Exploiting GSM Vulnerabilities

Ugo Reyne B00122680
Alexandre Carra B00122679
Putera Rameli B00095349

Department of Informatics
School of Informatics and Engineering
Technological University Dublin

Submitted to Technological University Dublin in partial fulfilment of the requirements for the degree of
Bachelor of Science (Honors) in Digital Forensics & CyberSecurity

Supervisor: Arnold Hensman

Submission Date:
20th May 2019

Declaration

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the award of Degree of **B.Sc. in Computer Science** in the Institute of Technology Blanchardstown, is entirely my own work except where otherwise stated, and has not been submitted for assessment for an academic purpose at this or any other academic institution other than in partial fulfilment of the requirements of that stated above.

Signed:_____ Dated: ____/_____/_____

Signed:_____ Dated: ____/_____/_____

Signed:_____ Dated: ____/_____/_____

Contents

Abstract

The purpose of the project is to identify how a user can be vulnerable to an attack through the GSM network. Throughout this report the reader will gain a basic understanding on how the GSM network works and how communication works through the GSM protocols. This project demonstrates the vulnerabilities found in GSM and how attacks can be performed by an attacker using various different methods of attacks.

The goal of the project is to identify the vulnerabilities and then show how the attacks work if one would be subject to one. Before any demonstrations of the vulnerabilities, knowledge of the GSM architecture needed to be gained. It was necessary to understand the protocols and the security features it implements in the world of mobile communications today.

For the purpose of this project, it was necessary to obtain some hardware and software tools to implement the vulnerabilities shown in the document. The hardware used for the project is a software-defined radio called “BladeRF”. This is a software defined radio which is a full-duplex transceiver that is used to receive and transmit frequencies. Along with the hardware used, the main software used in this project is Yate Base Station (BTS). This is used alongside the BladeRF to create a base station that mobile devices can connect to and start communicating with each other. By using the BladeRF and the software together, we can to show the vulnerabilities that are still present in the GSM protocol.

The goal was to exploit the named vulnerabilities detailed in the report and this was achieved through using the hardware obtained alongside the software found online.

Keywords: IMSI-Catchers, GSM, fake BTS, Rogue BTS, BTS, MS, BSC, Sniffing SMS, SMS capture, A5/1, RTL-SDR, BladeRF, HackRF, LimeSDR.

Chapter 1: Introduction

In recent times, there has been an increase of attacks through user's devices. Today, there are a large amount of vulnerabilities found in applications as well as devices. In this project the focus is on how devices are vulnerable to certain attacks if the attacker is using a software defined radio (SDR). Throughout this report we will cover the topics of GSM, SDR and Mobile telecommunications.

In order to exploit vulnerabilities found, we first needed to understand how communication works through by use of GSM. Our goal of the project is to test the vulnerabilities found in GSM and then exploit it by showing that communication can be eavesdropped, essentially a man-in-the-middle attack.

The report covers the information regarding the laws that are placed in Ireland when using software defined radio. The report also covers in detail the steps of configuring the software defined radio to use accordingly for mobile communications. The Yate Base Station (BTS) configuration is explained and the role of the base station is explained in the GSM architecture. Finally, the testing and results shown in the report shows the attacks in full effect and the mitigation of these attacks are explained to help people who have no knowledge of these attacks and how to defend against them.

Background

The GSM was the idea of Henry Kieffer who suggested to find a new spectrum for mobile 900MHz in 1975, The GSM standard started in 1987 and it refers to 2G cellular network which included data communication also. With EDGE (Enhanced Data rates for GSM Evolution) and before with GPRS (General Packet Radio Services).

The concept of GSM is based on Time Division Multiple Access (TDMA) transmission. The GSM protocol operates between the 900MHz and 1.8 GHz in Europe and 850 MHz and 1.9 GHz in US Bandwidth.

Legislation

Under the Wireless Telegraphy Act 1926 -1928:

“The possession and use of radio equipment in Ireland is governed by the Wireless Telegraphy Act 1926, (Act No 45 of 1926), (as amended), which stipulates that an appropriate Wireless Telegraphy licence must be held, unless licence exempted.” [2]

In this project we were extra careful when choosing which frequencies to use for mobile phones. The mobile phones in Ireland operates between 800 – 900MHz. After configuring the Yate Base Station, it was set up in a way that only the mobile devices that was used for the tests could only connect to the station. Any other devices that would try connecting to the station would be denied from the network.

863 - 870 MHz (continued)	25 mW ERP Power density : -4.5 dBm/100 kHz (Note 1.6)	Duty cycle: ≤ 0.1 % , or LBT + AFA (Note 1.1,1.4 and 1.5)	EN 300 220	DSSS and other wideband modulations other than FHSS European Legislation: Decision 2006/771/EC Decision 2011/829/EU Decision 2013/752/EU ERC/REC 70-03

Frequency band 863 – 870 [3]

915.2 – 920.8 MHz[22]	25 mW ERP except for the 4 channels identified in note 1.9 where100 mW ERP. applies	Duty cycle: ≤ 1% (note 10) For ER-GSM protection (918-920.8MHz, where applicable), the duty cycle is limited to ≤ 0.01% and limited to a maximum transmit on-time of 5ms/1s Maximum occupied bandwidth: 600 kHz except for the 1 channel identified in note 1.9 where ≤ 400 kHz applies	EN 300 220	ERC/REC 70-03

Frequency band 915.2 – 920.8

Statement of Ethics

The laws in Ireland are very clear with regards to the use of the named technologies for this project. It is illegal to use any of these on the public or in public. The project was taken place in an isolated area where and the devices that were used belonged to the group members. The connected devices to the created base station belonged to the group members and there was a specific deny all other IMSIs that were trying to connect to the Yate Base Station.

The methods shown in this thesis includes eavesdropping and attacking the GSM protocol. It is illegal to implement these methods to the public without a warrant and our group took all the necessary steps to make sure that the software defined radio never came into contact with the public.

Literature Review

Investigating Vulnerabilities in GSM Security Pannu, M;, Bird, R., Gill, B. and Patel K. 2015

The worldwide telecommunication uses a mixed of 2G(GSM), 3G(UMTS) and 4G(LTE). With the rise of attacks, the need of cyber-defence, and secure infrastructure is needed. When a new standard is released it offers a new layer of protection to the users. A lot of people have access to the 3G. Users are expected to be aware of potential threats but auditing the existent is the role of administrators. This paper shows the weaknesses in the GSM standard and show how to audit the GSM networks for vulnerabilities.

Demonstration of Vulnerabilities in GSM security with USRP B200 and Open-source Penetration Tools Dubey, A. Vohra, D., Vachhani, K. Rao, A. 2016

This paper highlights the vulnerabilities in the GSM architecture through a fake base transceiver station, the attack was possible by taking advantage of lack of two-way authentication. That show how to set up a fake Base Transceiver station(BTS) and send Malicious SMS.

How to intercepting GSM by Ioannis, G 2015

Secure communication is a big problem, since we live in technologically advanced community. The GSM protocol is used everywhere for phone communication. A lot of people have a phone/smartphone and we are not protected from intercepting or tracking a smartphone. In this paper is described how to create a GSM Scanner with GNURadio and Airpobe with a Universal Software Radio Peripheral(USRP), this also explain how to decrypt the gsm traffic by cracking the encryption key(A5/1) through 1.6TB rainbow tables and known plaintext attack.

Solutions to the GSM Security Weaknesses Toorani,. M, Agashar Beheshti Shiraz,. Ali.

Actually, the number of people who have register to a mobile provider service is increasing a lot. The GSM Network is vulnerable to several flaws. Many operators still using the 2G network. This paper presents the most important security breach of the 2G (GSM) system, it's also provided some solution to upgrade the security of the current 2G network.

Chapter 2: Design

Methodology

Before implementing any technologies to exploit the vulnerabilities mentioned earlier, we needed to understand how GSM works and is implemented in the world of mobile communications. In general, mobile phones will connect to a base station that has the strongest signal. A typical scenario in a real-world example would be that an attacker creates a fake base station that users connect to. The users are unaware that it is a fake base station, so this enables the hacker to sniff traffic, view conversations and even eavesdrop on phone conversations.

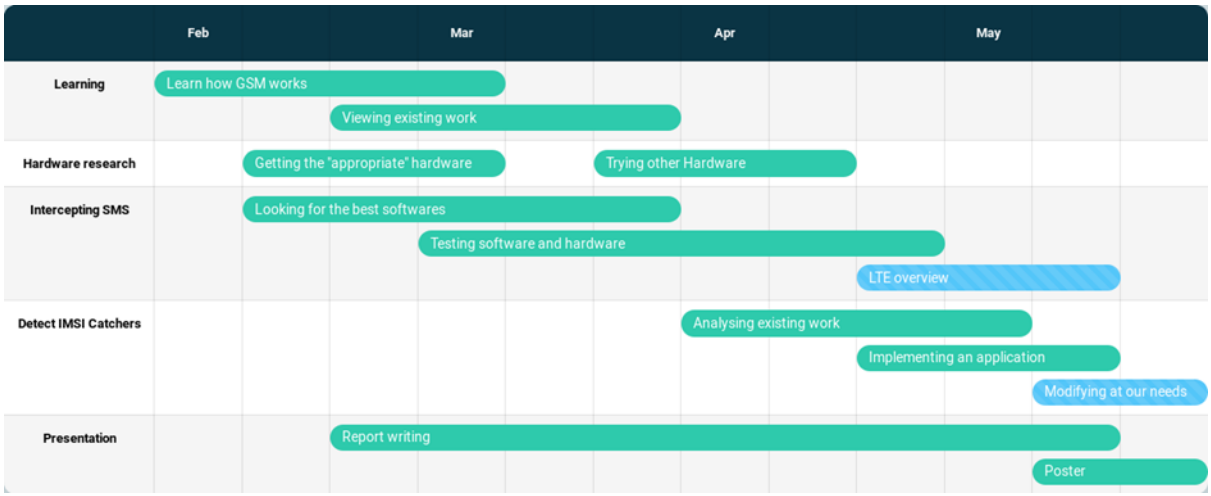


Figure 2.5 Project Stages

To perform such a project, learning about the GSM networks and looking at existing work is the first step, then the research and acquiring the hardware needs to be done before starting the testing.

