HOCHSCHULE
MITTWEIDA
UNIVERSITY OF
APPLIED SCIENCES

# BACHELORARBEIT

Herr
**Jan Bornemeyer**

## Möglichkeiten und Grenzen der Untersuchung des Netzwerkverkehrs mobiler Applikationen auf sicherheitsrelevante Aspekte mit Hilfe der Burp Suite

Mittweida, 2020

Fakultät Angewandte Computer- und Biowissenschaften

# BACHELORARBEIT

# Möglichkeiten und Grenzen der Untersuchung des Netzwerkverkehrs mobiler Applikationen auf sicherheitsrelevante Aspekte mit Hilfe der Burp Suite

Autor:
**Herr**

**Jan Bornemeyer**

Studiengang:
**IT-Sicherheit**

Seminargruppe:
**IF17wI-B**

Erstprüfer:
**Herr Prof. Dipl.-Ing. Ronny Bodach**

Zweitprüfer:
**Herr M. Sc. Stefan Schildbach**

Einreichung:
**Neuss, 30. Dezember 2020**

Verteidigung/Bewertung:
**Mittweida, 2021**

Faculty Applied Computer- and Biosciences

# BACHELOR THESIS

## Possibilities and limitations of mobile network traffic analysis for safety-relevant aspects using the burp suite

author:
**Mr.**

**Jan Bornemeyer**

course of studies:
**IT Security**

seminar group:
**IF17wI-B**

first examiner:
**Mr. Prof. Dipl.-Ing. Ronny Bodach**

second examiner:
**Mr. M. Sc. Stefan Schildbach**

submission:
**Neuss, 30. December 2020**

defence/ evaluation:
**Mittweida, 2021**

## Bibliografische Beschreibung:

Bornemeyer, Jan:
Möglichkeiten und Grenzen der Untersuchung des Netzwerkverkehrs mobiler Applikationen auf sicherheitsrelevante Aspekte mit Hilfe der Burp Suite. - 2020. - 4, 40, 10 S.
Mittweida, Hochschule Mittweida, Fakultät Angewandte Computer- und Biowissenschaften, Bachelorarbeit, 2020

## Referat:

Eine Untersuchung des Netzwerkverkehrs von 4 verschiedenen mobilen Applikationen mit Hilfe der Comminity Edition der Burp Suite der Firma PortSwigger auf sicherheitsrelevante Schwachstellen sowie das Aufzeigen der Grenzen bei der die Burp Suite zur Untersuchung von Netzwerkverkehr verwendet werden kann.

# Inhaltsverzeichnis

# Abbildungsverzeichnis

# Abkürzungsverzeichnis

| | |
|---|---|
| **HTTP** | Hypertext Transfer Protocol |
| **HTTPS** | Hypertext Transfer Protocol Secure |
| **URL** | Uniform Ressource Locator |
| **API** | Application Programming Interface |
| **CA** | Certificate Authority |
| **TLS** | Transport Layer Security |
| **AES** | Advanced Encryption Standard |
| **IDE** | Integrated Development Environment |
| **APK** | Android Package Kit |
| **SDK** | Software Development Kit |
| **SSL** | Secure Sockets Layer |
| **TLS** | Transport Layer Security |
| **VPN** | Virtual Private Network |

# 1 Einleitung

## 1.1 Motivation

In der heutigen Gesellschaft findet sich vermehrt ein Weggang von Desktop Computern oder Laptops hinzu mobilen Geräten in Form von Smartphones oder Tablets.[27] Dies liegt oft daran, dass diese Geräte in der Nutzung einfacher zu bedienen, besonders in Bezug auf die Installation und Benutzung von Applikationen sowie an der Vielzahl von Apps, etwa 2,5 Millionen beim Google Play Store und 2 Millionen beim Apple App Store. Täglich kommen in allen Stores etwa 4000-5000 neue Applikationen für verschiedene Märkte dazu. Mittlerweile ist es auch fast Standard, dass die trivialsten Applikationen eine Internetverbindung benötigen um Daten, auch Nutzerdaten, hin und her zusenden. Wegen der hohen Zahl neuer Apps ist es nahezu unmöglich, für jede Applikation einen vollständigen Sicherheitscheck durchzuführen und daher kommt es auch immer wieder dazu, dass sogar bei Apps im Bereich mobiles Banking oder Social Media schwerwiegende Sicherheitslücken ausgenutzt werden.

Bekanntes Beispiel ist hier Facebook. Nahezu 98% aller aktiven Nutzer haben Facebook über ein mobiles Gerät im Oktober 2020 aufgerufen.[26] So ist es auch kaum überraschend, dass die Produkte von Facebook zu den am meisten heruntergeladenen Applikationen sowohl im Google Play Store als auch im Apple App Store gehören. Jedoch musste das Unternehmen im Dezember 2019 und März 2020 zugeben, dass es einen massiven Diebstahl an Nutzerdaten kam. Ein Sicherheitsforscher Namens Bob Diachenko fand beide Datenlecks, bei denen es sich um insgesamt 300 Millionen Nutzerdaten inklusive Anmelde IDs, Telefonnummern, Email Adressen und privaten Details zum Profil handeln soll. Hier zeigt sich, dass sich auch bei großen Apps Fehler finden, die mit schädlicher Intention ausgenutzt werden können.

## 1.2 Zielstellung

Dieses Bachelorprojekt beschäftigt sich mit einem Aspekt der Sicherheit von mobilen Applikationen, dem Netzwerkverkehr von Apps, werde diesen näher Untersuchen und auf diverse Schwächen und Angriffsziele hinweisen, die je nach Einsatz der App sogar zu größerem Schaden führen könnten. Dies wird mit Hilfe der Burp Suite durchgeführt, einem multifunktionalen Werkzeug für die Cybersicherheit.

Die Zielstellung beinhaltet, dass man sich zuerst näher mit der Burp Suite auseinandersetzt, die Nutzung für das Android System und iOS erläutert, diverse Angriffsmethoden, die über den Netzwerkverkehr laufen vorgestellt werden und schließlich an 4 konkreten Beispielen die Nutzung der Burp Suite gezeigt wird.

## 1.3 Gliederung der Arbeit

Im Grundlagen Teil wird zu Anfang die Burp Suite und ihre verschiedenen Funktionen näher dargestellt. Es folgt eine Erläuterung über die genutzten mobilen Betriebssysteme sowie in welcher Form diese genutzt werden. Im nächsten Schritt werden die mobilen Applikationen vorgestellt, die im weiteren Verlauf der Arbeit weiter untersucht werden. Danach wird auf die Protokolle eingegangen, mit denen der Großteil des Netzwerkverkehrs mobiler Applikationen durchgeführt wird. Zum Abschluss der Grundlagen werden verschiedene Formen des Angriffes auf diesen Netzwerkverkehr aufgezeigt.

Im weiterführenden Teil der Arbeit wird dann die Untersuchung des Netzwerkverkehrs der vorgestellten mobilen Applikationen durchgeführt. Dafür wird das Vorgehen und die Methodik dargelegt, es wird gezeigt wie man verschiedene Android Systeme simuliert und diese sowie ein iPhone mit der Burp Suite nutzt. Schließlich wird dann der Netzwerkverkehr der vorgestellten Applikationen auf diverse Schwächen untersucht.

Zum Abschluss wird ein Fazit gezogen. Dies beinhaltet eine kurze Zusammenfassung, eine kritische Betrachtung der erreichten Ergebnisse von verschiedenen Seiten sowie ein Ausblick auf zukünftige Entwicklungen.

# 2 Burp Suite

Die Burp Suite ist ein multifunktionales Tool zur Analyse von Cybersicherheit der Firma Portswigger.

## 2.1 Portswigger

Portswigger wurde im Jahre 2008 von Dafydd Stuttard gegründet. Mit der Firma wollte er einen Werkzeugkasten für Sicherheitstest an Webanwendung entwickeln. Anfänglich war dies nach eigenen Aussagen ab 2004 Projekt als Hobby, er stellte jedoch fest, dass für ein solches Produkt eine Marktlücke vorhanden war. Stuttard fungiert heute als CEO sowie als Leiter der Softwareentwicklung. Der Hauptsitz der Firma ist in Knustford in Cheshire, UK mit einem Schwerpunkt auf Cyber Security, Information Technologie und Softwareentwicklung. [6]

Laut letzten Angaben hat Portswigger 27 Festangestellte bei einem jährlichen Umsatz von etwa 12 Millionen USD. [7]

Die Burp Suite wird von etwa 39000 Individuen in mehr als 120 Ländern genutzt, 80% der im Fortune 100 vertretenen Firmen, sowie 70% der Global Fortune 500 Banks und 100% der Financial Times Stock Exchange Banken nutzen das Programm zur Analyse und Informationsgewinnung für die Cybersicherheit. Für eine etwas genauere Auflistung aus welchen Sektoren Firmen die Burp Suite von Portswigger nutzen siehe Anlage 1. [6]

## 2.2 Burp Suite Grundlagen

Ein von der Firma Portswigger entwickeltes Tool für Web Application Security Testing, ist jedoch auch für jegliche Analyse von Hypertext Transfer Protocol (HTTP) und Hypertext Transfer Protocol Secure (HTTPS) nutzbar. Die Burp Suite ist in 3 verschiedenen Versionen mit unterschiedlichen Preisen erhältlich. Für diese Arbeit wird die kostenlose Community Edition verwendet.[6]

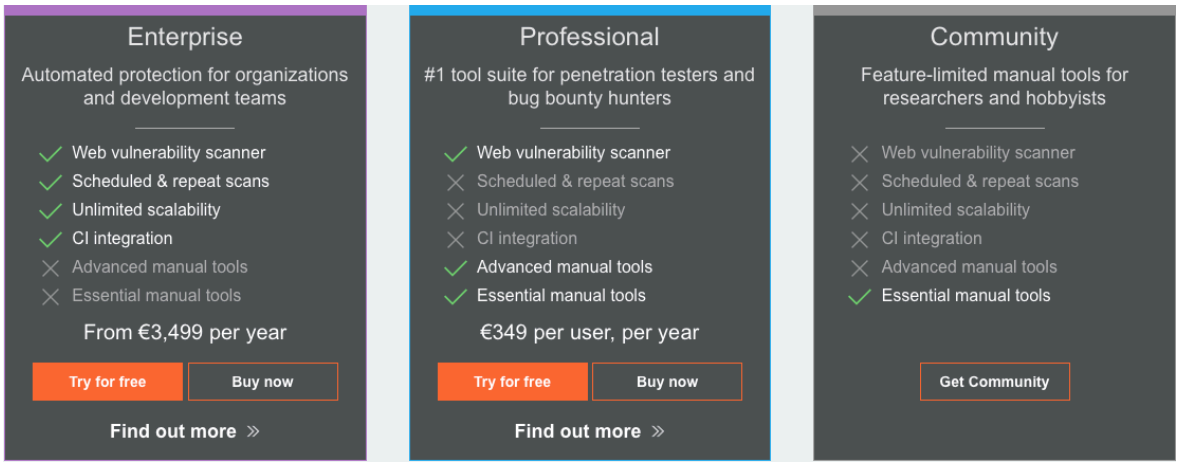



**Abbildung 1: Verschiedene Formen der Burp Suite mit offiziellen Preisen, https://portswigger.net/burp aufgerufen am 02.11.2020, 15:00**

## 2.3 Burp Suite Komponenten

Die Burp Suite kann grundlegend als ein multifunktionaler Werkzeugkasten angesehen werden, der verschiedene Werkzeuge bietet um den Netzwerkverkehr von Webseiten, Applikationen und mobilen Applikationen zu untersuchen und verschiedene Angriffe auf diesen zu testen. In der Folge werden die wichtigsten Werkzeuge für diese Zwecke vorgestellt.

In dem Bereich Target können grundlegende Informationen zur untersuchenden Web Applikation festgehalten werden. Alle aufgerufenen Uniform Ressource Identifier (URL), Formulare und vieles mehr werden erfasst und in der sogenannten „Site-Map“ dokumentiert. Der automatisierte Prozess hiervon kann sich mit Burp Spider erstellen. Hier lassen sich außerdem sogenannte „Scopes“ definieren, um bestimmte Teile der Web Applikation aus der Untersuchung auszuschließen.[8]

Das Tool Proxy ist der enthaltene Proxy Server. Dieser schaltet sich nach dem „Man-In-The-Middle“ Prinzip grundlegend zwischen Browser und Web Applikation und schneidet den HTTP und HTTPS Verkehr mit. Dies kann aber auch durch bestimmte Einstellungen dafür genutzt werden den Traffic mobiler Applikationen anzuzeigen. Über die „Interception“ Funktion lässt sich auswählen, ob nach jedem Request auf ein Weiterleiten der Anfrage gewartet werden soll oder ob die Anfrage protokolliert und automatisch weitergeleitet werden soll. Invisible Proxy ist ein Proxy für Web Applikationen, die keine Proxyeinstellung zulassen. Hierbei wird der Burp Proxy auf Port 80/443 konfiguriert. [8]

Die Scanner Funktionalität besteht aus zwei Schritten. Zuerst wird der Inhalt der Web Applikation (Links, Submitting Forms) durchsucht und das Ergebnis wird in einem Baumdiagramm dargestellt. In der 2. Phase, genannt „Auditing“, werden die gefundenen Ergebnisse und der bis dahin ausgetauschte Datenverkehr auf Schwachstellen überprüft. Dies ist nur in der Pro Version verfügbar und die Suite geht dabei grundlegend nach dem folgenden Schema vor. Im 1. Schritt (Passiv) werden die normalen Anfragen und Antworten untersucht, im 2. Schritt (Light Active) werden zusätzlich leicht veränderte Anfrage abgesendet und das folgende Verhalten untersucht, im 3. Schritt (Medium Active) werden gezielt Anfragen gestellt, die mit einer gezielten Attacke vergleichbar sind, im 4. Schritt (Intrusive Acitive) werden Anfragen so gestellt, dass man mit einer Beschädigung der Web Applikation rechnen muss und schließlich im letzten Schritt (Java Script Analysis) wird das clientseitige Javascript analysiert und auf eventuelle Schwachstellen geprüft. [8]

Intruder ermöglicht automatisierte angepasste Attacken gegen Web Applikationen. Aus einem bereits mitgeschnittenen Request wird ein Template für Intruder verwendet und die Standard Parameter durch diverse payloads ersetzt. Intruder schlägt hierbei automatisch die Position des payloads in den Parametern vor. Die Antworten der Web Applikation auf die modifizierten Requests können anschließend nach passenden Übereinstimmungen durchsucht werden und die folgenden Attacken können verbessert werden. Diese Funktion ist in der Community Edition nur stark eingeschränkt verfügbar. [8]

Über die Repeater Funktion können wiederholt HTTP Requests oder Websocket Nachrichten gesendet werden. Hier lässt sich eventuell ein verändertes Verhalten über die Quantität der Anfragen erkennen. [8]

Mit dem Burp Sequencer kann die Qualität der Zufälligkeit z.B. von Session Tokens überprüft werden. Hierbei wird eine bereits gesendete Antwort der Web Applikation selektiert und der Burp Sequencer wird die ursprüngliche Anfrage für diese Antwort wiederhott senden. Anschließend werden z.B. alle Tokens aus den Antworten extrahiert und auf Abhängigkeiten untersucht. [8]

Der Decoder, ein einfaches Burp Suite Tool mit dem man codierte Daten wieder in ihre Ursprungsform oder Rohdaten in verschiedene codierte oder gehashte Variationen umwandeln kann. Mit Hilfe heuristischer Methoden kann der Decoder selbstständig verschiedene codierte Formate erkennen. [8]

Mit dem Comparer vergleicht die Burp Suite Datensätze. Dies kann unter anderem hilfreich beim Vergleich von Site Maps mit unterschiedlichen Benutzern oder bei der Auswertung von Intruder Logs mit unterschiedlichen Angriffs Modi sein. [8]

Der Extender ermöglicht das Installieren und Verwalten von Burp Erweiterungen mit Hilfe von eigenem oder Third Party Code. Man kann dabei Erweiterungen laden und verändern, Details über installierte Erweiterungen einsehen, Erweiterung via „BApp Store“ installieren, die aktuelle Burp Extender Application Programming Interface (API) einsehen und Optionen zur Nutzung von Erweiterungen festlegen. [8]

