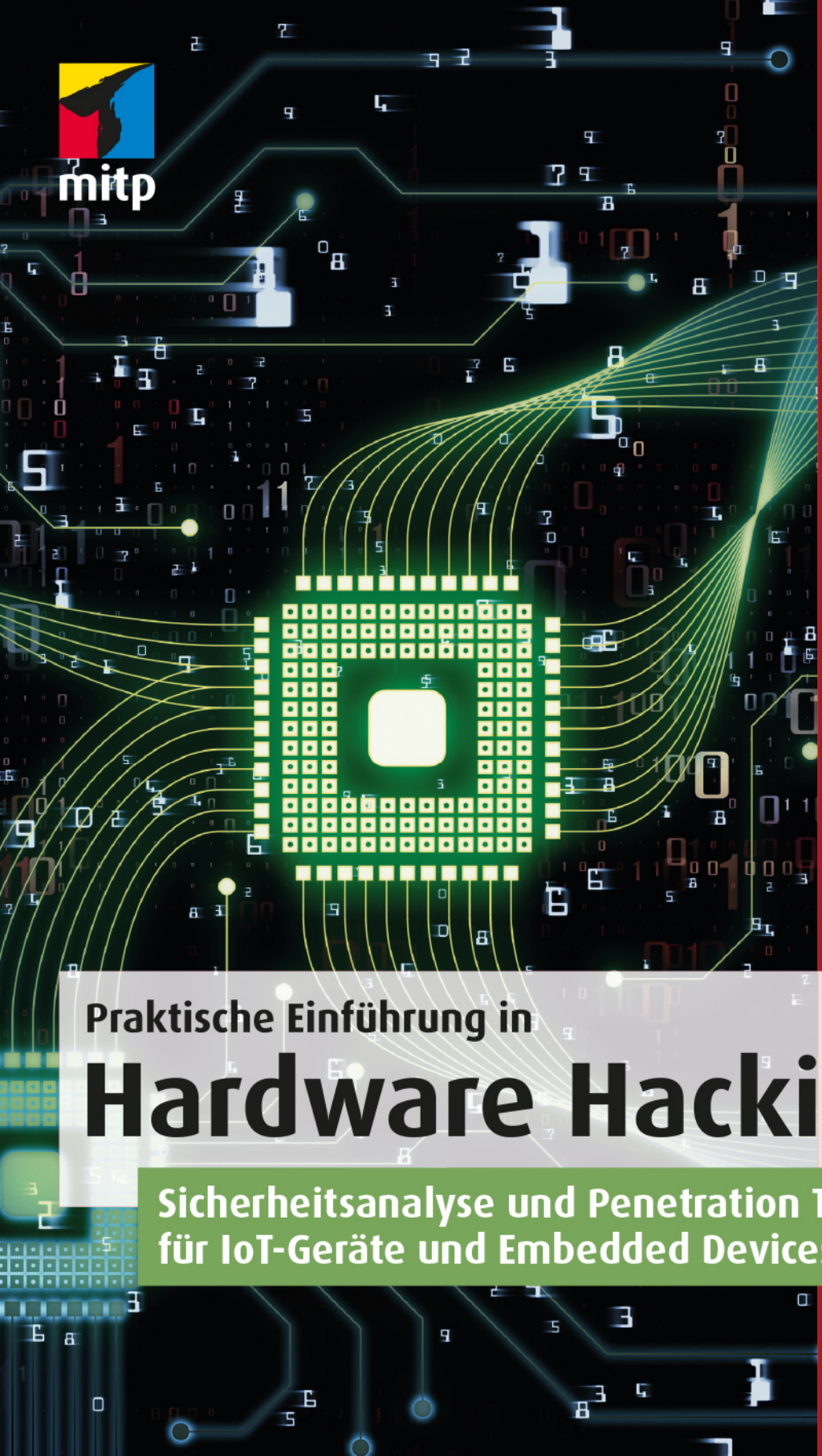

mitp

Marcel Mangel

Sebastian Bicchi

Praktische Einführung in

Hardware Hacking

Sicherheitsanalyse und Penetration Testing für IoT-Geräte und Embedded Devices

Marcel Mangel, Sebastian Bicchi

Praktische Einführung in

Hardware Hacking

Sicherheitsanalyse und Penetration Testing für IoT-Geräte und Embedded Devices

mitp

Bibliografische Information der Deutschen Nationalbibliothek
Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

ISBN 978-3-95845-817-8
1. Auflage 2020

www.mitp.de
E-Mail: mitp-verlag@sigloch.de
Telefon: +49 7953 / 7189 - 079
Telefax: +49 7953 / 7189 - 082

Lektorat: Janina Bahlmann
Sprachkorrektorat: Sibylle Feldmann
Covergestaltung: Christian Kalkert
Satz: III-satz, Husby, www.drei-satz.de
Bildnachweis: © agsandrew / stock.adobe.com

Inhaltsverzeichnis

Vorwort

Eine bei Minusgraden gehackte smarte Heizung führte in Finnland zu eingefrorenen Rohrleitungen in einer Wohnanlage mit einigen Hundert Einheiten. US-Behörden riefen Herzschrittmacher zurück, weil Hacker sie angreifen konnten. Zwei Forscher schafften es, einen Jeep Grand Cherokee über das Internet vollständig fernzusteuern, von Aircondition über Fahrtrichtung, Geschwindigkeit und Sitzverstellung bis hin zur Zentralverriegelung.

