WiFi Tracking

Jonas Primbs

Studiengang: Informatik M.Sc.

Ethical Hacking Seminar, SS 2019

Eberhard-Karls-Universität Tübingen

Zusammenfassung

In dieser Seminararbeit wird anhand eines Proof of Concepts demonstriert, wie WLAN-fähige Geräte getrackt werden können. Die große Verbreitung der dafür anfälligen Geräte mit der implementierten IEEE 802.11 Standard-Familie (Smartphones, Tablets, Laptops, ...) macht dieses kostengünstige und vollständig passive Trackingverfahren äußerst attraktiv. So kann es beispielsweise von Sicherheitsdiensten an Flughäfen genutzt werden, um Unbefugte beim Umgehen von Sicherheitsabsperrungen aufzuspüren oder um Bewegungsprofile von Kunden in einem Kaufhaus anzulegen, um deren Reaktion auf Werbung im Schaufenster zu analysieren. Es wird außerdem über den Umgang mit den dadurch erfassten Daten und über Präventionsmaßnahmen diskutiert.

1 Einführung

WLAN-Endgeräte sind heute allgegenwärtig: Smartphones, Tablets, Laptops, Smartwatches, Home Assistants, SmartTVs, ... So gut wie jeder besitzt ein solches Gerät und trägt meist mindestens eines davon bei sich. Die WLAN-Funktion wird dabei selten deaktiviert. So kann sich das Endgerät automatisch mit verfügbaren WLAN-Netzwerken in der Nähe verbinden. Worüber sich wohl die wenigsten Nutzer im Klaren sind: Der Mechanismus, wie sich Endgeräte mit verfügbaren WLAN-Netzwerken verbinden, ermöglicht ein zuverlässiges Tracking des Endgeräts. Trägt der Nutzer das Endgerät, beispielsweise sein Smartphone, bei sich, so lässt sich das dabei aufgezeichnete Bewegungsprofil des Endgeräts auf den Nutzer abbilden.

Das Tracking ist dabei ein vollständig passiver Angriff. Dadurch ist es für den Nutzer unmöglich festzustellen, ob er, beziehungsweise sein Endgerät, getrackt wird oder nicht.

Derartige Funktionen sind bereits seit Jahren in vielen Enterprise-Produkten namhafter Hersteller integriert und erfreuen sich besonders in öffentlichen Einrichtungen wie Bildungseinrichtungen, Flughäfen, Einkaufszentren und öffentlichen Plätzen großer Beliebtheit. So können beispielsweise Teilnehmer bei Veranstaltungen gezählt, sowie Besucherströme und Bewegungsverhalten erfasst und analysiert werden (siehe Abbildung 1).

Abbildung 1: Screenshot einer mit Cisco Meraki erstellten Heatmap. So kann festgestellt werden, an welchen Orten sich oft viele Besucher aufhalten. Quelle: `https://documentation.meraki.com/MR/Monitoring_and_Reporting/Location_Analytics`

Wie das beschriebene Trackingverfahren

funktioniert, soll zunächst in der Theorie erklärt werden. Um zu zeigen, wie einfach und kostengünstig ein solches Verfahren in der Praxis umgesetzt werden kann, werden drei ESP32-Mikrocontrollern mit integriertem WLAN Chip programmiert, um WLAN-Geräte aufzuspüren und deren Position zu trilaterieren.

Es soll außerdem der verantwortungsvolle Umgang mit den dabei erfassten Daten diskutiert und über den derzeitigen Umgang in der Praxis informiert werden. So sollen Administratoren von WLAN-Netzwerken, deren Access Points derartige Tracking-Features unterstützten, darüber informiert werden, welche Folgen das Aktivieren dieser Features aus ethischer und rechtlicher Sicht haben kann. Es sollen außerdem Präventionsmaßnahmen diskutiert werden, die es Nutzern von WLAN-Endgeräten ermöglichen, ein solches Tracking zu unterbinden.

2 Funktionsweise

Um zu verstehen, wie das Tracking von Endgeräten funktioniert, ist es nötig, die zugrundeliegende WLAN-Protokollfamilie IEEE 802.11 genauer zu betrachten. Speziell der Teil des Protokolls, welcher für den energieeffizienten Verbindungsaufbau zu bekannten Service Set Identifiern (SSIDs = "Name des WLAN-Netzwerks") mittels sogenannter Probe Requests zuständig ist.

2.1 Verbindungsaufbau nach IEEE 802.11

Dass ein WLAN-Client (z.B. ein Smartphone) überhaupt ein WLAN-Netzwerk finden kann, broadcasten Access Points in Intervallen Management-Frames, sogenannte "Beacons"[1]. Die Beacon-Intervalle variieren je nach Konfiguration. In der Regel liegt die Intervalldauer bei circa 100 ms. In einem solchen Beacon-Frame ist besonders die SSID wichtig. Sie erlaubt es WLAN-Clients, das Netzwerk zu identifizieren.