Der Clickbandit ist ein Werkzeug mit dem man sogenannte „clickjacking“ Angriffe simulieren kann. Dabei handelt es sich um Interface basierte Angriffe bei dem ein Nutzer durch das Klicken auf den Inhalt einer Webseite einer Aktion auf einer versteckten überliegenden Webseite zustimmt. Wenn man eine Webseite gefunden hat, bei der man davon ausgeht, dass dieser Angriff Erfolg haben könnte, kann man mit diesem Tool einen Testangriff durchführen. [8]

Der Burp Collaborator kann während eines Tests genutzt werden. Damit kann man unter andrem payload generieren und mit Hilfe der Collaborator Server schauen wie sich diese payload auf den Netzwerkverkehr auswirkt. Dies ist nur in der Pro Version verfügbar. [8]

Ein Werkzeug welches das Testen von iOS Apps mit der Burp Suite ermöglicht. Dies führt über mehrere Schritte dazu, dass der Burp Suite Mobile Assistant auf einem iOS Gerät mit Jailbreak installiert wird und man kann mit diesen Veränderungen zur Analyse der Apps am iPhone/iPad vornehmen. [8]

## 2.4 Burp Suite CA Certificate

Eine Certificate Authority (CA) ist eine Organisation im Internet, die elektronische Zertifikate herausgibt, mit denen sich Identitäten im Internet überprüfen lassen. Generell kann jede Organisation diese Zertifikate ausgeben, jedoch muss man dieser Organisation auch vertrauen können. Das Zertifikat ist grundlegend der öffentliche Schlüssel, ein Ablaufdatum für die Gültigkeit des Zertifikates, der Name des Besitzers sowie gegebenenfalls weitere Informationen über den Besitzer. Die Zertifizierungsstelle zeichnet diese Zertifikate gegen und garantiert somit deren Korrektheit. Durch diese Zertifikate werden betrügerische Webseiten unter anderem daran gehindert sich als legitime Webseiten auszugeben. Auch werden die übertragenen Daten mit Hilfe des öffentlichen Schlüssels des Zertifikates verschlüsselt und Man-In-The-Middle Angriffe werden unterbunden. [9]

Dies ist jedoch ein grundsätzlicher Widerspruch in sich selbst, da das Mitschneiden des Netzwerkverkehres zur Untersuchung in der Theorie ein Man-In-The-Middle Angriff ist und somit HTTPS Traffic nicht mitgeschnitten werden kann, da der Browser oder die Applikationen erkennen, dass die Kommunikation nicht direkt zwischen Client und Web Server abläuft und somit keine Verbindung aufgebaut werden kann. Um dieses Problem zu umgehen muss auf den zu nutzenden mobilen Geräten das Burp Suite CA Certificate installiert werden. Dieses Zertifikat erlaubt Burp ein Transport Layer Security (TLS) Zertifikat für jeden einzelnen Host zu erstellen und zu unterzeichnen und somit HTTPS Verbindungen zu ermöglichen. [10]

Es wird darauf hingewiesen, dass falls ein Angreifer Zugriff auf den privaten Schlüssel des Burp Zertifikates erhält, dass dieser, ohne bemerkt zu werden, einen Man-In-The-Middle Angriff durchführen kann. Zum Schutz nutzt Burp Suite daher ein einzigartiges CA Certificate für jede neue Installation. Dennoch muss der private Schlüssel auf dem Computer oder mobilen Gerät an einem Nutzer spezifischen Ort gespeichert werden. Burp empfiehlt daher das Zertifikat nicht zu installieren, falls nicht vertrauenswürdige Personen Zugriff auf die lokalen Daten haben. [10]

# 3 Mobile Betriebssysteme Android und iOS

In diesem Kapitel wird näher auf die genutzten Geräte und simulierten Geräte, die Programme, um die Geräte virtuell zu simulieren sowie die verwendeten Versionen von Android und Apple eingegangen. Für diese Arbeit liegt ein physisches iPhone vor und wird in der Folge auch genutzt, da ein Android Gerät nicht verfügbar ist, wird dieses in verschiedenen Formen simuliert. Die dafür verwendeten Android Geräte sind rein repräsentativ und können beliebig durch andere Geräte ersetzt werden, die die gegebenen Android Versionen unterstützen.

## 3.1 Android

Android ist ein Betriebssystem das Ende 2007 von der Open Handset Alliance (OHA) vorgestellt wurde und schließlich September 2008 veröffentlicht wurde. Die OHA hat damals 34 Gründungsmitglieder, unter anderem Google, Intel Corporation, Nvidia Corporation und Qualcomm. Ziel war ein freies und open source Betriebssystem für Geräte mit Touchscreen zu entwickeln. [19]

### 3.1.1 Android 6.0 „Marshmallow“ und Android 10.0 „Q“

Android 6.0, genannt Marshmallow wurde im Anfang des Jahres 2015 vorgestellt. Im Mai 2015 gab es die Beta und die breite Öffentlichkeit erhielt Oktober 2015 Zugang zu dieser Version, als erstes die Nutzer von Nexus Geräten. Es die Nachfolge Version von Android 5.0 „Lollipop“ und wird von Google als 6. große Android Veröffentlichung und 13. Android Version gelistet. [19]

Die neuen Hauptfeatures waren damals unter anderem eine neue Genehmigungsarchitektur, eine neue kontextuelle API, ein neues Energie Management System, native Erkennung und Nutzung der Fingerabdrücke, USB C Erkennung sowie die Möglichkeit Daten und Applikationen auf eine MicroSD Karte zu verschieben. [19]

Heute nutzen noch etwa 7,4% aller Android Geräte diese Version, obwohl sie offiziell nicht mehr von Google unterstützt wird und keine sicherheitsrelevanten Updates mehr erhält. [19]

Android 10.0, genannt Q, wurde Anfang 2019 vorgestellt, erhielt im Mai 2019 eine Beta und wurde schließlich 2019 veröffentlicht. Damals erhielten Geräte der Google Pixel Reihe als 1. Zugang zu dieser Version. Es ist die Nachfolger von Android 9.0 „Pie“. Google listet diese Version offiziell als 10. große Veröffentlichung sowie als 17. Android Version. [20]

Die neuen Hauptfeatures im Vergleich zur 9.0 Version waren neue Arten und Funktionen durch Fingerbewegung im Full Screen Modus, ein sogenannter Dark Mode, eine Überarbeitung der Sicherheitsfunktionen, so dass Apps nur Daten zugreifen können, wenn diese im Vordergrund genutzt werden und nur stark eingeschränkt die Möglichkeit haben im Hintergrund Aktivitäten durchzuführen sowie eine Veränderung am Speicherzugriff für Apps, so dass diese nur noch Speicher zugreifen dürfen den diese selbst erstellt haben. Zusätzlich wurde eine neue Verschlüsselungsmethode für Geräte eingeführt, die keine Möglichkeit zu einer Beschleunigung der AES Verschlüsselung über die Hardware besitzen, dieses neue System wird Adiantum genannt. Es ist anzumerken, dass mit dieser Version eine Verschlüsselung, Hardware basiert mit AES oder über das neue Adiantum, nun ohne Ausnahme durchgeführt werden muss. [20]

### 3.1.2 Nexus 6 und Pixel 2

Das Nexus 6 wurde von Google und Motorola entwickelt und von Motorola hergestellt. Es wird als „Phablet“ bezeichnet, eine Zwischenstufe zwischen Smartphone und Tablet. Es wurde im November 2014 veröffentlicht und die Produktion schließlich im Dezember 2015 eingestellt. Es ist der Nachfolger des Nexus 5 und das Vorgängermodell des Nexus 6P. Einführungspreis war 649$ USD für eine Version mit 32GB Speicher, sowie 699$ USD für eine Version mit 64GB Speicher. In China wurde das Gerät als Moto X Pro veröffentlicht und hatte keine Google Funktionen und Apps in einer abgespeckten Version von Android. Das Gerät kam mit Android 5.0 auf den Markt und erhielt als letztes offizielles Softwareupdate die Version 7.11 im Oktober 2017. Es sind keine genauen Verkaufszahlen bekannt, jedoch wurde das Verkaufsergebnis als eher enttäuschend beschrieben. [22]

Das Pixel 2 ist ein von Google entwickeltes und von HTC hergestelltes Smartphone. Die erste Veröffentlichung fand im Oktober 2017 statt, die Produktion wurde im April 2019 eingestellt. Es ist der Nachfolger des Pixels und der Vorgänger des Pixel 3. Zur Veröffentlichung kostete das Gerät 649$ USD für eine 64GB Version und 749$ USD für eine 128GB Variante. Das Pixel 2 wurde mit Android 8.0 ausgeliefert und soll noch bis Ende 2020/Anfang 2021 Software- und Sicherheitsupdates erhalten. Seit September 2019 ist für das Smartphone Android 10.0 verfügbar. [21]

### 3.1.3 Android Studio und Android Debug Bridge

Android Studio ist ein Integrated Development Enviroment (IDE) für Googles Mobiles OS Android. Es basiert auf JetBrains IntelliJ IDE und wurde im Jahre 2014 mit der Version 1.0 veröffentlicht und ist verfügbar für Windows, MacOS und Linux. Seit Mai 2019 ist die präferierte Entwickler Sprache Kotlin vor Java und C++. Oktober 2020 wurde die bisher letzte Version 4.1 veröffentlicht. Das Programm selbst ist kostenlos, es gibt jedoch ein Abonnement Modell, um erweiterte Funktionen freizuschalten. [12]

Die wichtigen Funktionen beinhalten eine Entwicklungsumgebung für Android, Android Wear und Android TV Apps mit einem Gradle-Based Build Support, Refrectoring für Android Apps, Lint Tool, ProGuard für Java Code, welches das automatisierte Aufräumen des Codes beinhaltet, Layout Editor mit einer Drag&Drop Funktion, Support für die Google Cloud Plattform, das Android Virtual Device, einen Emulator für Android Geräte sowie einen Android Package Kit (APK) Analyzer. Mit „.apk“ werden Android Programmpakate versehen. [11]

Bei Android Geräten ist es im Gegensatz zu iOS Geräten möglich, auch ohne den vorhandenen Appstore zu nutzen, eine Applikation zu installieren. Damit dies gemacht werden kann, muss die Datei im .apk Format vorliegen. Es gibt diverse Webseiten die einen Download von einer Vielzahl Programmen in diesem Format ermöglichen, es ist jedoch darauf hinzuweisen, dass dies keine offiziellen Webseiten von Google sind. Daher ist keine Sicherheit darauf gegeben, dass der Code nicht gegebenenfalls mit schädlicher Intention verändert wurde. Es finden sich dort auch Applikationen die explizit vom Google Playstore entfernt wurden, daher kann das System je nach Version eventuell direkt wieder entfernen. [12]

Die Android Debug Bridge ist eine Software Schnittstelle für das Android System und wird dafür genutzt, um auf Android Smartphones zuzugreifen und dort Befehle über das Terminal auszuführen. Es ist ein Bestandteil des größeren Android Software Development Kits. Es bietet die Möglichkeit Daten auf das Gerät zu übertragen, Befehle auszuführen und auf bestimmte Komponenten des Systems zuzugreifen. Die Android Debug Bridge besitzt keine grafische Oberfläche und kann nur über das Terminal verwendet werden. Es ist möglich über USB oder TCP eine Verbindung aufzubauen sowie eine direkte Nutzung mit einem vom Android Studio emulierten Gerätes. [13]

Die wichtigsten Befehle sind ./abd devices, welches alle angeschlossenen Geräte anzeigt, ./adb push <Datei> <Zielort> mit dem eine Datei auf das Android Gerät kopiert werden kann, Zielort ist zum Beispiel /sdcard/Downloads. Mit ./adb shell kann man Befehle direkt auf dem mobilen Gerät ausführen und schließlich mit ./adb install <Datei> kann eine Applikation im .apk Format installiert werden. [14]

## 3.2 iOS

iOS ist das von Apple entwickelte Betriebssystem für die Geräte der iPhone, iPad und iPod Touch Reihe. Im Gegensatz zu Android ist iOS nur für die eigene Apple Hardware verfügbar und kann nicht für andere Geräte lizensiert werden. iOS basiert grundlegend auf Apples eigenem macOS Betriebssystem. [24]

### 3.2.1 iPhone 8 und iOS 14

Das iPhone 8 wurden von Apple entwickelt und von Foxconn und Pegatron hergestellt. Die Veröffentlichung des Gerätes fand im September 2017 statt, die Produktion wurde im April 2020 eingestellt. Zur Auslieferung war iOS 11 installiert, seit Juni 2020 ist iOS 14 für das iPhone verfügbar. Die Preise starteten für das iPhone 8 starteten bei 699$ USD. Das Vorgängermodell war das iPhone 7, der direkte Nachfolger das iPhone X, nach Baureihe jedoch das iPhone SE der 2. Generation. Unterstützt werden alle Geräte ab dem iPhone 6S sowie den iPod Touch der 7. Generation. [23]

iOS 14 ist die 14. große iOS Veröffentlichung von Apple für iPhone und iPod Touch und der Nachfolger von iOS 13. Es wurde im Juni 2020 angekündigt und war ab September 2020 für alle unterstützten Geräte verfügbar. Zu den neuen Features gehören App Clips, durch die man kleine Apps durch QR-Codes starten kann, eine überarbeitete Version von CarPlay, CarKey mit dem man ein Auto über NFC mit einem Handy aufschließen und starten, eine Bild-in-Bild Funktion für verschiedene Apps, ein offline Übersetzer, Widgets und eine App Mediathek. [24]

# 4 Mobile Applikationen

In diesem Kapitel stelle ich kurz die von mir untersuchten Applikationen vor, dabei handelt es sich meiner Meinung nach um Applikationen, die zu den wichtigen Bereichen gehören. Eine Applikation aus dem Bereich mobiles Banking, eine aus dem Feld der News Apps, die App von Amazon, dem größten Onlinehändler in Deutschland, sowie aus aktuellem Anlass die Corona Warn App der Bundesregierung.

## 4.1 Sparkassen App

Die Sparkassen App ist für Android und iOS erhältlich und ist nach eigenen Angaben die in Deutschland meistgenutzte Banking App. Die wichtigen und grundlegenden Funktionen sind das Abrufen der Kontostände, Überweisungen, Postfächer, die Nutzung von ApplePay und GooglePay mit dem Smartphone über Sparkassen Konten sowie ein direkter Kontakt zu Beratern und Kundensupport. Die Applikation wird beim Start über ein Passwort entsperrt, bei neueren Geräten ist auch eine Entsperrung via Fingerabdruck möglich. Zusätzlich zu dieser Applikation stellt die Sparkasse auch eine pushTAN App bereit, mit der man die TAN für Transaktionen oder andere Handlungen erhält. [28][29]

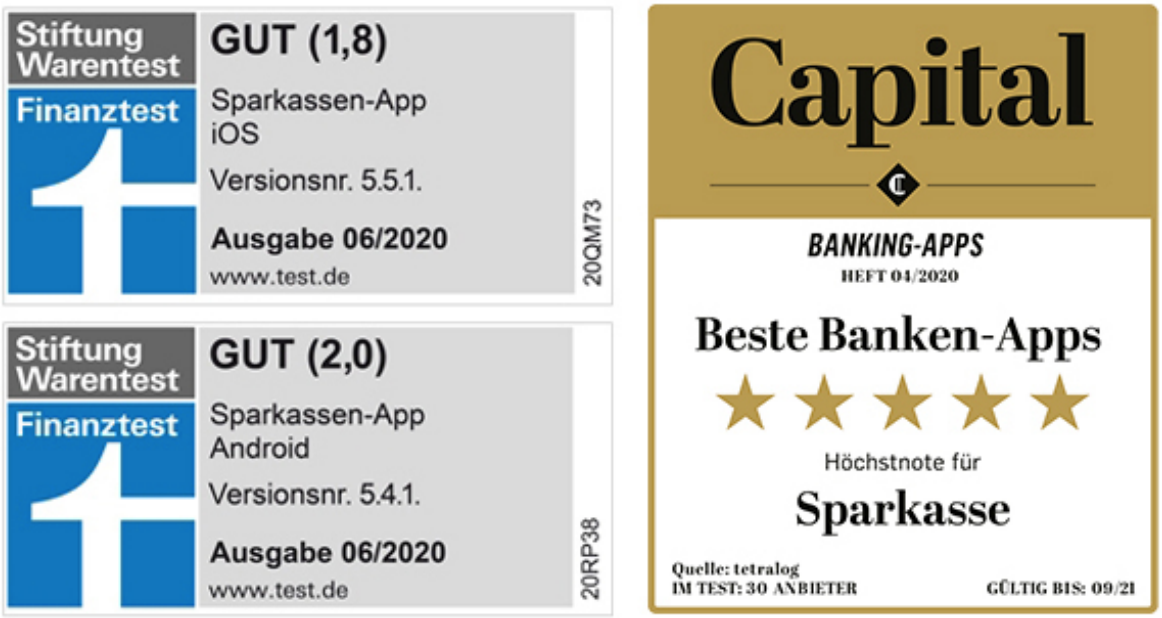



**Abbildung 2: Aktuelle Bewertungen der Sparkassen App, https://bit.ly/2JUvatq verfügbar am 02.11.2020, 15:10**

Die Sparkassen App ist im Google Play Store als Version 5.8.0 verfügbar, hat eine Größe von 40MB, erfordert Android 6.0 oder höher und hat 10.000.000+ Installationen (stand November 2020). Im Apple App Store findet man die App mit Version 5.9.0, einer Größe von 123MB, erfordert iOS 12.4 oder höher und wird auf Platz 3 unter allen Finanz Apps gelistet (stand November 2020). [28][29]

