Computer in regelmäßigen Abständen von 20 Minuten ein Datenpaket von einem auf dem Hotspot installierten Webserver herunterlädt.

Die Ergebnisse dieser Messreihen sind in Abbildung 5.6 dargestellt. Man kann erkennen, dass der Betrieb des Hotspots einen großen Einfluss auf den Energieverbrauch hat. Während der rote Graph wiedergibt, dass das iPhone nach 6 Stunden im Ruhezustand weniger als 5 % seiner Energie verloren hat, zeigt der grüne Graph, dass man beim Betrieb des mobilen Hotspots MyWi auf dem iPhone mit einem Energiebedarf von ca. 35% in 6 Stunden rechnen muss. Der Energieverbrauch bei eingeschaltetem Hotspot und zusätzlichen wiederholten Downloads (blauer Graph) ist nicht signifikant größer. Hierbei ergibt sich ein Energiebedarf von ca. 35 % in 5,5 Stunden.

Während der Jailbreak bzw. die Malware dahingehend angepasst werden könnte, dass im laufenden Betrieb keine optischen Hinweise mehr vorhanden sind (beispielsweise durch das Entfernen von App Icons zusätzlich installierter Soft-

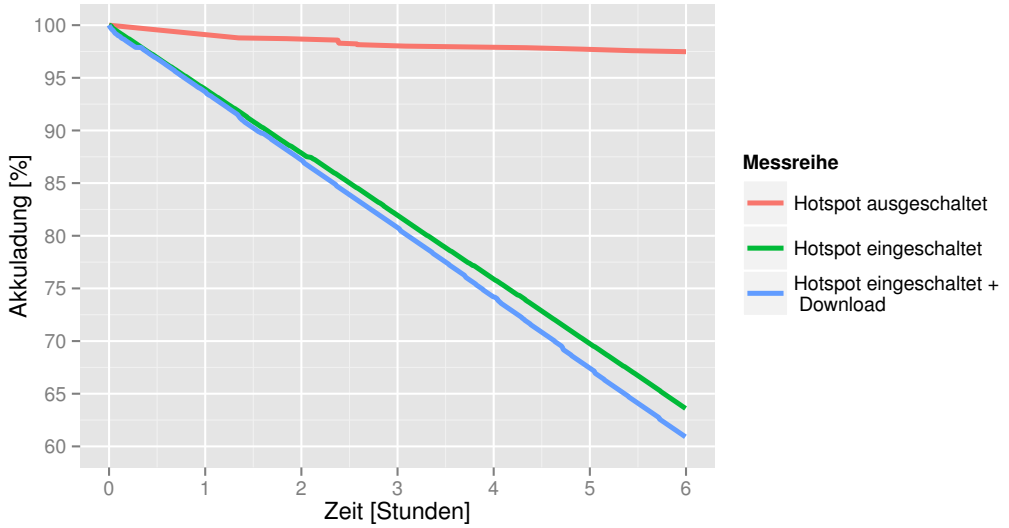


Abbildung 5.6: Betrachtung des Energieverbrauchs des mobilen Hotspots

ware) ist der erhöhte Energieverbrauch nicht zu vermeiden. In den meisten Fällen werden sich Benutzer bei Problemen mit der Akkulaufzeit auf die Suche nach energieraubenden Anwendungen machen, die möglicherweise im Hintergrund laufen. Die Malware würde nach einer entsprechenden Anpassung auf diese Weise nicht mehr gefunden werden können. Von daher ist davon auszugehen, dass möglicherweise ein weiterer bei Benutzern beliebter Schritt erfolgen wird, um dem Problem zu begegnen: der Neustart des Smartphones. Während viele Anwendungen und Dienste auf diese Weise gestoppt werden können, ist der automatische Start eines Dienstes nach einem Neustart des Betriebssystems ohne weiteres möglich. Selbst wenn diese Möglichkeit nicht für Entwickler von Apps und Diensten zur Verfügung stehen würde, so könnte sie dennoch im Rahmen der Malware implementiert werden. Die Malware, welche auf einem Jailbreak basiert, besitzt Administrationsrechte auf dem Smartphone und kann ohne weiteres Startskripte anlegen, verwalten und modifizieren, die während des Bootvorgangs ausgeführt werden. Auf diese Weise ist der erneute Start des mobilen Hotspots technisch leicht umsetzbar. Zusammenfassend kann also festgehalten werden, dass weder das Schließen von Apps, noch der Neustart des Telefons eine Änderung des Verbreitungsverhaltens der mobilen Malware mit sich bringen würde. Lediglich die radikalste aller Möglichkeiten, das komplette Ausschalten des Gerätes würde die weitere Verbreitung der Malware

von diesem Gerät unterbinden. Um die Malware endgültig vom Smartphone zu verbannen kann sie nicht wie eine normale App deinstalliert werden. Stattdessen ist eine komplette Neuinstallation des Betriebssystems nötig, um sicher alle Teile der Malware zu löschen.

5.6 Zusammenfassung

Die Suche geeigneter SSIDs zur Durchführung von Evil Twin-Angriffen ist zunächst entscheidend. Um nachzuweisen, dass diese Suche auch für gezielte Angriffe auf Einzelpersonen möglich ist, wurde im Rahmen einer Feldstudie an verschiedenen Orten gemessen, wie viele SSIDs aus WLAN Management Frames extrahiert werden können. Die Ergebnisse zeigen, dass innerhalb einer Stunde an allen Orten mehrere 100 SSIDs empfangen wurden, was für einen Angriff mehr als ausreichend sein dürfte.
Im zweiten Teil dieses Kapitels wurde die Entwicklung und Funktionsweise einer prototypischen Malware beschrieben. Die Malware nutzt die in allen aktuellen Smartphones vorhandene WLAN-Schnittstelle, eine Hotspot-Software und einen angepasst Jailbreak, um einen Mobile Evil Twin Angriff auszuführen und sich auf diese Weise auf andere Geräte weiter zu verbreiten. Nach Erläuterung der einzelnen Schritte, die für einen solchen Angriff notwendig sind, wurde im technischen Teil auf die Implementierung des Prototyps unter iOS eingegangen. Auf diese Weise konnte nachgewiesen werden, dass derartige Angriffe durch die Modifikation vorhandener Jailbreak-Lösungen möglich sind.
Auch zwei weitere Faktoren wurden betrachtet, die für eine schnelle Ausbreitung der Malware essenziell sind. Einerseits wurde hier die Dauer der Infektion bzw. die Übertragungsdauer der für die Infektion benötigten Daten betrachtet. Durch wiederholte Messungen konnte bestimmt werden, dass die Übertragung einer derartigen Malware für Entfernungen bis 15 m in unter 12 Sekunden möglich sind. Andererseits wurde der Energiebedarf eines Mobile Evil Twins untersucht. Der Betrieb einer Hotspot-Software hat einen erheblichen Einfluss auf den Energieverbrauch eines Smartphones. In Messungen wurde ermittelt, dass der Betrieb in etwa 35% der Akkuladung in den ersten sechs Stunden verbraucht. Gering waren hingegen die Unterschiede zu Messreihen mit wiederholten Datenübertragungen. Diese haben einen über die Gesamtlaufzeit ver-

nachlässigbaren Energieverbrauch. Beide Werte zeigen zwar einen erheblichen Einfluss auf den Energiebedarf, zeigen aber ebenso, dass die Verbreitung einer Malware mit Hilfe dieser Technik über den ganzen Tag hinweg möglich ist. Auch diese Werte gehen in die im folgenden Kapitel beschriebene Evaluation einer Malware-Ausbreitung ein.

Kapitel 6

Evaluation von Malware-Ausbreitungen

Im Folgenden soll die Ausbreitung einer mobilen Malware untersucht werden. Hierzu wird zunächst darauf eingegangen, warum sich der Einsatz von Simulationen als effektive und kostengünstige Alternative zu vielfachen Feldstudien bei der Untersuchung von Malware-Ausbreitungen erwiesen hat. Ebenso wird die Entwicklung eines speziell auf diese Anforderungen angepassten Simulationsframeworks beschrieben und im Anschluss dazu genutzt, um verschiedene Aspekte der im vorherigen Kapitel beschriebenen Malware zu untersuchen. Teile der Malware-Simulation dieses Kapitels sind auf der *International Conference on Wireless and Mobile Computing (WiMob)* [63] veröffentlicht worden.

6.1 Ziel der Evaluation

Für die Bewertung der Risiken mobiler Malware für den Endanwender in der heutigen Zeit soll die Ausbreitung der zuvor beschriebenen Malware untersucht werden. Hierbei soll zum einen auf die Ausbreitungseffizienz, aber ebenfalls auf die Ausbreitungsgeschwindigkeit neuartiger mobiler Malwares eingegangen werden. Die Untersuchung der Malware soll unter möglichst realen Bedingungen erfolgen. Hierfür sind verschiedene Charakteristika zu beachten, die speziell bei dieser Art der Ausbreitung vorzufinden sind. Auf diese zu untersuchenden Aspekte soll im Folgenden kurz eingegangen werden.

Die Ausbreitung soll in realen Umgebungen untersucht werden. Wie in anderen wissenschaftlichen Arbeiten erwiesen [40] bestehen große Unterschiede hinsichtlich einer Ausbreitung zwischen Bewegungsmodellen auf realen Straßennetzwerken im Vergleich zu simpleren Modellen auf einer freien Ebene. Ebenso sollen einfache, bislang verwendete Bewegungsmodelle mit komplexen Bewegungs- und Verhaltensmodellen verglichen und ihr Einfluss auf die Ausbreitung einer Malware untersucht werden.

Betrachtet man den einzelnen Verbreitungsschritt so sind ebenfalls besondere Betrachtungen notwendig, die ausschließlich für die Untersuchung derartiger Ausbreitungen benötigt werden. An dieser Stelle sind insbesondere zeitliche und räumliche Besonderheiten der Ausbreitung zu untersuchen. Im Vergleich zu anderen Infektionsmodellen müssen Übertragungszeiten und räumliche Nähe in Kombination betrachtet und bei der Evaluation berücksichtigt werden. Da es sich bei den betrachteten Endgeräten um akkubetriebene Smartphones und Tablets handelt ist auch der Energiebedarf im Rahmen der Evaluation nicht zu vernachlässigen. Aus diesem Grund sollen ebenfalls Untersuchungen hinsichtlich verschiedener initialer Ladungszustände durchgeführt werden.

Nicht zuletzt sollen darüberhinausgehend auch Untersuchungen dazu angestellt werden, inwieweit besondere Orte wie Cafés oder öffentliche Plätze die Ausbreitung mobiler Malware beeinflussen. Viele dieser Aspekte sind im Rahmen einer Evaluation in der echten Welt nicht durchzuführen. Daher wird im folgenden Abschnitt beschrieben, wieso Simulationen ein geeignetes Mittel für derartige Untersuchungen darstellen.

6.2 Warum Simulationen?

Im Folgenden sollen die Vor- und Nachteile von Simulationen bei verschiedenen Untersuchungen beleuchtet werden. Hierfür wird zunächst die Herangehensweise und Durchführung einer Feldstudie betrachtet. Hierbei müssen eine Vielzahl von Aufgaben bereits im Vorfeld geplant und erledigt werden. Für eine solche Studie müssen Teilnehmer über verschiedene Kanäle, wie Social Media und Mailinglisten rekrutiert werden. Die so gewonnenen Teilnehmer für die Studie müssen je nach Studie im nächsten Schritt instruiert werden. Ebenso müssen weitere Rahmenbedingungen der Studie geklärt werden. Nach Beendigung müssen in vielen Fällen Abschlussgespräche und Interviews geführt werden, um sowohl objektive als auch subjektive Ergebnisse anzufragen. In Studien, in denen auf den Smartphones der Probanden Software installiert oder verändert wurde, muss diese wieder in den Ursprungszustand versetzt werden. Ein weiterer Nachteil von Studien ist die Notwendigkeit, einen Anreiz für die Teilnahme an der Studie schaffen zu müssen. Dies gilt sowohl für die Gewinnung der Teilnehmer, als auch um die Quote der Studienabbrecher gering zu halten. In den meisten Fällen sind nur die Ergebnisse von Teilnehmern verwertbar, die die gesamte Studiendauer teilgenommen haben. Zusammenfassend kann festgehalten werden, dass auf diese Weise durchgeführte Studien zeit- und kostenintensiv sind. Darüber hinaus bleibt bis zum Abschluss der Studie unklar, ob überhaupt relevante Ergebnisse erzielt werden können.

Neben diesen Nachteilen haben Studien mit realen Personen naturgemäß auch Vorteile gegenüber Simulationen. In erster Linie ist hier das Verhalten der Benutzer zu nennen. In Simulationen wird versucht, das Verhalten eines Benutzers so realistisch wie möglich zu modellieren. Als Grundlage für die Implementierung des Verhaltens der Simulationsobjekte dienen oftmals Automaten, die als Zustände die verschiedenen Verhaltensmuster der simulierten Person darstellen. Zwischen den Zuständen gibt es Übergangswahrscheinlichkeiten, die im Vorfeld entweder durch Beobachtung oder Befragung ermittelt werden müssen.

Aber es gibt nicht nur auf den Realismus abzielende Vorteile von Feldstudien. Studien, in denen neuartige Systeme evaluiert werden, haben gegenüber Simulationen ebenfalls den Vorteil, dass im Nachgang subjektive Empfindungen

und Gefühle der Probanden abgefragt werden können. Dieser Aspekt entfällt bei der Simulation derartiger Systeme vollständig.

Um die in Kapitel 5 beschriebene Malware zu simulieren und ihre Ausbreitung in urbanen Regionen zu untersuchen muss eine Vielzahl von Parametern und Ausgangssituationen betrachtet werden. Feldstudien mit realen Personen und ihren mobilen Geräten sind für derartige Untersuchungen aus verschiedenen Gründen nicht praktikabel. Der bereits aufgeführte Nachteil der hohen Kosten und auch die sehr zeitintensive Vor- und Nachbereitung dieser Studien sind bei der Untersuchung von Malwareausbreitungen allerdings nur von untergeordneter Bedeutung. Viel entscheidender sind Sicherheitsrisiken und Bedenken hinsichtlich der Privatsphäre der teilnehmenden Benutzer und Unbeteiligter. Im Rahmen dieser Arbeit wird keine funktionsfähige und sich selbst verbreitende Malware entwickelt werden. Mit einer prototypischen Implementierung wurde unter Laborbedingungen gezeigt, dass diese neuartige Verbreitung mit Hilfe von heutigen Smartphones und Tablets möglich ist. Die Malware zu veröffentlichen und in Umlauf zu bringen wäre aus Sicherheitsgründen nicht zu verantworten. Alleine aus diesem Grund, kann die Ausbreitung nicht mit Hilfe von Feldstudien untersucht werden. Selbst wenn man genügend Teilnehmer rekrutieren könnte, so wäre die Ausbreitung auf Geräte von nicht an der Studie beteiligten Benutzer nicht zu verhindern und nicht kontrollierbar. Aus diesen Gründen ist eine Untersuchung dieser Art in öffentlichen Umgebungen mit öffentlichen Netzwerkinfrastrukturen undenkbar und müssen verworfen werden.

Die Sicherheits- und Privatsphärebedenken von Feldstudien sind bei der Simulation einer solchen Ausbreitung irrelevant. Keinerlei echte Benutzer und und mobile Geräte werden hierbei beteiligt. Ebenso sind Simulationen verschiedenster Parameter ohne große zeitliche und finanzielle Ressourcen durchführbar. Da im Rahmen dieser Arbeit verschiedene Ausgangssituationen untersucht und Parameterstudien durchgeführt werden sollen, bietet sich der Einsatz eines speziell auf diese Anforderungen angepassten Simulators an.

Hierfür wurde in einem ersten Schritt versucht, ein geeignetes Simulationsframework zu finden, welches die Anforderungen dieser Art der Simulation unterstützt. Im folgenden Abschnitt 6.3 werden diese Anforderungen an ein Simulationsframework zur Untersuchung der Ausbreitung von mobiler Malware beschrieben. Im Anschluss daran werden in Abschnitt 6.4 verschiedene

Simulatoren vorgestellt und hinsichtlich ihrer Eignung für Simulationen im Rahmen dieser Arbeit untersucht. In Abschnitt 6.5 wird die Entwicklung und der Einsatz des Mobile Security & Privacy Simulators (MOSP) dargestellt.

6.3 Anforderungen an ein Simulationsframework

Im den folgenden Abschnitten werden Anforderungen dargestellt und erläutert, wie sich diese für die Simulation von Malwareausbreitungen im Allgemeinen und der in Kapitel 4.3 beschriebenen Malware im Speziellen ergeben. Die folgenden Abschnitte beleuchten jeweils eine dieser grundlegenden Anforderung an ein entsprechendes Simulationsframework.

6.3.1 Zeit

Um die Ausbreitung einer Malware simulieren zu können muss ein hierfür entwickeltes Simulationsframework zeit-basiert arbeiten. Der zeitliche Verlauf ist ein entscheidender Faktor für die Beurteilung der Gefährlichkeit einer Ausbreitung. Die zeitliche Komponente der Ausbreitung soll sowohl simuliert als auch dargestellt werden können. Daher ist die Zeit als ein diskreter Simulationsparameter unbedingt erforderlich. Es wird entsprechend ein Framework benötigt, welches diskrete Zeitintervalle simulieren kann.

6.3.2 Verschiedene Kommunikationskanäle

Im Rahmen der Simulation wird nicht nur die Bewegung der einzelnen Benutzer bzw. ihrer mobilen Geräte berechnet. Ebenso wichtig sind Interaktion zwischen den simulierten Objekten. Die für die Simulation von Malwareausbreitungen wichtigste Interaktion ist die Kommunikation zwischen den simulierten Geräten. Der Austausch von Informationen zwischen den Benutzern selbst, aber auch zwischen Benutzern und weiteren Objekten, wie beispielsweise statischen Komponenten einer Netzwerkinfrastruktur müssen in die Simulation mit einbezogen werden. Durch die Bereitstellung weiterer Kommunikationskanäle können auch andere Kommunikationswege – z.B. solche mit unterschiedlicher Signallaufzeit oder Übertragungsgeschwindigkeit – modelliert werden.

Um auch komplexere Szenarien simulieren zu können, müssen Kommunikationskanäle unterschiedliche Routingkonzepte unterstützen. Die gängigsten und für die Simulation entscheidenden sind hierbei Unicast, (also das direkte Senden von Informationen von einem Objekt zu einem spezifizierten anderen) und Broadcast (also das Senden von Informationen von einem Objekt an alle anderen). Darüber hinaus ist aber wenigstens ein weiterer Mechanismus für die Simulation mobiler Malware erforderlich: die Kommunikation mit räumlich nahen Objekten. Nur mit Hilfe dieses Verbreitungsmechanismus kann die Übertragung von Daten zwischen mobilen Geräten mit einer begrenzten Reichweite simuliert werden. Genau dieser Aspekt wird durch die Verwendung des persönlichen Hotspots in der beschriebenen Malware ausgenutzt. Eine derartige Verbreitung muss entsprechend zwingend simuliert werden können.

6.3.3 Ereignisse

Die Modellierung von Ereignissen in der simulierten Welt ist ebenfalls eine wichtige Anforderung an den Simulator. Ereignisse können hierbei beispielsweise Konzerte oder andere Menschenansammlungen sein. Diese Ereignisse müssen sich sowohl zeit- als auch ortsabhängig modellieren lassen. Auf diese Weise können die im Folgenden beschriebenen Verhaltensmuster an konkrete Ereignisse angepasst werden und so das Verhalten des jeweiligen Benutzers / Objekts beeinflussen.

6.3.4 Einbeziehung realer geographischer Daten

Ein besonders wichtiger Aspekt bei einer möglichst realistischen Simulation von Smartphone-Nutzern ist die Bewegung dieser auf realen Straßennetzwerken. Wie Mascetti et al. in ihrer Arbeit [40] gezeigt haben, führt die Bewegung von Objekten auf realen Kartendaten zu signifikant anderen Ergebnisse als die zufällige Bewegung auf einer Ebene. Da mit Hilfe der Simulationen speziell urbane Regionen untersucht werden sollen, in denen sich viele Smartphone-Nutzer aufhalten, ist die Nutzung realer Kartendaten absolut notwendig. Neben der Bewegung auf Straßen bieten öffentlich verfügbare Kartendaten (wie beispielsweise die von OpenStreetMap [49] ebenso die Möglichkeit weitere Objekte, wie bestimmte Zonen (Fußgängerzonen etc.), Gebäude und andere

Points-of-interest (POI) zu modellieren. Auch diese Objekte können Einfluss auf die Bewegung und das Verhalten der simulierten Benutzer haben.

6.3.5 Modellierung verschiedener Verhaltensmuster

Durch die Modellierung verschiedener Verhaltensmuster werden verschiedene Dimensionen der Erweiterbarkeit des Simulators ermöglicht. Zum einen können auf diese Weise verschiedenartige Bewegungsarten erzeugt werden. Für verschiedene Verhaltensmodelle kann nicht nur die Bewegungsgeschwindigkeit variiert werden, sondern auch die Art der Bewegung. So sind nicht nur zielgerichtete Bewegungen möglich, sondern auch ein zielloses Herumlaufen oder ein simulierter Schaufensterbummel, also eine kontinuierliche, aber eher langsame Bewegungen mit vielen Zwischenstopps.
Über die Bewegung als Verhaltensmuster hinaus können weitere Parameter mit Hilfe verschiedener Verhaltensmuster definiert werden. Für die Simulation der Malwareausbreitung beispielsweise lassen sich ebenfalls verschiedene Nutzungsintervalle und Nutzungsdauern des Smartphones für unterschiedliche Personengruppen definieren.
González et al. haben in ihrer Arbeit [42] gezeigt, dass die Art der Bewegung einen signifikanten Unterschied in den Ergebnissen von Simulationen hervorruft. Hierfür haben sie die zufällige Bewegung von Objekten, die mit zuvor genutzten Modellen erzeugt wurden, mit realen, gesammelten Bewegungsdaten verglichen und signifikante Unterschiede festgestellt, die sich auch auf die Ergebnisse von Simulationen auswirken.
Auch Mascetti et al. haben in ihrer Arbeit [40] gezeigt, dass erhebliche Unterschiede zwischen zufälligen und möglichst realistisch generierten Bewegungsmustern bestehen. Hierfür nutzen sie den Kontext-Simulator Siafu [39] und erzeugen basierend auf zuvor modellierten Umgebungen Bewegungsdaten, die sie im Anschluss in einem Simulator verwenden und untersuchen.

6.3.6 Modellierung und Einbindung weiterer Dienste

Um auch übergreifende Systeme simulieren zu können, welche die simulierten Objekte steuern oder beeinflussen, muss der Simulator die Einbindung weiterer Dienste ermöglichen. Denkbar ist an dieser Stelle die Modellierung von

Webdiensten oder Portalen, welche Daten der simulierten Objekte sammeln, verarbeiten und anschließend Nachrichten an die entsprechenden Objekte schicken und somit das Verhalten verändern. Auch die Simulation eines verteilten Sicherheitssystems ist an dieser Stelle denkbar. Ein zentrales System könnte modelliert werden, welches sicherheitsrelevante Informationen von den simulierten Objekten sammelt, aggregiert, auswertet und Warnungen an die Objekte verteilt. Auch diese Warnungen könnten Verhaltensänderungen der Objekte hervorrufen, Kommunikationskanäle verändern oder andere Parameter der Umwelt beeinflussen.

6.4 Nachteile bestehender Simulationsframeworks

Im vorherigen Abschnitt wurden die Anforderungen an einen Simulator zur Untersuchung von Malwareausbreitungen beschrieben. Die sich aus den Anforderungen ergebenden Eigenschaften eines Simulationsframeworks werden im Folgenden mit bereits bestehenden Frameworks verglichen.
Es existieren verschiedene mathematische Modelle zur Beschreibung und Simulation der Ausbreitung von ansteckenden Krankheiten. Murray beschreibt in seinem Buch *Mathematical Biology: An Introduction* [45] einige Beispiele für solche Modelle. Ein besonders simples Modell ist das SI-Modell (Susceptible-Infective-Removed Model). In diesem Modell existieren drei individuelle Klassen von Personen: Susceptibles S, Infectives I und Removed R. Die verschiedenen Zustände können durch die simulierten Personen in folgender Reihenfolge durchlaufen werden:

$$S \longrightarrow I \longrightarrow R$$

Susceptibles: Diese Gruppe besteht aus bislang nicht infizierten, gesunden Personen. Alle Personen in dieser Gruppe sind anfällig für die untersuchte Infektion.

Infectives: Alle lebendigen, infizierten Personen sind Mitglied dieser Gruppe.

Removed: In dieser Gruppe sind all diejenigen Personen, die entweder einmal infiziert waren, nicht mehr infiziert und jetzt immun sind oder

die solange von der Gruppe der Infizierten getrennt waren, dass sie sich von der Infektion erholen konnten.

Nur einige dieser Annahmen des Modells sind sinnvoll auf die Untersuchung von mobiler Malware und ihrer Ausbreitungscharakteristika übertragbar. Betrachtet man die drei Gruppen des oben beschriebenen Modells, so ist die dritte Gruppe (*Removed*) irrelevant für die hier angestellten Betrachtungen. Ein Smartphone oder Tablet welches sich bereits infiziert hat, wird sich ohne die in Kapitel 5.4 beschriebenen Maßnahmen nicht selbst von der Malware befreien können. Nichtsdestotrotz dienten diese Modelle zunächst als Grundlage für die Simulation mobiler Malware.

Bulygin nutzte das Modell in seiner Arbeit [4] und simulierte mit Hilfe dieser Modelle die Ausbreitung von Bluetooth und MMS-Würmern. Der wesentliche Nachteil dieser mathematischen Modelle für die Nutzung im Rahmen dieser Arbeit ist, dass sie eine Ausbreitung ausschließlich über die Zeit beschreiben. Eine räumliche Ausdehnung wird in diesen Modellen nicht berücksichtigt. Untersuchungen darüber, in welchen Gebieten und Infrastrukturen sich eine mobile Malware besonders schnell oder langsam ausbreitet sind mit Hilfe dieser Modelle daher nicht möglich. Parallel zu mathematischen Modellen haben sich Agenten-basierte Modelle weiterentwickelt. Diese berücksichtigen neben der zeitlichen Komponente ebenfalls die räumliche Ausbreitung der Infektion. Die meisten dieser Simulatoren und Modelle basieren allerdings auf sehr schlichten Annahmen (ähnlich mathematischer Modelle). Einige dieser vereinfachenden Annahmen sind homogene Nutzergruppen und das Fehlen von Inkubations- und Ansteckungszeiten. Diese Aspekte sind für die Untersuchung der beschriebenen Malware unerlässlich und werden von daher zwingend in den Simulationen benötigt. Im Rahmen einer ausgiebigen Recherche konnte kein bestehendes Simulationsframework gefunden werden, welches die o.g. Anforderungen umfänglich erfüllt. Aus diesem Grund ist im Rahmen eines studentischen Projekts der Mobile Security & Privacy Simulator (MOSP) entstanden, der im Rahmen dieser Dissertation weitergeführt und erweitert wurde. Die Funktionsweise und seine Implementierung werden im folgenden Abschnitt näher beschrieben.

6.5 Der Mobile Security & Privacy Simulator

Nachdem kein passendes Simulationsframework gefunden werden konnte, welches für die Simulationen im Rahmen dieser Arbeit geeignet ist, wurde mit der Entwicklung des Mobile Security & Privacy Simulators begonnen. Die Arbeiten an diesem Projekt erfolgten in Zusammenarbeit mit meinem Kollegen Benjamin Henne. Ebenso waren Studenten im Rahmen eines Software-Projekts an der Entwicklung des Simulators beteiligt. Die Entwicklung des Simulators wurde auf der *IEEE Conference on Open Systems* veröffentlicht [24]. Eine Erweiterung des Simulators zur Auslagerung von Orten in andere Simulationen wurde auf der *International Conference on Digital Ecosystems and Technologies* [25] veröffentlicht.

Zum einen sollte der Simulator sämtliche der oben aufgeführten Anforderungen erfüllen. Darüber hinaus sollte er ebenfalls als Grundlage für zukünftige Untersuchungen in diesem Forschungsbereich dienen. Er sollte hierbei so flexibel wie möglich bleiben, so dass nicht nur Privatsphäre-bezogene Fragestellungen, sondern auch bestehende und zukünftige Sicherheitssysteme untersucht werden können, bei denen eine räumliche Komponente von entscheidender Bedeutung sind.

Ein weiterer Grund für die Entwicklung des Simulators lag in der Tatsache begründet, dass derartige Fragestellungen und Untersuchungen in Zukunft immer weiter zunehmen werden. Immer mehr Menschen nutzen Smartphones, Tablets, Smartwatches und andere internetfähige, mobile Geräte. Diese Geräte besitzen alle eine Vielzahl von Kommunikationsschnittstellen, wie etwa Mobilfunk, WLAN, Bluetooth, NFC etc. Alle diese Schnittstellen können sicherheitsrelevante Probleme mit sich bringen.

Ein weiterer Faktor, der weitere Untersuchungen in diesem Bereich bekräftigt, ist die im Vergleich zum Wachstum des Smartphonemarkts eher verhaltene Entwicklung bzgl. der Sicherheitssysteme und bzgl. des Bewusstseins für ein ebenfalls steigendes Maß an Sicherheitsbedürfnissen der Nutzer. Smartphones und andere internetfähige Endgeräte sind zu ständigen Begleitern im Alltag geworden. Mit ihnen werden verschiedenste Arten von Onlinediensten genutzt. Neben rein die Privatsphäre betreffenden Problemen in sozialen Netzwerken, können ebenso sicherheitsrelevante Probleme mit anderen Onlinediensten auf-

treten. Prägnante Beispiele für derartige Dienste sind Onlinebanking oder verschiedene Clouddienste zur Aufbewahrung von zum Teil privaten und schützenswerten Daten. Die Nutzung dieser Dienste und die damit verbundene Speicherung von schützenswerten Informationen auf dem jeweiligen mobilen Gerät machen diese zu einem lohnenswerten Ziel für Angreifer.

Zur Untersuchung dieser neuen Gefahren und ebenso zur Evaluierung der hierfür entwickelten Gegenmaßnahmen, Sicherheitssysteme und Algorithmen wurde der Mobile Security & Privacy Simulator entwickelt. In erster Linie diente die Entwicklung hingegen der Untersuchung der Ausbreitung der beschrieben mobilen Malware. Eine wichtige nichtfunktionale Anforderung für den Simulator war es, dass bei der Erstellung von Szenarien und Systemen nur die für die jeweilige Situation relevanten Aspekte modelliert werden müssen. Darüber hinaus sollte die Wiederverwendbarkeit bei der Erstellung von Simulationen gewährleistet werden. Auf diese Weise lassen sich bestehende Simulationen um zusätzliche Aspekte erweitern oder neue Sicherheitssysteme basierend auf anderen schnell modellieren und simulieren.

6.5.1 Implementierung

Nachdem im vorangegangenen Abschnitt Notwendigkeit eines eigenständigen Simulationsframeworks eingegangen wurde, werden in diesem Abschnitt technische Details erläutert. Zunächst wird hierbei die Basis des Simulators beschrieben. Im Anschluss werden verschiedene für die Betrachtungen in dieser Arbeit relevante Aspekte und die hierfür umgesetzten Lösungen näher beschrieben.

Der Mobile Security & Privacy Simulator (MOSP) ist in Python [1] implementiert. Als Basis für den MOSP dient das Simulationsframework SimPy [2]. SimPy selbst ist ein generisches Prozess-basiertes Simulationsframework zur Simulation diskreter Ereignisse. Auf Basis dieses Frameworks wurde der Simulator und die im Folgenden beschriebenen Eigenschaften entwickelt.

[1] `https://www.python.org/`

[2] `http://simpy.readthedocs.org/en/latest/`

Genauigkeit der Position

Sowohl die Genauigkeit der Position von Objekten, als auch die Genauigkeit der simulierten Bewegung dieser Objekte ist von entscheidender Bedeutung bei der Simulation von miteinander kommunizierenden Objekten mit geringer Signalreichweite. In Abschnitt 5.5 wurde gezeigt, dass sich die Übertragungsgeschwindigkeit für Daten stark vermindert, sobald ein gewisser Schwellwert zwischen beiden kommunizierenden Geräte überschritten wird. Bei Untersuchungen im Kontext dieser Arbeit werden in erster Linie WLAN-Verbindungen simuliert, bei welchen beide Kommunikationspartner mobile Endgeräte mit entsprechenden Schnittstellen sind. Entsprechend müssen Simulationen mit einer hohen Positionsgenauigkeit durchgeführt werden.
Oftmals genügen für Betrachtungen der Bewegung von Handys und Smartphones ungenauere Positionsdaten. Diese können beispielsweise durch Mobilfunkprovider gewonnen werden, in dem sie die Funkzellen aufzeichnen bzw. auswerten, in denen ein Handy im Verlauf einer betrachteten Zeitspanne eingebucht war. Durch dieses Vorgehen kann aber nur eine Genauigkeit auf mehrere hundert Meter erreicht werden. Diese Genauigkeit reicht für die Simulationen zur Malwareausbreitung mit dem oben beschriebenen Mechanismus nicht aus. Hierfür werden meter- bzw. sogar zentimetergenaue Positionen benötigt. Nur durch diese hohe Genauigkeit in Verbindung mit einer ebenso gut aufgelösten Zeitskala können die realen Eigenschaften dieser Geräte modelliert und simuliert werden.

Einbindung OSM / Ausbreitung Straßen

Die Simulationen zur Ausbreitung der Malware sollen in realen Straßennetzwerken durchgeführt werden. Hierfür müssen die entsprechenden Daten vor Beginn der Simulation geladen und verarbeitet werden. Als zuverlässige und kostengünstige Quelle für diese Geoinformationen hat sich das OpenStreetMap-Projekt (OSM) herausgestellt. Aus diesem Grund werden für Simulationen mit Hilfe des MOSP aufgearbeitete Kartenausschnitte des OSM-Kartenmaterials verwendet.
Das Kartenmaterial enthält lediglich eine Topologie des Straßennetzwerks. Die Kanten spiegeln die Straßen wieder, die Knoten hingegen Kreuzungen von Stra-

ßen. Zusätzlich befinden sich weitere Punkte, wie z.B. Points-of-interest (POI), in den Kartendaten. Alle Objekte in OSM können beliebig viele sog. Tags besitzen. Diese Tags beschreiben Namen, Bezeichnungen und weitere Metadaten. Ein Teil dieser Metadaten ist auch für die Simulationen im Rahmen dieser Arbeit relevant. Alle Kanten – also entsprechend alle Straßen – sollten einen Tag mit der Bezeichnung *Highway* besitzen. Der Wert dieses Tags gibt des Straßentyp an. Die folgenden Straßentypen sind in OSM definiert: *motorway*, *trunk*, *primary*, *secondary*, *tertiary*, *unclassified*, *residential* und *service*. Während *motorway* eine Autobahn bzw. entsprechend einen Freeway beschreibt, vermindert sich die Größe und Ausdehnung der Straßen bis hin zu *residential* und *service*. Diese beiden Straßentypen beschreiben Straßen in Wohngebieten und kleine Privatstraßen. Mit Hilfe der Abstufungen können die verschiedenen Straßenbreiten der unterschiedlichen Typen mit in die Simulation übernommen werden.

Durch die Einführung der Straßenbreite in der Simulation ist es nun ebenfalls möglich komplexere Situationen zu modellieren. So ist es denkbar, dass sich zwei Objekte entlang einer Straße in entgegengesetzte Richtungen bewegen. Bewegen sich hierbei beide Objekte auf derselben Straßenseite, so entspricht dies einer Simulation ohne Straßenausbreitung. Je nach Geschwindigkeit der beiden Objekte befinden sie sich entweder lange genug in Kommunikationsreichweite um einander zu infizieren, oder sie bewegen sich so schnell an einander vorbei, dass keine Infektion stattfinden kann. Bewegen sich die beiden Objekte hingegen sowohl auf derselben Straßenseite als auch in dieselbe Richtung, so ist alleine die Geschwindigkeit und der Abstand der beiden Objekte entscheidend. Ist der Abstand der beiden Objekte nicht zu groß oder ist der Geschwindigkeitsunterschied zwischen beiden nicht zu hoch, so kann eine Infektion stattfinden. Bewegen sich die beiden Objekte auf gegenüberliegenden Straßenseiten muss eine dritte Komponente – die Straßenbreite – beachtet werden. Auf diese Weise können auch Situationen entstehen, in denen sich beide Objekte mit der gleichen Geschwindigkeit im gleichen Abschnitt einer Straße bewegen. In diesem Fall ist die Breite der Straße entscheidend. Ist diese größer als die Kommunikationsreichweite, so kann keine Infektion stattfinden. Bewegen sich die beiden Objekte hingegen in entgegengesetzte Richtungen, so nimmt die Infektionsgefahr stark ab, da der Aufenthalt in Kommunikations-

reichweite durch die zusätzlich zu berücksichtigende Straßenbreite sehr kurz wird. Von daher ist in diesen Situationen mit einem deutlich geringeren Infektionsrisiko zu rechnen, als in den anderen Situationen.

Verlassen des Straßennetzwerks

Neben der Bewegung auf realen Kartendaten wurden weitere Möglichkeiten für eine möglichst realistische Verhaltensweise innerhalb der Simulation geschaffen. So können simulierte Objekte das Straßennetzwerk an zuvor definierten Orten verlassen und sich frei in einem ebenfalls zuvor festgelegten Bereich bewegen und aufhalten. Auf diese Weise sollen Parks und öffentliche Plätze simuliert werden, auf denen sich Menschen frei bewegen.
Aber nicht nur die Bewegung auf freien Flächen wurde ermöglicht. Auch Cafés und Einkaufsläden werden vereinfacht simuliert. Hierfür werden Knoten innerhalb des Straßennetzwerkes definiert, an denen sich diese speziellen Orte befinden sollen. Ist ein solcher Ort Ziel eines simulierten Objekts, so wird dieses beim Betreten des entsprechenden Knoten für eine bestimmte Zeit aus der globalen Simulation entfernt. In dieser Zeit befindet sich das Objekt im Café bzw. im Einkaufsladen. Nach Ablauf der Aufenthaltsdauer wird das Objekt der globalen Simulation wieder hinzugefügt und bewegt sich weiter wie zuvor. Ob sich das Objekt während des Aufenthalts selbst oder andere infiziert hat kann hierbei beispielsweise mit Hilfe eines vereinfachten mathematischen Modells berechnet werden. Nähere Informationen hierzu befinden sich in Abschnitt 6.6.

