The effectiveness of a homemade IMSI catcher build with YateBTS and a BladeRF

Kenneth van Rijsbergen[1]

[1]Student Msc System and Network Engineering, University of Amsterdam

***Abstract*—An IMSI-catcher, also known as a cell-site-simulator, is a telephone eavesdropping device mainly used by law enforcement. By simulating a cell-site it can force mobile phones to connect with the fake cell-site and force unencrypted (A5/0) communication. IMSI-catchers used to be expensive and only available to law-enforcement. Nowadays an IMSI catcher can be set up cheaply using a software defined radio and open source software such as OpenBTS. Deloitte would like to find out if this technology can be used for gathering OSINT (Open Source Intelligence) for red-teams.To set up the IMSI catcher the Blade-RF x40 was used running YateBTS. Then observations were made on how effective this IMSI catcher actually is in spoofing a modern phone. YateBTS is capable of simulating a 2.5G (GPRS) cell site and it was observed that phones always prefer the faster base station (regardless of signal strength). The general conclusion is that an IMSI catcher isn't as effective nowadays against modern phones, and require specific conditions along with forcing the use of 2G for the IMSI catcher to work.**

***Acknowledgements*—**I would like to thank the hacking team of Deloitte for enabling me to work and lending the necessary hardware to conduct the experiments. Also, I would like to thank Cedric van Bockhaven for helping with the initial setup of YateBTS, as well as pointing out the current status of A5/1.

Keywords—IMSI catcher, fake base station, GSM, SDR, MITM, OSINT, BladeRF, YateBTS

I. Introduction

An IMSI-catcher, also known as a cell-site-simulator, is a telephone eavesdropping device mainly used by law enforcement. The most commonly known IMSI-catching device is the "Stingray". It can run passively and collect IMSI's which can be used to identify a mobile subscriber. It can also actively intercept phone calls and SMS. By simulating a cell-site it forces mobile phones to connect with the stingray and to downgrade encryption to A5/0 (no encryption) or the weak A5/2 encryption.

An original stingray would have cost around $70,000 [8]. Nowadays it has been demonstrated that with a cheap Software Defined Radio (SDR) the same functionality can be achieved as the professional IMSI catchers.

Deloitte would like to find out in what way this technology can be used for gathering OSINT (Open Source Intelligence) for red-teams. A red team is a group of penetration testers that test an organisation's security from an outsider's point of view. Extracting phone numbers from a specific building and intercepting SMS would be useful to be able to do. Research will be done on how useful an SDR is to passively and actively capture GSM traffic for OSINT purposes.

A. Research questions

The main research question is:

"How may GSM be used for gathering Open source intelligence (OSINT) by a red team?"

This question is divided into the following sub-questions:

- How can an SDR be used to passively capture GSM traffic?
- How can an SDR be used to actively capture GSM traffic?
- What OSINT (Open Source Intelligence) may be extracted from this GSM data?

B. Related Work

Chris Paget demonstrated during DEFCON 18 (2010) how 2G traffic can be captured using a cheap SDR [20]. He demos this by creating a fake cell tower that can spoof an AT&T cell tower.

Another related work is the talk given by Karsten Nohl and Sylvain Munaut at the 27th Chaos Communication Congress [13]. At this talk, a demonstration was given on brute forcing captured GSM traffic (2G). For this, the "kraken" tool was used that can crack A5/1 encryption. They also demonstrated that a programmable phone of a few bucks was enough to capture this traffic.

In the paper titled "IMSI Catcher" by Daehyun Strobel [27] the authentication process of GSM and UMTS is analysed and how an IMSI catcher exploits these authentication weaknesses. It explains that the weakness of GSM mainly lies with one-sided authentication and the weak cryptography.

Finally, in the paper titled "Fake BTS Attacks of GSM System on Software Radio Platform" by Yubo Song, Kan Zhou and Xi Chen, an SDR was developed with a custom-made motherboard and an AM3517 single-board computer (SBC) [24]. With this SDR an IMSI catcher was made. This would spoof a cell site that had one of the weakest signals so other cell towers won't be interfered. Also, a selective jamming feature was created that can target an individual phone while leaving the other phones operational.