## 4.2 WELT News App

Die Welt App ist die zugehörige App zum Welt Fernsehsender, ehemals N24, welche Teil der Axel Springer Gruppe ist. Sie bietet Nachrichten zu allen grundlegenden Themen von Aktuell, Wirtschaft, Geld, Politik im In- und Ausland, Sport und viele mehr in der deutschen Sprache. Die App bietet ein Abo Modell für sogenannte Welt+ Artikel, diese sind aber oft auch für eine gewisse Zeit zu Anfang für alle verfügbar. Für die meisten Artikel gibt es eine Kommentar Funktion, die jedoch erst nach einer Registrierung freigeschaltet werden. Die Welt News App hat für Android die Version 6.4.0 und eine Größe von 24MB, erfordert Android Version 4.1 oder höher und hat 1.000.000+ Installationen (stand November 2020). Für iOS Geräte ist die App mit Version 5.5.0 erhältlich und hat eine Größe von 36MB, es wird iOS 13.0 oder höher vorausgesetzt (stand November 2020). Die App wird im Apple App Store auf Platz 2 aller Applikationen des Bereiches Zeitungen und Zeitschriften gelistet. [30][31]

## 4.3 Amazon App

Die Amazon App ist grundlegend eine angepasste Version der Amazon Webseite für mobile Geräte in Form einer Applikation. Man kann die exakt gleichen Aktionen durchführen unter anderem das Kaufen von Gegenständen, Objekte zur Beobachtung in Listen ablegen, Kontoinformationen ändern oder neu hinzufügen. Die Amazon App ist im Google Play Store mit Version 20.22 erhältlich und erfordert eine Android Version von 5.0 oder höher, die Größe variiert je nach Gerät und es gab bisher 100.000.000+ Installationen (stand November 2020). Im Apple App Store findet sich die Amazon App mit Version 15.21 wieder, besitzt eine Größe von 116MB und setzt iOS 12.0 oder höher voraus. Zusätzlich ist die App auf Platz 3 aller Applikationen für den Bereich Shopping (stand November 2020). [32][33]

## 4.4 Corona Warn App

Die Corona Warn App ist eine Applikation, die von der Bundesregierung in Verbindung mit dem Robert-Koch-Institut entwickelt wurde. Seit Juni 2020 ist sie in Deutschland verfügbar und seit Juli 2020 auch für andere EU-Staaten. Die Corona Warn App bietet eine Kontaktnachverfolgung für die Covid-19 Epidemie indem die App das Risiko einer Infektion nach Kontakt mit positiven Personen ermittelt und dem Nutzer mitteilt. Die Corona Warn App wurde stand September 2020 etwa 18,4 Millionen Mal heruntergeladen, 9,8 Millionen auf Android Geräten und 8,6 Millionen Mal auf Apple Geräte. [34][35]

# 5 Protokolle für den Netzwerkverkehr

In diesem Abschnitt gehen ich näher auf die beiden Formen des Netzwerkverkehres ein, die den wichtigen Teil des Traffics von mobilen Applikationen ausmachen. Dabei handelt es sich um HTTP und HTTPS.

## 5.1 HTTP

Das Hypertext Transfer Protocol ist ein zustandsloses Client-Server Protokoll auf Schicht 7 des OSI-Anwendungsmodell, der Anwendungsschicht. Zustandslos bedeutet in diesem Zusammenhang, dass Anfragen unabhängig von vorherigen Anfragen behandelt werden und keine Sitzungsinformationen ausgetauscht werden. Der gesamte Netzwerkverkehr läuft bei diesem Protokoll über Port 80. Haupteinsatz Feld ist das Übertragen von Daten von Webseiten in einem Webbrowser, aber es ist nicht darauf beschränkt. Daher nutzen zum Beispiel Applikationen diese Form und HTTPS zum Großteil für ihre Netzwerkübertragungen. Die große Schwäche an HTTP ist, dass der gesamte Datenverkehr zwischen Client und Server unverschlüsselt abläuft und es daher sehr anfällig für sogenannte „Man in the Middle“ Angriffe ist. Zum Schutz vor solchen Angriffen wurde HTTPS entwickelt.[15]

Der Grundsätzliche Ablauf findet wie folgt statt. Ein Nutzer sendet einen HTTP Request an einen Server, der darauf hin nach der gewünschten Datei sucht und mit einem HTTP Header antwortet. Wird die Datei gefunden sendet er in der Folge auch den sogenannten Message Body, der den gefragten Inhalt enthält. [16]

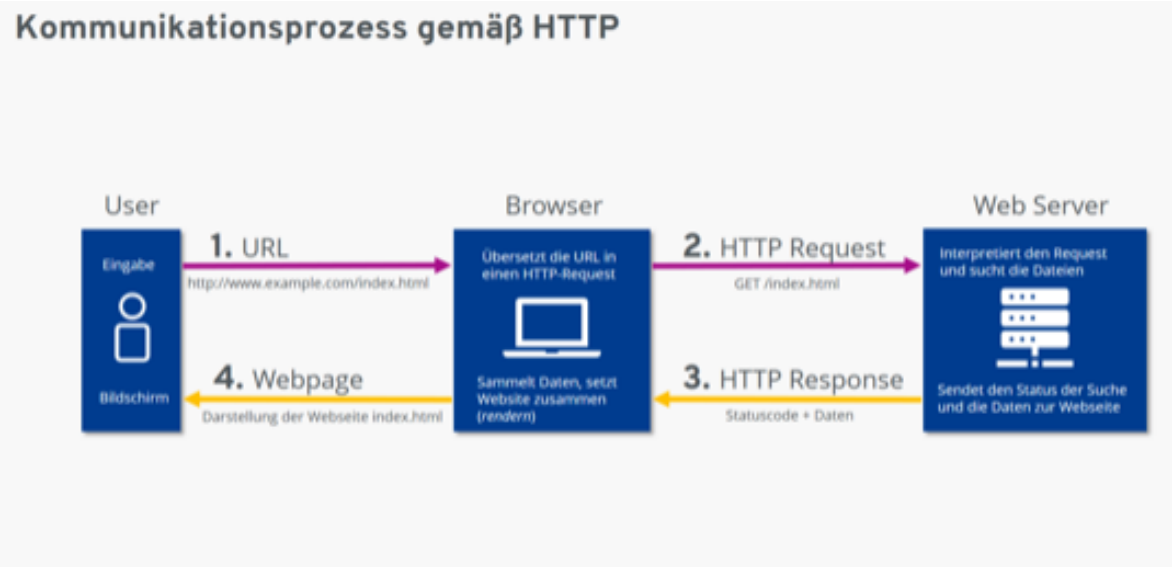



**Abbildung 3: Schematischer Ablauf eines Aufrufes einer Webseite via HTTP, [16]**

HTTP wurde 1991 eingeführt und der aktuelle Standard ist seit 2015 HTTP/2. Der Schritt zu HTTP/2 musste erfolgen, da Webseiten immer größer wurden und mehrere Megabyte an Daten versendeten. Mit HTTP/1 wären dies hunderte einzelne HTTP Requests die dem Protokoll nach alle nacheinander hätten abgearbeitet werden müssen. Dies hätte zu unglaublich langen Wartezeiten geführt. HTTP/2 setzt daher auf einige Änderungen. So basiert das Protokoll nun auf Binärdateien statt auf Textdateien und Multiplex erlaubt das Senden von mehreren HTTP Requets zwischen Client und Server parallel. Die HTTP Header wurden komprimiert, da diese ohnehin oft sehr redundante Informationen enthalten. Zudem wurde Server Push eingeführt, dabei wird es dem Server ermöglicht, dem Client Daten, ohne einen vorherigen HTTP Request zu senden, wenn der davon ausgeht, dass der Client diese sowieso noch anfragt.[15]

Heute läuft etwa 45% des gesamten HTTP Traffics in Form von http/2 ab. Für die Zukunft ist HTTP/3 geplant, bei dem ein Wechsel von TCP zu UDP als Grundlage stattfinden soll. Das momentane Problem an TCP ist, dass der Client jedes empfangene Datenpaket bestätigen muss, bevor das nächste gesendet werden kann, was zu sogenannten Head of Line Blocking Fehlern führen kann, falls eines der Pakate verloren geht und man darauf warten muss, dass dieses erneut gesendet wird. [16]

## 5.2 HTTPS

Das Hypertext Transfer Protocol Secure wurde als Verbesserungen des HTTP Netzwerkverkehres entwickelt, um Schutz vor Man in the Middle Angriffen zu bieten. Es läuft über Port 443 grundlegend ist es HTTP, mit einer zusätzlichen Verschlüsselung des kompletten Datenverkehres, um Angriffe auf diesen zu erschweren oder zu unterbinden. Dies wird dadurch erreicht, dass zu Beginn der Kommunikation der Server ein Zertifikat an den Client sendet, damit dieser seine Vertrauenswürdigkeit bestätigen kann. [17]

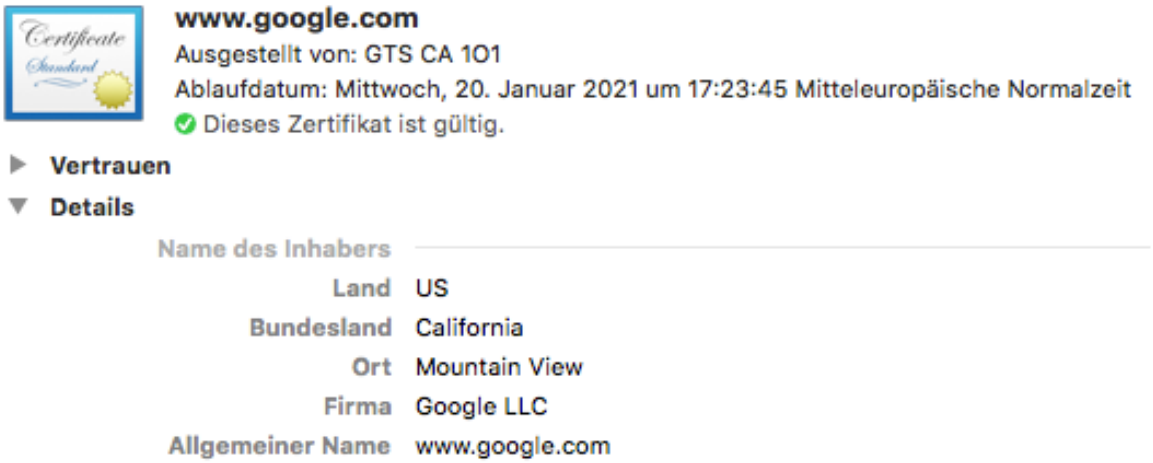



**Abbildung 4: Sicheres HTTPS Zertifikat von www.google.com, eigene Abbildung**

An Abbildung 4 kann man die wichtigen Inhalte eines Zertifikates sehen. So wird angezeigt wer das Zertifikat ausgestellt hat, bis wann es gültig ist und wer der Inhaber des Zertifikates ist.


**Abbildung 5: Warnung vor ungültigem HTTPS Zertifikat aufgrund vom Ablaufdatum, eigene Abbildung**

Abbildung 5 zeigt, wie es aussieht, wenn ein Browser, in diesem Fall Safari, ein Zertifikat als nicht sicher ansieht. Bei diesem Beispiel hat das Zertifikat seine zeitliche Gültigkeit verloren und die Firma selbst hat es nicht erneuert.

Der 2. Große Unterschied zwischen HTTP und HTTPS ist, dass HTTPS als Transportprotokoll SSL/TLS statt TCP nutzt. Der Ablauf in seinen Grundzügen findet wie folgt statt. Der Client kontaktiert den Server. Der Server sendet zu Beginn das Zertifikat, um seine Identität zu authentifizieren. Der Client prüft das Zertifikat und sendet dem Server eine Zufallszahl, die mit dem öffentlichen Schlüssel des Servers verschlüsselt wird. Server erzeugt mit der Zufallszahl, welche nur er mit seinem privaten Schlüssel entschlüsseln kann, einen Sitzungsschlüssel. Dieser dient als Grundlage für die Verschlüsselung der Kommunikation zwischen Server und Client. Der Server nutzt den Diffie-Hellmann-Schlüsselaustausch, um dem Client den Sitzungsschlüssel zuzusenden. Vorteil dieser asymmetrischen Verschlüsselung ist, dass die gesendeten Inhalte selbst nicht verschlüsselt werden, was dieses System deutlich schneller ablaufen lässt als andere Sicherheitssysteme, jedoch ist es im Vergleich zu HTTP langsamer, da Dinge wie Zertifikat, Schlüsselerzeugung und Austausch der Schlüssel einen gewissen Rechenaufwand erfordern.[18]

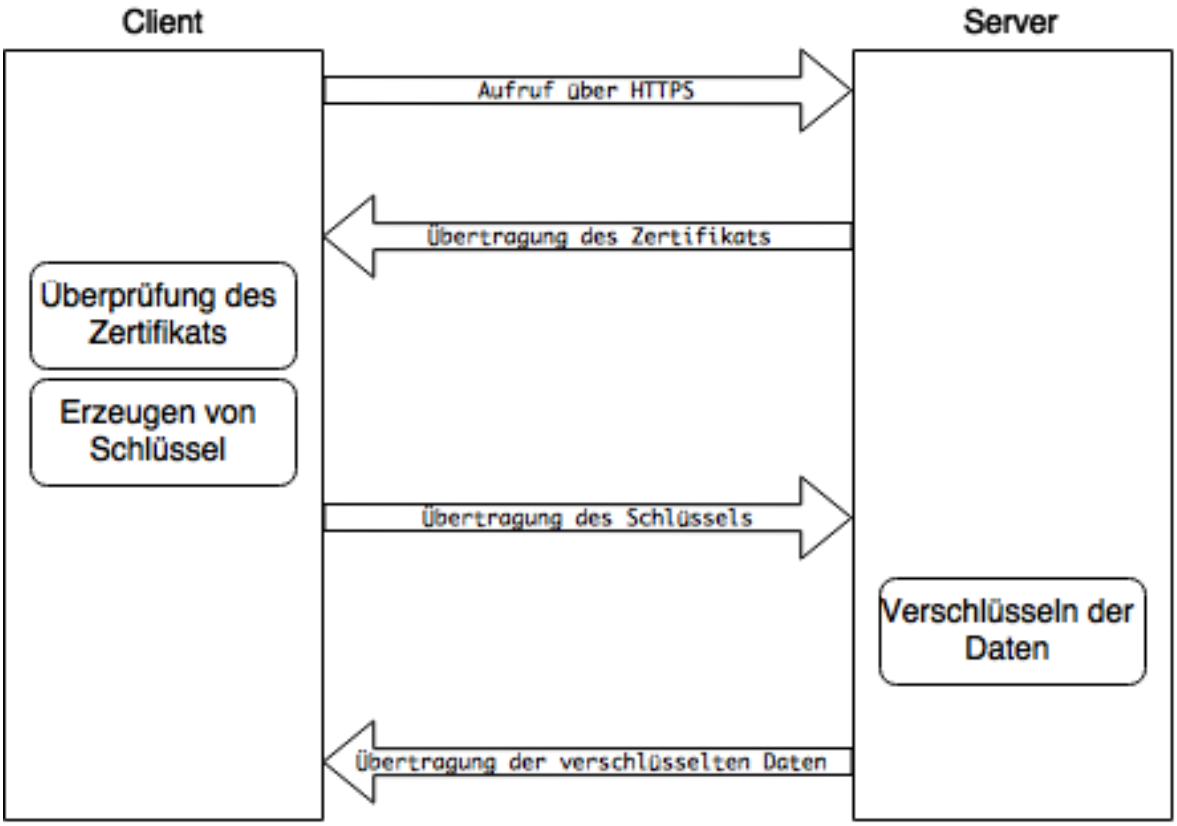



**Abbildung 6: Schematischer Ablauf des Übertragens von Daten via HTTPS, [18]**

Aufgrund seines Sicherheitsstandards wird HTTPS auch vermehrt in anderen Bereichen als nur dem Aufrufen von Webseiten eingesetzt. Daten können vor den meisten Angriffen gesichert versendet werden. Daher ist HTTPS auch meist der Standard für Webseiten oder Applikationen im Bereich des Bankings oder Einkaufens, da hier sensible Daten genutzt werden. [18]

# 6 Angriffsmethoden auf den Netzwerkverkehr mobiler Applikationen

Die größte und am meisten ausgenutzte Schwäche des Netzwerkverkehres von mobilen Applikationen ist der Man in the Middle Angriff. Im folgenden Kapitel wird näher auf die Grundlagen dieses Angriffs eingegangen und verschiedene Formen des Man in the Middle Angriffs beschrieben, die in der jüngeren Vergangenheit oft zum Einsatz kamen.