Simulationsobjekte

Alle simulierten Objekte basieren innerhalb der Simulation auf der selben Klasse `entity` von der sie abgeleitet werden. Dies schließt sowohl simulierte Menschen mit ihren Smartphones, als auch Serverdienste und Webportale ein. Die grundlegenden Eigenschaften für die Simulation erhalten die Objekte über die `entity`-Klasse. Ebenso werden Registrierungen mit Interaktionskanälen über Schnittstellen der `entity`-Klasse durchgeführt. Auf diese Weise werden die für die Simulation relevanten und grundlegenden Eigenschaften dieser Objekte durch das Simulationsframework bereitgestellt. Bei der Entwicklung der

Simulationen können nun entsprechende Klassen für menschliche und nichtmenschliche Entitäten erzeugt und um weitere Eigenschaften ergänzt werden. Durch die konsequente Anwendung von Vererbung in diesem Bereich ist es ebenfalls leicht möglich, verschiedene Arten von Objekten mit gemeinsamen Eigenschaften oder Parametern zu erstellen. Für die Simulationen im Bereich mobiler Malware sollen beispielsweise verschiedene Verhaltensmuster von Personen existieren. Hierfür wurde eine `SmartphoneUser`-Klasse erzeugt, die von weiteren Klassen beerbt wird. Auf diese Weise können verschiedene Klassen von Benutzern erzeugt werden, die grundlegend identische Eigenschaften besitzen, sich aber in ausgewählten Parametern und Eigenschaften voneinander unterscheiden. Die im Rahmen der Simulation von Malwareausbreitung modellierten unterschiedlichen Eigenschaften sind die folgenden:

Bewegung: Es werden verschiedene Arten der Bewegung im Straßennetzwerk ermöglicht. So sind zielgerichtete, zufällige und variierende Bewegungen möglich.

Smartphone-Nutzung: Es können unterschiedliche Nutzungsverhalten von Smartphones simuliert werden. Für die Simulation von Smartphone-Nutzern sind verschiedene Parameter bedeutend. Die Dauer pro Nutzungsfall ist ebenso so entscheidend wie das generelle Nutzungsverhalten über den gesamten Tag betrachtet.

Verhalten: Das Verhalten der Benutzer, bzw. deren Tagesablauf und andere Verhaltensmuster können ebenfalls in Gruppen modelliert werden.

Weitere Details zur Modellierung der Nutzergruppen und ihres Verhaltens folgen in der eingehenden Beschreibung der Simulation zur Ausbreitung mobiler Malware in urbanen Gebieten im nächsten Abschnitt 6.6.

Interaktionskanäle

Damit simulierte Objekte miteinander interagieren können, werden sie an sog. Interaktionskanälen registriert. Je nach Kanal sorgt der Simulator für den Auf- und Abbau einer entsprechenden Verbindung. Ein Objekt kann an beliebig vielen Interaktionskanälen angemeldet sein. So können alle möglichen Arten von

Interaktionen implementiert werden. Die wichtigste Interaktion zwischen den Objekten im Rahmen dieser Arbeit ist die Kommunikation untereinander. Aus diesem Grund soll im Folgenden das genutzte Kommunikationsmodell beschrieben werden.
Ein Kommunikationskanal soll in vereinfachter Form ein echtes Kommunikationsmedium mit den hierfür benötigten Eigenschaften widerspiegeln. In den Simulationen im Rahmen dieser Arbeit wird auf ein vereinfachtes Kommunikationsmodell zurückgegriffen, in welchem nur relevante und grundlegende Routingfunktionen unterstützt und modelliert werden.

Unicast: Ein simuliertes Objekt kommuniziert mit genau einem anderen. Hierbei ist es nicht relevant, ob es sich bei den Objekten um bewegliche Objekte wie Menschen bzw. Smartphone-Benutzer handelt oder um statische Objekte, wie beispielsweise installierte Netzwerkinfrastruktur.

Broadcast: Ein Objekt sendet eine Nachricht an alle erreichbaren anderen Objekte innerhalb der Broadcast-Domäne. m Normalfall entspricht das einer Nachrichten an alle Teilnehmer eines Kommunikationskanals.

Lokaler Broadcast: Beim lokalen Broadcast handelt es sich um einen herkömmlichen Broadcast, der eine geografische Begrenzung besitzt. Es wird in diesem Fall eine Nachricht an alle Teilnehmer des entsprechenden Kommunikationskanals gesendet, die sich in einem bestimmten Umkreis um den Absender befinden. Mit Hilfe dieser Funktion können Signalreichweiten von Kommunikationsschnittstellen mobiler Geräte modelliert werden.

Auf tiefere Ebenen von Protokollen wird an dieser Stelle verzichtet, da diese für die Betrachtungen im Rahmen dieser Arbeit nicht relevant sind und die Ergebnisse nicht beeinflussen.

Ausgaben

Um die Ergebnisse der Simulationen darstellen und weiterverarbeiten zu können wurden Ausgabekanäle entwickelt, die ebenfalls auf einer Registrierung der Objekte basieren. Auch diese werden von einer Basisklasse `output` abgeleitet

und stellen die für die jeweilige Ausgabe erforderlichen Methoden zur Verfügung. Die simulierten Objekte können auch bei der Auswahl der Ausgabe bei verschiedenen Ausgabekanälen registriert sein. Die folgenden Ausgabekanäle sind verfügbar:

Textausgabe: Die Ausgabe der Teilergebnisse und Endergebnisse in Textdateien ist der für diese Arbeit wichtigste Ausgabekanal. Die Ergebnisse können so mit Hilfe von Analysewerkzeugen weiterverarbeitet und grafisch dargestellt werden.

Konsolenausgabe: Zur Kontrolle einer laufenden Simulation können Ausgaben auf der Konsole hilfreich sein. Hiermit lassen sich Statusinformationen der aktuell laufenden Simulation anzeigen, um diese zu kontrollieren und evtl. auftretende Fehler frühzeitig zu erkennen.

GUI: Eine weitere Möglichkeit der Überwachung ist die Visualisierung der simulierten Objekte. Hierfür wurde ein Programm mit einer Kartendarstellung der aktuellen Simulation entwickelt, in welchem die einzelnen simulierten Objekte dargestellt sind. Auch diese Form der Ausgabe kann bei der Simulation hinreichend weniger Objekte zur Erkennung von Fehlern in der Simulationsmodellierung genutzt werden. Werden hingegen zu viele Objekte simuliert, so leidet die Übersichtlichkeit erheblich und Fehler sind nur in Ausnahmefällen zu erkennen.

Die verschiedenen Ausgabekanäle verfolgen höchst unterschiedliche Ziele. Während die einen zur Suche von Fehlern dienen, sind andere sinnvoll bei der Überwachung laufender Simulationen. Betrachtet man beispielsweise die aktuelle Position der einzelnen simulierten Objekte. Für das Endergebnis und weitere Auswertungen sind nicht alle Positionen von allen Objekten zu allen Zeitpunkten relevant. Für die Darstellung der Situation in einer grafischen Oberfläche hingegen schon. Entsprechend den Bedürfnissen können die einzelnen Objekte an verschiedenen Ausgabekanälen registriert werden.
Das entstandene Simulationsframework wurde neben den im Folgenden dargestellten Simulationen zur Analyse sicherheitsrelevanter Fragestellungen auch in Untersuchungen zum Schutz der Privatsphäre von Benutzern verwendet, die im Rahmen einer wissenschaftlichen Konferenz veröffentlicht [26] wurden.

6.6 Simulation mobiler Malware in urbanen Gebieten

Die im Rahmen dieser Arbeit entstandene und in Kapitel 4 vorgestellte mobile Malware soll mit Hilfe des beschriebenen Simulators untersucht werden. Insbesondere soll die Verbreitung der Malware in einem urbanen Gebiet mit einer entsprechend hohen Smartphonedichte analysiert werden. Die hierfür entwickelte Simulation für den Mobile Security & Privacy Simulator wird zunächst eingehend beschrieben. In einem weiteren Schritt werden vier Verhaltens- und Infektionsmodelle entwickelt, simuliert und miteinander verglichen. Abschließend werden die Ergebnisse der verschiedenen Simulationen dargestellt und ihre Bedeutung für das Risiko bei der alltäglichen Smartphone-Nutzung diskutiert. Teile dieses Abschnitts basieren auf einer wissenschaftlichen Veröffentlichung [63] im Rahmen der *International Conference on Wireless and Mobile Computing, Networking and Communications*.

In diesem Abschnitt wird die Simulation zusammen mit allen erforderlichen Voraussetzungen und Annahmen beschrieben. Zunächst wird auf die Modellierung der Umwelt und die hierbei getroffenen Annahmen eingegangen. In einem zweiten Schritt wird der Vorgang der Infektion zwischen zwei mobilen Geräten beschrieben.

6.6.1 Modellierung von Objekten und Umwelt

Um die Ausbreitung einer Malware untersuchen zu können, müssen für die hierfür benötigten Simulationen viele Parameter bestimmt und bestmöglich geschätzt werden. Eine dieser Schätzungen betrifft die Anzahl der Benutzer bzw. der Geräte, die potentiell durch die mobile Malware infiziert werden können. Um diese Anzahl zu bestimmen, wurden verschiedene Quellen herangezogen. Zunächst wurde die Gesamtzahl der Personen ermittelt, die im simulierten Bereich anzutreffen sind. Über verschiedene Statistiken wurde im Anschluss die ungefähre Anzahl der betroffenen Benutzer bzw. deren Smartphones ermittelt. Darüber hinaus wurde das Nutzungsverhalten anhand anderer wissenschaftlicher Studien mit in die Simulation einbezogen.

Grundlegende Rahmenbedingungen

Zunächst soll das Augenmerk auf die simulierte Umgebung und die an diesem Ort befindlichen Smartphones gelenkt werden. Zur Untersuchung der Ausbreitung in urbanen und damit bevölkerungsreichen Gebieten wurde der Stadtkern von Chicago, Downtown Chicago (The Loop) gewählt. Die zur Simulation benötigten Geodaten wurden aus den Karten von OpenStreetMap extrahiert und für das Simulationsframework aufbereitet. Ebenso wurden öffentliche Anlagen, Parks und eine Vielzahl von Cafés in den Geodaten so angepasst, dass sie innerhalb der Simulation durch die simulierten Objekte erreichbar sind.
Der nächste entscheidende Parameter für die Simulation ist die Anzahl der simulierten Benutzer. Wie auch in der Arbeit von Husted et al. [28] wurden hierfür Statistiken des öffentlichen Nahverkehrs der Stadt Chicago [72] herangezogen, um die Anzahl der zu simulierenden Objekte zu bestimmen. Ebenso wurden Statistiken zur Smartphone-Nutzung herangezogen, um diesen Parameter genauer bestimmen zu können. Hierbei hat sich ergeben, dass an einem normalen Werktag einschließlich Berufspendlern ca. 400.000 Personen in Downtown Chicago anzutreffen sind, die im Besitz eines Smartphones sind. Da sich die hier simulierte Malware nur unter iOS einsetzen lässt, musste die Anzahl der simulierten Benutzer entsprechend vermindert werden. Zum Zeitpunkt der Durchführung der Simulation hatte iOS einen Marktanteil von 12,4 % in den USA [8]. Die in Kapitel 4 beschriebene mobile Malware kann in ihrer derzeitigen Form jedoch nur eine spezielle Version des Betriebssystems iOS angreifen. Deswegen wurde die Anzahl der potentiell gefährdeten und entsprechend simulierten Geräte abermals vermindert. Unter Zuhilfenahme einer Statistik eines App-Entwicklers konnte festgestellt werden, dass die Version 4.3.3 von iOS zum Zeitpunkt der Datenerhebung eine Verbreitung von rund 17% unter allen iOS-fähigen Geräten [10] hatte.
Durch die Gesamtheit aller herangezogenen Statistiken und Studien wurde die Größe der Population mit infizierbaren mobilen Geräten auf 8.000 festgelegt. In Abschnitt 2.6.1 wurde bereits beschrieben, dass andere Versionen von iOS potentiell ebenso anfällig für derartige Angriffe. Aus diesem Grund und weil auch plattformübergreifende Malwares denkbar sind wurden die Simulationen im Rahmen dieser Arbeit mit einer Population von 10.000 Geräten durchgeführt. Hierbei ist zu beachten, dass die Größe der Population unter den be-

schriebenen Annahmen eine eher konservative Schätzung ist. Die Anzahl der infizierbaren Geräte kann durchaus bedeutend größer sein. Durch eine größere Population würde die Ausbreitung einer Malware begünstigt und somit beschleunigt werden.

Energiebedarf

Wie in Abschnitt 5.5 beschrieben, hat der Betrieb des mobilen Hotspots auf den infizierten Geräten einen signifikanten Einfluss auf die Akkulaufzeit der Smartphones. Der Betrieb der Hotspot-Software und der damit verbundene, erhöhte Batterieverbrauch werden ebenfalls in der Simulation berücksichtigt. Infizierte Geräte werden so modelliert, dass sie einen entsprechend höheren Energieverbrauch haben als nicht infizierte. Die Akkulaufzeit der Geräte, sowie aller anderen für die Simulation festgelegten Parameter sind in den Tabellen 6.1 und 6.2 dargestellt. Zusätzlich zum erhöhten Energiebedarf wird ein kleiner Teil der Energie pro Infektion abgezogen vom aktuellen Akkustand abgezogen, um der Übertragung der Malware Rechnung zu tragen.

Übertragung

Auch die Übertragungszeit der Malware, die durch wiederholte Messungen ermittelt und in Abschnitt 5.5 dargestellt wurde, wird im Rahmen der Simulation berücksichtigt. Die Infektionsdauer innerhalb der Simulation setzt sich hierbei aus zwei Komponenten zusammen: der Übertragungszeit selbst und der Dauer für die Installation der Malware auf dem neu infizierten Gerät. Es wird in diesem Modell von einer konservativen Dauer von 15 Sekunden ausgegangen. Dies teilt sich in eine Übertragungszeit von 12 Sekunden und eine Installations- und Ausführungszeit von 3 Sekunden auf.

Nutzergruppen

Um ein möglichst realistisches Bild der Bevölkerung dieses Stadtteils zu erzeugen, werden verschiedene Personengruppen modelliert. An einem normalen Werktag wird die Bevölkerung in diesem Stadtteil von Pendlern dominiert. Für die Simulationen wurden fünf verschiedene Nutzergruppen mit unterschiedlichen Eigenschaften erzeugt. Ebenso wurden vier Aktivitäten modelliert, welche

jede simulierte Person ausführen kann. Zunächst sollen diese vier Aktivitäten vorgestellt werden:

walking: Ein Benutzer wählt sich ein zufälliges Ziel im simulierten Gebiet und bewegt sich darauf zu. Am Ziel angelangt verbleibt der Benutzer für einen ebenfalls zufälligen Zeitraum an diesem Ort.

public space: Ein Benutzer bewegt sich in eine öffentliche Anlage oder einen Park. Diese sind in den Geodaten als solche markiert. Auch hier verbringt der Benutzer einen zufällig gewählten Zeitraum.

location: Ein Benutzer besucht einen speziellen Ort (in dieser Simulation speziell Cafés) und verbleibt auch hier für einen zufällig bestimmten Zeitraum. Beim Betreten eines Cafés wird die entsprechende Person aus der globalen Simulation entfernt und im Folgenden durch das nachfolgend beschriebene mathematische Modell für Cafés simuliert.

leave: Ein Benutzer verlässt die Simulation. Hierfür wird ein Knoten am Rand des simulierten Gebiets ausgewählt. Der Benutzer bewegt sich zu diesem Randknoten und wird anschließend aus der Simulation entfernt.

Geschlossene Systeme haben eine andere Charakteristik, was die Ausbreitung einer Infektion angeht. In einem geschlossenen System würden zwangsläufig früher oder später alle Geräte infiziert werden. In einem offenen System hingegen ist dies nicht der Fall. Benutzer können ebenso den Bereich der Simulation verlassen. Das Verlassen des simulierten Bereichs ist hierbei unabhängig von der Infektion. Ebenso können neue Benutzer die Simulation betreten und sich in die bereits simulierten Objekte eingliedern. Durch das Verlassen infizierter Benutzer verringert sich die Infektionsgefahr aller anderen Benutzer im simulierten Bereich. Um die Realität möglichst genau abzubilden ist die Simulation eines offenen Systems unerlässlich.
Jeder simulierten Person ist ein Zustandsautomat zugeordnet, welcher während der Simulation entscheidet, welche Aktion durch diese Person als nächstes ausgeführt wird. Neben den Aktivitäten selbst ist hier auch das Nutzungsverhalten des Smartphones gespeichert – also wie oft ein Benutzer sein Smartphone bzw. das Internet nutzt. Als Grundlage für die Nutzung des Smartphones wurde eine Studie von Karlson et al. [33] herangezogen. Diese Studie bezieht sich

auf Arbeitnehmer, die in ihrem Alltag verstärkt Online-Dienste, Email usw. nutzen. Aus diesem Grund wurden die in dieser Studie ermittelten Werte als obere Grenze angenommen. Die Werte der anderen Benutzergruppen wurden entsprechend konservativ als niedriger angenommen.
Die für diese Simulation erstellten Benutzergruppen sind die folgenden: *Power Users* stellen Benutzer dar, die ihr Smartphone überdurchschnittlich häufig nutzen. *Window Shoppers* zeichnen sich maßgeblich durch ihr Bewegungsverhalten aus. Sie bewegen sich langsamer und sollen Personen darstellen, die einen entspannten Einkaufsbummel machen. *Cafe Visitors* zeichnen sich durch häufige und lange Besuche der modellierten Cafés aus. Mit den *Average People* wurde versucht einen möglichst durchschnittlichen Benutzer zu modellieren. *Strolling People* ist eine Benutzergruppe die aus Spaziergängern besteht. Sie suchen selten Lokalitäten auf und zeichnen sich dadurch aus, dass sie sich überdurchschnittlich viel durch den simulierten Bereich bewegen.
Die genauen Werte, welche in der Simulation zum Einsatz gekommen sind, finden sich in den Tabellen 6.1 und 6.2. Die Spalte mir der Bezeichnung v gibt die Geschwindigkeit an, mit der sich ein Benutzer aus dieser Gruppe bewegt. Der Parameter p beschreibt den prozentualen Anteil dieser Nutzergruppe an der Gesamtpopulation. Die folgenden sechs Spalten geben jeweils die oberen und unteren Grenzen der Aufenthaltsdauern für die o.g. Orte an. In diesen Grenzen wird jeweils eine zufällige Zeit als Aufenthaltsdauer gewählt. Die Parameter t_0 und t_{loc} stellen die Nutzungsintervalle der mobilen Geräte dar. Es gibt zwei solcher Intervalle, weil davon ausgegangen wird, dass Smartphone-Nutzer öfter auf das Internet zugreifen, wenn sie sich in einem Café befinden. Die folgenden vier Spalten geben die Übergangswahrscheinlichkeiten für andere Aktionen an. Die Spalten b und b_i beschreiben die Akkulaufzeit. Einmal unter normalen Bedingungen (b) und einmal bei einem infizierten Gerät (b_i).

Infektion

Die initiale Infektion findet an einem sehr bevölkerten Ort – in diesem Fall einer Starbucks Filiale – statt. Auf diese Weise würde auch ein realer Angreifer vorgehen. Simuliert wird an dieser Stelle ein infiziertes Smartphone. Andere Smartphones infizieren sich im Café wenn sie die weiteren Bedingungen für eine Infektion erfüllen. In der Simulation wird zwischen zwei Infektionsumge-

Typ	v	p	**wait min**	**wait max**	**cafe min**	**cafe max**	**pub min**	**pub max**
Power Users	1,5	20	1	30	30	120	5	30
Window Shoppers	1,2	30	1	30	5	20	2	30
Cafe Visitors	1,1	10	2	10	30	120	30	240
Average People	1,2	20	1	10	5	120	5	120
Strolling People	1,0	20	1	20	20	60	20	60

Tabelle 6.1: Simulationsparameter verschiedener Nutzergruppen der Simulation zur Malwareausbreitung

Typ	t_{o}	t_{loc}	P_{walk}	P_{loc}	P_{pub}	P_{leave}	b	$\mathrm{b_i}$
Power Users	15	15	70	20	6	4	10	6
Window Shoppers	30	20	40	18	40	2	18	8
Cafe Visitors	30	20	8	50	40	2	18	8
Average People	30	30	32	32	32	4	36	10
Strolling People	30	30	58	20	20	2	36	10

Tabelle 6.2: Fortsetzung: Simulationsparameter verschiedener Nutzergruppen der Simulation zur Malwareausbreitung

bungen unterschieden. Die Infektion im öffentlichen Raum (auf der Straße, in Parks) hängt von den folgenden Größen ab: Position der Geräte, Nutzung des Smartphones, Abstand zwischen einem infizierten und einem nicht infizierten Gerät und dem Zeitraum, in dem sich die Geräte in Kommunikationsreichweite befinden.

Im Vergleich zu traditionellen Infektionsmodellen wird in dieser Simulation nur dann ein Infektionsvorgang gestartet, wenn das zu infizierende Gerät in der Nähe eines infizierten Geräts eine Verbindung zum Internet aufbauen will bzw. wenn das entsprechende Simulationsobjekt durch das Nutzungsmodell aktiviert wird. Durch diese realistische Einschränkung nimmt das Infektionsrisiko stark ab, da die reine Begegnung mit einem infizierten Gerät nicht mehr für eine Infektion ausreichend ist. Eine erfolgreiche Infektion findet nur dann statt, wenn sich die beiden beteiligten mobilen Geräte für mindestens 15 Sekunden in einem Umkreis von nicht mehr als 15 Metern befinden. Nur dann wird davon ausgegangen, dass genügend Zeit für die Übertragung der Malware und die anschließende Aktivierung zur Verfügung stand. Nach einer erfolgreichen Infektion kann das infizierte Geräte selbst wiederum andere Geräte anstecken. Infizierte Geräte werden in ihrer Bewegung und ihrem Verhalten nicht von den nicht infizierten unterschieden. Sie können ebenso Cafés und andere POIs besuchen. Außerhalb von Gebäuden basiert die Infektion in dieser Simulation also nicht auf mathematischen Modellen, sondern auf logischen, zeitlichen und räumlichen Bedingungen, sowohl des Infizierenden, als auch des Infizierten.

Modellierung von Cafés

Die Orte von Cafés sind zwar in den OpenStreetMap Daten enthalten, nicht aber ihre Ausbreitung und ihre Aufteilung. Daher werden sie im Rahmen dieser Simulationen nicht auf dieselbe Art simuliert, wie das Straßennetzwerk und öffentliche Plätze. Stattdessen kommt ein vereinfachtes mathematisches Modell zum Einsatz. Da keinerlei Daten über die Größen von Cafés in den OSM-Daten hinterlegt sind, wurden diese manuell in die Simulation eingefügt. Die Größe der Cafés wurde für die Simulation zwischen 30 und 300 m^2 angenommen.

Die Wahrscheinlichkeit, mit der sich Objekte innerhalb des Cafés mit der Malware infizieren ergibt sich folgendermaßen: Zunächst wird die Fläche pro infiziertem Gerät mit Hilfe der folgenden Formel ermittelt:

$$A_i = \beta \cdot \frac{l \cdot a}{i} \tag{6.1}$$

Die Fläche pro infiziertem Gerät ergibt sich aus der Anzahl der Stockwerke l multipliziert mit der Grundfläche a geteilt durch die Anzahl an Infizierten im Cafe i. Der Faktor β dient der Dämpfung, um sich gegenseitig störende Netzwerkinfrastrukturkomponenten zu berücksichtigen. Die Dämpfung ist aber auch deshalb notwendig, weil durch Formel 6.1 von einer gleichmäßigen Verteilung der Geräte auf die gesamte Ladenfläche ausgegangen wird. Da dies im Allgemeinen nicht der Fall sein wird und da durch eine andere Verteilung das Infektionsrisiko sinkt. Für die Simulationen bei diesen Untersuchungen wurde ein Dämpfungsfaktor von $\beta = 0.16$ festgelegt, basierend auf Versuchsreihen mit dem in Kapitel 4 beschriebenen Prototypen.
Aufgrund von gegenseitigen Störungen und um bei der Infektion innerhalb von Cafés eher konservative Abschätzungen zu erhalten wurde die Reichweite der mobilen Geräte r_{WiFi} ebenfalls auf 5m herabgesetzt.
Die räumliche Wahrscheinlichkeit für eine Ansteckung wird über das Verhältnis von der vom WLAN überstrahlten Fläche zur Fläche pro infiziertem Gerät A_i ermittelt. Der zeitliche Faktor wird über das Verhältnis von Zeitdauer des Aufenthalts t_{visit} zum Aktivierungsintervall des Geräts t_{loc} berechnet. Die Kombination dieser beiden Faktoren führt zur finalen Infektionswahrscheinlichkeit P_i:

$$P_i = \frac{t_{\text{visit}}}{t_{\text{loc}}} \cdot \frac{\pi r_{\text{WiFi}}^2}{A_i}. \tag{6.2}$$

6.6.2 Vergleich von Bewegungs- und Infektionsmodellen

In vielen Arbeiten [34, 74, 12, 13, 52, 46, 18, 57] wurden Simulationen eingesetzt, um die Ausbreitung von Malware zu untersuchen. Diese basieren alle auf den o.g. mathematischen oder sehr beschränkten Modellen. Um diese Modelle mit den Simulationen dieser Arbeit zu vergleichen, wurden die vereinfachten Modelle auch für den MOSP implementiert. Die Vereinfachungen

beziehen sich hierbei sowohl auf das Bewegungsmodell, als auch auf das Infektionsmodell. Das oben beschriebene, komplexe Bewegungsmodell wird im Folgenden mit rein zufälligen Bewegungen verglichen. Die Infektion wird im Vergleich zur oben beschrieben dahingehend vereinfacht, dass eine erfolgreiche Infektion stattfindet, sobald ein infiziertes und ein nicht infiziertes Gerät sich in Kommunikationsreichweite befinden. Mit Hilfe dieser jeweils zwei Varianten wurden vier unterschiedliche Simulationen modelliert und verglichen. Diese werden im Folgenden mit R-Z, R-D, F-Z und F-D bezeichnet und setzten sich folgendermaßen zusammen:

Zufällige Bewegung, instantane Infektion (R-Z): Dies ist das einfachste Modell, bei dem keine der oben genannten Verbesserungen enthalten ist.

Zufällige Bewegung, realistische Infektion (R-D): Als Bewegungsmodell wird das vereinfachte, zufällige Modell angewendet. Beim Infektionsmodell hingegen werden die oben genannten Verbesserungen bzgl. einer realistischen Ausbreitung angewendet.

Komplexes Bewegungsmodell, instantane Infektion (F-Z): Während in dieser Variante wiederum das vereinfachte Infektionsmodell zum Einsatz kommt, wird das vereinfachte Bewegungsmodell in dieser Variante durch das oben beschriebene Modell mit verschiedenen Bewegungs- und Verhaltensmustern ersetzt.

Komplexes Bewegungsmodell, realistische Infektion (F-D): In dieser Variante kommen alle genannten Verbesserungen zum Einsatz. Zum einen das komplexe Bewegungsmodell mit verschiedenen Nutzergruppen und Verhaltensmustern, zum anderen aber auch die beschriebene, auf räumliche Nähe und Zeit basierende Infektion.

In Diagramm 6.1 ist der Unterschied der vier genannten Szenarien dargestellt. Es wurde ein maximaler Zeitraum von 12 Stunden mit 10.000 Geräten simuliert. Die vier Szenarien teilen sich optisch im Diagramm in zwei Gruppen auf. Während die beiden Simulationen, die auf einer instantanen Infektion beruhen eine Infektion aller Geräte bereits nach weniger als einer bzw. weniger als 3

Stunden zu verzeichnen haben, ist ein deutlicher Unterschied zu den Simulationen mit dem vorgeschlagenen, realistischen Infektionsmodell zu erkennen. Betrachtet man die Bewegungsmodelle, so sind ebenfalls Unterschiede zwischen den beiden Modellen zu erkennen. Diese haben zwar ebenfalls einen Einfluss auf die Entwicklung der Ausbreitung, sind aber deutlich weniger ausgeprägt als die Differenzen der verschiedenen Infektionsmodelle.

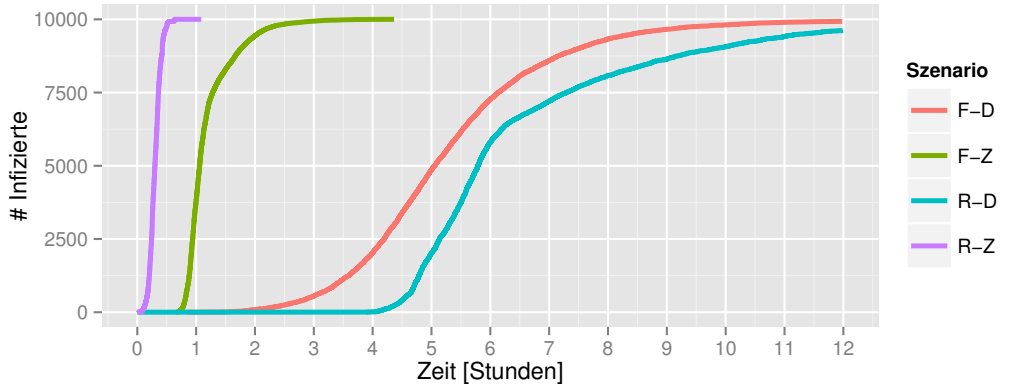


Abbildung 6.1: Vergleich der Ausbreitungsgeschwindigkeit der vier beschriebenen Szenarien R-Z, R-D, F-Z und F-D.

In Diagramm 6.2 sind die beiden Szenarien R-Z und F-Z noch einmal gegenübergestellt. Das Diagramm umfasst verschiedene Simulationen. Zunächst wurde überprüft, welchen Einfluss die Anzahl der simulierten Geräte auf den Gesamtverlauf einer Infektion haben. Wie im linken Bereich ersichtlich ist, findet eine vollständige Infektion aller simulierten Geräte nach weniger als einer Stunde statt. Dies geschieht unabhängig von Anzahl der simulierten Geräte. Auch die Simulation von 6.000, 4.000 und 2.000 Geräten führt zu einer vollständigen Infektion nach weniger als einer Stunde. Durch das Hinzufügen realistischer Bewegungen in Form des neuen Bewegungsmodells und durch die Modellierung von speziellen Orten, wie Cafés (siehe Diagramm, F-Z) verlangsamt sich der Verlauf der Infektion nur geringfügig. Eine vollständige Infektion ist dennoch nach etwas mehr als drei Stunden erreicht. Durch die Einführung des neuen Infektionsmodells ergibt sich hingegen eine deutliche Änderung hinsichtlich des zeitlichen Verlaufs der Malwareausbreitung. Die drei rechten Kurven zeigen den zeitlichen Verlauf der Infektion mit dem erarbeiteten realistischen Infektionsmodell. Man sieht hier bereits deutlich eine Verlangsamung der Aus-

breitungsgeschwindigkeit. Darüber hinaus wurden verschiedene Aktivierungsintervalle der Geräte simuliert. Auch hier kann ein deutlicher Einfluss auf die Ausbreitungsgeschwindigkeit beobachtet werden. Wie erwartet ergibt sich eine langsamere Ausbreitung bei wachsendem Aktivierungsintervall. Im Vergleich zur Ausbreitung bei instantaner Infektion ergeben sich Ausbreitungsgeschwindigkeiten, die für eine vollständige Infektion mehr als 10 Stunden in Anspruch nehmen.

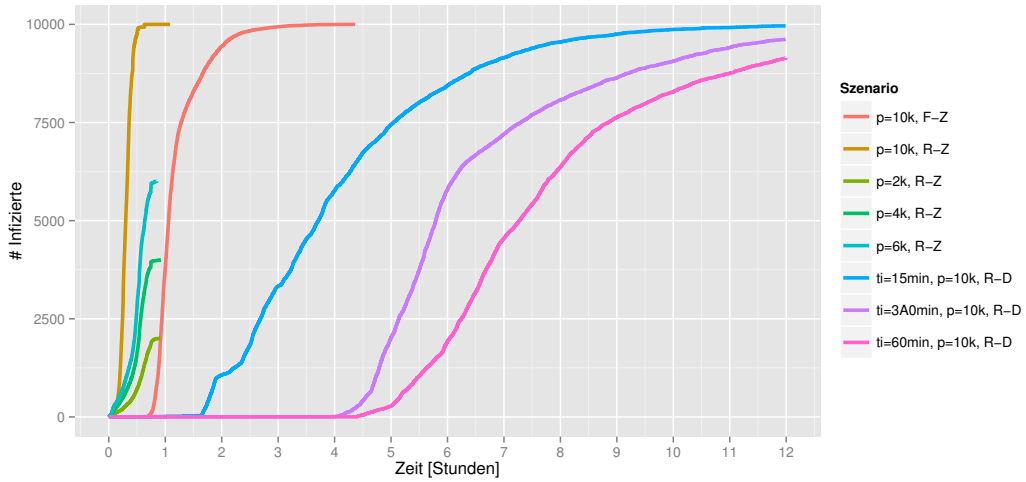


Abbildung 6.2: Vergleich der verschiedenen Modelle, mit Anpassungen verschiedener Parameter

Mit Hilfe der Diagramme 6.1 und 6.2 konnte gezeigt werden, dass die Ausbreitungsgeschwindigkeit erheblich durch die Einführung eines neuen Bewegungs- und Verhaltensmodells beeinflusst wird. Ebenso hat eine realistische Modellierung der Gerätenutzung erhebliche Auswirkungen auf die Ergebnisse der Simulationen. Somit ist zu erwarten, dass durch die Einführung und Nutzung dieser neuen Modelle im Rahmen der Simulation, genauere und realistischere Ergebnisse bezüglich der Ausbreitung mobiler Malware erzielt werden können.

Die Rohdaten der Ergebnisse aller durchgeführten Simulationen befinden sich in der Abgabeversion dieser Dissertation im beigefügten elektronischen Verzeichnis im Ordner *Simulationsergebnisse*.

6.7 Ergebnisse

Im Folgenden werden die Ergebnisse aller weiteren Simulationen zur Malwareausbreitung dargestellt und diskutiert. Mit Hilfe der Simulationen soll ein Gefühl dafür gewonnen werden, wie die Ausbreitung dieser neuen Art von Malware sich in urbanen Gebieten wie Downtown Chicago entwickeln kann. Hierfür werden zahlreiche Simulationen durchgeführt. Es wird unter Einbeziehung des im Rahmen dieser Arbeit entwickelten Modells untersucht, welchen Einfluss die Größe der Population auf die Ausbreitung der Malware hat. Ebenso wird untersucht, welchen Einfluss die Größe des Aktivierungsintervalls der mobilen Geräte auf die Ausbreitung hat. Auch werden offene Systeme geschlossenen gegenübergestellt und entsprechende Simulationsergebnisse miteinander verglichen. Der gesteigerte Energiebedarf der Malware wird ebenfalls betrachtet und es wird untersucht, wie sich dies auf die Ausbreitung der Malware auswirken kann. Schlussendlich wird eine Kartendarstellung präsentiert, die untermauert, dass die im Rahmen dieser Arbeit entwickelten Modelle zu signifikant anderen und realistischeren Ergebnissen führen, als bislang verwendete Modelle.

In den folgenden dargestellten Simulationen wird das vollständige Modell mit allen oben beschriebenen Eigenschaften genutzt. Zunächst soll hierbei auf die Größe der Population eingegangen und untersucht werden, wie sich diese auf die Ausbreitung auswirkt. In Abbildung 6.3 ist das Ergebnis einer Parameterstudie zur Population in einem geschlossenen System abgebildet. Das Diagramm zeigt die normalisierten Ergebnisse der Ausbreitung. Man kann erkennen, dass eine nahezu vollständige Infektion aller simulierten Geräte für Populationen größer als 6.000 Personen innerhalb der ersten 12 Stunden geschieht. Bei Populationen, die aus mehr als 10.000 Geräten bestehen, wird eine vollständige Infektion bereits innerhalb von 8 Stunden erreicht. Die in Abschnitt 6.6 bestimmte Population von 8.000 potentiell gefährdeten Smartphones würde also in diesem geschlossenen System eine nahezu vollständige Infektion aller Geräte bedeuten.

Wie beschrieben, wurden die weiteren Simulationen mit einer Population von 10.000 Geräten durchgeführt, um der steigenden Anzahl an Smartphones und evtl. vorhandener plattformübergreifender Exploits Rechnung zu tragen. Um

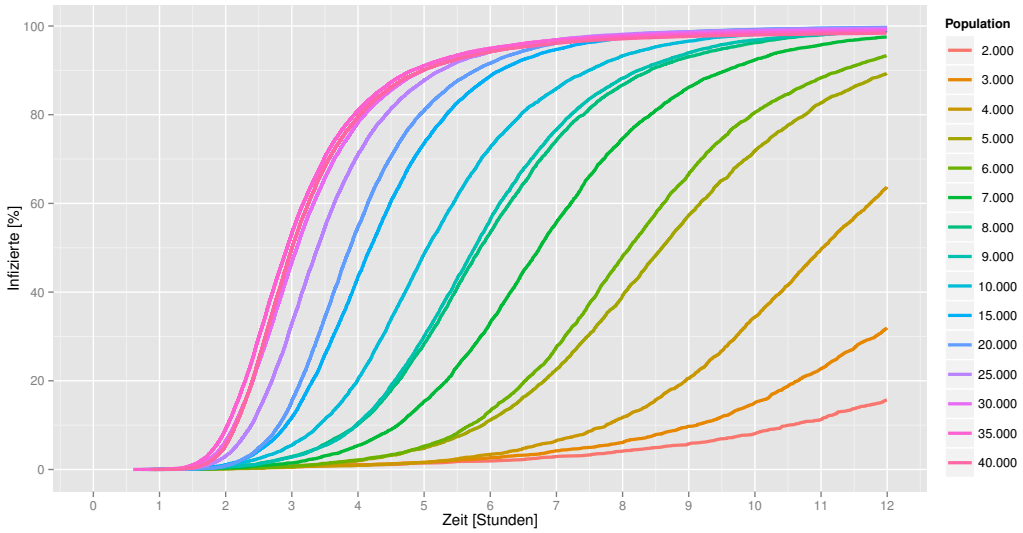


Abbildung 6.3: Parameterstudie zur Infektionsausbreitung bei veränderlicher Populationsgröße

den Effekt verschiedener Nutzungsintervalle von Smartphones in Bezug auf die in dieser Arbeit untersuchte Malware zu studieren, wurden ebenfalls Simulationen mit unterschiedlichen Aktivierungsintervallen durchgeführt. Da ein Smartphones nur bei aktiver Nutzung des Internets bzw. beim Aufbau einer neuen Verbindung anfällig für diese Art von Evil Twin-Angriffen ist, wurden bei konservativer Herangehensweise Simulationen mit Aktivierungsintervallen von 15, 30, 45 und 60 Minuten durchgeführt. In Abbildung 6.4 sind die Ergebnisse dieser Simulation dargestellt. Man kann beobachten, dass für lange Aktivierungsintervalle von mehr als 45 Minuten eine Ausbreitung vergleichsweise langsam stattfindet. Für kürzere Intervalle hingegen ergibt sich für die Ausbreitung schnell ein epidemischer Charakter mit vollständigen Infektionen innerhalb weniger Stunden.

