werden die Unternehmen, die die meisten Leserdaten besitzen, „die Zukunft des Buchmarktes“ bestimmen. Das heißt, dass Firmen wie Amazon bald die Richtung in Sachen Lesestoff angeben.[7] Schützen gegen die Datensammlung kann der Leser sich nur durch Lesen eines normalen Buches.[23] Maßnahmen, die der Sicherung der Daten dienen könnten, wären u. a. neue Verschlüsselungstechnologien sowie die „Sicherheit von Speichersystemen und qualifizierte […] Zugriffs- und Berechtigungslogiken“[26].

Die Welt des digitalen Buches ist ein „goldener Käfig“ geworden. Neben Komfort und Nutzen wirken E-Book-Reader durch dauerhaftes *Tracken* ihrer Leser freiheitsberaubend.[22]

> *„Auch die Kunst hat ihre Moral und viele Gesetze dieser Moral sind dieselben wie die Gesetze gewöhnlicher Ethik oder ihnen zumindest analog.“*[1]

Referenzen

1 Huxley A (2012) Schöne neue Welt. Frankfurt a. M., 68. Auflage, S. 9, 1
2 Plöger S, Schmidt F (2015) eBook-R… Computer Bild
3 ITwissen.info (2015) E-Book-Reader. http://www.itwissen.info/E-Book-Reader-ebook-reader-eReader.html
4 Colon A, Lendino J (2015) The best ebook readers of 2015. PCMag
5 Shahd M, Lutter T (2016) Nutzung von E-Books bleibt stabil. Bitkom, 11.10.2016, https://www.bitkom.org/Presse/Presseinformation/Nutzung-von-E-Books-bleibt-stabil.html
6 Lang T (2013) E-Books lesen persönliche Daten aus. PC Magazin, 18.3.2013, http://www.pc-magazin.de/ratgeber/e-books-persoenliche-daten-datenschutz-1473671.html
7 Probst M, Trotier K (2013) Leser, mach's dir selbst! Die Zeit Nr. 06/2013, http://www.zeit.de/2013/06/Internet-Buecher-schreiben
8 Alter A (2012) Your E-Book is reading you. The Wall Street Journal, 19.7.2012, https://www.wsj.com/articles/SB10001424052702304870304577490950051438304
9 Oppmann V (2009) Die neuen Vertriebskanäle des Lesens. The European, 4.11.2009, http://www.theeuropean.de/volker-oppmann/1893-wie
10 Pursche O (2013) eBooks: Anbieter lesen fleißig mit. Computer Bild, 7.10.2013, http://www.computerbild.de/artikel/cb-News-PC-Hardware-eBook-Reader-Hersteller-lesen-mit-7625845.html
11 Amazon.de (2015) Amazon.de-Datenschutzerklärung. https://www.amazon.de/gp/help/customer/display.html/ref=hp_left_v4_sib?ie=UTF8&nodeId=201909010 (1.5.2017)
12 Müller C, Spiegel S, Ullrich F (2010) E-Books in Deutschland: Der Beginn einer neuen Gutenberg-Ära? PricewaterhouseCoopers, Sept. 2010, http://www.pwc.de/de/technologie-medien-und-telekommunikation/assets/e-books_in_deutschland_-_beginn_einer_neuen_gutenberg-aera.pdf
13 Hoffelder N (2014) Adobe is spying on users, collecting data on their ebook libraries. The Digital Reader, 6.10.2014, http://the-digital-reader.com/2014/10/06/adobe-spying-users-collecting-data-ebook-libraries/
14 Schroeder T (2013) Der Kindle liest mit: Datenschutz adé! E-Reader FAQ, 10.9.2013, http://www.ereaderfaq.de/der-kindle-liest-mit-privatsphaere-ade/
15 unwatched.org (2012) Datenschutz: Welche eBook-Reader ihre Leser tracken. Archiviert unter https://web-beta.archive.org/web/20130901133230/http://www.unwatched.org/20121216_Datenschutz_Welche_E-Book-Reader_ihre_Leser_tracken
16 Bitomsky F (2014) Das Kindle-Format: Vor- und Nachteile des Amazon-Readers. Liber Laetitia, 25.7.2014, http://liber-laetitia.de/blog/kindle-format-vor-und-nachteile-des-amazon-readers/
17 Spiegel Online (2012) Amazon sperrt Kindle-Account. 23.10.2012, http://www.spiegel.de/netzwelt/web/amazon-sperrt-account-einer-kindle-nutzerin-samt-bibliothek-a-862926.html

erschienen in der FIfF-Kommunikation, herausgegeben von FIfF e.V. - ISSN 0938-3476 www.fiff.de

18 … will E-Books boykottieren. Golem.de, …e/1106/84107.html
19 …Orwell books from Kindle. The New …www.nytimes.com/2009/07/18/technology/companies/18amazon.html
20 Knop C (2013) Amazon kennt dich schon: Vom Einkaufsparadies zum Datenverwerter. Frankfurter Allgemeine Buch, Frankfurt a. M.
21 Hentschel A (2009) Amazon kennt Sie besser als Sie sich selbst. Focus Online, 6.2.2009, http://www.focus.de/digital/computer/chip-exklusiv/tid-13299/datenschutz-amazon-kennt-sie-besser-als-sie-sich-selbst_aid_367742.html
22 Pachali D (2013) Stiftung Warentest: Viele Mängel bei AGB und Datenschutz von E-Book-Portalen. iRights.info, 26.9.2013, http://irights.info/artikel/stiftung-warentest-viele-mangel-bei-agb-und-datenschutz-von-e-book-portalen/18062
23 Haupt J (2009) EFF: Datenschutz bei eBooks mangelhaft. lesen.net, 22.12.2009, http://www.lesen.net/diskurse/eff-datenschutz-bei-ebooks-mangelhaft-1918/
24 Rijmenam M (2015) How Amazon is leveraging big data. Datafloq, https://datafloq.com/read/amazon-leveraging-big-data/517
25 OECD (2012) E-books: Developments and Policy Considerations. OECD Digital Economy Papers, Nr. 208, OECD Publishing, Paris. DOI: http://dx.doi.org/10.1787/5k912zxg5svh-en
26 Beer N (2015) Ein zentraler Ort für alle meine Daten. Zeit Online, 17.4.2015, http://www.zeit.de/politik/deutschland/2015-04/fdp-digitalisierung-datenschutz-nicola-beer