GSM Architecture

GSM stands for Global System for Mobile Communications. Originally developed in 1984 for the use of telephone system across Europe, it is now the international standard for mobile devices.

The services that can be used through GSM includes voice communications like voice calls, short messaging service (SMS) and many others. The frequencies that are used in this project will be around the 900MHz band.

This section intends to explain the GSM architecture and the protocols used to ensure that communications work smoothly in present day.

A full network can be seen below with each of the components that make up the GSM architecture. A full network is made up of different components and interfaces. Each component in the network handles different tasks in the overall flow of communication.

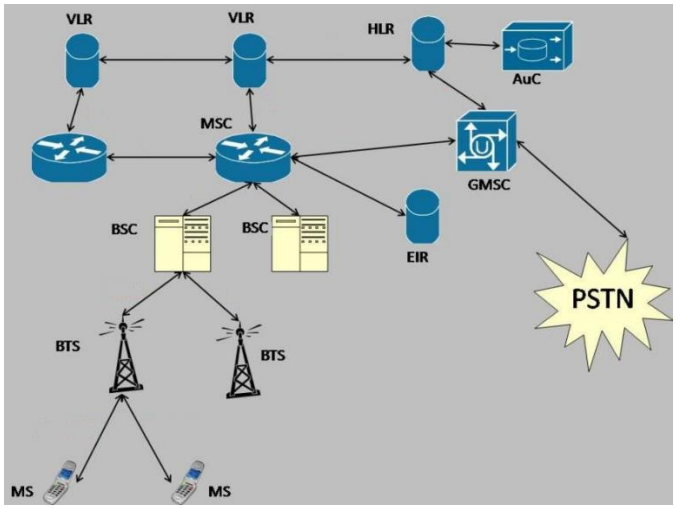


Figure 2.1 GSM Architecture

Mobile Station (MS):

This part from the GSM architecture includes two components:

1. **Mobile Equipment (ME)**

   Mobile equipment refers to the mobile device itself that is connected on the GSM network. Each mobile device can be identified through its International Mobile Equipment Identity (IMEI). This unique number is chosen during the mobile device's manufacturing process.

2. **Subscriber Identity Module (SIM)**

   The SIM is a small card which is in the phone and it stores information like, TMSI, IMSI, Ki (for encryption, Local Area Identify (LAI) and the Service Provider Name (SPN).

Base Transceiver Station (BTS):

The base transceiver station is basically a cell tower that the mobile stations can connect to. The BTS sends and receives data from the MS. It is an access point for the mobile phones to access the GSM network. The station handles specific things like encryption of communication like SMS. Speech encoding, multiplexing and modulation/demodulation of the radio signals.

Base Station Controller (BSC):

The base station controller handles multiple different base stations. Most importantly, it handles the radio channel allocations (on which frequency the base station will send and receive data), power and signal measurements from the mobile stations and the administration of frequencies.

Mobile Switching Center (MSC):

This component of the GSM network has an important role in the GSM architecture. It is found at the centre of the network. The MSC oversees routing calls, setting up calls and basic configurations of switching.

Gateway Mobile Switching Center (GMSC):

This component acts as a gateway between two networks. For example, if a user wants to place a regular call using landline, the call would go through this GMSC and then be passed or switched to the Public Switch Telephone Network (PSTN).

Home Location Register (HLR):

This component stores all the information about its connected subscribers on the network.

Visitor Location Register (VLR):

This is a database that is apart of the HLR but this component only stores information of the subscribers currently in its location area.

Equipment Identity Register (EIR):

The EIR contains 3 lists – black, grey and white. The blacklist contains all the IMEI's that are denied from the service of the network. The grey list contains the ones that are to be monitored continuously for suspicious activity and the white list contains all the IMEI's that aren't on the grey or blacklist.

Authentication Center (Auc):

Auc stores the Ki for each IMSI on the network which is used for authentication and encryption.

Short Messaging System (SMS)

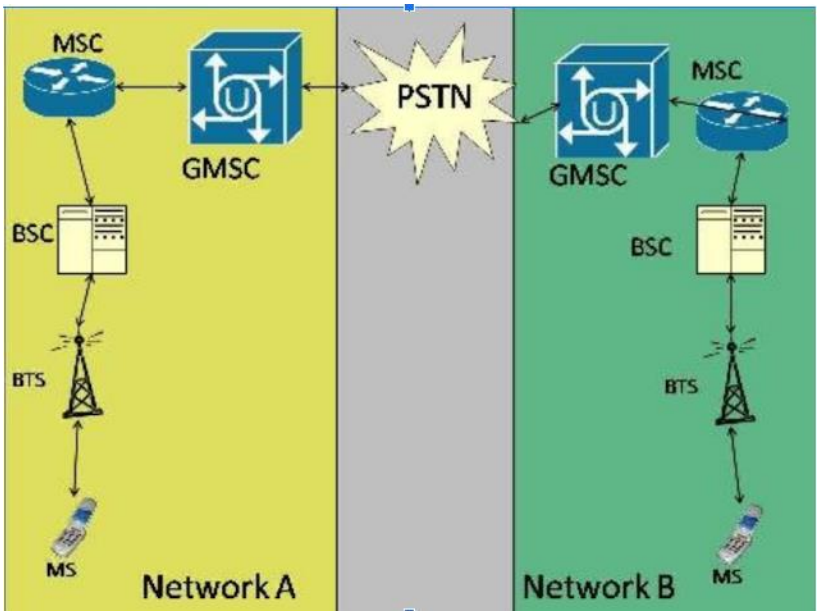


Figure 2.2 SMS in the network

Having explained each component in the GSM architecture, the figure above shows a more specific type of communication. This figure shows how one user on network A can contact the second user on Network B. It starts off with the MS on network A connects to the BTS and the message is passed to the BSC which assigns the radio channels and condenses the message which reduces the overall traffic passed onto it. The traffic moves on to the MSC and on arrival, the message is stored within the Short Messaging Service (SMSC) while the subscriber's details are validated using the two databases the VLR and the HLR. The EIR examines the subscribers to check if it is cloned or stolen. Once the message reaches the GMSC, from here the message will be passed to Network B through the PSTN. The PTSN used for public landlines acts as the connection between the two mobile phones. Once the message has passed into Network B, the message goes through each component of the GSM network but this time backwards until the message finally reaches the MS in Network B.

Mobile - Originated call

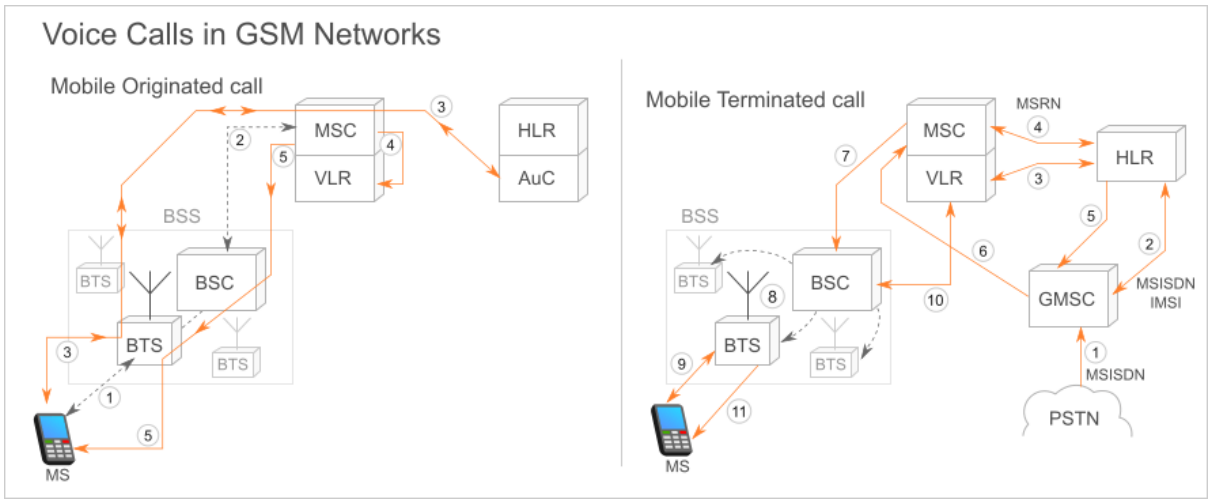


Figure 2.3 Calls in GSM

When a call originates from the MS from user 1, it is passed through all the components of the network before it gets to the second MS which is user 2. To connect to the network, the mobile must connect to a radio network, where the mobile device sends shares messages with the Base Station Subsystem of the radio network in order to send and receive signals. The BSC then takes the message and passes it to the MSC where it is responsible for voice and SMS within the network. The MSC connects to the PTSN and it links networks together to allow local and international communication to be made possible.

Vulnerabilities in GSM

The following explains the most common types of attacks used on GSM. These attacks in GSM have been known for many years now, yet mobile companies are very reluctant to upgrade the infrastructure due to costs and it has been documented that the intelligence agencies have asked the companies to keep the architecture the same in order for them to exploit the vulnerabilities for investigations.

Base Station Spoofing:

In this attack, the attack creates their own home-made base station which will act as a legitimate station for mobile phones to connect to. This fake base station can now eavesdrop on the devices connected to it and perform some man in the middle attacks.

SMS Exploitation:

When a user connects to the fake base station, SMS can be sent across the network through the base station that has been created. When a user sends a text, the messages can be eavesdropped on and thus other attacks like social engineering attacks can be carried out.

IMSI Catching:

The users MCC, MNC and MSIN are now available to the attacker as the base station can view these elements. From this point the attacker can find the exact SIM card that the user is using.