Während in den vorangegangenen Simulation von einem geschlossenen System ausgegangen wurde, soll im Folgenden untersucht werden, welchen Einfluss ein offenes System auf die Ergebnisse hat. Hierbei können Benutzer den simulierten Bereich verlassen und stellen zwar für den betrachteten Bereich keine Gefahr mehr dar, wohl aber für angrenzende Gebiete. In Abbildung 6.5 sind die Ergebnisse dieser Simulationen dargestellt. Während beim geschlossenen System keine simulierten Benutzer den simulierten Bereich verlassen können, beginnen die simulierten Pendler nach 6 Stunden Downtown Chicago zu verlassen. Wie

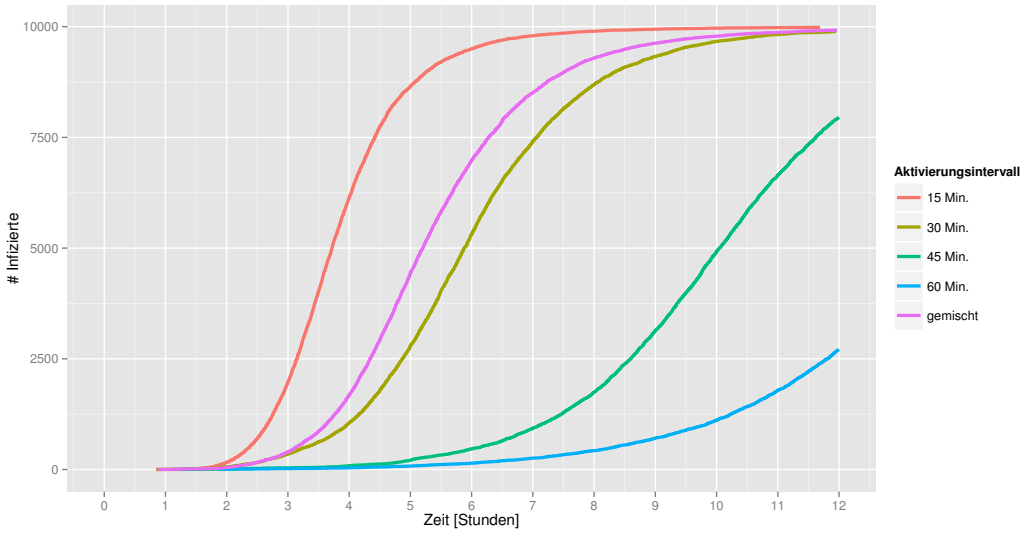


Abbildung 6.4: Infektionsausbreitung im Szenario F-D mit verschiedenen Aktivierungsintervallen der Smartphones

in vielen anderen Simulationen von geschlossenen Systemen kommt es auch hier zu einer nahezu vollständigen Infektion aller Geräte nach 9 bis 10 Stunden. Betrachtet man hingegen das offene System, so ergibt sich ein anderes Bild: Die Gesamtzahl der im Laufe der Simulation infizierten Geräte im betrachteten Bereich ist erwartungsgemäß kleiner als beim geschlossenen System. Die beiden Graphen *Infiziert verlassen* und *Gesund verlassen* zeigen, wie viele Geräte den simulierten Bereich zu einem Zeitpunkt verlassen haben. Man kann hier erkennen, dass die Ausbreitung durch das zunehmende Verlassen des simulierten Bereichs durch Pendler ab der 6. Stunde keinen epidemischen Charakter mehr hat. Bedenklich ist aber die Tatsache, dass mehr als 7.000 Geräte die Infektion aus dem betrachteten Bereich hinaustragen. Somit würde die Ausbreitung im betrachteten Bereich zwar verlangsamt, sie würde sich aber mit größerer räumlicher Ausbreitung fortführen.
Um zu untersuchen welchen Einfluss die initiale Akkuladung der Geräte auf die Ausbreitung hat, soll im Folgenden auf den Energieverbrauch der mobilen Geräte eingegangen werden. Die Akkulaufzeiten moderner Smartphones liegen bei normaler bis intensiver Nutzung im Schnitt bei einem Tag; bei seltener Nutzung kann eine Akkuladung hingegen auch zwei Tage oder mehr ausreichen. Durch die in Kapitel 4 vorgestellte Malware wird der Energieverbrauch deutlich gesteigert. Daher wird in folgender Simulation betrachtet, wie sich die

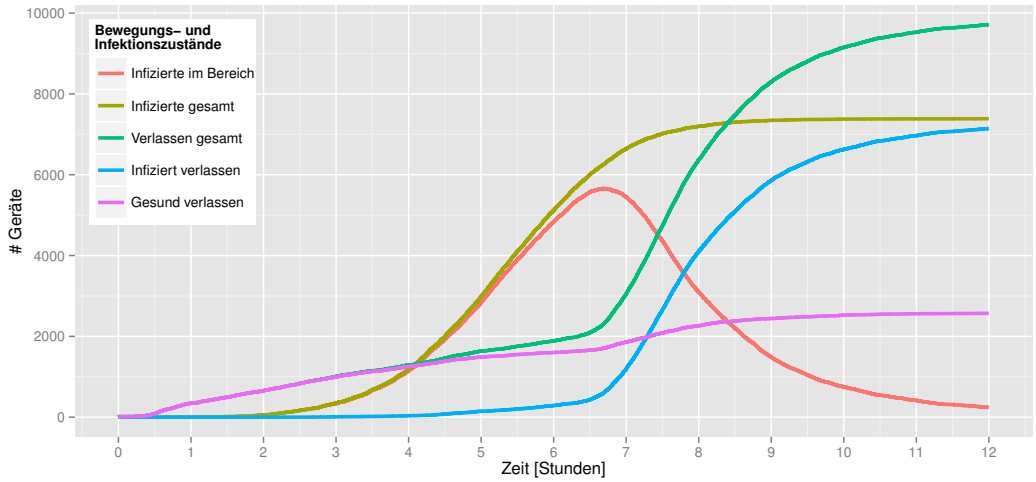


Abbildung 6.5: Simulation eines geschlossenen System im Vergleich zu einem offenen System

Ausbreitung entwickelt, wenn nicht davon ausgegangen wird, dass jedes Gerät zu Beginn eines Tages mit voller Akkuladung in den Tag startet. In diesen Simulationen schalten sich diverse Geräte im Laufe des Tages selbstständig aus. Diese Geräte werden in der Simulation nicht mehr berücksichtigt. Sie behalten den Infektionsstatus, den sie beim Abschalten hatten und können selbst keine weiteren Geräte mehr infizieren. In Abbildung 6.6 sind die Ergebnisse der entsprechenden Simulationen dargestellt. In den Simulationen wurden die initialen Akkuladungen mit einer Gauss-Verteilung im jeweiligen Bereich festgelegt. Es ist festzustellen, dass die initiale Akkuladung der einzelnen Geräte keinen erheblichen Einfluss auf die Charakteristik der Ausbreitung hat. Auch bei Akkuverteilungen mit sehr niedrigen Startwerten werden dennoch große Teile der Population (in allen Fällen mehr als zwei Drittel) infiziert.

Auf den in Abbildung 6.7 gezeigten Karten sind die Orte der Infektionen farblich markiert. Für jede Infektion wurde hierbei ein Punkt gezeichnet. Insgesamt ergibt sich hieraus eine sog. Heatmap, in denen Bereiche mit vielen Infektionen hell und Bereiche mit wenigen oder keinen Infektionen dunkel dargestellt sind. Während in Abbildung 6.7a das Ergebnis einer Simulation ohne die Modellierung spezieller Orte wie Cafés dargestellt ist, sind in Abbildung 6.7b alle oben beschriebenen Eigenschaften und Funktionen modelliert und enthalten. Es ist deutlich zu erkennen, dass in der linken Karte das Straßennetzwerk eindeu-

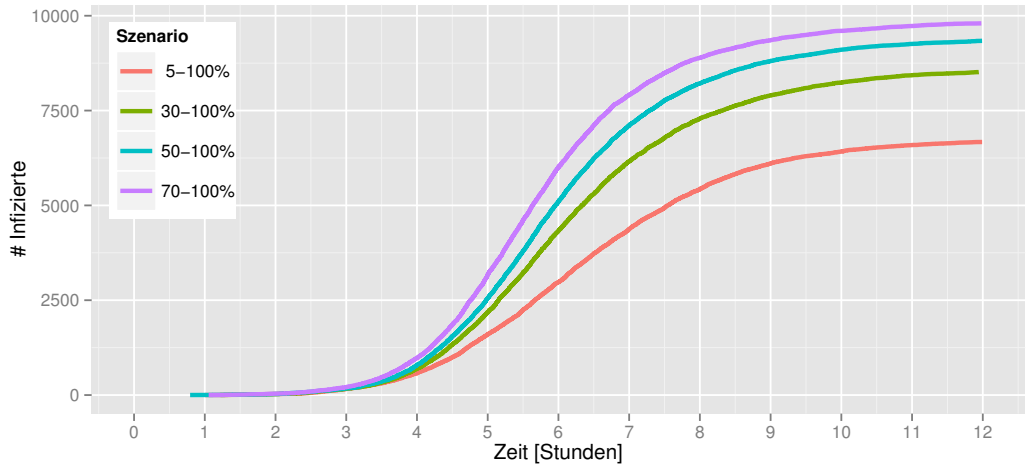
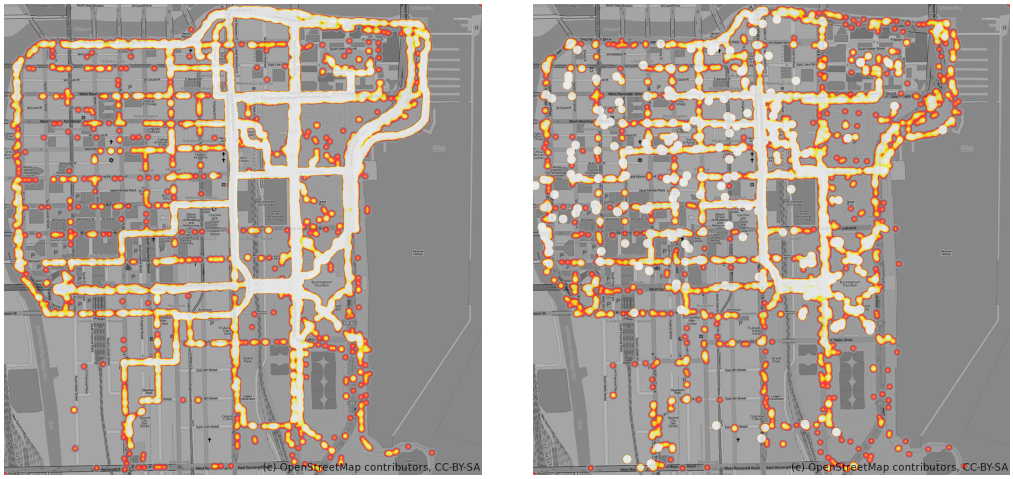


Abbildung 6.6: Infektionen bei der Simulation von 10.000 Geräten bei unterschiedlichen initialen Akkuladungen zu Beginn des simulierten Tages

tig als Ort für Infektionen hervorsticht. In der zweiten Karte hingegen sieht man den Einfluss spezieller Orte wie Cafés bzgl. des Risikos einer Infektion. Sehr viele Bereiche in und um die modellierten Cafés weisen hier die höchsten Infektionsraten auf.

6.8 Zusammenfassung

In diesem Kapitel wurde die in Kapitel 5 vorgestellte mobile Malware hinsichtlich ihres Verbreitungsverhaltens evaluiert. Es konnte im Rahmen von Simulationen gezeigt werden, dass eine derartige Malware unter den vorgestellten Bedingungen innerhalb weniger Stunden zu einer nahezu vollständigen Infektion der betroffenen Population führen kann. Es ist davon auszugehen, dass die Rahmenbedingungen für derartige Angriffe in Zukunft die Ausbreitung eher begünstigen als einschränken werden. Auch die verschiedenen Betrachtungen weiterer unterschiedlicher Ausgangsbedingungen haben das Ergebnis hinsichtlich des Risikos nicht positiv beeinflusst. Durch die Simulation spezieller Orte, an denen sich Menschenansammlungen verschiedener Größen bilden, konnte gezeigt werden, dass das Infektionsrisiko an diesen Orten erwartungsgemäß besonders hoch ist.
Um die Evaluation durchzuführen wurde zunächst beschrieben, wieso auf Simulationen bei der Entwicklung derartiger Sicherheitssysteme nicht verzichtet



(a) 10.000 Personen, Infektionen nur im Straßenentzwerk

(b) 10.000 Personen, alle Infektionsmöglichkeiten

Abbildung 6.7: Geografischer Vergleich der Infektionen bei der Simulation mit und ohne modellierten Lokalitäten

werden kann. Ziel einer solchen Untersuchung ist die Erprobung und Auswertung verschiedener Konfigurationen und Ausgangssituationen des Gesamtsystems. Es konnte mit Hilfe der Simulationen gezeigt werden, dass realistische Bewegungsmodelle einen erheblichen Einfluss auf die Ergebnisse der Simulationen haben. Durch das speziell für die Simulation der Malware entwickelte Infektionsmodell konnte darüber hinaus gezeigt werden, dass eine möglichst detailgetreue Modellierung der Infektion ebenfalls große Auswirkungen auf die Ergebnisse hat. Insbesondere konnte beobachtet werden, dass die Auswirkungen realistischer und detaillierter Infektionsmodelle einen größeren Einfluss auf die Ergebnisse haben als die zuvor eingeführten realistischeren Bewegungsmodelle.

Kapitel 7

Mechanismen zum Schutz vor Evil Twin Access Points

In diesem Teil der Arbeit wird zunächst dargestellt, welche Anforderungen an ein Schutzsystem bestehen und wie die zum Teil konträren Interessen bei der Verwendung eines solchen Systems zustande kommen. Es wird beschrieben, wie im Rahmen einer Studie die zur Entwicklung und Konfiguration notwendigen Daten erhoben wurden und wie mit Hilfe dieser Daten ein Schutzsystem gegen Evil Twin Access Points entwickelt wurde. Das *Mobile Evil Twin Detection System* (METDS) ist ein kontextbasiertes System, das Evil Twin Access Points während des Verbindungsprozesses aufspürt und hierbei vollständig autark arbeitet. Insbesondere wird keinerlei weitere Infrastruktur für den Betrieb des Systems benötigt. Teile dieses Kapitels basieren auf einer wissenschaftlichen Veröffentlichung [65] im Rahmen der internationalen Konferenz *Financial Cryptography and Data Security 2015*.

7.1 Anforderungen an ein Schutzsystem

Im Folgenden soll zunächst auf generelle Anforderungen bei der Entwicklung von Schutzmechanismen eingegangen werden. Das Spannungsfeld zwischen Sicherheit, Benutzbarkeit und Privatsphäre wird zunächst erläutert. Im zweiten Teil wird auf konkrete Anforderungen an ein Sicherheitssystem eingegangen, die sich in großen Teilen aus existierenden Lösungen und deren Unzulänglichkeiten ergeben.

7.1.1 Spannungsfeld bei der Entwicklung von Schutzmechanismen

Um Benutzer vor digitalen Gefahren im Allgemeinen und Evil Twin-Angriffen im Speziellen zu schützen, muss eine Abwägung verschiedener Interessen vorgenommen werden. Diese lassen sich grob in die drei Kategorien Sicherheit, Benutzbarkeit und Privatsphäre aufteilen. In Abbildung 7.1 sind diese drei Faktoren und ihre Interessenbereiche dargestellt. Ein System zum Schutz des Benutzers muss diese drei Faktoren berücksichtigen, um vom Endnutzer akzeptiert und genutzt zu werden. Das System soll grundsätzlich Sicherheit gegenüber Gefahren – hier im Speziellen Angreifern – bieten. Ein weiterer wichtiger Faktor ist die Benutzbarkeit eines solchen Systems. Ein Sicherheitssystem kann einem Anwender nur dann Schutz bieten, wenn es für den Endnutzer ausreichend einfach bedienbar ist. Andernfalls würde ein System vom Benutzer ignoriert oder schlimmsten Falls sogar deaktiviert werden. Die potentiell gute Schutzleistung eines solchen Systems wäre wirkungslos. Bietet das System Sicherheit und ist dabei auch durch Endanwender benutzbar, so kommt der dritte Faktor ins Spiel: die Sicherung der Privatsphäre. Ein Sicherheitssystem braucht Eingangsdaten, welche analysiert und ausgewertet werden. In diesen Eingangsdaten sind im Falle von Daten zur Erkennung von Angriffen auf IT-Systeme oftmals personenbezogene und andere schützenswerte Daten enthalten. Ein konkretes Beispiel für ein besonders häufiges personenbezogenes Datum in derartigen Daten sind IP-Adressen. Aber auch bei der Erkennung von Evil Twin Angriffen sind in den Kontextdaten schützenswerte Informationen, wie SSIDs oder BSSIDs enthalten sein.

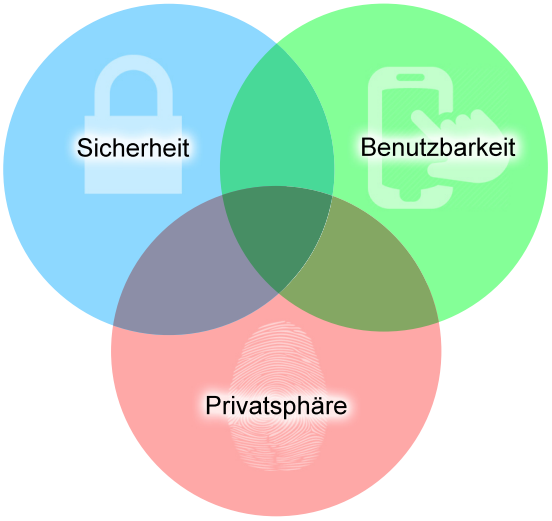


Abbildung 7.1: Spannungsfeld zwischen Sicherheit, Benutzbarkeit und Privatsphäre bei der Entwicklung von Schutzmechanismen

Im Folgenden sollen noch einmal die drei Faktoren beschrieben und ihre Überlappung mit den jeweils anderen Faktoren erläutert werden. Zunächst soll auf die drei Extrema dieses Systems eingegangen werden:

Beachtet man im Entwicklungsprozess eines solchen Sicherheitssystems nur den Faktor Sicherheit und lässt Anforderungen bezüglich der beiden anderen Faktoren außer acht, so kann ein durchaus sicheres System entstehen. Die Kosten auf Seiten der anderen beiden Faktoren dürften den Nutzen in diesem Fall aber übersteigen. Hinsichtlich der Privatsphäre wäre an dieser Stelle auch ein System denkbar, das Verbindungsdaten vieler Benutzer ungefiltert an ein zentrales System übermittelt. Hier könnten anhand der gesammelten Informationen Analysen durchgeführt und die Ergebnisse zurück an die entsprechenden Benutzer geliefert werden. Gerade die ungefilterte Weitergabe von personenbezogenen Daten birgt stets die große Gefahr, das private Daten in die Hände unbefugter Dritter gelangen. In Bezug auf die Benutzbarkeit eines solchen Systems können ebenfalls Probleme entstehen. So sind auch Systeme mit einer hohen Falsch-positiv-Rate denkbar. Auf diese Weise entgeht dem System zwar kein echter Angriff, der Benutzbarkeit und Akzeptanz eines solchen Systems ist dies hingegen nicht zuträglich.

Beachtet man bei der Entwicklung vorrangig den Faktor Benutzbarkeit, so ergeben sich zum einen die selben Probleme bezüglich der Privatsphäre, wie im vorherigen Beispiel. Darüber hinaus kann die Basisfunktion des Systems, die Sicherheit, unter einem solchen Vorgehen leiden. In einem auf diese Weise gestalteten System hätte die Benutzbarkeit höchste Priorität. Warnungen sind einem Benutzer nur dann anzuzeigen, wenn es keinerlei andere Möglichkeiten mehr gibt, Schaden vom Benutzer abzuwenden. In den meisten Fällen kann eine hundertprozentige Sicherheit in Bezug auf das Vorhandensein einer realen Gefahr nicht erzielt werden. Warnungen zu derartigen Vorfällen würden in einem solchen System nicht angezeigt werden, was die Gesamtsicherheit deutlich einschränkt.
Auch beim dritten Extrem, der reinen Ausrichtung auf die Sicherung der Privatsphäre, ist davon auszugehen, dass die beiden verbleibenden Faktoren unter einer sehr einseitigen Ausrichtung leiden werden. In diesem Fall scheinen die Auswirkungen hingegen nicht ganz so gravierend auszufallen. So ist ein Sicherheitssystem denkbar, welches bei der Analyse ausschließlich auf lokal vorhandene Daten setzt. Es verlassen entsprechend keinerlei Daten das lokale System. Darüber hinaus müssen auch zur Analyse keinerlei weitere Daten mit anderen Servern ausgetauscht werden. Ein solches System ist in Bezug auf die Sicherung der Privatsphäre vorbildlich. Je nach Gestaltung des Systems müssen auf Seiten der Sicherheit nur wenige Einbußen in Kauf genommen werden. Auch der Faktor Benutzbarkeit hängt seinerseits stark von der Gestaltung des Sicherheitssystems ab und wie gut das System mit ausschließlich lokal gesammelten und vorgehaltenen Daten Angriffe erkennen und vor ihnen warnen kann.
Das Ziel bei der Entwicklung des Sicherheitssystems im Rahmen dieser Arbeit ist es, alle drei Faktoren hinreichend zu berücksichtigen. Je nach Gewichtung der Einzelfaktoren kann sich zwar eine Tendenz in Richtung eines Faktors ergeben, trotzdem sollen stets alle Faktoren berücksichtigt werden. Es soll eine Lösung gefunden werden, die im Bereich der Überlappung aller Faktorenfelder in der Mitte von Abbildung 7.1 anzusiedeln ist.

7.1.2 Weitere funktionale Anforderungen

Neben den beschriebenen Faktoren, welche die Akzeptanz und Sicherheit des Systems maßgeblich beeinflussen, gibt es weitere funktionale Anforderungen,

die sich aus existierenden Lösungen und bestehenden Problemen dieser ergeben:

- **Keine zusätzliche oder angepasste Hardware:** Um den Einsatz und die Verbreitung des Systems positiv zu beeinflussen, ist eine Lösung zu bevorzugen, die keine zusätzliche Hardware benötigt. Auf diese Weise können sowohl die Kosten als auch der Aufwand einer Einführung des Systems minimiert werden.

- **Keine Online-Verbindungen:** Um den Mechanismus sicher gegen direkte Angriffe auf das Schutzsystem selbst zu realisieren, soll ein System entstehen, welches keinen Zugriff auf weitere Online-Dienste erfordert.

- **Mobil einsetzbar / energieeffizient:** Im Gegensatz zu vielen existierenden Lösungen soll das entstehende Sicherheitssystem auf mobilen Geräten einsetzbar sein. Es soll hierbei nicht nur auf Laptops, sondern insbesondere für Tablets und Smartphones geeignet sein. Aus diesem Grund muss bei der Entwicklung verstärkt auf die Energieeffizienz des Systems geachtet werden.

- **Schnelle Erkennung:** Während viele existierende Systeme erst während der Nutzung einer Verbindung Prüfungen durchführen und so potentiell gefährliche Access Points aufspüren, soll das im Rahmen dieser Arbeit entstehende Sicherheitssystem erst dann die Nutzung der Verbindung erlauben, wenn diese hinreichend evaluiert und als sicher eingestuft worden ist.

- **Erkennung ausschließlich über lokale Sensorik:** Es sollen zur Erkennung bösartiger Access Points nur lokal verfügbare Sensoren von Smartphones und Tablets eingesetzt werden. Sowohl relevante und brauchbare, wie auch nicht nutzbare Parameter müssen hierfür gefunden und hinsichtlich ihrer Eignung bewertet werden.

- **Benutzerfreundlichkeit:** Der bereits oben beschriebene Faktor der Benutzerfreundlichkeit zieht weitere funktionale Anforderungen nach sich. Während in einigen existierenden Sicherheitssystemen eine Nutzerinteraktion bei jeder aufzubauenden Verbindung benötigt wird, soll im hier

entstehenden System im positiv bewerteten und ungefährlichen Fall keinerlei Interaktion vonnöten sein. Nur bei einer als potentiell gefährlich eingestuften Situation soll der Benutzer informiert werden.

- **Evaluation:** Viele in Abschnitt 3.1 erwähnten Lösungen wurden nur unzureichend evaluiert. Aus diesem Grund soll das im Rahmen dieser Arbeit entstehende System mit realistischen Daten evaluiert werden.

Die beschriebenen Faktoren im Spannungsfeld in Kombination mit den beschriebenen funktionalen Anforderungen geben die weiteren Schritte zur Entwicklung eines Schutzsystems vor. In einem ersten Schritt werden im nächsten Abschnitt diejenigen Kontextparameter ermittelt, die bei der Erkennung bösartiger Access Points hilfreich sind.

7.2 Sammlung und Verifizierung vorhandener Kontextdaten

Für die Erkennung von bösartigen Access Points sollen im Rahmen dieser Arbeit verschiedenste Parameter des Kontextes genutzt werden. All diese Parameter müssen mit einem handelsüblichen, aktuellen Smartphone bestimmt werden können. Um untersuchen zu können, welche Parameter für die Erkennung von Angriffen geeignet sind, wurde im Vorfeld der Entwicklung eine Feldstudie durchgeführt, in der möglichst viele potentiell relevante Parameter durch die Teilnehmer der Studie gesammelt wurden. Die Bestimmung der Parameter welche für die Erkennung geeignet und nutzbar sind, ist nicht Teil dieser initialen Studie gewesen. Diese Entscheidung wird zusammen mit einer Diskussion der entsprechenden Parameter später in diesem Kapitel folgen.
Um zu evaluieren, ob die Sammlung verbindungsrelevanter Parameter auf Smartphones möglich und sinnvoll ist, wurde für die Studie eine mobile App für Android entwickelt. Mit Hilfe der App konnten im Studienverlauf Daten zu mehr als 220.000 WLAN-Verbindungen gesammelt werden. Ein weiterer Teil der Studie bestand aus einem Fragebogen zur Selbsteinschätzung der Teilnehmer bezüglich ihres Nutzungsverhaltens. Die Ergebnisse dieser Befragung werden in einem weiteren Abschnitt mit den durch die App gesammelten Da-

ten verglichen. Abschließend werden die relevanten Ergebnisse der Analyse dargestellt und diskutiert.

7.2.1 Feldstudie und mobile App

Neben der Sammlung von Verbindungsdaten soll anhand der gesammelten Verbindungs- und Konfigurationsdaten der Teilnehmer ebenso gezeigt werden, wie groß das Gefahrenpotential von Evil Twin-Angriffen in heutigen Hotspot-Infrastrukturen ist.
Die Studie wurde im Zeitraum von Dienstag, dem 17. Dezember 2013 bis Freitag, dem 28. Februar 2014 durchgeführt. In dieser Zeit konnten sich Teilnehmer für die Studie anmelden und die mit ihren Geräten gesammelten Daten zur Verfügung stellen. An der Feldstudie haben insgesamt 92 Smartphone-Nutzer teilgenommen. Die Studien-App wurde für Android entwickelt, da hier der Zugriff auf Verbindungsdaten und Netzwerkkonfigurationen im Gegensatz zu anderen mobilen Betriebssystemen wie iOS problemlos möglich ist. Alle Teilnehmer der Studie mussten die App auf ihrem Smartphone installieren und den Hintergrunddienst durch die App starten. Rekrutiert wurden die Teilnehmer über verschiedene Medien, wie Mailinglisten für Studieninteressierte der DCSec-Gruppe und der Verbreitung in sozialen Netzwerken.
Als Anreiz für die Teilnahme an der Studie erhielten alle Teilnehmer die Chance auf einen von fünf Gutscheinen für ein Online-Versandhaus. Da neben der Sammlung einzelner Konfigurationsdaten auch das Nutzungsverhalten über einen längeren Zeitraum beobachtet werden sollte, wurde das Gewinnspiel dahingehend angepasst, dass sich die Chance auf einen Gewinn erhöht, je länger man an der Studie teilnimmt und Daten an einen zentralen Dienst überträgt. Die Teilnehmer konnten während des Studienzeitraums stets mit Hilfe der App einsehen, wie viele Daten bereits gesammelt wurden und wie groß die Nutzungsdauer der App und des integrierten Dienstes ist.
In Abbildung 7.2 ist die Architektur der Studien-App dargestellt. Im linken Bereich sind alle Schnittstellen dargestellt über die Daten zur aktuellen Verbindung abgerufen werden. Der *Connection Recording Service* übernimmt hierbei die Steuerung der Informationssammlung und -verwaltung während jedes Verbindungsaufbaus mit einem WLAN. Zum Teil wird bei der Sammlung bestimmter Daten zusätzlich auf Receiver zurückgegriffen. Receiver sind in An-

droid Klassen, die sich gegenüber dem System registrieren und auf diesem Weg über bestimmte Systemereignisse informiert werden. Beispielsweise wird der *WiFiStatusReceiver* benötigt, um als Dienst über einen abgeschlossenen Verbindungsaufbau der WLAN-Schnittstelle informiert zu werden und der *NearbyWiFisListReceiver*, um die Ergebnisse eines WiFi-Scans entgegenzunehmen. Der dritte Receiver ist der *BootUpReceiver*. Er sorgt dafür, dass der Dienst nach dem Neustart des mobilen Geräts automatisch ebenfalls neu gestartet und aktiviert wird. Der *Watchdog Service* überwacht während des laufenden Betriebs, ob der Dienst läuft und korrekt arbeitet. Sollte dies zu einem Zeitpunkt nicht der Fall sein, so kann er den Dienst neu starten. Neben der grafischen Oberfläche des Dienstes und des eingebundenen Fragebogens sind noch Elemente zur Konfiguration und zur Auswertung der Laufzeit Bestandteil des Gesamtsystems. Zur Datenspeicherung nutzt der Dienst zum einen die lokale *Context DB*, auf die über den entsprechenden *Persistence Layer* zugegriffen wird. Zum anderen ist der *Sync Service* dafür zuständig, die gesammelten und aggregierten Daten in regelmäßigen Abständen an einen zentralen Service mit angebundener Datenbank zu senden. Der zentrale Server hierfür befindet sich im Netzbereich der Distributed Computing & Security Group (DCSec).

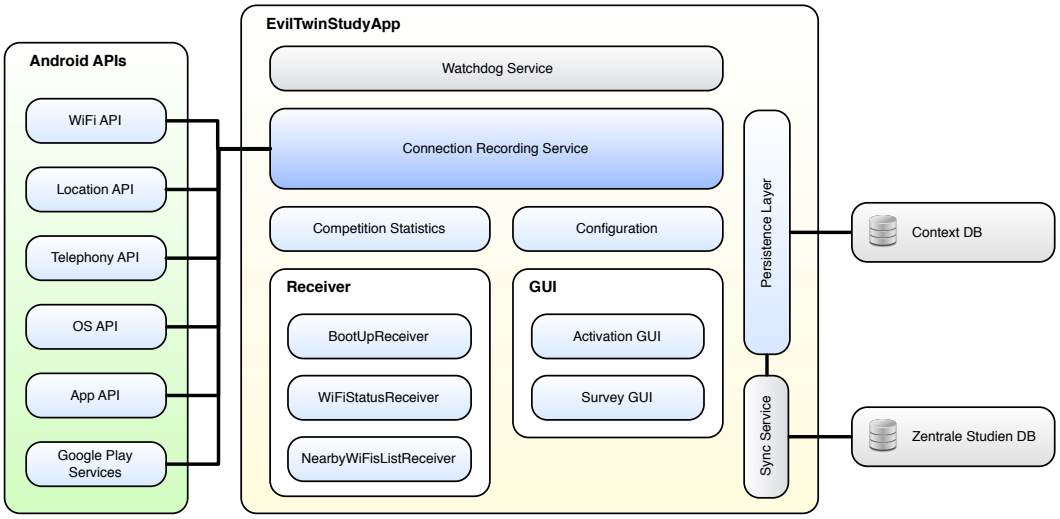


Abbildung 7.2: Gesamtarchitektur der Android-App zur Feldstudie und zur Sammlung von Kontextdaten

Der Synchronisations-Dienst der App hat in regelmäßigen Abständen die neu gesammelten Informationen an einen zentralen Service zur Speicherung aller in

der Studie gesammelten Daten übermittelt. Der Transport erfolgte hierfür zum Schutz der zum Teil persönlichen Daten SSL-verschlüsselt und wurde zusätzlich per SSL-Pinning gegen MITM-Angriffe geschützt. In der zentralen Datenbank wurden die Verbindungsdaten aller Teilnehmer gespeichert. Eine Beschreibung der gesammelten Kontextdaten findet sich im folgenden Abschnitt.

7.2.2 Gesammelte Verbindungs- und Kontextdaten

Mit Hilfe der App wurde die WLAN-Nutzung der Teilnehmer bis zu 74 Tage lang protokolliert. Im Durchschnitt haben die Teilnehmer hierbei 46,75 Tage an der Studie teilgenommen und für den gleichen Zeitraum Daten über ihr WLAN-Nutzungsverhalten aufgezeichnet. Ebenso wurden Kontextdaten bei jedem Verbindungsaufbau zu einem WLAN ermittelt, gespeichert und zu einem späteren Zeitpunkt an den zentralen Dienst übermittelt. Von den insgesamt 92 Teilnehmern der Studie haben 83 neben der Datensammlung durch den oben beschriebenen Dienst auch einen Fragebogen ausgefüllt, mit dem eine Selbsteinschätzung des WLAN-Nutzungsverhaltens und einige demografische Daten abgefragt wurden. Insgesamt konnten auf diese Weise mehr als 220.000 Verbindungen, Netzwerkumgebungen und weitere Metadaten zu WLAN-Verbindungen gesammelt werden, die in folgenden Arbeitsschritten zu einer Bewertung des Risikos für Evil Twin Angriffe und zur Analyse hinsichtlich eines Schutzsystems genutzt werden.
Die in der Studie verwendete App hat sämtliche Verbindungen zu Funknetzwerken überwacht und jeweils direkt nach einem erfolgreichen Verbindungsaufbau die folgenden Daten über entsprechende Schnittstellen ermittelt:

- Aktueller Zeitstempel
- Eindeutige Identifikation der Installation und des Geräts
- App-Version
- Hersteller und Modell des Smartphones
- Verbundenes Netzwerk
  - SSID
  - BSSID
  - Signalstärke (RSSI)
  - Unterstützte Geschwindigkeit des Netzwerks

- Netzwerke in der Umgebung
  - SSID
  - BSSID
  - Signalstärke (RSSI)
  - Weitere Eigenschaften (Verschlüsselungsstandard, Teil eines Extended Service Sets)
  - Frequenz
- Aktuelle Position
  - Location-Provider
  - Zeitstempel der Position
  - Breitengrad
  - Längengrad
  - Genauigkeit der Positionsbestimmung
- Status des Bildschirms (An / Aus)
- Sperrstatus des Geräts (An / Aus)
- Akkuladung
- Status der SIM-Karte
- Zellen-ID und Location Area Code der Mobilfunkverbindung
- Konfigurierte Netzwerke auf dem Gerät
  - SSID
  - Schlüsselmanagement des Netzwerks
  - Group Cipher

Im Folgenden sollen einige der wichtigsten, gesammelten Parameter näher erläutert werden. Neben zur Auswertung erforderlichen Metadaten wie einem aktuellen Zeitstempel, einer genauen Identifikation eines Geräts wird ebenso die Version der App gespeichert, um evtl. Verbesserungen von einer App-Version zur nächsten nachvollziehen und bewerten zu können. Der erste für ein späteres Erkennungssystem wichtige Parameter ist das derzeit verbundene Netzwerk. Es werden alle über die API zur Verfügung stehenden Parameter abgerufen und gespeichert. Ein weiterer erfasster Parameter sind die aktuellen WLANs

in der Umgebung des Smartphones zum Zeitpunkt des Verbindungsaufbaus. Hierbei wird nicht nur die SSID des jeweiligen Netzwerks gesichert, sondern auch weitere Parameter wie die BSSID, die Signalstärke, die Frequenz auf der der Access Point arbeitet und die unterstützten Verschlüsselungsstandards. Auch die aktuelle Position wird mit allen verfügbaren Daten gespeichert. Entsprechend wird nicht nur die Position selbst gespeichert, sondern ebenfalls die über die Schnittstelle verfügbare Genauigkeit der Position. In der Schnittstellenbeschreibung [21] heißt es zur Genauigkeit einer Position unter Android:

> "We define accuracy as the radius of 68% confidence. In other words, if you draw a circle centered at this location's latitude and longitude, and with a radius equal to the accuracy, then there is a 68% probability that the true location is inside the circle."

Der reale Aufenthaltsort liegt entsprechend mit einer Wahrscheinlichkeit von 68% in dem durch die Genauigkeit aufgespannten Kreis. Da neben dieser Art der Beschreibung des eigenen Standortes keine weiteren verfügbar sind, wird diese auch im Folgenden für die Bestimmung und Speicherung der Position verwendet, wann immer dies erforderlich ist. Für die nachfolgende Auswertung ist der sog. Provider ebenfalls entscheidend. Durch den Provider wird angegeben, mit Hilfe welcher Technik die Position bestimmt worden ist. Mögliche Provider sind die folgenden:

GPS Die Position wurde ausschließlich über das GPS-Modul des mobilen Geräts bestimmt.

Network Die Position wurde über das Mobilfunknetz in Verbindung mit Assisted GPS und WLANs bestimmt. Im Vergleich zur reinen Ortung per GPS ist dieses Verfahren nicht nur schneller sondern auch deutlich energiesparender.

Passive Es wurde kein Verfahren zur Bestimmung der Position aktiv genutzt bzw. angestoßen. Die Position kann in diesem Fall nur ermittelt werden, wenn eine andere App oder ein Systemdienst die Position bestimmen möchte. In einem solchen Fall wird diese auch passiv durch diesen Provider genutzt. Da in der Studien-App die aktuelle Position bestimmt werden soll, wird die Positionsbestimmung aktiv in Gang gesetzt und Positionen dieses Providers treten nicht auf.

Fuse Bei diesem Provider werden die Google Play Services [22] zur Bestimmung der Position verwendet. Sie nutzen eine Kombination aller anderen Provider und nutzen die generelle Verfügbarkeit im gesamten System zur Steigerung der Genauigkeit und zur Minimierung des Energieverbrauchs aus. Dieser Provider wird bei vorhandenen Google Play Services auf dem mobilen Gerät durch die Studien-App bevorzugt.

Weitere erfasste, gespeichert und übermittelte Parameter sind verschiedene Statusinformationen des Displays, des Akkus und der Mobilfunkverbindung. Auch die auf dem mobilen Gerät konfigurierten Funknetzwerke werden gespeichert. Hierbei werden wie bei den Netzwerken, die durch den Scan der Umgebung gefunden werden, alle verfügbaren Informationen gespeichert. Dies umfasst neben der SSID auch das Schlüsselmanagement und weitere Informationen wie beispielsweise die Group Cipher Suite, also das Verfahren welches der Access Point für alle verbundenen Stationen nutzt. Möglich sind hier beispielsweise das Temporary Key Integrity Protocol (TKIP), WEP oder auch das *Counter Mode mit Cipher Block Chaining Message Authentication Code Protocol* (CCMP) [41].

Alle im Rahmen dieser Studie gesammelten Daten können aufgrund der Vielzahl personenbezogener Daten nicht öffentlich zur Verfügung gestellt werden. Sie befinden sich in der Abgabeversion dieser Dissertation im beigefügten elektronischen Verzeichnis im Ordner *Kontext_ Studie*.

7.2.3 Fragebogen

Um am Gewinnspiel teilnehmen zu können mussten die Benutzer der App einen Fragebogen ausfüllen, welcher in die App integriert ist. Hierbei wurden zum einen demografische Daten abgefragt, aber auch eine Selbsteinschätzung der Benutzer hinsichtlich ihrer WLAN-Nutzung und ihrer Fähigkeiten in IT-bezogenen Fragestellungen verlangt. Insgesamt haben 83 Teilnehmer an der Befragung teilgenommen und jeweils alle Fragen beantwortet.
In Abbildung 7.3 ist die Altersverteilung der Studienteilnehmer dargestellt. Man sieht, dass die Altersgruppe der 21-30 jährigen mit knapp 64% mit Abstand am stärksten vertreten ist. Die Altersgruppe der 0-20 Jährigen ist mit knapp 11%, die Gruppe der 31-40 Jährigen mit knapp 16% vertreten. Die beiden Altersgruppen der 41-50- und der 51-60-Jährigen bestehen nur jeweils aus 4,8% der Teilnehmer. Somit ergibt sich zusammenfassend ein Durchschnittsalter von 28,24 Jahren. Ein solches Ergebnis war zu erwarten, da viele der rekrutierten Teilnehmer Studenten und Absolventen sind, die der entsprechenden Altersgruppe angehören. Für die Untersuchungen im Rahmen dieser Dissertation ist dieses Ergebnis gut geeignet, da es sich bei den am häufigsten vorkommenden Altersgruppen um diejenigen handelt, bei denen auch eine besonders hohe Technikaffinität und -nutzung erwarten kann.