II. Background

A. IMSI

IMSI stands for "International Mobile Subscriber Identity" and is globally unique for each subscriber. The IMSI consists of 15 digits which contain the Mobile Country Code (MCC), Mobile Network Code (MNC) and the Mobile Subscriber Identification Number (MSIN). The IMSI is stored on the Subscriber Identity Module (SIM).

Since the IMSI is tied to a specific subscriber, “catching” an IMSI allows law enforcement to identify users that were at a certain location. The workings of an IMSI catcher will be covered in more detail further on in this paper.

B. Mobile phone generations

1) 1G (first generation): The first generation of mobile phones was implemented in the 1980s. The data send from and to the phones where analogue and naturally had no security whatsoever. Also, for 1G networks, it was only possible to make voice calls. Text messaging was not possible yet for 1G networks.

Technologies: AMPS, NMT, TACS, C-450, Radiocom 2000, RTMI, JTACS, TZ-801, TZ-802, and TZ-803 [19].

2) 2G (second generation): In the 1990s the second generation of mobile phone technology was rolling out. Features such as SMS, data, MMS, voice mail and call forwarding were implemented. Also, the radio signals became digital and were encrypted. Later 2.5G and 2.75G were introduced which both implemented improved techniques of data transfer such as GPRS and EDGE. The Global System for Mobile Communication (GSM) standard is the most widely used 2G standard and as of 2007 the most widely used mobile phone protocol in general [11].

Standards other than GSM are: IS-95 (a.k.a. cdmaOne), PDC, iDEN and IS-136 (a.k.a. D-AMPS) [19].

3) 3G (third generation): 3G was slowly rolled out in the 00s. The International Telecommunication Union (ITU) set up specifications that label certain mobile networks as 3G. 3G mobile networks support Global positioning system (GPS), mobile television and video conferencing. It also offers way more data transfer bandwidth and speed. Also, the encryption standard is improved: using two-way authentication between the mobile phone and the base station and having improved encryption standards.

Standards: UMTS, W-CDMA, TD-SCDMA (only in China), HSPA, and HSPA+, CDMA2000 [19].

4) 4G (fourth generation): 4G is also specified by the International Telecommunication Union (ITU). One of the requirements of 4G is a speed of 100 Mbit/s in a car or train and 1 Gbit/s for pedestrians [12]. A 4G internal network is also completely IP based, so no more circuit-switched telephone. It has to be noted that the current 4G standards are not actually fully compliant yet with the ITU specifications. However, they are still considered 4G since they are the closest to 4G speeds and are substantially better than 3G technologies.

Standards: Long Term Evolution (LTE) and Mobile WiMAX [19].

5) 5G (fifth generation): The proposed new standard that is planned to be rolled out around 2020 [17].