Längst betrifft IT-Sicherheit nicht mehr nur den klischeehaften Hacker in seinem mit Pizzakartons zugemüllten dunklen Kabuff, sondern jeden von uns. Mit Cyber-Physical-Systemen haben Programmierfehler und Sicherheitslücken Auswirkungen auf das tägliche Leben.

Marcel Mangel und Sebastian Bicchi sind professionelle Penetration Tester, sie demonstrieren regelmäßig Sicherheitslücken. Auf Basis ihres Wissens und ihrer praktischen Erfahrungen ist das folgende Buch entstanden, in dem sie darlegen, wie Angreifer arbeiten – am Beispiel smarter Geräte.

Sie zeigen auf, wie oft simple Fehler zu gefährlichen Angriffen führen und wie die Angriffe funktionieren. Damit bieten sie Sicherheitstestern eine hervorragende Anleitung dazu, wie sie zur Qualitätssicherung vor der Markteinführung ihre Produkte testen sollten.

Gleichzeitig lernen die Entwickler, wie sie die Angriffsflächen in der Zukunft verringern und so die Sicherheit deutlich erhöhen.

Ich bin ein Verfechter von Qualitätssicherung in der Softwareentwicklung. Zur Qualitätssicherung gehört es, sowohl die Fehlermöglichkeiten zu kennen, um sie zu vermeiden, als auch durch gründliches Testen Fehler zu identifizieren. Zu beidem leistet dieses Buch einen wertvollen Beitrag.

Von Praktikern für Praktiker anhand praktischer Beispiele mit qualifiziertem theoretischem Hintergrund – eine lohnende Lektüre, für die ich den beiden Autoren sehr dankbar bin.

Prof. Dr. Tobias Eggendorfer
Lehrstuhl für IT-Sicherheit
Hochschule Ravensburg Weingarten

Einleitung

Die Vernetzung der Welt schreitet immer weiter voran. Das betrifft nicht nur traditionelle IT-Systeme, sondern mehr und mehr alle möglichen »smarten« Geräte. Diese können alle unter dem Stichwort **I**nternet **o**f **T**hings (IoT) subsummiert werden. Für die Hersteller technischer Geräte ist es heutzutage quasi ein Muss, diese in irgendeiner Weise »smart« zu machen. Ein Gerät wird in der Regel dadurch »smart«, dass es mit anderen Geräten oder Systemen vernetzt ist und Informationen austauschen kann.

Dieser Informationsaustausch geschieht dabei in aller Regel über standardisierte Protokolle und Infrastrukturen wie z. B. das Internet. Für potenzielle Angreifer eröffnet diese Vernetzung eine ganze Reihe neuer Angriffsmöglichkeiten. Anstatt besonders gehärtete klassische IT-Systeme anzugreifen, ist es für Hacker deutlich interessanter geworden, ihren Fokus auf IoT-Geräte zu verlegen. Darunter finden sich eine Menge Geräte, die praktisch überhaupt nicht abgesichert sind. Da kann es, wie das Mirai-Botnetz[1] eindrucksvoll unter Beweis gestellt hat, schon ausreichen, die Geräte einfach mit Standardpasswörtern zu übernehmen und für weitere Angriffe zu missbrauchen. Es wurden insgesamt 64 verschiedene Standardpasswörter getestet, um Zugriff auf die entsprechenden Geräte zu erlangen. Eine genauere Analyse des Sourcecodes sei dem Leser an Herz gelegt.

Die Situation verschärft sich jedoch dramatisch, wenn es sich bei den angegriffenen Geräten nicht mehr um Router oder Kameras handelt, sondern um Geräte aus der Medizintechnik. Auch in dieser Branche werden die Geräte immer »smarter« und bieten damit immer größere Angriffsflächen. Viele Hersteller sind sich aktuell gar nicht richtig im Klaren über die Gefahren, die von der zunehmenden Vernetzung ausgehen. Nicht nur die Medizintechnik ist ein Beispiel für die wachsende Verzahnung von Security und Safety. Weitere Bereiche, für die das gilt, sind die sogenannten »kritischen Infrastrukturen«. Darunter fallen z. B. Energieversorger, der öffentliche Personenverkehr oder auch Krankenhäuser. Werden solche Systeme kompromittiert, kann das fatale Folgen haben. Ein sehr bekanntes Beispiel ist der Stromausfall in Brasilien. Nachdem Angreifer Industrial-Control-Systeme unter ihre Kontrolle gebracht hatten, kam es zu flächendeckenden Stromausfällen, von denen Millionen Menschen in Brasilien über mehrere Stunden betroffen waren.[2]

1 https://github.com/jgamblin/Mirai-Source-Code
2 https://www.wired.com/2009/11/brazil/

Auch hier waren nicht abgesicherte Systeme, die über das Internet erreichbar waren Ausgangspunkt des Angriffs.