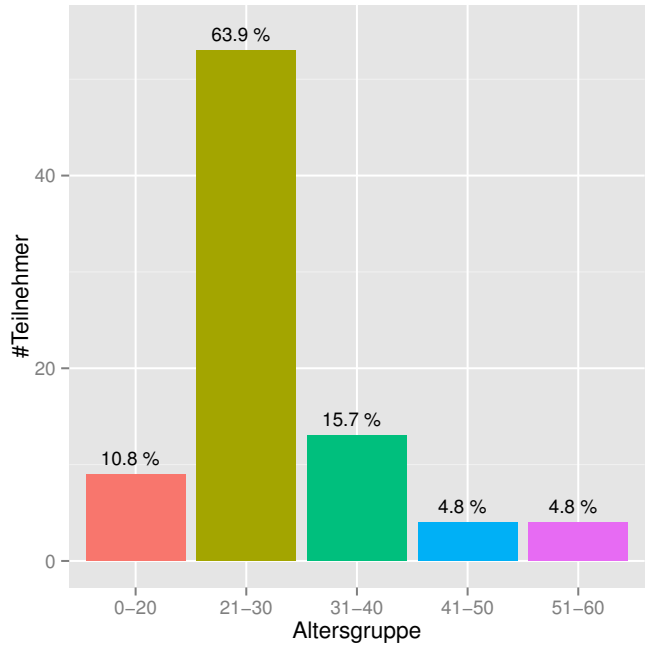


Abbildung 7.3: Altersgruppen der Teilnehmer

Abbildung 7.4 zeigt die Geschlechterverteilung der Studienteilnehmer. Mehr als 84% der Teilnehmer waren männlich. Dass diese ungleiche Verteilung negative Auswirkungen auf die Untersuchung hat, ist zu bezweifeln, da bislang keine Geschlechterunterschiede hinsichtlich der Nutzung von Smartphones und Hotspot-Netzwerken bekannt sind.

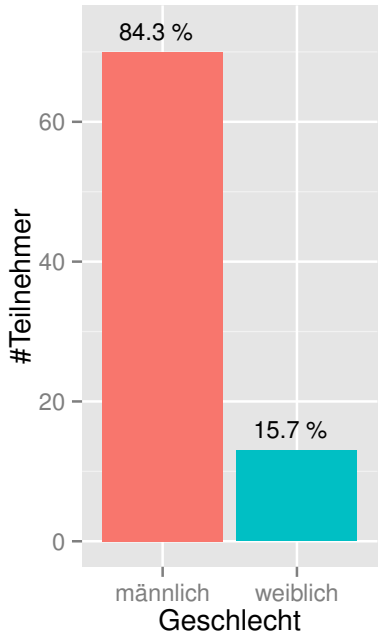


Abbildung 7.4: Geschlechterverteilung der Studienteilnehmer

Die Teilnehmer wurden ebenfalls nach ihrer derzeitigen Tätigkeit gefragt. Hierbei hat sich das folgende Bild ergeben: Mehr als die Hälfte der Teilnehmer waren Studenten. Dies liegt unter anderem an der Verbreitung der Werbung für diese Studie innerhalb der Mailingliste für Studieninteressierte. In dieser sind überwiegend Studenten enthalten. Knapp ein Drittel aller Teilnehmer arbeitet Vollzeit. Die verbleibenden 12% der Teilnehmer teilen sich in die Gruppen Schüler, Teilzeitkräfte, Selbstständige und derzeitig Erwerbslose auf.
Bei der Frage nach der Nutzungsart ihres Smartphones gaben über 72% an, das Smartphone ausschließlich privat zu nutzen. 26,5% gaben an, das Smartphone sowohl privat als auch beruflich einzusetzen. Nur einer der Teilnehmer (dies entspricht 1,2%) hat mit einem Smartphone teilgenommen, dass er ausschließlich für berufliche Zwecke nutzt.
Um einschätzen zu können, wie technikaffin die Teilnehmer der Studie sind, wurden mehrere auf diesen Aspekt abzielende Fragen gestellt. Zunächst wurden

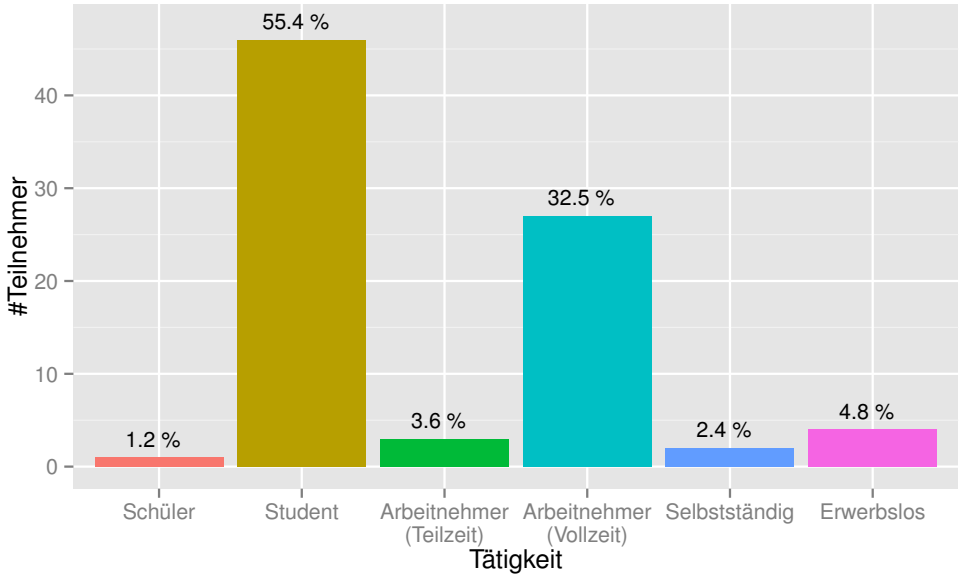


Abbildung 7.5: Tätigkeiten der Studienteilnehmer

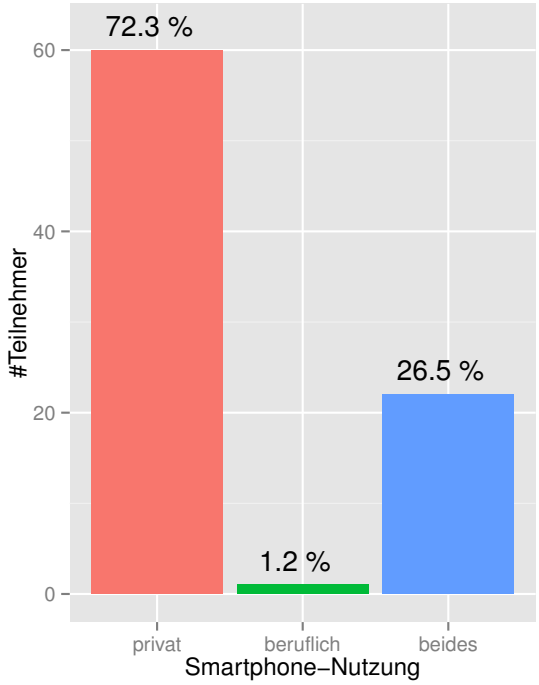


Abbildung 7.6: Nutzungsbereich des Smartphones

die Teilnehmer gefragt, ob sie jemals eine Ausbildung, ein Studium oder einen Beruf mit IT-Bezug ausgeübt haben. Die Angaben der Teilnehmer ergaben ein sehr ausgeglichenes Ergebnis. 48,2% der Teilnehmer gaben an, keine Ausbildung mit IT-Bezug begonnen oder abgeschlossen zu haben. Die verbleibenden 51,8% haben eine entsprechende Ausbildung genossen und können im Rahmen dieser Studie als etwas technikaffiner angesehen werden. Insgesamt ergibt sich an dieser Stelle ein ausgeglichenes Verhältnis, welches die reale Bevölkerung und ihre Smartphone-Benutzer gut repräsentieren kann.
Um darüber hinaus auch eine subjektive Selbsteinschätzung der Teilnehmer bezüglich ihrer Kenntnisse zu erhalten, sollten die Teilnehmer die folgenden Aussagen über sich selbst auf einer fünfstufigen Skala von „Ich stimme gar nicht zu“ bis „Ich stimme voll zu“ bewerten.

A1 „Ich habe ein sehr gutes Verständnis von Computern und dem Internet.“
A2 „Ich frage häufig Andere wenn ich Computerprobleme habe.“
A3 „Andere fragen häufig mich wenn sie Computerprobleme haben.“

Die Ergebnisse dieser Bewertungen sind in Abbildung 7.7 dargestellt. Auch hier zeigt sich, dass es sich bei den Teilnehmern der Studie vermehrt um Personen handelt, die sich selbst als erfahren im IT-Umfeld bezeichnen. Fast alle Teilnehmer behaupten von sich, ein sehr gutes Verständnis von Computern und dem Internet zu haben. Eine ähnliche Verteilung ergibt sich ebenso bei den Fragen zu entsprechenden Problemstellungen. Auch hier kann man in den Angaben zu Aussage 2 und 3 erkennen, dass nur wenige Teilnehmer Freunde oder Bekannte um Rat fragen. Vielmehr werden sie häufig durch Bekannte angesprochen, um bei Problemen mit dem Computer zu helfen. Gerade diese Personen und ihr täglicher Umgang mit Technik wie Smartphones und Computern machen sie sowohl als Zielgruppe für die untersuchten Evil Twin-Angriffe und entsprechend auch für diese Studie besonders interessant.

7.2.4 Auswertung der Verbindungsdaten

Die Studie hatte eine Laufzeit von 73 Tagen. In dieser Zeit wurden von den Teilnehmern die oben beschriebenen Daten an den zentralen Service gesendet. Im Laufe dieser Zeit konnten Daten zu 223.877 Verbindungen zu WLAN-Access Points gesammelt werden. Die Daten wurden von 92 Teilnehmern der Studie

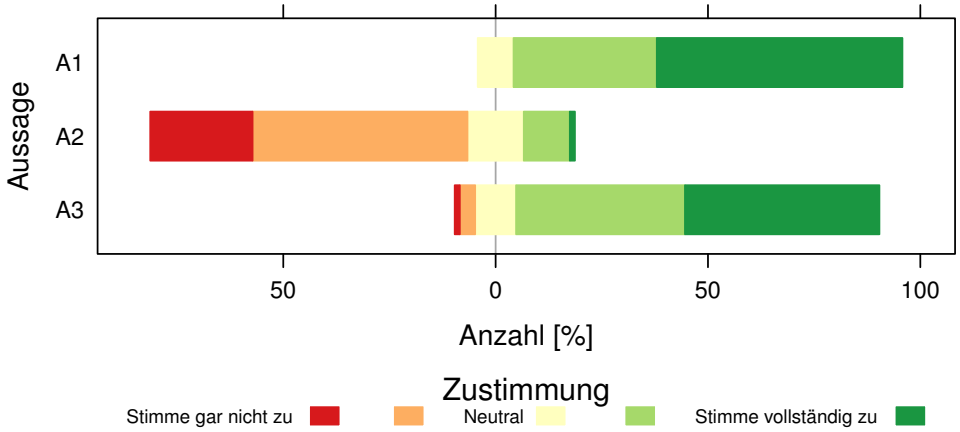


Abbildung 7.7: Selbsteinschätzung der Teilnehmer bezüglich ihrer Kenntnisse und Fähigkeiten mit Computern

gesammelt. Es wurden entsprechend durchschnittlich 2433 Verbindungen pro Benutzer aufgezeichnet. Mit Hilfe der gesammelten Daten wurde eine Vielzahl von Untersuchungen durchgeführt. Es wurde zum einen untersucht, wie groß die Bedrohung durch Evil Twin Access Points in der heutigen Zeit für Smartphone-Nutzer ist. Darüber hinaus wurde untersucht, welche Kontextdaten bei der Erkennung derartiger Angriffe eingesetzt werden können.
Eine erste Analyse befasste sich mit auf den Geräten konfigurierten Netzwerken. Diese sind wie in Kapitel 4.4 beschrieben für eine automatische Verbindung vorbereitet und bergen die beschriebenen Gefahren. Des Weiteren wurde erforscht, wie groß der Unterschied zwischen dem Nutzungsverhalten von öffentlichen Hotspots und der Menge an entsprechenden konfigurierten Netzwerken ist. Auch wurde die Tauglichkeit der verschiedenen Kontextparameter für das zu entwickelnde Erkennungssystem geprüft. Insbesondere wurden hierfür die gesammelten Positionsdaten und die aufgezeichneten WLAN-Umgebungen analysiert. Abschließend wird noch einmal auf die Eigenschaft moderner Smartphones eingegangen, sich automatisch mit bekannten Netzwerken zu verbinden.

Konfigurierte Netzwerke

Um die Gefahr, die von Evil Twin Access Points ausgeht beurteilen zu können, wurden die konfigurierten Netzwerke auf den Geräten der Teilnehmer ausgewertet. In Diagramm 7.8 sind die konfigurierten Netzwerke pro Benut-

zer dargestellt. Aufgeteilt wird die Gesamtzahl in unverschlüsselte (wie sie in öffentlichen Hotspots zum Einsatz kommen) und per WPA bzw. WPA2 verschlüsselte Netzwerke.

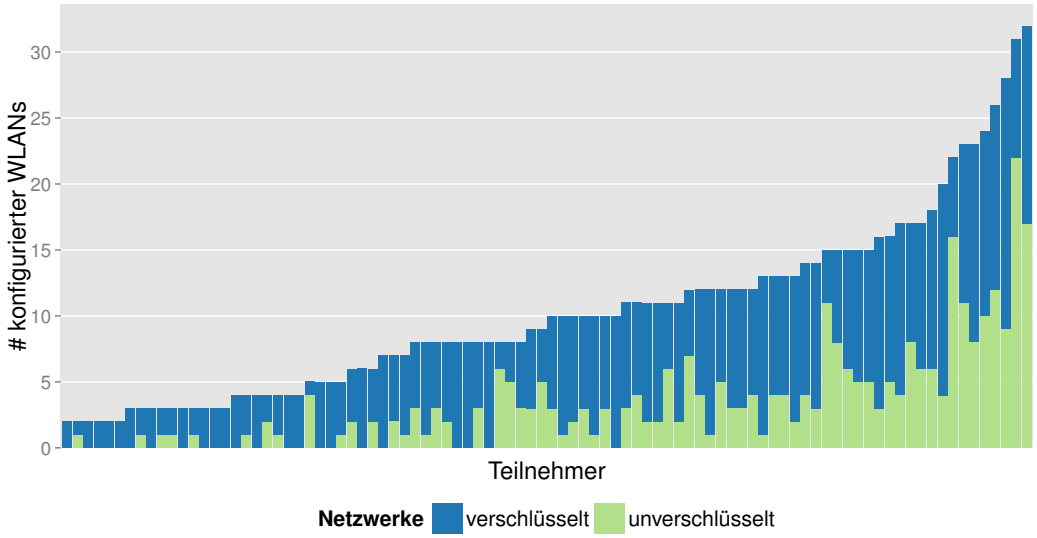


Abbildung 7.8: Anzahl und Typ konfigurierter Funknetzwerke auf den Geräten der Teilnehmer

Im Schnitt haben die Teilnehmer mehr als 10 Netzwerke auf ihren Geräten konfiguriert. Auf der rechten Seite des Diagramms finden sich Teilnehmer, die mehr als 20 und sogar bis hin zu 32 verschiedene Netzwerke auf ihrem Gerät konfiguriert haben. Bei diesen Smartphone-Nutzern kann es sich entweder um Benutzer handeln, die besonders viel und oft mit ihrem Smartphone online sind. Die hohe Anzahl an konfigurierten Netzwerken kann aber ebenso durch ein hohes Reiseaufkommen entstehen. An vielen verschiedenen Orten werden im Laufe der Zeit immer neue Netzwerke hinzugefügt, die bestehenden und nicht mehr gebrauchten aber nicht gelöscht. Auf diese Weise entsteht im Laufe der Zeit eine List mit einer beträchtlich hohen Anzahl an Netzwerken, die ein Sicherheitsrisiko im Alltag darstellen können.

Aus Sicht eines Evil Twin Angriffs ist die Tatsache entscheidend, dass mehr als 75% der Teilnehmer wenigstens ein unverschlüsseltes Netzwerk konfiguriert haben und damit potentiell gefährdet sind für Angriffe der Art, wie sie in Kapitel 4 beschrieben wurden. Einer der Teilnehmer hatte nicht weniger als

22 verschiedene unverschlüsselte Netzwerke auf seinem Gerät hinterlegt, mit denen das Smartphone automatisiert eine Verbindung eingeht.

Hotspots und Selbsteinschätzung der Teilnehmer

Neben der Anzahl an konfigurierten Netzwerken wurden im Rahmen der Studie ebenfalls die SSIDs und Verschlüsselungsmechanismen der gespeicherten Netzwerke erhoben. Diese gesammelten Daten wurden zur Untersuchung genutzt, wie gut sich die Teilnehmer der Studie selbst einschätzen konnten. Im Rahmen der Umfrage wurde durch alle Teilnehmer die Frage beantwortet, wie oft sie öffentliche Hotspots in ihrem Alltag nutzen.

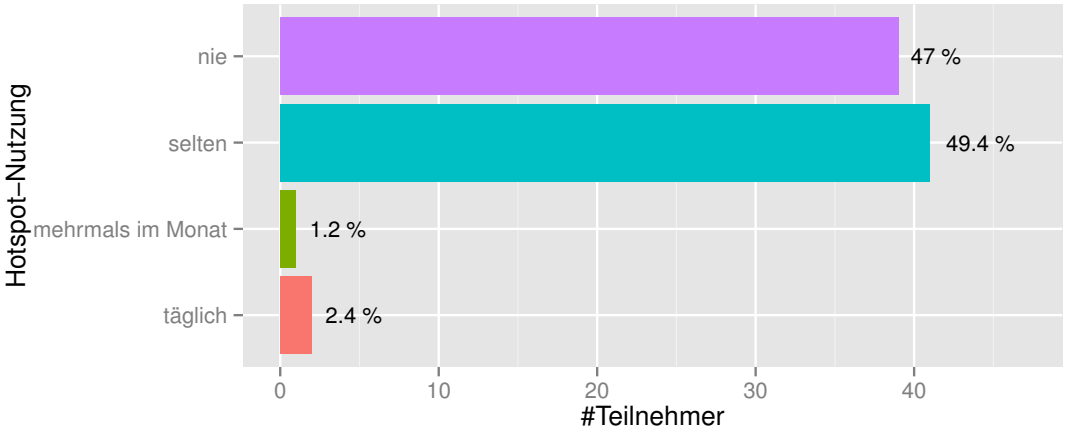


Abbildung 7.9: Selbsteinschätzung der Nutzung von Hotspots im Alltag

Weniger als 4% der Teilnehmer gaben an, dass sie öffentliche Hotspots mehr als einmal pro Monat benutzen. 49% der Teilnehmer nutzen diese Hotspots nach eigenen Angaben selten und die verbleibenden 47% gaben an, keinerlei Hotspots zu nutzen. Nur zwei der Teilnehmer gaben an, öffentliche WLAN Hotspots täglich zu nutzen. Und nur einer der Teilnehmer gab eine Nutzung mehrmals pro Monat an. Insgesamt wurden auf allen Geräten der 92 Teilnehmer 239 verschiedene Konfigurationen für unverschlüsselte Netzwerke vorgefunden, was die Größenordnung der Gefahr veranschaulicht.
Smartphone-Nutzer, die regelmäßig die Dienste von Hotspots nutzen, werden in der Regel wissen, dass sie unverschlüsselte Netzwerke auf ihren Geräten konfiguriert haben. Sie sind sich im besten Fall ebenfalls darüber im Klaren, dass ihr Smartphone sich automatisch mit diesen Netzwerken verbindet und

dass dies zu Gefahren im Alltag führen kann. Interessanter – weil deutlich gefährlicher – sind diejenigen Benutzer, die angaben keinerlei Hotspots in ihrem Alltag zu nutzen, trotzdem aber zahlreiche auf ihren Geräten konfiguriert hatten. 39 Teilnehmer gaben an, gar keine Hotspots zu nutzen. Aber nur 37,8% dieser Teilnehmer haben keinerlei unverschlüsselte WLAN-Netzwerke konfiguriert. Die verbleibenden 62,6% haben wenigstens eins, aber bis zu 20 verschiedene entsprechende Netzwerkkonfigurationen auf ihren Geräten. Im Schnitt hat diese Personengruppe 3.65 konfigurierte öffentliche Netzwerke. Das ist im Rahmen dieser Studie sogar mehr als die durchschnittliche Anzahl an konfigurierten unverschlüsselten Netzwerken bei Teilnehmern, die angegeben haben, Hotspots täglich oder mehrmals im Monat zu nutzen. Die durchschnittliche Anzahl dieser Gruppe beträgt lediglich 3,0. In der Gruppe der Teilnehmer die angab, Hotspot-Dienste nur selten zu nutzen hatten lediglich 5 Teilnehmer keinerlei öffentliche Netzwerke gespeichert. Im Schnitt hatten diese Teilnehmer 4.89 konfigurierte öffentliche Funknetzwerke.
Diese Zahlen zeigen deutlich, dass Benutzer über die Gefahren von Evil Twin-Access Points nicht ausreichend aufgeklärt sind. Ebenso veranschaulichen sie, wie verbreitet sich diese Gefahrenlage darstellt.

Positionsdaten

Abgesehen von der Betrachtung potentieller Gefährdungen durch konfigurierte Netzwerke lag ein anderes Augenmerk der Studie auf der Sammlung verschiedenster Kontextdaten, die zukünftig bei der Erkennung von Evil Twin Access Points hilfreich sein können.
Ein Parameter, der hierfür intuitiv hilfreich erschien ist die Position des Smartphones. Während der einzelnen Verbindungen innerhalb der Studienlaufzeit wurden insgesamt 163.043 Position aufgezeichnet. Das entspricht ca. 72,8% aller Verbindungen, die im Rahmen der Studie aufgezeichnet wurden. Die Teilnehmer der Studie hatten jederzeit die Wahl, die eigene Position zu entweder mit zu übermitteln oder diese aus den zu übermittelnden Daten auszuschließen.
Um die aktuelle Position des Smartphones während des Verbindungsaufbaus mit einem WLAN zu ermitteln, wurde in erster Linie die *Location API* der Google Play Services verwendet. In Fällen, in denen die Google Play Services

nicht verfügbar waren, wurde als Ausweichlösung auf die native Android Location API zurückgegriffen. Eine detaillierte Erklärung zur Funktionsweise der Google Play Services findet sich in Abschnitt 8.1.1. Obwohl die Genauigkeit der Position bei einer Bestimmung mit Hilfe von GPS höher ist, als bei der Bestimmung mit Hilfe des Network Providers, benötigt sie auch deutlich mehr Zeit und Energie. Mit den Google Play Services wurde ein gut nutzbarer Mittelweg sowohl für die Genauigkeit, als auch für die Ermittlungszeit und die hierbei verwendete Energie gefunden.
Auf den Geräten der großen Mehrheit der Teilnehmer waren die Google Play Services verfügbar. Entsprechend konnten 99,4% aller ermittelten Positionen mit Hilfe dieser API bestimmt werden. Im Schnitt wurde hierbei eine Genauigkeit von 115,06m erreicht. Ob diese Genauigkeit des Kontextparameters ausreichend ist, um ihn als Stütze für eine Entscheidung im Rahmen einer Angriffserkennung zu nutzen wird in Abschnitt 7.4 beschrieben und erörtert.

Netzwerkumgebung

Zur Entscheidung darüber, ob es sich bei einem zu verbindenden Hotspot um einen legitimen, originalen handelt oder um einen gefälschten und somit bösartigen, müssen neben den bereits genannten weitere Kontextparameter zum Einsatz kommen. Ein weiterer möglicher Parameter hierfür ist die zum Verbindungszeitpunkt vorhandene Netzwerkumgebung des Geräts. Im Speziellen sind an dieser Stelle umgebende Funknetzwerke gemeint. Diese wurden ebenfalls mit Hilfe der Studien-App während jedes Verbindungsaufbaus ermittelt. Betrachtet man alle während der Studie aufgezeichneten Verbindungen, so konnten im Schnitt 10.26 WLAN-Netzwerke pro Verbindung in der Umgebung ermittelt werden. Dieser Kontextparameter erscheint daher als vielversprechend für eine spätere Verwendung im Entscheidungsprozess.
In Abbildung 7.10 sind zwei Werte eines jeden Benutzers dargestellt. Zur rechten Ordinate gehörig und somit in rot dargestellt ist der Mittelwert der Anzahl aller Funknetzwerke in der Umgebung pro Benutzer. Zur linken Ordinate gehörig und somit als schwarze Balken dargestellt ist die Anzahl unterschiedlicher Access Points, mit denen sich ein Benutzer während der Studie verbunden hat. Hierbei werden alle Access Points betrachtet, insbesondere werden also auch die vielen Access Points großer Unternehmens- und Uni-Netzwerke unterschie-

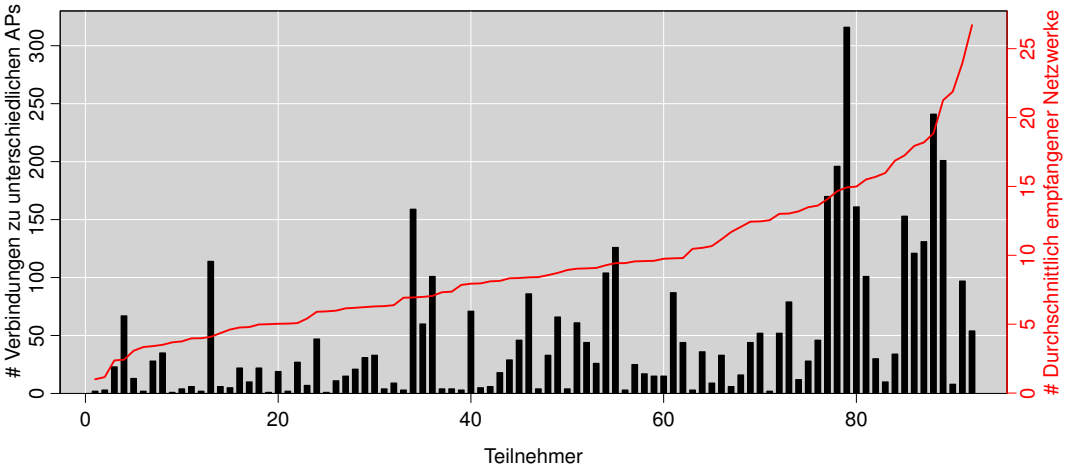


Abbildung 7.10: Anzahl der Verbindungen zu unterschiedlichen Access Points pro Benutzer zusammen mit der durchschnittlichen Anzahl an gefundenen Funknetzwerken in der Umgebung beim Verbindungsaufbau

den. Aus diesem Grund kommt es zu den zunächst sehr hoch erscheinenden Zahlen an dieser Stelle.

Betrachtet man die beiden gegeneinander aufgetragenen Werte für jeden einzelnen Benutzer, so ist ein leichter Trend dahingehend erkennbar, dass Benutzer, die sich mit vielen verschiedenen Access Points verbinden, durchschnittlich ebenfalls mehr Access Points in ihrer Umgebung ermitteln konnten. Diese Tatsache ist ebenfalls darauf zurückzuführen, dass viele dieser Verbindungen in Unternehmensnetzwerken aufgebaut wurden, in denen durch die Betreiber eine möglichst hohe WLAN-Abdeckung angestrebt wird. Die hierfür hohe Zahl eingesetzter Access Points wirkt sich entsprechend an dieser Stelle auf die Zahlen aus.

Eine andere interessante Tatsache ist es, dass durchschnittlich nur bei rund 19% aller aufgezeichneten Verbindungen weniger als fünf umgebende Netzwerke vorzufinden waren, die zur Bestimmung des Kontextes herangezogen werden können. Angesichts der Tatsache, dass auch weniger als fünf umgebende Netzwerke bei der Charakterisierung eines Kontextes hilfreich sein können, unterstreichen die Zahlen abermals, dass dieser Parameter für den Entscheidungsprozess von großer Relevanz ist.

Gefahrenquelle „Automatische Wiederverbindung“

Um die Gefahr der automatischen Wiederverbindung zu bekannten Netzwerken im Hinblick auf Evil Twin Access Points zu untersuchen wurde als weiterer Parameter der Status des Smartphone-Displays (an/aus) und der Verriegelungszustand (engl. lock state) des Gerätes (locked/unlocked) mit aufgenommen. Mit Hilfe dieser Parameter kann bestimmt werden, ob der Benutzer das Smartphone zum Zeitpunkt der Verbindung aktiv genutzt hat oder ob es sich um eine Verbindung ohne jegliche Nutzerinteraktion gehandelt hat.
Die Gefahr soll an einem Beispiel für einen Hotspot-Anbieter dargestellt werden. Hierfür wurde der größte Hotspot-Anbieter in Deutschland T-Mobile ausgewählt. Acht verschiedene Teilnehmer haben während der Studienlaufzeit insgesamt 476 Verbindungen zu Access Points dieses Anbieters aufgebaut. Alarmierend ist die Tatsache, dass 374 dieser Verbindungen mit ausgeschaltetem Display und dementsprechend ohne jegliche Benutzerinteraktion aufgebaut wurden. Es wurden also 78% all dieser Verbindungen durch das Smartphone selbst initiiert. Der jeweilige Benutzer wird in vielen dieser Fälle nicht einmal bemerkt haben, dass sich sein Smartphone mit den entsprechenden Access Points verbunden hat.

7.2.5 Zusammenfassung Studie

In diesem Abschnitt wurde die Studie vorgestellt, mit Hilfe welcher zum einen Daten gesammelt werden konnten, die das Gefahrenpotenzial von Evil Twin Angriffen aufzeigen. Zum anderen wurde mit Hilfe einer Android-App eine Vielzahl von Kontextparametern gesammelt, um sie auf ihre Eignung zur Angriffserkennung hin zu untersuchen. Zunächst wurde die Architektur und die Funktionsweise der hierfür entwickelten Android-App und des zentralen Webservice zur Sammlung aller Informationen vorgestellt. Alle im Rahmen dieser Studie erhobenen Kontextdaten wurden erläutert und die Ergebnisse eines in die App integrierten Fragebogens wurden diskutiert. Ebenso wurden Vergleiche zwischen den Selbsteinschätzungen der Teilnehmer und den gesammelten Informationen zum Nutzungsverhalten der Geräte durchgeführt und beschrieben. Im Rahmen einer Analyse der gesammelten Kontextdaten wurden für ein Erkennungssystem relevante Parameter identifiziert und ihre Eig-

nung diskutiert. Abschließend wurde die in Abschnitt 4.3 aufgezeigte Gefahr der automatischen Wiederverbindung mit bekannten Netzwerken mit Zahlen belegt. Die hierfür während der Studie gesammelten Daten zeigen klar, dass sich Smartphones mit konfigurierten öffentlichen Hotspots sehr oft völlig automatisiert und ohne jegliche Nutzerinteraktion mit diesen Hotspots verbinden. Es konnte gezeigt werden, dass sich Smartphones der Teilnehmer zu einem im deutschen Raum gängigen Hotspot-Netzwerk in mehr als 78% der Fälle automatisiert verbunden haben. Dies bestätigt die eingangs beschriebene Gefahr durch Man-in-the-middle Angriffe, die somit durch Angreifer an beliebigen Orten durchgeführt werden können. Ebenso unterstützen die Zahlen das Vorhaben, im Rahmen dieser Arbeit ein Erkennungssystem für Evil Twin Netzwerke zu erforschen, welches ohne weitere zusätzliche Infrastruktur und Dienste lokal auf dem Gerät des Benutzers betrieben werden kann. Zunächst wird noch kurz auf den generellen Entwicklungsprozess von Systemen eingegangen, bevor in den folgenden Abschnitten erläutert wird, wie ein Angreifer vorgehen kann, um Schutzfunktionen eines Erkennungssystems zu umgehen. Im Anschluss daran wird die Entwicklung und der Aufbau des im Rahmen dieser Dissertation entstandenen Sicherheitssystems beschrieben.

7.3 Entwicklungsprozess

Bei der Entwicklung neuartiger Systeme, wie das im Rahmen dieser Arbeit entstehende Sicherheitssystem sollen verschiedene Aspekte und Ausgangssituationen betrachtet werden. Um dies zu erreichen sind im Rahmen der Entwicklung vielfache, wiederholte Durchführungen und Verbesserungen erforderlich. Dies betrifft bei dieser Art der Entwicklung sowohl die Durchführung von Studien, als auch Simulationen von Teilsystemen.

Im ersten Schritt wird mit der Planung des Systems begonnen. An dieser Stelle werden für die Beobachtung und Messung relevante Parameter festgelegt und es wird bestimmt, wie diese durch das System genutzt werden sollen. Ergebnis dieses Arbeitsschritts sind im konkreten Fall Maßnahmen zur Erkennung von Evil Twin Access Points. Im nächsten Schritt wird das geplante System implementiert. In einer frühen Entwicklungsphase fließen die geplanten Aspekte des Systems in einen Prototyp ein. Später werden neu gefundene

und geplante Maßnahmen in Form von Updates für das bereits bestehende System bereitgestellt. Ergebnis dieses Schritts ist ein funktionsfähiges und stabiles System. Mit Hilfe des Prototyps wird das entwickelte System getestet. In diesem Schritt wird nicht ausschließlich die allgemeine Funktion des Systems und seine Stabilität getestet und beurteilt. Vielmehr wird bei der Entwicklung des Erkennungssystems zusätzlich die Erkennungsleistung des Systems beurteilt. Hierfür werden alle für die Erkennung relevanten Daten aufgezeichnet und analysiert. Sollten bei der Prüfung Probleme hinsichtlich der generellen Funktion oder der Stabilität auftreten, so gilt für diesen, wie auch für alle anderen Schritte, dass ein direkter Wiedereintritt in die Planungsphase möglich ist, um Verbesserungen für eine neue Version des Systems planen zu können. Auf Basis der gesammelten Daten werden im vierten und letzten Schritt des Kreislaufs Anpassungen am Gesamtsystem durchdacht. Dieser Schritt verschmilzt nach mehreren Durchläufen des Zyklus immer mehr mit dem nun wieder beginnenden Planungsprozess, in dem die Implementierung der neuen Konzepte vorbereitet wird.

Im Folgenden werden grundlegende Maßnahmen und Mechanismen zur Erkennung von Evil Twins dargestellt und die Entwicklung und Anpassung des Erkennungssystems beschrieben. Der o.g. Kreislauf wurde hierbei mehrfach durchlaufen, indem mit Hilfe eines Simulators die Erkennungsleistung des Systems getestet und anschließend verbessert wurde. Der beschriebene Kreislauf stellt entsprechend an vielen Stellen die Basis der nun folgenden Beschreibung dar.

7.4 Angriffstypen und Gegenmaßnahmen

Eines der Ziele dieser Arbeit ist die Entwicklung eines unabhängigen Systems, das den Benutzer vor der Verbindung mit Evil Twin Access Points schützen soll. Zunächst muss hierfür betrachtet werden, welche Möglichkeiten ein Angreifer hat, um einen derartigen Angriff durchzuführen. Es werden in den folgenden Abschnitten verschiedene Arten von Angriffen dargestellt. Beginnend mit einem sehr simplen Aufbau steigt die Komplexität der Angriffe stetig bis hin zu einer Komplexität, wie sie für die meisten Angreifer nicht zu realisieren ist.

Es wird beschrieben, welche technischen Fähigkeit und welches Equipment für die einzelnen Angriffsarten benötigt wird. Neben der Beschreibung der verschiedenen Angriffe sollen ebenfalls Möglichkeiten aufgezeigt werden, wie diese Angriffe durch einen Smartphone-Benutzer mit Hilfe des METDS erkannt werden können.

In den einzelnen Szenarien ist es das Ziel des Angreifers, einen echten Access Point zu fälschen. Hierbei wird davon ausgegangen, dass der Angreifer dies auf möglichst unauffällige Art und Weise durchführt, um nicht als Angreifer erkannt oder enttarnt zu werden. Einem Benutzer ist es alleinig anhand seines Smartphones in den folgenden Szenarien nicht möglich, zu entscheiden, ob es sich bei einem angezeigten Access Point um einen realen oder um eine Fälschung handelt.

7.4.1 Typ A: Fälschen einer SSID

Im ersten Szenario wird vom Angreifer nur ein sehr kleines Basis-Setup benötigt. Er eröffnet einen Access Point mit einer bekannten und verbreiteten SSID (wie beispielsweise *tmobile* oder *BTOpenzone*). Der Angreifer wartet nun darauf, dass sich Benutzer, die bereits einmal mit den o.g. Netzwerken verbunden waren, mit seinem Access Point verbinden. Wie in Abschnitt 5.1 dargestellt kann dieser Prozess dahingehend optimiert werden, dass speziell die SSIDs durch den Angreifer simuliert werden, die von mobilen Geräten in seiner Umgebung aktiv gesucht werden.

Da mobile Betriebssysteme nur die SSID für den Vergleich mit bekannten Netzwerken heranziehen, kann dieser Angriff an jedem beliebigen Ort durchgeführt werden. Es spielt also keine Rolle, an welchem Ort sich der originale AP befindet oder wo der Angriff selbst durchgeführt wird.

Obwohl diese Variante des Evil Twin Angriffs zur Zeit die trivialste und somit am häufigsten vorkommende ist, kann sie mit einfachen Mitteln unterbunden werden. METDS speichert neben der SSID auch die BSSID des verbundenen Access Points. Bei einer erneuten Verbindung mit einem Netzwerk wird zusätzlich zur SSID ebenfalls die BSSID verglichen. Bei einem wie oben beschriebenen einfachen Aufbau des Angriffs kann diese Art des Angriffs somit erkannt und dem Benutzer eine Warnung angezeigt werden. Der Verbindungs-

aufbau wird gestoppt und die Gefahr durch Man-in-the-middle Angriffe kann minimiert werden.

7.4.2 Typ B: Fälschen der BSSID eines Access Points

Hat ein Angreifer Wissen darüber erlangt, dass die BSSID Bestandteil der Prüfung eines entsprechenden Sicherheitssystems ist, so kann er seinen Angriff leicht erweitern. Ebenso wie die SSID, kann auch die BSSID auf einfache Art und Weise durch einen Angreifer gefälscht werden. Hierfür sind online freie Werkzeuge verfügbar. Für diese Art des Angriff sind entsprechend keine weiteren technischen Fähigkeiten erforderlich.