Verlässt ein WLAN-Client nach einem erfolgreichen Verbindungsaufbau die Reichweite des WLAN-Netzwerks und kommt wieder zurück, so müsste der WLAN-Client theoretisch nichts weiter tun, als auf ein Beacon-Frame des Access Points zu warten. So würde der WLAN-Client sofort wissen, dass sich das bekannte WLAN-Netzwerk in der Nähe befindet und kann mit diesem erneut eine Verbindung aufbauen.

Das Problem daran ist jedoch, dass der WLAN-Client hierfür auf dem WLAN-Kanal des Access Points lauschen muss. Pro WLAN-Antenne kann in der Regel nur auf einem Kanal gelauscht werden. Im 2,4 GHz Frequenzband, für welchen in der Europäischen Union dreizehn Kanäle existieren, scheint dies noch einigermaßen vertretbar zu sein, im 5 GHz Frequenzband mit seinen in der Europäischen Union neunzehn Kanälen, ist dies schon weniger vertretbar[2]. Grund hierfür ist, dass man in der Regel nicht mehrere Sekunden warten möchte, bis der WLAN-Client das bekannte WLAN-Netzwerk wiedergefunden hat.

Angenommen, das letzte bekannte Beacon-Intervall wäre 100 ms. Um das WLAN-Netzwerk zuverlässig wiederzufinden, würde es nicht ausreichen, 100 ms in den letzten bekannten WLAN-Kanal zu hören, um das Beacon-Frame abzupassen: Hätte sich beispielsweise die Konfiguration des Access Points seit der letzten Begegnung geändert, oder ein anders konfigurierter Access Point strahlt dieselbe SSID aus und man möchte sich mit diesem automatisch verbinden, so müsste sicherheitshalber für einen viel größeren Zeitraum, beispielsweise eine Sekunde gelauscht werden.

Selbst auf den letzten bekannten WLAN-Kanal ist kein Verlass: Da mittlerweile fast jeder Access Point die Autokanal-Funktion unterstützt, könnte sich der Kanal der SSID je nach Kanalauslastung jederzeit ändern.

Abgesehen von der höheren Wartezeit, bis sich der WLAN-Client mit dem bekannten WLAN-Netzwerk verbunden hat, hat diese Methode noch einen weiteren, für mobile Endgeräte viel entscheidenderen Nachteil: Der Energieverbrauch. Das permanente Abhören eines WLAN-Kanals und die Auswertung der empfangenen Signale, würde die Akkulaufzeit massiv reduzieren.

Die Lösung hierfür sind die sogenannten "Probe Requests".

2.2 Probe Requests

Probe Requests ermöglichen es WLAN-Clients, aktiv nach WLAN-Netzwerken zu suchen.

Hierfür broadcastet ein WLAN-Client die SSID eines bekannten WLAN-Netzwerks. Wie jedes Management-Frame, enthalten auch Probe Requests die MAC-Adresse des Absenders und des Empfängers. Die MAC-Adresse des Empfängers lautet in diesem Fall `FF:FF:FF:FF:FF:FF`, da das Frame gebroadcastet wird. Die MAC-Adresse des Absenders ist dann die MAC-Adresse des WLAN-Clients.

Empfängt ein Access Point eine solche Probe Request und stimmt die SSID mit der auszustrahlenden SSID überein, so sendet er eine Probe Response, adressiert an die MAC-Adresse des Clients. Der WLAN-Client kann dann direkt eine Verbindung mit dem Access Point aufbauen. Die für das Antworten notwendige MAC-Adresse des Access Points ist die Absender-MAC-Adresse der Probe Request.

Diese Art des Verbindungsaufbaus ermöglicht zudem ein weiteres Feature für WLAN-Netzwerke: Das Verstecken von SSIDs. Möchte der Besitzer eines WLAN-Netzwerks beispielsweise nicht, dass sein Netzwerk von jedem WLAN-Client sofort gefunden wird, so kann das Aussenden der Beacon-Frames einfach abgeschaltet werden, ohne dass bereits verbundene WLAN-Clients das WLAN-Netzwerk nicht mehr finden. Diese sind nun nicht mehr auf die Beacon-Frames des Access Points angewiesen, da sie nur noch Probe Requests mit der SSID des bekannten WLAN-Netzwerks aussenden und der Access Point beantwortet diese mit Probe Responses.

Um die Probe Requests sichtbar zu machen, lassen sich unter Linux mit dem Tool `airmon-ng` alle WLAN-Signale auf einem virtuellen Netzwerk-Interface ausgeben und mit Wireshark mitschneiden. Mit `wlan.fc.type_subtype eq 4` kann dann nach Probe Requests gefiltert werden (siehe Abbildung 2).