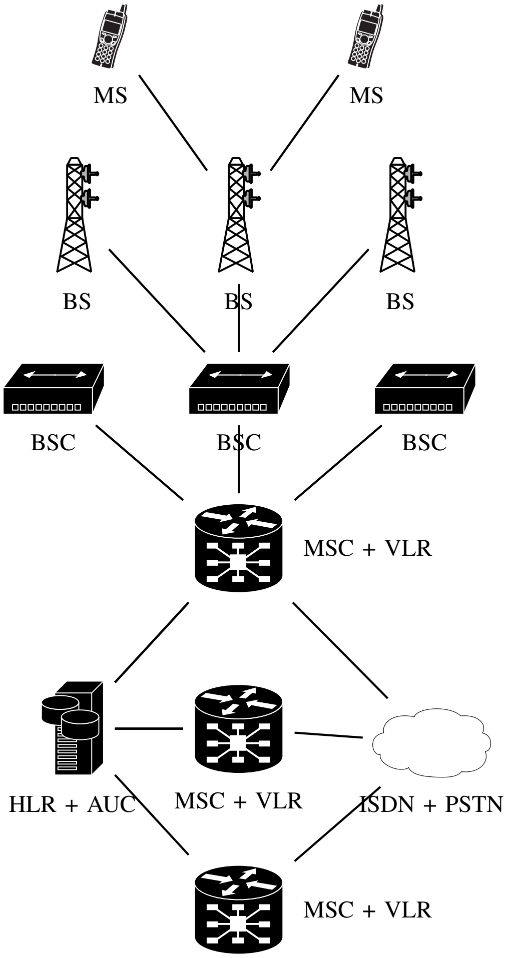


Figure 1: GSM network architecture

C. The GSM network

GSM (global system for mobile communications) will be the main focus of this paper. In this subsection, a brief overview will be given of the GSM network architecture and its lingo.

Figure 1 illustrates the hierarchy system of the GSM network. This network consist of the following components:

- **Mobile station (MS)** - The mobile station is the device that is able to access the GSM network via radio. The mobile station can be broken down into two separate parts, namely the mobile hardware and the SIM card.
- **Base Station (BS)** - This is the antenna and is also called the “cell tower” or “cell site”. One BS covers a particular cellular area in the cellular network. The size of this cell can vary from a few hundred meters to several kilometers. The size of the cell area depends on the landscape features and the population density of the area. In subway stations and large buildings, relay stations can be placed to act as repeaters. These relay stations then wire the signal to the nearest base station.

- **Base Station Controller (BSC)** - The base station controller controls several base stations. It handles the session handoffs between the different base stations when a user is moving through different cells. If, the base stations are not connected to the same BSC, then the handover is handled by the Mobile Switching Center (MSC).
- **Mobile Switching Center (MSC) and Visitor Location Register (VLR)** - The mobile switching center is responsible for managing the authentication, handover to the other BSCs and routing calls to the landline. To achieve this the MSC draws on the four different databases [27]: the HLR, VLR, AUC and the EIR.
  Each MSC has its own Visitor Location Register (VLR). The VLR holds subscriber information of subscribers that are under the care of the MSC (which are copied from the Home Location Register (HLR)). The VLR, for example, holds the Temporary Mobile Subscriber Identity (TMSI) which is a temporary alias for the IMSI. This is to reduce the frequent broadcasting of the IMSI.
- **Home Location Register (HLR)** - The HLR stores personal subscriber information like the IMSI and the phone number. There is only one HLR for every GSM network provider.
- **Authentication Center (AUC)** - The Authentication Center (AUC) handles the authentication process of a subscriber to the network. More specifically, the AUC holds the shared secret key Ki and generates the random challenge that is used to authenticate. The authentication process will be covered in more detail in chapter II-D.
- **Equipment Identity Register (EIR)** - The Equipment Identity Register (EIR) holds the International Mobile Equipment Identity (IMEI) numbers of banned or stolen phones. An IMEI number is a unique number assigned to every mobile phone. Dialing `*#06#` will show this number on the screen of the mobile phone. The EIR is not shown in figure 1 since it is not relevant to an IMSI catcher.

D. *GSM authentication sequence*

Authentication between the SIM and the GSM take place by a shared secret key; in other words using symmetric key cryptography. This shared secret key (Ki) is stored on the SIM card and on the Authentication Center. Figure 2 shows the GSM authentication sequence. The authentication goes as follows:

1) First the MS sends its security capabilities to the VLR.
2) The VLR sends the MS an Identity Request.
3) The MS replies with the IMSI.
4) Once the VLR received the IMSI, it will send an Authentication Vector Request to the AUC for this particular IMSI. The AUC will then do the following:
    a) It will generate a 128-bit random number (RAND)