Dieser Angriff ist bereits deutlich eingeschränkter als der vorherige. Geht man von zwei Benutzern großer Hotspot-Netzwerke (wie beispielsweise *BTOpenzone*) aus, die mit verschiedenen Hotspots des Anbieters verbunden waren, so können diese nicht mehr mit ein und demselben Angriff attackiert werden. Beide Benutzer haben nach der ersten Nutzung verschiedene BSSIDs auf ihren Geräten gespeichert, die bei einem Angriff mindestens bei einem der beiden eine Warnung auslösen würde. Trotzdem sind natürlich weiterhin gezielte Angriffe auf die Benutzer dieser Hotspots möglich.

Ein weiterer Schritt in Richtung einer zuverlässigen Erkennung von Evil Twin Access Points ist die Einbeziehung weiterer Umgebungsparameter in den Entscheidungsprozess. Hierfür kann die durch METDS ermittelte Umgebung aller WLAN-Netzwerke herangezogen werden. Es können alle SSIDs und BSSIDs zusammen mit den von den dazugehörigen Access Points unterstützten Verschlüsselungsmethoden gespeichert werden. Auf diese Weise kann der Kontext eines Access Points je nach Lage deutlich detaillierter erfasst und beschrieben werden. Zu Bedenken ist hierbei hingegen, dass Schwankungen beim Empfang und reale Veränderungen der WLAN-Umgebung berücksichtigt werden müssen, um fälschlich erzeugte Warnmeldungen des Systems zu minimieren.

Ein Angreifer müsste sich für einen erfolgreichen Angriff auf ein mit diesem Sicherheitssystem ausgestatteten Smartphone entweder sehr nah am originalen Access Point befinden oder einen Großteil der WLAN-Umgebung am Ort seiner Wahl simulieren.

7.4.3 Typ C: Fälschen einer Netzwerkumgebung

Selbst komplette Netzwerkumgebungen können mit geeigneter Hardware simuliert werden. Hierfür sind nicht nur erweiterte technische Fähigkeiten, sondern auch zusätzliche Hardware und Vorbereitung notwendig. Zur Simulation dieser WLAN-Umgebungen wird sowohl spezielle Hard- als auch Software benötigt, die es erlaubt, auf einem physikalischen Gerät mehrere virtuelle Schnittstellen bereitzustellen. Diesen virtuellen Schnittstellen können dann unterschiedliche SSIDs und BSSIDs zugewiesen werden. Eine mögliche Umsetzung dieses Konzepts ist die Nutzung der freien und quelloffenen Router Software DD-WRT [11]. Bei Verwendung kompatibler Hardware und durch eine entsprechende Konfiguration der virtuellen Schnittstellen lassen sich mit einem WLAN-Router zwei unabhängige WLANs mit unterschiedlichen SSIDs und BSSIDs erzeugen. Nachgestellt und getestet wurde dieser Aufbau mit dem Router Linksys WRT-54GL und der DD-WRT Software in der Version 24.
Um diese Art des Angriffs dennoch bestmöglich erkennen zu können, kann die Position des mobilen Geräts zum Zeitpunkt der Verbindung mit einem Access Point als weiterer Parameter hinzugefügt werden. Im Falle einer Abweichung zur zuvor gespeicherten Position, die einen Schwellwert überschreitet, soll eine Warnung ausgegeben werden. So kann dem Benutzer mitgeteilt werden, dass ein bereits bekannter Access Point an einem Ort gefunden wurde, der nicht dem Ursprungsort entspricht. Nun ist es am Benutzer zu entscheiden, ob dies ein gewünschtes Verhalten ist oder ob diese Situation nicht der Erwartung entspricht.
Bei der Positionsbestimmung gilt es, einige Besonderheiten zu beachten: Die Bestimmung der Position mit Hilfe eines in das Smartphone integrierten GPS-Moduls ist in vielen Fällen nicht möglich. Sobald sich der Benutzer in einem Gebäude aufhält oder aus anderen Gründen keine direkte Sichtverbindung zu ausreichend vielen GPS-Satelliten aufgebaut werden kann, ist eine Bestimmung dieser Art unmöglich. In diesen Fällen muss auf andere Verfahren zurückgegriffen werden. In solchen Fällen und zur schnelleren Positionsbestimmung wird durch verschiedene Frameworks ebenfalls immer wieder auf die umgebenden WLAN-Netzwerke zurückgegriffen. Diese werden mit einer zuvor aufgebauten Datenbank abgeglichen, um auf diese Weise den aktuellen Aufenthaltsort bestimmen zu können. Diese Variante hat den oben genannten Nachteil, dass es

sich nicht um eine hinreichend fälschungssichere Methode handelt. Eine dritte Variante zur Bestimmung des Aufenthaltsortes ist die Nutzung des Mobilfunknetzes. Hierbei wird zwar eine deutlich geringere Genauigkeit erreicht, die Fälschungssicherheit ist der vorangegangenen Methode aber deutlich überlegen. Die Simulation von Mobilfunknetzen stellt den Angreifer vor große Herausforderungen – sowohl technischer als auch finanzieller Art.

7.4.4 Typ D: Fälschen der gesamten Umgebung

Die letzte im Rahmen dieser Arbeit betrachtete Art des Angriffs umfasst die Fälschung oder Simulation der gesamten Umgebung. Dies schließt alle der oben beschriebenen Parameter mit ein. Dieser Angriff ist je nach Ausführungsort mit einem beträchtlichen Aufwand und erheblichen Kosten verbunden. Nur gezielte Angriffe auf Einzelpersonen in Kombination mit einer großen finanziellen Ausstattung sind in diesem Szenario denkbar.

Eine weitere Möglichkeit ist es, den Angriff am originalen Ort mit der eigenen Hardware durchzuführen. Dies bringt allerdings Probleme auf Netzwerkebene mit sich, die auftreten, wenn zwei Funknetzwerke gleichen Namens in einer Umgebung betrieben werden. Nimmt der Angreifer diese Probleme in Kauf, so muss er trotzdem stets dafür sorgen, dass der von ihm eingerichtete Access Point eine größere Signalstärke aufweist, als der originale. Ansonsten würden sich die mobilen Geräte der potentiellen Opfer mit dem originalen Access Point verbinden und ein Angriff wäre nicht möglich.

Wird der bösartige Access Point tatsächlich in direkter Nähe des originalen betrieben, so kann der Angriff durch den Betreiber des originalen Hotspots erkannt werden. Hierfür bestehen Sicherheitssysteme wie WLAN Intrusion Detection und Prevention Systeme vieler Hersteller wie AirTight[1] und Cisco[2] , die kontinuierlich die Umgebung scannen und so weitere Access Points in der Umgebung finden. Der Betreiber könnte in einem solchen Fall Maßnahmen ergreifen, die ein Angreifer in jedem Fall umgehen möchte.

[1] `http://www.airtightnetworks.com/home/solutions/wireless-intrusion-prevention`

[2] `http://www.cisco.com/c/en/us/products/collateral/wireless/adaptive-wireless-ips-software/data_sheet_c78-501388.html`

Angriffe dieses Typs können auch durch METDS nicht verhindert werden. In diesem Szenario sind hingegen nur sehr gezielte und risikoreiche Angriffe durchführbar.

Kapitel 8

Entwicklung des kontextbasierten Sicherheitssystems

Die Entwicklung des Sicherheitssystems ist Bestandteil dieses Kapitels. Zunächst werden die verschiedenen Zustände erläutert, die das System erkennen soll. Im Anschluss an die Beschreibung der Architektur folgt die Beschreibung des zugrunde liegenden Algorithmus. Im zweiten Teil des Kapitels wird die Portierung des Erkennungssystems auf die Android-Plattform beschrieben. Es werden verschiedene Möglichkeiten der Implementierung diskutiert und eine Einbettung in die grafische Oberfläche vorgestellt. Teile dieses Kapitels wurden im Rahmen der wissenschaftlichen Konferenz *Financial Cryptography and Data Security 2015* veröffentlicht [65].

Die Schwierigkeit bei der Erkennung von Angriffen und Bedrohungen auf Smartphones entsteht unter anderem aus einem Mangel an Informationen, die zum Zeitpunkt des Verbindungsaufbaus zur Verfügung stehen. Im konkreten Fall sind nur begrenzte Informationen über die Umgebung verfügbar, die mit Hilfe verschiedener Schnittstellen eines Smartphones ermittelt werden können. Dem im Rahmen dieser Arbeit entstandenen System zur Erkennung von Evil Twin Access Points sollen für seine Entscheidung möglichst viele dieser Umgebungsdaten zur Verfügung gestellt werden. Des weiteren soll das System unabhängig agieren können. Es sollen entsprechend keine weiteren Systeme und keinerlei weitere Infrastruktur nötig sein, um das System zu betreiben. Im Folgenden werden die Parameter beschrieben, die zur Erkennung von Evil Twin Access Points genutzt werden und es wird dargestellt wie sie im Rahmen des Erkennungssystems zum Einsatz kommen.
Die folgenden Betrachtungen sind im Gegensatz zu den vorausgegangen aus Sicht des Benutzers zu sehen. Zunächst wird von einem Benutzer ausgegangen, der sich mit seinem Smartphone in Kommunikationsreichweite zu einem Access Point befindet. Das Smartphone benötigt zu diesem Zeitpunkt Zugriff auf das Internet und versucht entsprechend, sich mit einem bekannten Access Point in der Umgebung zu verbinden. An dieser Stelle muss der dem METDS zugrunde liegende Algorithmus zwei Situationen unterscheiden

1. Soll sich das Smartphone zum ersten Mal mit einem AP verbinden, so liegen keinerlei Daten vor, die die Erkennung eines bösartigen Access Points ermöglichen können. Das System kann den Benutzer in dieser Situation nicht unterstützen. Optional können an dieser Stelle zwar Warnungen eingeblendet werden, die dem Benutzer mögliche Gefahren vor Augen führen. Der Erkennung und einer entsprechenden Verhinderung eines Angriffs dient dies zu diesem Zeitpunkt hingegen nicht. Das verwendete Prinzip des *Trust On First Use* (TOFU) ist dasselbe, welches ein Benutzer auch ohne das Erkennungssystem verfolgen müsste. Vor der ersten Verbindung mit einem neuen Netzwerk muss diesem explizit vertraut werden.

2. Soll sich das Smartphone hingegen mit einem AP verbinden, mit dem es in der Vergangenheit bereits mindestens einmal verbunden gewesen ist, so

können hierfür gesammelte und aggregierte Daten zur Verfügung stehen. Diese können für den Entscheidungsprozess der aktuellen Verbindung herangezogen werden. Hierfür werden die zur Verfügung stehenden Daten über den Access Point mit der aktuellen Umgebung des Smartphones verglichen.

Je nach vorliegender Situation muss der Algorithmus entsprechende Maßnahmen ergreifen um dem Benutzer auf diese Weise den bestmöglichen Schutz zu bieten.

8.1 Das Erkennungssystem

Für die Erkennung von Evil Twin Access Points verwendet der heuristische Ansatz des Algorithmus die folgenden Parameter:

- **SSID:** In erster Linie wird – wie in jedem mobilen Betriebssystem – die SSID zur Identifizierung eines Netzwerks genutzt.
- **BSSID:** Darüber hinaus vergleicht der Algorithmus die BSSID des aktuellen AP mit der für dieses Netzwerk und diese Umgebung gespeicherten.
- **Netzwerkumgebung:** Da BSSIDs ebenso leicht gefälscht werden können wie SSIDs (siehe Abschnitt 7.4.3) nutzt der Algorithmus zusätzlich WLANs aus der Umgebung. Hierfür wird während des Verbindungsaufbaus ein Scan nach verfügbaren Netzwerken über die WLAN-Schnittstelle des Smartphones gestartet. Die Ergebnisse dieses Scans beinhalten nicht nur die SSIDs der umliegenden Funknetzwerke, sondern auch die BSSIDs der Access Points und deren unterstützte Authentifizierungs- und Verschlüsselungsmethoden. Durch die Speicherung all dieser Informationen, lassen sich Netzwerkumgebungen deutlich detaillierter beschreiben und eine höhere Genauigkeit beim Vergleich zukünftiger Verbindungen erzielen.
- **Mobilfunkinformationen:** Da Smartphones in den meisten Fällen mit einem Mobilfunknetz verbunden sind, werden auch über diese Schnittstelle Informationen zur aktuellen Umgebung gesammelt. Der Algorithmus nutzt hierfür die Kennung der Funkzelle (Cell-ID), in welcher das

Smartphone zum Zeitpunkt des Verbindungsaufbaus eingebucht ist. Zusammen mit der ebenfalls über diese Schnittstelle verfügbaren Aufenthaltsbereichskennzahl (Location Area Code; LAC) kann hiermit ein eindeutiger, räumlicher Sektor innerhalb eines Aufenthaltsbereiches beschrieben werden. Diese Informationen werden durch den Algorithmus dazu genutzt, die Position des Geräts zu verifizieren.

- **Position:** Um die Ortungsfunktion von heutigen Smartphones auszunutzen wird bei Bedarf zusätzlich die geografische Position des Smartphones bestimmt. Dies geschieht, wie in Abschnitt 7.2.2 beschrieben, über die Google Play Services oder über die native *Location API*.

Während der Entwicklung des Erkennungsalgorithmus wurden weitere Parameter auf ihre Eignung zur Verbesserung der Erkennungsleistung hin untersucht. Zwei zunächst als hilfreich erscheinende Parameter waren die Signalstärken der umgebenden Funknetzwerke und die Frequenz auf der die entsprechenden Access Points arbeiten. Die Signalstärke erwies sich hierbei allerdings als wenig hilfreich. Sie hängt stark von der Ausrichtung und Lage des Gerätes selbst im Raum ab. Ebenso wirken sich schon kleinste Änderungen der Position stark auf die gemessenen Signalstärken aus. Aus diesen Gründen ist die Signalstärke nicht als relevanter Parameter nutzbar, um die Zuverlässigkeit des Gesamtsystems zu erhöhen. Das gleiche Bild hat sich durch die Berücksichtigung der Betriebsfrequenz ergeben. Auch mit Hilfe der Frequenz konnte die Zuverlässigkeit nicht gesteigert werden. Der Grund hierfür ist, dass moderne APs und WLAN-Router die Frequenz bei Bedarf ändern. Sie scannen hierfür selbstständig die in der Umgebung befindlichen Funknetzwerke und wählen selbständig einen wenig genutzten Funkkanal aus, um weitestgehend störungsfrei ihre Dienste anbieten zu können. Aus diesem Grund ist der Funkkanal zu variabel und somit für das System nicht von Nutzen. Beide Parameter haben bei Tests zu höheren Falsch-positiv-Raten geführt, was sowohl die Sicherheit, als auch die Benutzbarkeit des Systems negativ beeinflusst.
Nach Sammlung und Auswertung all dieser Kontextparameter erreicht das Erkennungssystem verschiedene Zustände. Je nach erreichtem Zustand muss das System angepasste Maßnahmen ergreifen, um den Benutzer zu schützen. Bei allen Ergebniszuständen des Systems muss beachtet werden, dass die Falsch-

Negativ-Rate möglichst niedrig gehalten werden muss, um die Akzeptanz und Benutzbarkeit des Gesamtsystems nicht negativ zu beeinflussen.

8.1.1 Zustände

Im Folgenden werden die verschiedenen Zustände beschrieben, die nach der Bewertung durch das Erkennungssystem erreicht werden können. Ebenso wird dargestellt, wie das System dem Benutzer in den verschiedenen Situationen die aktuelle Gefahrenlage darstellt und welche Optionen sich hieraus für den Benutzer ergeben.

Zustand „SSID unbekannt“

Im Falle einer Verbindung zu einem gänzlich unbekannten Netzwerk, wird als Ergebnis dieser Zustand erreicht. Es gibt in diesem Fall keinerlei Daten, die für die Erkennung eines potentiellen Angreifers hilfreich wären. Aus diesen Gründen ist an dieser Stelle lediglich eine optionale Warnung möglich, die den Benutzer über die Risiken einer solchen neuen Verbindung aufklärt. Im derzeitigen Stand des entwickelten Systems wird auf diese Art der Warnung verzichtet.
Bei erfolgreicher Verbindung werden alle oben beschriebenen Daten der Umgebung erfasst, um bei einer erneuten Verbindung mit diesem Access Point als Entscheidungsgrundlage herangezogen werden zu können. Nur so kann bei einer zukünftigen Verbindung der Kontext des Access Points durch das Erkennungssystem zuverlässig wiedererkannt und somit auf eine Warnung verzichtet werden.

Zustand „Unbekannte BSSID“

Wurde die SSID als bekannt markiert und ein entsprechender Datensatz in der Erkennungsdatenbank gefunden, so wird geprüft, ob die BSSID in Verbindung mit dieser SSID ebenfalls wiedererkannt werden kann. Befindet sich die BSSID nicht unter den bekannten Access Points des Funknetzwerks, so wird dem Benutzer eine Warnung angezeigt. Diese klärt ihn über die potentiell gefährliche Situation auf und gibt ihm zwei Entscheidungsmöglichkeiten:

- Entweder, der Benutzer weiß, dass es sich bei dem zu verbindenden Access Point um einen legitimen handelt. In diesem Fall wird die Verbindung hergestellt und sämtliche Kontextdaten werden durch das System gelernt, um diese bei einer erneuten Verbindung berücksichtigen zu können.

- Oder, der Benutzer gibt keine Einwilligung in die Verbindung mit diesem Access Point. In diesem Fall wird der Verbindungsaufbau unterbrochen und es findet kein Lernprozess der aktuellen Umgebung statt.

Dieses Vorgehen erweitert das *Trust On First Use*-Prinzip um Access Points. Hierdurch wird die Benutzbarkeit des Smartphones beeinträchtigt, wie das folgende Beispiel zeigt.

Beispiel: Ein Benutzer nutzt häufig sein Smartphone der Starbucks-Filiale seiner Heimatstadt. Das System hat sowohl die SSID des Starbucks-Netzwerks, als auch die BSSID des Access Points der Heimatfiliale gespeichert und kann anhand dieser und weiterer Kontextparameter das Netzwerk und seine Umgebung verifizieren. Befindet sich der Benutzer in Zukunft in einer anderen Filiale des Unternehmens, so wird sich das Smartphone nicht automatisch mit dem Netzwerk verbinden, da zwar die bekannte SSID vorgefunden wird, nicht aber der zu dieser SSID bekannte Access Point der Heimatfiliale. Es wird dem Benutzer eine Warnung angezeigt, dass es sich zwar um ein bekanntes Netzwerk handelt, der vorliegende Access Point hingegen unbekannt ist.

Mit Hilfe der in der Feldstudie gesammelten Daten über WLAN-Verbindungen konnte ermittelt werden, dass bei den Teilnehmern der Studie nur durchschnittlich 8,14 neue BSSIDs angelernt werden mussten. Da diese Situation bei Abwesenheit eines Evil Twins nur dann auftritt, wenn neue Orte besucht werden, erscheint die Einschränkung hinsichtlich der Benutzbarkeit als hinnehmbar. Der Zugewinn an Sicherheit durch die Reduzierung der Angriffsmöglichkeiten durch Evil Twins überwiegt an dieser Stelle.

Zustand „Unbekannte Umgebung“

Bei einer Verbindung zu einem Access Point, dessen Tupel – bestehend aus einer SSID und einer BSSID – dem Erkennungssystem bekannt ist, wird ein weiterer Test zur Verifikation der Umgebung durchgeführt. In diesem Schritt wird ein Scan der Funknetzwerkumgebung im Hintergrund durchgeführt. Das Ergebnis dieses Scans wird mit gespeicherten Access Point Profilen des entsprechenden Access Points verglichen. Ein Access Point-Profil hat hierbei die in Abbildung 8.1 dargestellte Gestalt.

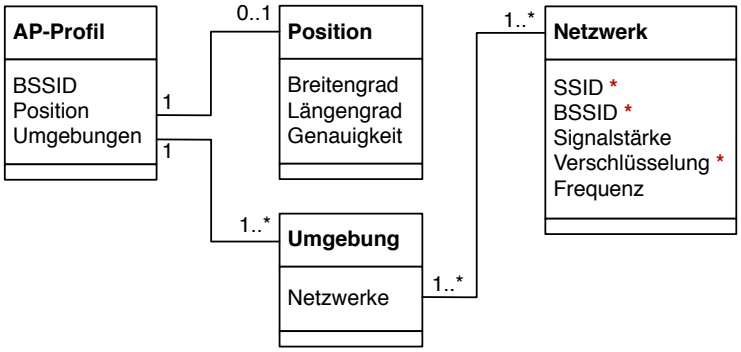


Abbildung 8.1: Klassendiagramm zur Beschreibung der Access Point-Profile innerhalb des Erkennungssystems. Die Attribute, die zur Generierung des Hashwerts herangezogen werden, sind mit einem Stern (*) markiert.

Neben der BSSID und einer optional zu speichernden Position des Access Points ist im Access Point-Profil eine Liste aller für diesen Access Point aufgezeichneten WLAN-Umgebungen enthalten. Eine hier gespeicherte WLAN-Umgebung besteht im Wesentlichen aus den in Abschnitt 7.2.2 beschriebenen und über die API verfügbaren Parameter.
Die Verwendung von Access Point-Profilen lässt sich damit begründen, dass die Ausbreitung eines WLANs in vielen Fällen so groß ist, dass sich für die verschiedenen Orte an denen es empfangbar ist, sehr unterschiedliche Funknetzwerkumgebungen ergeben. So ist bereits ein heimisches WLAN in vielen Räumen einer Wohnung oder eines Hauses empfangbar. In den unterschiedlichen Räumen ergeben sich zum Teil vollständig unterschiedliche Umgebungen. Dies liegt zum einen an eventuell nur schwach empfangbaren WLAN-Signalen aus der näheren Umgebung, wie z.B. Nachbarwohnungen oder anliegenden Häusern. Zum anderen hängt es aber auch mit stark unterschiedlich abschir-

menden Gebäuden und Wänden zusammen. Aus diesen und weiteren Faktoren ergeben sich für dasselbe WLAN zum Teil stark unterschiedliche Funknetzwerkumgebungen, die es beim Vergleich der Umgebungen zu beachten gilt.
Ein weiterer Vorteil der Access Point-Profile und ihrer Integration besteht in der daraus resultierenden Anpassungsfähigkeit des Erkennungssystems. Das Erkennungssystem kann sich durch das Hinzufügen, Ändern und Verwerfen von Access Point-Profilen sich verändernden Umgebungen anpassen. Wird beispielsweise ein Access Point mehrfach bei der Verbindung eines Netzwerks beobachtet, so kann dieser in Form eines Access Point-Profils dem aktuellen Netzwerk hinzugefügt werden. Ebenso kann ein mehrfach nicht beobachtetes Netzwerk aus den Profilen entfernt werden. Auf diese Weise kann sich das System langsam ändernden Umgebungen, in denen neue Access Points hinzugefügt oder entfernt werden anpassen und so seine Genauigkeit erhöht werden.
Zum Vergleich der WLAN-Umgebungen wird der Jaccard-Koeffizient verwendet. Kumar et al. beschreiben in ihrem Buch *Introduction to Data Mining* [37] den Jaccard-Koeffizient als eine Kennzahl für die Ähnlichkeit von Mengen. Zum Vergleich der WLAN-Umgebungen mit Hilfe des Jaccard-Koeffizienten werden im Vorfeld Hashwerte der gefundenen WLAN-Netzwerke gebildet. Wie bereits erwähnt, kommen beim Vergleich der WLAN-Umgebungen nicht alle verfügbaren Parameter zum Einsatz. Stattdessen beschränkt sich die Bildung des Hashwertes auf die Parameter SSID, BSSID und die verwendeten Verschlüsselungs- und Authentifizierungsmechanismen.
Der Jaccard-Koeffizient für die beiden wird folgendermaßen gebildet:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} \tag{8.1}$$

Er kann Werte zwischen 0 und 1 annehmen, wobei gilt, dass ein höherer Koeffizient eine größere Ähnlichkeit der Mengen darstellt. Das folgende Beispiel soll die Verwendung des Jaccard-Koeffizienten für den Vergleich von WLAN-Umgebungen veranschaulichen:

Beispiel Wir gehen davon aus, dass während eines Verbindungsaufbaus in der aktuellen Umgebung die WLANs mit den Hashwerten $H_{new,1}$, $H_{new,2}$, $H_{new,3}$ und $H_{new,4}$ ermittelt wurden. In der zum METDS gehörenden Datenbank sind für dieses Netzwerk zwei Access Point Profile P_1 und P_2 vorhanden.

Profil P_1 beinhaltet für WLANs die Hashwerte $H_{1,1}$, $H_{1,2}$, $H_{1,3}$ und $H_{1,4}$. Profil P_2 beinhaltet die Hashwerte $H_{2,1}$, $H_{2,2}$ und $H_{2,3}$.
Wir gehen im Folgenden davon aus, dass die beiden Access Point Profile P_1 und P_2 die folgenden Überlappungen haben:

$$H_{1,2} = H_{2,2} \qquad \wedge \qquad H_{1,3} = H_{2,3}$$

Darüber hinaus gibt es ebenfalls Überlappungen zwischen den Access Point Profilen und den neu ermittelten Hashwerten:

$$H_{new,1} = H_{1,1} \qquad \wedge \qquad H_{new,2} = H_{1,2} = H_{2,2} \qquad \wedge \qquad H_{new,3} = H_{1,4}$$

Bei dieser Konstellation ergeben sich die Jaccard-Koeffizienten der neu gesammelten Hashwerte und der bestehenden aus den Access Point Profilen folgendermaßen:

$$\begin{aligned} J(New, P_1) &= \frac{|\{H_{new,1}, H_{new,2}, H_{new,3}, H_{new,4}\} \cap \{H_{1,1}, H_{1,2}, H_{1,3}, H_{1,4}\}|}{|\{H_{new,1}, H_{new,2}, H_{new,3}, H_{new,4}\} \cup \{H_{1,1}, H_{1,2}, H_{1,3}, H_{1,4}\}|} \\ &= \frac{|\{H_{1,1}, H_{1,2}, H_{1,4}, H_{new,4}\} \cap \{H_{1,1}, H_{1,2}, H_{1,3}, H_{1,4}\}|}{|\{H_{1,1}, H_{1,2}, H_{1,4}, H_{new,4}\} \cup \{H_{1,1}, H_{1,2}, H_{1,3}, H_{1,4}\}|} \\ &= \frac{|\{H_{1,1}, H_{1,2}, H_{1,4}\}|}{|\{H_{1,1}, H_{1,2}, H_{1,3}, H_{1,4}, H_{new,4}\}|} \\ &= \frac{3}{5} = 0,6 \end{aligned}$$

Für das Access Point Profil P_2 hingegen ergibt sich folgendes:

$$\begin{aligned} J(New, P_2) &= \frac{|\{H_{new,1}, H_{new,2}, H_{new,3}, H_{new,4}\} \cap \{H_{2,1}, H_{2,2}, H_{2,3}\}|}{|\{H_{new,1}, H_{new,2}, H_{new,3}, H_{new,4}\} \cup \{H_{2,1}, H_{2,2}, H_{2,3}\}|} \\ &= \frac{|\{H_{new,1}, H_{2,2}, H_{new,3}, H_{new,4}\} \cap \{H_{2,1}, H_{2,2}, H_{2,3}\}|}{|\{H_{new,1}, H_{2,2}, H_{new,3}, H_{new,4}\} \cup \{H_{2,1}, H_{2,2}, H_{2,3}\}|} \\ &= \frac{|\{H_{2,2}\}|}{|\{H_{new,1}, H_{new,3}, H_{new,4}, H_{2,1}, H_{2,2}, H_{2,3}\}|} \\ &= \frac{1}{6} \approx 0,1667 \end{aligned}$$

Bestimmung des Grenzwerts Um zu bestimmen, ob eine Funknetzwerkumgebung durch METDS als bekannt angesehen wird, erfolgt die Berechnung der Jaccard-Koeffizienten für alle Access Point-Profile. Ist einer der sich hierbei

ergebenden Koeffizienten höher als ein festzulegender Grenzwert, so wird die Umgebung im Folgenden als bekannt angesehen.
Mit Hilfe zahlreicher Simulationen auf den Echtweltdaten, die im Rahmen der Studie gesammelt worden sind, konnte ein Jaccard-Koeffizient von 0,7 als am besten geeignet für das Erkennungssystem bestimmt werden. Er führt zum besten Verhältnis zwischen Falsch-Positiv- und Falsch-Negativ-Warnungen. Die Parameter des Erkennungssystems sind an dieser Stelle jedoch frei anpassbar. Für weitere Untersuchungen oder Erweiterungen des Systems können diese beliebig an die jeweiligen Bedürfnisse angepasst werden. Ebenso sind an dieser Stelle Optionen im aktiven Betrieb des Erkennungssystems denkbar, die den Benutzer entscheiden lassen, ob eine höhere Sicherheit und dafür mehr Warnmeldungen gewünscht sind oder umgekehrt.

Nutzung von Mobilfunkinformationen Im Falle einer unbekannten Funknetzwerkumgebung oder wenn eine Analyse der Umgebung aufgrund fehlender Informationen nicht durchgeführt werden konnte, so wird eine weitere Überprüfung der Umgebung durchgeführt: die Bestimmung der Position anhand des Mobilfunknetzes. Ist das betreffende Smartphone mit einem Mobilfunknetz verbunden, so wird durch die *Telephony API* von Android sowohl der Location Area Code (LAC) als auch die Mobilfunkzellenidentifikation (engl. Cell-ID) abgerufen. Mit Hilfe dieser Werte kann lediglich die Funkzelle bestimmt werden, in welcher sich das Smartphone derzeitig befindet. Funkzellen haben sehr unterschiedliche Ausbreitungen. Während sie in ländlichen Bereichen einen Durchmesser von mehreren Kilometern haben können, haben sie in urbanen Gebieten nur selten einen Durchmesser, der größer als 200m ist [66]. Diese Informationen über das Mobilfunknetz werden ebenso gespeichert, verifiziert und gelernt, wie die oben genannten Funknetzwerkdaten.
Sollte kein passendes Tupel an Mobilfunkinformationen in der METDS-Datenbank gefunden werden, so wird dem Benutzer eine Warnung angezeigt und der Verbindungsaufbau mit dem zu verbindenden Access Point wird sofort unterbrochen. In seiner derzeitigen Konfiguration wird keine Warnmeldung angezeigt, wenn ausschließlich die Prüfung der Mobilfunkinformationen erfolgreich ist, nicht aber die Überprüfung der Funknetzwerkumgebung. Das Fälschen von Mobilfunkzellen liegt außerhalb der Möglichkeiten von Angreifern, die im Rah-

men dieser Arbeit abgewehrt werden sollen. Abgesehen davon handelt es sich hierbei ebenso um einen der Parameter des Erkennungssystems, der zu einem späteren Zeitpunkt angepasst werden kann.

Zustand „Unbekannte Position“

Die Bestimmung der Position eines Smartphones ist im Vergleich zu den bereits genannten Methoden zur Kontextbestimmung zeitintensiv und verbraucht deutlich mehr Energie. Insbesondere ist dies beim Einsatz des GPS-Moduls der Fall, wenn die Position exakt bestimmt wird. Aus diesen Gründen kommt die Bestimmung der Position im METDS nur dann zum Einsatz, wenn die vorangegangen Prüfungen negativ ausgefallen sind oder die entsprechenden Parameter nicht bestimmt werden konnten.

Zur Bestimmung der Position wird zunächst geprüft, ob auf dem Smartphone die Google Play Services verfügbar sind. Sind diese verfügbar, so kann die Position deutlich schneller (in den meisten Fällen in unter einer Sekunde) bestimmt werden. Ermöglicht wird dies den Google Play Services, indem sie die Position für viele Apps und Systemdienste bestimmen. Der Dienst nutzt einen Zwischenspeicher und kann auf diese Weise bei neuen Anfragen von Apps und Diensten eine Position, die noch aktuell genug ist, innerhalb kürzester Zeit liefern. Nur falls diese Position nicht mehr aktuell genug ist, wird mit Hilfe von Mobilfunkinformationen, WLANs und GPS eine neue, aktuelle Position mit der benötigten Genauigkeit bestimmt. Da die Google Play Services von vielen Apps und Diensten genutzt werden, ist die Wahrscheinlichkeit für eine aktuelle Position im Zwischenspeicher groß. Auf diese Weise kann die Genauigkeit der bestimmten Positionen verbessert und gleichzeitig ein niedrigerer Energieverbrauch erreicht werden.

Sollte eine Bestimmung der Position mit Hilfe der Google Play Services nicht möglich sein, so wird ersatzweise die native *Android Location API* genutzt. In diesem Fall müssen die Nachteile der nativen Schnittstelle gegenüber dem Google Play Services Framework in Kauf genommen werden. Die Studie hat gezeigt, dass die Google Play Services inzwischen so verbreitet sind, dass 99% der Positionsbestimmungen während der Studie mit deren Hilfe erfolgen konnten. Dies zeigt einmal mehr, dass die Nutzung der nativen Schnittstelle ausschließ-

lich als Ausweichlösung dient, die nur äußerst selten zum Einsatz kommen würde.
Nachdem die Position über einen der beschriebenen Wege bestimmt worden ist, wird sie mit der für den aktuellen Access Point gespeicherten verglichen. Wenn die bestimmte Position innerhalb eines spezifischen Radius um den gespeicherten Wert liegt, so wird die neue Position als bekannt angesehen. Diese Methode wird genutzt, um der teilweise fehlenden Genauigkeit bei der Positionsbestimmung Rechnung zu tragen. Die Verbindung zum Access Point wird zugelassen und die neue Position wird mit der gespeicherten zusammengeführt. In diesem Fall wird keine Warnung ausgegeben.
Die Bestimmung des Radius, in dem zwei Positionen als gleich angesehen werden, ist an dieser Stelle entscheidend. Einerseits soll die Genauigkeit und somit die Sicherheit möglichst groß sein. Andererseits soll die Anzahl unbegründeter Warnungen möglichst klein gehalten werden. In einer Parameterstudie konnte gezeigt werden, dass Radien kleiner als 100m zu viele Falsch-Positiv-Meldungen erzeugen. Um auch an dieser Stelle ein konservatives Erkennungssystem zu entwickeln, wurde deshalb der Radius zunächst auf 100m festgelegt. Dieser Wert wird auch in den Simulationen der folgenden Abschnitte verwendet, kann aber jederzeit im Rahmen der Konfiguration des Erkennungssystems angepasst werden.

8.1.2 Gesamtarchitektur

Im Folgenden wird die Architektur des Mobile Evil Twin Detection Systems und seine Komponenten näher beschrieben. In Abbildung 8.2 ist die Gesamtarchitektur dargestellt.
Die zentrale Komponente des Erkennungssystems ist die *ETPEngine*. Hier werden die Verbindungen zu WLANs mit Hilfe des *connectionHandlers* kontrolliert. Sobald eine Verbindung aufgebaut werden soll tritt der *connectionValidator* in Kraft und prüft durch die oben genannten Methoden, ob eine Verbindung ungefährlich ist und durchgeführt wird oder ob sie zunächst verhindert und dem Benutzer eine Warnung angezeigt wird. Eine weitere Komponente der *ETPEngine* ist der *KnowledgeImprovementHandler*. Dieser steuert das Lernen der verschiedenen Kontexte (WLAN, Mobilfunknetz, Position). Die *ETPEngine* nutzt für die Bestimmung der Kontexte die im linken Teil

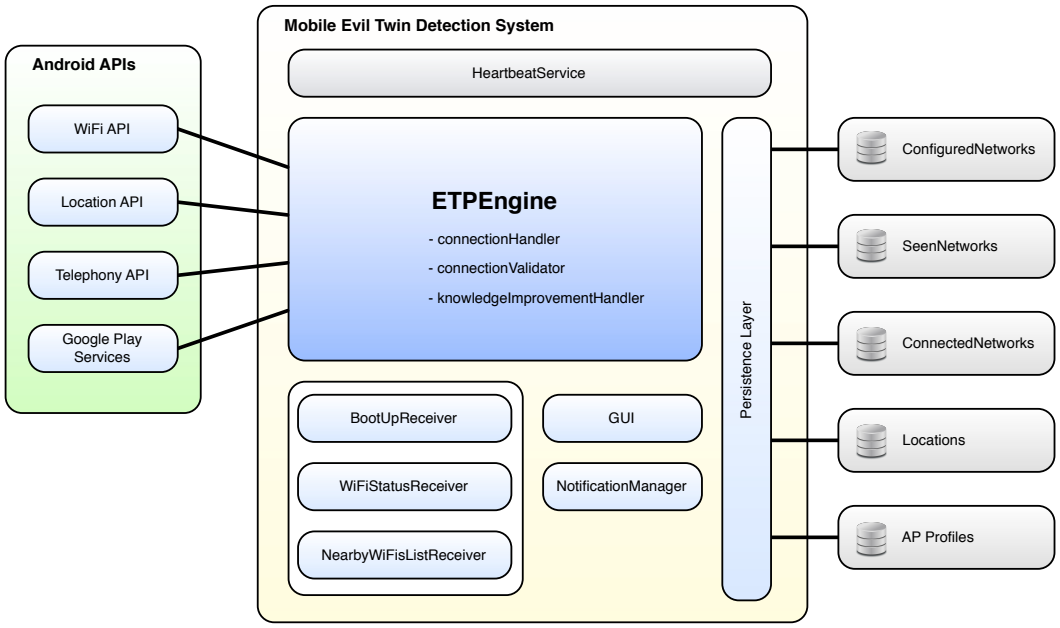


Abbildung 8.2: Überblick über die Gesamtarchitektur des Erkennungssystems

der Abbildung dargestellten nativen Schnittstellen des Android-Systems: `WiFi`, `Location`, `Telephony` und die *Google Play Services*. Ebenfalls werden zur Ermittlung der benötigten Kontextdaten die beiden Receiver *WifiStatusReceiver* und *NearbyWiFisListReceiver* benötigt. Der *WifiStatusReceiver* wird durch den zugrundeliegenden Dienst als erstes registriert. Er wird über Veränderungen des WLAN-Status durch das Betriebssystem benachrichtigt. Wird eine Verbindung zu einem WLAN aufgebaut, so ändert sich der Status der Verbindung und der *WifiStatusReceiver* setzt die Kontextvalidierung in Gang. Der *NearbyWiFisListReceiver* wird für die Ermittlung der umgebenden WLANs benötigt und benachrichtigt die *ETPEngine* nach einem erfolgreich durchgeführten Scan der Umgebung über dessen Ergebnis.

Der *HeartbeatService* ist ein weiterer Dienst, der zu Beginn der Nutzung im Hintergrund gestartet wird. Er überprüft, ob der METDS-Dienst läuft und startet ihn für den Fall eines unerwarteten Fehlers neu. Auf diese Weise kann sichergestellt werden, dass das Erkennungssystem und der mit ihm verbundene Schutz des Benutzers vor Angreifern dauerhaft aufrecht erhalten werden kann. Für den Fall eines Neustarts des Android-Smartphones, wird beim ersten Start ebenfalls ein *BootUpReceiver* registriert. Dieser sorgt nach einem Neustart des

Geräts dafür, dass der METDS-Dienst ebenfalls automatisch neu gestartet wird und seine Schutzaufgabe weiterhin erfüllen kann.
Zur Speicherung aller Kontextinformationen beinhaltet das METDS einen *PersistenceLayer*, welcher für die Speicherung aller relevanten Informationen und dessen Verknüpfungen verantwortlich ist. Gespeichert werden die Daten in sqlite-Datenbanken für die konfigurierten, beim Scan gefundenen und verbundenen Netzwerke. Darüber hinaus werden Informationen zu den Positionen der Access Points und die beschriebenen Access Point Profile gespeichert.
Des Weiteren ist die graphische Oberfläche Bestandteil des Gesamtsystems. Alle Informationen und Einstellungsmöglichkeiten des METDS lassen sich hierüber einsehen und steuern. Der *NotificationManager* sorgt mittels nativer Android-Benachrichtigungen dafür, dass der Benutzer des Systems stets über den aktuellen Stand des METDS aufgeklärt ist. Wird bei der Verbindung zu einem WLAN eine potentiell gefährliche oder verdächtige Umgebung erkannt, so wird mit Hilfe des *NotificationManagers* eine Benachrichtigung in der Mitteilungszentrale von Android angezeigt. Der Benutzer kann direkt in der Benachrichtigung entscheiden, ob die Verbindung zugelassen oder abgebrochen werden soll.