Encryption in GSM

Inside the GSM protocol the cipher used between the BTS and the MS is the A5 encryption

There is three different cipher for this encryption:

- A5/0: means "no encryption". Data is sent unencrypted. In some countries, this is the only allowed mode (India is such a country).
- A5/1: is the old "strong" algorithm, used in Europe and North America.
- A5/2: is the old "weak" algorithm, (it is not recommended in the GSM specifications)
- A5/3: is the newer algorithm for GPRS/UMTS

The A5/1 encryption is actually a stream cipher used in communication over-the-air(OTA) in GSM cellular network standard. Created in 1987 it was at the beginning kept secret but was reverse in 1999 by Marc Briceno and fell in the public knowledge domain. A5/1 is globally used in Europe and USA, United Kingdom and Australia.

It used a 64 bits key but this implementation in GSM used only 54 bits (10 bits is down to 0).

It works with three Linear-feedback shift register (LFSR). A conversation over GSM is made by a division in temporary block(burst), each burst is sent every 4.615 milliseconds and contain 114 bits. A session key K is used for the communication.

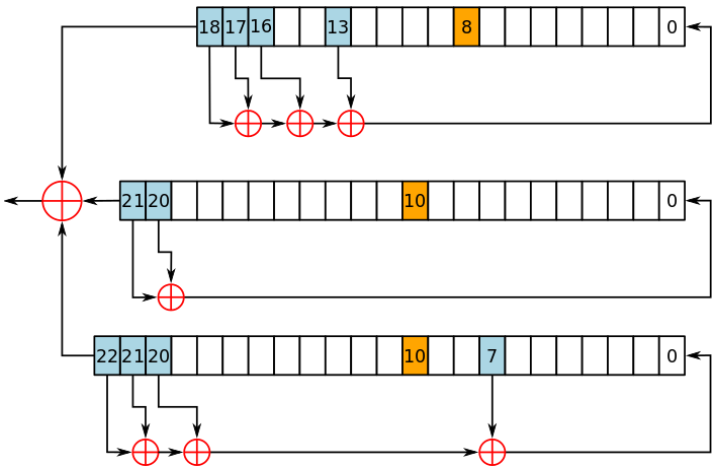


Figure 2.4 Schematic A5/1 Encryption

This is a schematic of the A5/1 with these three shift registers. A register is shifted, if the orange bit (also called clocked bit) match with the majority of the 3 bits in orange. Let is insert a bit according to a XOR between the two other bits in blue (also called tapped bits). Each tapped bit is XORed with the next one that produce a keystream that is also XORed with other keystream.

Chapter 3: Implementation

The BladeRF was picked as the final hardware to implement the project. The HackRF and the LimeSDR mini, were both tested but at the end it was concluded that the hardware was not capable of implementing the project. The HackRF is only a half-duplex which means it could not act as a fake base station and it was limited in its capabilities. The LimeSDR mini was not able to run after many attempts at trying to compile software to run through the hardware. There was a shortage of references online on how to attempt to install and configure software. It is still relatively new, but in a few years, there will be more documentation to help with the set-up procedure. This is why the BladeRF was used for the Active Testing.

Software Defined Radio

The frequency ranges and the prices in the table below can change between the models of each type of hardware.

	RTL-SDR	HackRF	LimeSDR Mini	BladeRF
Frequency Range	22MHz-2.2GHz	1MHz-6GHz	100kHz-3.8GHz	300MHz-3.8GHz
Duplex	N/A	Half	Full	Full
Open Source	No	Full	Full	Schematic, Firmware
Price	$10	$299	$299	$420

Figure 2.6 Table of existing SDR hardware

To start the implementation of a GSM network, a full documentation on the different technology is needed. Different types of hardware exist with different specifications, some are half duplex, they can only receive or transmit at a time and some of them are full duplex, they can receive and transmit at the same time. In Europe, the GSM frequencies are around 900MHz, the hardware should then include these frequencies.

All the implementation is performed on Kali Linux.

Passive - RTL-SDR

The passive IMSI catcher capture IMSI numbers for each phone that initialize a connection with a base station.

Hardware

Figure 2.7 RTL-SDR

For this passive IMSI, an SDR receiver to catch GSM frequencies is needed as hardware. The RTL-SDR, very cheap and can receive from 22 MHz to 2,2 GHz in the GSM frequencies. For this project, the DVB-T will be used.

Software

To make this IMSI, some software needs to be installed. The first software is “GNU Radio”, a framework for signal processing and “gr-gsm”, tools for “GNU Radio”. These two software brings the ability to retrieve GSM transmissions.

The command to install “GNU Radio”:

$ sudo apt install gnuradio gnuradio-dev

To install “gr-gsm”, check and install the missing software and libraries needed:

$ sudo apt install git cmake autoconf libtool pkg-config g++ gcc make libc6 libc6-dev libcppunit-1.14-0 libcppunit-dev swig doxygen liblog4cpp5v5 liblog4cpp5-dev python-scipy gr-osmosdr libosmocore libosmocore-dev

Then build and install “gr-gsm”:

$ git clone https://git.osmocom.org/gr-gsm

$ cd gr-gsm

$ mkdir build

$ cd build

$ cmake ..

$ make

$ sudo make install

$ sudo ldconfig

To analyze the GSM transmission, “Wireshark” can be used to view the GSM packets captured with “gr-gsm”. To make it more readable, a python script “IMSI-catcher” exists on GitHub to analyze and extract IMSI numbers from the GSM packets captured.

To install “Wireshark”:

$ sudo apt install wireshark

To install python and get the python script for the IMSI-catcher:

$ sudo apt install python-numpy python-scipy python-scapy

$ git clone https://github.com/Oros42/IMSI-catcher.git

With this setup, GSM communications can be intercepted but the communication is usually encrypted, so the only information that we can get with the passive implementation is the IMSI, country of the subscriber, brand, operator, Mobile Country Code (MCC), Mobile Network Code (MNC), Local Area Code (LAC) and CellId (CID).

Active - BladeRF

The active IMSI catcher brings the ability to create a portable GSM base station to create a private GSM network. The network created is vendor free. With this system a man in the middle attack is possible and messages and calls can be intercepted. It is important to note that the GSM protocol works by full duplex action which means is receiving and sending data out at the same time. Therefore, it was decided that the BladeRF was the most suitable hardware for this project because of its capabilities

Hardware

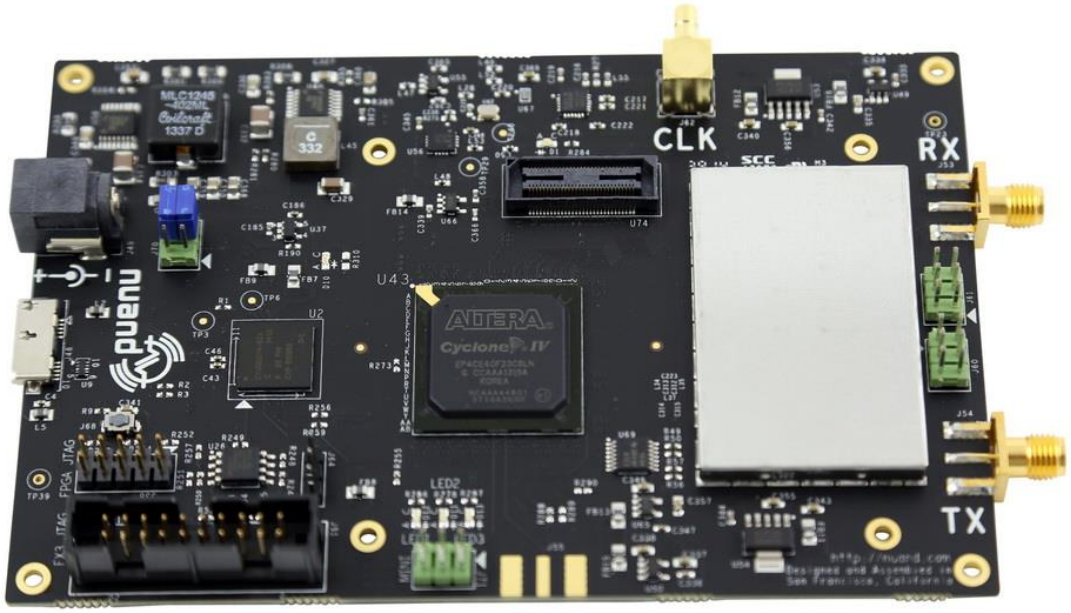


Figure 2.8 BladeRF

A full duplex software defined radio is required to create a GSM base station. The BladeRF with a range between 300 MHz to 3.8 GHz and full duplex. It is not the cheapest full duplex hardware, but it is the most documented and developed in its software. This hardware is the most expensive of all the SDR cards shown in this thesis.

Software

Firstly, make sure that all necessary software are installed:

$ sudo apt-get install git apache2 php bladerf libbladerf-dev libbladerf0 automake

(php 5 or 7 is preferred) we need apache for the web user interface. On some kali versions, **"libbladerf0"** needs to be install using the .deb file located in *https://pkgs.org/download/libbladerf0.*

With all this, the communication with the BladeRF is already possible, the command "dmesg | grep usb" should show that the blade is connected.

Then create a group named "yate" to allow non super user to run "Yate":

$ sudo addgroup yate

$ sudo usermod -a -G yate $user

After this step, we need to create a udev rule to ensure a user can access to the BladeRF hardware

create the file 90-yate.rules in /*etc/*udev/

$ sudo nano /*etc/*udev/90-yate.rules

and put this content inside :

 # Nuand BladeRF

ATTR{idVendor}=="1d50", ATTR{idProduct}=="6066", MODE="660", GROUP="yate"

Those values can be obtained with the "dmesg | grep usb" command. This provide an *88-nuand.rules* udev rule.

Unplug the device and enter the following command:

$ sudo udevadm control --reload-rules

This makes the hardware accessible for configuring.