Die Hersteller werden jedoch immer mehr zur Verantwortung gezogen. Während man das Thema Cyber Security vor einigen Jahren noch sehr stiefmütterlich behandeln konnte, wird es in den kommenden Jahren mehr und mehr an Wichtigkeit gewinnen. Zum Teil wird dies auch schon durch die Einhaltung von Compliance-Richtlinien wie der Datenschutzgrundverordnung erzwungen.

Ziel des Buchs

Als wir anfingen, uns mit dem Thema Penetration Testing für Internet-of-Things-Geräte zu beschäftigen, gestaltete es sich als überaus schwierig, Bücher zu diesem Thema zu finden. Insbesondere auf dem deutschsprachigen Markt war zu diesem Zeitpunkt kein geeignetes Buch verfügbar. Auch im englischsprachigen Raum gab es zwar schon einige Bücher, doch diese wurden alle nicht dem gerecht, wonach wir suchten. Im Internet auf diversen Blogs fanden wir einige sehr interessante und hilfreiche Informationen, jedoch keine, die das Thema umfassend abdeckten. Mit diesem Buch versuchen wir, genau diese Lücke zu schließen. Das Buch soll zum einen eine umfassende Informationsquelle zum Thema sein und zum anderen als eine Art Referenzwerk zum praktischen Testen von Geräten dienen.

Die OWASP IoT Top 10 stellen ein erwähnenswertes Projekt dar, in dem man sich ebenfalls mit der Sicherheit von IoT-Geräten beschäftigt. OWASP steht für **O**pen **W**eb **A**pplication **S**ecurity **P**roject, und es handelt sich dabei um eine Non-Profit-Organisation, die insbesondere durch die »OWASP Top 10«, eine Liste mit den am häufigsten vorkommenden Schwachstellen in Webanwendungen, bekannt geworden ist. Neben dieser Liste für Webanwendungen gibt es die »OWASP Mobile Top 10« sowie seit 2014 auch die »OWASP IoT Top 10«, in der die zehn am häufigsten vorkommenden Schwachstellen für IoT-Geräte aufgelistet sind. Nach der letztmaligen Aktualisierung im Jahr 2018 sind dies:

1. Schwache, erratbare oder hartcodierte Passwörter
2. Unsichere Netzwerkdienste
3. Unsichere Schnittstellen innerhalb des IoT-Ökosystems
4. Unsichere Update-Mechanismen
5. Verwendung unsicherer oder veralteter Komponenten
6. Nicht ausreichender Schutz von Benutzerdaten
7. Unsichere Transfers und unsichere Speicherung von Daten
8. Unzureichendes Management der Geräte
9. Unsichere Standardeinstellungen
10. Unzureichender Schutz gegen physische Angriffe

Aufbau des Buchs

Das Buch behandelt in acht Kapiteln die einzelnen Aspekte, die beim Testen eines IoT-Geräts vonnöten sind.

Kapitel 1 – Vorbereitung

In diesem Kapitel werden organisatorische sowie technische Projektvorbereitungen dargestellt, die für ein erfolgreiches Security-Testing-Projekt unabdingbar sind. Neben dem Thema Scoping wird auch intensiv auf den Aufbau eines entsprechenden Testing-Labors eingegangen. Abschnitt 1.3 »Das Labor« zeigt den praktischen Aufbau eines Elektroniklabors, welche Geräte benötigt werden und wofür. Die Grundbegriffe der Elektronik werden erläutert sowie Arbeitsweisen im elektronischen Labor.

Kapitel 2 – OSINT

Zu Beginn dieses Kapitels wird der Begriff *OSINT* (**O**pen **S**ource **Int**elligence) im Kontext der Sicherheit vernetzter Geräte erklärt. Weiterhin wird beschrieben, wie die systematische Sammlung und Analyse offen zugänglicher Informationen durchzuführen ist und wie diese zur Sicherheitsanalyse verwendet werden können.

