AV-TEST
The Independent IT-Security Institute
Magdeburg Germany

Internet of Things

Sicherheitsevaluation von 7 Fitness-Trackern unter Android und der Apple Watch

Eric Clausing

Michael Schiefer

11. Juli 2016

1 Einleitung

Wie sich im vergangenen Jahr gezeigt hat, ist das Interesse an und die Beliebtheit von Fitness-Trackern weiter gestiegen und ihr Erfolgszug scheint vorerst kein Ende zu nehmen. Wir nehmen diesen Sachverhalt zum Anlass, aufbauend auf unserem ersten Test von 2015 [Clausing et al., 2015], einen weiteren Test durchzuführen, der neue Entwicklungen in diesem Bereich berücksichtigen soll und weitere aktuelle Geräte in den Fokus unserer Untersuchungen rückt. Zu diesem Zweck haben wir sieben neue Produkte aus verschiedenen Preissegmenten und mit unterschiedlicher Ausstattung ausgewählt, die wir in unseren Testverfahren detailliert sicherheitstechnisch untersuchen. Für diesen Test haben wir unsere Testmethodik weiter verfeinert und um einige wichtige Aspekte ergänzt, wie beispielsweise die Sicherheit der Online-Kommunikation. In diesem ersten Kapitel soll kurz das Thema eingeleitet sowie die Motivation und die geleisteten Vorarbeiten kurz umrissen werden. Im zweiten Kapitel gehen wir grundsätzlich auf unser Testkonzept ein und erläutern den verwendeten Testaufbau und die Testdurchführung. Das dritte Kapitel dient der Präsentation und Erläuterung der Testergebnisse. Im letzten Kapitel wird eine abschließende Zusammenfassung geboten und ein Ausblick auf zukünftige Tests gegeben.

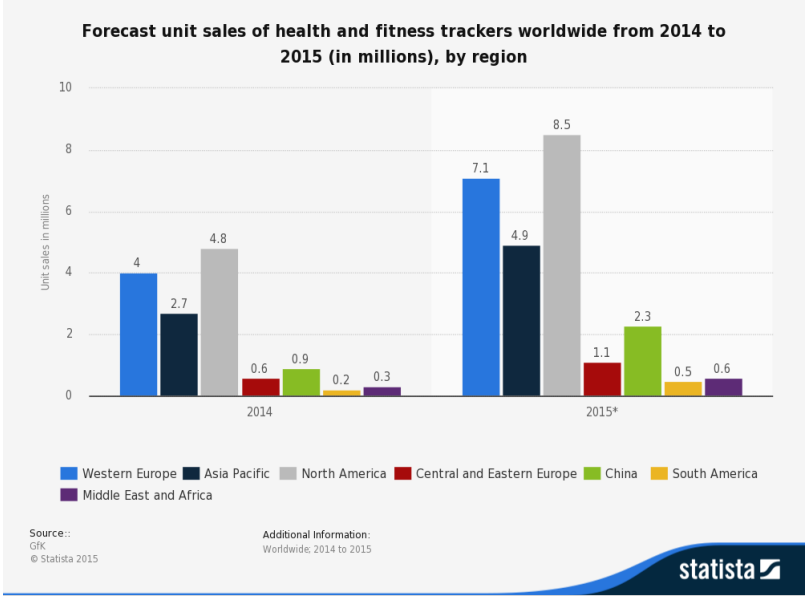

Abbildung 1.1: Weltweite Verkaufszahlen von Fitness-Trackern für 2014 und 2015

1.1 Motivation und Vorarbeiten

Wie bereits erwähnt und in Abbildung 1.1 zu erkennen hat sich die Bedeutung von Fitness-Trackern weltweit weiter erhöht und die Verkaufszahlen steigen stetig an. Wie bereits in unserem ersten Test [Clausing et al., 2015] berichtet wurde, sind Pläne, diese Tracker auch für versicherungstechnische Zwecke zu verwenden, in den USA bereits in Ausführung. Mehrere große Versicherungsanbieter haben bereits Rabatt-Programme aufgelegt, die das Tragen eines Fitness-Trackers und die Bereitstellung der gesammelten Daten mit Gutscheinen und Vergünstigungen belohnen. In Deutschland wird zumindest die Anschaffung eines Trackers bereits von mehreren Versicherungsunternehmen subventioniert [Dörner, 2015, Vers, 2015], auch wenn diese nach eigener Aussage an den Daten selbst (noch) nicht interessiert sind. Allein für diesen Zweck muss ein Fitness-Tracker schon gewissen Maßstäben entsprechen, aber neuerliche Entwicklungen in juristischer Hinsicht rücken die Anforderungen an Manipulationsschutz und robuster Authentifizierung weiter in den Mittelpunkt. Mehrere Nachrichten-Websites berichten von aktuellen Fällen, in denen die Daten, die von Trackern erfasst wurden, als Beweismittel vor Gericht angebracht wurden [Fleischer, 2016, Gardner, 2016]. Wie unsere Tests zeigen, sollte jedoch eine gewisse Skepsis gegenüber dem Manipulationsschutz und der robusten Authentifizierung bestehen bleiben, da einige der Produkte, die sich derzeit auf dem Markt befinden, noch ein gutes Stück weit entfernt von dem sind, was als Beweismittel in einer Gerichtsverhandlung zugelassen werden sollte.

Unser Testkonzept und unsere Testdurchführung basiert dabei auf unserem ersten Test [Clausing et al., 2015]. Einige Testkriterien wurde dabei angepasst und/oder verfeinert, um bestimmte Aspekte mehr in den Vordergrund zu rücken und andere, weniger relevante Aspekte etwas zurückzustellen.

2 Testkonzept

In diesem Kapitel beschreiben wir die Testumgebung inklusive Testaufbau und -durchführung. Außerdem werden die getesteten Produkte aufgeführt und ihre zur Verfügung stehende Funktionalität aufgelistet und kurz erläutert. Die zugehörige und getestete Software wird aufgeführt und die entsprechend betrachtete Software-Version vermerkt.

2.1 Testaufbau

Der Testaufbau für die zweite Testrunde wurde im Vergleich zum ersten Testaufbau leicht erweitert. Da wir in diesem Test auch die potentielle Manipulierbarkeit der Datenübertragung prüfen wollen, wird der Testaufbau um einen weiteren Rechner ergänzt, der zwischen Smartphone und Internet als Man-in-the-Middle dienen soll. Abbildung 2.1 illustriert den Aufbau. Über diesen Rechner wird der Netzwerkverkehr geleitet, beobachtet und gegebenenfalls gezielt manipuliert. Als Tool zur Umsetzung des Man-in-the-Middle nutzen wir *mitmproxy* [mitmproxy, 2014], welches aufgrund der guten Konfigurierbarkeit und einfachen Handhabung optimal geeignet ist. Es handelt sich bei *mitmproxy* um ein Linux-basiertes Tool, welches ermöglicht, auch HTTPS-Verbindungen durch einen Man-in-the-Middle-Angriff aufzubrechen und den Klartextinhalt von Requests und den dazugehörigen Responses zu betrachten und zu verändern. Wir überprüfen damit, ob für einen Nutzer oder Angreifer die Möglichkeit besteht, die von der App synchronisierten Daten abzufangen und zu verändern, ohne dass der dazugehörige Server die Manipulation bemerkt. Wir verwenden für unsere Tests *mitmproxy* im Transparent-Modus mit custom Gateway, wie in Abbildung 2.2 dargestellt. Ansonsten ändert sich am grundsätzlichen Versuchsaufbau dieses zweiten Tests nichts: Es wird weiterhin der entsprechende Tracker mit einem Smartphone (wahlweise mit installierter Original- oder Test-App) über Bluetooth verbunden, die stattfindende Bluetooth-Kommunikation beobachtet, die Original-App analysiert und die Online-Kommunikation zwischen App und Server auf Schwachstellen, d.h. unverschlüsselte Verbindungen, untersucht.

