Demystifying SDR Hacking: A Deep Dive into Wireless Protocols Part:3

KISHORERAM

KISHORERAM

Follow

7 min read

·

Oct 7, 2023

Listen

Share

More

Aircraft Communication

Aircraft communication is a critical aspect of aviation that ensures the safe and efficient operation of aircraft. The International Telecommunication Union (ITU) has assigned aircraft analog voice dialogue in the High Frequency (HF) band between 3–30MHz and in the Very High Frequency (VHF) band at 118–137 Mhz.VHF signals are only line-of-sight but offer much higher audio quality. This makes them ideal for aircraft communication where clear and immediate transmission is vital. In this context, VHF is often preferred despite its shorter range.

Press enter or click to view image in full size

Photo by Bao Menglong on Unsplash

Common Aircraft Communication Frequencies

APP (Approach Control Frequency: 120.05 MHz): This frequency is used by the approach control department to guide aircraft as they approach for a landing.

ATIS (Automatic Terminal Information Service: 128.05 MHz): This is a continuous broadcast of recorded non-control information in busier terminal areas. It’s essential for arriving and departing aircraft as it provides information like weather conditions, active runways, available approaches, and other necessary data.

TWR (Tower Frequency: 118.15 MHz): This frequency is used by the control tower at an airport to provide instructions to aircraft in the vicinity of the airport.

These frequencies is for Coimbatore International Airport .To find the aircraft control tower frequency at your local airport, go to the airport’s website and look for the air traffic control or pilot information section.

HDSDR

Press enter or click to view image in full size

HDSDR (High Definition Software Defined Radio) is a freeware Software Defined Radio (SDR) program for Microsoft Windows. It can be used to listen to VHF aircraft communication.

To listen to aircraft communication using HDSDR, follow these steps:

Open HDSDR and set the mode to AM.

Set the frequency manager to Air.

Enter the frequency of the aircraft communication that you want to hear.

Click the Start button to start listening.

You should now be able to hear the aircraft communication. You may need to adjust the volume and other settings to get the best sound quality.

Replay Attacks in Wireless Devices with RTL-SDR

Wireless devices such as bells, switches, and car remotes have become an integral part of our daily lives due to their convenience. However, these devices can be susceptible to a type of security vulnerability known as a replay attack.A replay attack involves capturing a valid transmission and retransmitting it at a later time. This could potentially allow unauthorized access or control over the wireless device.For replaying first we need to identify the frequency it uses. This can be achieved using an RTL-SDR dongle, a cost-effective device capable of receiving radio signals. Once the frequency is identified, the signal can be recorded using the RTL-SDR dongle. After the signal has been recorded, it can be replayed using a Raspberry Pi as a transmitter.However, it’s important to note that many modern wireless devices employ security measures to prevent such attacks. One such measure is rolling code security, which is commonly used in car remotes. This security feature changes the code transmitted by the remote every time it’s used, preventing unauthorized individuals from replaying the signal to gain access.

Press enter or click to view image in full size

Source:https://www.pngegg.com/

Jamming Signals Using a HackRF One

What is Jamming?

Jamming is the act of intentionally interfering with a radio signal. It can be used to prevent a signal from being received, to corrupt the signal, or to make the signal difficult to understand. Jamming can be used for legitimate purposes, such as to prevent interference between different radio systems, but it can also be used for malicious purposes, such as to disrupt communications or to prevent navigation.

Press enter or click to view image in full size

Source:https://www.everythingrf.com/

To jam signals using a HackRF One, you will need the following:

A HackRF One

GNU Radio

The gr-osmosdr package

The osmocom\_siggen\_nogui utility

Commands

Sudo apt install gnuradio

Sudo apt install gr-osmosdr

Osmocom\_siggen\_nogui -h

The following command will generate a continuous wave signal at 100 MHz with a power of 10 dBm:

Osmocom\_siggen\_nogui -a hackrf -f 100e6 –sweep -x 2e6 -y 10 -v

Osmocom\_siggen\_nogui: This is the command to run the signal generator application. Osmocom-siggen is a versatile signal generator tool that can be used to generate a variety of signals, including constant waves, sinusoidal signals, uniform noise, Gaussian noise, frequency sweeps, GSM bursts, and two-tone signals.

-a hackrf: This option specifies the hardware to use, in this case, the HackRF One. The HackRF One is a software-defined radio (SDR) device that can be used to transmit and receive radio signals. It is a popular choice for jamming because it is relatively inexpensive and easy to use.

-f 100e6: This option sets the frequency to 100 MHz. The frequency of the signal is the number of times per second that the signal oscillates. The frequency of the signal you want to jam will depend on the type of signal you are targeting. For example, if you are targeting a GSM signal, you would need to set the frequency to 900 MHz.