8.1.3 Zusammenfassung

Der Algorithmus des beschriebenen Erkennungssystems versucht, möglichst viele kontextuelle Parameter für seine Entscheidung zu nutzen und hiermit zu entscheiden, ob ein Evil Twin Access Point vorliegt oder nicht. Alle hierfür erforderlichen Parameter werden über Sensoren und Schnittstellen eines gebräuchlichen Android-Smartphones ermittelt. Die gesammelten Daten und erstellten Profile des METDS werden in einer lokalen Datenbanken gespeichert, die das Gerät zu keinem Zeitpunkt verlassen. Auf diese Weise ist einerseits die Privatsphäre des Benutzer geschützt und darüber hinaus ist das Gesamtsystem vollständig abgeschlossen und benötigt für seine Entscheidungen keine weitere Kommunikation mit zusätzlichen Servern oder zusätzlicher Infrastruktur.
Ein weiteres Augenmerk lag während der Planung auf der Energieeffizienz des Erkennungssystems. Es soll auf Smartphones laufen und darf entsprechend nur einen geringen Anteil der zur Verfügung stehenden Akkukapazität für sich beanspruchen. Wäre dieser Aspekt bei der Entwicklung nicht beachtet worden,

so hätte ein möglicherweise entstandenes System durch seinen Energiehunger und die daraus resultierende, verkürzte Akkulaufzeit dazu geführt, dass die Akzeptanz für ein solches System vermindert wird. Aus diesem Grund werden energiesparende Sensoren bevorzugt durch den Erkennungsalgorithmus eingesetzt. Nur als Ausweichlösung werden energiehungrige Sensoren wie GPS eingesetzt.
Die durch das Erkennungssystem erzeugten Warnungen wurden ebenfalls beschrieben und sollen den Benutzer vor eventuellen Gefahren warnen. Es obliegt dem Benutzer, die Warnmeldungen zu ignorieren oder einen Abbruch des Verbindungsaufbaus zu veranlassen. Abschließend wurde die Gesamtarchitektur, ihre Komponenten und deren Zusammenspiel erläutert.

8.1.4 Entwicklung von Erkennungsstrategien

In diesem Abschnitt wird der zugrundeliegende Erkennungsalgorithmus diskutiert und erläutert, wieso im Rahmen dieser Arbeit auf eine zentrale Instanz zur Datenanalyse verzichtet wurde. Die verschiedenen während der Entwicklung und Implementierung des Systems untersuchten Methoden werden dargestellt und erläutert welche Gründe zu dem letztendlich entstandenen System geführt haben.

Entwicklung des Erkennungsalgorithmus

Bei der Entwicklung des Erkennungssystems standen eine Vielzahl von Datenquellen zur Verfügung. Die im Rahmen der Feldstudie aus Abschnitt 7.2 ausgewählten und aufgezeichneten Daten konnten uneingeschränkt bei der Entwicklung und Konfiguration des Systems genutzt werden. In einem ersten Schritt wurden diejenigen Parameter ausgewählt, die bereits im Vorfeld der Studie als vielversprechende Parameter ausgemacht worden waren. Hierzu zählten neben der SSID und der BSSID auch die aktuelle Position des Geräts. In einer ersten Variante des Erkennungssystems kamen nur diese drei Parameter zum Einsatz. Die Reihenfolge der Überprüfung wurde zunächst folgendermaßen festgelegt: Die SSID ist der Name des zu verbindenden Netzwerks. Entsprechend wird dieser als erstes dazu genutzt, zu überprüfen, ob bereits im Vorfeld eine Verbindung zu diesem Netzwerk aufgebaut wurde. In einem zweiten Schritt kommt

die Überprüfung der BSSID hinzu. Es wird also geprüft, ob sich das Smartphone in der Vergangenheit bereits einmal mit dem vorliegenden Access Point verbunden hat. Der dritte Parameter ist die Position. Mit Hilfe dieser kann bestimmt werden, ob sich ein bereits bekannter Access Point an der selben Position befindet, wie der aktuelle zu verbindende.

Sowohl die SSID als auch die BSSID des zu verbindenden Access Points lassen sich unmittelbar bestimmen. Die Position hingegen ist, wie beschrieben, mit einem erhöhten Zeit- und Energieaufwand verbunden. Aus diesem Grund wurde ein alternativer Parameter verwendet, der mindestens so schnell bestimmbar ist, dabei aber einen deutlich geringeren Energiebedarf aufweist. Hierfür eignet sie die Bestimmung der umgebenden Funknetzwerke. Auf diese Weise lässt sich in erster Linie zwar nicht die Position bestimmen, in belebten und urbanen Gebieten ergeben sich durch die Kombination vorgefundener Funknetzwerke eindeutige Merkmale für den aktuellen Kontext. In einem zweiten Schritt wurde entsprechend die Bestimmung der umgebenden Funknetzwerke in den Algorithmus integriert. Hierfür wird beim Verbindungsaufbau ein Scan gestartet, um die umgebenden Netzwerke zu finden. Diese werden im Anschluss mit den Netzwerken des gespeicherten Kontextes verglichen. In einem weiteren Schritt wurden weitere Kontextinformationen für die Entscheidung herangezogen. So wird zusätzlich die aktuelle Funkzelle des Mobilfunknetzes dazu genutzt, den Kontext weiter einzugrenzen. Mit Hilfe der Mobilfunkzelle kann je nach Mobilfunknetz nur sehr grob die Position bestimmt werden. In Kombination mit allen anderen Parametern kann auf diese Weise hingegen ein detaillierterer Kontext gebildet werden.

Im Rahmen von Simulationen auf den in der Feldstudie gesammelten Echtweltdaten wurde ebenfalls untersucht, wie sich eine initiale Lernphase auf die Erkennungsrate des Gesamtsystems auswirkt. Für diese Untersuchung wurden alle innerhalb einer vorgegebenen Periode aufgebauten Verbindungen durch das System akzeptiert und die entsprechenden Kontextinformationen gelernt. Es wurde im Rahmen dieser Untersuchung davon ausgegangen, dass in dieser ersten Phase keine Verbindungen zu bösartigen Access Points aufgebaut werden. In dieser Periode wurden dem Benutzer entsprechend keinerlei Warnungen angezeigt. Durch die Simulationen verschieden langer initialer Lernphasen konnte keine gesteigerte Genauigkeit erzielt werden. Zwar war das Erkennungs-

system in den meisten Fällen direkt nach der initialen Lernphase mit deutlich mehr Informationen über Kontexte bestückt als das System ohne diese Lernphase. Der Unterschied zwischen einem System mit und einem ohne initialer Lernphase war nach kurzer Zeit vernachlässigbar. Bei längerer Benutzung des Systems haben sich keine Vorteile durch die initiale Lernphase gezeigt. Aus diesem Grund wird in der aktuellen Implementierung auf eine Lernphase verzichtet. Über einen Parameter in der Konfiguration kann diese initiale Lernphase auf einfache Art und Weise wieder aktiviert und ihre Länge festgelegt werden.
Der die vorangegangenen Abschnitte beschriebene und für die folgenden Simulationen verwendete Algorithmus 1 des Erkennungssystems ist im Folgenden dargestellt.

Zentralisiertes Erkennungssystem

Bei der Entwicklung von Sicherheitssystemen, die dem Schutz vor Angriffen dienen stellt sich unweigerlich die Frage, ob ein autonomer oder ein verteilter Ansatz verfolgt werden soll. Bei einem autonomen System ist jedes mit dem Sicherheitssystem ausgestattete Gerät auf sich alleine gestellt. Es besitzt ausschließlich Informationen, die dem Grundsystem zuzuordnen sind und solche Informationen, die es während der Laufzeit selbst gesammelt oder aggregiert hat. In einem verteilten System hingegen kommunizieren die dem Sicherheitssystem angeschlossenen Geräte entweder miteinander oder mit einer zentralen Instanz. Jede einzelne Instanz kann über die Informationen des Gesamtsystems verfügen und rechenintensive Aufgaben können an zentraler Stelle durchgeführt und die Ergebnisse den Teilnehmern zur Verfügung gestellt werden. Im Folgenden sollen die Vor- und Nachteile der jeweiligen Ansätze beschrieben werden. Ebenso wird dargestellt, aus welchen Gründen der autonome Ansatz für das im Rahmen dieser Dissertation entstandene Sicherheitssystem gewählt wurde.
Für die Verwendung eines verteilten Systems spricht die potentiell bessere Möglichkeit der Erkennung von Unregelmäßigkeiten. Die durch mindestens zwei aber bis hin zu tausenden Personen gesammelten Daten liefern ein deutlich umfangreicheres Bild, als es einzelnen Teilnehmern autonom möglich ist. Aber nicht nur eine vermeintlich bessere Erkennungsleitung spricht für diesen

```
if SSID ist bekannt then
    if BSSID ist bekannt then
        if Netzwerkumgebung ist bekannt then
            Verbessere Netzwerkumgebung
            Verbessere CellInfo
            Rückmeldung Verbindung OK
        else
            if CellInfo ist bekannt then
                Verbessere CellInfo
                Rückmeldung Verbindung OK
            else
                if Position ist verfügbar then
                    if Position ist bekannt then
                        Verbessere Position
                        Rückmeldung Verbindung OK
                    else
                        Warnung: Unbekannte Umgebung
                        Verbindungsabbruch
                else
                    Warnung: Keine Position verfügbar
                    Verbindungsabbruch
        Entscheidungsdialog "Neuer Access Point":
        Rückmeldung Verbindung OK & Lerne neuen Access Point
        - oder -
        Verbindungsabbruch
else
    Entscheidungsdialog "Neues Netzwerk":
    Rückmeldung Verbindung OK & Lerne neues Netzwerk
    - oder -
    Verbindungsabbruch
```

Algorithmus 1 : Erkennungsalgorithmus des METDS

Ansatz. Auch ein Monitoring der aktuellen Gefahrenlage wäre mit einer zentralisierten Datenhaltung deutlich einfacher möglich. Aus diesem Monitoring generierte Bericht könnten dazu dienen, gezielt Gegenmaßnahmen zu planen. Es gibt viele weitere Möglichkeiten, die für einen derartigen verteilten Ansatz sprechen.

Demgegenüber stehen hingegen viele Probleme und kritische Fragestellungen, die die Nutzung eines verteilten Ansatzes schwierig erscheinen lassen. Die zunächst positiv erscheinende zentrale Datensammlung der Daten, die der beschriebene Algorithmus benötigt, bringt aus Sicht des Datenschutzes erhebliche Probleme mit sich. Die Sammlung zum Teil privater Informationen, wie SSIDs, BSSIDs und Ortsinformationen an einer zentralen Stelle birgt die große Gefahr des Missbrauchs dieser Daten durch Angreifer, staatliche Organe oder für wirtschaftliche Interessen von Unternehmen. Es müsste für die Verwendung eines zentralisierten Ansatzes ein Konzept erarbeitet werden, welches die gesammelten privaten Daten anonymisiert oder pseudonymisiert, gleichzeitig aber die Erkennungsleistung des Gesamtsystems durch diese Maßnahmen nicht schmälert. Ähnliche Konzepte sind im Rahmen eines Grid-Projektes [38] der D-Grid Initiative erforscht und implementiert worden. Auch im Rahmen dieses Projekts wurden Fragestellungen bezüglich der Datenhoheit vielfach und intensiv diskutiert. Im Rahmen des D-Grids mit seinem vergleichsweise kleinen Nutzerkreis war eine Einigung möglich [69]. Eine entsprechende Vereinbarung und gesetzliche Regelung für ein Sicherheitssystem, wie es im Rahmen dieser Dissertation erarbeitet wird, dürfte deutlich schwieriger zu erarbeiten und umzusetzen sein. Ein weiterer wesentlicher Nachteil hinsichtlich des Datenschutzes stellt die Akzeptanz der Nutzer eines solchen Systems dar. Der Nutzer muss zur Teilnahme am entsprechenden Sicherheitssystem explizit der Datensammlung und -nutzung zustimmen. Ob Benutzer ein derartiges System akzeptieren und einsetzen würden ist fraglich.

Aber nicht nur der Datenschutz stellt große Herausforderungen dar. Auch die Anforderungen an die Sicherheit eines zentralisierten Systems übersteigen die eines autonomen Systems in großem Maße. Während bei einem autonomen System die Daten nur lokal auf den entsprechenden mobilen Geräten vorgehalten werden, sind bei einem Angriff auf ein solches Gerät auch nur die Daten eines einzelnen Benutzers gefährdet. Bei einem Angriff auf ein zentrales Sys-

tem hingegen sind die Daten aller teilnehmenden Nutzer betroffen. Ein solches zentrales System muss entsprechend hohe Sicherheitsstandards erfüllen. Es gilt aber nicht nur die Datenhaltung zu sichern. Ebenso sind Angriffe hinsichtlich der Integrität der Daten zu verhindern. Ein Angreifer könnte den Datenbestand des zentralen Systems verändern. Hierbei könnte er zusätzliche Daten zu nicht existierenden Angriffen einfügen oder bestehende Daten löschen, um einen bereits erkannten Angriff erneut zu ermöglichen. Aus diesen Gründen muss die Integrität der Daten auf Seiten des Betreibers sichergestellt werden.

Ein weiteres Problem besteht in den Vertrauensverhältnissen. Bekannt aus dem PKI-Bereich bringen die Vertrauensbeziehungen zwischen Benutzern und zentralen vertrauenswürdigen Instanzen stets weitere Probleme mit sich, die sich auch im Rahmen eines solchen Systems ergeben würden. Auch hier müsste einer zentralen Instanz vertraut werden. Ebenso ist eventuell der Aufbau einer hierarchischen Struktur erforderlich, was bezüglich der Vertrauensverhältnisse einer PKI gleichkommen würde.

Bezogen auf das konkrete Sicherheitssystem würde sich noch ein weiteres sicherheitstechnisches Problem ergeben: der Austausch der Daten. Während die Übertragung der gesammelten Daten vom Benutzer in Richtung zentralem Service zu einem Zeitpunkt durchgeführt werden kann, wenn eine gute und stabile Verbindung zum Internet besteht, ist dies bei der Verhinderung von Angriffen während des Verbindungsaufbaus zu WLANs nicht möglich. Zu diesem Zeitpunkt besteht potentiell keine stabile Verbindung zum Internet, über die Daten zum verbundenen Access Point sicher übertragen werden können. Um dennoch die Sicherheit der Übertragung zu gewährleisten müssten die Daten vor der Verbindung mit dem Access Point über das Mobilfunknetz mit dem zentralen Dienst abgeglichen werden. Die Probleme an dieser Stelle ergeben sich einerseits dadurch, dass in manchen Fällen keine Verbindung zum Mobilfunknetz besteht. In diesen Fällen ist keine sichere Übertragung der Daten möglich und es kann durch das Gesamtsystem keine Entscheidung über den Status der aktuellen Verbindung gefällt werden. Anderseits führt der stetige Austausch von Informationen zwischen dem zentralen Service und dem mobilen Gerät des Benutzers zu einem erhöhten Datenverbrauch, für den der Benutzer gegenüber seinem Mobilfunkanbieter aufkommen muss. Je nach Anzahl der Verbindungen zu Access Points und Menge der selbst gesammelten

Daten können hier erhebliche Datenmengen auftreten. Die Akzeptanz eines solchen Systems dürfte ebenfalls unter dieser Tatsache leiden.

Sollten alle zuvor genannten Probleme beim Aufbau und Betrieb eines solchen zentralen Systems zufriedenstellend gelöst werden können, so besteht weiterhin das Problem der Finanzierung. Während bei einem autonomen System keinerlei zusätzliche Infrastruktur benötigt wird, fallen bei einem zentralen System neben der benötigten Hardware auch Kosten für eine Anbindung ans Internet und für Personal zur Wartung und zum Betrieb an. Es bleibt also die Frage offen, wie hoch entsprechende Kosten wären und wer hierfür aufkommen würde. Ein System, welches der Benutzer bezahlen muss, wird einer schnellen Verbreitung im Wege stehen.

Aus den genannten Gründen wurde auf eine zentrale Instanz bei dem im Rahmen der Dissertation erarbeiteten System verzichtet. Die Vorteile der autonomen Lösung überwiegen und nur auf diese Weise kann ein leichtgewichtiges und schnell an aktuelle Bedürfnisse anpassbares System entwickelt werden.

8.2 Portierung auf Android

Bei der Implementierung des Erkennungssystems unter Android konnten dank der zielgerichteten Entwicklung des Systems im Vorfeld große Teile direkt für den Einsatz im mobilen Bereich übernommen werden. Vor der eigentlichen Portierung gilt es jedoch, die Vor- und Nachteile der verschiedenen Lösungsansätze zu beleuchten.

Es gibt grundsätzlich zwei Arten der Implementierung des Erkennungssystems für die mobile Plattform Android. Einerseits kann das Erkennungssystem in Form einer normalen Android-App auf dem Smartphone installiert werden. Ein anderer Weg der Implementierung ist die Integration in das mobile Betriebssystem selbst. Die verschiedenen Möglichkeiten sollen im Folgenden erläutert und diskutiert werden. Im Anschluss daran wird beschrieben, welche Möglichkeit der Implementierung im Rahmen dieser Arbeit gewählt wurde und welche Gründe es hierfür gibt. Des Weiteren wird auf die Benutzerfreundlichkeit des portierten Erkennungssystems und seine Darstellung im Betriebssystem eingegangen. Abschließend wird erläutert, welche Aufgaben und Herausforderungen

noch zu erfüllen sind, um das im Rahmen dieser Arbeit prototypisch implementierte System zu einem produktiven Einsatz zu führen.

8.2.1 Unterschied zwischen App- und Systemintegration

Die bereits kurz beschriebenen Möglichkeiten der Bereitstellung des Erkennungssystems unter Android sollen nun zusammen mit ihren Vor- und Nachteilen erläutert werden. Hierbei wird zunächst auf die Nachteile einer Programmierung als App und den entsprechenden Vorteilen einer Systemintegration eingegangen.

Ein wesentlicher Nachteil bei einer Realisierung als App ist die Tatsache, dass auf diese Weise ausschließlich auf die native Android API zurückgegriffen werden kann. Die direkte Kommunikation sowohl mit der WLAN-Schnittstelle selbst als auch mit Gerätetreibern ist nicht vorgesehen. Aus diesen Gründen müssen Einschränkungen bei der Entwicklung in Kauf genommen werden. Eine dieser Einschränkungen besteht darin, dass mit Hilfe der nativen Android-API lediglich auf veränderte Zustände der WLAN-Verbindung reagiert werden kann. Entsprechend wird eine vollständig aufgebaute Verbindung, also eine Verbindung die sämtliche hierfür definierte Zustände `Associating`, `Associated`, `Authenticating` und `Completed` erfolgreich durchlaufen hat, evaluiert und im Anschluss zunächst wieder getrennt. Im Rahmen der Evaluierung werden auch die ebenfalls benötigten weiteren Kontextparameter ermittelt. Bei positivem Ergebnis der Prüfung wird die Verbindung ein weiteres Mal aufgebaut. Bei einer Realisierung in Form der Integration in Android bestehen weiterhin mehrere Möglichkeiten. Zum einen könnte hierfür der zugrundeliegende wpa_supplicant [1] angepasst werden. Zum anderen wäre die Anpassung der Android API hinsichtlich der Wartbarkeit und der Standardkonformität die bessere Wahl. Hier könnte man den Verbindungsaufbau, ähnlich wie in der App-basierten Lösung steuern und beeinflussen, könnte aber einen weiteren Nachteil der App-basierten Lösung, eine mögliche Deinstallation der App, verhindern.

Der App-basierte Lösungsansatz bietet neben den oben beschriebenen Nachteilen auch zahlreiche Vorteile gegenüber einer Systemintegration. Für die Ver-

[1] `http://w1.fi/wpa_supplicant/`

breitung eines derartigen Sicherheitssystems ist die Lösung in Form einer App auf jeden Fall zu bevorzugen. Diese kann mit geringem Aufwand im Google Play Store veröffentlicht, vermarktet und somit verbreitet werden. Ebenso besteht bei diesem Vorgehen der Vorteil, schnell und unkompliziert Updates und Erweiterungen des Erkennungssystems an Benutzer verteilen zu können. Bei einer in das System integrierten Lösung sind in den meisten Fällen längere Wartezeiten bis zu einer Veröffentlichung zu erwarten, da die Verteilung von Betriebssystemupdates in der Regel über die jeweiligen Hersteller der Smartphones durchgeführt wird. Diese nehmen in vielen Fällen weitere Anpassungen vor, können aber auch entscheiden, dass ein Update für Geräte bestimmter Typen nicht mehr verfügbar gemacht wird. Auch Aktualisierungen und Erweiterungen des Erkennungssystems wären in einem solchen Fall nicht mehr möglich. Bei einer Realisierung als App sind keine Anpassungen am Betriebssystem notwendig, wodurch der Verteilung im Google Play Store nichts im Wege steht.
Ein weiterer im Rahmen dieser Dissertation gewichtiger Vorteil der App-basierten Lösung ist die Möglichkeit das System schnell und einfach auf verschiedenen Plattformen zu testen. Auf diese Weise ist es nicht nur möglich, das System auf vielen verschiedenen Hardwareplattformen / Smartphones zu testen um Eigenheiten bestimmter Hard- und Softwarekombinationen zu ermitteln und zu berücksichtigen. Es ist darüber hinaus auch deutlich einfacher das System unter verschiedenen Versionen des Android-Betriebssystems zu testen. Auf diese Weise kann die Lauffähigkeit und Stabilität des Systems in den gängigen und aktuellen Versionen von Android untersucht werden.
Aus den beschriebenen Gründen wurde im Rahmen dieser Arbeit auf die Integration in das Betriebssystem verzichtet und die App-basierte Lösung angestrebt.

8.2.2 Umsetzung und Darstellung im Betriebssystem

Bei der Portierung des im Rahmen der Simulation entstandenen und verwendeten Erkennungssystems ergaben sich nur wenige Detailprobleme, für die es Lösungen zu suchen galt.
In Android ist die direkte Verbindung mit einem spezifizierten Netzwerk nicht möglich. Es kann lediglich der wpa_supplicant angestoßen werden, sich mit

einem verfügbaren WLAN zu verbinden. Hierbei sucht dieser nach WLANs in der Umgebung und verbindet sich mit dem ihm bekannten, gespeicherten Netzwerk mit der höchsten Signalstärke. Da bei dem im Rahmen dieser Arbeit entwickelten Erkennungssystem in Form einer App zunächst die Verbindung unterbrochen wird, muss im zweiten Schritt aber genau eine solche Verbindung zu einem spezifizierten Netzwerk hergestellt werden. Hierfür wurde eine Lösung gefunden, die auf der Deaktivierung von gespeicherten Netzwerken beruht. In Android gibt es die Möglichkeit, bestimmte BSSIDs für zukünftige Verbindungen zu unterdrücken. Hierfür werden diese in eine Blacklist des wpa_supplicant aufgenommen. Die Aufnahme einer BSSID auf diese Blacklist verhindert aber leider nicht gänzlich eine zukünftige Verbindung mit diesem Access Point. Stattdessen wird die Liste dazu verwendet, Access Points zu erkennen, die bei einer Verbindung mit einer niedrigeren Priorität ausgewählt werden sollen. Eine BSSID gelangt auf diese Blacklist in dem sie einerseits manuell hinzugefügt wird. In der Regel wird die Blacklist aber durch den wpa_supplicant selbst gefüllt. Er nutzt die Liste, um den Zugriff auf Access Points zu vermeiden, bei denen ein vorangegangener Verbindungsversuch fehlgeschlagen ist. Hierfür wird einer BSSID ein Wert zugeordnet, der angibt, wie oft schon ein Problem mit dem entsprechenden Access Point erkannt wurde. Je höher diese Zahl ist, desto später wird der wpa_supplicant diesen Access Point verwenden, solange ein anderer Access Point verfügbar ist. Der Name Blacklist ist an dieser Stelle irreführend, weil durch entsprechende Einträge nicht die Verbindung zu den Access Points verhindert wird, sondern ausschließlich die Reihenfolge der Access Points festgelegt wird, mit welcher die nächste Verbindung aufgebaut wird. Die Funktion zum direkten Hinzufügen von Access Points auf die vom wpa_supplicant geführten Liste ist im zentralen `WifiManager` nicht vorgesehen. Stattdessen wird hier die Methode `disableNetwork` angeboten, mit der die Verbindung zum entsprechenden Netzwerk unterbunden werden kann. Das Erkennungssystem deaktiviert entsprechend alle konfigurierten abgesehen von dem zu verbindenden. Nach der Verbindung mit dem spezifizierten Netzwerk werden alle übrigen Netzwerk reaktiviert. Auf diese Weise kann das angestrebte Verhalten erzielt werden.

Um dem Benutzer die Möglichkeit zu geben, möglichst einfach und schnell auf Warnungen und Nachfragen des Systems reagieren zu können wurden die

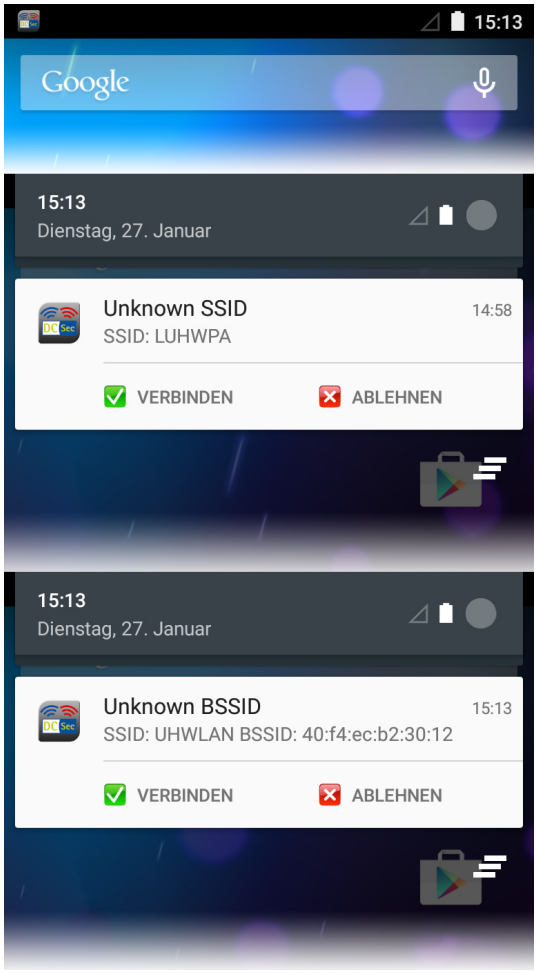


Abbildung 8.3: Darstellung des METDS in der Android-Oberfläche. Im oberen Teil die dezente Benachrichtigung über ein Ereignis, auf das reagiert werden kann. Im mittleren und unteren Bild befinden sich verschiedene Warnmeldungen.

Android-typischen Benachrichtigungen (Notifications) zur Anzeige und zur Bestätigung von Verbindungen verwendet. In Abbildung 8.3 wird die Darstellung von Warnungen im *Notification Center* von Android gezeigt. Die Benutzer sehen anhand eines in der Statusleiste eingeblendeten Icons, dass das METDS eine Verbindung geprüft hat und hierbei nicht mit ausreichender Sicherheit die Gefährlichkeit der Verbindung ausschließen kann. Durch eine Wischgeste vom oberen Bildschirmrand kann der Benutzer seine Benachrichtigungen anzeigen. Durch eine hohe vergebene Priorität der METDS-Benachrichtigungen werden die ebenfalls in Abbildung 8.3 dargestellten vollständigen Warnmel-

dungen zusammen mit den zur Reaktion benötigten Buttons im Notification Center im oberen Bereich angezeigt. Der Benutzer kann bereits aus dem *Notification Center* heraus die Verbindung zum entsprechenden Netzwerk bestätigen und somit zulassen oder ablehnen. Bei einer positiven Bewertung der Situation wird die Verbindung wie oben beschrieben erneut aufgebaut und ein Lern- und Anpassungsprozess des aktuellen Kontextes wird gestartet. Bedingt durch den neuen Verbindungsaufbau kann es in Netzwerken mit mehreren zulässigen Access Points dazu kommen, dass im Anschluss an eine Bestätigung der Verbindung erneut eine Warnungsmeldung über einen unbekannten Access Point angezeigt wird. In diesem Fall wurde bei der erneuten Verbindung nicht der bestätigte Access Point der ersten Verbindung genutzt, sondern (beispielsweise durch geänderte Empfangsbedingungen) ein anderer. Diese Einschränkung wird in dieser prototypischen Implementierung in Kauf genommen und wird im Rahmen dieser Arbeit nicht weiter beachtet, da sie die grundsätzliche Funktionsweise des Systems nicht beeinträchtigt. Sollte der Benutzer die Verbindung hingegen ablehnen, so wird keine Verbindung aufgebaut und das System wartet auf einen neuen zu bewertenden Kontext.
Abbildung 8.4 zeigt die zum Erkennungssystem gehörige App und ihre Oberfläche. In ihrer derzeitigen Version werden an dieser Stelle lediglich einige Statistiken über die aktuelle und vergangene Verbindungen angezeigt. Ebenso ist es dem Benutzer an dieser Stelle möglich, vergangene Warnmeldungen in der Historie zu betrachten. Im Rahmen einer Produktivführung könnten an dieser Stelle ebenfalls Einstellungen und Anpassungen des System platziert werden.

8.2.3 Produktivführung

In den vorangegangen Abschnitten wurde bereits angedeutet, dass die im Rahmen dieser Dissertation entstandene Implementierung des Systems für Android prototypischer Natur ist. Im Rahmen einer Produktivführung müssten weitere Verbesserungen der App vorgenommen werden, auf die an dieser Stelle eingegangen werden soll.
Beim vorgestellten Konzept handelt es sich um ein System zur Steigerung der Sicherheit in Funknetzwerken. Um die Sicherheit zu steigern muss hingegen auch das System selbst hinreichend gegen Angriffe und Manipulationsversuche von außen geschützt werden. Auf diese sicherheitsrelevanten und zum Teil

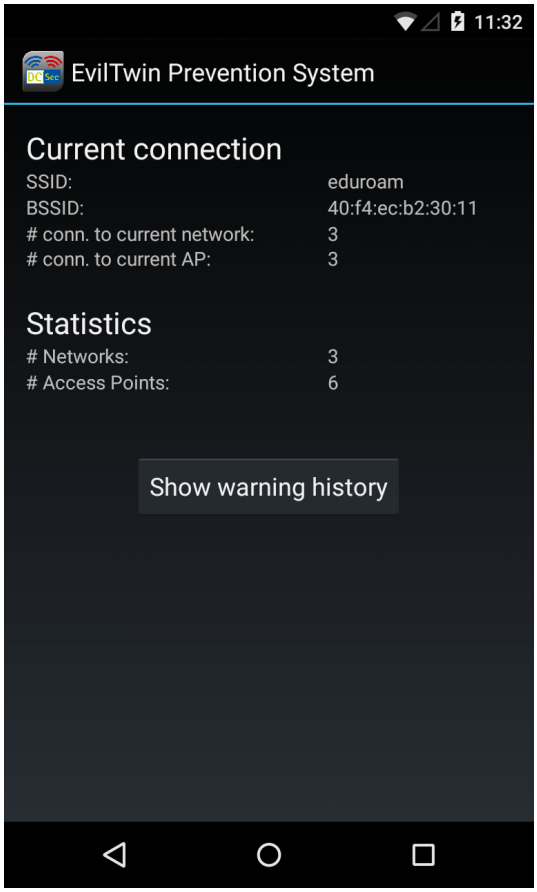


Abbildung 8.4: Oberfläche der METDS-App

nicht-funktionellen Anforderungen wurde bislang nicht eingegangen, da sie die Funktion des Gesamtsystems nicht beeinflussen dürfen. Im Rahmen einer Produktivführung muss die Datenhaltung der App vor Missbrauch und Manipulation geschützt werden. Um die während der Auswertung von Kontexten gesammelten Daten zu schützen könnte zukünftig eine verschlüsselte Datenbank auf dem mobilen Gerät eingesetzt werden.

Ein weiterer wichtiger Aspekt für die Verbreitung des Systems in Form eines öffentlichen Releases ist die Stabilität der App und des damit verbundenen Dienstes. Um eine hinreichende Stabilität gewährleisten zu können, müssen zahlreiche weitere Tests durchgeführt werden. Zum einen müssen verschiedene Smartphones mit entsprechend unterschiedlichen Firmwares getestet werden, um diesbezügliche Kompatibilitätsproblem ausschließen zu können. Darüber hinaus müssen auch die verschiedenen unterstützten Android Versionen und auch die Kombination aus Hersteller und Version getestet werden. Bei Tests

auf verschiedenen Systemen haben sich zum Teil Unterschiede ergeben, die eine Anpassung des Dienstes erforderlich gemacht haben. Vor einer Veröffentlichung müssten entsprechend möglichst viele Kombinationen untersucht werden, um derartigen Problemen begegnen zu können.

Kapitel 9

Evaluation des Erkennungssystems

In diesem Kapitel wird die Funktion des beschriebenen Erkennungssystems untersucht und evaluiert. Hierzu wird das Erkennungssystem simuliert. Die Daten, die der Simulation zugrunde liegen, wurden im Rahmen der Feldstudie durch die Teilnehmer gesammelt. Es werden im Folgenden der Simulator und die durchgeführten Simulationen beschrieben. Ebenso werden die Ergebnisse dargestellt und es wird gezeigt, dass das entwickelte Erkennungssystem unter den zuvor genannten Bedingungen zuverlässig Evil Twin Access Points erkennen kann. Ein weiteres Augenmerk bei der Untersuchung wird auf der Benutzbarkeit des Systems liegen. Es wird ebenso gezeigt, dass die Erkennungsrate des Systems so hoch ist, dass der Einsatz auf mobilen Geräten von Endnutzern möglich ist, ohne den gewohnten Einsatz der Geräte im Alltag maßgeblich einzuschränken. Teile dieses Kapitels basieren auf einer wissenschaftlichen Veröffentlichung [65] im Rahmen der internationalen Konferenz *Financial Cryptography and Data Security 2015*.

9.1 Simulation auf Echtweltdaten

Um die Funktionsweise des beschriebenen Erkennungssystems und des zugrunde liegenden Algorithmus zu evaluieren, wurde das Gesamtsystem simuliert. Als Datengrundlage wurden die Verbindungsdaten genutzt, die im Rahmen der Studie erfasst wurden.

Der vorgestellte Erkennungsalgorithmus nutzt zur Erkennung von Angriffen heuristische Methoden. Es wird versucht in möglichst kurzer Zeit mit dem begrenzten Wissen über die Umgebung des Smartphones eine Entscheidung darüber zu treffen, ob die Verbindung zu einem Access Point gefährlich erscheint oder ob die Verbindung als sicher eingestuft werden kann. Ein Sicherheitssystem hat zwar stets den Zweck, den Benutzer bestmöglich vor Angriffen zu schützen, zugleich ist aber zu beachten, dass die Anzahl an fälschlich erzeugten Warnmeldungen hierbei minimiert werden muss. Nur so kann die Nutzbarkeit eines derartigen Systems sichergestellt und der Einsatz im Alltag ermöglicht werden.

Die im Rahmen der Studie gesammelten Verbindungsdaten zu insgesamt 2326 verschiedenen Access Points beinhalten aufgrund der gesammelten Rohdaten der Verbindungen keine Aussagen darüber, ob sich die Smartphones der Teilnehmer in dieser Zeit mit bösartigen Access Points verbunden haben. Sollte dies der Fall gewesen sein, so ist die Wahrscheinlichkeit für Angriffe des Typs A und B sehr hoch. Es wird nicht davon ausgegangen, dass ein derartig gezielter und aufwändiger Angriff (wie er für die Angriffstypen C und D notwendig wäre) gegen einen der Teilnehmer der Studie während der Laufzeit durchgeführt wurde. Für die Evaluation des Gesamtsystems wird daher davon ausgegangen, dass keinerlei Angriffe aufgezeichnet wurden. Sollten entgegen dieser Annahme doch Angriffe des Typs A oder B gegen einen der Teilnehmer durchgeführt worden sein, so beeinträchtigt dies die nachfolgenden Simulationen zur Evaluation nur marginal. In diesen Fällen würden die gefälschten und potentiell bösartigen Access Points lediglich als weitere legitime Access Points behandelt werden. Da an dieser Stelle die Suche nach geeigneten Kontextparametern und die Anpassung des Erkennungssystems im Vordergrund stehen, haben die oben beschriebenen, kleinen Anomalien in den Verbindungsdaten nahezu keinen Einfluss auf das Gesamtergebnis der im Folgenden beschriebe-

nen Simulationen. Die Wahrscheinlichkeiten für Angriffe der Typen C und D erscheinen für die Teilnehmer der Studie derartig gering, dass sie an dieser Stelle vernachlässigt werden.

9.2 Beschreibung des Simulators

Der im Rahmen dieser Evaluation entstandene Simulator ist nicht mit dem in Kapitel 6 beschriebenen Simulator zur Untersuchung von Malware-Ausbreitungen zu vergleichen. Während der in Kapitel 6 beschriebene Simulator die Bewegung und Interaktion zwischen Smartphone-Benutzern auf Basis verschiedener Modelle simuliert verfolgt der hier beschriebene Simulator ein anderes Ziel. Basierend auf den gesammelten Echtwelt-Daten wird hier ausschließlich die Auswertung der Verbindungs- und Kontextdaten und das im Verlauf entstehende Wissen des Erkennungssystems simuliert. Die Simulationen verfolgen entsprechend das Ziel, die Parameter des Erkennungssystems zu bestimmen und den Algorithmus anzupassen. Der hierfür entstandene Simulator wurde, wie auch die zentrale Komponente zur Angriffserkennung und alle benötigten Bibliotheken, in Java implementiert. Auf diese Weise ist eine Portierung des Systems auf die Android-Plattform einfacher möglich, als bei der Verwendung anderer Programmiersprachen und Frameworks. Das zur Simulation entstandene System kann so auf einfache Art und Weise portiert und auf realen Smartphones genutzt werden. Die zugrundeliegende Architektur des Simulationsframeworks und des Simulators sind in Abbildung 9.1 dargestellt.
Die verschiedenen Datenbanken, welche mit dem *Data Access Layer* verbunden sind, beinhalten alle für die Simulationen erforderlichen Rohdaten. Die Verbindungsdaten befinden sich in der *Connection DB*, welche alle Informationen zu Verbindungen beinhaltet, die während der Studie aufgezeichnet, gespeichert und an den zentralen Server übermittelt wurden. Die Konfigurationsdatenbank enthält gerätespezifische Informationen, wie konfigurierte Netzwerke der Teilnehmer und weitere Metadaten der Verbindungen. In der dritten Datenbank sind alle Netzwerke gespeichert, die während der aufgebauten Verbindungen in der Umgebung der Teilnehmer aufgezeichnet werden konnten. Die Datenbanken sind über Schnittstellen an das Simulationsframework angebunden, um leicht austauschbar zu sein.

