Replay-Angriffe über Funk

Wieder eingespielt

Alexander Poth

Über Funk sind IoT-Geräte besonders anfällig. Werden etwa Embedded Devices darüber gesteuert, bietet das Angreifern die Möglichkeit zu Replay-Angriffen.

Die letzten Artikel der losen Serie zum Thema IoT-Hacking beschrieben typische Angriffsvektoren auf Hard- und Software. Flashspeicher und die darin enthaltene Firmware bieten einem Angreifer oftmals große Angriffsflächen, was aber einen physischen Zugang zum Gerät voraussetzt. Anders verhält es sich mit drahtlosen Netzen, deren Datenverkehr sich auch ohne direkten Zugriff mitschneiden und manipulieren lässt, sofern sich der Angreifer in ihrer Reichweite befindet.

Neben dem allgegenwärtigen WLAN-Standard kommen bei IoT-Geräten auch Funktechniken wie Bluetooth Low Energy, GSM/GPRS und ZigBee zum Einsatz. Oftmals werden auch proprietäre Protokolle im ISM-Band bei beispielsweise 433 oder 868 MHz verwendet. Dieser Artikel gibt einen Einblick in einen Angriff auf ein IoT-Gerät über den 433-MHz-Standard, während sich ein weiterer Artikel in einer der nächsten Ausgaben Bluetooth Low Energy widmet.

Das Identifizieren und Mitschneiden drahtlos übertragener Daten benötigt kein teures Equipment, da sich ein Großteil der Funksignalverarbeitung heute per Software erledigen lässt. Solche Software-defined Radios führen einen großen Teil der Signalverarbeitung auf einem gewöhnlichen Rechner durch. Den Empfang und das Mitschneiden sämtlicher Funksignale der verwendeten IoT-Standards übernehmen günstige Geräte wie Nooelecs NESDR SMArt v4, das auch für diesen Artikel zum Einsatz kommt (siehe Abbildung 1).

SDRs (Software-defined Radios) wie dieses basieren auf dem verbreiteten RTL-2832U-Chipsatz von Realtek und werden deshalb auch als RTL-SDR bezeichnet. Für den Einsatz des Nooelec RTL-SDR ist ein passender Treiber zu installieren, der eine Schnittstelle für eine Reihe von Programmen bereitstellt. Der mit RTL-SDR empfangbare Frequenzbereich liegt in der Regel zwischen 700 kHz und 1,7 GHz.

Zudem benötigt man eine Antenne. Ihre Länge sollte ein Teiler der Wellenlänge sein, beispielsweise die Hälfte oder ein Viertel der Wellenlänge. In Deutschland genutzte FM-Radiofrequenzen, die bei etwa 100 MHz liegen, haben eine Wellenlänge von circa drei Metern. Die optimale Antennenlänge entspricht bei einem Viertel also etwa 75 cm. Geeignete Antennen lassen sich online beziehen oder selber bauen.

Vieles nicht verschlüsselt

Neben digitalen Daten lassen sich auch analoge oder Audiosignale in sämtlichen Frequenzbereichen analysieren. UKW-Rund-, Amateur-, aber auch Flugzeugfunk

iX-TRACT

- Kommuniziert ein IoT-Gerät über Funk, benötigt ein Angreifer keinen physischen Zugang zum Gerät.
- Digitale Signale lassen sich mit geeigneten Receivern mitschneiden und per Software analysieren.
- Sind die Signale erst einmal decodiert, kann man sie auch wieder senden und damit Geräte steuern.

sind mögliche Einsatzbereiche. Wer sich einen Überblick über die genutzten Frequenzen in Deutschland verschaffen will, für den empfiehlt sich der offizielle Frequenzplan der Bundesnetzagentur (siehe ix.de/zuef). Es sei darauf hingewiesen, dass es gesetzlich untersagt ist, in gewissen Bändern ohne Genehmigung respektive Funkzeugnis zu empfangen oder zu senden. Wer mit SDRs oder anderen Geräten Funksignale empfangen oder absetzen will, sollte unbedingt die aktuelle Gesetzeslage berücksichtigen.