## 6.1 Man in the Middle

Der „Man in the Middle“ ist eine Bezeichnung für einen Angriff auf generellen Netzwerkverkehr und findet sich auch oft bei Angriffen auf den Netzwerkverkehr wieder. Grundlegend wird dabei der Netzwerkverkehr zwischen Person und Person B von einer dritten Person abgefangen und gegebenenfalls verändert bevor er die jeweilige andere Person erreicht oder es kann sogar dazu kommen, dass die dritte Person sich dazu entscheidet Netzwerkpakete zu verwerfen. Dies ist natürlich auch für Traffic zwischen Client und Server und ähnlichem möglich. Um dies für mobile Geräte und deren Applikationen zu erreichen, ist es am einfachsten, wenn man sich als Proxy zum mobilen Gerät schaltet und somit der gesamte Netzwerkverkehr über diesen Proxy abläuft. Es ist in vereinfachter Form genau das, was die Burp Suite unternimmt, um den Netzwerkverkehr von Applikationen zu untersuchen und daher ist es auch mit gewissen Hürden in der Sicherheit versehen und wird offiziell nicht von Google in Android und Apple in iOS unterstützt. Das Ziel des Man in the Middle Angriffes für mobile Applikationen kann das abgreifen von unverschlüsselten Log-In Daten oder Kontoinformationen, die die App an den Server sendet oder mit denen der Server an den Client antwortet, Verhindern das Pakete vom Client zum Server oder vom Server zum Client gesendet werden, um zum Beispiel einen Login zu unterbinden oder sogar mit falschen Paketen zu antworten, um unter anderem einen Client auf eine falsche Webseite zu leiten. Die einfachsten Mittel gegen Man in the Middle Angriffe sind das komplette Verschlüsseln der sensiblen Informationen innerhalb des Netzwerkverkehres. [1][2]

## 6.2 Replay Attack

Replay Attack ist eine Form des Man in the Middle Angriffes. Bestimmte Informationen, die man durch das Mitschneiden des Netzwerkverkehres erhalten hat, werden zu einem späteren Zeitpunkt erneut genutzt, um dem Server eine sichere Verbindung vorzutäuschen. In dieser Variation kann man auch die Daten abfangen und selber an das gewünschte Ziel weiterleiten, so dass der eigentliche Client denkt, dass er könnte keine Verbindung aufbauen während der Server annimmt, dass er mit dem eigentlichen Client in Kontakt steht. Durch das Nutzen von Session IDs, wie sie von HTTPS genutzt werden, kann diese Form des Angriffes leicht unterbunden werden, da die alten Pakete so nicht erneut genutzt werden können.[1]

## 6.3 Phishing Attack

Bei dieser Variante des Man in the Middle Angriffes wird ein Nutzer dazu verleitet seine Login Daten durch eine gefälschte Applikation sensible Daten preiszugeben. Dies tritt eher selten auf Apple Geräten auf, da dort nur Applikationen über den App Store installiert werden können, wohingegen bei Android Geräten auch Applikationen über den Browser des Gerätes heruntergeladen und installiert werden können. So kann es sein, dass man zum Beispiel dadurch, dass man durch das Weiterleiten von falschen Paketen während eines Man in the Middle Angriffes auf eine gefälschte Webseite gelenkt wird und dort dann statt der echten Banking App eine Fälschung herunterlädt, die dann die Login Daten abgreift. [1]

## 6.4 Session Hijacking

Beim Session Hijacking versucht man während eines Man in the Middle Angriffes das sogenannte HTTP Cookie zu erhalten. Dieses Cookie enthält die Session ID, welche schon bei der Replay Attack eine Rollte spielte. Falls es dem Angreifer gelingt diese Session ID zu erbeuten, kann er sich während der aktuellen laufenden Sitzung ebenfalls als der authentifizierte Nutzer ausgeben. Dies kann zum Beispiel schwere Konsequenzen haben, falls die Applikation es erlaubt bei einem angemeldeten Nutzer das Passwort zu ändern, ohne das aktuelle einzugeben. Damit kann der Angreifer den eigentlichen Nutzer aus seinem Account aussperren. [3]

## 6.5 Traffic Analysis

Dies ist wahrscheinlich die grundlegendste Form eines Man in the Middle Angriffes und im Grunde auch die am schwersten zu entdeckende, denn es kann Wochen oder sogar Monate ablaufen, bis Schaden entsteht. Der komplette Netzwerkverkehr wird mitgeschnitten und je nachdem wie Apps bestimmte Daten verschlüsseln oder nicht verschlüsseln werden immer mehr Daten über den Nutzer gesammelt. Dies kann dann genutzt werden, um Rückschlüsse auf das Nutzerverhalten zu ziehen und wird meist als Grundlage für Social Engineering Angriffe genutzt. [4]

## 6.6 Account Lockout Attack

Dieser Angriff während einer Man in the Middle Aktion basiert auf einem oft genutzen einfachen Sicherheitssystem für Login Authentifizierungen. Sollte bei einem Login der Nutzername korrekt sein, dass Passwort hingegen falsch und sollten mehrere falschen Logins versuche in kurzer Zeit folgen, so wird es für eine bestimmte Zeit unterbunden sich in den Account einzuloggen, sogar mit einem korrekten Passwort. Oft muss dann das Passwort zurückgesetzt werden. Der Angreifer kann in diesem Fall oft den Nutzernamen mitschneiden, da diese in den meisten Fällen im Gegensatz zu Passwörtern nicht verschlüsselt werden und macht sich dann das oben beschriebene System zu nutzen, um den Nutzer, wenn auch temporär, aus seinem Account auszusperren. [5]

# 7 Untersuchung des Netzwerkverkehrs der mobilen Applikationen mit Hilfe der Burp Suite

Das 7. Kapitel beschäftigt sich schließlich mit der Untersuchung des Netzwerkverkehres der vorgestellten Applikationen. Zuerst wird gezeigt, wie für diese Arbeit die Android Systeme simuliert werden und wie für diese und iOS 14.0 der Burp Suite Proxy eingerichtet wird. Danach wird die Burp Suite genutzt, um den Netzwerkverkehr der Applikationen genauer zu betrachten. Ein besonderes Augenmerk wird dabei auf die Pakete gelegt, die während kritischen sicherheitsrelevanten Momenten versendet werden, wie beim Einloggen in einen Account oder der Registrierung eines Accounts. Auch wird sich damit beschäftigt inwieweit man mit der Burp Suite eine Aussage darüber treffen kann, ob Pakete sicher sind oder ob Grenzen bei der Untersuchung bestehen.

## 7.1 Simulieren einer Android Umgebung und Einrichten des Burp Suite Proxy

In diesem Abschnitt wird erklärt, wie man die virtuellen Android Geräte erstellt und diese so eingerichtet, dass sie mit der Burp Suite genutzt werden können. Dafür werden die im Vorfeld genannten Programme Android Studio und Android Debug Bridge sowie die Burp Suite genutzt. Der Grund für die Nutzung von 2 verschiedenen Android Versionen liegt daran, dass die Einrichtung für 6.0 deutlich einfacher ist als für 7.0 und höher, aufgrund von eingeführten Sicherheitsfeatures. Jedoch lehnen wegen der veränderten Sicherheitsarchitektur bestimmte Apps auf den höheren Versionen das von der Burp Suite erstellte CA Certifcate ab. Daher sollte man, solange es noch möglich ist, mit den Apps auf der Version 6.0 arbeiten, sofern diese vorhanden sind. Da für diese Arbeit ein iPhone 8 vorlag, wurde dieses anstelle einer virtuellen Umgebung genutzt, daher wird in der Folge nur dargelegt, wie man den Proxy für die iOS 14.0 einrichtet.

### 7.1.1 Android 6.0

Zuerst richtet man das Android Studio für die jeweilige eigene Nutzung entsprechend ein. Nach dem Start des Programmes wählt man die Einstellungen und den SDK Manager aus und entscheidet sich dort für die Android Versionen, die man in der Folge nutzen möchten. Diese werden dann automatisch am auf den genannten Pfad heruntergeladen, entpackt und installiert.

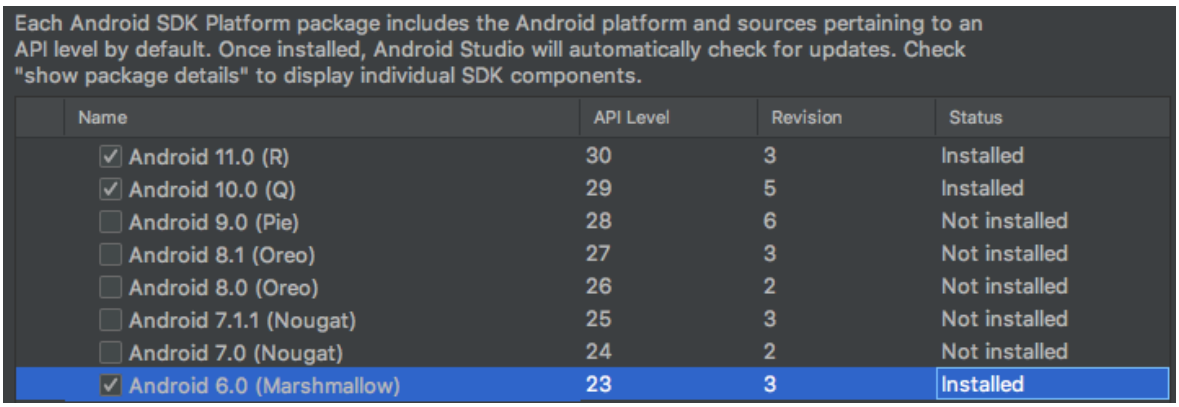



**Abbildung 6: Verschiedene Android Versionen für Android Studio, eigene Abbildung**

Danach wählt man die Einstellungen aus und navigiert zum AVD (Android Virtual Device) Manager, um das virtuelle Android Gerät zu erstellen. Man hat die Auswahl zwischen TV, Phone, Tablet, WearOS und Automotive. Es werden die Abmessungen des Gerätes angegeben sowie die zusätzliche Information ob der Google PlayStore auf diesem simulierten Gerät verfügbar ist. Wählt man das Gerät seiner Wahl aus, gibt man in der Folge die Version des mobilen Betriebssystems an, die man dafür haben möchte. So sind jedoch nur Kombinationen möglich, die die physischen Geräte auch durchführen können. Das Pixel 2 zum Beispiel wurde mit Android 8.0 ausgeliefert, jedoch liefen die Testgeräte zu Anfang mit 7.0, welches dann auch für das Gerät verfügbar ist.

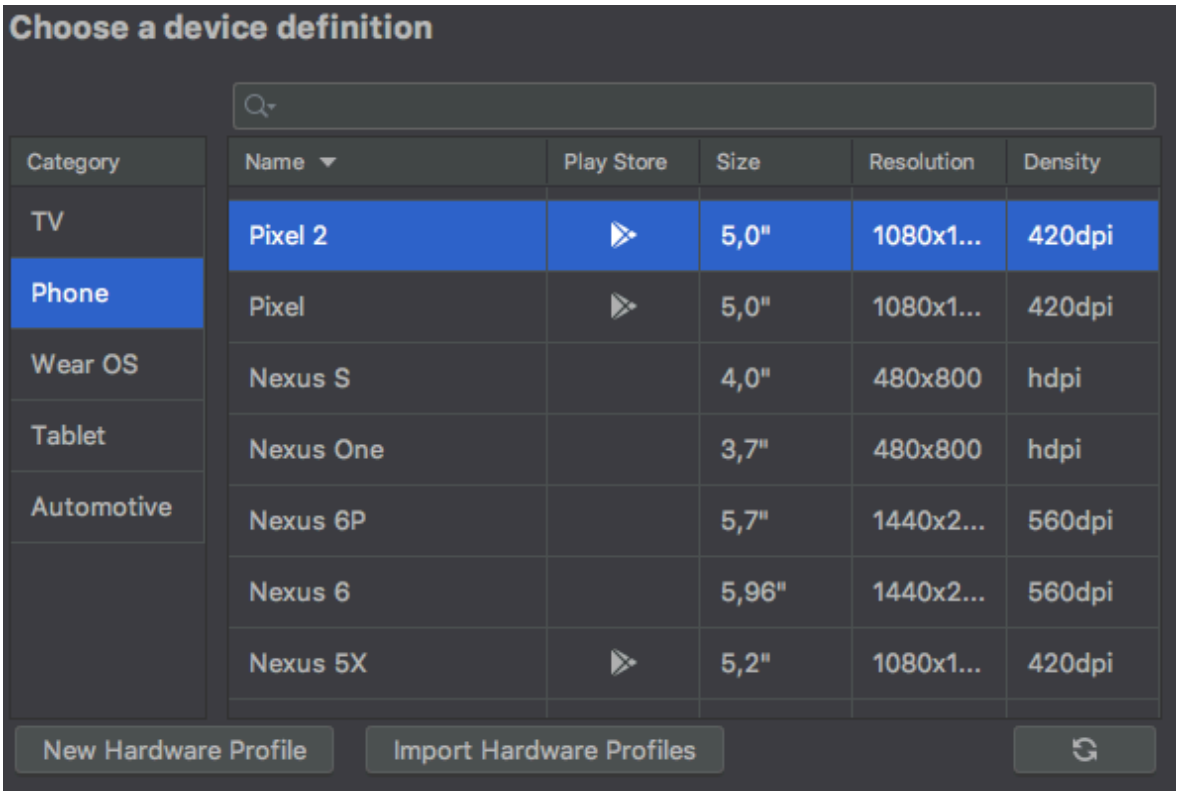



**Abbildung 7: Verschiedenen Geräten die im Android Studio verfügbar sind inklusive der von mit genutzten Pixel 2 und Nexus 6, eigene Abbildung**

Hat man das Gerät erstellt, kann man es über den AVD Manager starten und man sollte man sich mit Grundlegenden Funktionen vertraut machen (Siehe Abbildung 3 und 4). Der Mauszeiger kann die Bewegungen des Fingers für den Touchscreen ersetzen, bestimmte Funktionen, für die man jedoch mehrere Finger benötigt, lassen sich so nicht durchführen. Auch sollte man austesten ob das Gerät korrekt mit dem Internet verbunden ist. Die durch das Android Studio simulierten Geräte nutzen das gleiche Netzwerk des Computers, das virtuelle Gerät zeigt dies jedoch als mobile Daten an. Ein Bild des virtuellen Pixel 2 und Nexus 6 finden sich in unter Anlange 3 und 4.