Kapitel 3 – Hardware

Das Kapitel »Hardware« erklärt zunächst die notwendigen Elektronikgrundlagen und die verschiedenen Bauelemente in einer kleinen Baustein-Lehre. Zusätzlich werden Bussysteme und Schnittstellen theoretisch beleuchtet und integrierte Bausteine sowie verschiedene Aspekte der Herstellung, wie Layout und Schemata einer Platine, beschrieben.

Kapitel 4 – Physische Sicherheit

Das Kapitel »Physische Sicherheit« beschäftigt sich mit der Sicherheit des Gehäuses und Tamper-Protection-Maßnahmen. Außerdem werden Hardwaredesign-Grundlagen erklärt (8-/32-Bit-Controller) und verschiedene Angriffspunkte erläutert. Zu guter Letzt wird gezeigt, wie bei einem physischen Zugriff auf das Gerät die Firmware extrahiert werden oder ein Zugriff auf andere Daten in Bausteinen erfolgen kann (JTAG/SWD/UART/SPI).

Kapitel 5 – Firmware

Die Firmware ist quasi das Herzstück eines jedes IoT-Geräts und besitzt damit einen besonderen Stellenwert bei der Sicherheitsanalyse eines solchen. In diesem Kapitel werden zunächst unterschiedliche Möglichkeiten dargestellt, an die Firmware eines Geräts zu gelangen. Im Anschluss beschreibt das Kapitel ausführlich das Entpacken, die Analyse und die Emulation von Firmware-Images.

Kapitel 6 – IoT-Referenzarchitekturen und Netzwerkprotokolle

Das Kapitel beginnt mit einer Einführung in das Thema Protokolle und stellt zwei verschiedene in der Praxis relevante IoT-Referenzarchitekturen vor, bevor auf zwei konkrete Netzwerkprotokolle (Bluetooth Low Energy und Zigbee) im Detail eingegangen wird. Diese beiden Protokolle werden von zahlreichen IoT-Geräten verwendet und stellen damit ein oftmals zentrales Thema bei vielen Security Assessments dar. Neben den Grundlagen der beiden Protokolle werden praktische Angriffe und Tools vorgestellt.

Kapitel 7 – MQTT

Das achte Kapitel widmet sich dem wohl wichtigsten Protokoll auf Anwendungsebene im IoT-Umfeld: MQTT. Hier steht insbesondere eine ausführliche Darstellung der Funktionsweise sowie der wesentlichen Pakettypen im Vordergrund.

Kapitel 8 – Apps

Das App-Kapitel umfasst einen kurzen Einstieg in die OWASP-App *Security* und erklärt, welche Sicherheitsanforderungen an Apps allgemein gestellt werden.

Vertiefend wird auf die Spezifika vernetzter Geräte und Apps eingegangen, insbesondere auf sämtliche Verbindungen (direkt und indirekt) zum vernetzten Gerät.

Kapitel 9 – Backend, Web und Cloud

In diesem Kapitel werden zum einen die Rahmenbedingungen erläutert, die beim Testen von Backend-Systemen zu beachten sind, und zum anderen wird die am weitesten verbreitete Methodik zum Testen von Applikationen nach OWASP erläutert.

Zielgruppe

Das Buch richtet sich in erster Linie an Personen, die Penetration Tests von Internet-of-Things-Geräten durchführen möchten.

Neben den eigentlichen Testern kann das Buch durchaus auch für Projektmanager oder Security-Verantwortliche aufseiten der Hersteller interessant sein.

Was Sie benötigen

Als Leser dieses Buchs sollten Sie bereits über grundlegende Kenntnisse in IT-Sicherheit, insbesondere in den Bereichen Netzwerk- und Applikationssicherheit, verfügen. Zudem wird ein routinierter Umgang mit Linux vorausgesetzt.

Trotzdem legen wir Wert darauf, dass auch interessierte Einsteiger den Inhalten des Buchs gut folgen können.

Über die Autoren

Marcel Mangel verfügt über mehr als eine Dekade praktischer Erfahrung in IT-Sicherheit. Sowohl im defensiven als auch im offensiven Bereich war er mehrere Jahre für renommierte Unternehmen tätig und hat neben einem Master in Informatik noch über eine ganze Reihe von anerkannten Zertifikaten. Nebenbei arbeitet Marcel Mangel bereits seit mehreren Jahren im Rahmen von Lehraufträgen und Gastvorlesungen an diversen Universitäten und Fachhochschulen als Dozent. Zurzeit hält er die Master-Vorlesung »Vertiefung der IT-Sicherheit« an der Fachhochschule Rosenheim.