Peter Wohlgenannt

Auf der Spur digital terrestrischer Fußabdrücke

Ein Großteil der Bevölkerung aus Industrieländern ist durch mobile Geräte wie Smartphones und Tablet-Computer befähigt, ständig auf das Internet zuzugreifen. Die Bundesanstalt Statistik Österreich *hat für das Jahr 2015 erhoben, dass österreichweit 72,3% aller Personen im Alter zwischen 16 und 74 Jahren innerhalb von drei Monaten mittels einem mobilem Gerät (nachfolgend als* Station *bezeichnet) das Internet benutzt haben. In der Gruppe der 16- bis 24-Jährigen nutzten sogar 97,7 % mobiles Internet.[1] Viele dieser Personen dürften u. a. per WLAN über eigene und fremde* Access Points *(APs) den Internetzugang hergestellt haben. Betrachtet man den technischen Vorgang bzw. das für den Verbindungsaufbau benutzte Kommunikationsprotokoll, gelangt man zur Erkenntnis, dass ein mobiles Gerät auf der Suche nach verwendbaren APs stetig Klartextinformation zu dessen* MAC-Adresse *(dem eindeutigen Identifikator des verbauten WLAN-Adapters) sowie teilweise die Namen bevorzugter Netzwerke aussendet. Gelingt die Zuordnung einer MAC-Adresse zu einer bestimmten Person, können in weiterer Folge sensible Information (wie Standortdaten) zu dieser Person gesammelt bzw. bereits erhobene Daten mit ihr verknüpft werden.*

Das Protokoll IEEE 802.11

Das Protokoll 802.11 für Kommunikation in Funknetzwerken[2], hauptsächlich unter den Begriffen WLAN und Wi-Fi bekannt, wurde durch das Institute of Electrical and Electronics Engineers (IEEE) entwickelt. Die Datenübertragung erfolgt, aufgeteilt auf 14 Kanäle (wobei Kanal 14 nur in Japan verwendet wird), hauptsächlich im 2,4-GHz- und eher selten auch im 5-GHz-Band.

Eine *MAC-Adresse* (Media-Access-Control-Adresse) setzt sich aus sechs normalerweise durch Doppelpunkt voneinander getrennten Bytes in hexadezimaler Schreibweise zusammen. Über die ersten drei Bytes lässt sich der Hersteller jeder individuellen WLAN-Karte eruieren[3] (z. B. 00:07:E9:XX:XX:XX ist dem Hersteller Intel zuzuordnen). Über die Broadcast-Adresse FF:FF:FF:FF:FF:FF werden alle Geräte in einem lokalen Netzwerk adressiert. Häufig ist die MAC-Adresse auf einem am Netzwerkgerät angebrachten Sticker aufgedruckt oder kann über das Betriebssystem abgefragt werden (z. B. per Linux-Befehl *ifconfig*, Windows-Befehl *ipconfig*).

Die *SSID* (Service Set ID) ist der durch den AP-Betreiber frei wählbare Name für den AP. Es ist möglich, mehrere Namen für ein und dasselbe Gerät zu vergeben und unterschiedliche Restriktionen dafür vorzugeben (Stichwort *Virtual Local Area Networks*). Anhand der SSID wählt der Nutzer den AP, mit dem er sich verbinden möchte.

Bei der *BSSID* (Basic Service Set ID) handelt es sich um die eindeutige MAC-Adresse des AP, welche der Nutzer normalerweise nicht zu sehen bekommt – gleichnamige APs aber unterscheidbar macht. Werden mehrere APs unter derselben SSID (also demselben Namen) betrieben, um damit beispielsweise eine größere Fläche abdecken zu können, wird diese als *ESSID* (Extended SSID) bezeichnet. Der Nutzer kann nicht zwischen SSID und ESSID unterscheiden.

Die zwischen Netzwerkgeräten übertragenen Pakete werden in drei unterschiedliche Typen unterteilt:

1. *Management Frames* werden für Authentifizierung, Verbindung sowie Synchronisation benötigt und sind immer im Klartext verfügbar (z. B. Beacon Frames, Probe Request Frames, Authentication und Deauthentication Frames). Zur Informationsgewinnung und Durchführung der hier vorgestellten Angriffe werden ausschließlich die in den Management Frames enthaltenen Daten genutzt.

2. *Control Frames* müssen ebenfalls in unverschlüsselter Form vorliegen, regeln den Datenfluss und werden vor allem zur Vermeidung von Kollisionen benötigt.

3. *Data Frames* sind die eigentlichen Nutzdaten (auch als *Payload* bezeichnet), welche zumeist verschlüsselt sind und den nutzerspezifischen Inhalt befördern.

Jedes 802.11-Paket enthält unter anderem die Absender- und Ziel-MAC-Adresse in unverschlüsselter Form.

Aktive und passive Access Points

Einerseits bieten APs ihre Dienste mobilen Geräten passiv per Beacon Frame über einen Broadcast an (Abbildung 1, rechts), welcher u. a. Information über den Netzwerk-Namen und zur erforderlichen Authentifikation (offen/frei zugänglich, im Allgemeinen WEP-, WPA- bzw. WPA2-verschlüsselt) liefert. Hierbei ist anzumerken, dass der Broadcast des Netzwerk-Namens aus Sicherheitsgründen unterdrückt sein könnte. Dies bedeutet, dass der AP sich nicht anbietet, sondern an einer Verbindung interessierte Stations aktiv nach ihm suchen müssen.[4] Aktives Suchen bedeutet, dass die Station auf sämtlichen Kanälen per Probe Request nach bereits bekannten (E)SSIDs sucht und bei dieser Gelegenheit auch die MAC-Adresse mitschickt (siehe Abbildung 1, links). Die meisten Geräte speichern standardmäßig eine Liste mit APs, mit welchen sie sich schon einmal erfolgreich verbunden haben – die *Configured Network List* (*CNL)*. Diese Information liegt in den übertragenen Paketen immer unverschlüsselt vor, auch wenn die Nutzdaten durch Verschlüsselung gegen unbefugte Einsicht geschützt sind.

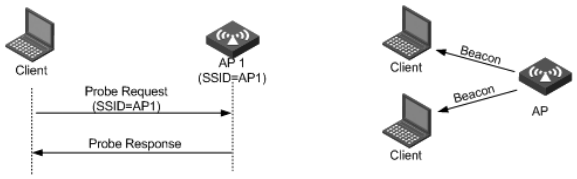

Abbildung 1: Aktive Suche nach APs (links) vs. sich passiv anbietender AP (rechts)
© New H3C Technologies Co., Limited

WLAN-Sniffen – verbreitete Hard- und Software

Das Vorhandensein dieser Klartextinformation ermöglicht es, passiv von APs gesendete Beacon Frames als auch Probe Requests und dazugehörige Probe Responses mitzulesen, was auch als WLAN-Sniffen oder Snooping (engl. für schnüffeln) bezeichnet wird. Für das passive und aktive Sniffen bedarf es spezieller Hard- und Software. Der Autor dieses Artikels verwendet hauptsächlich das Betriebssystem *Kali Linux 2016.1* mit dem vorinstallierten Softwarepaket *Aircrack-Suite*[5] sowie den bereits gepatchten Treibern für kompatible WLAN-Karten in Verbindung mit einer USB-WLAN-Karte der Marke/Type *Alfa Awus036h* (Chipset RTL8187L, Treiber r8187).