Um Applikationen auf ein virtuelles Gerät ohne Google AppStore zu installieren, muss man zuerst die gewünschte App als apk Datei herunterladen. Die von mir verwendete Seite ist www.apk-dl.com, es finden sich aber auch viele verschiedene Webseiten, die diese Funktion anbieten. Hat man die Applikation heruntergeladen und das virtuelle Gerät läuft, dann kann man die Android Debug Bridge nutzen, um es auf dem Gerät zu installieren. Die Android Debug Bridge wird zusammen mit Android Studio installiert und findet sich unter MacOS am folgenden Pfad: /Users/[user]/Library/Android/sdk/platform-tools/. Die Android Debug Bridge ist wie im Vorfeld schon erwähnt eine Anwendung ohne grafische Oberfläche und kann daher nur über das Terminal verwendet werden. Man sollte mit ./adb devices schauen ob das virtuelle Gerät mit der Debug Bridge verbunden ist, sollte dies der Fall sein kann man mit ./adb install <Download Pfad> die heruntergeladene Applikation installieren.


**Abbildung 8: Nutzung der Android Debug Bridge zum Installieren der Welt News App auf ein virtuelles Nexus 6, eigene Abbildung**

Um nun den Datenverkehr der Applikationen abzugreifen muss man nun noch entsprechend die Burp Suite für die Android Umgebung einrichten. Dafür startet man die Burp Suite und wählt Proxy gefolgt von Optionen aus. Unter Proxy Listeners wählt man schließlich den vorhanden aus und ändert diesen oder fügt einen neuen hinzu. Man muss Port und Adresse entsprechend der genutzten Geräte einstellen, da der Emulator auf demselben Gerät läuft sollte es Port 8080, der Localhost, und als Adresse 127.0.0.1 sein. Die gleichen Einstellungen sollte man beim virtuellen Gerät auch über das Android Studio einstellen. Dafür wählt man das „…“ Symbol am Gerät aus, wählt Einstellungen aus und navigiert zum Reiter Proxy. Dort gibt man unter Manual Proxy Configuration die gleichen Einstellungen aus. Aktiviert man nun den Burp Suite Proxy und wird beim virtuellen Gerät eine Aktion mit Netzwerkverkehr durchgeführt, sollte dies in der HTTP History angezeigt werden.

Es ist jedoch noch nicht möglich HTTPS Traffic abzufangen, so können zum Beispiel bei der Welt News App die neusten Nachrichten abgeglichen werden, aber ein Login in seinen Account ist nicht möglich. Damit dies auch möglich ist, muss das Burp CA Certificate auf dem virtuellen Gerät installiert werden. Um dies zu tun wählt man in der Burp Suite den Reiter Proxy aus und dort dann den Reiter Options. Dort findet sich dann ein Knopf mit „Import/Export CA Certificate“. Mit dessen Hilfe exportiert man dann das Zertifikat im .cer Format. Um dieses auf das Gerät zu bewegen nutzt man wie zuvor die Android Debug Bridge. Mit ./adb push <Pfad des Zertifikates> <Pfad auf dem Gerät> kopiert man das Zertifikat auf das Gerät, als Pfad bietet sich /sdcard an, da man von der SD Karte einfacher das Zertifikat installieren kann. Um zu überprüfen ob der push Befehl funktioniert hat, kann man ./adb shell nutzen und dann mit dem cd Befehl zur SD Karte des Gerätes navigieren.

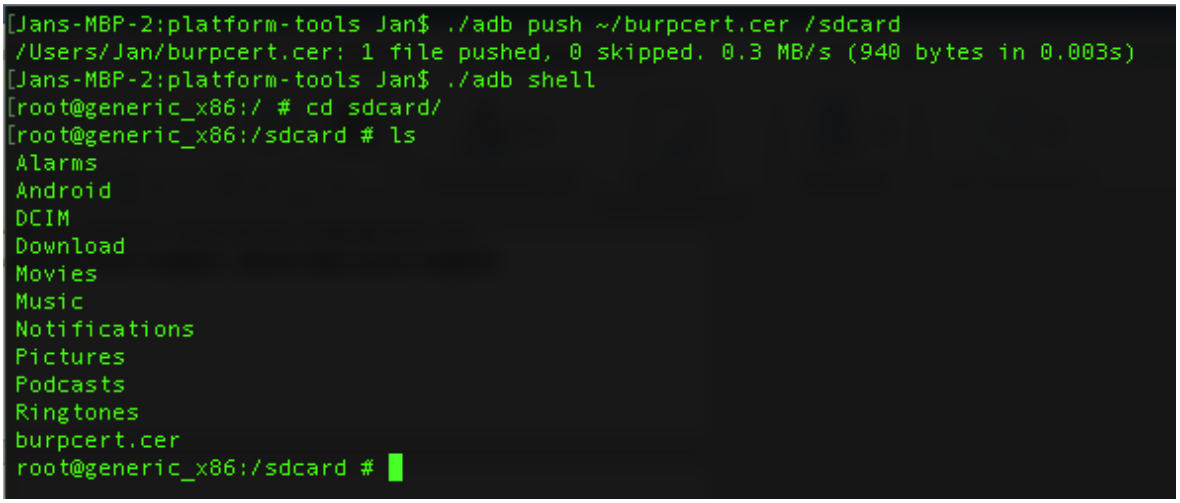



**Abbildung 9: Nutzung der ADB um das Burp CA Certificate auf dem virtuellen Gerät zu installieren, eigene Abbildung.**

Das Zertifikat muss nun noch installiert werden. Dies funktioniert je nach Gerät anders. Beim Nexus 6 wählt man folgendes aus, Settings -> Security ->I nstall from sdcard -> Internal Storage -> Burp Zertifikat -> Name für das Zeritifkat auswählen. Zum Überprüfen wechselt man zu trusted credentials- > User und dort sollte sich dann das Zertifikat von PortSwigger finden. Für das Pixel 2 navigiert man über Settings, Security, Encryption & Credentials zu Install from SD Card. Hier kann man nach dem Zeritfikat suchen und bestätigt die Installation. Zum Überprüfen wählt man auch Trusted Credentials gefolgt von User aus und sollte dort dann auch das Zertifikat von PortSwigger finden. Hat man diese Schritte durchgeführt kann man nun auch HTTPS Verbindungen abfangen.

### 7.1.2 Android 10.0

Mit Android 7.0 gab es einige Änderungen an den von Google implementierten Sicherheitsprotokollen. Von Nutzern installierte Zertifikate werden von Apps mit einem API Level von 24 und höher nicht mehr als sicher angesehen und somit nicht mehr akzeptiert. Die einzige Ausnahme ist, wenn die Applikation selbst das bestimmte Zertifikat erlaubt. Dieses Problem lässt sich leicht bei physischen Geräten umgehen, für virtuelle Geräte gibt es von PortSwigger keine offizielle Lösung.

Um das Problem zu umgehen benötigt man mehrere zusätzliche Vorbereitungen und Schritte im Vergleich zu Android 6.0. So sollte Magisk auf dem Gerät installiert sein. Dies ist ein Source Tool für Android Geräte mit dem man unter anderem Veränderungen stärkeren Ausmaßes vornehmen kann, wie unter anderem das verschaffen von Root Access, Read Only Partitionen verändern und Boot Scripte ausführen. Dieses Tool sollte man dafür nutzen „Magisk Trust User Certs“ zu installieren.

Zu Anfang installiert man das Burp CA Certificate wie bei Android 6.0 auf dem Gerät. Android speichert und ändert den Namen des Zertifikates in einen Hash mit dem Zusatz .0 am Ende. Um den exakten Namen herauszufinden navigiert man mit der Android Debug Shell, ./adb shell, durch das Gerät. Im Pfad /data/misc/user/0/cacerts-added/ sollten mit ls alle installierten Zertifikate angezeigt werden. Da dies in meinem Fall nur das Burp CA Certificate ist, lässt sich der Name einfach finden. Mit dem Magisk Trust User Certs und dem gehashten Namen für das Zertifikat, kann man dieses mit Root Access als sicheres Zertifikat hinzufügen und es wird dann auch genauso wie bei der 6.0 Variante als sicheres Zertifikat in den Einstellungen angezeigt.

### 7.1.3 Einrichten des Burp Suite Proxy für iOS 14.0

Da ein physisches iPhone vorhanden ist, ist der Prozess, um ein iPhone mit der Burp Suite zu nutzen deutlich einfacher als im Vergleich zu Android. Zuerst muss man in der Burp Suite unter dem Reiter Proxy auf den Reiter Einstellungen gehen und dort den Proxy Listener entsprechend einstellen. Dort fügt man entweder einen neuen hinzu oder bearbeitet einen der bereits vorhandenen Proxy. Man stellt einen beliebigen Port ein, der nicht in Benutzung ist, in meinem Fall habe ich zum Beispiel Port 8082 verwendet. Für die Adresse sollte man am einfachsten auf „all Interfaces“ wechseln, es kann aber auch die spezifische Adresse des iPhones verwendet werden (Siehe Anlage 2). Nun muss noch das iPhone entsprechend auf diesen Proxy eingestellt werden. Dafür navigiert man über Einstellungen zu Wlan und verbindet sich dort mit dem entsprechenden Wlan, welches auch vom Computer genutzt wird. Es wird das i-Symbol neben dem Namen ausgewählt und dort findet sich ein Feld HTTP-Proxy. Dieses muss ausgewählt werden und dort dann Manuell aktiviert werden. Der Server ist die IP-Adresse des Computers mit der Burp Suite, der Port ist, der der beim Proxy Listener ausgewählt wurde. Aktiviert man nun den Intercept in der Burp Suite, sollten bereits die Pakete auftauchen, wenn man entsprechende Funktionen von Apps nutzt, jedoch wie vorher lassen sich so erstmal nur HTTP Verbindungen anzeigen und der Versuch eine HTTPS Verbindung aufzubauen wird fehlschlagen. Es muss, wie auch bei Android, noch das Burp CA Certificate installiert werden. Bei einem physischen Gerät ist es jedoch deutlich einfacher. Bei korrekten Proxy Einstellungen und aktiviertem Intercept muss über den Browser des Gerätes nur die Adresse http://burpsuite aufgerufen werden. Dort kann man oben rechts CA Certificate anklicken und den Download des Profils bestätigen. Nun muss das Ca Certificate noch installiert werden. Dafür wählt man über Einstellungen und Allgemein Profile aus und installiert das Portswigger Ca und aktiviert das Profil (Siehe Anlage 5). Für neuere iOS Versionen muss nun noch ein zusätzlicher Schritt durchgeführt werden, da für das Zertifikat das sogenannte „volles Vertrauen für Root Zertifikate“ aktiviert werden muss. Dafür navigiert man über Einstellungen, Allgemein, Info zu Zertifikatvertrauenseinstelllung und legt dort den Schalter für das PortSwigger CA um. Nun kann auch jeglicher HTTPS Traffic über die Burp Suite mitgelesen werden.

## 7.2 Untersuchung der mobilen Applikationen

Im folgenden Abschnitt wird schließlich der Netzwerkverkehr der vorgestellten Applikationen auf die benannten Schwächen mit der Burp Suite untersucht.

### 7.2.1 Netzwerkverkehr der Sparkassen App

Die Sparkassen App scheint gegen die von der Burp Suite genutzte Form des Man in the Middle Angriffes vollkommen sicher zu sein. Die App lässt sich normal nutzen, sowohl auf Android als auch iOS, jedoch löst jede Anfrage oder der Abgleich von Daten den immer gleichen Fehler aus. Es scheint nicht so, dass die Verbindung unterbrochen, sondern dass erst gar keine Verbindung aufgebaut wird, da sich nicht ein Paket mit Hilfe der Burp Suite anzeigen lässt. Sofern der Proxy deaktiviert wird, funktioniert die Sparkassen App wieder vollkommen normal und alle Funktionen lassen sich durchführen.

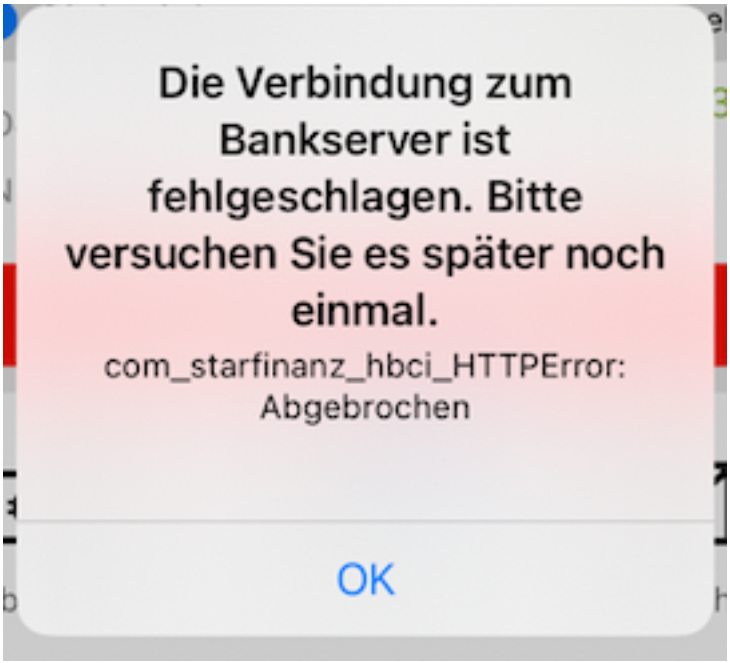



**Abbildung 10: HTTP Error der Sparkassen App der Auftritt, wenn ein Proxy genutzt wird, eigene Abbildung**

Die Frage an dieser Stelle ist, ob die Sparkassen Applikationen von Grund auf jegliche zusätzliche Zertifikate ablehnen oder ob es spezifische wenige Zertifikate betrifft, so zum Beispiel das Burp CA Certificat. Auch kann mit dieser Methode keinerlei Aussage darüber getroffen werden, inwieweit die Pakete der Sparkassen sicher sind oder nicht, in dieser Form nicht einsehen kann. Da es jedoch eine Banking App ist und diese, auch was Sicherheit angeht, Bestnoten erhält, kann man davon ausgehen, dass die Sparkassen App gegen das Mitschneiden des Datenverkehrs robust gestaltet ist und dort wenig bis keine Schwächen zeigt. Interessanter weise lässt sich die App auf dem iPhone noch ohne Probleme nutzen, wenn ein oder mehrere Virtual Private Networks (VPN) dazwischengeschaltet werden, somit scheint es erstmal kein Problem zu sein, dass kein direkter Kontakt zwischen App und Sparkassen besteht. Somit werden von der Applikation wohl Proxys beim Verbindungsaufbau erkannt und das Paket wird sofort verworfen. Fraglich ist warum, falls dieser tatsächlich erkannt wird, der Nutzer nicht davor gewarnt wird und es nur eine Nachricht zu einem HTTP Error gibt und man es doch bitte später nochmal probieren sollte. Ein Hinweis darauf, dass gegebenenfalls das mobile Gerät kompromittiert ist, kann sicherlich nicht schaden, auch wenn es in den meisten Fällen sicherlich ein anderes Problem ist.

### 7.2.2 Netzwerkverkehr der WELT News App

Die Welt App zeigt gleich schon zu Anfang eine große Schwäche. Ich war bereits schon bei der App angemeldet, habe mich jedoch abgemeldet und eine neue Registrierung durchgeführt. Für die Registrierung sind die Anrede, der Name, der Nachname, das Land, die Emailadresse und die Zweifache Eingabe eines Passwortes nötig. Vorname, Name und Emailadresse werden nicht im Vorfeld auf Gültigkeit überprüft, so können auch vorerst nicht vorhandene Emailadressen genutzt werden. Der gegebenenfalls viel schlimmere Fehler ist jedoch, dass sofern man die Registrierung abschließen möchte, die eingegebenen Daten im Klartext Format übertragen werden. Im abgefangenen Paket kann man erkennen, dass für die Registrierung HTTP/1.1 verwendet wird, also kein gesichertes HTTPS und das alle angegeben Informationen einfach ausgelesen werden können, sofern ein Man in the Middle Angriff während der Registrierung abläuft. Man kann eindeutig in Abbildung 11 sehen, dass für die Registrierung der Name „Test Test“, die E-Mail-Adresse bachelor.bo.test@gmail.com und das Passwort „Test1234“ verwendet wurde. Es ist noch anzumerken, dass die Welt App die Registrierungen und Anmeldungen nicht selber durchführt, sondern hierfür einen Dienst Namens „mypass.de“ verwenden. Dieser verwaltet dem Anschein nach die vertraulichen Daten für mehrere verschiedene Online Angebote der Axel Springer AG.