Für den SDR-Betrieb eignen sich verschiedene kostenlose Programme, die es für die üblichen Betriebssysteme gibt. Um einen Überblick über die unterschiedlichsten SDR-Werkzeuge zu geben, sollen ein paar gängige Tools vorgestellt werden. Den Anfang macht das Programm SDRSharp für Windows.

Die Software erlaubt das Abhören beziehungsweise Durchsuchen sämtlicher Frequenzbereiche, die die verwendete Hardware unterstützt. Neben den üblichen Radiosendern stößt man hier oft auf weitere unverschlüsselte Kommunikation im Amateur- und Betriebsfunk. Auch wenn die Suche nach solchen Frequenzen der nach der Nadel im Heuhaufen gleicht, sind mit etwas Mühe interessante Funde gewiss. Funkgespräche von Institutionen der öffentlichen Verkehrsmittel, Personenschutz, Flugverkehr oder Privatpersonen

Quelle: Nooelec

Software-defined Radios wie Nooelecs NESDR SMArt v4 verleihen einem gewöhnlichen Rechner die Fähigkeit zur Funksignalverarbeitung (Abb. 1).

lassen sich mit Geduld identifizieren. Dadurch sind Gespräche mithörbar, die eigentlich verschlüsselt sein sollten.

Mit SDRs lassen sich nicht nur analoge Radiosignale identifizieren. Digitale Signale im 433-MHz-Standard, wie sie beispielsweise Garagentorsteuerungen, Eingabegeräte wie Tastaturen, Wetterstationen, aber auch Funksteckdosen verwenden, sind häufig unverschlüsselt, also im Klartext empfangbar. Besonders Geräte im Smarthome-Bereich verwenden oft 433-MHz-Frequenzen für die unterschiedlichsten Anwendungen.

Werkzeuge für den 433-MHz-Standard

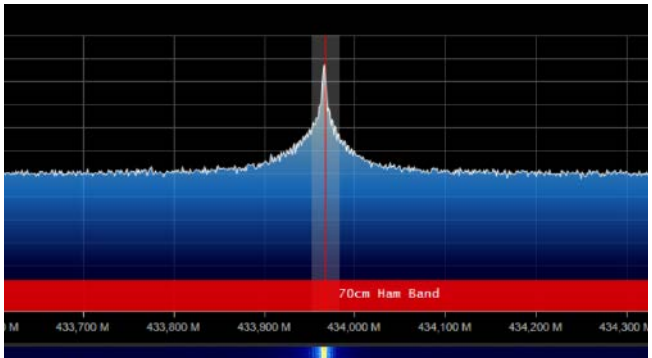
Mit entsprechenden Transmittern lassen sich auch Daten an die jeweiligen Geräte senden. Bei einer Wetterstation erfasst ein Sensor beispielsweise Temperatur und Luftfeuchtigkeit und sendet die Daten an die Station. Außenstehende können diese Datenpakete aufzeichnen und als Replay-Attacke manipuliert erneut an die Station senden. Zwar mag eine falsche Temperaturanzeige keine Gefahr für den Nutzer darstellen. Das unautorisierte Öffnen des Garagentors allerdings, das Ausschalten der Alarmanlage oder das Mitschneiden von Tastatureingaben stellt für den Nutzer ganz sicher ein hohes Risiko dar. Im Folgenden wird das Vorgehen erläutert.

Einen Überblick über Signale im 433-MHz-Standard in der unmittelbaren Umgebung verschafft man sich am einfachsten mit dem Linux-Tool rtl-sdr, das man mit `apt-get install rtl-sdr` installiert. Nach dem Einbinden des RTL-SDR-Receivers startet man den Befehl `rtl_433`. Er schneidet empfangene 433-MHz-Signale mit und decodiert sie automatisch. Die Software unterstützt einige Protokolle, die viele Hersteller verwenden. Listing 1 zeigt, wie `rtl_433` Daten, die an eine umliegende Wetterstation gesendet werden, mitliest.