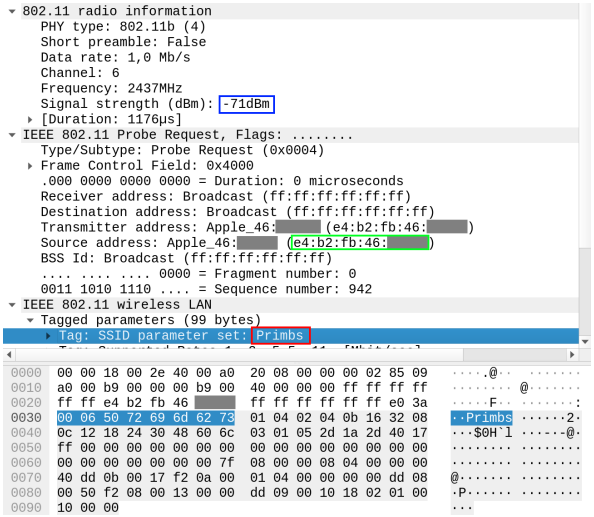

Abbildung 2: Probe Requests enthalten die MAC-Adresse des Absenders (grün) und die gesuchte SSID (rot). WLAN-Chips bieten zudem Informationen über die Signalstärke (blau). [Aus Gründen der Privatsphäre wurden die MAC-Adressen maskiert]

2.3 Missbrauch für Tracking

Dass WLAN-Clients Probe Requests aussenden, kann jedoch dazu missbraucht werden, um jene Clients zu tracken.

Da die WLAN-Clients als Absender der Probe Requests ihre MAC-Adresse verwenden und es sich dabei in der Regel um eine eindeutige hardwarespezifische Adresse handelt, muss hierfür nur auf eingehende Probe Requests gewartet und deren Absender-MAC-Adressen geloggt werden. Eine aktive Kommunikation zwischen dem abhörenden Gerät und dem WLAN-Client ist hierfür nicht erforderlich, wodurch das Tracking des Clients nicht erkannt werden kann.

Da es sich bei dem dritten bis vierundzwanzigsten Bit um einen vom Institute of Electrical and Electronics Engineer (IEEE) vergebene Organizationally Unique Identifier (OUI)[3] handelt, kann aus der MAC-Adresse zudem der Hersteller des Endgeräts ausgelesen werden. In manchen Fällen erlaubt dies auch Rückschlüsse darüber, um welchen Gerätetyp es sich bei dem getrackten Gerät handelt. Ist die MAC-Adresse beispielsweise der Firma Nokia zugeordnet, lässt sich darauf schließen, dass es sich um ein Smartphone handelt. Läuft man an einer Wohnung vorbei und empfängt eine Probe Request von einem Gerät der Firma Panasonic, so liegt die Vermutung nahe, dass in der Wohnung ein SmartTV der besagten Firma zu finden sein wird.

Die Liste der vergebenen OUIs ist auf

der Website des IEEEs öffentlich einsehbar[1]. Kostenlose APIs, die die Abfrage vereinfachen, lassen sind bei Drittanbietern finden[2].

2.4 Erhebung von Geopositionen

Mit dem Wissen aus den vorherigen Abschnitten lässt sich bisher nur erkennen, ob sich ein WLAN-Client in der Nähe des sogenannten Sniffers, also des nach Probe Requests lauschenden Geräts, befindet oder nicht. Um Bewegungsprofile der getrackten WLAN-Clients anlegen zu können, ist es jedoch notwendig, deren Geoposition herauszufinden.

Dafür liefern die WLAN-Chips die notwendigen Informationen. Selbst günstige WLAN-Chips liefern nämlich eine sogenannte Received Signal Strength Indication (RSSI)[4]. Diese liefert, wie der Name schon sagt, die Signalstärke der empfangenen WLAN-Frames und zwar auf einer logarithmischen Skala, welche in der Regel in einem Bereich von 0 dBm (sehr gut) bis -100 dBm (sehr schlecht) liegt.

Anhand der Signalstärke lässt sich nun (vorerst unter der Annahme, dass jeder WLAN-Client gleichstarke Signale aussendet) ungefähr die Distanz zum Endgerät abschätzen, da die Signalstärke mit zunehmender Distanz zum Sniffer abnimmt. Um daraus Geopositionen herzuleiten, benötigt man nun mindestens drei (zur zweidimensionalen Abbildung) oder vier Sniffer (zur dreidimensionalen Abbildung). Mittels Trilaterationsalgorithmen ist es nun möglich, die Position des zu trackenden WLAN-Clients zu bestimmen[5].

Hierfür ist es notwendig, dass die Sniffer dieselbe Probe Request aufgezeichnet haben und über die zugehörige RSSI verfügen. Außerdem muss die Position der Sniffer bekannt sein. Um Messungenauigkeiten zu reduzieren, sollten die Antennen der Sniffer nicht derart ungünstig ausgerichtet sein, dass richtfunkähnliche Effekte entstehen. Würden beispielsweise Richtfunkantennen verwendet werden und eine der Antennen würde direkt auf den WLAN-Client zeigen, während alle anderen von dem Client wegzeigen, würde dies das Messergebnis massiv verfälschen.

[1] `http://standards-oui.ieee.org/oui.txt`
[2] `https://macvendors.com/`

2.5 Zusammenfassung

In diesem Abschnitt wurden die theoretischen Grundlagen für WLAN-Tracking eingeführt: Probe Requests sind für einen energieeffizienten und schnellen Verbindungsaufbau von WLAN-Clients mit durch Access Points ausgestrahlten SSIDs konzipiert. Ohne sie wären auch versteckte SSIDs nicht realisierbar. Durch das Aussenden der Probe Requests leakt ein WLAN-Client jedoch nicht nur seine bekannten WLAN-Netzwerke, sondern auch seine hardwarespezifische MAC-Adresse.