Compiling Yate

To compile Yate:

$ svn checkout http://yate.null.ro/svn/yate/trunk yate

$ svn checkout http://voip.null.ro/svn/yatebts/trunk yatebts

$ cd yate

$ svn update

$ ./autogen.sh

$ ./configure --prefix=/usr/local

$ make

$ sudo make install

$ sudo ldconfig

To compile YateBTS:

$ cd ../yatebts

$ svn update

Now because of compatibility issues to do with the compiler (gcc version 5) we need to fix few files.

The file is in “mbts/GPRS/MSInfo.cpp”

Replace the line:

GPRSLOG(INFO,GPRS_MSG|GPRS_CHECK_OK) << "Multislot assignment for "<<this<<os;

With the line:

GPRSLOG(INFO,GPRS_MSG|GPRS_CHECK_OK)<<"Multislot assignment for "<<this<<(!os.fail());

It just adds “.str()” at the end of “os” for the compatibility.

For a gcc in the version 6, 7 or 8, the next lines need to be modified, in the file "mbts/SGSNGGSN/Sgsn.cpp", replace the lines:

SGSNLOGF(INFO,GPRS_OK|GPRS_MSG,"SGSN","Removing SgsnInfo:"<<ss);

SGSNLOGF(INFO,GPRS_OK|GPRS_MSG,"SGSN","Removing gmm:"<<ss);

by:

SGSNLOGF(INFO,GPRS_OK|GPRS_MSG,"SGSN","Removing SgsnInfo:"<<(!ss.fail()));

SGSNLOGF(INFO,GPRS_OK|GPRS_MSG,"SGSN","Removing gmm:"<<(!ss.fail()));

This applies with the other file as we need to add .str() to "ss" for compatibility reasons.

After fixing this bit of code the installation can now begin.

$ ./autogen.sh

$ ./configure --prefix=/usr/local

$ make

$ sudo make install

$ sudo ldconfig

Some permissions need to be fixed to allow local users to be able to configure:

$ sudo chown root:yate /*usr/*local/etc/yate*.conf

$ sudo chmod g+w /*usr/*local/etc/yate/*.conf

Now, few starter configurations are needed in the file "/usr/local/etc/yate/ybts.conf" to set the BladeRF to the right frequency range:

Radio.Band=900

Radio.C0=100

Identity.MNC=01

Identity.LAC=1000

The most important part is to set the “French mode” also called no encryption:

Cipher.Encrypt=no

After set the transceiver scheduling priority in the same file:

radio_read_priority=highest

radio_send_priority=high

Configuring SNMP port number in “/usr/local/etc/yate/ysnmpagent.conf”

port=161

remote_port=162

Now configuring subscriber devices (this allows a user to connect by making a regular expression) in “/usr/local/etc/yate/subscribers.conf”:

regexp=.*

It will allow any user to connect to the BTS.

The last thing before monitoring with Wireshark is to edit the file “/usr/local/etc/yate/ybts.conf” and the section [tapping]. Make sure to have this option is set:

GSM=yes

GPRS=yes

TargetIP=127.0.0.1

To be able to connect to the web interface:

$ cd /var/www/html/

$ ln -s /usr/local/share/yate/nipc_web/

$ chmod -R a+rw /usr/local/etc/yate/

Chapter 4: Testing and Results

Passive Testing

Now that everything is prepared, we are now able to work with the GSM network. Firstly, launch the command:

$ sudo grgsm_scanner

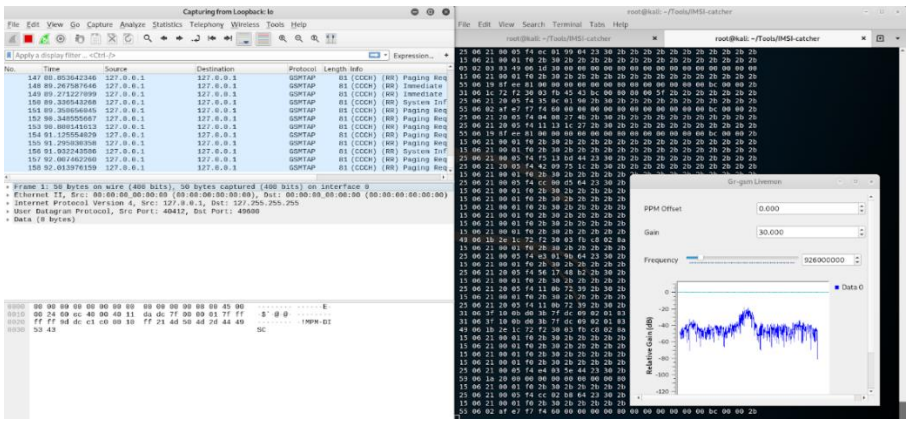
This command shows GSM antennas with different frequencies available in the area.

```
root@kali:~/Tools/IMSI-catcher# grgsm_scanner -b GSM900 -v

ARFCN:  979, Freq:  926.0M, CID: 11804, LAC:  1019, MCC: 272, MNC:   3, Pwr: -27
  |---- Configuration: 1 CCCH, not combined
  |---- Cell ARFCNs:
  |---- Neighbour Cells: 975, 976, 977, 978, 980, 981, 982, 983, 984, 985, 986, 988, 992, 993

ARFCN:  983, Freq:  926.8M, CID: 11805, LAC:  1019, MCC: 272, MNC:   3, Pwr: -37
  |---- Configuration: 1 CCCH, not combined
  |---- Cell ARFCNs:
  |---- Neighbour Cells:

ARFCN:   67, Freq:  948.4M, CID:     0, LAC:     0, MCC:   0, MNC:   0, Pwr: -29
  |---- Configuration: Unknown
  |---- Cell ARFCNs: 51, 74
  |---- Neighbour Cells:
```

By choosing one of the frequencies shown with the last command, we can launch a live data gathering with:

$ sudo grgsm_livemon -f 949.8M

Wireshark can also be launched listening on the loopback. All this shows all the data exchanged under this frequency.



“grgsm_livemon” opens a graphical window showing a graph on the frequency chosen. Here the frequency can be easily modified.

The next command is to get IMSI numbers from new connections in the area:

$ python simple_IMSI-catcher.py –sniff

Active Testing

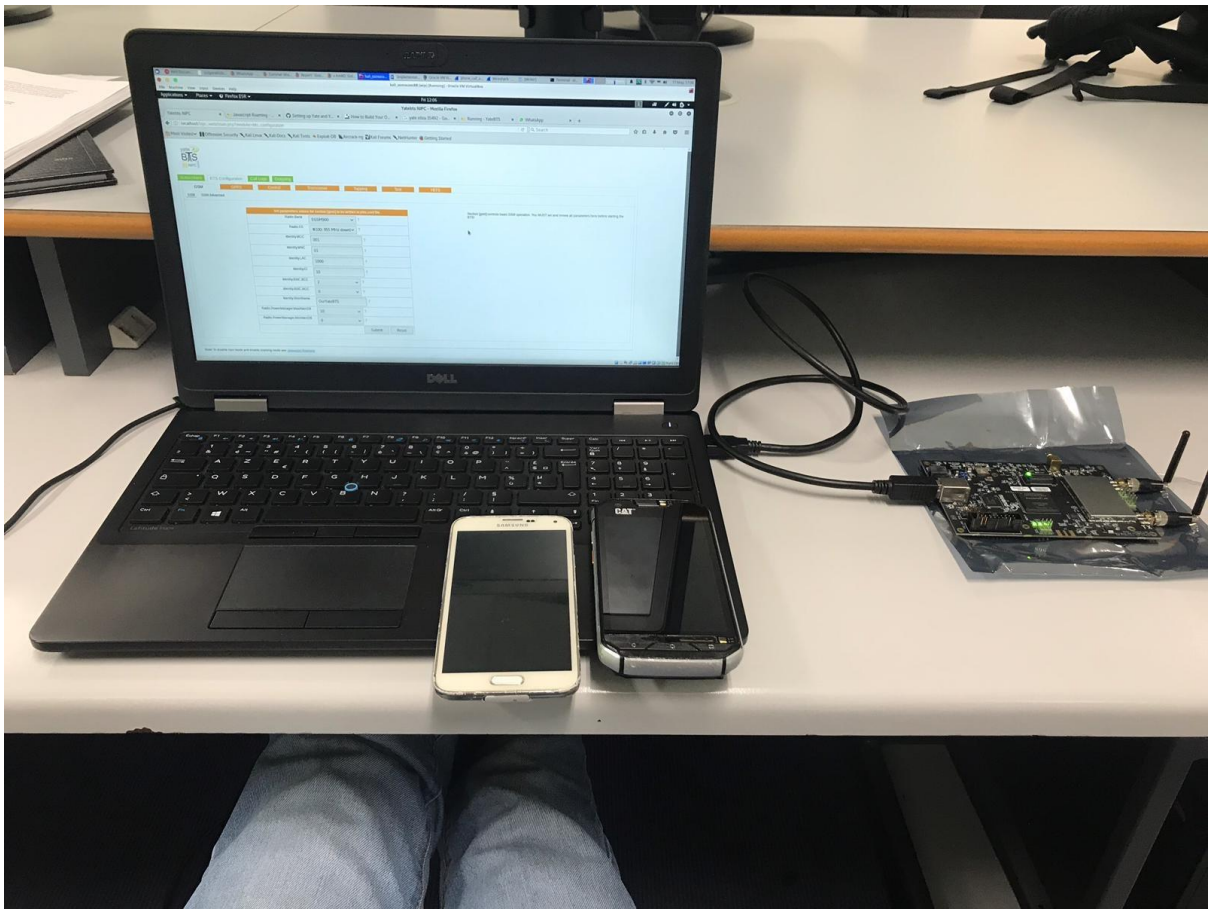


Figure 2.9 Test Setup

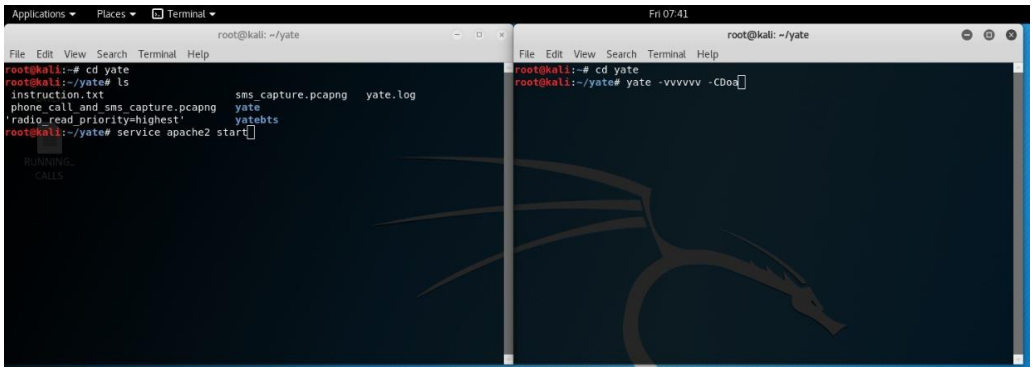
Now that the software and hardware have all been configured correctly, the next step is to run and do some tests on the vulnerabilities mentioned. So, the goal for the tests is to show the vulnerabilities and to exploit them.