2.2 Testdurchführung

Die Testdurchführung gliedert sich in mehrere Einzelschritte, die schließlich das Gesamtergebnis für die Untersuchung liefern. Diese Schritte sind:

- Analyse der Original-App
- Analyse der Bluetooth-Kommunikationen zwischen Tracker und Smartphone mit Original- und/oder Test-App

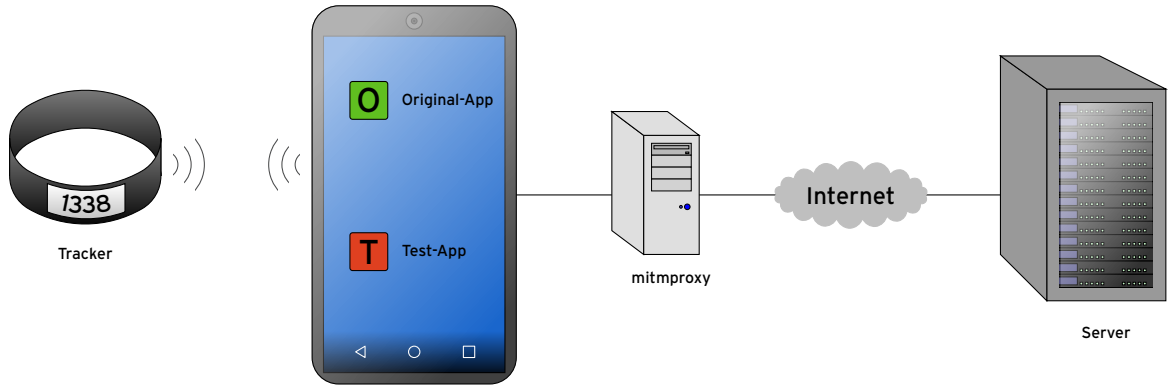

Abbildung 2.1: Schematischer Testaufbau

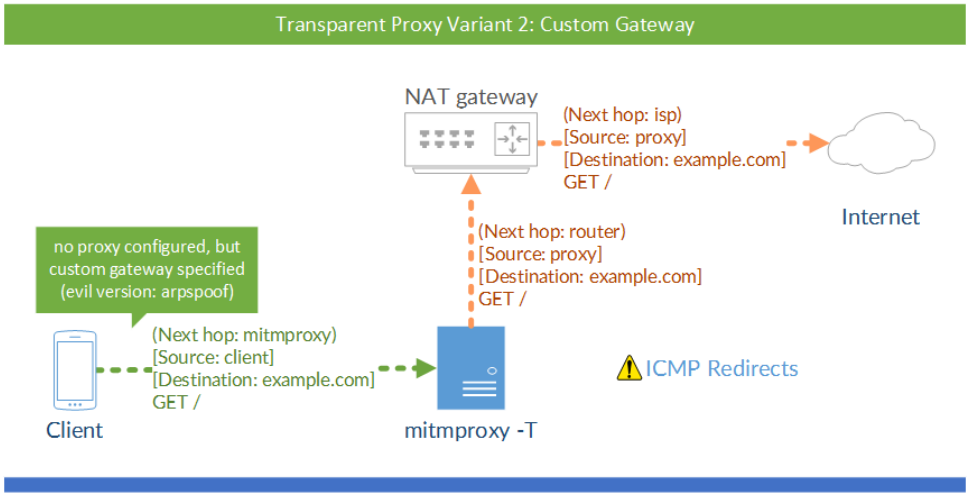

Abbildung 2.2: Illustration des für unsere Tests verwendeten *mitmproxy*-Transparent-Modus mit custom Gateway (Quelle: http://docs.mitmproxy.org/en/latest/_images/proxy-modes-transparent-2.png**)**

- Analyse der Online-Kommunikation der Original-App

Im ersten Schritt wird die Original-App auf potentielle Schwachstellen untersucht. Dabei wird auf allgemeine Sicherheitsmerkmale, wie beispielsweise die Verwendung von Code-Obfuscation oder die Absicherung der lokal gespeicherten Nutzerinformationen geprüft. Im zweiten Schritt wird dann die Bluetooth-Kommunikation betrachtet. Dabei wird beobachtet, welche Informationen und in welcher Form sie zwischen Tracker und Smartphone bzw. Original-App ausgetauscht werden. Anschließend wird versucht mit der Test-App die Kommunikation nachzustellen (z.B. durch replay von einem zweiten Smartphone), um so zu ermitteln, ob eine robuste Authentifizierung verwendet wird und es einem Angreifer potentiell möglich wäre, auf diese Weise Zugriff auf Trackerfunktionen und gespeicherte Inhalte zu erlangen. In dieser zweiten Testrunde wird in diesem Schritt zusätzlich noch überprüft, ob es möglich ist, mit der in Android 5.0 neu hinzugekommenen Bluetooth-Peripheral-Funktionalität [Peripheral, 2014] (welche einem Smartphone theoretisch erlaubt sich als Fitness-Tracker auszugeben) der Original-App einen legitimen Tracker vorzutäuschen. Im dritten und letzten Schritt wird die komplette Online-Kommunikation untersucht, die die Original-App zu Hersteller-Servern

unterhält. Dabei wird erst einmal überprüft, ob die wichtigen Verbindungen zur Nutzerauthentifizierung sowie Cloud-Synchronisierung und Firmwareaktualisierung durch das HTTPS-Protokoll abgesichert sind. Zusätzlich überprüfen wir in dieser zweiten Testrunde auch die Möglichkeit, diese gesicherten Verbindungen vom Server und/oder der Original-App mitlesen und theoretisch auch manipulieren zu können.
Die Testmethodik für die *Apple Watch* unterscheidet sich dabei in einigen Aspekten von der für die anderen getesteten Produkte. Die Ergebnisse und mehr Details zur Durchführung des Tests der *Apple Watch* werden in Abschnitt 4 diskutiert.

2.3 Produkte

Die Liste aller getesteten Produkte und ihrer Features ist in Tabelle 2.1 aufgeführt. Wie man erkennen kann, befinden sich diesmal insgesamt acht Produkte unterschiedlicher Hersteller im Test, von denen sieben mit Android kompatibel sind. Die Auswahl der Geräte erfolgte aufgrund ihrer Bekanntheit und Verbreitung. Abhängig vom Preis unterscheiden sich die untersuchten Produkte auch in ihrer Ausstattung stark voneinander.

	Apple *Watch*	Basis *Peak*	Microsoft *Band 2*	Mobile Action *Q-Band*	Pebble *Time*	Runtastic *Moment Elite*	Striiv *Fusion*	Xiaomi *MiBand*
Bluetooth 4.0 (Low Energy)	✔	✔	✔	✔	✔	✔	✔	✔
WLAN	✔	✘	✘	✘	✘	✘	✘	✘
Mikrofon	✔	✘	✔	✘	✔	✘	✘	✘
Display	✔	✔	✔	✔	✔	✘	✔	✘
Pulsmessung	✔	✔	✔	✘	✘	✘	✘	✘
Schrittzähler	✔	✔	✔	✔	✔	✔	✔	✔
Schlafüberwachung	✘	✔	✔	✔	✔	✔	✔	✔
Integriertes GPS	✘*	✘	✔	✘	✘	✘	✘	✘
Magnetometer	✘	✘	✘	✘	✔	✘	✘	✘
Barometer	✘	✘	✔	✘	✘	✘	✘	✘
UV-Sensor	✘	✘	✔	✔	✘	✘	✘	✘
Umgebungslichtsensor	✔	✘	✔	✔	✔	✘	✘	✘
Kapazitive/Galvanische Sensoren	✘	✔	✔	✘	✘	✘	✘	✘
Hauttemperatursensor	✘	✔	✔	✘	✘	✘	✘	✘

✔integriert/unterstützt ✘nicht integriert/unterstützt

* falls möglich, wird GPS vom Smartphone verwendet

Tabelle 2.1: Überblick über die Trackerfunktionen

In Tabelle 2.2 werden die den Produkten zugehörigen Applikationen und die getestete Version aufgeführt. Alle getesteten Versionen werden offiziell im Google Play Store zum Download angeboten oder sind direkt Teil des Smartphone-Betriebssystems. Aufgrund des zeitlichen Versatzes der Tests sind nicht in alle Fällen die zum Zeitpunkt der Veröffentlichung aktuellsten Versionen getestet worden.

App-Bezeichnung	Version
Apple *Apple Watch*	
Watch	2.1
Basis *Peak*	
Basis Peak	1.17.1
Microsoft *Band 2*	
Microsoft Health	1.3.20213.1
Pebble *Time*	
Pebble Time	3.9.1-966-bc5f043
Runtastic *Moment Elite*	
Runtastic Me	1.5.3
SportPlus *Q-Band*	
i-gotU Life	1.2.1506.947
Striiv *Fusion*	
Striiv Activity Tracker	1.0.1024p
Xiaomi *MiBand*	
Mi Fit	1.8.441

Tabelle 2.2: Überblick der den Trackern zugehörigen getesteten Applikationen

3 Testergebnisse

Die Tabelle 3.1 fasst die wesentlichen Ergebnisse unserer Untersuchung in den drei Kategorien *Tracker*, *Applikation* und *Online-Kommunikation* zusammen und präsentiert die Einzelergebnisse der jeweiligen Produkte. Die einzelnen Kategorien, ihre Unterpunkte und die erreichten Ergebnisse werden in den folgenden Abschnitten erläutert.

