

Ziggy's of the world

geeks, computers, and bread

Sniffing GSM traffic with HackRF.

GSM, Hacking, HackRF

While my friend and colleague Simone (<https://twitter.com/evilsocket>) was visiting our ZIMPERIUM – Enterprise Mobile Security (<https://www.zimperium.com/products>) TLV office, we got our hands on HackRF (<https://greatscottgadgets.com/hackrf/>) and hacked together the unguarded borders of Radio Frequencies. Simone had the great patience to try and explain me the boring world of complex numbers and friends (more on that here (<http://greatscottgadgets.com/sdr/>)), but my dyslexia won over again and left me to fill the gap by following Simone's steps (and some mistakes, eh Simone?) and use my 'trial & error' approach until success. This tutorial is the result of our collaborate GSM hacking session, presented with the hope it will be useful for others.

Tools used:

- hackrf_kalibrate (<https://github.com/scateu/kalibrate-hackrf>)
- gnuradio-companion (<https://gnuradio.org/redmine/projects/gnuradio/wiki/GNURadioCompanion>)
- gr-gsm (<https://github.com/ptrkrysik/gr-gsm>)
- gqrx (<https://github.com/csete/gqrx>)
- wireshark (<https://www.wireshark.org/>)

Install Requirements:

First thing, you want to make sure you have all the required software installed, you can install most of them and their dependencies using your distribution package manager. Let's start with the libraries and tools for the hackrf itself, on a Debian/Ubuntu distro

you'll install them like so:

```
1 | sudo apt-get install hackrf libhackrf-dev libhackrf0
```

Once these libraries are installed, you can plug your hackrf into one of your USB ports and execute the **hackrf_info** command, at this point you should see something like the following:

```
1 | # hackrf_info
2
3 | Found HackRF board.
4 | Board ID Number: 2 (HackRF One)
5 | Firmware Version: 2014.08.1
6 | Part ID Number: 0x00574746 0x00574746
7 | Serial Number: 0x00000000 0x00000000 0x14d463dc 0x2f4339e1
```

You will now install **gnuradio** which is the software we'll use to decode the RF signals, **gqrx** a tool to visualize signal power on certain frequencies and everything else that will be needed in the next steps:

```
1 | sudo apt-get install gnuradio gnuradio-dev gr-osmosdr gr-osmosdr g
```

Proceed with **gr-gsm**, the GnuRadio blocks that will decode GSM packets:

```
1 | sudo apt-get install git cmake libboost-all-dev libcppunit-dev swi
2 | git clone https://github.com/ptrkrysik/gr-gsm.git
3 | cd gr-gsm
4 | mkdir build
5 | cd build
6 | cmake ..
7 | make
8 | sudo make install
9 | sudo ldconfig
```

Now create the file `~/.gnuradio/config.conf` and paste the following contents into it:

```
1 | [grc]
2 | local_blocks_path=/usr/local/share/gnuradio/grc/blocks
```

Finally install **kalibrate-hackrf**, a tool that will hop among known GSM frequencies and will tell you which your country is using:

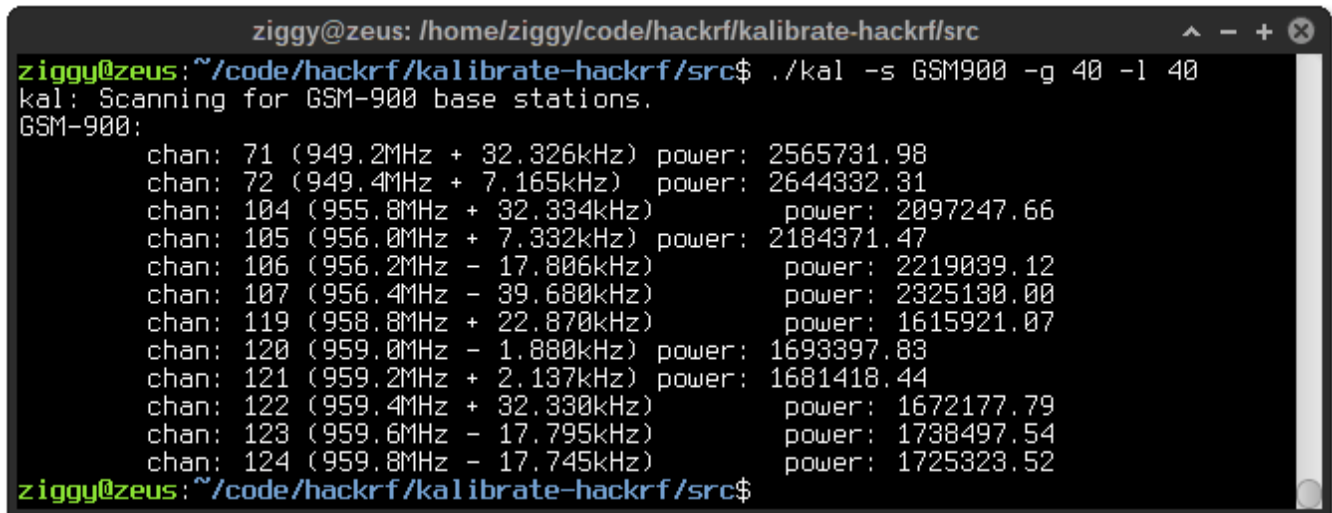
```
1 | git clone https://github.com/scateu/kalibrate-hackrf.git
2 | cd kalibrate-hackrf
3 | ./bootstrap
4 | ./configure
5 | make
6 | sudo make install
```

Finding GSM Frequencies:

Each operator in each country uses a different frequency in the GSM possible spectrum, which usually starts from 900Mhz. You can use `hackrf_kalibrate` to find the frequencies you want to sniff:

```
1 | ./kal -s GSM900 -g 40 -l 40
```

Note the two gain values, those are important in order to get some results. Leave `kalibrate` running and after a while you should see an output similar to this:

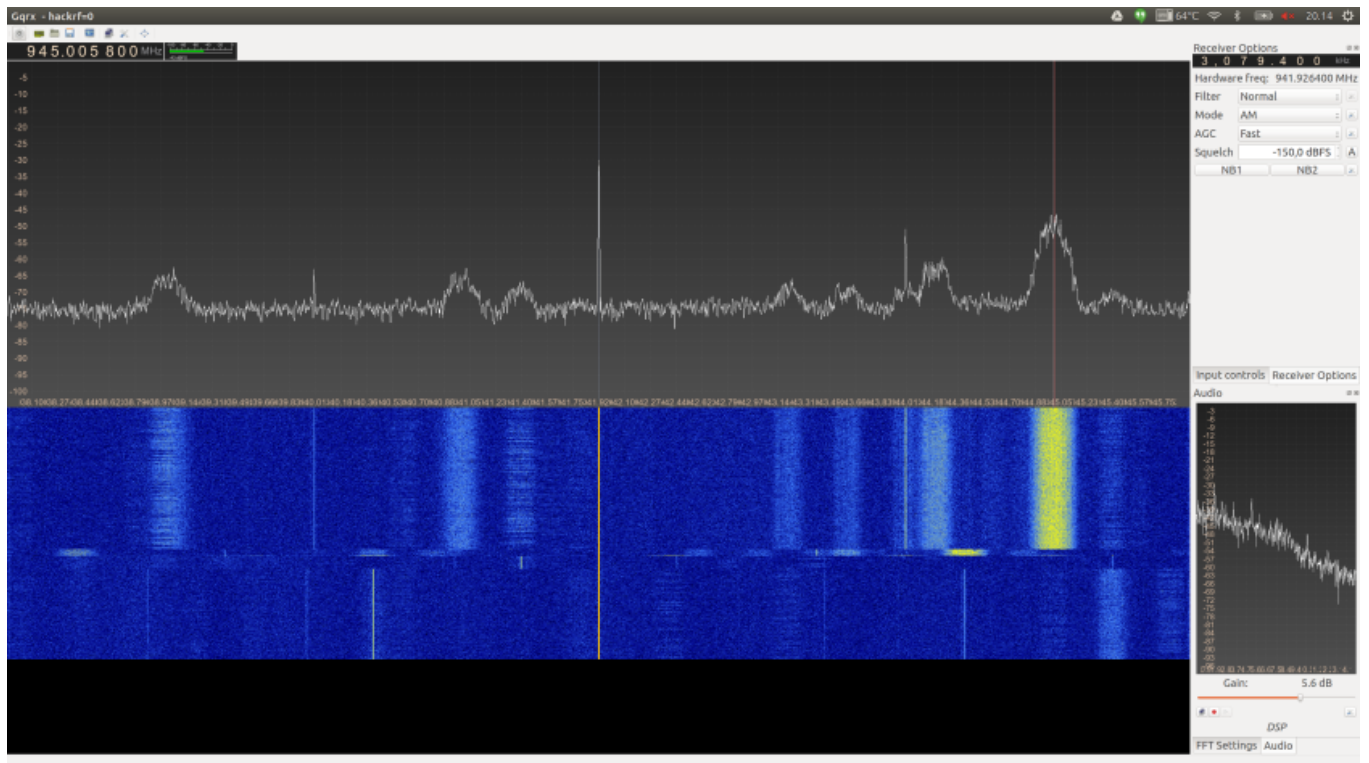


```
ziggy@zeus: /home/ziggy/code/hackrf/kalibrate-hackrf/src
ziggy@zeus:~/code/hackrf/kalibrate-hackrf/src$ ./kal -s GSM900 -g 40 -l 40
kal: Scanning for GSM-900 base stations.
GSM-900:
  chan: 71 (949.2MHz + 32.326kHz) power: 2565731.98
  chan: 72 (949.4MHz + 7.165kHz) power: 2644332.31
  chan: 104 (955.8MHz + 32.334kHz) power: 2097247.66
  chan: 105 (956.0MHz + 7.332kHz) power: 2184371.47
  chan: 106 (956.2MHz - 17.806kHz) power: 2219039.12
  chan: 107 (956.4MHz - 39.680kHz) power: 2325130.00
  chan: 119 (958.8MHz + 22.870kHz) power: 1615921.07
  chan: 120 (959.0MHz - 1.880kHz) power: 1693397.83
  chan: 121 (959.2MHz + 2.137kHz) power: 1681418.44
  chan: 122 (959.4MHz + 32.330kHz) power: 1672177.79
  chan: 123 (959.6MHz - 17.795kHz) power: 1738497.54
  chan: 124 (959.8MHz - 17.745kHz) power: 1725323.52
ziggy@zeus:~/code/hackrf/kalibrate-hackrf/src$
```

(<https://z4ziggy.files.wordpress.com/2015/05/kalibrate.png>)

You will have to use the proper GSM parameter ('-s') to correspond to your local operator. Consult [this \(http://www.worldtimezone.com/gsm.html\)](http://www.worldtimezone.com/gsm.html) list for verification.