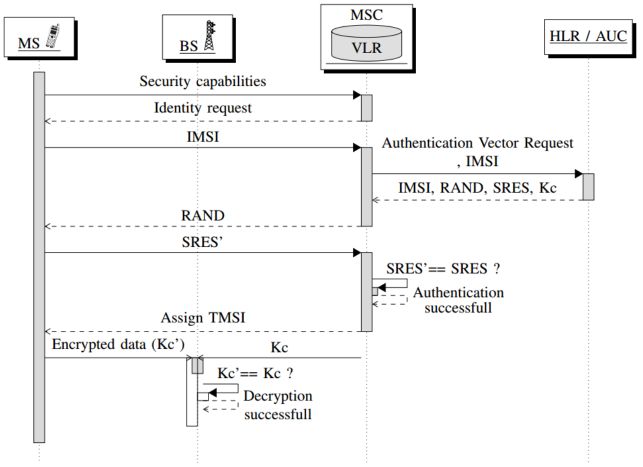


Figure 2: GSM authentication sequence

    b) Then it signs the RAND with the shared secret key (Ki) that belongs to this IMSI.
    c) The signed RAND is then used to create a 32 bit matching signed response (SRES) and a 64-bit session key (Kc).
    d) It sends the IMSI, RAND, SRES and Kc back to the VLR.
5) The VLR will then send the (unsigned) RAND to the MS.
6) The MS signs this RAND with the shared secret key (Ki). If the Ki is the same as the Ki which is stored on the AUC, then it should generate exactly the same SRES.
7) The MS sends back its version of the SRES (SRES').
8) The VLR then checks whether SRES' == SRES. If they are, then the authentication is successful.
9) The VLR assigns a TMSI to reduce the transmission of the IMSI over the air.
10) The VLR also sends the Kc (session key) to the BS.
11) Traffic that is being transmitted between the MS and the BS will now be encrypted using the Kc. As long as Kc' == Kc, the data can be decrypted by the BS.

E. *GSM Encryption*

GSM uses different algorithms to generate the different authentication and encryption functions. Since GSM has been around for more than a decade there have been some changes in the strength of some of the algorithms.

1) A5: A5 is the encryption standard that is used to encrypt the wireless transmissions between the Mobile station (MS) and the Base station (BS). A5/1 and A5/2 were developed in the 80s and their source code was kept secret. Finally, in 1999 these protocols were reverse engineered and published.

A5/0 Is referred when there is no encryption used.

A5/1 Made for the European and U.S. market and developed in 1987. Its workings were kept secret until it been reverse engineered in 1999 and published. This cypher nowadays is weak and can be practically cracked in a reasonable time using various attacks [4]. The most practical attack was demonstrated by Karsten Nohl, where A5/1 can be cracked within seconds using a rainbow table and the tool kraken. Despite this vulnerability, A5/1 is still used about 42% of the time in the Netherlands as of 2016 [23].

A5/2 This cypher was developed for export markets and turned out to be extremely weak. The cryptanalysis of Ian Goldberg and David Wagner in 1999 revealed that the cypher can be cracked in real-time [30]. A5/2 has been discontinued by the GSM association (GSMA) since 2006. This means that phones made since 2006 do not have A5/2 functionality. If a cell-site only supports A5/2 then that phone will fall back to A5/0 (no encryption) [1].

A5/3 A5/3 was actually designed for 3G networks but is also in use for GSM (2G) systems as an upgrade to the old A5/1 encryption. A5/3 was first based on the MISTY block cypher. Later the MISTY block cypher was simplified to make it more hardware friendly; this variant is called the KASUMI block cypher. The KASUMI cypher can be attacked faster than an exhaustive search by using a "sandwich" crypto attack. This is an adapted way of implementing a related-key boomerang attack and reduces the time complexity from 2^{76} to 2^{32}. This allows the KASUMI cypher to be cracked on a single duo-core PC within 2 hours. However, this is not an attack that would be practical in a real-life GSM implementation since it relies on chosen messages and both related keys [6]. But the point of the attack was to show that the modification of MISTY did have an impact on the strengths of the algorithm. It is unknown if anyone has successfully applied this attack in a real-life environment.

2) A3: The A3 standard is used to calculate the SRES from the Ki and RAND. A3 is based on the COMP128 algorithm and also had its weaknesses. Just like the A5 standard, A3 was initially kept secret. This algorithm was later reversed engineered in the 90s [10].

3) A8: A8 is used to calculate the session key (Kc) from the Ki and RAND and is also based on the COMP128 algorithm. Both A3 and A8 are executed on the AUC and the Mobile device.