Vorbereitend muss die Netzwerkkarte dann nur noch in den *Monitor Mode* geschaltet, anschließend können die eintreffenden Datenpakete aufgezeichnet werden. Der Monitor Mode fängt sämtliche an der WLAN-Karte vorbeifliegenden Pakete ein, auch wenn sie nicht an diese adressiert sind.

Standardmäßig aktiviertes WLAN

Bei der Firmware *Android* ab der Version 4.3 alias *Jelly Bean*[6] ist das WLAN als ständig aktiv vorkonfiguriert, auch wenn der Nutzer es vermeintlich ausgeschaltet hat. Dies dient den dazu berechtigten Programmen, den Standort ohne aktiviertes GPS-Modul feststellen zu können. Der Vorteil liegt im geringeren Stromverbrauch. Möchte der Nutzer dies unterbinden, muss er in der Rubrik *Erweiterte WLAN-Einstellungen* unter dem Punkt *WLAN im Ruhemodus aktiviert lassen* die Option „Nie“ auswählen.

```
CH  9 ][ Elapsed: 1 min ][ 2007-04-26 17:41 ][ WPA handshake: 00:14:6C:7E:40:80

BSSID              PWR RXQ  Beacons    #Data, #/s  CH  MB   ENC  CIPHER AUTH ESSID

00:09:5B:1C:AA:1D   11  16       10        0    0  11  54.  OPN              NETGEAR
00:14:6C:7A:41:81   34 100       57       14    1   9  11e  WEP  WEP         bigbear
00:14:6C:7E:40:80   32 100      752       73    2   9  54   WPA  TKIP   PSK  teddy

BSSID              STATION            PWR   Rate    Lost  Packets  Probes

00:14:6C:7A:41:81  00:0F:B5:32:31:31   51   36-24    2       14
(not associated)   00:14:A4:3F:8D:13   19    0-0     0        4    mossy
00:14:6C:7A:41:81  00:0C:41:52:D1:D1   -1   36-36    0        5
00:14:6C:7E:40:80  00:0F:B5:FD:FB:C2   35   54-54    0       99    teddy
```

Abbildung 2: airodump-ng Konsolenausgabe Quelle: Ausgeschnitten aus http://www.aircrack-ng.org/doku.php?id=airodump-ng Aircrack-ng, CC BY-NC-SA 4.0

Die in Geräten großer Hersteller wie Apple und Microsoft eingesetzte Firmware dürfte sich bezüglich der WLAN-Ortung ähnlich verhalten und durch den Nutzer mehr oder weniger transparent einstellbar sein. Es folgt daraus, dass ein mobiles Gerät, beispielsweise ein Smartphone, auch dann WLAN-Signale aussenden könnte, wenn der Nutzer seiner Meinung nach die WLAN-Karte deaktiviert hat.

Reichweite von WLAN-Geräten

Die typische Reichweite eines AP beträgt ca. 35 Meter in Räumen und ca. 100 Meter im freien Gelände.[7] Auch die in den Smartphones verbauten WLAN-Karten dürften stark in ihrer Reichweite variieren und ebenfalls max. 100 Meter an Reichweite erreichen können[7], zumal diese im Gegensatz zu vielen APs über keine leistungsstarken, externen Antennen verfügen.

Zuordnung einer bekannten Zielperson zu einem noch nicht identifizierten mobilen Gerät

Wie bereits erwähnt, muss die WLAN-Karte des Angreifers zuerst in einen Modus gebracht werden, welcher sämtlichen Datenverkehr mithört. Am konkreten Beispiel der Konsolenapplikationen *Aircrack-Suite* gelingt dies mit dem Kommando `sudo airmon-ng <start|stop> <interface>` (wobei es sich bei *<interface>* um die Aircrack-Suite-kompatible WLAN-Karte handelt). Anschließend befindet sich die Karte im sogenannten *Monitor Mode*, wobei die neu hinzugekommene Schnittstelle als *mon<Nr>* bezeichnet wird. Mit dem Kommando `airodump-ng --write </path/filename> mon<Nr>` können nun die in Reichweite der WLAN-Karte befindlichen APs und Stations über die im Klartext vorliegenden Control- und Management-Frames bei ihren Aktivitäten beobachtet und in Dateien gespeichert werden. Abbildung 2, der Aircrack-Webseite entnommen, enthält Information zu den aktuell beobachteten APs und Stations (PWR = Signalstärke, CH = Kanalnummer, ENC = verwendete Verschlüsselung).

Versuchsaufbau – mobile Überwachungseinheit

Um die nachstehend beschriebenen Angriffe auch tatsächlich durchführen zu können, wären folgende Eigenschaften betreffend unsere mobile Überwachungseinheit wünschenswert:

1. Auf dem Überwachungsgerät sollte das OS Kali Linux lauffähig sein.
2. Das Überwachungsgerät sollte möglichst klein und bestenfalls unauffällig sein.

Einerseits bieten sich dafür Kleinstcomputer wie beispielsweise der *Raspberry Pi*[8] an. Andererseits würde sich speziell für die nachfolgend beschriebenen Angriffe ein modifiziertes Smartphone sehr gut eignen, da ein solches über einen Touchscreen verfügt. Dadurch kann individuell auf den Programmablauf Einfluss genommen werden (vor allem auf den Beginn und den Abbruch eines Angriffs).

Zur Implementierung wurde ein ausgedientes Smartphone der Marke/Type *LG-E960* aka *Google Nexus 4* mit dem Betriebssystem *Nethunter*[9] bespielt, welches auf (der *Debian*-Distribution) Kali Linux basiert. Außerdem wird zusätzlich ein Micro-USB-OTG-Hub, ein USB-Akkupack und eine USB-WLAN-Karte der Marke/Type *TP-Link TL-WN722N* eingesetzt.