**Abbildung 11: Übertragung der Daten zur Registrierung bei der Welt App, im unteren Teil stehen die sensiblen Informationen im Klartext, eigene Abbildung**

Dies kann in der heutigen Zeit zu fatalen Schäden, da besonders E-Mail-Adresse und Passwort so auch für andere Dinge genutzt werden können. Es wird empfohlen für jede Webseite und Applikation ein anderes Passwort oder zu mindestens eine Variation des Passwortes zu nutzen, jedoch geht man Umfragen nach davon aus, dass etwa 6 von 10 Personen das gleiche Passwort für mehrere Dinge nutzen und sogar eine von 10 Personen das gleiche Passwort für alle Anwendungen nutzt.[25] In Kombination mit der E-Mail-Adresse, bei der auch nahezu immer die gleiche verwendet, kann man sich so sehr einfach Zugang zu den verschiedensten Anwendungen verschaffen, wie zum Beispiel Zugriff zum Amazon Konto. Somit könnte in dieser Form direkt zu Anfang schon eine Traffic Analysis Form des Man in the Middle Angriffes in vielen Fällen zu teilweise massiven Schäden führen. Es sollte auch darauf hingewiesen werden, dass der Mypass Dienst explizit herausstellt, dass eine Registrierung bei einem Dienst auch dafür sorgt, dass man bei allen anderen registriert ist. Somit können so auch die hier abgegriffenen Kontodaten für weitere Dinge verwendet werden. So ist es zum Beispiel möglich über Rechnung etwas im Bild Shop zu kaufen, nachdem man sich ohne Adresse bei der Welt App registriert hat.


**Abbildung 12: Angebote die mit denselben Zugangsdaten über myPass genutzt werden können, https://www.mypass.de aufgerufen am 10.11.2020, 14:30**

Das gleiche Problem tritt auch bei der normalen Anmeldung auf. Das Passwort wird bei der Eingabe, wie bei so ziemlich jeder anderen mobilen oder Webanwendung auch, nur als Pünktchen dargestellt. Wird das Paket jedoch abgefangen, sind sowohl der Anmeldename als auch das Passwort in Klartext dargestellt. Man kann in Abbildung 12 eindeutig den Nutzernamen bachelor.bo.test@gmail.com erkennen sowie das Passwort „Test1234“.

```
POST /aao/app/login;jaesaionid=A384D1C864BDCD259FA7E3156B6048E0?aecurity=low&aervice=
https%3A%2F%2Faecure.mypaaa.de%2Fappconnect-3.0%2Fahop%2FWONP%2Fhtml%2Faucceaa&deviceClaaa=amartphone&oaType=ioa
&remId=16204775555114141611&traceeid=ECFU31GJG8CN8VHB27PIP9LLUUM1SJ0A HTTP/1.1
Host: aecure.mypaaa.de
Content-Type: application/x-www-form-urlencoded
Origin: https://aecure.mypaaa.de
Accept-Encoding: gzip, deflate
Cookie: wt_trackId=370137189758193; JSESSIONID=A384D1C864BDCD259FA7E3156B6048E0; remid=16204775555114141611;
NSC_MCWT_QDQ-QSPE-TTP=ffffffff09e9155b45525d5f4f58455e445a4a423660; kameleoonVisitorCode=_ja_5ncfltlbkawgmoz9
Connection: close
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
User-Agent: Mozilla/5.0 (iPhone; CPU iPhone OS 14_2 like Mac OS X) AppleWebKit/605.1.15 (KHTML, like Gecko)
Mobile/15E148
Referer:
https://aecure.mypaaa.de/aao/app/login?aecurity=low&aervice=https%3A%2F%2Faecure.mypaaa.de%2Fappconnect-3.0%2Fah
op%2FWONP%2Fhtml%2Faucceaa&deviceClaaa=amartphone&oaType=ioa&remId=16204775555114141611&traceeid=ECFU31GJG8CN8VHB
27PIP9LLUUM1SJ0A
Content-Length: 80
Accept-Language: de-de

username=bachelor.bo.test%40gmail.com&paaaword=Teat1234&_eventId_aubmit=&lt=elal
```

**Abbildung 13: Paket für die Anmeldung der Welt News App, Username und Password in Klartext sichtbar, eigene Abbildung**

Angesichts solcher gravierenden Fehler mutet es schon ein wenig merkwürdig an, dass der 1. Satz in der Datenschutzerklärung WELT Digital „WELT nimmt Datenschutz ernst…“ lautet. Besonders da hier ein System genutzt wird, dass es erlaubt sich auf mehreren verschiedenen Diensten mit ein und demselben Nutzernamen und Passwort anzumelden und dies dann so wenig geschützt abgegriffen werden kann ist sicherlich das Gegenteil von erhöhtem Datenschutz. Aus Sicht des Unternehmens ist es wahrscheinlich mit der Anzahl der Anwendungen und der damit hohen Anzahl an Anmeldungen und Registrierungen nicht verwunderlich, dass nur HTTP für die Anmeldung genutzt wird. Wie schon im Kapitel dazu erwähnt, erfordert HTTPS einen deutlich höheren Rechenaufwand und ist damit langsamer als HTTP, was besonders bei immer höherer Anzahl an Anfragen deutlich wird. Jedoch kann Geschwindigkeit kein einzelner Grund sein, da hier sogar noch das veraltete und langsamere HTTP/1.1 statt HTTP/2 verwendet wird. So ist es Aufgrund des 1.1 Standards sogar vergleichsweise einfach an die Anmeldedaten zu kommen, sogar wenn man die Anmeldung verpasst hat. Fast jedes Paket enthält den ungeschützten Cookie mit der Session ID, was einen Angriff nach dem Session Hijacking erlaubt, welches dann dafür genutzt werden kann den Nutzer abzumelden und somit eine neue Anmeldung zu forcieren, die dann abgegriffen werden kann.

Meiner Meinung nach zeigt die Welt News App schwerwiegende Schwachstellen, die sich Aufgrund der Nutzung des Mypass Dienstes auch in anderen Applikationen der Axel Springer AG finden, sowohl als mobile als auch als Webapplikationen. Im Vergleich zu der Sparkassen App, welche nicht einmal das Abfangen eines einzigen Paketes erlaubte, ist die Welt App für einen fähigen Angreifer über das Man in the Middle Prinzip fast wie ein offenes Buch. Mit dem richtigen Timing ist nichts mehr nötig als das Mitlesen von Paketen um den Zugriff auf mehrere verschiedene Apps zu erhalten. Das in der heutigen Zeit für einen Prozess der E-Commerce enthält noch der HTTP/1.1 Standard verwendet wird, ist kaum zu erklären und hat mit dem generellen Verständnis von Datenschutz nichts zu tun.

### 7.2.3 Netzwerkverkehr der Amazon App

Die Amazon App Test beginnt genau wie der Welt App Test. Da ich bereits über ein Konto verfüge habe ich mich abgemeldet und werde einen neuen Nutzer registrieren. Der Ablauf zeigt die nahezu gleichen Ergebnisse wie bei der Welt Applikation. Die Amazon App nutzt fast ausschließlich den HTTP/1.1 Standard und verschlüsselt bei der Registrierung anscheinend keinerlei Daten.

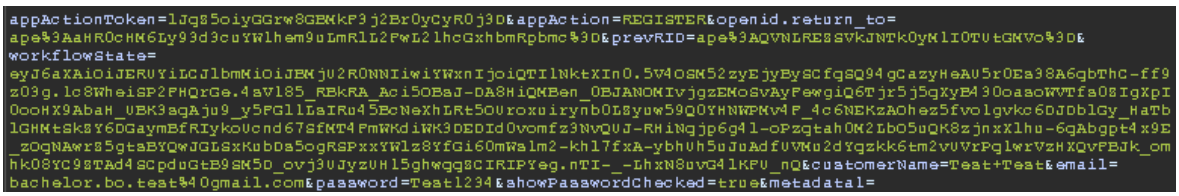



**Abbildung 14: Ausschnitt (Inhalt des gesamten Paketes zu groß) des Paketes für die Registrierung eines Amazon Konto, Accountdaten in Klartext zu sehen, eigene Abbildung**

Ebenso wie bei der Welt App werden alle 3 zur Registrierung genutzten Daten unverschlüsselt angezeigt, in diesem Fall der Name „Test Test“, die E-Mail-Adresse bachelor.bo.test@gmail.com und das Passwort „Test1234“. Das diese Daten, die dann auch für die Registrierung eines Amazon Konto zu Test Zwecken genutzt wurden, offenliegen kann die gleichen Folgen haben, wie bei der Welt App, gegebenenfalls sogar noch schädlicher, da ein Amazon Konto für weitaus mehr Dinge genutzt werden kann, jedoch hat auch Amazon selbst gewisse Schranken für den Fall eingeführt, dass einer Person das Konto entwendet wird. So ist es zum Beispiel nicht möglich etwas über die gespeicherten Zahlungsmethoden zu kaufen und dies an eine neue Adresse zu senden. In diesem Fall muss die Zahlungsmethode neu eingegeben werden, was bei normalen Bankkonten oder Kreditkarten nicht trivial ist und eine Bestellung auf Rechnung ist meist nicht verfügbar. Jedoch bietet Amazon auch die Möglichkeit mit anderen Online-Bezahldiensten zu zahlen, zum Beispiel Paypal oder Klarna. Hier könnte wieder das Prinzip in Frage kommen, dass eine Vielzahl von Personen die gleichen Daten zur Anmeldung für mehrere verschiedene Dienste nutzen und somit ein Angreifer im Zweifel diesen nutzen könnte. Oft ist es auch der Fall, dass sobald Angreifer an die Daten eines Händlers kommen, dessen Konto mit einer horrenden Zahl an gefälschten Angeboten zu Dumping Preisen fluten und die Auszahlungen auf ein Dummy Konto leiten. So entstand in der Vergangenheit schon ein gewisser Schaden, da anscheinend obwohl der Verlust des Kontos schnell bemerkt wurde, der Amazon Kundenservice auf solche Fälle nicht gut vorbereitet schient und es teilweise bis zu einer Woche dauerte bis das Amazon Konto gesperrt wurde und keine neuen Angebote mehr eingestellt werden konnten.

Im nächsten Schritt teste ich was in Paketen bei einer Anmeldung sichtbar ist. Hier zeigt sich ein entscheidender Unterschied zur Welt App. Zwar ist die E-Mail-Adresse, die ein Teil der Anmeldung ist, immer noch sichtbar, dass Passwort hingegen wurde verschlüsselt.

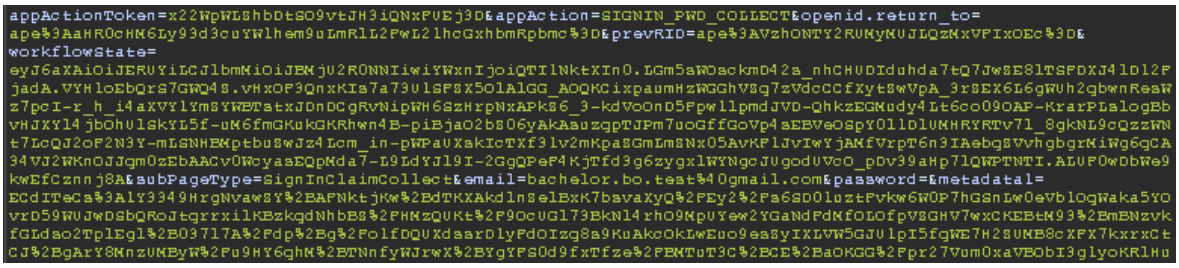

appActionToken=x22WpWL8hbDtSO9vtJH3iQNxFUEj3D&appAction=SIGNIN_PWD_COLLECT&openid.return_to=
ape%3AaHR0cHM6Ly93d3cuYWlhem9uLmRlL2PwL2lhcGxhbmRpbmc%3D&prevRID=ape%3AVzhONTY2RUMyMUJLQzMxVPIxOEc%3D&
workflowState=
eyJ6aXAiOiJERUYiLCJlbmMiOiJBMjU2R0NNIiwiYWxnIjoiQTIlNktXIn0.LGm5aWOackmD42a_nhCHUDIduhda7tQ7JwSE8lTSFDXJ4lDl2F
jadA.VYHloEbQrS7GWQ48.vHxOF3QnxKIa7a73UlSFSX5OlAlGG_AOQKCixpaumHzWGGhV8q7zVdcCCfXytSwVpA_3rSEX6L6gWUh2gbwnReaW
z7pcI-r_h_i4aXVYlYm8YWBTatxJDnDCgRvNipWH68zHrpNxAPk86_3-kdVoOnD5Fpwllpmd JVD-QhkzEGMudy4Lt6coO9OAP-KrarPLalogBb
vHJXYl4jbOhUlSkYL5f-uM6fmGKukGKRhwn4B-piBjaO2b806yAkAauzgpTJPm7uoGffGoVp4aEBVeOSpYOllDlUNHRYRTv7l_8gkNL9cQzzWN
t7LcQJ2oP2N3Y-mLSNHEMptbu8wJz4Lcm_in-pWPaUXakIcTXf3lv2mKpa8GmLm8Nx05AvKPlJvIwYjAMfVrpT6n3IAebq8Vvhgbgr MiWg6gCA
34VJ2WKnOJJgm0zEbAACv0WcyaaEQpMda7-L9IdYJl9I-2GgQPeP4KjTfd3g6zygxlWYNgcJUgodUVcO_pDv39aHp7lQWPTNTI.ALUP0wDbWe9
kwEfCznnj8A&subPageType=SignInClaimCollect&email=bachelor.bo.test%40gmail.com&password=&metadata1=
ECdITeCa%3Aly3349HrgNvaw8Y%2BAFNktjKw%2BdTKXAkdln8elBxK7bavaXyQ%2PEy2%2Pa6SDOluztPvkw6W0P7hGSnLw0eVblOgWaka5YO
vrD59WUJwDSbQRoJtqrrxilKBzkgdNhbE8%2PHMzQUKt%2P9OcUGl73BkNl4rhO9MpUYew2YGaNdPdMfOLOfpV8GHV7wxCKEBtM93%2BmBNzvk
fGLdso2TplEgl%2BO37l7A%2Pdp%2Bg%2PolfDQUXdaarDlyPdOIzq8a9KuAkcOkLwEuo9ea8yIXLVW5GJUlpI5fqWE7H28UNB8cXPX7kxrxCt
CJ%2BgArY8MnzUMByW%2Pu9HY6qhM%2BTNnfyWJrwX%2BYgYFSOd9fxTfze%2PEMTuT3C%2BCE%2BaOKGG%2Ppr27Vum0xaVBObI3glyoKRlHu

**Abbildung 15: Ausschnitt eines Paketes bei der Anmeldung der Amazon App, E-Mail-Adresse im Klartext, Passwort verschlüsselt, eigene Abbildung**

Die Verschlüsselung des Passworts scheint auch nicht sehr trivial zu sein, nach mehreren Tests mit verschiedenen E-Mail-Adressen und Passwörtern unterschiedlicher Länge zeigt sich, dass das verschlüsselte Passwort immer die gleiche Länge hat. Es ist anzunehmen, dass hier ein Hash inklusive Padding zum Einsatz kommt, was es je nach der genutzten kryptographischen Hashfunktion nahezu unmöglich macht das originale Passwort wiederherzustellen. Dies ist im Vergleich zu der Welt App fast schon ein Quanten Sprung was die Sicherheit angeht. Fraglich ist hier jedoch auch, warum dies nicht auch für die E-Mail-Adresse genutzt wird. Das diese wie schon bei der Registrierung in Klartext vorliegt, kann zum Beispiel auch zum Verlust des Kontos führen, falls man während eines Angriffes Zugriff zum E-Mail-Konto erhält und beides dazu nutzt, das Passwort des Amazon Kontos zurückzusetzen.

Sicher scheinen hingegen das Aufrufen und gegebenenfalls das Verändern von Zahlungsinformationen und Lieferadressen zu sein. Zwar lässt sich in Paketen sehen, dass diese Informationen aufgerufen werden, die Informationen selbst sind jedoch wie die Passwörter verschlüsselt und in den Paketen nicht als Klartext sichtbar. Dem Anschein nach wird hier jedoch tatsächlich nur das gehasht, was auch tatsächlich vorhanden ist, da sich bei Tests mit unterschiedlichen Längen bei den Adressen auch unterschiedlich längen in der Verschlüsselung zeigen. Da jedoch die Adresse aus unterschiedlichen Komponenten besteht ist nicht klar ob diese einzeln verschlüsselt werden und dann zusammengefügt werden oder ob diese erst zusammengefügt werden und dann verschlüsselt.