- sweep: This option indicates that a frequency sweep will be performed. This means that the signal will sweep through a range of frequencies. Frequency sweeps can be used to interfere with a wider range of signals, but they can be less effective at jamming specific signals.

-x 2e6: This option specifies the start frequency of the sweep, which is 2 MHz.

-y 10: This option sets the stop frequency of the sweep, which is 10 MHz.

-v: This option enables verbose mode, providing additional information about the signal generation process.

To jam a GSM signal at 900 MHz, you would use the following command:

Osmocom\_siggen\_nogui -a hackrf -f 900e6 –sweep -x 890e6 -y 910e6 -v

This command would sweep the signal from 890 MHz to 910 MHz, which would interfere with the GSM signal at 900 MHz.

Limitations

The primary limitation of jamming signals using a HackRF One is the strength of the signal. The HackRF One is a relatively low-power device, so it can only jam signals that are close to the receiver. Additionally, the HackRF One is susceptible to interference from other radio signals, so it may not be effective in noisy environments. It is important to note that jamming signals is illegal.

GPS Spoofing with HackRF One: A Deep Dive into Location Deception

Press enter or click to view image in full size

Photo by Tobias Rademacher on Unsplash

GPS spoofing is a fascinating and somewhat controversial topic in the world of software-defined radio (SDR) and radio communications.GPS spoofing is the act of manipulating GPS signals to deceive a receiver into believing that it is located at a different location than it actually is. This can be done for a variety of purposes, such as to evade tracking or to disrupt critical infrastructure. One way to spoof GPS signals is to use a software-defined radio (SDR) device such as the HackRF One. The HackRF One is a low-cost SDR device that can be used to transmit and receive radio signals across a wide frequency range. To spoof GPS signals with the HackRF One, you will need to install the gps-sdr-sim software. This software can be used to generate simulated GPS signals that can then be transmitted by the HackRF One. To start spoofing GPS signals, you will need to compile the gpssim.c file. Once the gpssim.c file has been compiled, you can use it to generate a simulated GPS signal file. This file can then be transmitted by the HackRF One using the hackrf\_transfer command.

Step #1: Install HackRF One

Step #2: Install GPS Spoofing Software

Create a Directory:

Kali > mkdir GPS\_SPOOF

Navigate to the New Directory:

Kali > cd GPS\_SPOOF

Download GPS Spoofing Software:

Kali > sudo git clone <https://github.com/osqzss/gps-sdr-sim.git>

Navigate to the Software Directory:

Kali > cd gps-sdr-sim

Compile the Software:

Compile the gpssim.c file to create an executable named gps-sdr-sim:

Kali > sudo gcc gpssim.c -lm -O3 -o gps-sdr-sim -DUSER\_MOTION\_SIZE=4000

Step #3: Locate the Satellite

Proceed with GPS spoofing, you need information about GPS satellites positions. This information is obtained from GPS broadcast ephemeris files, which can be downloaded from sources like NASAs CDDIS archive. Make sure to download the most recent daily file.

NASA CDDIS Ephemeris Files:

<https://cddis.nasa.gov/archive/gnss/data/daily/>

<https://cddis.nasa.gov/archive/gnss/data/daily/2022/brdc/>

Once you have the ephemeris file, you can use it to generate a simulated pseudorange and Doppler for the satellites in your range.

Select a Location. Choose the location you want to spoof. You can use services like Google Maps to obtain GPS coordinates.

Start GPS Spoofing

To initiate GPS spoofing, use the following command, providing the ephemeris file and GPS coordinates:

Kali > sudo ./gps-sdr-sim -b 8 -e <ephemeris\_file> -l <latitude>, <longitude>, <altitude>

This command creates a simulation file named gpssim.bin containing spoofed GPS data.

Transmit the Spoofed GPS Signal

Now, you can transmit the spoofed GPS signal using the HackRF One:

Kali > sudo hackrf\_transfer -t gpssim.bin -f 1575420000 -s 2600000 -a 1 -x 0

This command sends the spoofed GPS signal, making any GPS receiver tracking it believe that it’s at the specified location.

GPS spoofing can be used to conceal our location and prevent tracking by governments and malicious actors.

Thanks For Reading ☺

Don’t miss out on my upcoming articles! Follow me on Medium for more insightful content. Clap and share this article to spread the knowledge among fellow bug bounty hunters and cybersecurity enthusiasts.

If you have any further questions or would like to connect, feel free to reach out to me

My LinkedIn handle: <https://www.linkedin.com/in/kishoreram-k/>