Opfer des Replay-Angriffs soll eine Funksteckdose sein. Hat man den RTL-SDR-Empfänger mit dem PC oder Note-

Listing 1: rtl_433

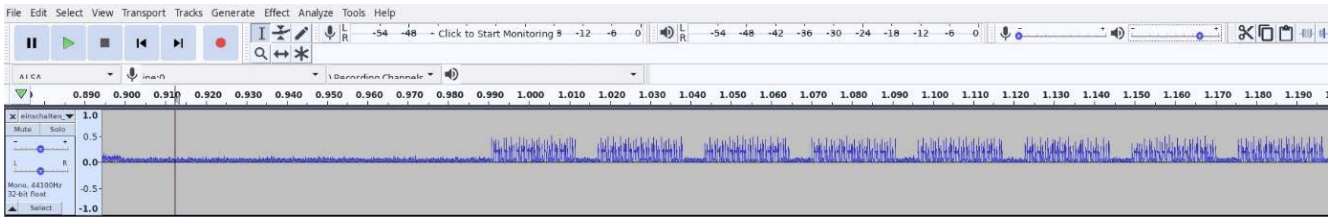
```
root@kali:~/Desktop/SDR# rtl_433
rtl_433 version unknown inputs file rtl_tcp RTL-SDR SoapySDR
Use -h for usage help and see https://triq.org/ for documentation.
Trying conf file at "rtl_433.conf"...
Trying conf file at "/root/.config/rtl_433/rtl_433.conf"...
Trying conf file at "/usr/local/etc/rtl_433/rtl_433.conf"...
Trying conf file at "/etc/rtl_433/rtl_433.conf"...
Registered 122 out of 149 device decoding protocols [ 1-4 8 11-12 15-17 19-21 23 25-26 29-36 38-60 63 67-71 73-100 102-105 108-116 119 121 124-128 130-149 ]
Detached kernel driver
Found Rafael Micro R820T tuner
Exact sample rate is: 250000.000414 Hz
[R82XX] PLL not locked!
Sample rate set to 250000 S/s.
Tuner gain set to Auto.
Tuned to 433.920MHz.
Allocating 15 zero-copy buffers
-----------------------------------------------------------------------------------------------------------------------
time      : 2020-06-04 11:43:28
model     : Schrader     type      : TPMS          flags     : 07          ID        : 1F37F5E
Pressure  : 212.5 kPa    Temperature: 23 C        Integrity : CRC
-----------------------------------------------------------------------------------------------------------------------
time      : 2020-06-04 11:43:29
model     : Schrader     type      : TPMS          flags     : 07          ID        : 1F37F5E
Pressure  : 212.5 kPa    Temperature: 23 C        Integrity : CRC
```



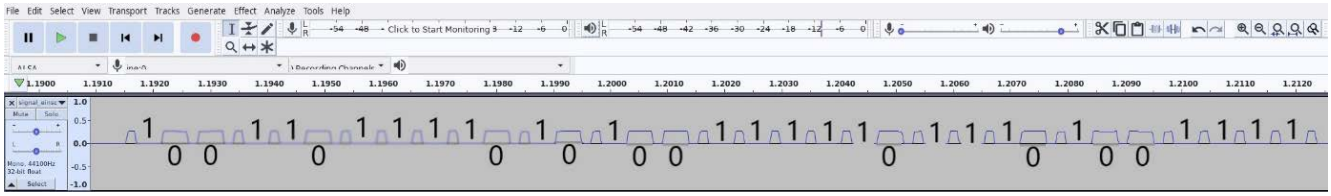
Listing 2: rtl_fm

```
rtl_fm -f 433.966M -M am -r 24000 -g 10 > signal_einschalten.dat
rtl_fm -f 433.966M -M am -r 24000 -g 10 > signal_ausschalten.dat
```

Die Aktivität der Steckdosenfernbedienung zeigt SDRSharp mit einem deutlichen Ausschlag bei etwa 433 MHz an (Abb. 2).



Audacity stellt die aufgezeichneten Signale grafisch dar. Hier zeigt sich, dass das einmalige Betätigen der Fernbedienung acht identische Datenpakete erzeugt (Abb. 3).