Sebastian Bicchi begann bereits in seiner Jugend mit dem Aufspüren von Sicherheitslücken in Webseiten und Anwendungen. Nach seinem IT-Security-Studium gründete er in Wien gemeinsam mit Studienkollegen das Unternehmen »Security Research« (sec-research.com) und beschäftigte sich insbesondere mit den technischen Aspekten der IT-Security. Die Synergien seiner Ausbildungen in verschiedenen Bereichen (Elektrotechnik/industrielle Informationstechnik, Informationstechnologien und Telekommunikation, IT-Security) schufen die Basis für sein fundiertes Wissen über IoT- und Hardware-Hacking. Sebastian Bicchi betätigt sich darüber hinaus freiwillig in nicht kommerziellen Organisationen für Informationssicherheit, zum Beispiel als Co-Chapter-Lead bei OWASP für das Chapter Vienna.

Danksagungen

Marcel Mangel

Mein besonderer Dank gilt meiner Lebensgefährtin Sarah sowie meiner gesamten Familie, die mich bei der Erstellung des Buchs außerordentlich unterstützt haben.

Daneben möchte ich mich ganz herzlich bei meinem Koautor und Freund Sebastian Bicchi bedanken, ohne den dieses Buch nicht entstanden wäre.

Außerdem bedanke ich mich sehr herzlich beim mitp-Verlag für die Möglichkeit der Veröffentlichung sowie insbesondere bei unserer Lektorin Frau Janina Bahlmann für die tolle Unterstützung und Zusammenarbeit.

Und zu guter Letzt bedanke ich mich ebenfalls sehr herzlich bei Christian Salzmann, der die Abschnitte über Bluetooth und Zigbee beigesteuert hat, und bei Tobias Eggendorfer für den stetigen Austausch und das Vorwort.

Sebastian Bicchi

Mein besonderer Dank gilt meiner Freundin Sandra, die mich in allem, was ich mache, unterstützt und mir, während ich dieses Buch geschrieben habe, alle nur erdenklichen Lasten abgenommen hat.

Weiterhin möchte ich mich bei Marcel Mangel bedanken – für die Möglichkeit, dieses Buch zu schreiben, und für die Zusammenarbeit bei der Erstellung des Buchs und allen weiteren spannenden Projekten, die wir zusammen bearbeitet haben.

Ein großes Dankeschön möchte ich an dieser Stelle auch dem Verlag und Frau Bahlmann für ihre Unterstützung während des gesamten Erstellungsprozesses des Buchs und für das außerordentlich gute Feedback ausrichten.

Mein weiterer Dank gilt den Personen, die meine Ausbildung zu einem besonderen Weg gemacht und mir letztendlich ermöglicht haben, dieses Buch zu schreiben: Manuel Koschuch (FH Campus Wien) und Bernd Stanzl (HTL Mödling). Beide haben mich über alle Maßen unterstützt, sämtliche Fragen, auch über den Lehrplan hinaus, immer mit genügend Zeit und Geduld beantwortet und mich inspiriert, Wissen weiterzugeben.

Ich möchte mich ebenfalls bei Zlatko Sehanovic und Herbert Dowalil für das Vorab-Review und das inhaltliche Feedback bedanken.

Kapitel 1

Vorbereitung

1.1 Organisatorische Vorbereitungen

Bei der Durchführung eines Penetration Tests auf einem IoT-Gerät ist ein gut strukturierter Projektablauf maßgeblich für den Erfolg.

Ein wesentlicher Aspekt, dem ausreichend Aufmerksamkeit geschenkt werden sollte, ist das Thema »Scoping«, also die Festlegung der thematischen Abdeckung bzw. des Geltungsbereichs des Tests. Das Scoping muss zwingend in enger Abstimmung mit dem Auftraggeber erfolgen. Nach unserer Erfahrung liegt hier eine häufige Ursache in der Unzufriedenheit vieler Auftraggeber am Ende eines Projekts. Wichtig ist es, sicherzustellen, dass der Auftraggeber und der Tester die »gleiche Sprache sprechen«. Häufig sind die Tester so tief technisch verwurzelt, dass es bei Projektbesprechungen zu Missverständnissen kommt und insbesondere Fachbegriffe unterschiedlich ausgelegt werden. Deshalb ist es zu Beginn eines Projekts sinnvoll, einige wesentliche Begriffe kurz zu definieren und für ein gemeinsames Verständnis zu sorgen.

Darüber hinaus sollten folgende Punkte abgestimmt werden:

- Testmethodik (BlackBox, GreyBox oder WhiteBox)
- Testtiefe
- Testziele

Bei einem *BlackBox*-Test verfügt der Tester über keinerlei nicht öffentliche Informationen über das zu testende Gerät bzw. System. Der Tester schlüpft also in die Rolle eines externen Angreifers, der sich alle notwendigen Informationen selbstständig besorgen muss. Im Gegensatz dazu stehen dem Tester bei einem *WhiteBox*-Test viele nützliche nicht öffentliche Dokumente wie z. B. Netzwerkdiagramme, Funktionsdiagramme, interne technische Dokumentationen, Datenflussdiagramme oder auch der Quellcode der Applikationen zur Verfügung. Der *GreyBox*-Ansatz stellt den Mittelweg zwischen BlackBox und WhiteBox dar. Hier verfügt der Tester über einige nicht öffentliche Informationen. Im Regelfall stellt der Auftraggeber zumindest Netzwerkdiagramme und interne technische Dokumentationen zur Verfügung, während der Quellcode von Applikationen nicht herausgegeben wird.