	Basis *Peak*	Microsoft *Band 2*	Mobile Action *Q-Band*	Pebble *Time*	Runtastic *Moment Elite*	Striiv *Fusion*	Xiaomi *MiBand*
Tracker							
Kontrollierte Sichtbarkeit	✘	✔	≈	✔	✘	✘	✘[1]
BLE-Privacy	✘	✔	✘	✘	✘	✘	✘
Kontrollierte Verbindbarkeit	✔	✔	✘	✔	✘	✘	✔
Adäquate Authentifizierung	✔	✔	✘	✔	✘	✘	≈
Manipulationsschutz	✔	✔	✘	✔	✘	✘	≈
Applikation							
Absicherung lokale Speicherung	✔	✔	✔	✔	✔	✔	✘
Code-Obfuscation	≈	✘	✔	✔	≈	✘	✔
Log-/Debug-Ausgabenfrei	✘	✘	✔	✘	✘	✘	✘
Online-Kommunikation							
Verschlüsselte Verbindung	✔	✔	≈	✔	✔	≈	≈
Manipulationsschutz	✔	≈	≈	✔	≈	≈	≈

✔ Ja ≈ Teilweise ✘ Nein

[1] immer unsichtbar nach Pairing

Tabelle 3.1: Zusammenfassung der Testergebnisse

3.1 Tracker

Die Kategorie *Tracker* umfasst relevante Kriterien, die es erlauben, das allgemeine Sicherheitsniveau des Trackers an sich einzuschätzen. Das betrifft im Speziellen und vorrangig die Sichtbarkeit und Verfolgbarkeit des Trackers für Dritte, den Schutz gegen unberechtigten Zugriff sowie den Schutz gegen Manipulation durch Dritte als auch durch den Nutzer selbst.

Kontrollierte Sichtbarkeit

Aktive Bluetooth-LE-Geräte sind standardmäßig für jedes andere Gerät sichtbar, da sie aktiv sogenannte Advertisement-Pakete versenden, die andere Geräte über ihre Anwesenheit sowie Verbindungsadresse informieren. Sicherheitstechnisch hat dieses Verhalten potentiell zwei Bedrohungsmöglichkeiten zur Folge. Zum einen stellt dieser Umstand eine gewisse Gefahr für die eigene Privatsphäre dar – Ein Gerät, welches zu jeder Zeit sichtbar ist, kann einfach selbst Opfer von ungewolltem Tracking werden. Gerade wenn das Gerät zusätzlich noch mit einer festen MAC-Adresse als Verbindungsadresse arbeitet, ist das Anfertigen von Bewegungsprofilen sehr einfach. Zum anderen ist ein stets sichtbares Gerät ein leichteres Ziel für Angriffe, da es von potentiellen Angreifern leicht entdeckt werden kann. Zwar kann das Gerät dann immer noch über geeignete Mechanismen ungewollte Verbindungen abblocken, einfacher ist es aber, bei diesen Verbindungsversuchen gar nicht erst auffindbar zu sein.
In unseren Tests haben wir überprüft, welche Produkte einen adäquaten Mechanismus umsetzen, um die Sichtbarkeit einzuschränken, wenn gerade keine Verbindung zu einem registrierten Smartphone besteht. Dabei konnte festgestellt werden, dass lediglich 3 der getesteten Produkte eine Einschränkung der Sichtbarkeit vornehmen. Bei diesen Geräten handelt es sich um die Produkte von Microsoft, Pebble und Mobile Action. Wobei nur bei den ersten beiden der Mechanismus auch tatsächlich einwandfrei funktioniert. Das *Q-Band* von Mobile Action sieht zwar eine Aktivierung über die Betätigung eines Tasters auf der Vorderseite des Trackers vor, in unseren Tests war es allerdings mehrfach möglich, das Gerät auch ohne vorherige Betätigung des Tasters zu finden. Außerdem war es in Kenntnis der physikalischen Adresse des Trackers jederzeit möglich, direkt mit ihm eine Verbindung aufzunehmen ohne vorher scannen zu müssen. Ob seine Sichtbarkeit dabei aktiviert war oder nicht, spielte keine Rolle. Alle anderen Produkte schränken ihre Sichtbarkeit standardmäßig nicht ein. Das bedeutet, dass sie grundsätzlich immer verfolgbar sind, solange keine aktive Verbindung zum Smartphone besteht. Trotzdem sind sie nicht alle automatisch angreifbar, da einige von ihnen wirksame Mechanismen implementieren, die nicht legitimierte Verbindungen von Fremdgeräten abweisen. Abschnitt 3.1 geht darauf näher ein.

BLE-Privacy

Wie im vorangegangenen Abschnitt bereits erwähnt kann das aktive Advertisement in Verbindung mit der Verwendung der echten physischen MAC-Adresse als Verbindungsadresse eine gute Möglichkeit für das gezielte Tracking eines Bluetooth-Gerätes darstellen. Um dem entgegenzuwirken wurde in Android 5.0 ein Sicherheits-Feature integriert, das eine periodische Randomisierung der Verbindungsadresse bewirkt und zusätzlich die

Verwendung der echten physikalischen MAC-Adresse als Verbindungsadresse untersagt. Auf diese Weise soll verhindert werden, dass anhand der durch das BLE-Advertisement verteilten Verbindungsadresse ein Bewegungsprofil für ein spezifisches Gerät erzeugt werden kann.
In unserem Test haben wir überprüft, welche Produkte dieses Feature bereits unterstützen und daher aktiv zum Schutz der Privatsphäre des Nutzers beitragen. Wie sich allerdings herausstellte, implementiert nur eines der getesteten Produkte dieses Feature – das Microsoft *Band 2*. Die fehlende Nutzung dieses Features bei allen anderen Produkten hat besonders für die Produkte ohne Einschränkung ihrer Sichtbarkeit die Folge, dass diese sehr leicht, über eine lange Zeit hinweg und relativ einfach nachverfolgbar werden und so die Erstellung von Bewegungsprofilen möglich wird.
In unseren Tests stellte sich allerdings auch heraus, dass das Fehlen des BLE-Privacy-Features tatsächlich praktische Vorteile bringen kann. Mit Android 5.0 kann ein Android-Smartphone nun auch als Bluetooth-LE-Peripheral verwendet werden. Das bedeutet kurz gefasst, dass es nun grundsätzlich möglich ist, die Bluetooth-Funktionalität und das Verhalten beispielsweise eines Fitness-Trackers mit einem Smartphone mit Android 5.0 nachzuempfinden: Der Device-Name kann geändert, die laufenden Services können umgesetzt, die dazugehörigen Characteristics definiert und die Logik nachimplementiert werden. Auf diese Weise ist es theoretisch möglich sich mithilfe eines Smartphones als Fitness-Tracker auszugeben. So könnte ein Angreifer als Fitness-Tracker getarnt Verbindung zu einem Telefon aufnehmen, oder ein Nutzer könnte seinen eigenen Fitness-Tracker nachempfinden, der dann aber die Gesundheitsdaten ausgibt, die der Nutzer ihm vorgibt. In der Praxis scheiterte unser Versuch jedoch, einen ‚Fake-Tracker' aufzusetzen – und zwar an der fehlenden Unterstützung der Produkt-Apps und Tracker von BLE-Privacy. Die Android API verbietet nämlich die Deaktivierung dieses Features bei der Implementierung eines BLE-Peripherals und erlaubt auch zusätzlich nicht das freie Setzen der zum Advertisement verwendeten Adresse. Da aber die meisten im Test befindlichen Produkt-Apps nach festen MAC-Adressen oder zumindest einem Adressschema suchen, um ihre zugehörigen Tracker zu identifizieren, wird der ‚Fake-Tracker' schon von vornherein aussortiert und als ‚echter' Tracker ausgeschlossen. So sind die Produkte ohne das BLE-Privacy Feature vorerst sicher vor dieser Art Spoofing.