Innerhalb der Datenpakete lassen sich die einzelnen Bits identifizieren (Abb. 4).

Listing 3: Ein- und ausschalten

```
Einschalten:
[00] {34} 9b d4 fb a7 c0 : 10011011 11010100 11111011 10100111 11
Ausschalten:
[00] {34} 9b d4 f7 af c0 : 10011011 11010100 11110111 10101111 11
```

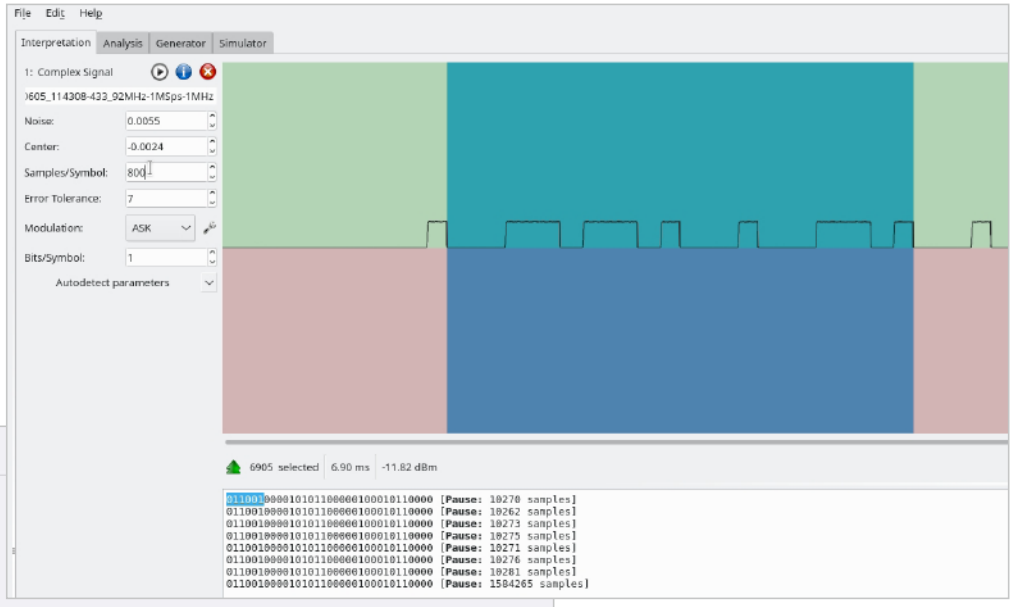
book verbunden, erzeugt die Betätigung der Steckdosenfernbedienung einen deutlichen Ausschlag in SDRSharp bei etwa 433 MHz (siehe Abbildung 2).

Replay-Attacken durchgespielt

Ebenfalls im installierten Paket rtl_sdr enthalten ist das Werkzeug rtl_fm, das die übertragenen Datenpakete im Rohformat aufzeichnet. In Listing 2 nimmt es das Signal der Fernbedienung zum Ein- und Ausschalten der Steckdosen auf und schreibt sie in die Dateien signal_einschalten.dat und signal_ausschalten.dat (siehe Listing 2).

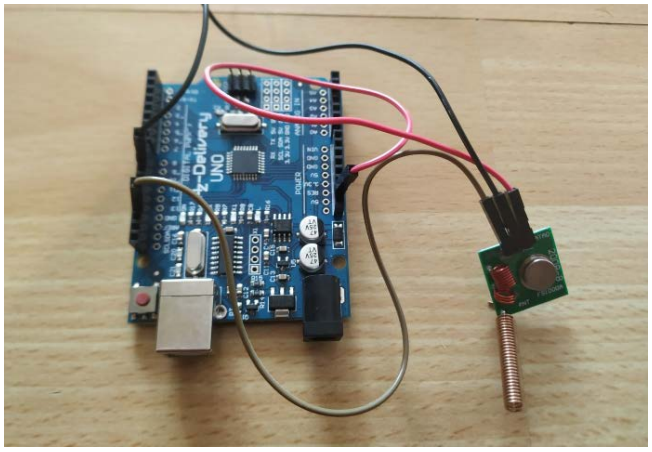
Wer keine visuell interpretierbaren Ergebnisse bekommt, sollte mit dem Verstärkungsparameter `-g` experimentieren. Ein Verstärkungsfaktor von 10 führt in diesem Beispiel zu gut interpretierbaren Datenaufzeichnungen. Um das aufgezeichnete Signal visuell zu analysieren, empfiehlt sich zum Beispiel die Verwendung des Linux-Programms Audacity.