Bei der Abstimmung der Testtiefe geht es in erster Linie darum, festzulegen, inwieweit identifizierte Schwachstellen ausgenutzt werden sollen. In der Regel wer-

den dazu sogenannte Proof-of-Concept-Exploits erstellt, die belegen, dass die entsprechende Schwachstelle ausnutzbar ist.

Es ist sinnvoll, für jedes Projekt Ziele zu definieren, die beim Testen verfolgt werden sollten. Diese können auch in Form von Fragen formuliert werden.

Einige sinnvolle Fragen, deren Beantwortung nach Abschluss des Tests möglich sein sollte, sind:

- Ist es möglich, unautorisierten Zugriff auf das Gerät zu bekommen?
- Ist es möglich, an sensitive Daten zu gelangen?
- Ist die Angriffsfläche des Produkts minimal gehalten, um potenzielle Angriffe zu erschweren?

Das Motiv vieler Auftraggeber ist zunehmend das Thema »Compliance«, z. B. im Rahmen einer PCI-Zertifizierung oder der Datenschutzgrundverordnung. Aus unserer Sicht sollte die Einhaltung der Compliance zwar mit berücksichtigt werden, jedoch niemals das einzige Motiv für eine Überprüfung sein. Die Sicherheit des IoT-Geräts bzw. dessen Ökosystem sollte immer das zentrale Motiv eines solchen Tests darstellen.

1.2 Grundlegender Ablauf des Penetration Tests

Obwohl ein Penetration Test dynamisch abläuft und eine Aufgabe in die andere greift, gibt es ein grundsätzliches Vorgehensmuster:

1. **Auftragsvergabe, Vorbereitung, Klärung des Scopes:** Bevor Sie mit dem Test beginnen, müssen sämtliche formellen Punkte geklärt werden. Dazu zählt der Testmodus, der Scope (Testziel), die Angriffsfreigabe, und natürlich müssen Sie die Testobjekte (IoT-Geräte) erhalten.
2. **Informationsbeschaffung:** Wie funktioniert das Gerät? Welche öffentlichen Informationen gibt es? Dieser Teil wird in Kapitel 2 behandelt.
3. **Inbetriebnahme nach Vorgabe:** Nehmen Sie das Gerät wie durch den Hersteller vorgesehen in Betrieb. Zeichnen Sie dabei jeglichen Netzwerkverkehr auf – die erste Inbetriebnahme beinhaltet sehr oft sicherheitskritische Abläufe wie Updates, Kopplung oder Registrierung. Die erste Inbetriebnahme bzw. Aufzeichnung wird im nächsten Abschnitt erläutert.
4. **Firmware-Analyse:** Dieser Schritt hängt davon ab, ob Sie die Firmware im Rahmen der OSINT-Analyse oder anderweitig bereits erhalten konnten. Ist dies der Fall, kann die Firmware-Analyse durchgeführt werden – siehe hierzu Kapitel 5. Haben Sie die Firmware noch nicht, ziehen Sie die Hardware- oder App-Analyse vor.
5. **Hardwareanalyse und physische Sicherheit:** Im Rahmen der Hardwareanalyse sind Sie unter Umständen in der Lage, die Firmware zu extrahieren. Führen Sie diese Analyse durch und verbinden Sie den Vorgang mit der Analyse der physischen Sicherheit (Kapitel 3 und Kapitel 4).