This section of the thesis explains the process of broadcasting the base station and what happens when an attacker has users connected to their fake base station. The tests will cover sniffing, capturing traffic and spoofing of the GSM network. These tests are explained to show the different ways that users are vulnerable through the GSM protocols.

The phones that were used for the testing of the project are Samsung S5, CATs60 and OnePlus3. It is important to note that some the Samsung S5 picked up the fake base station as “00101” and the CATs60 and OnePlus3 picked up the station as “Test PLMN”.

YateBTS

Now with all the configurations being made, we can now run Yate and test the home-made GSM network. Before starting yate the apache server service must be started. The command **service apache2 start** is used. Open a new terminal and go into the yate folder and from here enter **yate -vvvvvv -CDoa.** This command



After the Yate service has started, the next step is to confirm that the fake base station is being broadcasted. This can be confirmed by using **sudo yate -s -vv**. Below we see that the mobile base station has connected to Yate base station.

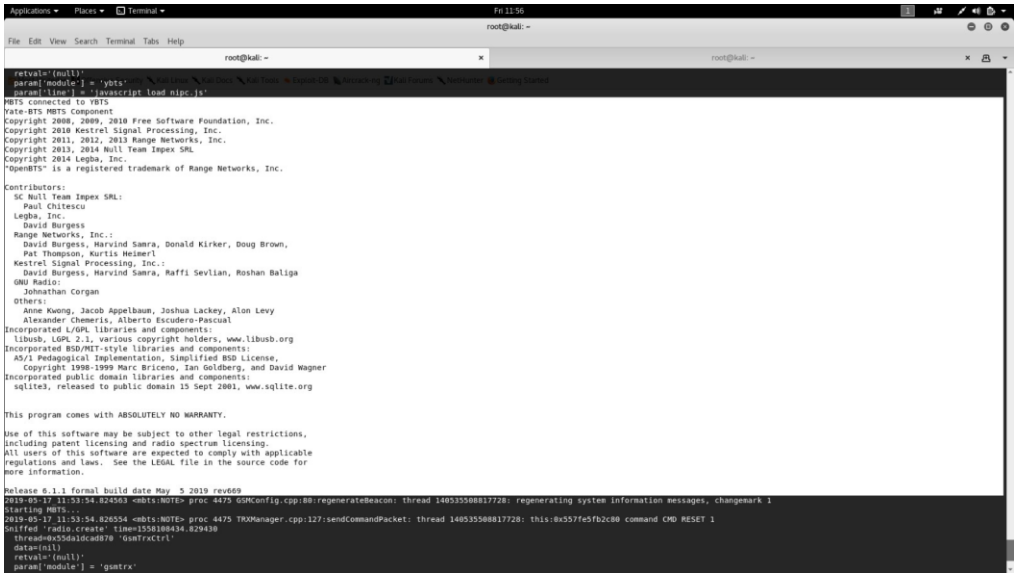


Figure 3.1 Successful connection to YateBTS

The YateBTS setup has been successful and it is running correctly with no error messages.

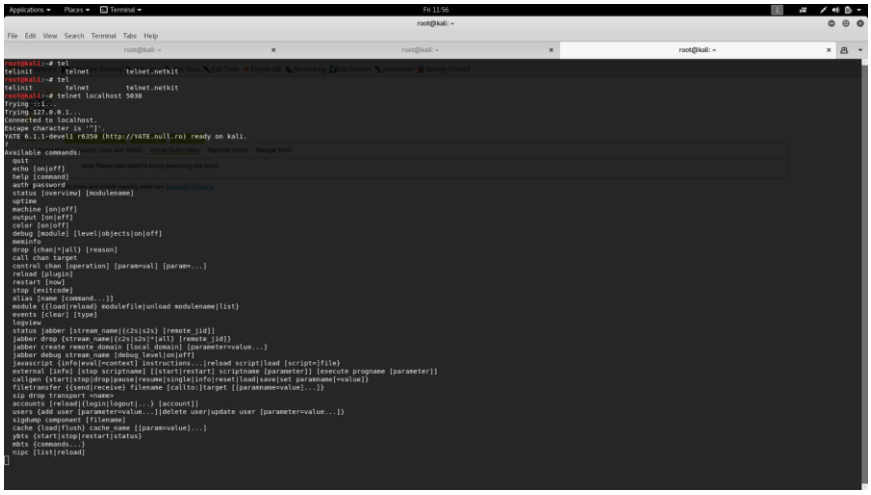


Figure 3.2 Telnet to BTS

For users who prefer to configure the YateBTS through a console must use the telnet command to connect to the base station. Configurability through the console is not available and the user must use **telnet localhost 5038**. See figure 3.2 above to see the available commands to configure the YateBTS.

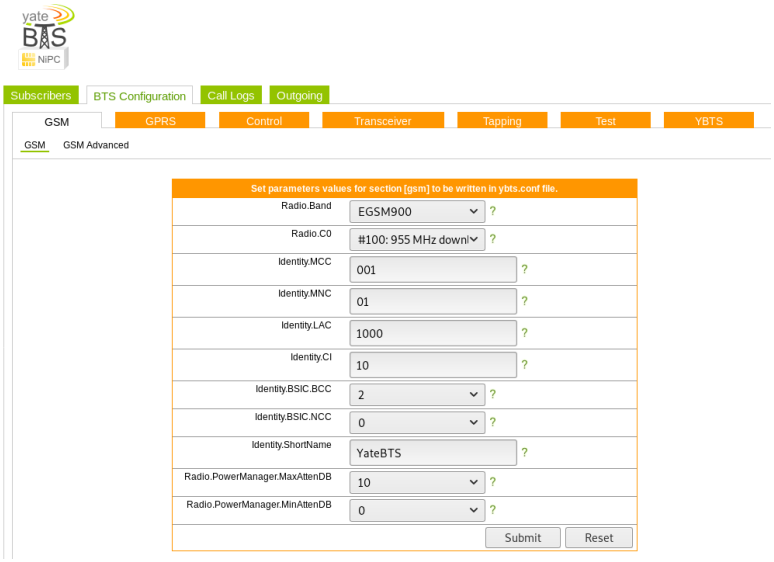


Figure 3.3 YateBTS

Now to check that YateBTS is running the “” is used. From figure 3.3 we can see YateBTS is running successfully. The next step after connecting to the Yate Base Station is to check the configuration of the BTS. For Ireland the frequencies used is mostly on the radio band of EGSM900 and the rest of the configuration can be left at the default settings. The broadcasting of the base station should now be working. Mobile Country Code (MCC) is shown by “001” and the Mobile Network (MNC) “01”. The base station can.

Creating Fake Base Station

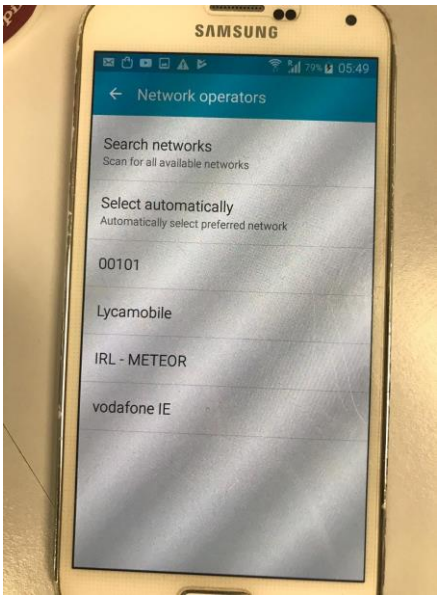


Figure 3.4 BTS network connection

After starting Yate and broadcasting, the base station should now be available for users to connect freely. This is seen in figure 3.0, where the base station is shown as “00101”.

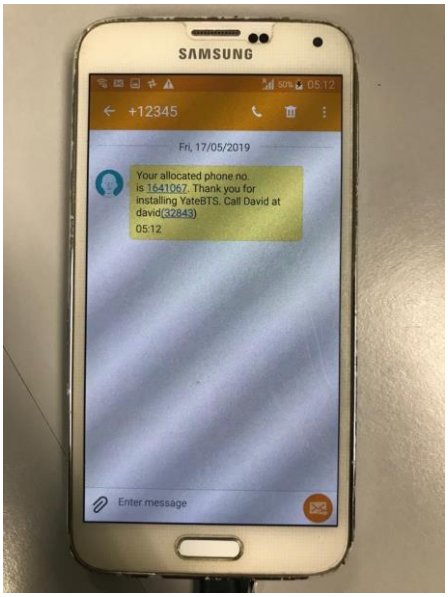


Figure 3.6 Connected to base station

After choosing the “00101” network the user has now successfully connected to the Yate Base Station. The user receives a text message to confirm they have connected, and they receive an allocated number – 1641067.

Sending SMS through YateBTS

When the two test phones are connected to the network, a simple SMS is sent to each phone to test the network connection. The figure below shows that the mobile devices can communicate with each other. The subscribers can now confirm that communication by text is functioning properly.