Kontrollierte Verbindbarkeit

Um die Angreifbarkeit eines Trackers zu vermindern, ergibt es durchaus Sinn, nicht jeden Verbindungsversuch von jeglichem Gerät zu akzeptieren, sondern auch schon vor dem eigentlichen Authentifizierungsprozess bestimmte Geräte auszuschließen. Am einfachsten lässt sich ein solcher Mechanismus über ein Pairing und Bonding erreichen. Hierbei werden zwei Geräte miteinander bekannt gemacht und können, basierend auf den beim Pairing und Bonding ausgetauschten Informationen, bei einer späteren Verbindungsanfrage entscheiden, ob sie mit einem bekannten Gerät kommunizieren und die Verbindung akzeptieren oder im anderen Fall ablehnen.
In unserem Test haben wir geprüft, welche Produkte ein Pairing/Bonding oder ein ähnliches adäquates Verfahren verwenden, um sicherstellen zu können, dass keine unberechtigten Geräte eine Verbindung zum Tracker aufbauen können. Dabei waren bei vier der sieben Produkte Mechanismen zu finden, die den Tracker vor nicht legitimierten Verbindungsversuchen schützen sollen. Das Basis *Peak* oder auch das Mircosoft *Band 2* beispielsweise gehen den klassischen Weg eines exklusiven Bondings. Der Nutzer bestätigt dieses beim ersten

Verbinden mit dem Tracker und diese Verbindung wird anschließend als die einzige legitimierte betrachtet. Alle anderen Verbindungsversuche werden abgewiesen. Das Pebble *Time* lässt zwar die Nutzung mit mehreren Geräten zu, lässt sich aber jede Verbindung zu jedem neuen Gerät durch den Nutzer physisch bestätigen. Das Xiaomi *MiBand* macht es einfach aber effizient und ist, sobald es einmal mit einem Gerät das Pairing und Bonding durchlaufen hat, für andere Geräte nicht mehr sicht- und verbindbar.

Adäquate Authentifizierung

Für ein sicherheitskritisches Szenario bietet es sich grundsätzlich immer an, eine robuste Authentifizierung zu verwenden, um effektiv den Zugriff auf bestimmte Ressourcen kontrollieren und nachvollziehen zu können. Da im Fall der hier getesteten Fitness-Tracker sensible Daten in Form von Nutzerverhalten und Gesundheitsdaten verarbeitet werden, ist von der Notwendigkeit einer geeigneten Authentifizierung auszugehen. Da mittlerweile auch ausreichend belegt ist, dass diese Daten für kriminelle als auch kommerzielle Zwecke durchaus interessant sind, kann ein solches System eigentlich nicht mehr ohne eine adäquate, also ausreichend sichere, Authentifizierung ausgeliefert werden.
In unserem Test wurde überprüft, ob eine effektive Authentifizierungsmethode umgesetzt wird, die sowohl auf Seiten des Trackers durchgeführt wird, um den Zugriff durch die dazugehörige Applikation zu verifizieren, sowie durch die Applikation, um die Echtheit des Trackers zu gewährleisten. Wie in Tabelle 3.1 zu sehen ist, konnten wir im Test bei vier von sieben Produkten einen Mechanismus identifizieren, der beidseitig sicherstellen soll, dass die Authentizität der beiden Kommunikationspartner Tracker und Original-App überprüfbar ist. Auch wenn die verwendete Methode nicht in allen Fällen als robust angesehen werden kann, haben wir das bloße Vorhandensein und die korrekte Funktion gelten lassen. Gerade die Authentifizierung beim Produkt von Xiaomi ist eher simpel, da sie nur aus einer Verknüpfung der Nutzerdaten (User-ID, User-Alias, Gewicht, Größe, etc.) besteht. Es bestünde also grundsätzlich die Möglichkeit des Erratens, wenn das Format bekannt ist. Die übrigen drei Produkte von Mobile Action, Runtastic und Striiv bieten dagegen keinen funktionierenden Authentifizierungsmechanismus an, so dass die Trackerfunktionalität direkt nach der Verbindung offen liegt.

Manipulationsschutz

Wie bereits zuvor erwähnt, werden Fitness-Tracker und ähnliche Produkte heutzutage auch bereits in Szenarien verwendet, in denen die durch sie erfassten Daten nicht mehr nur der Information des Nutzers dienen, sondern auch von Versicherungen, Arbeitgebern und sogar Gerichten verwendet werden. In all diesen Einsatzbereichen kann eine mutwillige Manipulation der erfassten Daten durch einen Dritten oder auch den Nutzer selbst zu ernsthaften finanziellen und/oder juristischen Konsequenzen führen. Es ist daher absolut notwendig einen geeigneten Mechanismus zu implementieren, der eine effektive Integritätssicherung der erfassten Daten realisieren kann.
Für dieses Kriterium haben wir getestet, ob es für ein gegebenes Produkt möglich ist, die vom Tracker erfassten und lokal gespeicherten Fitnessdaten oder die Trackerfunktion an sich zu manipulieren, oder ob ein Mechanismus (Integritätssicherung oder Zugriffsschutz) existiert, der dies unterbindet. Auch hier sind es wieder die 4 Produkte von Basis, Microsoft, Pebble und Xiaomi, welche in diesem Bereich einen mindestens

grundsätzlichen Schutz umsetzen. Für die ersten 3 Produkte konnte im Test keine Möglichkeit der Manipulation der gespeicherten Daten noch der Trackerfunktionalität an sich identifiziert werden. Für das Gerät von Xiaomi hingegen konnten nach Überwindung der relativ schwachen Authentifizierung mehrere Funktionen des Trackers manipuliert werden. So ist es möglich den Tracker vibrieren zu lassen, Weckzeiten einzustellen oder zu löschen oder gar den Tracker vollständig auf Werkseinstellungen zurückzusetzen. Da diese Funktionen nur durch eine relativ schwache Authentifizierung geschützt sind, muss hier eine teilweise Abwertung erfolgen. Bei den Geräten von Mobile Action sowie Striiv ist weder eine Authentifizierung noch eine anderweitige Absicherung vor Manipulation gegeben, sodass hier kein erkennbarer Schutz bewertet werden kann. Beim *Fusion* von Striiv konnten wir die Werte für Körperabmessungen des Nutzers auf übermenschliche Werte ändern, die dann 1:1 in die Berechnung von Distanz und Kalorienverbrauch einflossen. Bei dem Produkt von Mobile Action war es uns ebenfalls möglich, die auf dem Tracker gespeicherten Nutzerinformationen zu Gewicht, Größe, Schrittweite usw. zu ändern ohne uns vorher authentifizieren zu müssen. Diese Werte gehen auch dort direkt in die Berechnung des Kalorienverbrauchs und der zurückgelegten Strecke ein. Abbildung 3.1 zeigt eine Auflistung der Steuerbefehle für das *Q-Band*, die wir bei unseren Tests identifizieren und auch ausführen konnten. Bei dem Runtastic-Produkt war ebenfalls keine Authentifizierung feststellbar und einige Trackerfunktionen konnten auch von einem fremden Smartphone aus initiiert werden.

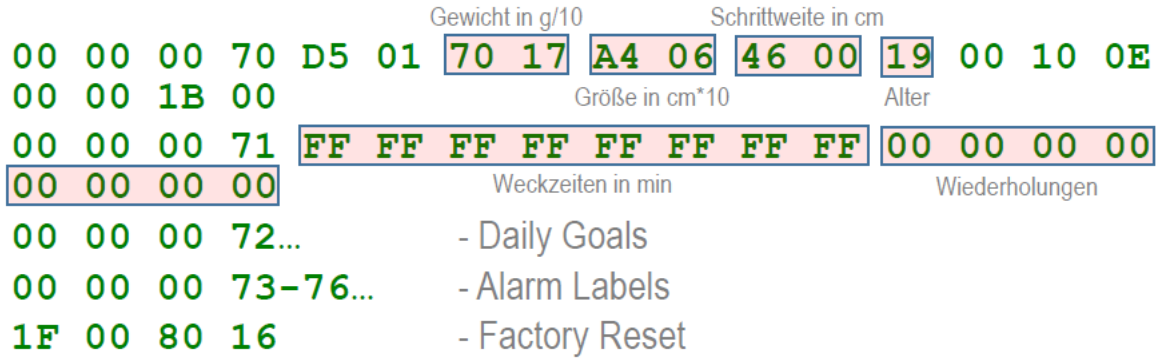

Abbildung 3.1: Beispielliste der identifizierten Steuerbefehle für das *Q-Band*, die ohne Authentifizierung ausgeführt werden konnten

3.2 Applikation

In dieser Kategorie werden die Punkte zusammengefasst, die für eine Einschätzung der Sicherheit der Produkt-Applikation relevant sind. Dabei betrachten wir, ob die lokale Speicherung (von sensiblen Daten) auf dem Smartphone gesichert erfolgt und wie einfach es für Dritte ist, die Funktionsweise der App nachzuvollziehen und zu analysieren.