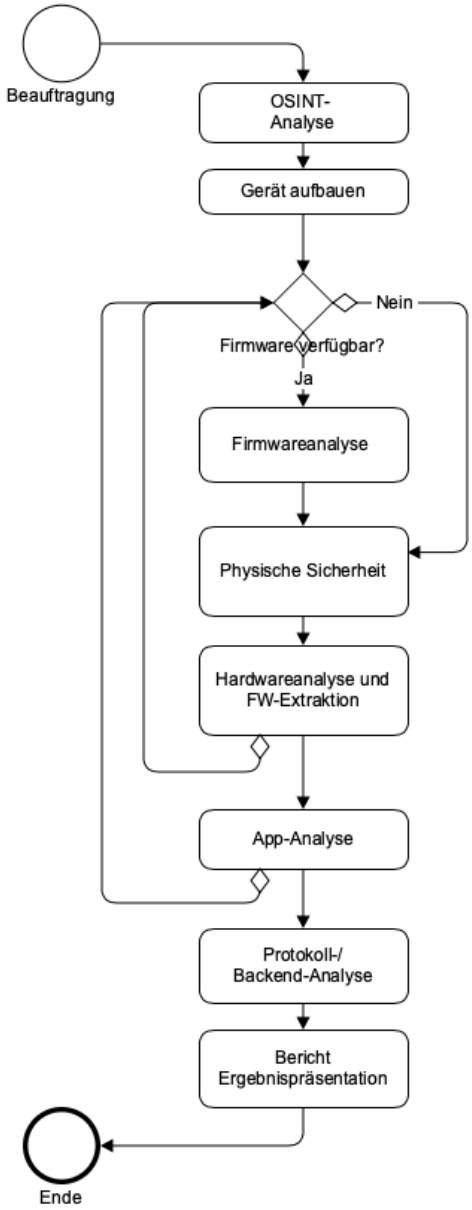

Abb. 1.1: Ablaufdiagramm eines IoT-Penetration Tests

Wichtig

Wenn Sie nur ein Testobjekt zur Verfügung haben, verschieben Sie die Hardwareanalyse ans Ende des Tests, da das Testobjekt beschädigt werden könnte und Sie danach nicht mehr in der Lage sein könnten, die anderen Tests durchzuführen!

6. **App-Analyse:** Führen Sie die App-Analyse wie in Kapitel 8 beschrieben durch.
7. **Protokollanalyse:** Im Rahmen der vorangegangenen Tests sollten nun einige Protokollangriffspunkte sichtbar geworden sein. Führen Sie jetzt den Penetration Test auf die IoT-Protokolle und Backends durch, wie in den Kapitel 6, Kapitel 7, und Kapitel 9 beschrieben.
8. **Schwachstellen zusammenfügen:** Das holistische Bild ist sehr wichtig, da reale Angriffe sehr oft eine Kombination unterschiedlicher Schwachstellen ausnutzen. Verbinden Sie alle gefundenen Schwachstellen und beschreiben Sie eventuelle Zusammenhänge.
9. **Reporting:** Erstellen Sie einen Bericht und führen Sie die Nachbereitung mit dem Auftraggeber durch. Das Reporting besteht normalerweise aus einer Präsentation der Schwachstellen und Empfehlungen dazu, wie diese behoben werden können. Auch wird oft ein Nachtest besprochen, um die Behebung der Schwachstellen zu prüfen.

Dieser gesamte Testablauf ist grafisch in Abbildung 1.1 dargestellt.

Aufzeichnung der ersten Inbetriebnahme

Bevor der eigentliche Test durchgeführt wird, sollten Sie das Gerät zunächst wie durch den Hersteller vorgesehen in Betrieb nehmen. So erfahren Sie, wie das Gerät grundsätzlich funktioniert bzw. wie die vom Hersteller gedachte Arbeitsweise ist.

Die erste Inbetriebnahme ist von besonderer Bedeutung: Oft werden hier einmalige Vorgänge gestartet, die sich später nicht einfach reproduzieren lassen, zum Beispiel ein Firmware-Update oder die Registrierung des Geräts mit dem Server. Daher ist es notwendig, die Inbetriebnahme von Beginn an ausreichend zu dokumentieren. Dazu zählt auch insbesondere der Netzwerkverkehr des Geräts.

Je nachdem, wie das Gerät mit dem Netzwerk zu verbinden ist, stehen verschiedene Aufzeichnungsmöglichkeiten zur Auswahl. Verbindet sich das Gerät mittels WLAN mit dem Netzwerk, kann die Konfiguration in Abbildung 1.2 verwendet werden.

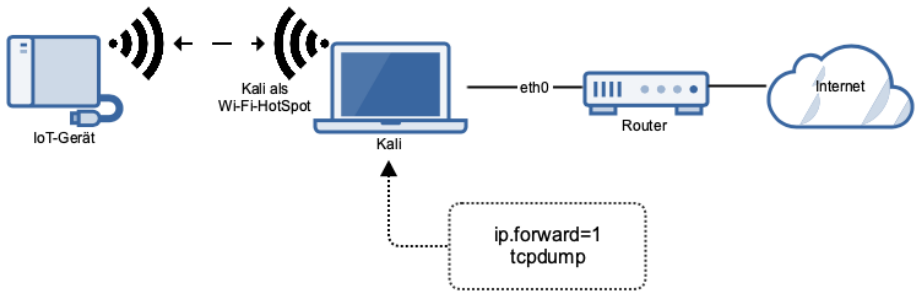

Abb. 1.2: Capturen der Wi-Fi-Verbindung

In dieser Konfiguration können Sie folgenden Befehl verwenden, um die Aufzeichnung durchzuführen:

```
tcpdump -i wlan0 -s 65535 -w ./capture-01.pcap
```

Besitzt das Gerät hingegen einen Ethernet-Anschluss bzw. muss die Ersteinrichtung über Ethernet durchgeführt werden, können Sie einerseits auf Hardware wie das Hak5 PacketSquirrel[1] zurückgreifen, das in Abbildung 1.3 dargestellt ist.