F. IMSI catchers

In the thesis by Joseph Ooi, titled "IMSI Catchers and Mobile Security" much research was done on the workings of IMSI catchers and their deployment [19]. IMSI catchers are primarily deployed by law enforcement. The most widely known manufacturer is Harris Corporation that manufactured the StingRay devices. The FBI was willing to drop charges to avoid sharing information about the cell-site-simulators [21]. This can be due to the fact that Harris Corporation requires law enforcement to sign an NDA to prevent sharing details of the device [5].

Manufacturers other than Harris are [19]:

- Digital Receiver Technology, Inc. (DRT) who make the "DRTBOX".
- Meganet Corporation, who sell the VME Dominator which can monitor multiple calls and can call and send a text on behalf of the user.
- Gamma Group that also sells an IMSI catcher measuring only 41 x 33 x 18 cm which can be body-worn. These, however, do not intercept calls, but only capture IMSI numbers.
- Septier
- PKI, that has IMSI catchers that can jam 3G signals and force phones to use 2G.

1) Criminal use: In 2014, the FCC established a task force to investigate the extent of use of IMSI catchers by criminal gangs and foreign intelligence services [28]. Meanwhile, in 2012, there have been reports of the widespread use of IMSI catchers in the Czech Republic [29].

G. Workings of an IMSI catcher

First, an IMSI catcher has to find a GSM frequency that isn't in use yet so it will not interfere with other base stations. It can use a frequency reserved for testing or use the buffer channels that reside between the different operators.

Next, the 3G and 4G frequencies may be jammed so the MS will downgrade to 2G. The IMSI catcher can instruct the phone which crypto to use. So it can simply declare that it has no encryption capabilities and the connection will default to A5/0 (no encryption).

Once an IMSI catcher has caught a phone, it will keep the mobile phone "captive" by broadcasting the neighboring cell list as empty and therefore keeping the cell connected with the tower. The most straight-forward way to forward outgoing calls, text and data to the network is by using another SIM card or an outgoing VOIP trunk. The IMSI catcher can disable the caller ID presentation to mask the man in the middle.

If a phone is already connected to a legitimate tower it will not immediately connect to the fake base station, even if that signal strength is higher. To mitigate this, the IMSI catcher may change its LAC (Local area code) to one that is different from all the other base stations. Since the signal strength is higher, the phone established it moved to a different cell area and this will force the phone to do a location update [25].

1) Passive attacks: A passive attack device would only catch IMSI numbers. This is simple and involves performing the first steps of connecting to a base station.

1) First a fake base station needs to be set up.
2) The mobile device will connect with the fake base station.
3) The mobile device will then send its security capabilities (the IMSI catcher can ignore this).
4) The IMSI catchers replies with an ID request (SendIndentification) to the phone.
5) Finally, the phone will send back the IMSI in plain text.

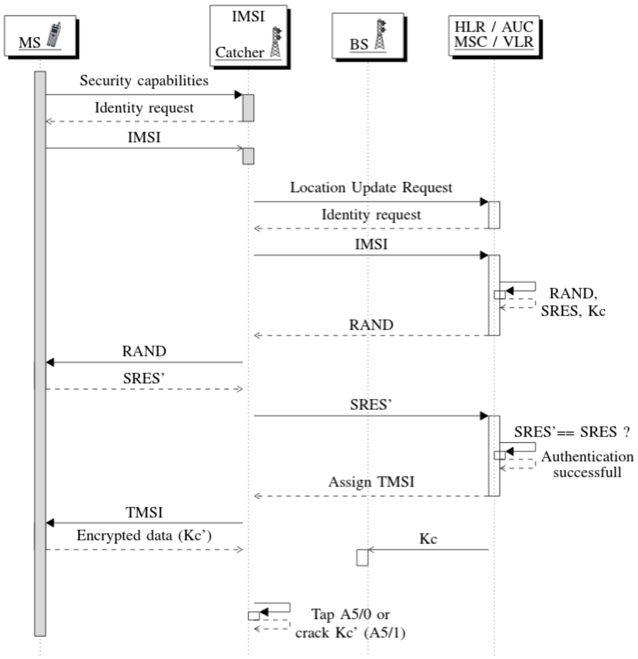


Figure 3: IMSI catcher

This IMSI can then be used by law-enforcement to determine the phone number and the owner of this IMSI.