Absicherung lokale Speicherung

Die Applikationen, die zu den Fitness-Trackern geliefert werden, speichern im Normalfall einiges an Daten lokal auf dem Smartphone zwischen. Dies kann notwendig sein, um die erfassten und mit dem Tracker synchronisierten Daten vorzuhalten, bis über eine Internetverbindung mit der Cloud synchronisiert werden kann, um Credentials für einen Offline-Login in der App zur Verfügung zu stellen oder einfach um Statistiken vorzuhalten. In all diesen Fällen muss sichergestellt werden, dass der Zugriff von anderen Applikationen oder in manchen Fällen dem Nutzer selbst verhindert wird. Standardmäßig werden solche Daten durch Android verwaltet und im App-Verzeichnis gesichert abgelegt. Auf Smartphones ohne Root-Rechte ist dies auch als zuverlässig sicher anzusehen. Allerdings haben Applikationen unter Android auch die Möglichkeit, ungeschützte Speicherbereiche für beispielsweise temporäre Speicherung zu verwenden. Dabei kommt es auch vor, dass eigentlich schützenswerte Nutzerinformationen in ungeschützten Bereichen abgelegt werden und somit praktisch für jede App lesbar sind.
Wir untersuchen jede App nach Daten, die (gewollt oder ungewollt) ungesichert lokal auf dem Smartphone abgespeichert werden und potentiellen Angreifern sensible Informationen über den Nutzer oder das Nutzerverhalten liefern können. Von den getesteten Produkten konnte nur bei einem eine ungeschützte Speicherung sensibler Daten festgestellt werden – beim Xiaomi MiBand. Dieses legt, vermutlich als Relikt aus der Debug-Version, ein sehr ausführliches Log über die gesamte App-Aktivität an. Enthalten sind dabei synchronisierte Daten, Online-Kommunikation (inklusive URL und übermittelten Daten), Nutzerinformationen, wie Alias, Körperabmessungen usw., die auch für den Authentifizierungsprozess verwendet werden. Dieses Log wird einfach auf die SD-Karte gespeichert und liegt dort im Klartext für jede andere App frei lesbar.

Code-Obfuscation

Wie weiter oben bereits erwähnt, ist es notwendig den Tracker und den Zugriff auf die von ihm erfassten Daten zu sichern. Das schließt den oben genannten Manipulationsschutz und die Authentifizierung mit ein. Die Mechanismen, die diese Aspekte umsetzen, sind nur selten streng nach dem Kerckhoff'schen Prinzip [Kerckhoff, 1883] umgesetzt, und ihre Sicherheit basiert in der Praxis oft nur darauf, dass ihre Funktionsweise geheim bleibt. Ein Angreifer hat somit die Möglichkeit durch gezieltes Reverse-Engineering die entsprechenden Mechanismen zu identifizieren und zu rekonstruieren. Der beste Ansatz dafür ist die Applikation. Diese ist (für alle getesteten Produkte) frei erhältlich, und mit der reichhaltigen Auswahl an zusätzlich frei erhältlichen Reverse-Engineering-Tools ist es ein Leichtes, damit an den Quellcode (oder zumindest eine gute Repräsentation dessen als Dekompilierung) der Original-App zu gelangen. Einem Angreifer ist es dann ein Leichtes, die entsprechenden Funktionen zu finden und deren Arbeitsweise nachzuvollziehen. Code-Obfuscation kann diese Möglichkeit zwar nicht gänzlich ausschließen und dient eher als Aufwandssteigerung beim Reverse-Engineering, einen wenig versierten Angreifer kann der wesentlich erhöhte Aufwand aber abschrecken. Da dieser Schutzmechanismus auch zusätzlich einen sehr geringen Aufwand zur Umsetzung benötigt und keinerlei negativen Einfluss auf die Leistung der Applikation hat, wird er von uns als Must-Have angesehen und jede App wird daraufhin überprüft. Wie sich im Test herausstellt, setzen allerdings nur drei der sieben Apps eine ausreichend gute Obfuscation um. Dabei handelt es sich um die Apps von Mobile Action, Pebble und

Xiaomi. Das Basis- und das Runtastic-Produkt setzen zwar auch Obfuscation ein, dies aber nur teilweise und in zu geringem Maße, sodass die App-Analyse hier nicht deutlich genug gestört wird. Die Produkte von Microsoft und Striiv hingegen verwenden überhaupt keine Obfuscation und gestalten eine App-Analyse daher relativ einfach. Abbildung 3.2 zeigt einen sehr kleinen Ausschnitt (etwa 10 %) aus dem Klassendiagram einer durch Code-Obfuscation ‚verschleierten' App. Der hohe Aufwand, der bei der Suche nach einer bestimmten Funktion in einem so geschützten Quellcode erbracht werden muss, ist gut vorstellbar. Wenn dann noch die Aufrufe auf die System-API gut verschleiert sind, wächst der Aufwand in enorme Höhen.

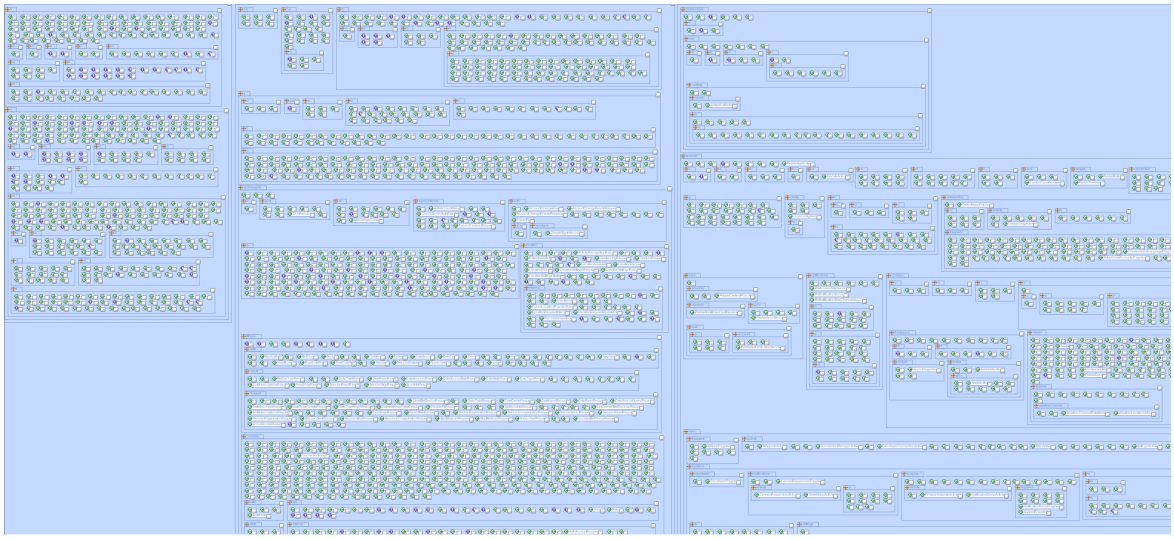

Abbildung 3.2: Ausschnitt aus dem Klassendiagramm einer App mit starker (wenn auch nicht vollständiger) Code-Obfuscation

Log-/Debug-Ausgabenfrei

In unseren Tests hat sich gezeigt, dass viele Apps noch große Teile Debug-Code enthalten und mitunter teils sehr ausführliche Log-Ausgaben von diesen verursacht werden. Im Extremfall führen solch exzessive Log-Ausgaben dazu, dass andere Mechanismen, insbesondere Code-Obfuscation, fast oder vollständig ausgehebelt werden, weil der Angreifer sämtliche benötigten Informationen direkt aus dem Log der Original-App lesen kann, ohne sich den Quellcode auch nur anschauen zu müssen. Von den sieben getesteten Produkten konnten wir nur bei einem Produkt eine fast gänzliche Freiheit von unnötigen Debug-Ausgaben feststellen und zwar beim *Q-Band* von Mobile Action. Alle anderen Produkte haben teilweise extrem ausführliche Logcat-Ausgaben und liefern damit viele Hinweise zur Funktionsweise der App oder den Stellen im Quellcode, an denen bestimmte Funktionen gesucht werden sollten.

3.3 Online-Kommunikation

Die Kategorie ‚Online-Kommunikation' stellt schließlich die Betrachtung aller eingehenden und ausgehenden Verbindungen dar, die durch das Produkt initialisiert werden. Dabei betrachten wir, ob und wie gut die Ver-

bindungen durch Verschlüsselung gesichert sind und ob es trotz Verschlüsselung grundsätzlich möglich ist, über einen Man-in-the-Middle-Angriff Daten mitzulesen oder gar unbemerkt zu manipulieren. Auch hierbei soll wieder die Gefahr der Manipulation durch Dritte sowie durch den Nutzer selbst untersucht werden.