Durch das Belauschen der Probe Requests und das Messen ihrer Signalstärke, ist es jedoch möglich, den Gerätetyp des WLAN-Clients zu erahnen und dessen Position zu bestimmen. Mit dem mehrfachen Bestimmen der Daten kann ein Bewegungsverlauf des WLAN-Clients aufgezeichnet und so auf den Nutzer des Endgeräts zurückgeführt werden.

3 Proof of Concept

Um zu demonstrieren, dass ein solches Tracking prinzipiell funktioniert und mit kostengünstiger und frei erhältlicher Hardware umsetzbar ist, wurde das in Abschnitt 2.3 und 2.4 beschriebene Verfahren umgesetzt.

3.1 Versuchsaufbau

Der Versuchsaufbau ist in Abbildung 3 grafisch dargestellt. Als Sniffer wurden hierfür drei ESP32-WROOM-32 Mikrocontrollern der Firma Espressif verwendet[3]. Diese sind bei zahlreichen Händlern bereits unter €10 erhältlich. Der Vorteil an diesem Mikrocontroller ist der geringe Anschaffungspreis, der geringe Stromverbrauch, wodurch der Sniffer über mehrere Stunden über eine Powerbank betrieben werden kann, und zusätzlich zum integrierten WLAN auch Bluetooth.

Der WLAN-Chip wird in diesem Fall dazu verwendet, die WLAN-Kanäle 1 bis 13 jeweils für 200 ms zu belauschen, um eingehende Probe Requests mitzuschneiden und auszuwerten. Hierfür ist es notwendig, den WLAN-Treiber auf den promiscuous Mode zu schalten, um Zugang zum

[3] `https://www.espressif.com/en/products/hardware/esp32/overview`

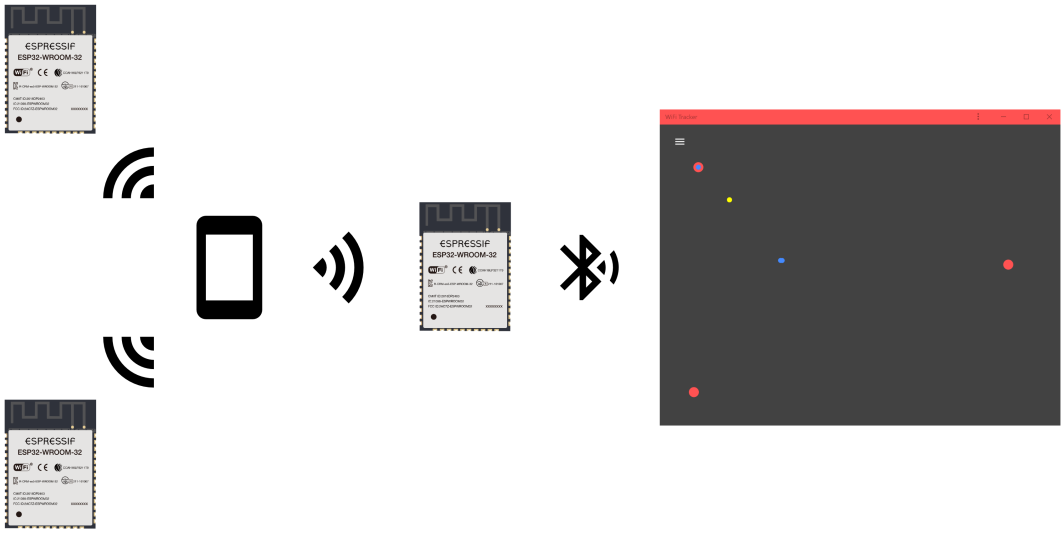

Abbildung 3: Versuchsaufbau: Drei ESP32 Mikrocontroller lauschen mit ihrem integrierten WLAN nach Probe Requests und übermitteln die Daten per Bluetooth-LE an eine Webanwendung, die die Position des getrackten WLAN-Clients trilateriert und grafisch ausgibt.

Data Link Layer (Layer 2) des OSI-Modells[6] zu erhalten. Die Zeitspanne von 200 ms erwies sich dabei aus stochastischen Gründen als sinnvoll, da sie sich in der Praxis als ein guter Kompromiss zwischen Kanallauschdauer (und somit Scandauer) und Erkennungsrate erwiesen hat.

Der integrierte Bluetooth-Chip ist für das Sniffing eigentlich nicht notwendig. Er ermöglicht jedoch eine Übertragung der Trackingdaten out-of-band, ohne hierfür den Sniffing-Vorgang unterbrechen zu müssen. Um auch hier energieeffizient zu arbeiten, wurde das Bluetooth Low Energy (BLE) Protokoll verwendet. Übertragen wurden die Informationen nach jedem abgeschlossenen Kanal-Scan im recht kompakten JSON-Format über den in BLE vorgesehen Generic Attribute Profile (GATT) Layer[7]. Die übertragenen Informationen sind im Folgenden aufgelistet:

- Timestamp
- Array von Client-Informationen:
  - RSSI
  - MAC-Adresse

Der Timestamp ermöglicht dabei eine Zuordnung, in welchem Zeitraum der Scan stattgefunden hat.