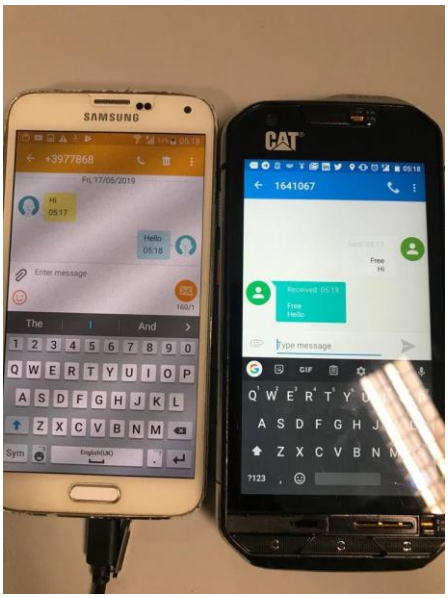


Figure 3.7 SMS test

It is worth to note that a user can test out if the network functions properly using only one mobile device. ELIZA is an AI chatterbot that is built into YateBTS to test the SMS capabilities of a network. When the user has successfully connected to the Yate network, they can then proceed by sending a message to ELIZA on the number 35492. In the figure below, the conversation between one user and ELIZA can be seen to show that the SMS test can be done using one mobile device.

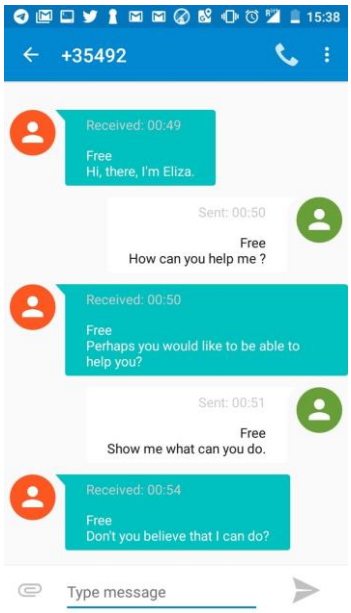


Figure 3.8 ELIZA AI chatterbox

Testing Phone Call

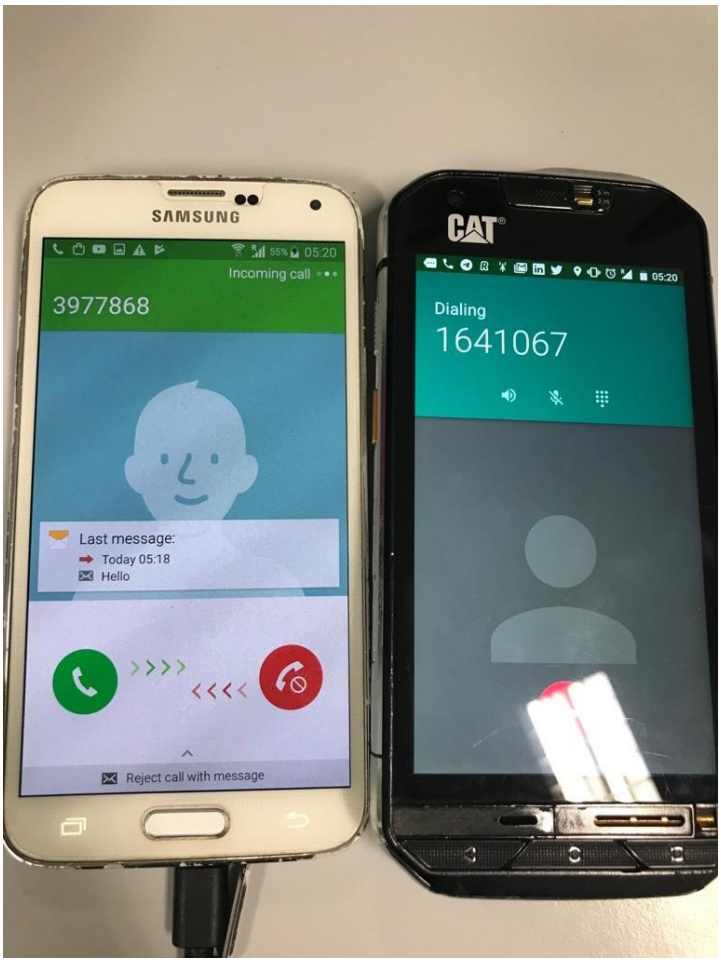


Figure 3.9 Phone call test

After confirming that send SMS messages between the two mobile devices, the next step was to perform a call test. Using the default numbers assigned by the base station, the call was placed to confirm that users connected on the network are able to communicate each other through voice calls.

SMS Man in the Middle Attack

When configuring the base station, the encryption was set to none. So, the next test was to perform a man in the middle attack through SMS. The test plan for this attack included the two phones, wireshark and the fake network. Wireshark was set up to capture traffic through the Loopback interface. Once this is done, the attacker can now start the capture and wait for the communication to begin. When the message has been sent, the filter shown "**gsmtap && !icmp && !dns**" was used to filter out icmp and dns information and only show gsm data. To find SMS texts the filter "**gsm_sms**" is used to again filter out all the GSM traffic to only show SMS in GSM.

In figure 3.9 below the attacker has filtered out the traffic to show GSM SMS protocol. The second line of the captured packets can be further analysed under specific headers. When the header of **TP-User-Data** is clicked, the text message can now be seen in plaintext where it shows "SMS text: Hi". The reply can be seen in figure 4.0, the reply "Hello" can be seen by using the same steps to view the first text.

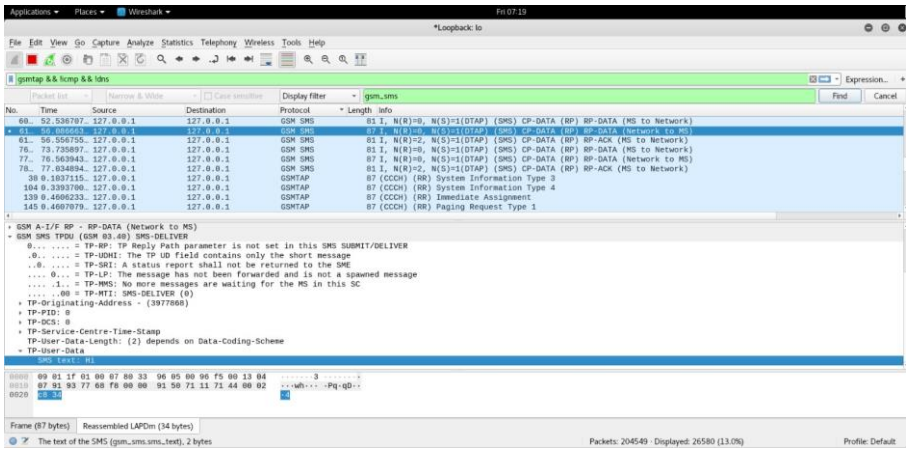


Figure 3.9 SMS text

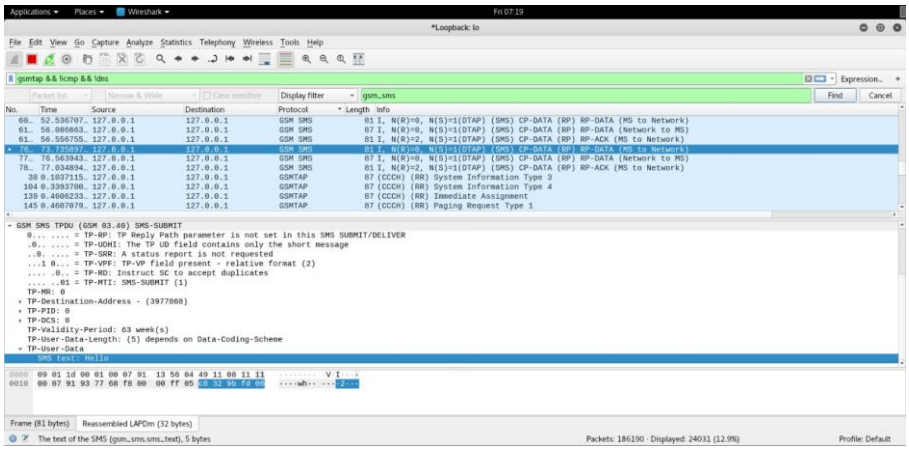


Figure 4.0 SMS reply

Chapter 5: Mitigation

In the world today, 2g encryption has been overtaken by 3g, 4g and in the near future 5g. However, users are still vulnerable to attacks through the GSM architecture. The average user will not have knowledge about the vulnerabilities and thus are susceptible to attacks. After doing some research, a question had to be asked; How would one know they are connected to a legitimate base station?

After some research there were a few applications found for mobile phones which were free to download off the android play store.

Cell Spy Catcher

Cell Spy catcher is an application on the google play store that is free to download. It allows a user to check if their mobile phone is connected to a rogue base transceiver by checking if the base transceiver station is registered on the OpenCellID Database. OpenCellID is a public database of base transceiver stations that is trusted by the telephony providers services. The application detects if the user is connected to a fake base station based on the location of the phone and the location of the base station they have connected to.