Verschlüsselte Verbindung

Für die Übertragung von sensiblen Daten über das Internet sollte heutzutage in jedem Fall eine aktuelle Verschlüsselung verwendet werden. Die von Fitness-Trackern erfassten und gespeicherten Gesundheits- und Bewegungsdaten sind dabei keine Ausnahme. Auch wenn eine Verschlüsselung allein heute nicht mehr ausreicht, um von einer vollständigen Sicherheit ausgehen zu können, ist sie doch essentiell für ein adäquates Sicherheitskonzept. In unserem Test überprüfen wir, ob zumindest die Verbindungen zur Synchronisierung, zum Nutzer-Login und zur Aktualisierung der Firmware des Trackers über das HTTPS-Protokoll abgesichert werden. Wie schon in unserem ersten Test kann auch dieses Mal wieder festgestellt werden, dass es kein Produkt gibt, das vollständig auf die Absicherung der Internet-Kommunikation verzichtet. Und zumindest für die Verbindungen für Login und Synchronisation verwenden alle Produkte eine abgesicherte Verbindung über HTTPS. Bei 4 Produkten, namentlich Basis, Microsoft, Pebble und Runtastic, konnte überhaupt keine ungesicherte Verbindung beobachtet werden. Nach dem, was wir in unserem Test feststellen konnten, kommunizieren diese Produkte ausschließlich über gesicherte Verbindungen. Bei den übrigen drei Produkten konnten stellenweise HTTP-Verbindungen beobachtet werden, die nicht der Authentifizierung oder Synchronisation dienten. So wurde bei Mobile Action beispielsweise die Verfügbarkeit einer Firmwareaktualisierung über HTTP angefragt, bei Xiaomi sind mehrere Verbindungen zu Drittdomains zu verzeichnen, die wahrscheinlich für statistische Zwecke genutzt werden, und bei Striiv werden einige Verbindungen lediglich über HTTP betrieben. Grundsätzlich ist in diesem Bereich ein relativ hohes Schutzniveau festzustellen, was heutzutage aber für die Kommunikation über das Internet Standard sein muss.

Manipulationsschutz

Auch wenn die Online-Kommunikation durch ein Verbindungsprotokoll mit Verschlüsselung abgesichert ist, existieren trotzdem Mittel und Wege, sich in so gesicherte Verbindungen einzuschleichen und Inhalte mitzulesen oder zu verändern. In unseren Tests verwenden wir ein Test-Setup (siehe Abschnitt 2.1), das uns ermöglicht gesicherte Verbindungen aufzubrechen und die gesendeten Inhalte abzufangen, zu verändern und weiterzusenden. Auch für diesen Fall gibt es Sicherheitsmechanismen (wie das sogenannte *SSL Pinning* [Pinning]), die solche Man-in-the-Middle-Angriffe erkennen und verhindern können. Obwohl die Notwendigkeit von beispielsweise SSL-Pinning unter Android relativ gering ist, da die Installation eines Root-Zertifikats ohne Nutzerwissen praktisch sehr schwierig ist, möchten wir in unserem Test die vorliegenden Produkte auf zusätzliche Absicherungen der verschlüsselten Verbindungen überprüfen, die eine Manipulation der übermittelten Daten verhindern sollen. Wir versuchen dabei lediglich in den gesicherten Verbindungen mitzulesen. Sollte dies erfolgreich durchführbar sein, gehen wir zumindest von einer theoretischen Chance auf Manipulation aus. Wie in Tabelle 3.1 zu sehen ist, können wir in dieser Kategorie bei nur 2 Produkten einen ausreichenden Manipulationsschutz feststellen. Bei diesen zwei Produkten von Basis und Pebble wird schon beim Login

korrekt der Man-in-the-Middle erkannt und die Verbindung unterbunden. Bei allen anderen Produkten war es uns zumindest möglich die gesicherte Verbindung mitzulesen und teilweise auch erfolgreich zu manipulieren. Abbildungen 3.3 und 3.4 zeigen einen Ausschnitt aus einem abgefangenen Authentifizierungs-Request, sowie einen Ausschnitt aus einer mitgelesenen Synchronisierung. Da wir dafür ein eigenes Root-Zertifikat auf dem Test-Smartphone installieren mussten, ist die praktisch ausgehende Bedrohung von dieser Art Angriff für Android-Geräte eher gering einzuschätzen. Da sie jedoch nicht völlig von der Hand zu weisen ist und bei Betrachtung des Nutzers als Angreifer doch sehr relevant wird, nehmen wir an dieser Stelle trotzdem eine leichte Abwertung vor.

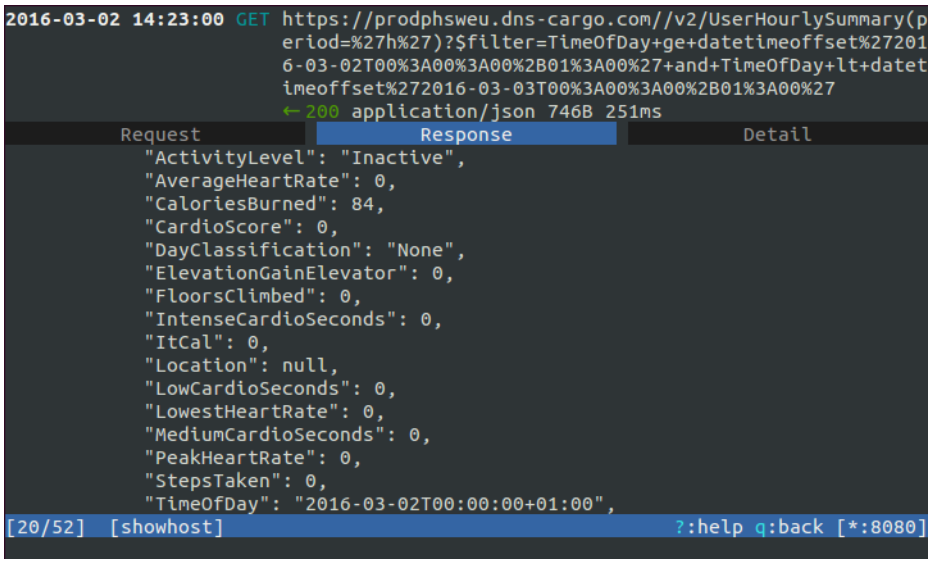

Abbildung 3.3: Ausschnitt aus Auth-Request (Screenshot aus *mitmproxy* UI)

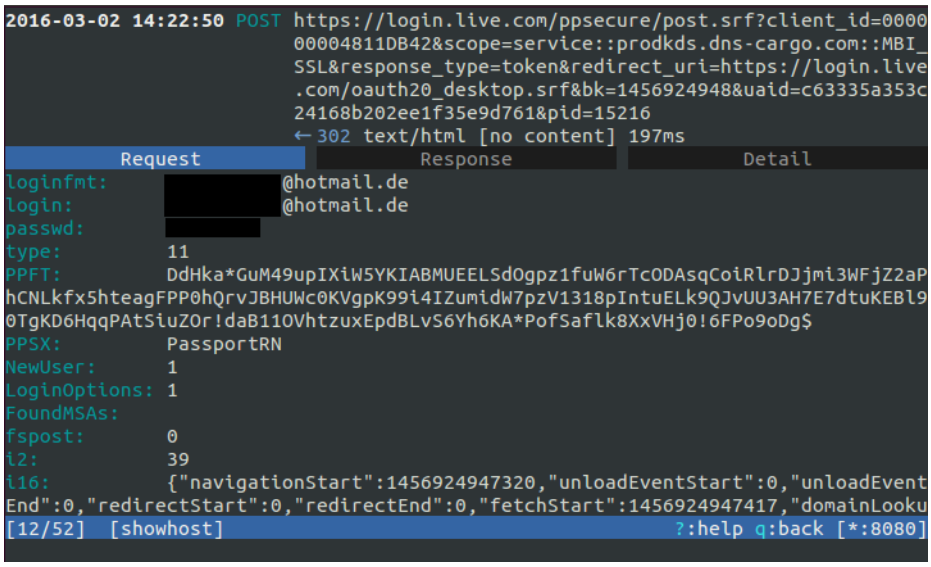

Abbildung 3.4: Ausschnitt aus Synchronisierung (Screenshot aus *mitmproxy* UI)

4 Apple Watch

Wie zu Beginn erwähnt waren unsere Testmöglichkeiten für das Produkt von Apple stärker beschränkt als bei den Android-kompatiblen Systemen. In der Folge ist auch die Ergebnistabelle 4.1 deutlich übersichtlicher.