In der mit dem JavaScript-Framework Vue.js[4] geschriebenen Webanwendung kann nun ein Webbrowser, der die Web Bluetooth Schnittstelle unterstützt[5], mit den Sniffern hergestellt werden. Nach erfolgreichem Pairing, werden direkt die Tracking-Informationen übertragen und der Sniffer (als Access Point bezeichnet) wird grafisch als per Drag & Drop in einem 2D-Canvas positionierbaren Punkt dargestellt. Anhand der MAC-Adressen werden anschließend die RSSI-Werte von den verschiedenen Sniffern für einen WLAN-Client zusammengetragen und über die Positionen der Sniffer mit Formel 1 trilateriert.

$$p_c = \sum_{i=1}^{n} \frac{p_{s_i}(r_{cs_i} + 100)}{\sum_{j=1}^{n}(r_{cs_j} + 100)} \tag{1}$$

[4] `https://vuejs.org/`

[5] Webbrowser mit Web Bluetooth: `https://caniuse.com/#feat=web-bluetooth`

Die Variablen sind dabei wie folgt definiert:

- n ist die Anzahl an Sniffern, die den Client getrackt haben.
- p_c ist die Position eines Clients als Vektor.
- p_{s_i} ist die Position des Sniffers i als Vektor.
- r_{cs_i} ist der RSSI-Wert von Client zu Sniffer i.

Der Quellcode des Mikrocontrollers und der Webanwendung ist auf GitHub[6] zu finden. Eine Live-Demo der Webanwendung ist ebenfalls online verfügbar[7].

3.2 Einschränkungen

Die Resultate des Trackings sind zu Demonstrationszwecken zwar eindrucksvoll, für reale Anwendungen jedoch nur eingeschränkt zu gebrauchen. So funktioniert zwar die Erkennung von WLAN-Clients in der Nähe außerordentlich gut, jedoch nur für Geräte, die im 2,4 GHz Frequenzband funken, da der ESP32 Mikrocontroller nicht über einen Empfänger für das 5 GHz Frequenzband verfügt. An dieser Stelle wäre der Einsatz besser ausgestatteter und somit auch teurerer Hardware notwendig. Die 2,4 GHz-basierten WLAN-Standards wie IEEE 802.11b und IEEE 802.11g sind jedoch nahezu in jedem verfügbaren WLAN Access Point verfügbar und gängiger Standard. Daher ist es sehr unwahrscheinlich, dass ein WLAN-Client sich noch nie mit einem 2,4 GHz-basierten WLAN-Netzwerk verbunden hat. Somit werden wohl nur die wenigsten WLAN-Clients nur im 5 GHz-Frequenzband nach bekannten SSIDs suchen.

Die Positionsbestimmung hingegen ist sehr ungenau, und zwar aus mehreren Gründen: Einerseits müsste der Abstand zwischen den Sniffern deutlich größer sein, sodass sich die RSSI-Werte deutlicher voneinander unterscheiden. Hierfür ist jedoch die viel geringere Reichweite von Bluetooth gegenüber WLAN der limitierende Faktor: Sind die Sniffer weiter auseinander, so bricht die Bluetooth-Verbindung zu dem Gerät, auf dem die Webanwendung läuft ab und für die Trilateration fehlen die notwendigen Daten. Dies ist ein Grund, warum sich für ein solches Tracking kabelgebundene oder mobilfunkbasierte Übertragungstechniken anbieten oder, falls ein Live-Tracking nicht notwendig ist, die Speicherung für eine spätere Auswertung auf einem Lokalen Speichermedium.

Formel 1 berücksichtigt außerdem durch die logarithmische Skala der RSSI-Werte nur eingeschränkt den radialen Abfall der Signalstärke. Der dadurch im Modell implizit angenommene lineare Abfall an RSSI-Werten mit zunehmendem Abstand zum WLAN-Client entspricht schlicht nicht der Realität, ist jedoch effizient zu berechnen und für Demonstrationszwecke vollkommen ausreichend.

Auch signalschluckende und -reflektierende Gegenstände wie Wände werden nicht berücksichtigt – Ein Problem, welches sich jedoch durch eine vorherige Kalibrierung und eine bekannte Umgebung lösen ließe. Bei fest installierten Access Points, beispielsweise in Kaufhäusern wäre dies mit entsprechender Software einfach zu realisieren.

In dem Modell wird zudem davon ausgegangen, dass sich die getrackten WLAN-Clients nur innerhalb des durch die Sniffer aufgespannten Polygons befindet. Dieses Problem lässt sich jedoch ebenfalls mit aufwendigeren Trilaterationsalgorithmen lösen[8].

4 Diskussion

Wir wissen nun, dass das Tracking von WLAN-Clients selbst mit kostengünstiger Hardware möglich ist und derartiges Tracking vom WLAN-Client nicht erkannt werden kann, da es sich um ein vollständig passives Verfahren handelt. Mit Hilfe entsprechender Software lässt sich außerdem die Genauigkeit der ermittelten Positionen erhöhen.