Angriff Nr. 1 – die WLAN-AP-Replay-Attacke

Auf Abbildung 2 ist zu erkennen, dass verschiedene Stations per Probe Request nach konkreten APs Ausschau halten (z. B. mossy) und versuchen, sich damit zu verbinden. Die WLAN-AP-Replay-Attacke nutzt genau diesen Umstand aus; sie gliedert sich in zwei Phasen:

Zuerst werden an einem der Zielperson vertrauten Ort, z. B. der Privatadresse, die verfügbaren APs erhoben. Wenn diese Liste bevorzugter Netzwerke mehrere individuelle Namen für SSIDs führt, kann von einem gerätespezifischen Wi-Fi-Fingerprint gesprochen werden. Mathieu Cunche[10] hat dafür ein Shell-Skript geschrieben (welches wiederum ein Perl-Skript startet), das die APs in vereinfachter Weise lediglich mit den Attributen SSID und `0` für offen bzw. `1` für gesichert in eine Textdatei (im folgenden Beispiel nach `APFinger.txt`) schreibt .

```
root@kali: #
./WiFi\_AP\_fingerprinter.sh
APFinger.txt mon0
----------------------------------
FAU\_WiFi;1
myWifi;1
Kitzmann Guest;0
FBI\_Surveillance - Van\_02;0
```

Alternativ könnten diese Daten auch online über die Webseite *Wigle*[11] beschafft werden. Wigle steht für *Wireless Geographic*

Logging Engine und bietet über seine Webseite die Möglichkeit, im Zuge von *Wardriving*[12] gesammelte Access-Point-Daten hochzuladen und auch abzufragen. Zum Zeitpunkt der Erstellung dieses Artikels sind ca. 340 Millionen Datensätze weltweit erfasst worden.[11]

Zu einem späteren Zeitpunkt begibt man sich in die Nähe der Zielperson, um die ihr bekannten APs als verfügbar vorzutäuschen und währenddessen mitzuprotokollieren, welche Geräte versuchen, sich zu verbinden. Für die Replay-Attacke bietet sich insbesondere der Arbeitsplatz dieser Person an. Wie von Golle und Partridge[13] gezeigt, ist die Wahrscheinlichkeit, dass zwei Personen am gleichen Ort wohnen und auch arbeiten, sehr gering.

Praktisch läuft das durch Cunche in zwei Skripten automatisierte Verfahren folgendermaßen ab: Die in der Wi-Fi-Fingerprint-Textdatei gespeicherten APs werden ausgelesen und durch das zur Aircrack-Suite gehörige Tool *airbase-ng* vorgetäuscht.[10] Das hoffentlich in Reichweite befindliche Smartphone erkennt die SSIDs und versucht, sich im besten Fall als Einziger zu verbinden. Dieser Verbindungsversuch wird schlussendlich protokolliert und unter *Displaying results* im Terminal ausgegeben.

```
root@kali: #
./WiFi\_AP\_replayer.rb APFinger.txt mon0
Creating fake AP : myWifi (privacy=1)
Creating fake AP : Kitzmann usw.
Analyzing results ...
Displaying results ...
C0:EE:FB:XX:XX:XX myWifi
```

Lediglich die Station mit der MAC-Adresse *C0:EE:FB:XX:XX:XX* hat versucht, sich mit *myWifi* zu verbinden. Es könnte sich daher um unsere Zielperson handeln. Es wäre nun möglich, das Ergebnis der Zuordnung mit dem im nachfolgenden Abschnitt vorgestellten Angriff zu verifizieren, welcher jedoch auch für sich alleine stehen kann.

Angriff Nr. 2 – die „Stalker-Attacke"

Auch dieser Angriff[10] gliedert sich in zwei Phasen, diesmal in eine Beschaffungs- und eine Analysephase. Es wird im Folgenden von einem Smartphone mit aktiviertem WLAN ausgegangen, wobei es sich natürlich um jede Art von mobilem Gerät handeln könnte.

1) Im Zuge der Beschaffungsphase wird die Zielperson im öffentlichen Raum über einen nicht genau definierten, aber möglichst langen Zeitraum unauffällig verfolgt. Es muss dabei zum einen darauf geachtet werden, nicht das Signal zum betreffenden Smartphone zu verlieren, zum anderen, der Zielperson nicht aufzufallen. Die MAC-Adressen der aktiv suchenden Smartphones und die Kontaktlänge in Sekunden werden dabei in eine Textdatei gespeichert. Die Erfassung der Kontaktlänge hängt mit dem Umstand zusammen, dass im Zuge der Verfolgung vermutlich auch andere Personen in den Fokus der mobilen Überwachungseinheit geraten. Das Kommando zur Aufzeichnung lautet:

```
root@kali: #
./WiFi\_monitor.sh capture\_file.txt mon0
```

Cunche hat einen Versuch unternommen,[10] bei welchem er sich zwei Stunden lang planlos durch eine große Stadt bewegt und zwei Aufzeichnungen erstellt hat. Bei der ersten Aufzeichnung wurden 1.644, bei der zweiten 460 Geräte erfasst. In der ersten Aufzeichnung betrug die Kontaktlänge bei 80 % der erfassten Geräte weniger als 500 Sekunden; einige Kontakte bestanden jedoch auch deutlich länger.

2) Nach Beendigung der Aufzeichnung werden die dabei aufgenommenen Daten mit dem dafür geschriebenen *Ruby*-Programm *Analyze_capture.rb* hinsichtlich Kontaktdauer absteigend sortiert. An erster Stelle sollte abhängig von der Dauer der Beschaffungsphase mit hoher Wahrscheinlichkeit die gesuchte MAC-Adresse zu finden sein.

```
root@kali: #
./Analyze\_capture.rb capture\_file.txt
MAC addr : Contact Length ( sec )
[C0:EE:FB:XX:XX:XX] : 1023.129089
[1C:4B:D6:XX:XX:XX] : 13.435345
[F8:1E:DF:XX:XX:XX] : 0.12231
...
```

ARP Poisoning bzw. *Man-In-The-Middle-Attacke* betreffend die erlangte, mutmaßliche MAC-Adresse

Falls Zugang zu dem von der Zielperson verwendeten AP besteht (wenn dieser unverschlüsselt ist) oder der Angreifer sich diesen verschafft, könnte im Anschluss zur weiteren Verifizierung und ersten Datensammlung durch *ARP-Poisoning* eine *Man-In-The-Middle-Attacke* durchgeführt werden, denn nun ist die mutmaßliche MAC-Adresse bekannt.

Im LAN sind nur MAC-Adressen relevant, während IP-Pakete an die IP-Zieladresse geliefert werden. Um eine Verknüpfung zwischen diesen beiden Adresstypen herzustellen, wird das *Address Resolution Protocol (ARP)* eingesetzt. Ein Netzwerkgerät, das ein IP-Paket abzuliefern hat, kann über ARP einfach alle Hosts im LAN fragen, welche MAC-Adresse zu dieser IP-Adresse gehört. Da die Antwort auf eine solche Anfrage nicht kryptographisch geschützt ist, kann ein im LAN sitzender Angreifer alle solchen Anfragen mit seiner eigenen MAC-Adresse beantworten. Dies bezeichnet man als *ARP Spoofing* oder *ARP-Poisoning*. Wenn er schnell genug ist, kann er so den gesamten IP-Verkehr im LAN über sich umleiten, da die zeitlich erste Antwort zählt. Der Angreifer kann so im LAN als *Man-in-the-middle* agieren, d. h. er schaltet sich einfach in die Leitung zwischen zwei Teilnehmer A und B, gibt sich A gegenüber als B, und B gegenüber als A aus, und kann so jegliche Netzwerkkommunikation mitlesen und auch verändern.[14]

Eine äußerst effektive Methode, um sehr schnell an persönliche Daten zu gelangen, stellt das im Rahmen einer Bachelorarbeit im Jahr 2011 realisierte Tool *DroidSheep*[15] von Andreas Koch dar. Es handelt sich dabei um eine Android-App, mit welcher sich Session-IDs, die u. a. von Amazon, Facebook und Google eingesetzt werden, abfangen lassen. Der Angreifer ist dadurch in der Lage, dem Server gegenüber die Identität des Opfers vor-

zutäuschen und Zugang zum Account zu erlangen, was als *Session Hijacking* bezeichnet wird. Dadurch wäre es möglich, die Personaldaten inklusive Kontaktlisten und Nachrichten in Erfahrung zu bringen.