	Apple *Watch* und *iOS* 9.2
Tracker	
Kontrollierte Sichtbarkeit	✔
BLE-Privacy	≈
Kontrollierte Verbindbarkeit	
Online-Kommunikation	
Verschlüsselte Verbindung	≈
Manipulationsschutz	≈

✔Ja ≈Teilweise ✘Nein

Tabelle 4.1: Zusammenfassung der Testergebnisse zur Apple Watch

Die Apple Watch bietet mit der Aktivierung des Flugmodus die Möglichkeit WLAN, aber auch Bluetooth, zu deaktivieren. Auch beherrscht sie prinzipiell BLE-Privacy, nach jedem von uns beobachteten Start meldete sie sich mit einer anderen MAC-Adresse. Nach dem Aktivieren des Flugmodus meldete sie sich nicht mehr. Wurde jedoch der Flugmodus wieder deaktiviert, so meldete sich die Apple Watch wieder, nun jedoch mit der echten, unveränderten MAC-Adresse der eingebauten Bluetooth-Komponente.

Auch auf der Seite der Online-Kommunikation kann keine komplette Entwarnung gegeben werden. Zwar kommuniziert das Smartphone fast ausschließlich per TLS, aber es konnten doch vereinzelt unverschlüsselte HTTP-Verbindungen beobachtet werden. Insbesondere das Update der Apple Watch auf das neue WatchOS 2.1 wurde, basierend auf unseren Beobachtungen, mit über HTTP übertragenen Dateien durchgeführt. Es muss allerdings auch beachtet werden, dass eine solche Übertragung nicht direkt ein Sicherheitsproblem darstellt. Wenn die Dateien beispielsweise auf spezielle Weise signiert oder durch gesicherte Checksummen

abgesichert sind, so könnte ein Angreifer vielleicht durch Manipulation ein Update verhindern, aber nicht seine manipulierte Version auf die Hardware schleusen.
Fast alle verschlüsselten Verbindungen sind laut unseren Ergebnissen abgesichert, sofern unser Root-Zertifikat nicht installiert war. Doch es konnten auch nicht abgesicherte TLS-Verbindungen gefunden werden. Einige dieser Verbindungen scheinen mit der Apple Watch verbunden zu sein, eine genaue Trennung war jedoch nicht möglich. So lassen sich in den POST-Daten an gsp-ssl.ls.apple.com Text-Schnipsel wie „com.apple.GeoServices“, „com.apple.NanoWeatherKit“, „Watch1,1“ und „2.1.13S6618“ finden, wobei beachtet werden sollte, dass die verwendete Apple Watch laut internen Daten Version 1.1 ist und „13S661“ in Verbindung mit WatchOS 2.1 steht. In der Antwort befinden sich dann u.a. Angaben über den Aufenthaltsort, exakt bis auf die Straßennummer. Um es noch einmal zu erwähnen, potentielle Angreifer können diese Verbindung nach unseren Tests aufbrechen und diese Daten mitlesen. Doch auch andere Hosts sind anfällig, z.B. gsp10-ssl.ls.apple.com. Im Prinzip sind aufbrechbare Verschlüsselungen nicht per se gefährlich, solange keine wichtigen Daten über diese Verbindungen laufen. Im Falle des genauen Aufenthaltsortes kann dies allerdings anders aussehen. Nach Aktivierung eines Profils mit Root-Zertifikat auf dem Smartphone ließen sich viele Verbindungen mitlesen. Für diese gesamten Analysen kam nicht `mitmproxy`, sondern ein selbst entwickeltes Programm zum Einsatz.
Eine weitere Prüfung betrifft die kontrollierte Konnektivität. Eine bereits angelernte Apple Watch mit einem anderen Account zu verbinden ist nicht gerade trivial. Ist die Watch mit einem Account verbunden, so lässt sie sich trotz Werksreset nicht einfach einem anderen Account zuordnen. Dies ist ein Diebstahlschutz, der von Apple vorgesehen ist. In unseren Tests war es allerdings nicht eindeutig möglich zu identifizieren, ob dieser Mechanismus auch tief in die Apple Watch eingebaut wurde oder ob er lediglich auf die App auf dem Smartphone beschränkt ist. Für einen Diebstahlschutz ist dies meist egal, da Kunden die originale iPhone-App mit dem verkauften Diebesgut verwenden würden. Bei bösartigen Apps von Angreifern kann dies anders aussehen. Ohne eindeutige Aussage wurde in der Tabelle 4.1 kein Eintrag vorgenommen.

Zusammenfassend kann gesagt werden, dass basierend auf den von uns untersuchten Punkten die Apple Watch durchaus eine gute Sicherheit bietet, es aber auch kleinere Probleme gibt, die im Falle der Adresse auch einen größeren Einfluss haben könnten.

5 Bedeutung für die Privatsphäre

In diesem Kapitel möchten wir kurz auf die Bedeutung der Verwendung eines Fitness-Trackers für die Privatsphäre eingehen. Wie in Tabelle 3.1 ersichtlich ist, haben wir diesen Punkt nicht direkt in die Bewertung einfließen lassen, da es absurd erscheint Geräte nach ihrem Einfluss auf die Privatsphäre zu bemessen, die dafür entwickelt werden, ihren Benutzer 24 Stunden am Tag, 7 Tage die Woche in allen Lagen seines Lebens zu überwachen und dies zu protokollieren. Jedem Nutzer sollte klar sein, dass diese Geräte Daten sammeln und jedem Nutzer sollte damit auch bewusst sein, dass diese Vorgehensweise grundsätzlich einen Einschnitt in die Privatsphäre darstellt. Allerdings können diese Tracker und gerade auch die Geräte, die eher den Funktionsumfang einer waschechten Smartwatch mitbringen, heutzutage so unglaublich viel, dass vielen Nutzern nicht klar sein dürfte, was eigentlich alles von ihnen preisgegeben wird.

Gehen wir dabei zunächst noch einmal auf die Produktapplikationen ein. Welche Rechte fordern diese überhaupt ein und welche Bedeutung haben diese für den Nutzer? Tabelle 5.1 zeigt eine unvollständige Auflistung der sensibelsten Permissions, die von den von uns getesteten Produkten eingefordert werden. Natürlich sind viele von diesen notwendig, um dem Nutzer den vollen Funktionsumfang zur Verfügung stellen zu können. So kann ein Tracker natürlich nur dann erhaltene SMS anzeigen, wenn er auch das Recht hat diese zu lesen. An dieser Stelle dürfte aber vielen Nutzern die Weitläufigkeit der erteilten Permissions schon nicht mehr vollständig klar sein. Mit den SMS-bezogenen Permissions WRITE, SEND, RECEIVE und READ hat die entsprechende App nämlich dauerhaft Zugriff auf den vollständigen SMS-Verkehr und kann sogar selbstständig SMS verfassen und absenden. Da die meisten Apps zusätzlich noch die Möglichkeit haben im Hintergrund laufen zu dürfen (über RECEIVE_BOOT_COMPLETED) ist dies theoretisch jederzeit möglich. Darüber hinaus dürfen Applikationen mit CALL_PHONE- und PROCESS_OUTGOING_CALLS-Rechten auch eigenständig Telefonate initiieren, beenden oder sogar heimlich durchführen. Zusammen mit der Möglichkeit jederzeit auf den Standort (über GPS oder verbundene Netzwerke) des Nutzers zuzugreifen, seinen Kalender zu lesen (und auch zu manipulieren) und jeden seiner Kontakte zu kennen, kommt insgesamt eine kritische Masse an Möglichkeiten zusammen. Infolgedessen wird eine solche App, die parallel dazu rund um die Uhr eine Vielzahl von sensiblen, privaten Daten sammelt, doch potentiell sehr mächtig.