4.1 Praktische Anwendung

Derartige Trackingverfahren sind bereits seit längerem in verbreiteten kommerziellen Systemen wie Cisco Meraki im Einsatz [9]. Das integrierte Location Analytics Feature verwendet zur Positionsbestimmung nicht nur WLAN, sondern auch Bluetooth, was aufgrund des mit zunehmender Distanz stärkeren Signalabfalls zu einer noch höheren Präzision führt. Anwendungsfälle sind etwa Flughäfen.

[6] `https://github.com/JonasPrimbs/WiFiTracker`
[7] `https://wifi.primbs.dev/`

Durch das Live-Tracking von Personen können diese etwa beim Umgehen von Sicherheitsbarrieren erkannt werden. Cisco bewirbt beispielsweise sein Meraki-System damit, dass bereits der Flughafen von Amsterdam damit vernetzt wurde. Das entsprechende Marketing-Video wurde jedoch mittlerweile wieder entfernt, es ist jedoch noch in einer Aufzeichnung eines Vortrags auf dem Chaos Communication Congress 2018 zu sehen [10].

Ein weiteres, durchaus beliebtes Szenario ist der Einsatz in Supermärkten. So können Bewegungsprofile von Kunden angelegt und deren Kaufverhalten analysiert werden [11].

Durch Tricks wie das Einloggen in einen WLAN-Hotspot mittels Facebook-Account soll zusätzlich ein Zusammenführen der Offline- auf Online-Tracking-Verläufe ermöglicht werden. Selbes kann auch durch ein Bezahlen mit Kundenkarte oder elektronische Zahlungsverfahren wie Kreditkarte oder Mobile Payment in Kombination mit Zeitstempeln und Bewegungsprofilen erreicht werden. Bezahlt beispielsweise ein Kunde zum Zeitpunkt x mit seiner Kreditkarte und weiß man, dass ein WLAN-Client sich zum Zeitpunkt x an der Kasse befunden hat, lässt sich der gesamte Bewegungsverlauf des WLAN-Clients dem Besitzer der Kreditkarte zuordnen.

Doch damit nicht genug: Die damit erfassten Bewegungsprofile mit zugeordneter Person werden oft mit Drittdiensten wie Skyfii.io geteilt, welche sich um das Zusammenführen von Online- und Offline-Trackingdaten und deren Auswertung kümmern (siehe Abbildung 4).

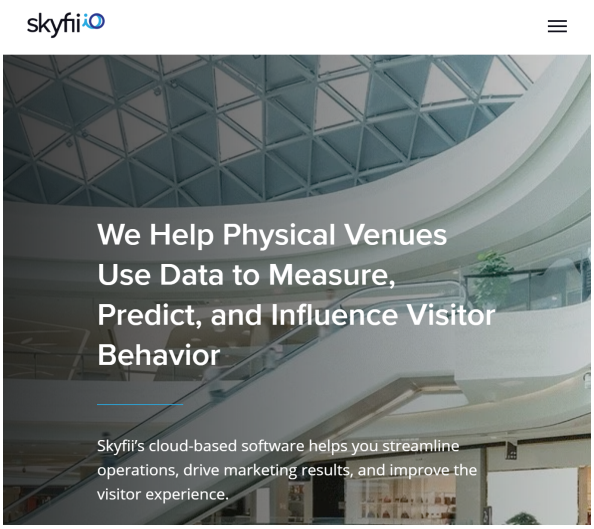

Abbildung 4: Startseite von Skyfii.io: Hier wird damit geworben, dass mit der Auswertung von “Physical Data” das Verhalten von Kunden gemessen, vorhergesagt und beeinflusst werden kann.

4.2 Rechtlicher Hintergrund

Die juristische Legitimation ist hingegen umstritten. Das Abfangen von Probe Requests könnte in Deutschland beispielsweise unter § 202b StGB. fallen, wonach das „[...] unbefugt[e] [...] [Beschaffen von] nicht für [einen] bestimmte Daten [...] aus der elektromagnetischen Abstrahlung einer Datenverarbeitungsanlage [...] mit Freiheitsstrafe bis zu zwei Jahren oder mit Geldstrafe bestraft [wird] [...].“ Dagegen steht jedoch, dass es sich bei den Probe Requests um Broadcasts handelt und diese im Falle des Anbietens eines WLAN-Hotspots schon irgendwie für den Sniffer bestimmt sind.

Laut § 98 Abs. 1 TKG hingegen „[...] hat der Anbieter bei jeder Feststellung des Standortes des Mobilfunkendgerätes den Nutzer durch eine Textmitteilung an das Endgerät [...] zu informieren“. Wie dies technisch zu realisieren ist, ist jedoch fragwürdig, schließlich kann aus der MAC-Adresse eines WLAN-Clients nicht automatisch die Mobilfunknummer (für eine SMS-Mitteilung), die E-Mail-Adresse oder Ähnliches abgeleitet werden.