Konkrete Nutzung dieser Verknüpfung zwischen realer Person und mobilem Gerät

Die Kenntnis über die Verknüpfung zwischen MAC-Adresse und Person könnte in weiterer Folge u. a. dafür genutzt werden, um sich beim Aufscheinen der betreffenden MAC-Adresse an einem bestimmten Ort, z. B. per E-Mail, über diesen Umstand informieren zu lassen. Damit sind unter Verwendung mehrerer dieser Detektoren (in anderen Quellen meist *Drohnen* genannt) die Voraussetzungen für Standortbestimmungen und Bewegungsprofile geschaffen.

Implementierung der MAC-Adressen-Alarmierung per E-Mail

Zur Umsetzung der E-Mail-Alarmierung wurde im Zuge dieser Arbeit nach einem Log-File-Überwachungsprogramm recherchiert, welches eine Blacklist (in unserem Fall bestehend aus MAC-Adressen) entgegen nehmen und darauf basierend eine Alarmierung durchführen kann. Das *Perl*-Programm *swatch*[16] alias *swatchdog* erfüllt diese Anforderungen. Nachdem das Hauptprogramm samt einiger anderer vorausgesetzter Programme (insbesondere *tail*) installiert wurde, muss eine Konfigurationsdatei erzeugt werden, welche folgenden textuellen Aufbau hat (*swatch* Konfigurationsdatei mit einem vollständigen Eintrag):

```
watchfor /C0:EE:FB:xx:xx:xx/
        echo=red
        mail addresses=xxx, subject=xxx
watchfor ...
```

watchfor / hier steht die gesuchte MAC-Adresse /
echo=red für die Terminalausgabe in der Farbe rot
mail addresses=xxx und subject=xxx für die Mailweiterleitung per *sendmail-Dienst*

Anmerkung: Im Betreff könnte z. B. der Name einer Drohne sowie einer Zielperson inklusive Angaben zum Standort untergebracht werden.

Wie bisher beschrieben, funktioniert die Ausgabe des Suchtreffers auf der Konsole ohne weitere Installations- und Konfigurationsarbeiten. Zur Umsetzung der E-Mail-Alarmierung muss jedoch zusätzlich ein Mail-Server auf dem Überwachungsgerät eingerichtet werden. Diesbezüglich eignet sich der Mail Transfer Agent *exim4*, welcher folgende Möglichkeiten bietet:

- Die IP-Adresse für eingehende SMTP-Verbindungen auf den *localhost 127.0.0.1* festzulegen und externe Verbindungsversuche abzulehnen – was aus sicherheitstechnischer Sicht sehr wünschenswert und für den beschriebenen Zweck perfekt geeignet ist.
- Die ausgehenden E-Mails über einen externen Mail-Server, z. B. von *Gmail*, an den Empfänger-Server weiterzuleiten.

Das Programm *exim4* ist mit der richtigen Anleitung[17] schnell installiert und konfiguriert. Problematisch ist lediglich, dass die Zugangsdaten zum Postfach des Mail-Providers im Klartext in einer Textdatei abgelegt werden müssen.

Im Zuge des ersten praktischen Versuchs ist aufgefallen, dass die E-Mails nicht unmittelbar versendet werden. Nach längerem Stöbern in der Konfiguration wurde festgestellt, dass der Standardwert für die E-Mail-Warteschlange 30 Minuten beträgt. Nach Abändern des Eintrages `QUEUEINTERVAL` auf `2s` (2 Sekunden) funktionierte dann alles wie gewünscht. In diesem Zusammenhang sind die Befehle `exim -bp` zum Anzeigen der Anzahl wartender E-Mails und `exim -qff` zum Leeren der Warteschlange sehr hilfreich. Beispiel einer Konsolenausgabe nach Ausführung von *swatchdog*:

```
root@kali: # swatchdog --config-file=/etc/swatch.conf
--tail-file=capture.csv --tail-args -f
*** swatchdog version 3.2.4 (pid:1903) started at Die
Mai 31 11:24:35 CEST 2016
C0:EE:FB:XX:XX:XX, 2016-05-28 16:31:38, 2016-05-28
17:08:34, -80, 100, (not associated) ,
```

Der tail-Befehl bekommt die vom Programm *airodump-ng*[18] erzeugte csv-Datei *capture* als Log-File übergeben (siehe dazu auch Abbildung 2).

Zuordnung eines bestimmten mobilen Geräts zu einer unbekannten Person

Bisher wurde beschrieben, wie es gelingen kann, eine bekannte Person mit ihrer digitalen/elektronischen Ausstrahlung zu verschmelzen. Nun wird der umgekehrte Fall erörtert: ob und wie anhand einer bekannten MAC-Adresse die dazugehörige, unbekannte Person eruiert werden kann. Dazu wird hauptsächlich auf die frei verfügbare Software *Snoopy-ng* im Zusammenspiel mit dem Analyse-Programm *Maltego* eingegangen und auf Wilkinson[7] Bezug genommen. Dieser bezeichnet die Ortung von Personen durch deren digitale/elektromagnetische Ausstrahlung als *Digital Terrestrial Tracking* (DTT) und deren digitale/elektronische Spur als *Digital Terrestrial Footprint* (DTF). Ihm gelingt es, in sehr anschaulicher Weise den Vergleich zwischen dem eindeutigen physikalischen und dem digitalen Profil zu ziehen – nachfolgend eine sinngemäße Übersetzung aus seiner Arbeit[7]:

> *Der Digital Terrestrische Fußabdruck (DTF) ist zwischen dem physikalischen und dem Online-Fußabdruck einzuordnen. Das physikalische Tracking von Personen bezieht sich auf deren biometrische Merkmale. Online-Tracking erfolgt über individuelle, digitale Netzwerkspuren wie u. a. IP-Adressen, Cookies, Social Media Accounts. Mit Digital Terrestrischem Tracking (DTT) ist die geografische Lokalisation einer Person, basierend auf der Ausstrahlung individueller Signaturen mitgeführter Geräte, gemeint.*

Mögliche Einsatzszenarien

Zum einen gibt es die bereits angesprochene Einsatzmöglichkeit zur Standortbestimmung von mobilen Geräten (welche sich beispielsweise auf einer hinterlegten Blacklist befinden könnten). Zum anderen wäre es auch denkbar, Orte mit Drohnen zu bestücken, an denen ein bislang noch unbekannter Täter schon mehrfach Straftaten begangen hat, von dem vermutet wird, dass er auch in Zukunft solche Straftaten ausführen wird. Scheint im Zeitraum der Tatausführungen immer wieder die gleiche MAC-Adresse (bestenfalls inklusive der präferierten SSIDs) auf, kann es über diesen Ansatzpunkt zur Ausmittlung der Täterschaft kommen. Beispielsweise ist damit die Überwachung einer bestimmten Örtlichkeit möglich, an welcher ein Sexualstraftäter wiederholt seine Übergriffe setzt.