Die aufgezeichneten Rohdaten lassen sich in Audacity als Raw-Paket mit dem 16-Bit-PCM-signed Encoding im Little-Endian-Format importieren und betrachten. Am Beispiel des Einschaltsignals kann man erkennen, dass das einmalige Betätigen der Fernbedienung zum Versand von acht identischen Datenpaketen führt (siehe Abbildung 3). Viele Hersteller implementieren solche Wiederholungen, um den Erhalt des Befehls sicherzustellen, da im viel genutzten 433-MHz-Band oft Pakete verloren gehen.



Der Universal Radio Hacker lauscht auf dem Gerät RTL-SDR und zeichnet die eingehenden Signale auf. Im Einschaltsignal der Fernbedienung findet er acht Datenpakete (Abb. 5).

In einem Rutsch analysiert der Universal Radio Hacker die in den Signalen befindlichen Binärdaten. Die zugeordneten Bits hebt er beim Markieren ebenfalls farbig hervor (Abb. 6).



Der an den Arduino Uno angeschlossene Transmitter genügt, um Geräte mit abgefangenen und aufgezeichneten Signalen fernzusteuern (Abb. 7).

Listing 4: Steckdose im 300-Millisekunden-Takt ein- und ausschalten

```
#include <RCSwitch.h>
RCSwitch rcSwitch = RCSwitch();
const int senderPin = 9;
const int wait = 300;
void setup() {
  rcSwitch.enableTransmit(senderPin);
  rcSwitch.setProtocol(1,280); //280 entspricht der Impulslänge
  rcSwitch.setRepeatTransmit(8); //Daten werden 8x gesendet
}
void loop() {
  rcSwitch.send("011001000010101100000100010110000");
  delay(wait);
  rcSwitch.send("011001000010101100001000010100000");
  delay(wait);
}
```

Mit etwas Erfahrung lassen sich Binärdaten im einzelnen Datenpaket manuell identifizieren. Lage und Länge des Signals bestimmten hier die Zustände 0 und 1 (siehe Abbildung 4).

Auch wenn das manuelle Aufzeichnen und Auslesen der Binärdaten einen gewissen Lerneffekt mit sich bringt, ist es gerade bei größeren Datenpaketen sehr mühselig. Aus diesem Grund empfiehlt es sich, das der Software zu überlassen, etwa den Decodierer `rtl_433`, der über einen integrierten Pulse-Analyzer verfügt. Ihn startet man mit dem Befehl `rtl_433 -A`. Betätigt man danach die Ein-/Ausschalttaste der Fernbedienung, zeichnet der Decodierer das Signal automatisch in seiner Binärform auf und gibt es aus (siehe Listing 3).

Mehrere Wege führen zum Bitstrom

Vergleicht man die Binärausgabe des Einschaltsignals in Listing 3 mit der Übersetzung in Abbildung 4, sieht man die Übereinstimmung. Das Signal zum Ausschalten der Steckdose unterscheidet sich lediglich in drei Bits.

Grundsätzlich lässt sich die Analyse des Spektrums ebenso wie das Decodieren auch mit anderen Werkzeugen bewerkstelligen. Wer die gezeigten Schritte mit einem einzigen Tool durchführen will, kann zum Universal Radio Hacker greifen. Das kostenlos für Windows, Linux und macOS verfügbare Programm bietet eine Vielzahl von Analysemöglichkeiten (siehe ix.de/zuef). Nach seiner Installation und dem Einbinden des RTL-SDR kann es das Signal zum Einschalten der Steckdose aufzeichnen und danach analysieren. Auch der Universal Radio Hacker identifiziert beim Betätigen der Fernbedienung acht Pakete (siehe Abbildung 5).