Dabei muss man den Herstellern der Produkte allerdings zugutehalten, dass sie laut unserer Tests offenbar auf weitere exzessive Datensammlung, z.B. zum Nutzerverhalten oder Ähnliches, zu verzichten scheinen. Zumindest konnte in unseren Betrachtungen in dieser Hinsicht nichts Auffälliges und Ungewöhnliches festgestellt werden. Andererseits muss aber auch in Betracht gezogen werden, dass die Produkte sowieso schon auf legitimem Wege dermaßen viele Daten sammeln, dass es vermutlich auch unnötig wäre, noch mehr erfahren zu wollen bzw. der Informationsgehalt nur noch schwerlich erhöht werden könnte. Wahrscheinlich sind sich viele Nutzer auch nicht im Klaren darüber, welchen tatsächlichen Informationsgehalt ihre Daten besitzen. Natürlich kann jedermann nachlesen, welche Sensoren in einem spezifischen Gerät integriert sind, und welche

Rohdaten damit erfasst werden, ist für die meisten auch noch ersichtlich. Tatsächlich sind es aber nicht die Rohdaten, wie Puls, Schrittzahl, Kalorienverbrauch oder Standort, als Einzelinformation gesehen, sondern die Kombination dieser Informationen und die beinahe unendliche Menge an Assoziationen und Verknüpfungen, die sich daraus mithilfe von modernen Big Data-Analysesystemen und maschinellem Lernen ableiten lassen. So ist es beispielweise möglich aus den erfassten Daten Zustände des Nutzers abzulesen, die dieser vielleicht nicht unbedingt preisgeben möchte – z.B. ob er gerade mit dem Rauchen aufgehört hat oder wann er sich in einem alkoholisierten Zustand befindet [Kawamoto et al., 2014]. Selbst Verbindungen zwischen verschiedenen Nutzern lassen sich finden. So wird auch von der Möglichkeit berichtet, anhand einer Auswertung ihrer Fitnessdaten die berufliche Verbindung zwischen zwei Personen finden zu können [Tsubouchi et al., 2013]. Dabei kann man sogar getrost davon ausgehen, dass diese Beispiele nur die Spitze des Eisberges darstellen, und der Hersteller eines Fitness-Trackers eine Vielzahl von Dingen über den Nutzer weiß, dessen sich dieser selbst nicht einmal bewusst ist.

	Basis *Peak*	Microsoft *Band 2*	Mobile Action *Q-Band*	Pebble *Time*	Runtastic *Moment Elite*	Striiv *Fusion*	Xiaomi *MiBand*
App-Permission							
ACCESS_COARSE_LOCATION	✘	✔	✘	✔	✔	✔	✔
ACCESS_FINE_LOCATION	✘	✔	✘	✘	✘	✔	✔
ACCESS_GPS	✘	✘	✘	✔	✘	✘	✘
ACCESS_ASSISTED_GPS	✘	✘	✘	✔	✘	✘	✘
CALL_PHONE	✔	✔	✔	✔	✘	✘	✔
PROCESS_OUTGOING_CALLS	✘	✔	✘	✘	✘	✘	✘
READ_CONTACTS	✔	✔	✔	✔	✔	✔	✘
READ_CALENDAR	✘	✔	✘	✔	✘	✔	✘
READ_SMS	✘	✔	✘	✔	✘	✔	✘
READ_EXTERNAL_STORAGE	✔	✘	✔	✔	✔	✔	✔
RECEIVE_BOOT_COMPLETED	✔	✔	✔	✘	✔	✔	✔
RECEIVE_SMS	✘	✔	✔	✔	✘	✔	✘
SEND_SMS	✘	✔	✘	✔	✘	✘	✘
WRITE_SMS	✘	✘	✘	✔	✘	✘	✘
WRITE_EXTERNAL_STORAGE	✔	✔	✔	✔	✔	✔	✔

✔Ja ✘Nein

Tabelle 5.1: Rechte der Produktapplikationen

6 Zusammenfassung und Ausblick

In unserem zweiten Test zu Fitness-Trackern haben wir acht Produkte auf ihr allgemeines Sicherheitsniveau getestet. Dabei fielen insbesondere die Produkte von Basis, Pebble, Apple und Microsoft positiv auf. Sie leisten sich keine wirklich großen Schwächen und befinden sich insgesamt auf einem guten Schutzniveau ohne wirklich gravierende Schwachstellen. Bei den anderen Produkte konnten zumindest kleinere Schwachstellen im Bereich der Authentifizierung und dem Schutz vor Manipulation des Trackers nachgewiesen werden. Auch in diesem Test fällt wieder positiv auf, dass sich kein Produkt große Schwächen bei der Absicherung der Internet-Kommunikation leistet. Alle Produkte nutzen zumindest für die wichtigen Aspekte der Nutzerauthentifizierung und Datensynchronisation gesicherte HTTPS-Verbindungen. Wie unser um einen Man-in-the-Middle erweiterter Test allerdings zeigt, war es jedoch bei allen Produkten möglich, abgesehen von Basis und Pebble, uns in die Verbindung einzuschleichen und diese mitzulesen. Wie zuvor aufgeführt gelang uns dies aber nur mit der Installation eines eigenen Root-Zertifikats, was für einen Angreifer unter Android nicht ohne weiteres möglich ist und daher von uns auch nicht als gravierend eingeschätzt wurde.

Eine weitere Erwähnung sollte das Produkt von Xiaomi finden, welches zwar laut unserer Testauswertung (siehe Abbildung 3.1) objektiv gesehen viele Schwächen aufweist, aber als mit Abstand günstigstes Gerät im Test einen subjektiv guten Gesamteindruck hinterlässt, da in vielen Bereichen schon konzeptuell an die Sicherheit gedacht wurde. Die praktische Umsetzung des Sicherheitskonzeptes lässt dann allerdings zu wünschen übrig, wodurch das Produkt objektiv gesehen auf kein gutes Gesamtergebnis kommen kann.

Negativ fällt insbesondere das Produkt von Striiv auf. Schon im Vorjahrestest [Clausing et al., 2015] wurde das baugleiche Acer *Leap* ausgiebig getestet und wies dabei eklatante Schwächen auf. Wie der diesjährige Test zeigt, hat das baugleiche Striiv *Fusion* exakt die gleichen Probleme, wie vor einem Jahr schon das Acer *Leap*. Am Sicherheitskonzept des Basisgerätes wurde folglich nichts geändert. Mit den Erfahrungen aus dem Test des Acer *Leap* konnten wir beim Test des Striiv *Fusion* sogar noch etwas mehr in die Tiefe gehen und fanden dabei weitere gravierende Schwachstellen und Manipulationsmöglichkeiten. Für zukünftige Tests planen wir, die Bedeutung für die Privatsphäre noch näher zu beleuchten, und wir werden darüber hinaus den Test für die Online-Kommunikation weiter verschärfen, sollte dies notwendig erscheinen. Außerdem soll die Tracker-Firmware als weiterer Aspekt in den Fokus der Untersuchung rücken und die Betrachtung des Nutzers als primärer Angreifer unsere Testmethodik in stärkerem Maße beeinflussen. Dies bedeutet, dass insbesondere auch Manipulationsmöglichkeiten an den Produktapplikationen näher untersucht werden sollen.

Literaturverzeichnis

[Clausing et al., 2015] Clausing, E., Schiefer, M., Lösche, U., and Morgenstern, M. (2015). Internet of things - security evaluation of nine fitness trackers. Online. last access March 2nd, 2016. Available from: https://www.av-test.org/fileadmin/pdf/avtest_2015-06_fitness_tracker_english.pdf.

[Dörner, 2015] Dörner, S. (2015). Diese Krankenkassen bezuschussen die Apple Watch. Online. last access March 2nd, 2016. Available from: http://www.welt.de/wirtschaft/article144818188/Diese-Krankenkassen-bezuschussen-die-Apple-Watch.html.

[Fleischer, 2016] Fleischer, J. (2016). Fitness trackers can be used against you in a court of law. Online. last access March 2nd, 2016. Available from: http://www.wsbtv.com/news/news/local/fitness-trackers-can-be-used-against-you-court-law/nqHp4/.

[Gardner, 2016] Gardner, L. (2016). Fitness tracker data used in court cases. Online. last access March 2nd, 2016. Available from: http://www.news4jax.com/news/investigations/fitness-tracker-data-now-used-as-evidence-in-court-cases.

[Kawamoto et al., 2014] Kawamoto, K., Tanaka, T., and Kuriyama, H. (2014). Your activity tracker knows when you quit smoking. ACM, New York, NY, USA.

[Kerckhoff, 1883] Kerckhoff, A. (1883). La cryptographie militaire. Journal des sciences militaire.

[mitmproxy, 2014] mitmproxy (2014). mitmproxy. Online. last access March 2nd, 2016. Available from: https://mitmproxy.org/.

[Peripheral, 2014] Peripheral (2014). Android 5.0 APIs. Online. last access March 2nd, 2016. Available from: http://developer.android.com/about/versions/android-5.0.html.

[Pinning] Pinning. Certificate and public key pinning. Online. last access March 2nd, 2016. Available from: https://www.owasp.org/index.php/Certificate_and_Public_Key_Pinning.

[Tsubouchi et al., 2013] Tsubouchi, K., Kawajiri, R., and Shimosaka, M. (2013). Working-relationship detection from fitbit sensor data. Proceedings of the 2013 ACM conference on Pervasive and ubiquitous computing adjunct publication. ACM.

[Vers, 2015] Vers (2015). Mehrere Krankenkassen bezuschussen Fitness-Armbänder. Online. last access March 2nd, 2016. Available from: http://fitnessarmband.eu/krankenkassen-bezuschussen-fitness-armbaender/.