Eine weitere Einsatzmöglichkeit besteht in der Überwachung von Konsumenten, welche sich z. B. innerhalb eines Einkaufszentrums befinden. Nebst dem aktuellen Aufenthaltsort könnte anhand der Bewegungsprofile unter Umständen sogar vorausgesagt werden, wohin sie sich bewegen.

Emissionen anderer Geräte

Aus Gründen der Übersichtlichkeit und der Gefahr des Abschweifens wurde bislang nicht auf den Umstand eingegangen, dass auch andere am Körper getragene Geräte über Technologien wie *Bluetooth* oder *NFC/RFID* Emissionen verursachen (z. B. ein Headset, ein Reisepass[19], eine Bankomatkarte oder ein Fitnessarmband). Auch diese Daten können natürlich erfasst werden, wobei deren Ausstrahlungsstärke teilweise massiv schwächer als bei der WLAN-Technologie ausfällt[7], siehe dazu Tabelle 1.

Gerät	Reichweite
Wi-Fi	bis zu 100 m
Bluetooth	ca. 50 m
RFID	10 cm bis 200 m
NFC	ca. 10 cm

Tabelle 1: Reichweite von Wi-Fi, Bluetooth, RFID und NFC

Snoopy-ng

Die Software *Snoopy-ng*[20] wurde im Jahre 2012 ursprünglich als Proof of Concept (PoC) auf der Security Conference *44CON* in London von zwei Mitarbeitern (Penetration Tester) der Cyber-Security-Firma *SensePost* (u. a. Glenn Wilkinson) vorgestellt. Snoopy-ng ist ein in der Programmiersprache *Python* geschriebenes Framework, welches dafür ausgelegt ist, möglichst viele Daten von am Körper getragenen Geräten zu sniffen bzw. zu snoopen.

Dabei kommt eine *Client-Server-Infrastruktur*[21] zum Einsatz. Die Clients werden *Drohnen* genannt, wobei es sich in der Regel um kleine elektronische Geräte mit diversen Sensoren handelt. Die Drohnen (z. B. ein adaptiertes Smartphone, ein Raspberry Pi, ein Laptop, …), mit der Client-Software bespielt, nehmen lediglich den Traffic entgegen und leiten ihn an einen analysierenden Server weiter bzw. führen Befehle (z. B. NMAP-Scan, Auslieferung von Malware, Modifikationen am Traffic) von diesem aus. Siehe dazu auch Abbildung 3.

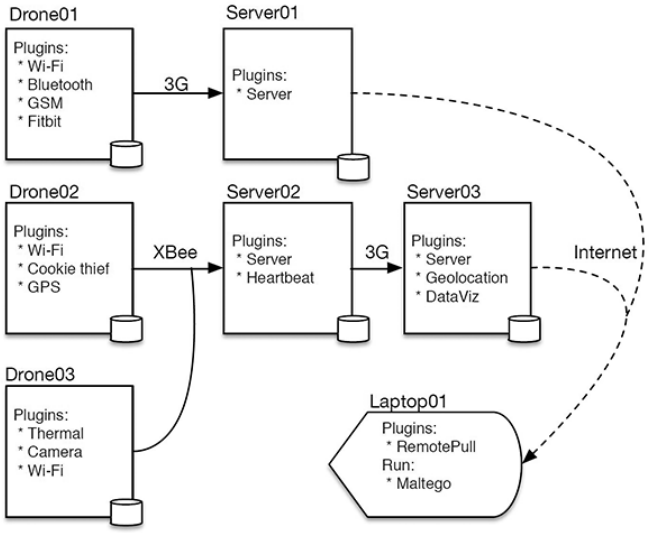

Abbildung 3: Mögliches Snoopy-Setup, bei welchem drei Drohnen ihre gesammelten Daten über unterschiedliche Technologien (3G und XBee) an zwei Server übertragen. Ein Client-Laptop ruft die Daten zum Zwecke der Auswertung bei diesen Servern ab. Pfeile mit durchgehenden Linien kennzeichnen einen **push** *(Auslieferung von Daten), während Pfeile mit gestrichelter Linie einen* **pull** *(Abholen von Daten) symbolisieren. © Glenn Wilkinson / SensePost 2014 [Ref. 7]*

Alternativ können die erfassten Daten jedoch auch in eine lokale Datenbank geschrieben werden, welche im Arbeitsverzeichnis unter der Bezeichnung *snoopy.db* abgelegt wird. Es handelt sich dabei standardmäßig um eine *SQLite*-Datenbank.

Den Opfern kann unter Vortäuschung unverschlüsselter, präferierter Netzwerke Zugriff auf das Internet gewährt werden, wobei der Server alle Daten aufzeichnet. Unter anderem sind folgende Features enthalten: *SSL Strip*, *Traffic-Inspektor* für *PDF* und *VoIP*, *Social Media Plugins* für z. B. Facebook.

In nachfolgender Textbox wird `Snoopy` mit dem WLAN-Plugin `wifi` auf Schnittstelle `mon0` unter der Bezeichnung `myDrone` und der Spezifizierung des Standortes *bregenz* gestartet:

```
root@kali: # snoopy -v -m wifi:iface=mon0 -d myDrone -l
bregenz
[+]Starting Snoopy with plugins: wifi
[+]Capturing local only.
Saving to 'sqlite:///snoopy.db' ...
```

Die dabei gespeicherten Daten können entweder über einen entsprechenden Datenbank-Viewer betrachtet oder mit Hilfe des Analysetools *Maltego* ausgewertet werden. Für das Programm *Maltego* ist im Programmpaket von *Snoopy-ng* ein Plugin (mit Symbolen) enthalten, welches die Datenbank-Aufzeichnung grafisch aufbereiten kann. Anhand der erlangten Information könnte auf die Identität der Zielperson geschlossen werden.