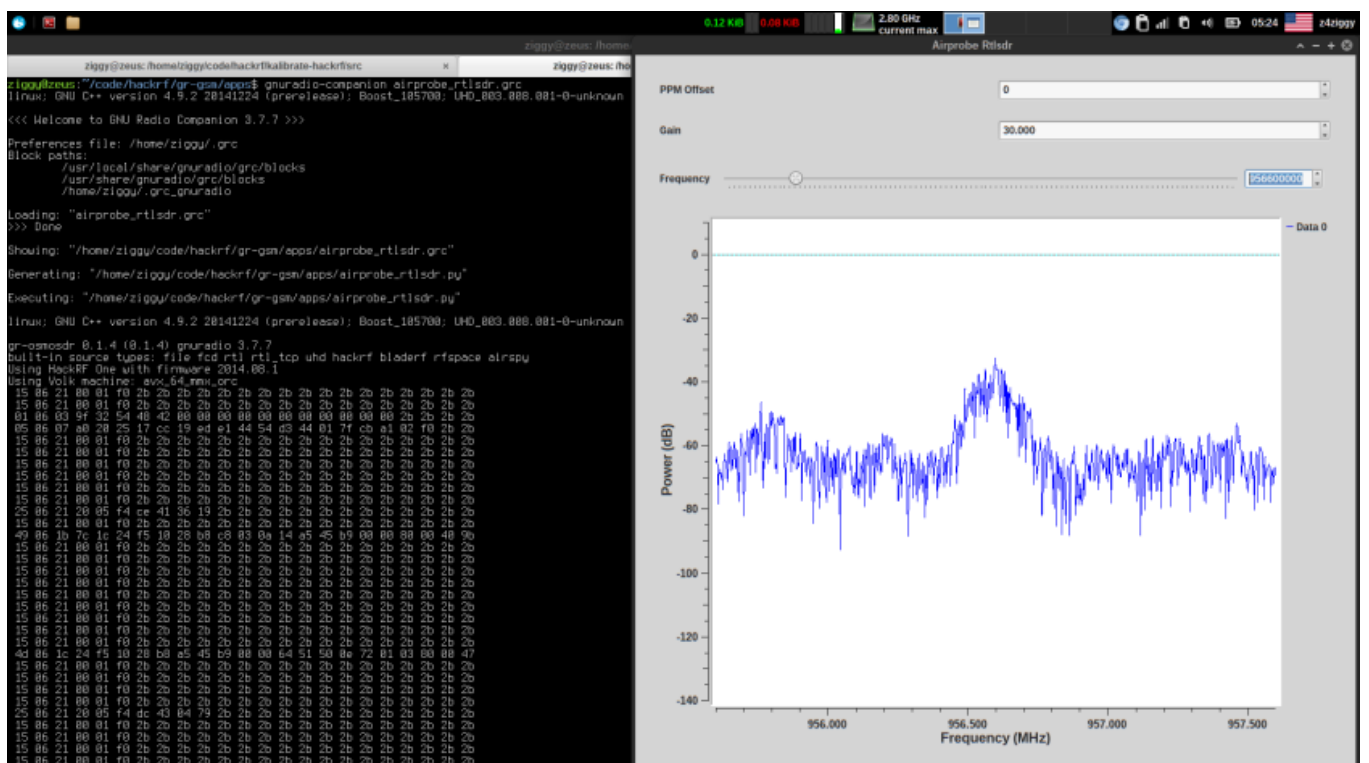
Sometimes you might want to **see** the frequencies in order to ensure correct results from `hackrf_kalibrate`, or to save yourself from calculating the correct frequency given by `hackrf_kalibrate` (notice the +/- Khz sign of each result – this means the top peak with the corresponding power, not 100% correct frequency). Open `gqrx` and tune it to the first frequency you got from `hackrf_kalibrate`, for example 940.6Mhz, and you'll see something like the following picture:



(<https://z4ziggy.files.wordpress.com/2015/05/waterfall.png>)

In the above screenshot you can visually see the activity is around 945Mhz.

Once you know the GSM channels frequencies, you can start gr-gsm by running the python script `./airprobe_rtlsdr.py` or load the `airprobe_rtlsdr.grc` file using gnuradio-companion and set one of the channel frequencies you just found in the frequency field. Don't forget to add 'gain' value again, move back to the frequency field and start pressing the UP/DOWN arrows on your keyboard to start scrolling the frequencies in 200Khz steps until you start seeing some data in your console window. The whole process should look something like this:

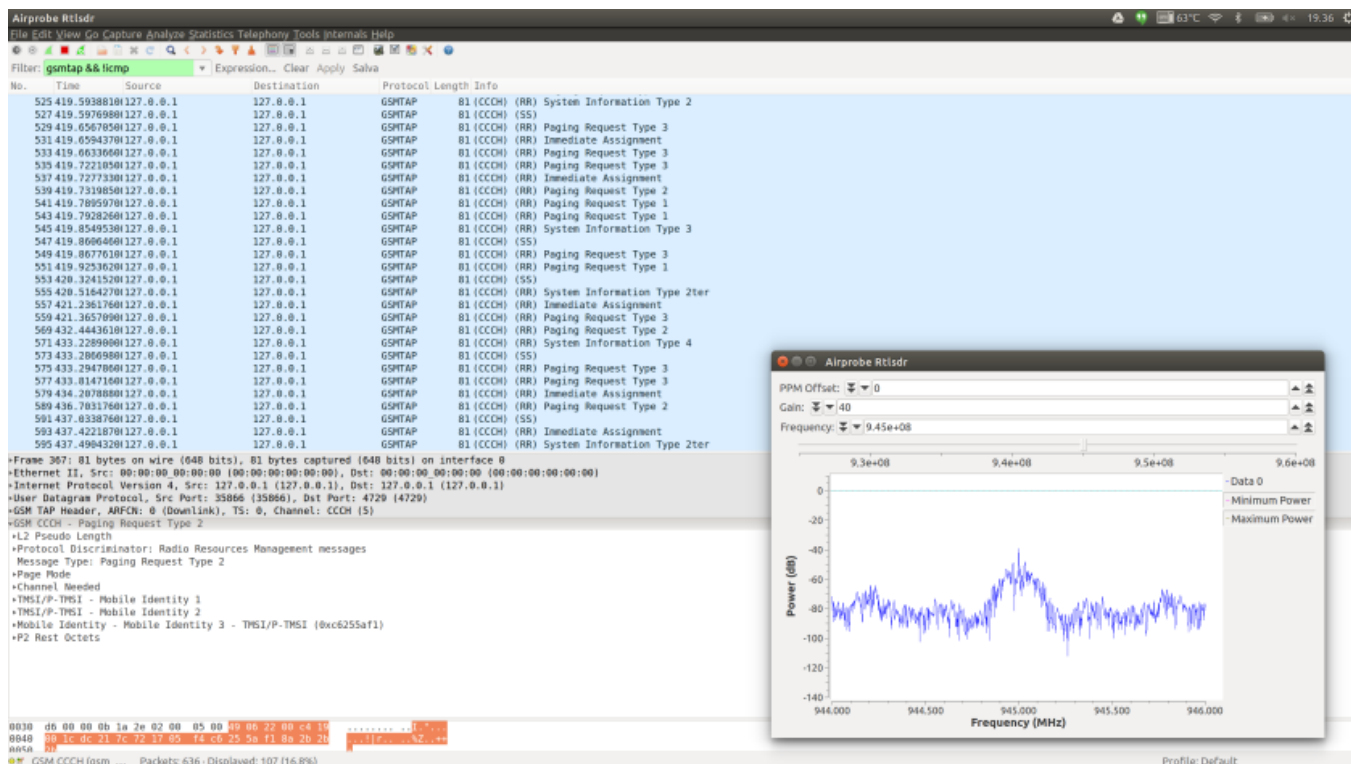


(<https://z4ziggy.files.wordpress.com/2015/05/gr-gsm.png>)

Now you only need to launch **wireshark** from another terminal tab with the following command:

```
1 | sudo wireshark -k -Y 'gsmtap && !icmp' -i lo
```

If gr-gsm did his job, you should be able to see decoded GSM traffic sniffed by your hackrf.



(<https://z4ziggy.files.wordpress.com/2015/05/wireshark.png>)

About these ads (<https://wordpress.com/about-these-ads/>)
You May Like

1.  [Gravity is personalizing the internet. Discover how they do it.](https://gravity.com) a year ago gravity.com Gravity.com (sponsored)

May 17, 2015 May 20, 2015 · z4ziggy · Tagged GSM, Hacking, HackRF · 1 Comment

One thought on “Sniffing GSM traffic with HackRF.”

1. SKOPERST

May 19, 2015 at 11:44 am · Reply

But what exactly is innovative in getting the GSM broadcast channel? Too many done it before. All testing equipment and modem manufacturers does it better and 20 years back.

Unless ofcourse your final goal is to exploit those poor Chinese modems.

Blog at WordPress.com. | The Illustratr Theme.

 Follow

Follow “Ziggy's of the world”

Build a website with WordPress.com