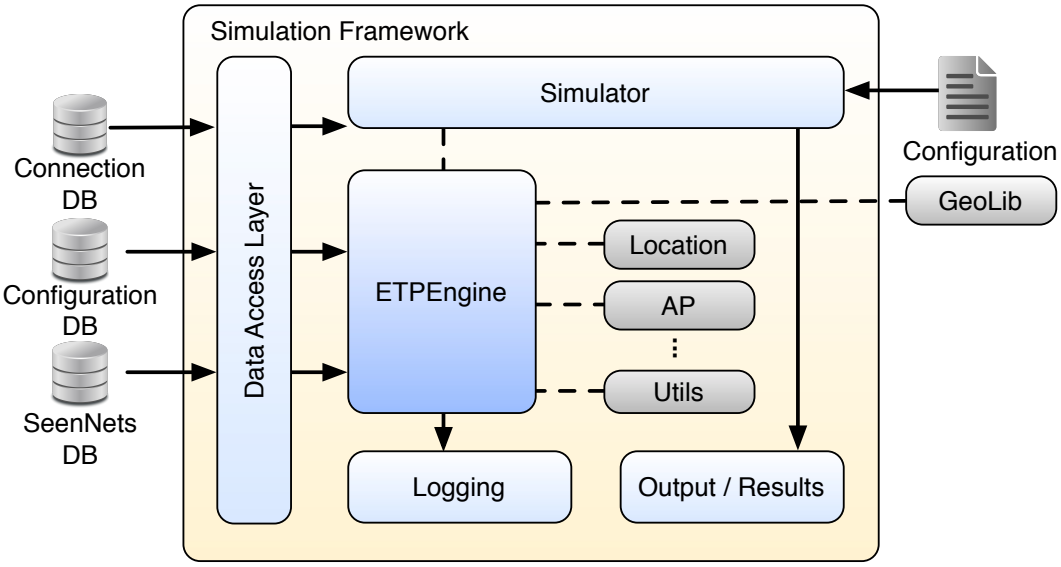


Abbildung 9.1: Architektur des Simulators zur Bestimmung geeigneter Kontextdaten und Parameter für ein Erkennungssystem

Der Simulator selbst verwaltet den Dateneingang, orchestriert weitere Komponenten des Frameworks und liefert das Ergebnis des Erkennungsprozesses an die Ausgabeeinheit. Die Ausgabeeinheit *Output* ist hierbei ebenso leicht austauschbar. Zur Auswertung der Simulationen wurde an dieser Stelle eine Textausgabe in eine Datei implementiert, um die Daten mit Hilfe anderer Programme weiter zu bearbeiten, analysieren und grafisch darzustellen. Ausgaben in anderen Formaten und über weitere Schnittstellen sind an dieser Stelle denkbar und leicht zu integrieren. Eine weitere Aufgabe der Simulator-Komponente ist das Einlesen der Konfiguration. Hier werden alle zur Simulation und für Parameterstudien benötigten Parameter und Einstellungen festgelegt.

Die zur Erkennung von Angriffen wesentliche Komponente ist die *ETPEngine*. Sie beinhaltet neben der Logik des Systems auch den zugrundeliegenden Erkennungsalgorithmus. Die *ETPEngine* nutzt für die Erfüllung ihrer Dienste eine Vielzahl weiterer Komponenten und Klassen. Hierbei handelt es sich im Wesentlichen um Klassen zur Speicherung und Analyse der Verbindungs- und Funknetzwerkumgebungsdaten. Die *Logging*-Komponente dient der Unterstützung während der Entwicklungszeit von Simulationen und Parameterstudien. Hiermit können weitere Ausgaben erzeugt und gespeichert werden, die zur

Überprüfung der korrekten Funktionsweise benötigt werden, nicht aber zum Ergebnis der Simulationen selbst gehören.

9.3 Ergebnisse der Simulationen

Um die Effektivität des Erkennungssystems unter Beweis stellen zu können wurden in einem ersten Schritt Simulationen mit dem oben beschriebenen Simulationsframework durchgeführt. Im Rahmen der Simulation wurden für alle betrachteten Benutzer die Daten genutzt, die während der Studie gesammelt werden konnten. Für jede aufgebaute Verbindung, welche in der Datenbank gespeichert ist, wurden die zugehörigen gesammelten Kontext- und Metadaten der Verbindung als Eingabe für die *ETPEngine* verwendet. Dies entspricht einer Verfahrensweise, als würde sich das Erkennungssystem auf einem mobilen Gerät befinden. Der Unterschied besteht lediglich in der Quelle der Sensorinformationen. Während die Daten im Android-System direkt über die verschiedenen Sensoren und Schnittstellen abgegriffen werden würden, werden sie im Rahmen der Simulation aus der Datenbank geladen. Der Bezug der Daten aus einer lokalen Datenbank hat für die Analyse des Erkennungssystems einen weiteren Vorteil. Während die Abfrage der Sensoren und beispielsweise die Ermittlung der Position über die Google Play Services durchaus mit einem kleinen Zeitaufwand verbunden ist, stehen die Daten für die Simulationen direkt und ohne Verzögerung bereit. Auf diese Weise kann die Geschwindigkeit der Simulationen gesteigert werden.
Alle Daten werden aus den im vorherigen Abschnitt beschriebenen Datenbanken entnommen und durch die *ETPEngine* dazu genutzt, eine Entscheidung darüber zu treffen, ob es sich bei einer Verwendung des Algorithmus auf einem Smartphone um eine sichere oder eine potentiell gefährliche Verbindung mit einem Access Point handelt. Bei dieser Art der Simulation muss bedingt durch den Versuchsaufbau auf einige Verhaltensweisen der realen Welt verzichtet werden. So ist in den für die Simulation vorliegenden Daten nicht ersichtlich, ob die Verbindung zu einem Access Point nach einer Warnung durch den Benutzer abgelehnt worden wäre. Während der Studie wurden jegliche Daten im Hintergrund aufgezeichnet. Die Teilnehmer sollten im Verlauf der Studie möglichst gar keine Notiz von der Studien-App nehmen, um so das normale

und unverfälschte Verhalten der Benutzer aufzeichnen zu können. Im Rahmen dieser Simulationen kann also ausschließlich davon ausgegangen werden, dass die aufgezeichneten Verbindungen legitime und somit vom jeweiligen Benutzer akzeptierte sind.
Als Datengrundlage für die Simulationen wurden die Verbindungen aller Teilnehmer genutzt, die länger als 10 Tage an der Studie teilgenommen und sich in dieser Zeit mit öffentlichen Netzwerken verbunden haben. Teilnehmer die kürzer als 10 Tage an der Studie teilgenommen und Daten geliefert haben sind für die folgenden Betrachtungen nicht sinnvoll, da sich das Erkennungssystem durch einen initialen Lernprozess und eine fortlaufende Anpassung erst stabilisieren muss, bevor die tatsächliche Effizienz des Systems beobachtet werden kann. Es wurden nur Verbindungen zu öffentlichen, unverschlüsselten Netzwerken betrachtet, da diese das primäre Ziel für einen allgemeinen, nicht zielgerichteten Angriff darstellen würden.
Es sind zwar ebenso Angriffe auf WPA/WPA2-geschützte Netzwerke denkbar, diese werden im Rahmen dieser Dissertation aber nicht betrachtet. Der Grund hierfür liegt darin begründet, dass großflächige Angriffe, die nicht zielgerichtet auf bestimmte Personen durchgeführt werden durch das System verhindert werden sollen. In Netzwerken, die mit den Verschlüsselungstechniken WPA und WPA2 gesichert sind, findet neben einer Authentifizierung des Benutzers ebenso eine Authentifizierung des Access Points statt. Auf diese Weise sind Evil Twin Angriffe in diesen Umgebungen ohne die Nutzung weiterer Techniken wie Social Engineering nicht ohne weiteres durchzuführen. Im Rahmen dieser Arbeit werden entsprechend nur Angriffe auf die weit verbreiteten und durch große Anbieter zur Verfügung gestellten öffentlichen und unverschlüsselten Hotspots betrachtet. Die zuvor beschriebenen Techniken und Parameter des Erkennungssystems sind auch für die hier durchgeführten Simulationen übernommen worden. Weitere Details zur Konfiguration des Simulators und des Erkennungssystems werden in Anhang D dargestellt und erläutert.

9.3.1 Bewertung aus Sicht des Gesamtsystems

Die Ergebnisse einer Simulation über die Gesamtlaufzeit von 74 Tagen ist in Abbildung 9.2 dargestellt. Hier werden die Anzahl an Warnungen pro Tag und Benutzer gezeigt. Die hierbei erzeugten Gruppen reichen von den Benutzern,

die keine (dunkelgrün). bzw. lediglich eine Warnmeldung an einem spezifischen Tag erhalten hätten (grün) bis hin zu Benutzern, die an einem einzigen Tag auf mehr als 10 Warnungen hätten reagieren müssen (dunkelrot).

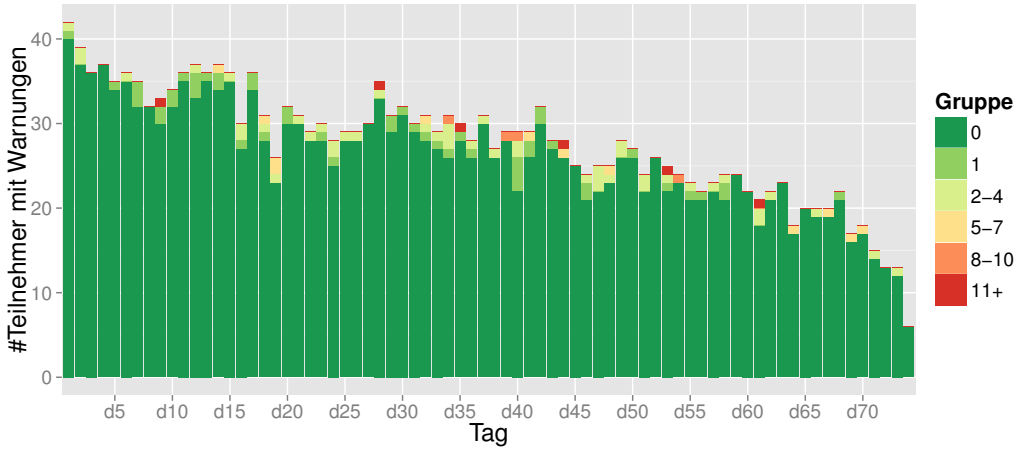


Abbildung 9.2: Anzahl der Warnungen, die einem Benutzer pro Tag angezeigt werden. Die Benutzer sind gruppiert nach Anzahl der erhaltenen Warnungen pro Tag.

Man kann leicht erkennen, dass die Anzahl der Benutzer, die an einem spezifischen Tage nicht eine einzige Warnmeldung erhalten hätten massiv überwiegt. Sie hätten das zugrunde liegende und im Hintergrund operierende Angriffserkennungssystem gar nicht wahrgenommen. Um diejenigen Benutzer besser studieren zu können, die wenigstens eine Warnmeldung angezeigt bekommen würden, wurde die Gruppe derjenigen Benutzer, die keine Warnmeldung erhalten hätten, in Abbildung 9.3 ausgelassen.
Auf diese Weise ergibt sich ein klareres Bild der verbliebenen Benutzer hinsichtlich der Beeinträchtigung im Alltag. Alle in Abschnitt 8.1 beschriebenen Warnmeldungen sind in den hier dargestellten Diagrammen enthalten. Die Kombination aus Abbildung 9.2 und 9.3 macht deutlich, dass im Durchschnitt über alle betrachteten Tage nur 3,77% der Benutzer, die öffentliche Hotspots in ihrem Alltag nutzen, eine Warnmeldung erhalten hätten. Alle verbleibenden Benutzer hätten das im Hintergrund laufende Erkennungssystem am jeweiligen Tag nicht wahrgenommen. Betrachtet man hingegen die Benutzer, die von Warnungen betroffen sind, so erscheint die Anzahl der Warnungsmeldungen zunächst im Einzelfall hoch. Betrachtet man beispielsweise in Abbildung 9.3

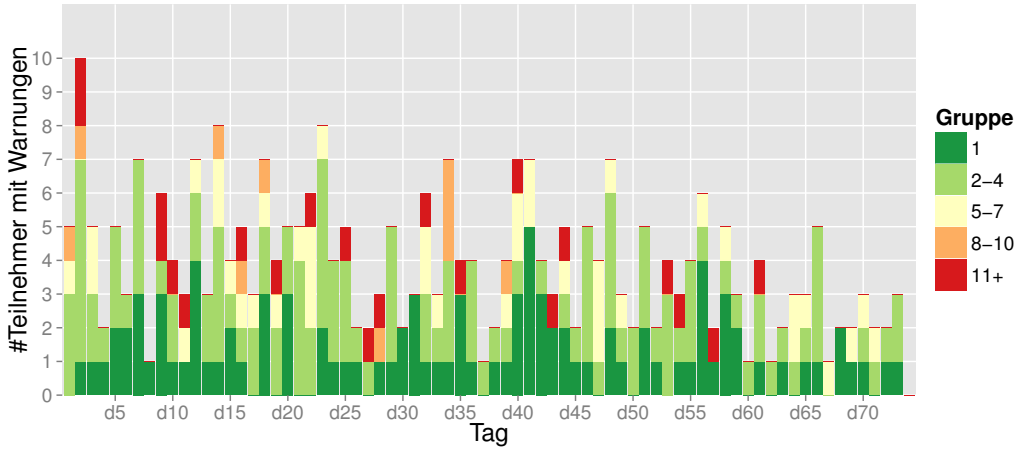


Abbildung 9.3: Anzahl der Warnungen, die ein Benutzer am Tag sieht. Die Benutzer sind gruppiert nach der Anzahl der erhaltenen Warnungen pro Tag. Die Gruppe derjenigen Benutzer, die keine Warnung erhalten hätten, sind in dieser Darstellung nicht mit aufgenommen worden, um die verbleibenden deutlicher darstellen zu können.

den zweiten Tag der Studie, so hätten bei realer Nutzung des Erkennungssystem an diesem Tag 10 der Teilnehmer Warnmeldungen erhalten. Zwei von ihnen hätten an diesem Tag sogar mehr als 10 Warnmeldungen erhalten. Ein Benutzer hätte an diesem Tag auf 8 bis 10 Warnungen reagieren müssen. Die verbleibenden sieben Benutzer hätten 4 Warnmeldungen oder weniger beachten müssen. Diese verhältnismäßig hohe Anzahl lässt sich insbesondere in den ersten Tagen der Nutzung des METDS durch Warnmeldungen bzgl. unbekannter BSSIDs erklären. Diese Art von Warnungen zu unbekannten BSSIDs haben eine Falsch-Positiv-Rate von 0%. Das heißt, dass jede Warnung, die auf dem Gerät eines Benutzers angezeigt wird auch wirklich einen bisher unbekannten Access Point zu einem bereits bekannten Netzwerk darstellt. Diese Warnungen sind somit nicht nur ein Nebeneffekt des Erkennungssystems, sondern ein wichtiger Aspekt für die Sicherheit in nicht vertrauenswürdigen Netzwerken. Verbindungen zu unbekannten Access Points sollten nicht ohne das Wissen des Benutzers eingegangen werden. Aus diesem Grund sind die oben beschriebenen Warnungen zu tolerieren. Sie stellen einen echten Mehrwert hinsichtlich der Sicherheit jedes einzelnen Benutzers dar.

In den vorangegangenen Abbildungen sind die entsprechenden Warnungen zu unbekannten BSSIDs enthalten. Um genauer analysieren zu können, wie groß der Einfluss dieser Meldungen auf die Gesamtzahl der Meldungen ist, werden in Abbildung 9.4 die Warnungen zu unbekannten BSSIDs nicht dargestellt. Alle verbleibenden Warnungen sind weiterhin enthalten. Da für diese Simulationen davon ausgegangen wird, dass während der Aufzeichnung dieser Verbindungen keinerlei Angriffe auf die Teilnehmer stattgefunden haben, können die in diesem Diagramm enthaltenen Warnungen als falsch-positive Warnungen angesehen werden.

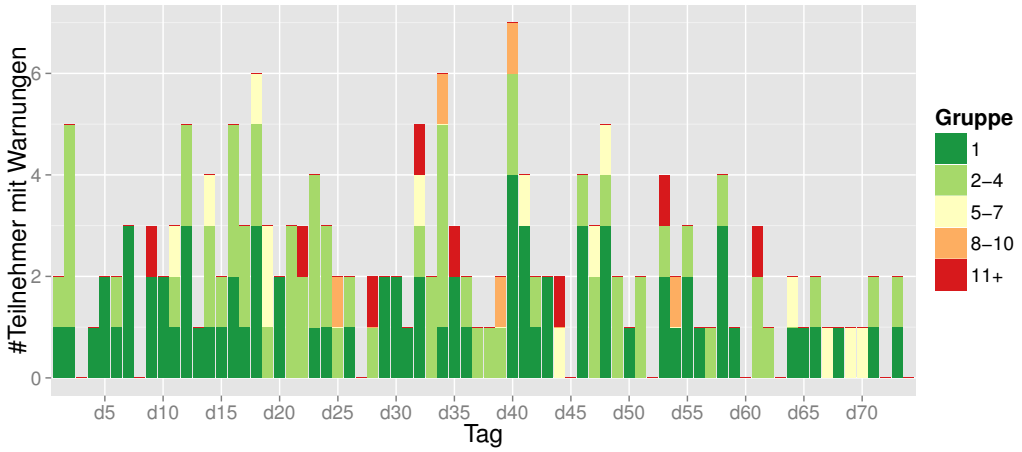


Abbildung 9.4: Anzahl der Warnungen, die einem Benutzer pro Tag angezeigt werden. Die Benutzer sind gruppiert nach Anzahl der erhaltenen Warnungen pro Tag. Warnungen zu unbekannten Access Points sind in dieser Grafik nicht enthalten.

Im Gegensatz zur eindeutigen Abgrenzung zwischen bekannten und unbekannten Access Points ergibt sich bei der Entscheidung der hier aufgeführten Warnmeldungen ein komplexeres Zusammenspiel. Die durch die verschiedenen Sensoren aufgezeichnete Umgebung hat in diesen Fällen einen derart großen Unterschied zur originalen, gespeicherten Umgebung aufgewiesen, dass der Algorithmus diese nicht mehr als bekannt angesehen und eine entsprechende Warnung abgesetzt hat.
Mit Hilfe des Simulators wurde nicht nur erforscht, welche Kontextparameter für eine Entscheidung sinnvoll herangezogen werden können. Ebenso wurden Schwellwerte für Parameter und andere Konfigurationen getestet, eingestellt

und überprüft. Ziel all dieser Simulationen war es, eine Einstellung all dieser Parameter zu finden, die die Erkennung von Angriffen der Typen B, C und D ermöglicht, aber gleichzeitig eine möglichst geringe Falsch-Positiv-Rate hat. Angriffe der Typen B, C und D sind bislang nicht objektiv in der realen Welt beobachtet worden. Sollten Smartphones in Zukunft mit Sicherheitssysteme (wie dem hier entwickelten) ausgestattet sein, so wären diese Angriffstypen hingegen eine logische Konsequenz eines Angreifers. Dieser wird immer den Weg des geringsten Widerstands gehen, um so einen Angriff zu ermöglichen. Aus diesen Gründen ist die im Rahmen der Simulationen gefundene Abstimmung der Konfiguration, Parameter und Schwellwerte eine erste Abschätzung, die dazu führen soll, aktuelle und zukünftige Angriffe so weit zu erschweren wie möglich. Zeitgleich wurde das Ziel der Benutzbarkeit im Auge behalten, um sowohl ein angenehmes Arbeiten mit dem System, als auch eine hohe Akzeptanz zu gewährleisten. Eine Anpassung verschiedener Parameter in den Systemen der Endanwender ist nicht nur denkbar sondern auch wünschenswert, um den diversen Anforderungen und Bedürfnissen der einzelnen Benutzer gerecht zu werden.

Um noch einmal herauszustellen, wie wenige Warnmeldungen die Benutzer in einem realen System auf ihrem Smartphone akzeptieren müssten, wurden in einer weiteren Untersuchung nur diejenigen Benutzer betrachtet, die an einem spezifischen Tag Daten zur Studie beigetragen haben. Es handelt sich also bei der folgenden Betrachtung ausschließlich um Benutzer die sowohl am jeweiligen Tag aktiv ihr Smartphone genutzt haben und die sich im Laufe der Studie mit öffentlichen WLANs verbunden haben. Basierend auf diesen Einschränkungen ergeben sich äußerst akzeptable Zahlen hinsichtlich der Benutzbarkeit und der Akzeptanz des Systems. Als Tagesdurchschnitt wurden nur für lediglich 5,81% der aktiven Teilnehmer Warnungen ausgegeben. Die verbleibenden 94,19% der haben im Durchschnitt keine Warnungen an einem spezifischen Tag erhalten. Dies kann einerseits damit zusammenhängen, dass an diesem Tag keine Verbindungen mit öffentlichen Netzwerken aufgebaut wurden, oder aber auch dass während des Verbindungsaufbaus die Umgebung durch das METDS als sicher eingestuft wurde und auf eine Warnung verzichtet wurde.

9.3.2 Bewertung aus Nutzersicht

Um eine Bewertung des Erkennungs- und Warnungssystems aus der Sicht eines einzelnen Nutzers durchzuführen werden zunächst wieder alle Benutzer der Studie betrachtet. In der folgenden Abbildung 9.5 sind die Anzahlen der Warnmeldungen pro Benutzer dargestellt. Weil die Anzahl der potentiellen Warnmeldungen von der Gesamtanzahl der Verbindungen eines Benutzers abhängig ist, wurde diese Größe in den Betrachtungen eliminiert, in dem die Anzahl der Warnungen pro 100 Verbindungen des jeweiligen Benutzers dargestellt wird.

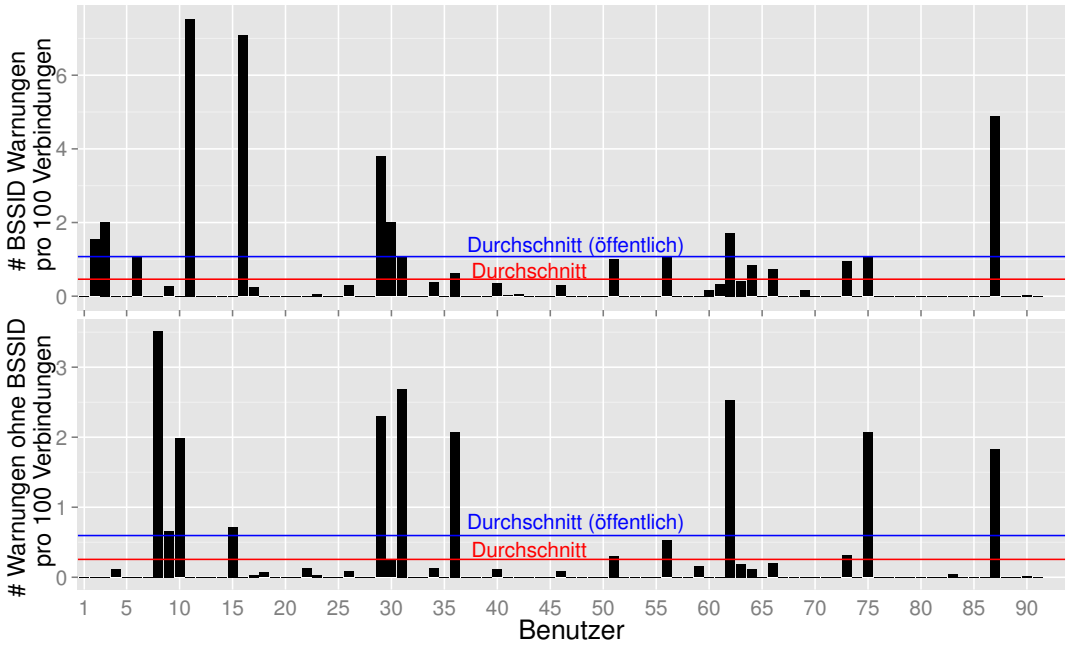


Abbildung 9.5: Warnungsmeldungen pro Benutzer, aufgeteilt nach unbekannten BSSIDs und sonstigen Warnmeldungen

Im oberen Bereich des Diagramms ist die relative Anzahl an BSSID-Warnungen pro Teilnehmer der Studie abgebildet. Im unteren Bereich sind alle anderen Warnungstypen in gleicher Weise dargestellt. Aus dieser Darstellung können verschiedene Erkenntnisse gezogen werden. Zum einen kann wie erwartet festgestellt werden, dass die Anzahl an Warnungen bezüglich unbekannter BSSIDs gegenüber den restlichen Warnungstypen überwiegen. Allgemein lässt sich (bis auf wenige Ausnahmen) ebenso ein niedriges Niveau bzgl. der Anzahl an Warnmeldungen feststellen.

Typ	Maximum	⌀ alle	⌀ öffentliche
BSSID	7,53	0,46	1,07
andere	3,52	0,26	0,60

Tabelle 9.1: Durchschnittliche Warnmeldungen pro 100 Verbindungen

In Tabelle 9.1 sind die entsprechenden Werte notiert. Selbst die beiden Maxima der verschiedenen Warnungstypgruppen liegen mit 7,53 und 3,52 Warnmeldungen pro 100 Verbindungen noch in einem für den Endnutzer tolerierbaren Rahmen. Die Durchschnittswerte über alle Teilnehmer dieser Warnungshäufigkeit liegen mit 0,46 BSSID-Warnungsmeldungen und 0,26 sonstigen pro 100 Verbindungen in einem äußerst niedrigen Bereich. Selbst wenn ausschließlich diejenigen Benutzer für den Durchschnitt herangezogen werden, die im Laufe der Studie öffentliche Netzwerke und Hotspots genutzt haben, so sind die Durchschnittswerte mit 1,07 BSSID- und 0,6 sonstigen Warnmeldungen immer noch in einem für den Endnutzer akzeptablen Bereich.

9.4 Zeitmessungen

Ein Erkennungssystem, welches auf mobilen Geräten als Hintergrundprozess betrieben wird und bei jedem Verbindungsaufbau Verbindungsparameter überprüft, muss sowohl schnell als auch energieeffizient arbeiten. Nur ein schnell arbeitendes System beeinträchtigt den Benutzer im normalen Umgang mit seinem Smartphone nicht und ermöglicht ihm das unterbrechungs- und störungsfreie Arbeiten. Aus diesem Grund wurde eine Zeitmessung des METDS durchgeführt. Es wurde wiederholt die Zeit gemessen, die den gesamten METDS-Durchlauf umfasst. Gestartet wurde die Messung jeweils zum Zeitpunkt, wenn METDS die initial aufgebaute Verbindung untersucht und im Anschluss trennt. Gestoppt wurde die Messung jeweils zum Zeitpunkt, wenn durch das METDS ein Ergebnis ermittelt wurde, welches nach Beendigung der Messung dem Benutzer in Form einer Benachrichtigung angezeigt wurde. Durchgeführt wurden alle Messungen mit einem LG Nexus 4 mit der Android Version 5.0.1. Zur Messung wurde ein Timer entwickelt, welcher zu den oben beschriebenen Zeitpunkten die aktuelle Systemzeit in Millisekunden speichert und aus der

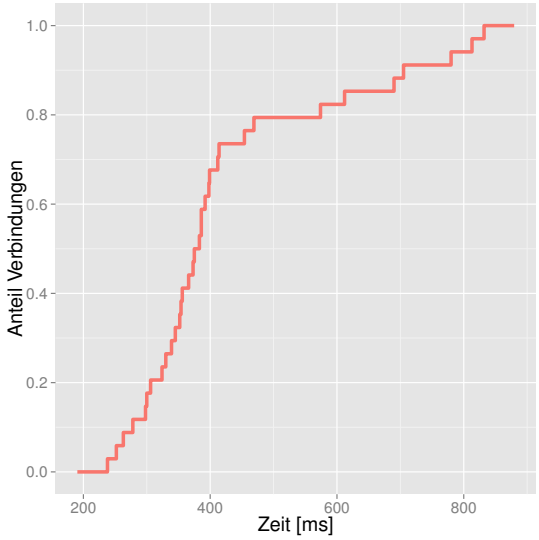


Abbildung 9.6: CDF-Diagramm zur Verzögerung des Verbindungsaufbaus durch die Verifikation der Verbindung des METDS

Differenz der ermittelten Werte die benötigte Zeit für den Vorgang ermittelt hat.
Abbildung 9.6 zeigt in einem CDF-Diagramm das Laufzeitverhalten des Erkennungssystems. Es zeigt, mit welcher Wahrscheinlichkeit welche zusätzliche Zeit pro Verbindungsaufbau zu erwarten ist. Man sieht, dass das System in keinem Fall länger als 868 ms benötigt hat, um eine Verbindung zu evaluieren. In 80 % der Fälle konnte die Evaluation bereits in unter 600 ms abgeschlossen werden. Die entstehenden Wartezeiten sind allesamt unter einer Sekunde und dürften vom Benutzer während des Verbindungsaufbaus kaum wahrgenommen werden können. Auch wenn diese Zeit im Fall von technisch bedingten Verbindungsabbrüchen häufiger benötigt wird, steht sie in keinem Verhältnis zum Zugewinn an Sicherheit durch das System.

9.5 Zusammenfassung

Der im Rahmen dieses Kapitels vorgestellte Simulator dient dazu den Algorithmus des entstandenen Erkennungssystems zu entwickeln und die Parameter zu untersuchen und festzulegen. Es wurde eine Bewertung des Systems sowohl aus Sicht des Systems, als auch aus Nutzersicht vorgestellt. Hierbei wurde ausschließlich auf Echtweltdaten zurückgegriffen, die im Rahmen der beschrie-

benen Feldstudie gesammelt werden konnten. Als Ergebnis lässt sich für das Gesamtsystem festhalten, dass unter Verwendung der gesammelten Daten im Schnitt nur 5 % der Teilnehmer an einem speziellen Tag eine Warnung erhalten hätten. Alle anderen Teilnehmer hätten das Erkennungsystem an diesem speziellen Tag nicht bemerkt. Aber auch aus Nutzersicht ergibt sich, dass die Anzahl der Warnungsmeldungen so niedrig ist, dass einem Alltagseinsatz des Erkennungssystem von diesem Standpunkt aus nichts entgegen spricht.
Um ebenfalls zu evaluieren, welchen Einfluss ein derartiges Erkennungssystem im Alltag der Smartphone-Nutzung hat, wurden Zeitmessungen des Erkennungssystems durchgeführt. Hierbei wurde jeweils die Zeit gemessen, die das System braucht, um bei einem Verbindungsversuch über die jeweilige Situation zu entscheiden. Aus den Messungen ergab sich, dass das System in keinem Fall länger als eine Sekunde und in 80% der Fälle sogar in unter 600 ms eine Entscheidung treffen konnte. Diese Zeiten sind für den Benutzer bei normaler Nutzung an dieser Stelle nicht wahrnehmbar.

Kapitel 10

Zusammenfassung und Ausblick

In diesem letzten Teil der Dissertation sollen noch einmal alle wichtigen Arbeiten und ihre Ergebnisse kurz zusammengefasst werden. Es werden darüber hinaus aus den Ergebnissen gewonnene Empfehlungen gegeben, die für die Sicherheit zukünftiger mobiler Betriebssysteme und Hotspot-Infrastrukturen förderlich sind. Ebenso wird ein Ausblick auf zukünftige Möglichkeiten mit Hilfe des entstandenen Erkennungssystems gegeben und aufgezeigt, dass auch die weitere Aufklärung der Benutzer über derartige Gefahren zu einer Verbesserung der derzeitigen Situation erforderlich ist.

10.1 Zusammenfassung

In der vorliegenden Arbeit wurde gezeigt, dass ein Schutz gegen Evil Twin Access Points in heutigen Hotspot-Infrastrukturen unbedingt erforderlich ist. Ein entsprechendes System wurde für das mobile Betriebssystem Android entwickelt und seine Funktionsfähigkeit unter Beweis gestellt.

Im ersten Teil dieser Arbeit wurden zunächst die Gefahren heutiger Hotspot-Umgebungen erforscht. Im Anschluss wurde eine neuartige un im Rahmen dieser Dissertation entstandene mobile Malware vorgestellt, die Unzulänglichkeiten moderner mobiler Betriebssysteme zur Verbreitung ausnutzt. Sie erweitert die bereits bekannten Evil Twin-Angriffe lokaler, fest installierter Access Points um eine mobile Komponente und sorgt dafür, dass auch Mobiltelefone zu Evil Twin-Hotspots werden können. Im Rahmen von Simulationen konnte gezeigt werden, dass die Ausbreitung einer solchen mobilen Malware in urbanen Gebieten nicht nur sehr schnell stattfindet sondern auch, dass in sehr kurzer Zeit ein erheblicher Teil der potentiell gefährdeten Gesamtbevölkerung davon betroffen ist. Es wurden zahlreiche Simulationen mit verschiedensten Startbedingungen und Verbreitungsoptionen durchgeführt, die alle zum oben genannten Ergebnis führten.

Diese Beobachtungen und das Fehlen adäquater Sicherheitssysteme in aktuellen mobilen Betriebssysteme haben die Entwicklung eines solchen Sicherheitssystems im Rahmen dieser Dissertation vorangetrieben. Ziel des Schutzsystems ist es, Smartphone-Nutzer vor allgemeinen Evil Twin-Angriffen zu schützen. Um dies erreichen zu können wurde zunächst untersucht, mit Hilfe welcher zur Verfügung stehenden Informationen eine Aussage über das Vorhandensein eines Evil Twin Access Points getroffen werden kann. Ebenso wurde der Einsatz eines verteilten Systems mit zentraler Datensammlung und -auswertung diskutiert und zugunsten eines leichtgewichtigen autonomen Systems verworfen, das für die Verrichtung seiner Aufgabe keinerlei Informationen von Drittsystemen benötigt. Das entstandene autonome System nutzt hierbei ausschließlich Daten, die durch lokale Sensoren gesammelt werden. Ebenso ist das Erkennungssystem in der Lage, sich an verändernde Situationen und Umgebungen anzupassen.

Im Rahmen einer Studie wurde die Annahme bestätigt, dass Smartphone-Nutzer in aktuellen WLAN-Umgebungen nicht nur gefährdet sind, sondern auch, dass sie sich dieser Gefahr in vielen Fällen nicht bewusst sind. So konnte im Rahmen der Studie nachgewiesen werden, dass in 78 % der Fälle Verbindungen zu Access Points durch das Smartphone initiiert wurden. Diese Verbindungen wurden ohne jegliche Nutzerinteraktion und entsprechend ohne das Wissen des Benutzers aufgebaut. Zeitgleich wurden in der Studie Daten realer Verbindungen von Benutzern zu WLANs gesammelt und konnten zur Konfiguration des Erkennungssystems genutzt werden. Auf diese Weise konnte das System dahingehend angepasst werden, dass Evil Twin Access Points einerseits zuverlässig erkannt werden, andererseits aber die Benutzbarkeit des Systems nicht unter einer Flut falscher Warnmeldungen leidet.
Das entstandene Erkennungssystem wurde in einem weiteren Schritt auf die Android-Plattform portiert. In Form einer App bietet es dem Benutzer den oben beschriebenen Schutz und ist leichtgewichtig und flexibel genug, um auf zukünftige, neue Bedrohungen hin angepasst werden zu können. Ebenso arbeitet das System schnell genug, um im Alltag nicht als störende Verzögerung wahrgenommen zu werden.
Die Evaluation des Erkennungssystems mit Hilfe eines Simulators wurde ebenfalls auf den im Rahmen der Studie aufgezeichneten Echtweltdaten durchgeführt. Hierbei hat sich für das Gesamtsystem ergeben, dass im Schnitt lediglich 5 % der Benutzer Warnungen an einem speziellen Tag erhalten hätten. Der Großteil der Nutzer wäre entsprechend überhaupt nicht durch das System eingeschränkt worden. Auch aus Nutzersicht haben sich diese Zahlen bestätigt, womit einem Einsatz im Alltag auch von diesem Standpunkt her nichts entgegen spricht.

10.2 Empfehlungen zur Steigerung der Sicherheit in Hotspot-Umgebungen

Eine der größten Gefahrenquellen, die im Rahmen der Dissertation identifiziert wurde ist die Eigenschaft aller mobilen Betriebssystem, sich automatisch mit bekannten Netzwerken zu verbinden. Ob ein Netzwerk als bekannt angesehen wird bestimmt sich alleine aus dem Namen des Netzwerks (der SSID). Die

größte Gefahr stellt in diesem Zusammenhang der Aspekt dar, dass dies auch für unverschlüsselte Netzwerke gilt. Eine der größten Gefahrenquellen könnte entsprechend dadurch vermieden werden, dass die automatische Verbindung mit unverschlüsselten und somit offensichtlich unsicheren Netzwerken nicht mehr stattfindet. In Bezug auf private Netzwerke hätte diese Einschränkung in der heutigen Zeit keine nennenswerten Auswirkungen, sind doch nahezu alle privaten Netzwerke mittlerweile per WPA oder WPA2 abgesichert. Auch der verbreitete Einsatz von Sicherheitssystemen, wie dem hier vorgestellten, kann die allgemeine Sicherheit in solchen Infrastrukturen erhöhen.
Betrachtet man hingegen nicht die Nutzerseite, sondern widmet sich den Anbietern der Infrastrukturen, so sind auch hier Verbesserungen möglich. Um die Erkennung von Evil Twins zu ermöglichen reicht es in den meisten Fällen aus, ein WLAN Intrusion Detection System zu installieren. Dieses beobachtet das eigene Netzwerk und kann nicht autorisierte Access Points schnell und zuverlässig erkennen und melden. Sowohl die Einrichtung eines solchen Systems als auch der Betrieb verursacht Kosten. Aus diesen Gründen ist es an den Anbietern, sich mittel- und langfristig über den Aufbau und Betrieb von AAA-Infrastrukturen in Verbindung mit WPA2-gesicherten WLANs Gedanken zu machen.
In Anbetracht der Ergebnisse der durchgeführten Feldstudie ist ein weiterer wesentlicher Aspekt bei der Verbesserung der Sicherheit die weitere, gründliche Aufklärung der Benutzer hinsichtlich der Gefahren. Die Kombination aus gesammelten Echtweltdaten und den Antworten der Befragung hat große Unterschiede zwischen dem vermuteten und dem realen Verhalten aufgezeigt. Nahezu alle Teilnehmer der Studie waren sich der Gefahr in keinster Weise bewusst. Aus diesen Gründen ist die weitere Aufklärung der Benutzer ein ebenso wichtiger Ansatzpunkt wie die Verbesserung der Sicherheitssysteme auf Nutzer- und Betreiberseite.

10.3 Ausblick

Das in dieser Dissertation entstandene und evaluierte, prototypische Erkennungssystem für Evil Twin Access Points kann bei der zukünftigen Entwicklung und weiteren Erforschung mobiler Sicherheitssysteme als Basis genutzt

werden. Die Anzahl an Hotspots großer Anbieter und der durch Städte und Kommunen angebotenen frei zugänglichen Funknetzwerke nimmt stetig zu. In Kombination mit dem ebenfalls steigenden Datenverbrauch mobiler Anwendungen ergibt sich ein stark anwachsendes Gefahrenpotential durch die beschriebenen Angriffe.