Abbildung 4 zeigt das Ergebnis einer von Wilkinson über das Programm *Maltego* durchgeführten Analyse hinsichtlich Stations, welche nach demselben Netzwerk mit der Bezeichnung *RBS-1-1111* suchen. Da dieser AP über die geografische Lokalisation der *Royal Bank of Scotland Branch*, Standort Liverpool Street zugeordnet werden konnte, könnte es sich bei den Besitzern der Stations um Arbeitskollegen handeln.

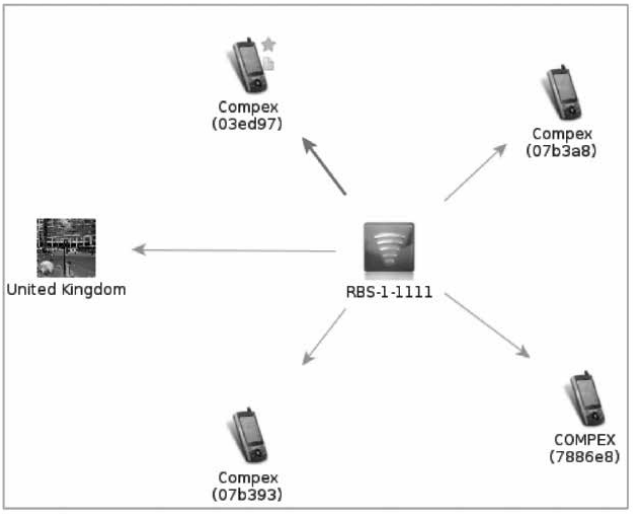

Abbildung 4: Maltego Snapshot – die als Mobiltelefon dargestellten Stations führen allesamt den AP **RBS-1-1111** *in ihrer Liste präferierter Netzwerke, welcher laut Wigle-Datenbank einem Geldinstitut in England zuzuordnen ist. © Glenn Wilkinson / SensePost 2014 [Ref. 7]*

Diskussion

Von uns allen am Körper getragene, elektronische Geräte können unsere geografische Lokation und darüber hinaus zahlreiche persönliche Informationen an einen Angreifer verraten. Sie ermöglichen es, nach erfolgreicher Zuordnung zwischen einer konkreten Person und der individuellen MAC-Adresse Bewegungsprofile zu erstellen und Standortbestimmungen durchzuführen.

Die Zuordnung zwischen Person und Gerät ist jedoch nicht unkritisch. Möglicherweise führt die Zielperson gar kein Smartphone mit sich. Oder die für den Angriff benötigten elektronischen Komponenten (z. B. der WLAN-Adapter) wurden von der Zielperson deaktiviert. Dadurch könnte es, speziell bei der vorgestellten Stalker-Attacke, zu einer falschen Zuordnung kommen. Denn hier wird lediglich die nach Verbindungszeit an erster Position gereihte MAC-Adresse als der Zielperson zurechenbar angenommen. Es wurde deshalb die Verifizierung der Zuordnung zwischen Zielperson und MAC-Adresse mittels Durchführung und Abgleich der Ergebnisse beider Angriffe und eventuell zusätzlicher Absicherung durch eine *Man-In-The-Middle*-Attacke angeregt.

Hinsichtlich der klassischen Handypeilung über den von der Zielperson verwendeten Betreiber hat das vorgestellte Verfahren hauptsächlich folgende zwei Nachteile:

1. Die Zuordnung zwischen Zielperson und Gerät ist beim vorgestellten Verfahren möglicherweise nicht mit Sicherheit zu klären. Der Telefonbetreiber knüpft die Information zur ausgelieferten SIM-Karte jedoch eindeutig an den Vertragspartner (eine mögliche Ausnahme sind Prepaid-SIM-Karten).
2. Den von uns vermutlich nur vereinzelt eingesetzten Drohnen steht ein flächendeckendes Netz an Mobilfunk-Sendemasten gegenüber, bei welchen sich der Kunde zumeist automatisch einbucht.

Dem können jedoch folgende Vorteile der vorgestellten Verfahren gegenübergestellt werden:

1. Die Veranlassung des Betreibers zur Handypeilung ist nur über staatliche Institutionen wie die Polizei möglich. Auf das Drohnen-Netzwerk (wie in Abbildung 3 gezeigt) hat dessen Administrator jederzeit und unbeschränkt Zugriff.

2. Das vorgestellte Verfahren beschränkt sich nicht nur auf die Standortdaten des Geräts und somit der Zielperson, sondern ermöglicht zusätzliche Informationsgewinnung in großem Umfang.

Dies kann zur Bekämpfung von Kriminalität und Terrorismus dienen oder aber auch für gezielte Werbung und kriminelle Zwecke missbraucht werden. Mit den vorgestellten Techniken, Gratis-Tools, günstiger Hardware und dem entsprechenden Know-how kann diese Überwachungs-Infrastruktur theoretisch durch jedermann realisiert werden. Die praktische Umsetzung des Verfahrens dürfte jedoch mit großem Aufwand für den Aufbau, die Verwaltung und Wartung verbunden sein.

Gegenmaßnahmen

Angriffe, die auf die WLAN-MAC-Adresse abzielen, können unter anderem teilweise verhindert werden durch

1. sich immer wieder ändernde MAC-Adressen,
2. das periodische Löschen von Listen präferierter Netzwerke oder
3. die Aussendung zufälliger Probe Requests zur Detektion von Replay-Angriffen.

Zu 1. und 2. hat der Programmierer Jorrit Jongma (bekannt unter dem Namen *Chainfire*) für Android das Tool *Pry-Fi*[22] entwickelt, welches nach einem pseudo-zufälligen Verfahren in kurzen zeitlichen Intervallen die MAC-Adresse ändern und abgespeicherte Listen präferierter Netzwerke löschen kann. Zu Punkt 3 beschreibt Thomas Kropeit[23] u. a. eine Methode zur Erkennung von Replay-Attacken. Er stellt fest, dass beim Senden einer zufällig generierten SSID lediglich eine Wahrscheinlichkeit von etwa 2^{-209} besteht, dass ein AP mittels Probe Response darauf antwortet. Es kann dadurch mit hoher Wahrscheinlichkeit festgestellt werden, ob gerade ein Replay-Angriff durchgeführt wird. Diese Gegenmaßnahmen werden jedoch durch folgende Umstände erschwert:

1. Ein Smartphone muss dazu im Falle von Android *gerootet*[24] sein. Änderungen, welche die MAC-Adresse betreffen, bedürfen der vollständigen Kontrolle über das Betriebssystem und dessen Ressourcen. Erfahrungsgemäß verfügt die große Masse der Nutzer jedoch nicht über die dafür erforderlichen Fertigkeiten. Außerdem könnte das *Rooten* zu Garantieverlust sowie zum *Bricken*[22] des Geräts führen; ein *Hardbrick* ist hierbei der schlimmste Fall und bedeutet, dass das Gerät komplett unbrauchbar ist und keine Möglichkeit der Reparatur mehr besteht.