Selbst wenn das Informieren über das Feststellen des Standorts erfolgen würde, so wäre laut § 6 Abs. 1 DSGVO „[d]ie Verarbeitung [von personenbezogenen Daten] [...] nur rechtmäßig, wenn [...] [d]ie betroffene Person [...] ihre Einwilligung zu der Verarbeitung der sie betreffenden personenbezogenen Daten [...] gegeben [hat]“. Dagegen wird jedoch oft damit argumentiert, dass die MAC-Adressen hierbei anonymisiert übertragen werden, wodurch es sich nicht mehr um personenbezogene Daten handelt. Die Anonymisierung der MAC-Adressen wird praktisch jedoch oft mit Hashing-Verfahren wie MD5[12] umgesetzt[10]. Mit einer p3.16xlarge Instanz bei Amazon Web Services (AWS) war es jedoch bereits 2017 möglich, 450 Milliarden

MD5-Hashes pro Sekunde zu berechnen [13]. Bei MAC-Adressen, welche über eine Länge von nur 6 Bytes verfügen folgt eine Anzahl von 256^6 verschiedenen MAC-Adressen (nicht vergebene und anderweitig reservierte miteinbezogen). Bei besagter Hashgeschwindigkeit ergibt sich daraus eine maximale Rechenzeit von unter 11 Minuten, wofür bei AWS lediglich circa 4.25 USD fällig werden[14]. Man kann daher also höchstens von einer Pseudonymisierung sprechen, nicht von einer Anonymisierung.

4.3 Präventionsmaßnahmen

Das Wissen um die massenhafte Analyse derart sensibler Daten legt die Frage nahe, wie sich der Einzelne dagegen schützen kann. Die einfache Antwort darauf lautet: WLAN und Bluetooth vollständig auszuschalten. Nur so lässt sich zuverlässig verhindern, dass Probe Requests ausgesendet und das WLAN-Gerät getrackt werden kann.

Doch es gibt auch andere Methoden, um sich dem Tracking zu entziehen: Für Android gibt es beispielsweise die App „Wi-Fi Privacy Police“[8]. Diese schränkt zumindest das Aussenden von Probe Requests bekannter WLAN-Netzwerke auf diejenigen ein, welche geographisch verfügbar sein sollten. Für die App ist zudem kein gerootetes Gerät erforderlich, eine Installation über den Google Play Store ist aber trotzdem nicht möglich.

Für iOS gibt es leider keine derartige App, auch nicht für gejailbreakte Geräte. Allerdings erkannte Apple als einer der ersten Hersteller das Privatsphäre-Problem und versuchte mit voreingestellter MAC-Randomisierung dagegen vorzugehen.

Bei MAC-Randomisierung wird als Absender-Adresse immer eine andere, zufällig generierte MAC-Adresse verwendet. So soll ein einzelner WLAN-Client durch die Verwendung wechselnder MAC-Adressen nicht eindeutig identifiziert werden können.

In Android wird dieses Feature seit Version 8 standardmäßig für Probe Requests verwendet, wenn noch keine Verbindung mit einem WLAN-Netzwerk besteht. Seit Android 9 kann MAC-Randomisierung optional auch dauerhaft in den Entwickleroptionen aktiviert werden[15]. In iOS wurde dieses Feature standardmäßig bereits in iOS 8 integriert[16].

Während des Testens des Trackings viel jedoch auf, dass dies das Tracking eines WLAN-Clients zwar ein wenig erschwert, es jedoch nicht unmöglich macht. Bleibt man beispielsweise mit seinem Smartphone stehen und es sendet mehrere Probe Requests mit unterschiedlichen randomisierten Absender-Adressen aus, so liegt der Schluss nahe, dass es sich um dasselbe Gerät handelt oder um mehrere Geräte, die derselbe Besitzer bei sich trägt. Durch die Namen der gesuchten SSIDs, ließe sich auch ein digitaler Fingerprint des Clients anlegen, welcher unabhängig von der MAC-Adresse ist. Mit Hilfe von Machine Learning Verfahren lässt sich dies sicher automatisieren.

Hinzu kommt außerdem, dass sich MAC-Randomisierung oft auch einfach erkennen lässt und selten richtig implementiert ist. Dazu kam eine Studie, in welcher sich die Autoren tiefer mit dem Thema befasst haben[17].

5 Conclusion

Es wurde gezeigt, dass das Tracking von WLAN-Clients heutzutage nicht nur möglich und mit kostengünstiger Hardware umsetzbar, sondern an öffentlichen Einrichtungen gängige Praxis ist. So nutzen es etwa Kaufhäuser, um das Kaufverhalten ihrer Kunden zu analysieren und zu beeinflussen. Die so – in der Regel widerrechtlich – erhobenen Daten werden pseudo-legitim mit Dritten geteilt, um damit präzise Profile von Nutzern anlegen können.

Technisch gibt es keine andere zuverlässige Möglichkeit, sich dem Tracking zu entziehen, als sein WLAN vollständig und überall auszuschalten. Überall aus dem Grund, da man ja nicht wissen kann, ob in der Nähe des Zuhauses getrackt wird. Aufgrund des in der EU-Datenschutz-Grundverordnung, Artikel 17 manifestierten “Rechts auf Vergessenwerden”, gibt es jedoch eine juristische Möglichkeit, erfasste Trackingdaten löschen zu lassen. Hierfür gibt es eine Website[9] des Future of Privacy Forums, welche es unter Angabe der eigenen MAC-Adresse erlaubt, dem Tracking zu widersprechen. Dieser Widerspruch

[8] Verfügbar im F-Droid Store: `https://f-droid.org/en/packages/be.uhasselt.privacypolice/`

[9] `https://optout.smart-places.org/`

gilt dann für das Erheben der Trackingdaten aller teilnehmenden (also nicht allen!) Mobile Location Analytics Firmen.