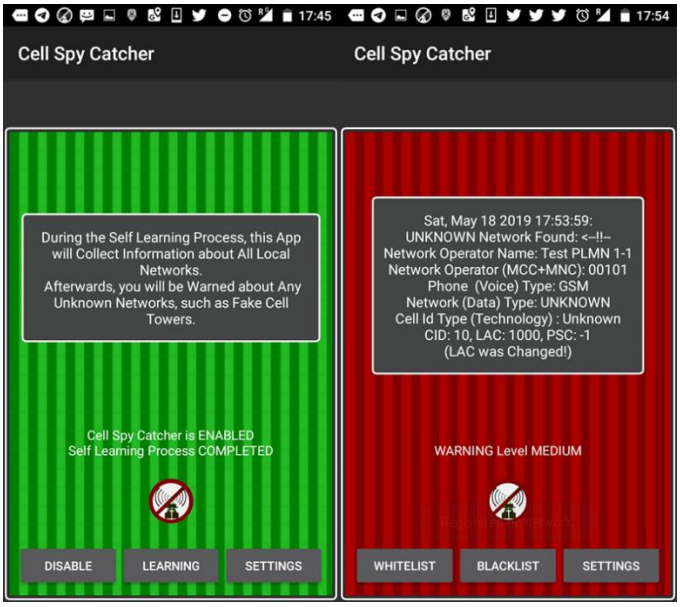


Figure 4.1 Cell Spy Catcher

The figure above shows that if a subscriber is connected to a genuine transceiver station, then the background will be green to indicate it isn't a fake one. However, if the users' mobile phone is connected to a fake base station, it will come up as red. The fake base station, "Test PLMN 1-1" with the same MCC and MNC – "00101" that was set up for this project was recognized as an unknown network and not being registered in the OpenCellID Database. The figure explains also that the local area code (LAC) was changed and that base station has a warning level of "MEDIUM".

SnoopSnitch

SnoopSnitch is a very powerful service which needs to have root access on the mobile device for it to be able to run. The application uses records of countries based on the registered base stations through gsmap.org. It allows the user to check for a silent SMS attack which allows an attacker to get the IMSI of the user. The application can be downloaded through the google play store but because of some security reasons the application will have some restrictions on some of the features. SnoopSnitch can be downloaded online as well and it will have all the full access and no restrictions compared to the one on google play store.

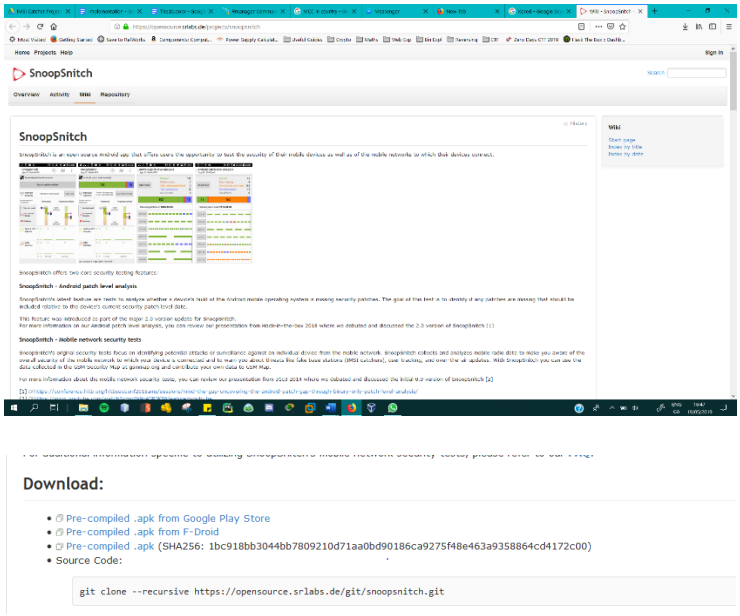


Figure 4.2 SnoopSnitch download

The figure shows the mobile station in its normal state. It shows the operator being Vodafone and the other registered licenced provider, Three network, in the area. The figure shows a comparison between the two providers and the level of protection it has against impersonation and interception.

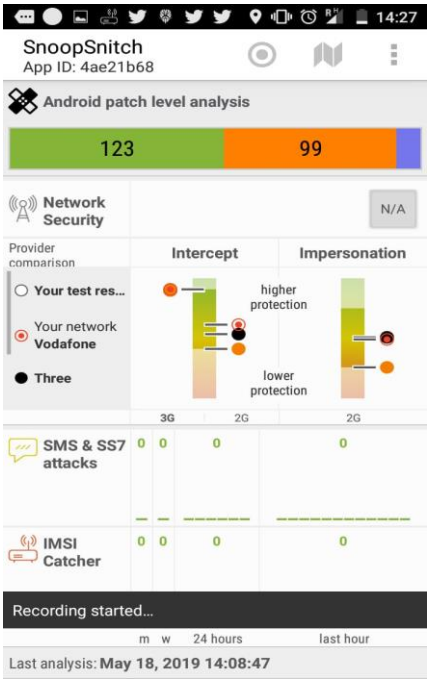


Figure 4.3 Connected providers

Here, the figure contains a map of where the mobile phone is located and the level of protection against attacks per area.

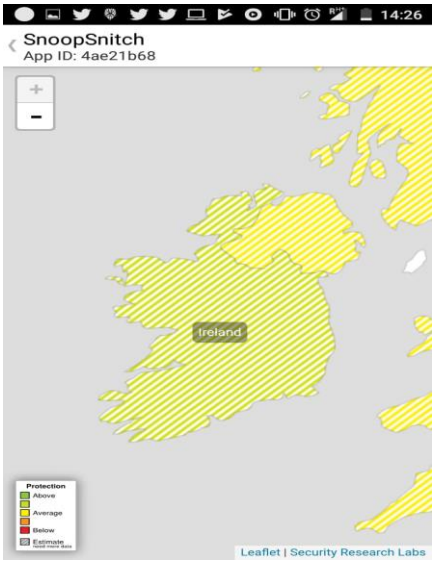


Figure 4.4 SnoopSnitch Map

The figure below shows the phone being connected to a fake base station and there is no comparison between the operators available. This is usually a clue to say that the user is connected to a fake base station. This means that the base station is not located in the area that it broadcasts itself to be.

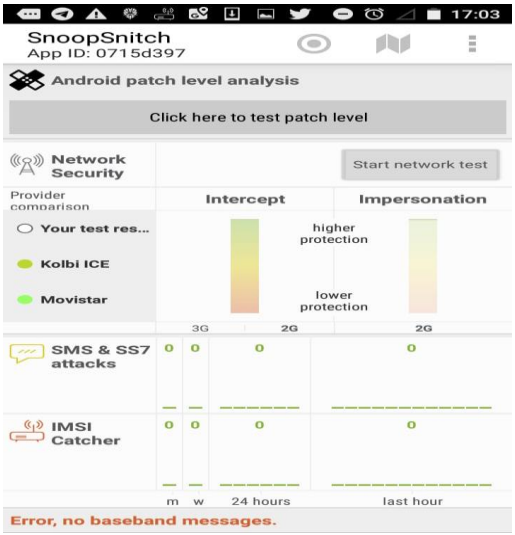


Figure 4.5 Fake BTS analysis

In figure 4.6 it shows an SMS that the user receives when a network test is used. When the network test is finished and successful, the user receives a text and the SnoopSnitch service is running also. In figure 4.7 the network scan is not possible as there were no baseband messages received during the active network test.

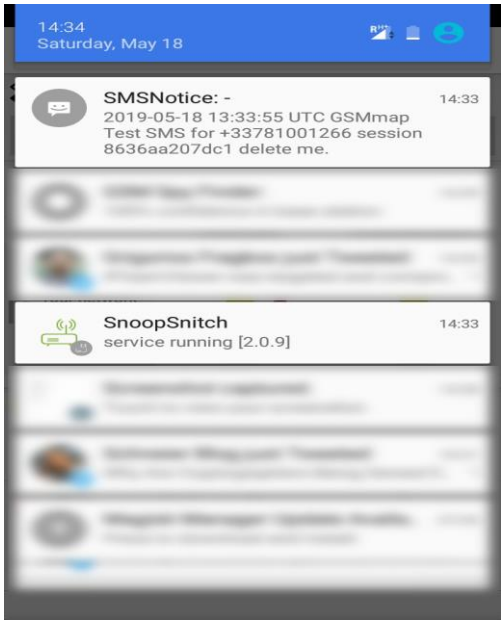


Figure 4.6 SMS notice

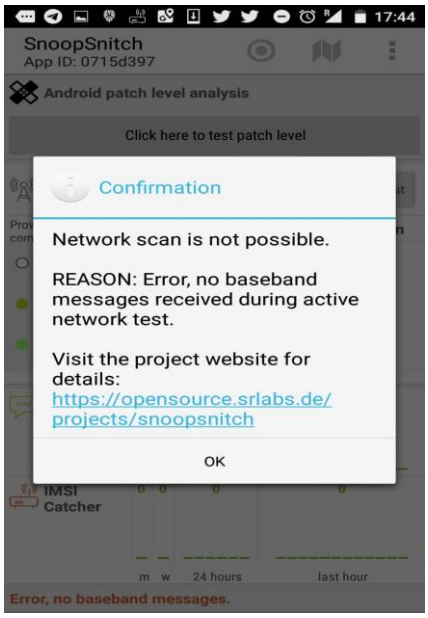


Figure 4.7 Failed configuration

The vulnerabilities are found at the core of the GSM architecture. These problems still exist and haven't been fixed which is why attackers are still able to exploit these vulnerabilities. If a fake base station has a powerful enough signal, then users will automatically be connected to the network. To add more security, it's possible to add two-way authentication between the BTS and MS for more security. Another option would be upgrading from using 2g to 4g or 5g and when texting uses a secure encrypted channel of communication through use applications such as Telegram, Signal or WhatsApp.

To prevent people from eavesdropping on communications, there are hardware used to detect some unusual frequencies. There is some hardware that also can detect an IMSI catcher such as Anti-Spy Wireless RF Signal Detector Bug T-8000 by Shenzhen Qin Pu Technology Co., Ltd. This hardware is used to detect radio frequencies and it is able to see if it is a fake base transceiver station.

Comparison

The RTL-SDR (Realtek RTL2832U also called DVB-T) is a chipset that is normally used for receiving TV signals. Further research and past projects from other people have shown that the RTL is capable of more than just receiving tv signals. From the frequencies between 500KHz to 1.75 GHz it can be used with **kalibrate-rtl**, **gr-gsm** or **airprobe** to decode GSM traffic. These tools can be found on GitHub This is the cheapest option to use and can be used for anyone who wants to discover how the GSM protocol works.