2. Nicht alle Geräte erlauben ein Löschen der Netzwerklisten.

Fazit und Ausblick

Die Untersuchung der von mobilen Geräten ausgesendeten Signale zeigt, dass diese zur Gewinnung sensibler Daten verwertet werden können. Es ist dadurch möglich, Information zu einer bestimmten Person sowie über Menschenmassen zu sammeln. Die Zuordnung von eindeutigen Gerätenummern zu individuellen Personen kann mit den vorgestellten Techniken gelingen, ist jedoch keineswegs trivial.

Das ständige Mitführen gesprächiger mobiler Geräte wie Smartphones gefährdet daher die informationelle Selbstbestimmung. Dem Großteil der Bevölkerung fehlt das Wissen um die automatisiert geführte, kabellose Kommunikation und die Auswirkungen der am eigenen Gerät eingestellten (Standard-)Konfiguration. Dadurch werden die angesprochenen massiven Eingriffe in die Privatsphäre ermöglicht, welche nicht unter Kontrolle der Behörden/Gerichte oder an die Gesetze gebundener Konzerne wie Telefonbetreiber stehen. Aufbau und Einsatz einer autonomen Überwachungsinfrastruktur sind verhältnismäßig kostengünstig und unter Einsatz allgemein zugänglicher, freier Software realisierbar. Mit den vorgestellten Gegenmaßnahmen können sich Nutzer von betreffenden Endgeräten aber unter bestimmten Rahmenbedingungen dagegen schützen.

Die große Anzahl potentieller Opfer sowie die Möglichkeit, tief in den privaten Bereich gehende Information zu beschaffen, macht eine weitere Vertiefung dieses Themas, schon aus Gründen des Selbstschutzes, lohnenswert. Insbesondere möchte ich wesentlich mehr Erfahrungen im praktischen Umgang mit dem Framework Snoopy-ng und weiterer dafür verfügbarer Plugins für Sensoren wie Bluetooth und NFC sammeln. Auch die grafische Analyse mit dem Programm Maltego sowie der Aufbau eines größeren Snoopy-Setups mit mehreren, über Internet angebundenen Clients und Servern dürfte einige Herausforderungen mit sich bringen.

Referenzen

1 *Statistik Austria (2015) IKT-Einsatz in Haushalten. http://www.statistik.at*

2 *IEEE (2012) IEEE Standard for Information technology–Telecommunications and information exchange between systems–Local and metropolitan area networks–Specific requirements-Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. IEEE Std 802.11-2012 (Revision of IEEE Std 802.11-2007), März 2012, S. 1–2793*

3 *Heise Online (2016) MAC-Adressen. Heise Netze, http://www.heise.de/netze/tools/mac/ (23.6.2016)*

4 *Cunche M, Kaafar MA, Boreli R (2013) Linking wireless devices using information contained in Wi-Fi probe requests. Pervasive and Mobile Computing 11(April 2014):56–69*

5 *http://www.aircrack-ng.org/*

6 *https://www.android.com/versions/jelly-bean-4-3/*

7 *Wilkinson G (2014) Digital terrestrial tracking: The future of surveillance. DefCon 22, Las Vegas, 7.-10.8.2014, https://www.defcon.org/images/defcon-22/dc-22-presentations/Wilkinson/DEFCON-22-Glenn-Wilkinson-GRW-WP.pdf*

8 *https://www.raspberrypi.org/*

9 *https://www.kali.org/kali-linux-nethunter/*

10 *Cunche M (2013) I know your MAC address: targeted tracking of individual using Wi-Fi. International Symposium on Research in Grey-Hat Hacking, Grenoble, Nov. 2013, https://hal.archives-ouvertes.fr/file/index/docid/858324/filename/Wi-Fi_Stalking.pdf*

11 *https://wigle.net/*

12 *Jäger S (2015) Wardriving – die unterschätzte Gefahr. FIfF-Kommunikation 2015(4):30–36, https://www.fiff.de/publikationen/fiff-kommunikation/fk-2015/fk-2015-4/fk-2015-4-content/fk-2015-4-p30*

13 *Golle P, Partridge K (2009) On the anonymity of home/work location pairs. Proc. 7th Int. Conf. on Pervasive Computing, S. 390–397, Springer-Verlag, Berlin, Heidelberg*

14 *Schwenk J (2014) Sicherheit und Kryptographie im Internet. Springer, Wiesbaden*

15 *http://droidsheep.de*

16 *https://sourceforge.net/projects/swatch/*

17 *https://wiki.debian.org/GmailAndExim4*

18 *http://www.aircrack-ng.org/doku.php?id=airodump-ng*

19 *Chothia T, Smirnov V (2010) A traceability attack against e-passports. In Sion R (eds) Financial cryptography and data security. Lecture Notes in Computer Science 6052, Springer, Berlin, Heidelberg, S. 20–34*

20 *https://github.com/sensepost/snoopy-ng*

21 *https://www.sensepost.com/blog/2014/release-the-hounds-snoopy-2.0/*

22 *http://forum.xda-developers.com/showthread.php?t=2631512*

23 *Kropeit T (2015) Don't trust open hotspots: Wi-Fi hacker detection and privacy protection via smartphone. Bachelorarbeit, Ruhr-Universität Bochum, 1.3.2015, https://www.emsec.rub.de/media/attachments/files/2015/03/BA_Kropeit.pdf*

24 *Cumplido T (2015) Android rooten – Vorteile, Nachteile und CyanogenMod. Heise Download, 10.5.2015, https://www.heise.de/download/specials/Android-rooten-Vorteile-Nachteile-und-CyanogenMod-3169058. Der Begriff Root kommt aus der Linux-Welt – Android basiert auf dem Linux-Kernel – und bezeichnet den Benutzer mit erhöhten System-Rechten, vergleichbar mit dem Admin-Konto unter Windows.*

Peter Wohlgenannt

Peter Wohlgenannt trat nach seinem Abitur und anschließendem Wehrdienst im Jahr 2000 der österreichischen Bundesgendarmerie bei. Im Anschluss an die Tätigkeit als Ermittler und Spurensicherer im Kriminaldienst wurde er im Jahr 2006 als Tatortbeamter beim Landeskriminalamt eingeteilt. In den letzten acht Jahren verschob sich sein Interesse hin zur Computerkriminalität, weshalb er ab 2009 als IT-Beweissicherer tätig war. Aktuell ist er stellvertretender Leiter der Kriminaltechnik und studiert seit 2014 im Rahmen des Projekts *Open Competence Center for Cyber Security* den Bachelorstudiengang Informatik/IT-Sicherheit an der Friedrich-Alexander-Universität in Erlangen-Nürnberg.