Um datenschutzrechtliche Klagen zu vermeiden sei es Administratoren von WLAN-Netzwerken zudem empfohlen, entsprechende Location Analytics Dienste abzuschalten.

Literatur

[1] Patrick Schnabel: *„WLAN-Beacons“* aus „Elektronik Kompendium“.
Unter `https://www.elektronik-kompendium.de/sites/net/2010231.htm`
(Aufgerufen am 07.08.2019)

[2] Patrick Schnabel: *„WLAN-Frequenzen und -Kanäle“* aus „Elektronik Kompendium“.
Unter `https://www.elektronik-kompendium.de/sites/net/1712061.htm`
(Aufgerufen am 07.08.2019)

[3] Wikipedia: *„MAC-Adresse - Herstellerkennung“*.
Unter `https://de.wikipedia.org/wiki/MAC-Adresse#Herstellerkennungen`
(Aufgerufen am 07.08.2019)

[4] Dr. Frederik Lipfert: *„RSSI“*.
Unter `https://www.speedcheck.org/de/wiki/rssi/`
(Aufgerufen am 07.08.2019)

[5] Markus Pinl: *„Positionsbestimmung“* aus „Seminar Mobile Systeme“, WS 2004/2005, Universität Koblenz-Landau.
Verfügbar unter `https://userpages.uni-koblenz.de/~zoebel/ws2004/Positionsbestimmung.pdf`

[6] Stefan Luber, Andreas Donner: *„Was ist das OSI-Modell?“*.
Unter `https://www.ip-insider.de/was-ist-das-osi-modell-a-605831/`
(Aufgerufen am 08.08.2019)

[7] Texas Instruments: *„Generic Attribute Profile (GATT)“*.
Unter `http://dev.ti.com/tirex/content/simplelink_cc2640r2_sdk_1_40_00_45/docs/blestack/ble_user_guide/html/ble-stack-3.x/gatt.html`
(Aufgerufen am 08.08.2019)

[8] Aiqing Zhang, Xinrong Ye, and Haifeng Hu: *„Point In Triangle Testing Based Trilateration Localization Algorithm In Wireless Sensor Networks“* aus *„TRANSACTIONS ON INTERNET AND INFORMATION SYSTEMS VOL. 6, NO. 10“*, 2012
Verfügbar unter `http://www.itiis.org/digital-library/manuscript/file/418/TIIS_Vol6No10P7Oct2012.pdf`

[9] Cisco Systems: *„Location Analytics“*
Unter `https://meraki.cisco.com/solutions/location-analytics`
(Aufgerufen am 10.08.2019)

[10] Clemens Hopfer: *„Track me if you ... oh“*
Unter `https://media.ccc.de/v/35c3chaoswest-25-track-me-if-you-oh-`
(Aufgerufen am 10.08.2019)

[11] Lisa Forster: *„Tracking im Supermarkt: Wie Händler ihre Verkäufe durch Kundenortung ankurbeln wollen“*
Unter `https://heise.de/-3973727`
(Aufgerufen am 10.08.2019)

[12] Stefan Luber und Peter Schmitz: *„Was ist MD5?“*
Unter `https://www.security-insider.de/was-ist-md5-a-811330/`
(Aufgerufen am 10.08.2019)

[13] Iraklis Mathiopoulos: *„Running hashcat v4.0.0 in Amazon's AWS new p3.16xlarge instance“*
Unter `https://medium.com/@iraklis/running-hashcat-v4-0-0-in-amazons-aws-new-p3-16xlarge-instance-e8fab4541e9b`
(Aufgerufen am 10.08.2019)

[14] Amazon Web Services: *„Amazon EC2 P3-Instances“*
Unter `https://aws.amazon.com/de/ec2/instance-types/p3/`
(Aufgerufen am 10.08.2019)

[15] Android Source: *„Privacy: MAC Randomization“*
Unter `https://source.android.com/devices/tech/connect/wifi-mac-`

randomization
(Aufgerufen am 11.08.2019)

[16] Nick Arnott: *„What's really happening with iOS 8 MAC address randomization?“* Unter https://www.imore.com/closer-look-ios-8s-mac-randomization
(Aufgerufen am 11.08.2019)

[17] Martin, Jeremy and Mayberry, Travis and Donahue, Collin and Foppe, Lucas and Brown, Lamont and Riggins, Chadwick and C. Rye, Erik and Brown, Dane: *„A Study of MAC Address Randomization in Mobile Devices and When it Fails“* aus *„Proceedings on Privacy Enhancing Technologies“*, 03/2017
Verfügbar unter https://www.researchgate.net/publication/314361145_A_Study_of_MAC_Address_Randomization_in_Mobile_Devices_and_When_it_Fails