Abb. 1.3: PacketSquirrel

Mit einer entsprechenden Konfiguration (entnehmen Sie diese bitte der Anleitung bzw. dem Hak5-Forum) und einem USB-Stick zeichnet das PacketSquirrel den Verkehr zwischen dem Gerät und dem Netzwerk zuverlässig auf. Diese Methode ist komfortabel, da kein Computer benötigt wird, jedoch muss das PacketSquirrel hierzu gekauft werden.

Sie können die Aufzeichnung jedoch auch mit einem Computer oder dem Raspberry Pi durchführen. Hierzu benötigen Sie eine zweite Ethernet-Karte – hat Ihr Computer diese nicht eingebaut, können Sie auch eine externe USB-Ethernet-Karte verwenden, wie in Abbildung 1.4 gezeigt.

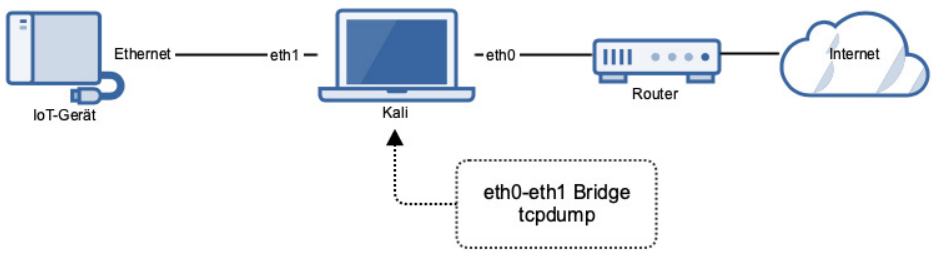

Abb. 1.4: Paket Capture Ethernet

Die beiden Karten müssen nun zu einer Netzwerkbrücke konfiguriert werden, sodass Pakete einfach zwischen den Ports durchgereicht werden. Installieren Sie

1 https://docs.hak5.org/hc/en-us/categories/360000982574-Packet-Squirrel

hierzu die bridge-utils – unter Debian oder Kali mit dem Kommando `aptitude install bridge-utils`.

Um nun eine Netzwerkbrücke zwischen den beiden Karten – zum Beispiel eth0 und eth1 – zu erstellen, verwenden Sie folgende Kommandos:

```
brctl addbr br0
brctl addif br0 eth0 eth1
```

Dadurch erhalten Sie ein neues virtuelles Netzwerkgerät namens br0, das die beiden realen Ethernet-Adapter eth0 und eth1 miteinander verbindet. Mittels `tcpdump` können Sie nun den Verkehr aufzeichnen.

```
tcpdump -i br0 -s 65535 -w ./capture-01.pcap
```

Wenn sich das IoT-Testgerät über Bluetooth mit einer App verbindet, sollten Sie ein erweitertes Setup verwenden. Aktivieren Sie hierzu bei Android zunächst die Funktion *Bluetooth HCI Snooping* in den Entwickleroptionen. Dadurch wird der Bluetooth-Verkehr aufgezeichnet (Details hierzu finden Sie im Abschnitt 6.3.5).

Verwenden Sie dann das Setup, das Abbildung 1.5 darstellt.

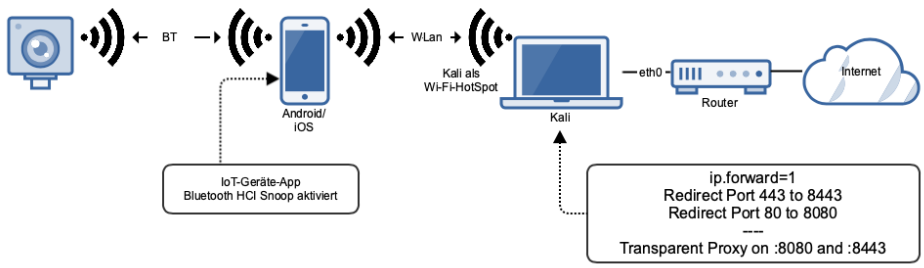

Abb. 1.5: Paket Capture Bluetooth

Zusätzlich sollten Sie zuvor noch die Einrichtungsschritte aus dem Abschnitt 8.4.1 »Netzwerkverkehr« durchführen und gleichzeitig den Netzwerkverkehr der App mitschneiden.

1.3 Das Labor

Das Labor ist der Arbeitsbereich für die Tests und sollte daher entsprechend ausgestattet sein. Im Labor werden Sie die Geräte zerlegen, die Hardware analysieren sowie Bluetooth-, WLAN- und LAN-basierte Angriffe durchführen.