The HackRF is a half-duplex software defined radio. This means it can receive and transmit frequencies, however, not at the same time. The HackRF has the largest frequency range as it goes from 1MHz to 6GHz. Kalibrate-rtl can be used together with the HackRF to sniff and capture GSM traffic.

LimeSDR and LimeSDR Mini they are both two full duplex devices. On paper, they have all the specification to run a fake BTS. They have frequency ranges of 100KHz to 3.8GHz in full duplex so they can broadcast in the 900MHz bandwidth (the one is used in the GSM Protocol). The lack a of public documentation and implementation makes it difficult to get the software defined radio working. It is relatively new on the market and was released at around the same time the project had started. The LimeSDR mini wasn't able compatibility issues with the software that was chosen and because there is almost no complete documentation online, it was not possible to implement the fake base station using LimeSDR. During the project there was some conversations with a few with professionals of radio frequencies through the Hack In the Box Conference in Amsterdam. They advised that they were not able to implement the fake base station project and said there was not enough documentation as the device is still relatively new.

BladeRFx40 is also a full duplex transceiver that has frequency range of 300MHz to 3.8GHz and that is perfectly compatible with YateBTS. The BladeRF was chosen as it is the most documented of all the software defined radio devices. There were clearer set up and compiling instructions available online and there were some projects that specifically detail the set up of the fake base station.

Chapter 6: Conclusion

The objective of this was to research on GSM and find existing vulnerabilities and show how an attacker can exploit them through the 2G network architecture. The attacks mentioned in this report, surprisingly are relatively affordable depending on the hardware someone uses. From the cheapest RTL-SDR costing as low as 7 euros. to the most expensive which is BladeRF costing from 450 euros and above. With the right refences, one can easily recreate building a home-made rogue base station.

The goals of this project were to show how a user can be vulnerable and detail the attacks through 2G. This was possible by implementing the following:

1. Understanding the GSM architecture and how homemade base stations could act as a man in the middle attack.
2. SMS man in the middle attack was easy and successful to implement. The use of wireshark to monitor traffic and capture is what helped perform the attack.
3. Using half duplex SDR's to sniff GSM traffic that is sent across the network. The packet sniffing allowed the obtaining of the information of the user.

A conclusion can now be reached that sniffing, breaching, spoofing 2G is not expensive at all and can be done by anyone with little knowledge of how the GSM architecture works. Resources online and through GitHub are readily available for anyone to use whether for educational purposes or for malicious intent. This thesis explained certain ways that a user can stay safe from these types of attacks. The certain mitigation methods that were explained could be applied and could be of help when a person is unsure of connecting to networks around them.

In general, a person should disable 2G in the network settings in their phone. This method alone already eliminates the risk of attacks. However, it is important to note that there are still countries using 2G as their main telecommunications technology for their citizens.

Further Work

Another option for further work will be the intercepting phone calls. It is still to be discovered with the LimeSDR. Some online research show that is possible to make an LTE(4G) Network.

Further research on Software Defined Radio that is able to receive and transmit a wide range of frequency such as WIFI (2.4 GHz and 5GHz), Bluetooth(2.485GHz), AM(535-1605 kHz) and FM(88-108 MHz). It is possible using a microcontroller such as the raspberry pi and an SDR device to capture and replay a signal to open house doors, cars, etc.

Another possibility of a project is to work on low frequencies and then inspect and analyze and NFC (13.56MHz) card.

Another idea of research is how is it possible to make a forensics investigation against an attack over the air (radio frequencies).

Acronyms

RF: Radio Frequency

SDR: Software Defined Radio

IMSI:International Mobile Subscriber Identity

TMSI: Temporary Mobile Subscriber Identity

IMEI: International Mobile Equipment Identity

MSIN: Mobile Subscriber Identification Number

BTS: Base transceiver station

BSC: Base Station Controller

MS: Mobile Station basically the phone or smartphone

ME: Mobile Equipment

SPN: Service Provider Name

GMSC: Gateway Mobile Switching Center

HLR: Home Location Register

VLR: Visitor Location Register

EIR: Equipment Identity Register

Auc: Authentication Center

PSTN: Public Switch Telephone Network

SMS: Short Messaging System

SMSC: Short Messaging Service

NIB: network In the Box

NIPC: Network Int a Portable Computer

MCC: mobile country code

MNC: mobile network code

LAC: Local Area Code

ARFCN: Absolute radio-frequency channel number

GSM: (Global System for Mobile Communications) in short this is the technology being used to transmit and receive voice and text.

EGSM900: extension of the GSM-900 frequency range.

GPRS: (General Packet Radio Service) is a packet oriented mobile data service. This is how you get your Internet on a 2g or 3g cell network.

EDGE: Enhanced Data Rates for GSM Evolution

Cell ID: unique number used to identify each BTS

CID: CellID (identifier number)

LAC: Location Area code, allow to group all every cell in the same area.

MCC: Mobile Country Code it's always the same number whenever you are inside the country (IRE = 272)

MNC: Mobile Network code, allow to identify the provider of service (like eir, three …)

LAI: Local Area Identify LAI

A5/0: means "no encryption". Data is sent unencrypted. In some countries, this is the only allowed mode (India is such a country).

A5/1: is the old "strong" algorithm, used in Europe and North America.

A5/2: is the old "weak" algorithm (it is not recommended in the GSM specifications)

A5/3: is the newer algorithm for GPRS/UMTS

KC: symmetric encryption key used in GSM protocol

TDMA: Time Division Multiple Access

LFSR: Linear-feedback shift register

ARFCN: Absolute Radio Frequency Channel Number

SIM: Subscriber Identity Module

OTA: Over the Air

AI: Artificial Intelligence

Bibliography

Irishstatutebook.ie. (2018). Wireless Telegraphy Act, 1926. [Online]
Available at: http://www.irishstatutebook.ie/eli/1926/act/45/enacted/en/print#sec12
[Accessed 17 May 2019].

Commission for Communications Regulation. (2018). Licensing - Commission for Communications Regulation. [Online]
Available at: https://www.comreg.ie/industry/radio-spectrum/licensing/ [Accessed 17 May 2019]

Commission for Communications Regulation. (2018). Permitted Short Range Devices in Ireland - Commission for Communications Regulation. [Online]
Available at: https://www.comreg.ie/publication/permitted-short-range-devices-ireland [Accessed 17 May 2019].

Baranoff, B. (2019). OpenAirInterface 4G/LTE with LimeSDR_Mini. [Online] Linux hacking. Available at: https://bastienbaranoff.wordpress.com/2018/10/23/openairinterface-4g-lte-with-limesdr_mini [Accessed 17 May 2019].

Extreme subversive tendencies. (2016). Building a portable GSM BTS using the Nuand BladeRF, Raspberry Pi and YateBTS (The Definitive and Step by Step Guide). [Online]
Available at:
https://blog.strcpy.info/2016/04/21/building-a-portable-gsm-bts-using-bladerf-raspberry-and-yatebts-the-definitive-guide [Accessed 17 May 2019].

Wiki.myriadrf.org. (2019). LimeSDR-Mini - Myriad-RF Wiki. [Online]
Available at: https://wiki.myriadrf.org/LimeSDR-Mini [Accessed 17 May 2019].

Ieeexplore.ieee.org. (2008). Solutions to the GSM Security Weaknesses - IEEE Conference Publication. [Online]
Available at:
https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4756489 [Accessed 17 May 2019].

Anon, (2015). [Online]
Available at:
https://www.researchgate.net/publication/283723257_Investigating_Vulnerabilities_in_GSM_Security [Accessed 17 May 2019].

Fr.wikipedia.org. (2018). Base transceiver station. [Online]
Available at: https://fr.wikipedia.org/wiki/Base_transceiver_station [Accessed 17 May 2019].

Cryptome.org. (2009). CRACK A5. [Online] Available at: https://cryptome.org/jya/crack-a5.htm [Accessed 17 May 2019].

Opensource.srlabs.de. (2018). Wiki - A5/1 Decryption - SRLabs Open Source Projects. [Online] Available at: https://opensource.srlabs.de/projects/a51-decrypt [Accessed 17 May 2019].

scribd.com. (2019). How to Capture-Analyze-crack GSM | Short Message Service | Gsm. [Online] Available at: https://www.scribd.com/document/249405291/How-to-Capture-Analyze-crack-GSM [Accessed 17 May 2019].

Crazy Danish Hacker. (2019). GSM Sniffing: Requirements - Software Defined Radio Series #2. [Online] Available at:
https://www.youtube.com/watch?v=3dridHDUHJQ&list=PLRovDyowOn5F_TFotx0n8A79ToZYD2lOv [Accessed 17 May 2019].

Mcc-mnc.com. (2013). Most up to date list of MCC and MNC codes: mobile country codes – mobile network codes. [Online] Available at: http://mcc-mnc.com [Accessed 17 May 2019].

En.wikipedia.org. (2019). Mobile country code. [Online]
Available at: https://en.wikipedia.org/wiki/Mobile_Network_Code [Accessed 17 May 2019].

Kamau, C. (2015). Sniffing GSM Traffic - The poetry of (in)security. [Online] Ckn.io. Available at: https://www.ckn.io/blog/2015/11/01/sniffing-gsm-traffic [Accessed 17 May 2019].

Margaritelli, S. (2016). How to Build Your Own Rogue GSM BTS for Fun and Profit. [Online] evilsocket. Available at: https://www.evilsocket.net/2016/03/31/how-to-build-your-own-rogue-gsm-bts-for-fun-and-profit/ [Accessed 17 May 2019].

Nuand. (2017). Nuand/BladeRF. [Online]
Available at:
https://github.com/Nuand/bladeRF/wiki/Setting-up-Yate-and-YateBTS-with-the-bladeRF [Accessed 17 May 2019].

Brmlab.cz. (2016). GSM [brmlab]. [Online] Available at: https://brmlab.cz/project/gsm/start [Accessed 17 May 2019].

F - Droid. (2019). Snoop Snitch. [online]
Available at: https://f-droid.org/en/packages/de.srlabs.snoopsnitch/ [Accessed 17 May 2019].