Anzumerken ist noch, dass die Nutzung von HTTP/1.1 ähnlich wie bei der Welt App dazu führen kann, dass ein Angriff nach dem Muster des Session Hijacking stattfindet. Die Session IDs liegen ungeschützt vor und ermöglichen daher einem mit dieser Methode versierten Angreifer ein leichtes Ziel.

Abschließend kann man sagen, dass die Amazon App eine deutlich größere Sicherheit bietet als die Welt App. Beide Applikationen nutzen jedoch für den Großteil des Datenverkehrs den HTTP/1.1 Standard, die Amazon App setzt aber durch das Verschlüsseln von Passwörtern oder sensiblen Zahlungsinformationen ein höheres Maß an Sicherheit voraus. Jedoch wird hier auch bei der Registrierung der Account einer hohen Gefahr ausgesetzt, da alle Daten unverschlüsselt vorliegen. Auch ist die Amazon App im Vergleich zur Sparkassen App deutlich anfälliger, da bei der letzteren die höheren Sicherheitsstandards dafür sorgen, dass das Burp CA Certificate nicht akzeptiert wird.

### 7.2.4 Netzwerkverkehr der Corona Warn App

Die Corona Warn App scheint ähnlich wie die Sparkassen App gegen diese Form von Angriffen vollkommen gefeilt zu sein. Es gibt keinerlei Registrierung mit sensiblen Daten, diese könnten dann auch nicht ähnlich wie bei der Amazon App oder der Welt App abgegriffen werden. Das System zum Austausch von Daten mit anderen Nutzern läuft über Bluetooth ab und kann daher nicht über den Proxy der Burp Suite abgefangen und analysiert werden. Diese zufälligen Bluetooth IDs werden auch nur für begrenzte Zeit gespeichert und scheinen keine Möglichkeit zu Rückschlüssen auf Identität oder Standort zuzulassen. Zudem nutzt die App ein dezentrales System, es werden zu keinem Zeitpunkt Daten, die Standort, Infektion und Identität betreffen, von Servern geholt oder an diese gesendet. Somit können diese auch mit einem Abgriff des Netzwerkverkehres nicht erbeutet werden. Es gibt anscheinend nur einen Angriffspunkt. Die App erlaubt es, durch Eingabe einer TAN oder eines QR-Codes, ein positives Covid-19 Testergebnis in die App zu speichern und dies mit einer entsprechenden zentralen Stelle abzugleichen, ob die TAN oder der QR-Code korrekt sind. Jedoch scheint die App hier ähnlich wie die Sparkassen App zu erkennen, das keine direkte HTTPS Verbindung besteht und somit wird jeglicher Verbindungsaufbau sofort unterbrochen (siehe Unterschied Abbildung 16 und Anlage 6). Die einzigen Pakete, die korrekt abgefangen werden können, sind diejenigen die man erhält, wenn man auf den Reiter „Häufige Fragen“ wechselt, dies ist dann aber auch nur eine Kopie der Webseite von „bundesregierung.de“. Dort finden sich meiner Ansicht nach keinerlei Interessante Informationen.

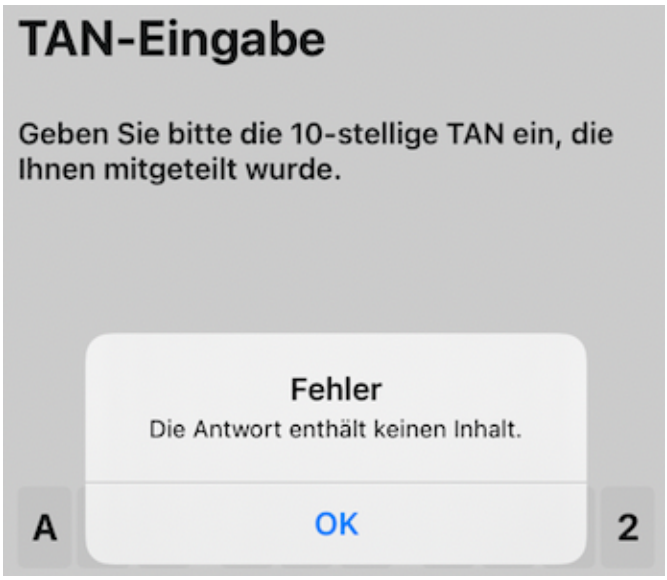



**Abbildung 16: HTTPS Verbindung kann zum Abgleichen der TAN der Corona Warn App nicht aufgebaut werden, eigene Abbildung**

Die Corona Warn App ist meiner Ansicht nach bis zu einem gewissen Grad vollkommen sicher gegen Angriffe des Man in the Middle Bereiches. Der grundlegende Aufbau der App ist durch dezentrales System sowie die Nutzung von Bluetooth zur Datenübertragung zwischen den einzelnen Corona Warn App Instanzen wirkt dem Mitschneiden oder Abgreifen von Daten komplett entgegen. Auch werden zur Nutzung der App keinerlei Registrierung mit sensiblen und persönlichen Informationen vorausgesetzt. Zugleich gibt es bestimmte Daten die über normalen Netzwerkverkehr abgeglichen werden müssen, in diesen Instanzen wird nach meinem Verständnis das Burp Suite CA Zertifikat wie bei der Sparkassen App nicht akzeptiert. Aber man sollte auch anmerken, dass man keinen Einblick darin hat, ob diese Pakete sicher sind. Es könnte theoretisch sein, dass die Pakete unsicher sind, es sollte aber auch in diesem Fall unwahrscheinlich sein, dass sich dort Angriffsziele finden lassen

# 8 Fazit

Ziel dieser Arbeit war die Untersuchung des Netzwerkverkehrs mit Hilfe der Burp Suite auf sicherheitsrelevante Aspekte und das Aufzeigen der Grenzen dieses Werkzeuges. Zu diesem Zwecke wurden 4 in Deutschland sehr beliebte Applikationen ausgewählt und deren Netzwerkverkehr in der Burp Suite näher betrachtet. 2 der 4 Applikationen zeigten sicherheitsrelevanten Schwächen von signifikantem Ausmaß und die anderen 2 Applikationen zeigten, dass bei einem hohen Sicherheitsstandard grundlegend keine Aussage darüber getroffen werden kann, ob die einzelnen Pakete im Netzwerkverkehr sicher sind oder nicht.

## 8.1 Zusammenfassung

Die Burp Suite ist momentan ein mächtiges Werkzeug für die Analyse von Netzwerkverkehr. In dieser Arbeit habe ich fast ausschließlich nur einen Teil dieses Werkzeugkastens genutzt und konnte dennoch, nach meiner Ansicht nach, gravierende Sicherheitslücken in mobilen Applikationen aufdecken, die Millionen an Downloads in Deutschland haben. Es scheint schon fast fahrlässig, wenn man einen Vergleich zwischen den Sicherheitsstandards von auf der einen Seite Sparkassen und Corona Warn App und auf der anderen Seite WELT News und Amazon App zieht, und sich anschaut, wie trivial bei der WELT News App der Zugriff auf sensible Daten gelingt. Hier zeigt sich, dass selbst Applikationen, die schon vor langer Zeit veröffentlicht wurden und seitdem auch immer regelmäßig Updates erhalten, dennoch oft große Sicherheitslücken haben können. Diese Schwachstellen werden dann auch leider meist erst entdeckt, wenn es zu spät ist und ein Angriff mit massiver Auswirkung bereits stattgefunden hat. Daher erweist sich die Burp Suite sogar schon in der kostenlosen Community Edition als sehr wertvoll. Es war mir möglich bei 2 von 4 bekannten Applikationen zu zeigen, dass Angriffe über den Netzwerkverkehr erfolgreich durchgeführt werden können.

## 8.2 Kritische Betrachtung

Die Burp Suite ist in ihrer Nutzung sehr intuitiv und hat mit einer breit aufgestellten Dokumentation die Antwort für die meisten Probleme. Eine aktive Community entwickelt und stellt anderen Nutzern kostenlos Erweiterungen zu Verfügung, die viele Prozesse erleichtern. Die Community Edition bietet kostenlos alle nötigen Tools für die Arbeit an den Grundlagen und die Pro Version hat, was die Rezensionen angeht, ein ausgezeichnetes Preis/Leistung Verhältnis sowie durch zusätzliche Tools alles was man für die Analyse der Cybersicherheit von Programmen, mobilen Applikationen und Webanwendungen braucht.

Die Community Edition hat meiner Ansicht nach 2 entscheidende Schwächen. Die erste ist, dass, wenn man das Programm schließt, alle Einstellungen und Individualisierungen beim nächsten Start erneut vorgenommen werden müssen. Dies beinhaltet unter anderem die Einstellungen an den Proxys, die jedes Mal neu eingerichtet werden müssen, um mit den vorhandenen Geräten zu funktionieren. Die 2. große Schwäche betrifft die Untersuchung des Netzwerkverkehrs. Hier ist es fraglich wie lange in dieser Form noch eine Nutzung des Proxys zum Mitschneiden des Datenverkehrs möglich ist. Bei 2 der 4 Applikationen war es bereits aufgrund von Sicherheitsstandards nicht mehr möglich den Netzwerkverkehr mitzuschneiden. Somit kann eigentlich auch keine Aussage darüber getroffen werden inwieweit die Pakete sicher sind. So ist es mittlerweile auch schon mit deutlichen Problemen verbunden das Burp CA Certificate auf eine virtuelle Android Version 7 oder höher zu bringen. Man kann davon ausgehen, dass dies in Zukunft wahrscheinlich mit noch mehr Hürden versehen ist oder eventuell gar nicht mehr in der jetzigen Variante möglich ist. Die Frage, wie lange noch ein solches Zertifikat auf Apple Geräten akzeptiert wird, die eigentlich für einen höheren Sicherheitsstandard als Android bekannt sind, muss auch gestellt werden.

## 8.3 Ausblick

Abschließend lässt sich sagen, dass der Datenschutz auf mobilen Applikationen in Zukunft eine immer größere Rolle spielen wird. Heute ist es mir schon möglich eine Vielzahl an Aktionen des täglichen Lebens darüber zu steuern, vom Banking, dem Einkaufen, Kommunikation, das Bezahlen an der Einkaufskasse, dem Öffnen und Schließen von Haustüren, das Starten des Autos. Dies wird Zukunft wahrscheinlich noch mehr zunehmen und das führt zwingend zu der Frage, wie sicher sind die Daten, die ich dort überall hin und her gesendet werden. Programme wie die Burp Suite sind hier sicherlich zwingend notwendig, um Schwachstellen zu finden und diese frühzeitig zu schließen, um im gesamten für eine höhere Sicherheit zu sorgen.

# Literatur

[1] Bojjagani S., Sastry, V.N.: A Threat Model for Vulnerability Assessment and Penetration Testing of Android and iOS Mobile Banking Apps, University of Hyderabad, India, Dezember 2017

[2] Mistry, Dahiya: A Comprehensive Study of Vulnerability Assessment of Existing Banking Apps, Gujarat Forensic Sciences University, India, April 2018

[3] Qin, Zhang: Vulnerability Detection on Android Apps-Inspired by Case Stuidy on Vulnerabitlity related with Web Functions, Beijing University of Posts and Telecommunications, China, May 2020

[4] Cai, Chen: AppCracker: Widespread Vulnerabilities in User and Session Auhentication in Mobile Apps, Shanghai Tech University, China, November 2014

[5] Mendoza, Gu: Mobile Application Web API Reconnaisance: Web-to-Mobile Inconsistencies & Vulnerabilites, Texas A&M University, May 2018

[6] PortSwigger: General Informations. https://portswigger.net/about, verfügbar am 01.11.2020, 09:35

[7] Crunchbase: Organization PortSwigger. https://www.crunchbase.com/organization/portswigger, verfügbar am 01.11.2020, 09:35

[8] PortSwigger: Documentation - Burp Suite tools. https://portswigger.net/burp/documentation/desktop/tools verfügbar am 01.11.2020, 10:50

[9] Computerweekly: Zertifizierungsstelle - Certificate Authority. https://bit.ly/2JeAwjh verfügbar am 02.11.2020, 09:05

[10] PortSwigger: Installing Burp's CA certificate. https://bit.ly/3qft4Vy verfügbar am 02.11.2020, 09:15

[11] Android Studio: Meet Android Studio. https://developer.android.com/studio/intro verfügbar am 02.11.2020, 09:45

[12] Wikipedia: Android Studio. https://de.wikipedia.org/wiki/Android_Studio verfügbar am 02.11.2020, 09:45

[13] Droidwiki: Android Debug Bridge.
https://www.droidwiki.org/wiki/Android_Debug_Bridge verfügbar am 02.11.2020, 10:00

[14] Android Studio: Android Debug Bridge.
https://developer.android.com/studio/command-line/adb verfügbar am 02.11.2020, 10:00

[15] Mozilla: HTTP.
https://developer.mozilla.org/de/docs/Web/HTTP verfügbar am 03.11.2020, 08:30

[16] Ionos: Was ist HTTP?
https://bit.ly/3qkhYPd verfügbar am 03.11.2020, 08:30

[17] Ionos: TLS - Wie das Internet verschlüsselt wird.
https://bit.ly/3lraMgs verfügbar am 03.11.2020, 08:50

[18] Ionos: HTTPS - Was es bedeutet und warum es wichtig ist.
https://bit.ly/37of0R0 verfügbar am 03.11.2020, 08:50

[19] Android: Android 6.0 Marshmallow
https://www.android.com/intl/de_de/versions/marshmallow-6-0/ verfügbar am 04.11.2020, 09:00

[20] Android: Android 10.0.
https://www.android.com/intl/de_de/android-10/
verfügbar am 04.11.2020, 09:15

[21] Wikipedia: Pixel 2.
https://en.wikipedia.org/wiki/Pixel_2 verfügbar am
04.11.2020, 09:30

[22] Wikipedia: Nexus 6.
https://en.wikipedia.org/wiki/Nexus_6 verfügbar am
04.11.2020, 09:45

[23] Wikipedia: iPhone 8.
https://en.wikipedia.org/wiki/IPhone_8 verfügbar am
04.11.2020, 10:00

[24] Apple: iOS 14.
https://www.apple.com/de/ios/ios-14/ verfügbar am
04.11.2020, 10:05

[25] InfoSecurity: Two in Three Users Reuse Passwords.
https://bit.ly/3prTXEG verfügbar am 20.12.2020, 14:30

[26] Statista: Device Usage of Facebook Users

https://bit.ly/37RW6DJ verfügbar am 20.12.2020, 14:35

[27] BroadbandSearch: Mobile vs. Desktop Internet Usage

https://bit.ly/2X1Lyf3 verfügbar am 20.12.2020, 14:50

[28] Google Play Store: Sparkassen App.