Das in dieser Dissertation entstandene Erkennungssystem ist unabhängig von anderen Systemen und zusätzlicher Infrastruktur. Es verwendet ausschließlich lokal verfügbare Sensoren und kann im Rahmen zukünftiger Forschung um zusätzliche Sensorik erweitert werden. Ebenso ist der Algorithmus in zukünftigen Versionen an neue Anforderungen und Gegebenheiten anpassbar. Eine für die zukünftige Forschung interessante Erweiterungsmöglichkeit wäre zum einen das anonyme Logging. Zum anderen ist die Erweiterung des Systems zur Erkennung weiterer Angriffe denkbar. Durch eine Erweiterung um eine Logging-Komponente könnten in Zukunft Daten über erkannte Angriffe anonym gesammelt werden, ohne hierbei die Privatsphäre des einzelnen Benutzers zu gefährden. Mit Hilfe der auf diese Weise gesammelten Daten könnten langfristige Trends hinsichtlich der Gefahrenlage erkannt und erforscht werden. Aber auch die Erkennung weiterer Angriffstypen auf mobile Geräte ist denkbar. Hierfür ist zukünftig zu erforschen, welche Angriffe mit Hilfe welcher Datenquellen für diese Erkennung in Frage kommen und wie sich die Analysen und Auswertungen hinsichtlich eines autarken, mobilen Systems eignen.

Das Ziel wird es in Zukunft sein, den Endnutzern Schutzsysteme wie das hier entstandene zur Verfügung zu stellen. Hierfür gibt es verschiedene Möglichkeiten. Die für den Endnutzer einfachste Möglichkeit wäre die Integration des Erkennungssystems in die mobilen Betriebssysteme. Auf diese Weise könnten sie den Benutzer bestmöglich vor derartigen Angriffen schützen. Betrachtet man die Bemühungen der Anbieter der mobilen Betriebssysteme hinsichtlich dieser Problematik in den vergangenen Jahren, so ist an dieser Stelle nicht mit einer baldigen Integration zu rechnen. Die andere Möglichkeit ist die Verbreitung des Erkennungssystems in Form einer App mit den beschriebenen Einschränkungen. Hierfür muss die prototypische Implementierung der App weiterentwickelt werden. Ebenso muss auf Seiten der Endnutzer ein aktives Interesse an derartigen Sicherheitssystemen erzeugt werden, weil sie durch den Nutzer aktiv eingerichtet werden müssen. Auch hierfür ist die erwähnte Auf-

klärung der Benutzer hinsichtlich der Gefahren in nicht vertrauenswürdigen Netzwerken voranzutreiben.

Anhang A

iOS Jailbreaks

A.1 iOS-Version mit verfügbaren Jailbreaks

Die folgende Tabelle A.1 zeigt, welche Versionen von iOS Jailbreak-Werkzeuge existieren. Bis auf sehr wenige Ausnahmen sind diese Werkzeuge zum Entsperren für jede Version verfügbar gewesen und sind es bis heute. Die Ausnahmen sind vorwiegend zu Beginn der Smartphone-Ära zu erkennen. Ebenso wurden für kleine Versionssprünge, welche sich zeitlich kurz vor einem großen Update befanden keine Jailbreaks veröffentlicht, um Apple die Chance zu nehmen, die für den Jailbreak verwendeten Schwachstellen in letzter Sekunde zu schließen. Teile der hier dargestellten Tabelle und weitere Informationen stammen von einer durch die Jailbreak-Community betriebenen und gepflegten Internetseite *theiPhoneWiki* [68]. Für iOS 8.3, also die zum Zeitpunkt der Abgabe dieser Dissertation aktuelle Version, ist zu diesem Zeitpunkt noch kein Jailbreak verfügbar.

	iPhone Gerätegeneration										
iOS Version	**2G**	**3G**	**3GS**	**4** GSM	**4** CDMA	**4S**	**5**	**5C**	**5S**	**6**	**6** Plus
1.0	✗										
1.0.0	✗										
1.0.1	✗										
1.0.2	✓										
1.1	✗										
			⋮				⋮				

iOS Version	iPhone Gerätegeneration										
	2G	**3G**	**3GS**	**4** GSM	**4** CDMA	**4S**	**5**	**5C**	**5S**	**6**	**6** Plus
1.1.1	✓										
1.1.2	✓										
1.1.3	✓										
1.1.4	✓										
1.1.5	✗										
2.0	✓	✓									
2.0.1	✓	✓									
2.0.2	✓	✓									
2.1	✓	✓									
2.1.1	✓	✗									
2.2	✓	✓									
2.2.1	✓	✓									
3.0	✓	✓	✓								
3.0.1	✓	✓	✓								
3.1	✓	✓	✓								
3.1.2	✓	✓	✓								
3.1.3	✓	✓	✓								
4.0		✓	✓	✓							
4.0.1		✓	✓	✓							
4.0.2		✓	✓	✓							
4.1		✓	✓	✓							
4.2.1		✓	✓	✓							
4.2.6					✓						
4.2.7					✓						
4.2.8					✓						
4.2.9					✓						
4.2.10					✓						
4.3			✓	✓	✗						
4.3.1			✓	✓	✗						
4.3.2			✓	✓	✗						
4.3.3			✓	✓	✗						
4.3.4			✓	✓	✗						
4.3.5			✓	✓	✗						
5.0			✓	✓	✓	✓					
5.0.1			✓	✓	✓	✓					
			⋮				⋮				

iOS Version	iPhone Gerätegeneration										
	2G	**3G**	**3GS**	**4** GSM	**4** CDMA	**4S**	**5**	**5C**	**5S**	**6**	**6** Plus
5.1			✓	✓	✓	✗					
5.1.1			✓	✓	✓	✓					
6.0			✓	✓	✓	✓	✓				
6.0.1			✓	✓	✓	✓	✓				
6.0.2							✓				
6.1			✓	✓	✓	✓	✓				
6.1.1						✓					
6.1.2			✓	✓	✓	✓	✓				
6.1.3			✓	✓	✓	✓	✓				
6.1.4							✓				
7.0				✓	✓	✓	✓	✓	✓		
7.0.1								✓	✓		
7.0.2				✓	✓	✓	✓	✓	✓		
7.0.3				✓	✓	✓	✓	✓	✓		
7.0.4				✓	✓	✓	✓	✓	✓		
7.0.5								✓	✓		
7.1				✓	✓	✓	✓	✓	✓		
7.1.1				✓	✓	✓	✓	✓	✓		
7.1.2				✓	✓	✓	✓	✓	✓		
8.0						✓	✓	✓	✓	✓	✓
8.0.1						✓	✓	✓	✓	✓	✓
8.0.2						✓	✓	✓	✓	✓	✓
8.1						✓	✓	✓	✓	✓	✓
8.1.1						✓	✓	✓	✓	✓	✓
8.1.2						✓	✓	✓	✓	✓	✓
8.1.3						✗	✗	✗	✗	✗	✗
8.2						✗	✗	✗	✗	✗	✗
8.3						✗	✗	✗	✗	✗	✗

Tabelle A.1: Alle bisher erschienen iOS Versionen und die Verfügbarkeit eines Jailbreaks (Stand 04/2015)

A.2 Werkzeuge

Die folgende Tabelle A.1 gibt eine Übersicht über alle bis heute veröffentlichten iOS Versionen für das iPhone. Nicht betrachtet werden iOS Versionen die ausschließlich für andere Geräte als das iPhone veröffentlicht wurden. Zusätzlich ist das entsprechende Tool dargestellt, welches für einen Jailbreak dieser iOS Version veröffentlicht wurde.

iOS Version	Jailbreak Tool
Version	JailbreakTool
1.0	AppTappInstaller,iBrickr
1.0.x	AppTappInstaller,iBrickr
1.1.1	AppSnapp
1.1.2	OktoPrep
1.1.3	iLiberty/iLiberty+,SoftUpgrade,ZiPhone
1.1.4	iLiberty/iLiberty+,PwnageTool,ZiPhone
2.0	PwnageTool,QuickPwn
2.0.x	PwnageTool,QuickPwn
2.1	PwnageTool,QuickPwn
2.1.1	PwnageTool
2.2	PwnageTool,QuickPwn
2.2.1	PwnageTool,QuickPwn
3.0	purplera1n,PwnageTool,redsn0w
3.0.1	redsn0w
3.1	blackra1n,PwnageTool,redsn0w
3.1.2	blackra1n,PwnageTool,redsn0w,Star,sn0wbreeze,Spirit
3.1.3	PwnageTool,redsn0w,Star,sn0wbreeze,Spirit
4.0	limera1n,PwnageTool,redsn0w,sn0wbreeze,Star
4.0.1	limera1n,PwnageTool,sn0wbreeze,redsn0w,Star
4.0.2	limera1n,redsn0w
4.1	limera1n,PwnageTool,redsn0w,sn0wbreeze,greenpois0n,Star
4.2.1	PwnageTool,redsn0w,sn0wbreeze,greenpois0n,Star
4.2.6	greenpois0n,PwnageTool,redsn0w,Saffron,sn0wbreeze,unthredera1n
4.2.7	redsn0w,Saffron,sn0wbreeze,unthredera1n
4.2.8	PwnageTool,redsn0w,Saffron,sn0wbreeze,unthredera1n
4.2.9-10	redsn0w,unthredera1n
⋮	⋮

iOS Version	Jailbreak Tool
4.3	PwnageTool,redsn0w,Saffron,sn0wbreeze,unthredera1n
4.3.1-3	PwnageTool,redsn0w,Saffron,sn0wbreeze,unthredera1n
4.3.4-5	PwnageTool,redsn0w,unthredera1n
5.0	Absinthe,PwnageTool,redsn0w,sn0wbreeze,unthredera1n
5.0.1	Absinthe,redsn0w,sn0wbreeze,unthredera1n,PwnageTool
5.1	redsn0w,sn0wbreeze,unthredera1n
5.1.1	redsn0w,sn0wbreeze,unthredera1n,PwnageTool,Absinthe
6.0	evasi0n,redsn0w,sn0wbreeze
6.0.1	evasi0n,redsn0w,sn0wbreeze
6.1	evasi0n,redsn0w,sn0wbreeze
6.1.1	evasi0n
6.1.2	evasi0n,redsn0w,sn0wbreeze
6.1.3	evasi0n,p0sixspwn,redsn0w,sn0wbreeze
6.1.4	p0sixspwn
7.0	evasi0n7
7.0.1-6	evasi0n7
7.1	Pangu
7.1.1	Pangu
7.1.2	Pangu
8.0-8.1	Pangu8,TaiG
8.1.1	TaiG
8.1.2	TaiG

Tabelle A.2: Alle bisher erschienen iOS Versionen und die zur Verfügung stehenden Jailbreak-Werkzeuge (Stand 04/2015)

Anhang B

Evil Twin-Malware Skripte

Das folgende Skript stellt den Daemon dar, der die Installation und die Ausführung der für die Evil Twin Verbreitung benötigten Software steuert. Er installiert die benötigten Pakete, kopiert Konfigurationen an die vorgesehenen Stellen und startet im Anschluss die erforderliche Software zum Betrieb des Mobile Evil Twins.

```
1  #!/bin/sh
2  name=evilTwinInstall
3  sleep 10
4  if [ -f /tmp/evilTwinInstall.success ]
5  then
6    launchctl remove evilTwinInstall
7  else
8
9    dpkg -i /tmp/openssl_0.9.8k-9_iphoneos-arm.deb
10   dpkg -i /tmp/openssh_5.8p1-9_iphoneos-arm.deb
11
12   dpkg -i /tmp/firmware_4.3.3_iphoneos-arm.deb
13   dpkg -i /tmp/libxml2-lib_2.6.32-3_iphoneos-arm.deb
14   dpkg -i /tmp/libxml2_2.6.32-6_iphoneos-arm.deb
15   dpkg -i /tmp/bzip2_1.0.5-7_iphoneos-arm.deb
16   dpkg -i /tmp/pcre_7.9-3_iphoneos-arm.deb
17   dpkg -i /tmp/sqlite3-lib_3.5.9-2_iphoneos-arm.deb
18   dpkg -i /tmp/sqlite3_3.5.9-12_iphoneos-arm.deb
19
20   dpkg -i /tmp/lighttpd_1.4.18-6_iphoneos-arm.deb
21   dpkg -i --force-all /tmp/com.saurik.substrate.safemode_0.9.3900_iphoneos-
         arm.deb
22   dpkg -i /tmp/mobilesubstrate_0.9.3901_iphoneos-arm.deb
23   dpkg -i /tmp/com.intelli.statusbaricons_0.95_iphoneos-arm.deb
24   dpkg -i --force-all /tmp/com.mywi4.ondemand_4.50.6_iphoneos-arm.deb
25   dpkg -i /tmp/com.mywi4_5.03.2_iphoneos-arm.deb
```

```
26
27   mv /tmp/www /var/
28   mv /tmp/com.mywi.plist /private/var/mobile/Library/Preferences/com.mywi.
         plist
29   mv /tmp/lighttpd.conf /etc/lighttpd.conf
30   mv /tmp/mypfrules.conf /etc/mypfrules.conf
31
32   launchctl submit -l webserver -- /usr/sbin/lighttpd -f /etc/lighttpd.conf
33   /Applications/MyWi.app/MyWiApp_ startmywi
34   pfctl -F all
35   pfctl -f /etc/mypfrules.conf
36   touch /tmp/evilTwinInstall.success
37 fi
```

Listing B.1: `evilTwinInstall` -Daemon zur Installation der benötigten Softwarepakete und weiterer Daten

Anhang C

Studien-Fragebogen

Die folgenden Fragen sind im Rahmen der Studie mit Hilfe des in die Studien-App integrierten Fragebogens durch die Teilnehmer der Studie beantwortet worden. Die Teilnehmer haben die Fragen lokal auf ihren Geräten beantwortet. Im Anschluss an die Befragung wurden die gesammelten Antworten gesichert an einen zentralen Server zur Speicherung der Ergebnisse gesendet. Der Wortlaut der folgenden Fragen entspricht genau den Originalfragen. Die Antworten auf die Fragen sind in den nun folgenden Fragebogen integriert. Sie stehen direkt neben der Antwort. Bei Fragen mit der Eingabemöglichkeit für eine Zahl ist eine kurze Statistik bestehend aus Minimum, Maximum, Durchschnitt und Standardabweichung dargestellt.

Frage 1

Bitte gib Dein Geschlecht an:

- ☐$_{70}$ männlich
- ☐$_{13}$ weiblich
- ☐$_{0}$ keine Angabe

Frage 2

Bitte gib Dein Alter an: _____

min	18
max	56
Ø	28,24
σ	8,37

Frage 3

Welche Tätigkeit hast Du zuletzt ausgeübt oder übst Du zur Zeit aus?

- ☐$_{1}$ Schüler
- ☐$_{46}$ Student
- ☐$_{3}$ Arbeitnehmer (Teilzeit)
- ☐$_{27}$ Arbeitnehmer (Vollzeit)
- ☐$_{2}$ Selbstständig
- ☐$_{0}$ Rentner
- ☐$_{0}$ Hausfrau/-mann
- ☐$_{4}$ Erwerbslos / Arbeitslos

Frage 4

Wie nutzt Du Dein Smartphone vorwiegend?

- ☐$_{60}$ privat
- ☐$_{1}$ beruflich
- ☐$_{22}$ beides

Frage 5

Hast Du jemals eine Ausbildung, einen Studiengang oder einen Beruf im IT-Bereich absolviert bzw. ausgeübt oder tust Du dies derzeit?

- ☐$_{43}$ Ja
- ☐$_{40}$ Nein

Frage 6

Bitte bewerte die folgende Aussage über Dich selbst: Ïch habe ein sehr gutes Verständnis von Computern und dem Internet."

- ☐$_{48}$ Stimme vollständig zu
- ☐$_{28}$ Stimme zu
- ☐$_{7}$ Neutral
- ☐$_{0}$ Stimme nicht zu
- ☐$_{0}$ Stimme gar nicht zu

Frage 7

Bitte bewerte die folgende Aussage über Dich selbst: Ïch frage häufig Andere wenn ich Computerprobleme habe."

- ☐$_{1}$ Stimme vollständig zu
- ☐$_{9}$ Stimme zu
- ☐$_{11}$ Neutral
- ☐$_{42}$ Stimme nicht zu
- ☐$_{20}$ Stimme gar nicht zu

Frage 8

Bitte bewerte die folgende Aussage über Dich selbst: Ändere fragen häufig mich wenn sie Computerprobleme haben."

- ☐$_{38}$ Stimme vollständig zu
- ☐$_{33}$ Stimme zu
- ☐$_{8}$ Neutral
- ☐$_{3}$ Stimme nicht zu
- ☐$_{1}$ Stimme gar nicht zu

Frage 9

Wie oft verbindet sich Deiner Meinung nach Dein Smartphone mit einem WLAN pro Tag? _____

min	0
max	100
Ø	10,88
σ	17,33

Frage 10

Wie häufig nutzt Du öffentliche Hotspots (z.B. bei Starbucks, Mc Donalds / in Bahnhöfen)?

- ☐$_{2}$ täglich
- ☐$_{1}$ mehrmals im Monat
- ☐$_{41}$ selten
- ☐$_{39}$ nie

Anhang D

Konfigurationsdetails des Erkennungssystems

In Tabelle D.1 sind die wichtigsten Konfigurationsobjekte des Erkennungssystems aufgelistet. Sie stellen die für die Simulationen verwendeten beispielhaften Werte dar, die sich im Laufe der Entwicklung und Erprobung als am besten geeignet herausgestellt haben. Im Folgenden soll kurz auf die einzelnen Objekte eingegangen werden.

Zunächst geben die drei Objekte `ACCOUNT_UNENCRYPTED`, `ACCOUNT_WPA_PSK` und `ACCOUNT_WPA_ENTERPRISE` an, welche Art von Netzwerken durch das System geprüft werden sollen. Wie beschrieben untersucht das Erkennungssystem in seiner derzeitigen Version ausschließlich unverschlüsselte Netzwerke. Es sind somit Vorbereitungen getroffen worden, das das Erkennungssystem in zukünftigen Versionen ebenfalls die beiden bislang nicht betrachteten Typen von Netzwerken überwachen kann. Der Wert `BSSID_DELETION_THRESHOLD` gibt an, wie oft ein Access Point in einer Umgebung vermisst werden muss, bis er dauerhaft aus dem Access Point Profil entfernt wird. Dadurch werden sich langsam verändernde Umgebungen berücksichtigt und das System adaptiert sich der neuen Situation. Ebenso verhält es sich mit dem Wert `BSSID_ADDITION_THRESHOLD`. Dieser gibt an, wie oft ein bislang unbekannter Access Point in einer bekannten Umgebung vorgefunden werden muss, bis er als fest zu dieser Umgebung zugehörig angesehen wird. Erst dann wird er zum Access Point Profil der entsprechenden Umgebung hinzugefügt. Der Wert `MAXIMUM_DISTANCE_THRESHOLD` gibt an, ab welcher Entfernung in Metern zwei

Positionen von Access Points vom Algorithmus nicht mehr als identisch angesehen werden. Auf diese Weise können Ungenauigkeiten bei der Positionsbestimmung berücksichtigt und ausgeglichen werden. Die Länge der Lernphase für einen Access Point wird mit dem Objekt `LEARNING_PHASE_NEW_AP_LENGTH` festgelegt. Angegeben ist hierbei die Zeit in Millisekunden. Der exemplarisch eingestellte Wert von 604.800.000 Sekunden entspricht genau sieben Tagen. Die letzten zwei Parameter geben an, welches Verfahren zur Bestimmung der Ähnlichkeit zweier Netzwerkumgebungen genutzt wird. In diesem Fall ist die Berechnung des Jaccard-Koeffizient aktiviert und mit einem Schwellwert von 0,7 eingestellt. Auch dieser Wert hat sich wie die anderen genannten im Laufe der Entwicklung und Erprobung als zielführend erwiesen (siehe hierzu Abschnitt 8) und wurde aus diesem Grund für die Simulationen verwendet.

Konfigurationsobjekt	**Wert**
ACCOUNT_UNENCRYPTED	TRUE
ACCOUNT_WPA_PSK	FALSE
ACCOUNT_WPA_ENTERPRISE	FALSE
BSSID_DELETION_THRESHOLD	-3
BSSID_ADDITION_THRESHOLD	3
MAXIMUM_DISTANCE_THRESHOLD	100,0
LEARNING_PHASE_NEW_AP_LENGTH	604.800.000
USE_JACCARD_ALGORITHM	TRUE
JACCARD_ENVIRONMENT_OK	0,7

Tabelle D.1: Konfigurationsobjekte des Erkennungssystems

Literaturverzeichnis

[1] ABIresearch. Growing Demand for Mobility will Boost Global Wi-Fi Hotspots to Reach 6.3 Million in 2013. Online: `https://www.abiresearch.com/press/growing-demand-for-mobility-will-boost-global-wi-f`, November 2013 (abgerufen am 02.11.2014).

[2] Bauer, K., Gonzales, H., and McCoy, D. Mitigating Evil Twin Attacks in 802.11. In *2008 IEEE International Performance, Computing and Communications Conference* (Dec. 2008), pp. 513–516.

[3] Bose, A., Hu, X., Shin, K. G., and Park, T. Behavioral detection of malware on mobile handsets. In *Proceedings of the 6th International Conference on Mobile Systems, Applications, and Services* (New York, NY, USA, 2008), MobiSys '08, ACM, pp. 225–238.

[4] Bulygin, Y. Epidemics of Mobile Worms. In *2007 IEEE International Performance, Computing, and Communications Conference* (Apr. 2007), pp. 475–478.

[5] Bundesamt für Sicherheit in der Informationstechnik. IT-Grundschutz Kataloge - 14. Ergänzungslieferung-2014, 2014.

[6] Chen, E., and Ito, M. Using end-to-middle security to protect against evil twin access points. In *World of Wireless, Mobile and Multimedia Networks Workshops, 2009. WoWMoM 2009. IEEE International Symposium on a* (June 2009), pp. 1–6.

[7] Claudia Eckert. *IT-Sicherheit – Konzepte - Verfahren - Protokolle.* Oldenbourg Wissenschaftsverlag, 2011.

[8] comScore. comScore Reports December 2011 U.S. Mobile Subscriber Market Share. Online: http://www.comscore.com/Press_Events/Press_Releases/2012/2/comScore_Reports_December_2011_U.S._Mobile_Subscriber_Market_Share, Februar 2012 (abgerufen am 03.06.2012).

[9] Conti, M., Dragoni, N., and Gottardo, S. MITHYS: Mind The Hand You Shake - Protecting mobile devices from SSL usage vulnerabilities. In *Proceedings of the 9th International Workshop on Security and Trust Management, STM 2013, Egham, UK* (2013), Springer, pp. 65–81.

[10] David Smith. iOS Version Stats. Online: http://david-smith.org/iosversionstats, März 2012 (abgerufen am 16.03.2012).

[11] DD-WRT. Unleash your Router. Online: http://www.dd-wrt.com, Dezember 2014 (abgerufen am 12.03.2015).

[12] De, P., Liu, Y., and Das, S. K. An Epidemic Theoretic Framework for Evaluating Broadcast Protocols in Wireless Sensor Networks. In *2007 IEEE International Conference on Mobile Adhoc and Sensor Systems* (Oct. 2007), pp. 1–9.

[13] Eidenbenz, S. Modeling Propagation Dynamics of Bluetooth Worms (Extended Version). *IEEE Transactions on Mobile Computing 8*, 3 (Mar. 2009), 353–368.

[14] EPFSUG is the European Parliament Free Software User group. Temporary Switch-off of the EP Public WI-FI Network. EP Private Wi-Fi Network Still Available. Online: http://epfsug.eu/wws/arc/epfsug/2013-11/msg00038.html, November 2013 (abgerufen am 03.02.2015).

[15] Eric Amberg. *Linux-Server mit Debian 7 GNU/Linux: Das umfassende Praxis-Handbuch.* mitp, 2014.

[16] F-Secure. Cabir worm description. Online: http://www.f-secure.com/v-descs/cabir.shtml, June 2004 (abgerufen am 10.04.2015).

[17] Felt, A. P., Finifter, M., Chin, E., Hanna, S., and Wagner, D. A survey of mobile malware in the wild. In *Proceedings of the 1st ACM Workshop on Security and Privacy in Smartphones and Mobile Devices* (New York, NY, USA, 2011), SPSM '11, ACM, pp. 3–14.

[18] Fleizach, C., Liljenstam, M., Johansson, P., Voelker, G. M., and Mehes, A. Can you infect me now? In *Proceedings of the 2007 ACM workshop on Recurring malcode* (2007), WORM '07, p. 61.

[19] Golem. Kinderleichter Hack von Mailaccounts im EU-Parlament. Online: `http://www.golem.de/print.php?a=102926`, November 2013 (abgerufen am 03.02.2015).

[20] Gonzales, H., Bauer, K., Lindqvist, J., McCoy, D., and Sicker, D. Practical defenses for evil twin attacks in 802.11. In *2010 IEEE Global Telecommunications Conference* (Dec. 2010), pp. 1 –6.

[21] Google. Android Location API. Online: `http://developer.android.com/reference/android/location/Location.html`, Dezember 2014 (abgerufen 07.12.2014).

[22] Google Inc. Google Play Services. Online: `https://developer.android.com/google/play-services/`, Januar 2015 (abgerufen 07.12.2014).

[23] Hak5 LLC. WiFi Pineapple. Online: `http://hakshop.myshopify.com/collections/wifi-pineapple-kits`, Januar 2015 (abgerufen am 05.01.2015).

[24] Henne, B., Szongott, C., and Smith, M. Towards a mobile security & privacy simulator. In *2011 IEEE Conference on Open Systems (ICOS2011)* (Langkawi, Malaysia, Sept. 2011).

[25] Henne, B., Szongott, C., and Smith, M. Coupled multi-agent simulations for mobile security amp; privacy research. In *Digital Ecosystems Technologies (DEST), 2012 6th IEEE International Conference on* (June 2012), pp. 1–6.

[26] Henne, B., Szongott, C., and Smith, M. Snapme if you can: Privacy threats of other peoples' geo-tagged media and what we can do about it. In *Proceedings of the Sixth ACM Conference on Security and Privacy in Wireless and Mobile Networks* (New York, NY, USA, 2013), WiSec '13, ACM, pp. 95–106.

[27] Ho, Y. L., and Heng, S.-H. *Mobile and ubiquitous malware.* MoMM '09. ACM Press, New York, New York, USA, 2009.

[28] Husted, N., and Myers, S. Mobile location tracking in metro areas: malnets and others. In *Proceedings of the 17th ACM conference on Computer and communications security* (2010), ACM, pp. 85–96.

[29] IEEE Computer Society. IEEE Standard for Information technology — Telecommunications and information exchange between systems — Local and metropolitan area networks— Specific requirements / Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, Juni 2007.

[30] Institut für Demoskopie Allensbach. Allensbacher Computer- und Technik-Analyse 2014 (ACTA). Online: `http://www.ifd-allensbach.de/uploads/tx_reportsndocs/PD_2014_21.pdf`, Dezember 2014 (abgerufen am 02.01.2015).

[31] iOS Developer Library. *File System Programming Guide.* Apple Inc.

[32] Juan Alegre-Sanahuja, Javier Camacho, Juan Carlos Cortés López, Francisco-José Santonja, and Rafael Jacinto Villanueva Micó. Agent-Based Model to Study and Quantify the Evolution Dynamics of Android Malware Infection. *Abstract and Applied Analysis 2014* (2014).

[33] Karlson, A. K., Meyers, B., Jacobs, A., Johns, P., and Kane, S. K. Working overtime: Patterns of smartphone and pc usage in the day of an information worker. In *Pervasive* (2009), pp. 398–405.

[34] Khouzani, M. H. R., Sarkar, S., and Altman, E. Maximum damage malware attack in mobile wireless networks. In *Proceedings of the 29th*

conference on Information communications (2010), INFOCOM'10, IEEE Press, pp. 749–757.

[35] Kim, J., Sridhara, V., and Bohacek, S. Realistic mobility simulation of urban mesh networks. *Ad Hoc Netw. 7* (March 2009), 411–430.

[36] Kindberg, T., Mitchell, J., Grimmett, J., Bevan, C., and O'Neill, E. Authenticating public wireless networks with physical evidence. In *Wireless and Mobile Computing, Networking and Communications, 2009. WIMOB 2009. IEEE International Conference on* (Oct 2009), pp. 394–399.

[37] Kumar, Vipin, Steinbach, Michael, and Tan, Pang-Ning. *Introduction to Data Mining.* Addison Wesley, 2005.

[38] Leibniz-Rechenzentrum (LRZ) der Bayerischen Akademie der Wissenschaften. Ein Grid-basiertes, föderiertes Intrusion Detection System zur Sicherung der D-Grid Infrastruktur. Online: `http://www.grid-ids.de/`, Juni 2012 (abgerufen am 10.04.2015).

[39] Martin, M. An Open Source Context Simulator. Online: `http://siafusimulator.sourceforge.net/`, 2014 (abgerufen am 03.10.2014).

[40] Mascetti, S., Freni, D., Bettini, C., Wang, X. S., Jajodia, S., and Milano, U. D. On the Impact of User Movement Simulations in the Evaluation of LBS Privacy-Preserving Techniques. In *Proceedings of the 1st International Workshop on Privacy in Location-Based Applications, Malaga, Spain* (October 2008).

[41] Matthew S. Gast. *802.11 Wireless Networks: The Definitive Guide.* O'Reilly Media, Inc., 2005.

[42] MC, G., CA, H., and AL, B. Understanding individual human mobility patterns. *Nature 453*, 7196 (06 2008), 779–782.

[43] Mikko Hypponen. Malware Goes Mobile. *Scientific American*, 295 (November 2006), 70–77.

[44] Mónica, D., and Ribeiro, C. Wifihop - mitigating the evil twin attack through multi-hop detection. In *Proceedings of the 16th European Conference on Research in Computer Security* (Berlin, Heidelberg, 2011), ESORICS'11, pp. 21–39.

[45] Murray. *Mathematical Biology*, vol. 17. Springer, 2002.

[46] Nguyen, H.-N., and Shinoda, Y. Modeling Malware Diffusion in Wireless Networks with Nodes' Heterogeneity and Mobility. In *Proceedings of 19th International Conference on Computer Communications and Networks* (Aug. 2010), pp. 1–8.

[47] Oberhaitzinger, B., Gerloni, H., Reiser, H., and Plate, J. *Praxisbuch Sicherheit für Linux-Server und -Netze*. Carl Hanser Verlag, 2004.

[48] OpenSignal. The many faces of a little green robot. Online: `http://opensignal.com/reports/fragmentation.php`, August 2012 (abgerufen am 06.06.2014).

[49] OpenStreetMap. Online: `http://openstreetmap.org`, 2014 (abgerufen am 16.01.2015).

[50] Panda Security. Eeki.a. Online: `http://www.pandasecurity.com/homeusers/security-info/215107/Eeki.A`, November 2009 (abgerufen am 07.03.2012).

[51] Peter Eckersley, and Jeremy Gillula. Is Your Android Device Telling the World Where You've Been? Online: `https://www.eff.org/deeplinks/2014/07/your-android-device-telling-world-where-youve-been`, Juli 2014 (abgerufen am 25.08.2015).

[52] Ramachandran, K., and Sikdar, B. Modeling Malware Propagation in Networks of Smart Cell Phones with Spatial Dynamics. In *26th IEEE International Conference on Computer Communications* (May 2007), pp. 2516–2520.

[53] Rivest, Ronald L., and Shamir, Adi. How to Expose an Eavesdropper. *Commun. ACM 27*, 4 (Apr. 1984), 393–394.

[54] Roth, V., Polak, W., Rieffel, E., and Turner, T. Simple and effective defense against evil twin access points. In *Proceedings of the first ACM conference on Wireless network security - WiSec '08* (Mar. 2008), p. 220.

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

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

[57] Singh, K., Sangal, S., Jain, N., Traynor, P., and Lee, W. Evaluating Bluetooth as a medium for botnet command and control. In *Proceedings of the 7th international conference on Detection of intrusions and malware, and vulnerability assessment* (Berlin, 2010), Springer-Verlag, pp. 61–80.

[58] Song, Y., Yang, C., and Gu, G. Who is peeping at your passwords at starbucks? - to catch an evil twin access point. In *DSN'10* (2010), pp. 323–332.

[59] Starbucks Corporation. WLAN-Zugang. Online: `http://www.starbucks.de/coffeehouse/wireless-internet`, Dezember 2014 (abgerufen am 03.01.2015).

[60] Statistisches Bundesamt. Wirtschaftsrechnungen - Private Haushalte in der Informationsgesellschaft - Nutzung von Informations- und Kommunikationstechnologien 2013, 2014.

[61] Symantec. SymbOS.Cabir. Online: `http://www.symantec.com/security_response/writeup.jsp?docid=2004-061419-4412-99`, 2004 (abgerufen am 16.12.2014).

[62] Symantec. Android Threat Set to Trigger On the End of Days, or the Day's End. Online: http://www.symantec.com/connect/blogs/android-threat-set-trigger-end-days-or-day-s-end, Mai 2011 (abgerufen am 16.12.2014).

[63] Szongott, C., Henne, B., and Smith, M. Evaluating the threat of epidemic mobile malware. In *Proceedings of the 8th IEEE International Conference on Wireless and Mobile Computing, Networking and Communications* (2012).

[64] Szongott, C., Henne, B., and Smith, M. Mobile Evil Twin Malnets - The Worst of Both Worlds. In *Proceedings of the 11th International Conference on Cryptology and Network Security* (2012), Springer, pp. 126–141.

[65] Szongott, Christian, Brenner, Michael, and Smith, Matthew. METDS - A Self-contained, Context-Based Detection System for Evil Twin Access Points. In *Proceedings of the 19th International Conference on Financial Cryptography and Data Security* (2015).

[66] Telekom. Fakten zum Thema Technik (1) - Wie das mobile Telefonieren funktioniert. Online: http://www.telekom.com/static/-/9982/3/fakten-mobilfunktechnik-si, 2014 (abgerufen am 08.04.2015).

[67] Tews, E., Weinmann, R.-P., and Pyshkin, A. Breaking 104 bit wep in less than 60 seconds. In *Information Security Applications*. Springer, 2007, pp. 188–202.

[68] TheiPhoneWiki Team. The iPhone Wiki. Online: https://theiphonewiki.com/wiki/Jailbreak, Dezember 2014 (abgerufen am 10.04.2015).

[69] W. Hommel, N. gentschen Felde, F. von Eye, J. Kohlrausch, M. Bräck, and C. Szongott. GIDS - Produktivsystem (Meilenstein 36). Tech. rep., D-Grid, 2012.

[70] Wang, P., González, M. C., Hidalgo, C. A., and Barabási, A.-L. Understanding the spreading patterns of mobile phone viruses. *Science (New York, N.Y.) 324*, 5930 (May 2009), 1071–6.

[71] Webkit Development Team. The webkit open source project. Online: http://www.webkit.org/, Dezember 2014 (abgerufen am 12.12.2014).

[72] Wendell Cox Consultancy. Urban Transport Fact Book. Online: http://www.publicpurpose.com/ut-cr-chicago.pdf, 2003 (abgerufen 21.03.2012).

[73] Wireshark Foundation. Wireshark. Online: https://www.wireshark.org, Dezember 2014 (abgerufen am 10.04.2015).

[74] Yan, G., Flores, H. D., Cuellar, L., Hengartner, N., Eidenbenz, S., and Vu, V. Bluetooth worm propagation. In *Proceedings of the 2nd ACM symposium on Information, computer and communications security - ASIACCS '07* (2007), p. 32.

Lebenslauf

Persönliche Daten

Name:	Christian Szongott
Geburtsdatum:	14. Juni 1981
Geburtsort	Hildesheim
Familienstand:	verheiratet

Schulische Ausbildung

07/1987 – 06/1991	Grundschule Didrik-Pinning, Hildesheim
07/1991 – 06/1993	Orientierungsstufe Ost Hildesheim
07/1993 – 06/2000	Scharnhorstgymnasium Hildesheim

Wehrdienst

09/2000 - 06/2001	1./PzArtLehrBtl 95, Munster

Berufsausbildung

10/2001 - 09/2003	Ausbildung zum technischen Assistenten für Informatik, Teutloff Bildungszentrum Hildesheim

Wissenschaftliche Ausbildung

10/2003 - 03/2007	Bachelorstudium der Informatik Gottfried Wilhelm Leibniz Universität Hannover Abschluss: Bachelor of Science
04/2007 - 06/2009	Masterstudium der Informatik Gottfried Wilhelm Leibniz Universität Hannover Abschluss: Master of Science

Wissenschaftliche Berufstätigkeit

seit 07/2009	Wissenschaftlicher Mitarbeiter und Doktorand Forschungszentrum L3S, Gottfried Wilhelm Leibniz Universität Hannover, Fakultät für Elektrotechnik und Informatik, Institut für verteilte Systeme Mitarbeit in den Projekten GIDS, DGI2, DGI2-AA, Kooperationsprojekt mit IT.N

Veröffentlichungen mit wiss. Qualitätssicherung

C. Szongott, M. Brenner, M. Smith(2015): *METDS - A Self-Contained, Context-based Detection System for Evil Twin Access Points*, Proceedings of the 19th International Conference on Financial Cryptography and Data Security 2015 (FC'15)

B. Henne, C. Szongott, M. Smith(2013): *SnapMe if You Can: Privacy Threats of Other Peoples' Geo-tagged Media and What We Can Do About It*, Proceedings of the 6th ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec 2013)

C. Szongott, B. Henne, M. Smith(2012): *Mobile Evil Twin Malnets - The Worst of Both Worlds*, Proceedings of the 11th International Conference on Cryptology and Network Security (CANS 2012)

C. Szongott, B. Henne, M. Smith(2012): *Evaluating the threat of epidemic mobile malware*, Proceedings of the 8th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob 2012)

B. Henne, C. Szongott, M. Smith(2012): *Coupled Multi-Agent Simulations for Mobile Security & Privacy Research*, Proceedings of the 6th IEEE International Conference on Digital Ecosystems and Technologies (IEEE DEST 2012)

M. Smith, B. Henne, C. Szongott, G. v. Voigt(2012): *Big Data Privacy Issues in Public Social Media*, Proceedings of the 6th IEEE International Conference on Digital Ecosystems and Technologies (IEEE DEST 2012)

N. gentschen Felde, W. Hommel, H. Reiser, F. von Eye, J. Kohlrausch, C. Szongott(2012): *Das Datenschutzkonzept für das föderierte Frühwarnsystem im D–Grid und seine technische Umsetzung*, GI–Edition – Lecture Notes in Informatics (LNI): Proceedings; 5. DFN–Forum Kommunikationstechnologien – Beiträge der Fachtagung

B. Henne, C. Szongott, M. Smith(2011): *Towards a Mobile Security & Privacy Simulator*, Proceedings of the 2011 IEEE Conference on Open Systems (ICOS2011)

C. Kunz, C. Szongott, J. Wiebelitz, C. Grimm(2009): *Design and Implementation of a Grid Proxy Auditing Infrastructure*, Proceedings of the 5th IEEE International Conference on eScience Workshops (eScience 2009)

N. gentschen Felde, W. Hommel, J. Kohlrausch, J. Köcher, C. Szongott, F. von Eye(2011): *Ein föderiertes Intrusion Detection System für das D-Grid*, Paulsen, Christian (Herausgeber): Sicherheit in vernetzten Systemen: 18. DFN Workshop