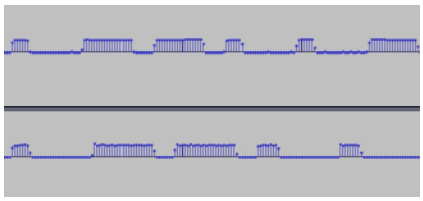
Nach dem Einstellen der Symbollänge decodiert der Universal Radio Hacker sämtliche Binärdaten der Pakete automatisch. Sie gleichen exakt den inversen Binärdaten, die die Analysen in Audacity und rtl_433 ergeben haben (siehe Abbildung 6).

Ist ein Angreifer in der Lage, solche Signale mitzuschneiden, kann er damit und unter Verwendung geeigneter Transmitter Geräte ansteuern. SDRs, die neben der Receiver-Funktion auch als Transmitter fungieren können, etwa HackRF, BladeRF oder USRP B200, wären mögliche Kandidaten. Für das Senden auf der 433-MHz-Frequenz gibt es aber eine preiswertere Methode.

Auch als Eigenbau

Ein Arduino-Board samt 433-MHz-Transmitter kann die mitgeschnittenen Pakete ebenfalls wiedergeben und damit die Steckdosen steuern (siehe Abbildung 7). Solche Transmitter lassen sich bereits für ein paar Euro über Onlineshops beziehen.

Da die erhältlichen 433-MHz-Transmitter meist über keine Antenne verfügen, ist eine solche von Hand anzulöten, um ein optimales Signal zu emittieren. Für das Programmieren des Arduino ist zudem noch die Open-Source-Bibliothek rc-switch in die verwendete IDE, etwa die Arduino IDE, einzubinden (siehe ix.de/zuef). Der Code in Listing 4 schaltet die Steckdose im 300-Millisekunden-Takt ein und aus.



Grafische Werkzeuge wie Audacity und der Universal Radio Hacker eignen sich auch für einen Vergleich der Impulslängen, um damit das eigene Signal an das Original anzugleichen (Abb. 8).

Unter Umständen sind die Impulslängen im Code so anzupassen, dass sie bestmöglich den aufgezeichneten Impulsen entsprechen. Unterschiedliche Impulslängen lassen sich bequem in Audacity oder im Universal Radio Hacker analysieren. Abbildung 8 zeigt ein Beispiel für die unterschiedlichen Impulslängen: Oben ist das Einschaltsignal der Fernbedienung zu sehen, unten das vom Arduino erzeugte Signal. Hat man den Code entsprechend angepasst und auf den Arduino geladen, schaltet der die Steckdose im 300-ms-Zyklus ein und aus.

Fazit

Allein dieses Beispiel verdeutlicht die Dringlichkeit von Sicherheitsstandards bei Funktechniken. Denn Replay-Angriffe lassen sich in der Regel durch die gängigen Verschlüsselungsmethoden vermeiden. Die Erfahrung zeigt jedoch, dass die meisten im Umlauf befindlichen Produkte, die solche Funkstandards verwenden, unzureichend oder gar nicht gesichert sind. Dadurch ließen sich bereits in unterschiedlichen Hardware-Audits Bestell-Pager etwa für Fast-Food-Restaurants, Einbauschalter zur heimischen Lichtsteuerung, Türklingeln, aber auch Alarmanlagen steuern.

Ab der nächsten Ausgabe unterbricht ein Tutorial, das sich mit dem Absichern von IoT-Geräten beschäftigt, die IoT-Hacking-Serie, bevor es mit dem Hacking von Gerätezugängen über Bluetooth Low Energy weitergeht. (sun@ix.de)

Quellen

Alle Downloads und Ressourcen: ix.de/zuef

Alexander Poth

ist IT-Security-Analyst bei NSIDE ATTACK LOGIC. Zu seinen Schwerpunkten zählt die Sicherheit von IoT-Geräten.