https://bit.ly/38BYA8c verfügbar am 20.12.2020, 15:00

[29] Apple App Store: Sparkassen App

https://apple.co/34MhS9T verfügbar am 20.12.2020, 15:10

[30] Google Play Store: WELT News App

https://bit.ly/38Ds2e6 verfügbar am 20.12.2020, 15:15

[31] Apple App Store: WELT News App

https://apple.co/3aRTffX verfügbar am 20.12.2020, 15:20

[32] Google Play Store: Amazon App

https://bit.ly/2WLeY0K verfügbar am 20.12.2020, 15:25

[33] Apple App Store: Amazon App

https://apple.co/3rwWRJZ verfügbar am 20.12.2020, 15:30

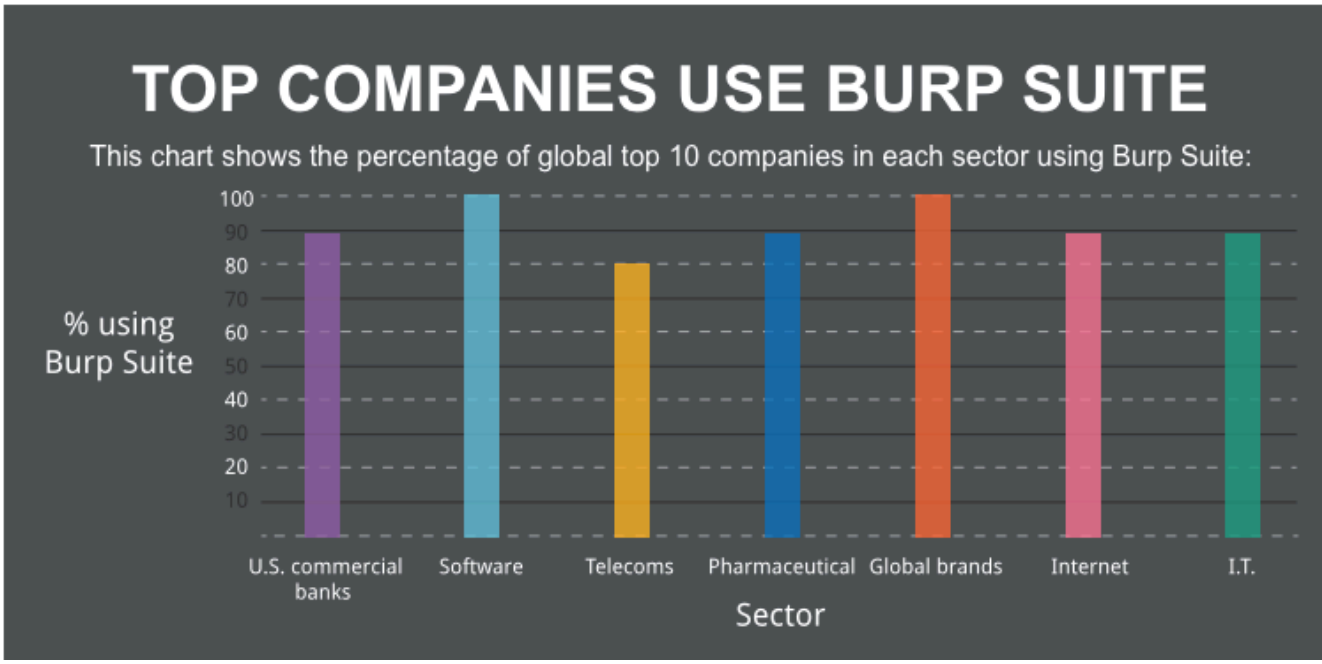
[34] Google Play Store: Corona Warn App

https://bit.ly/38zuylw verfügbar am 20.12.2020, 15:35

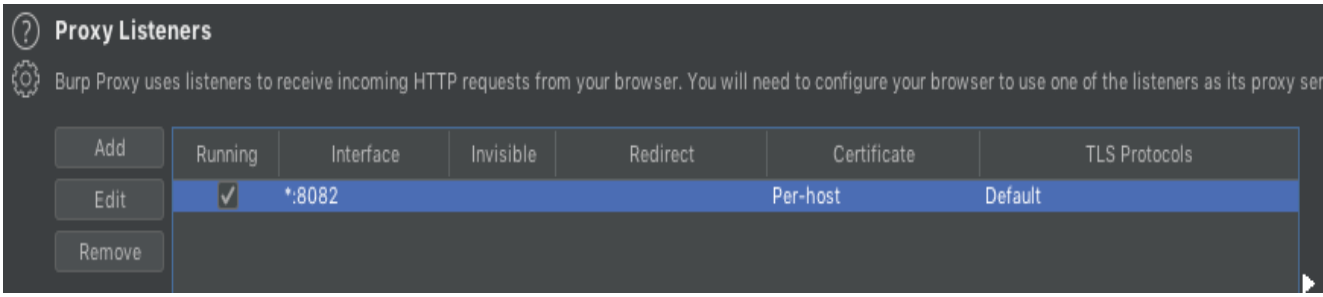
[35] Apple App Store: Corona Warn App

https://apple.co/3pkiyv0 verfügbar am 20.12.2020, 15:40

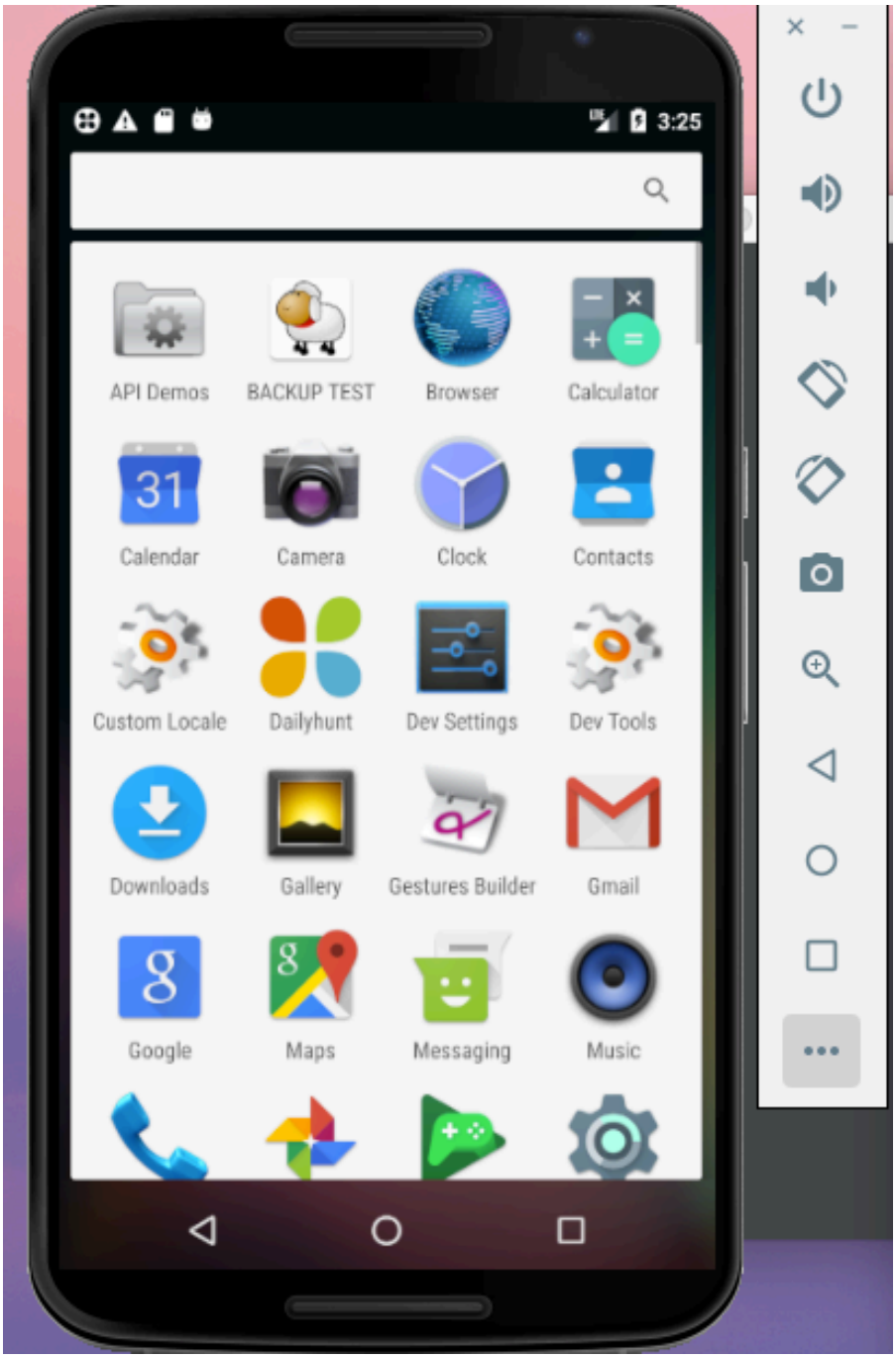
# Anlagen



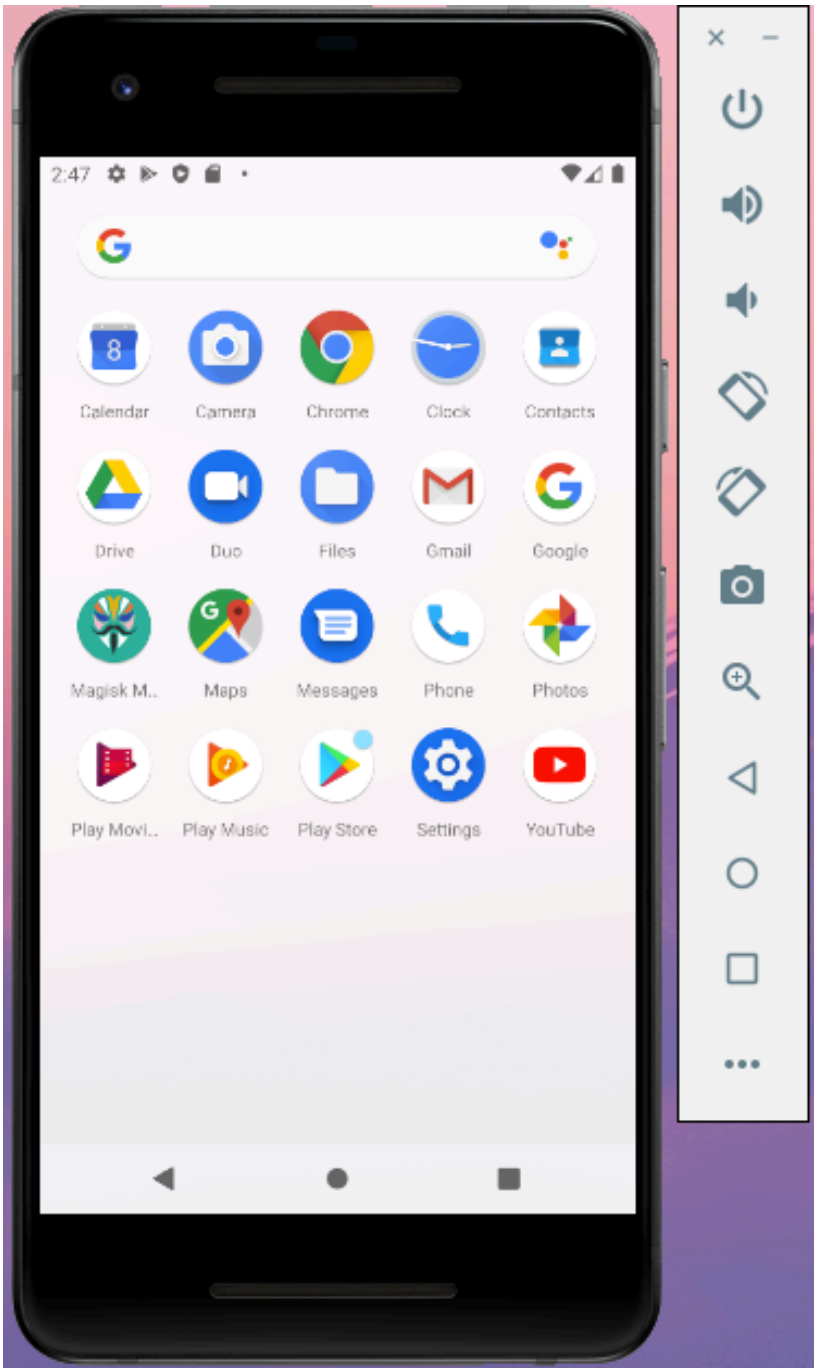

Anlage 1: Graph welche Unternehmen aus welchen Sektoren die Burp Suite nutzen, https://portswigger.net/about zuletzt aufgerufen 09.11.2020



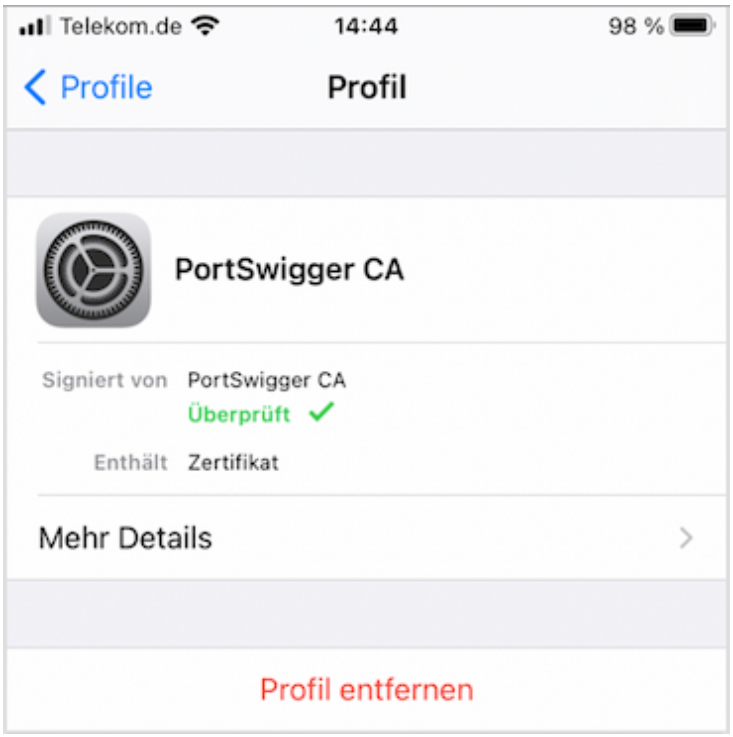

Anlage 2: Einstellung des Proxy Listeners der Burp Suite auf Port 8082 und all Interfaces, eigene Abbildung




Anlange 3: Virtuelles Nexus 6 mit Hilfe des AVM über das Android Studio, eigene Abbildung.

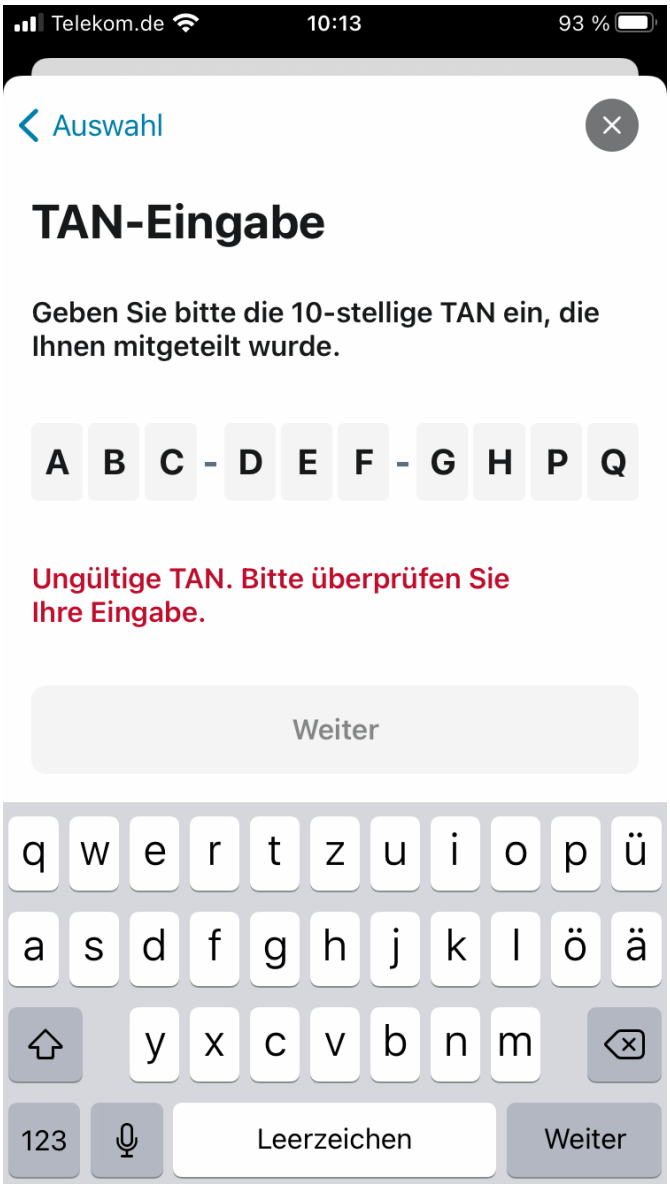



Anhang 4: Virtuelles Pixel 2 mit Hilfe des AVM über Android Studio, eigene Abbildung




Anlage 5: Das Burp Ca Certificate, auf dem iPhone 8 installiert, eigene Abbildung


Anlage 6: Diverse Informationen des Burp CA Certificate auf dem iPhone 8, eigene Abbildung




Anhang 6: TAN wird bei der Corona Warn App nicht akzeptiert, eigene Abbildung

# Selbstständigkeitserklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und nur unter Verwendung der angegebenen Literatur und Hilfsmittel angefertigt habe.

Stellen, die wörtlich oder sinngemäß aus Quellen entnommen wurden, sind als solche kenntlich gemacht.

Diese Arbeit wurde in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegt.

Neuss, der 29. Dezember, 2020

Jan Bornemeyer