2) Active attacks: A more active attack would be to make a full network connection with the cell tower so an MITM can be achieved. In figure 3 this MITM can be seen. What the fake base station will do is simply forward the authentication traffic to the real network and thus impersonating the victims' mobile phone. It also forwards the traffic from the network to the victims' phone.

The encryption used between the fake base station and the mobile station can be disabled (A5/0) or set to a weak cypher (A5/1). This allows for incoming and outgoing information to be intercepted and read.

3) Handover attack: This attack applies to cell sites that support both 3G and 2G. When a phone moves to a new cell-site it needs to be hand over to a different base station. If a phone moves from a 3G site (UMTS) to a 2G site (GSM) no new authentication to the network takes place. Instead, it will convert the session key of GSM (Kc) to a UMTS session key (CK) and Integrity key (IK). Vice versa also applies. This means that any pre-handover or post-handover UMTS communication can be broken if the handover involves a 2G site [15].

III. Methodology

A. Equipment

The SDR used to set up the fake base station is the Blade-RF x40. This can transceive on a frequency of 300MHz to 3.8GHz. It is a full-duplex transceiver, meaning it can receive and transmit at the same time. This is necessary for running a base station. The BladeRF uses USB3.0 for fast data transfer and in 2016 the BladeRF costs $420,- [18].

The HackRF One is used to analyze the radio signals generated by the Blade-RF. The HackRF is a half-duplex transceiver, runs with USB 2.0 and has a frequency range of 1MHz to 6GHz. In 2016 the HackRF One was about $299,- [7].

Finally, for testing the base station, two phones were used:

- HTC Desire HD. Released in 2010 and supports up to 3.5G (HSDPA). This phone was running Android 4.2.2 with the JellyTime R9 custom ROM. The "Network Cell Info Lite" app was used to analyze the cellular network it was connected to.
- Nokia 6021. This is a phone released in 2005 and supports up to 2.75G (EDGE). No custom software was installed on this phone.

For the experiment, a valid SIM card of the provider Tele2 was used. Tele2 makes use of the T-Mobile network. This SIM card was used in both phones.

The computer used to run the SDR equipment was an Asus K53S laptop. This is a USB 2.0 laptop with an i7 quad-core processor and 4GB RAM. The operating system installed was Ubuntu 14.04 LTS (Trusty Tahr).

B. Sniffing

The radio frequencies used for GSM may vary among different countries. On http://www.worldtimezone.com/gsm.html a list can be extracted for each country. The Cellular bands used in the Netherlands can be seen in table I.

Table I: GSM bands used in the Netherlands retrieved from WorldTimeZone.

2G	3G	4G LTE
900/1800	KPN 900/2100 T-Mobile 900/2100 Vodafone 2100	KPN 800(band 20) 1800(Band 3) 2600(Band 7) Vodafone 800(band 20) 1800(Band 3) 2600(Band 7) Tele2 800(band 20) 2600(Band 7) T-Mobile 900(band 8) 1800(Band 3) 2600(Band 7)

With the Hack-RF one can analyse these different GSM bands. Figure 4 shows a waterfall display of the 900MHz band where the different channels can be seen. The 900 MHz GSM band is about 100 MHz wide and each channel has a width of 200 kHz.

The 900 MHz frequency was in the past primarily used for 2G technologies, but recently providers are also allowed to offer 3G on these frequencies [31]. Due to its lower frequency, GSM 900 has a higher range than GSM 1800. The advantage of GSM 1800 is that it is faster but it is more difficult for these higher frequencies to penetrate buildings. Most modern phones are dual band, meaning the 900 and 1800 frequencies can be combined.

For the sniffing experiment, GSM 900 was used.

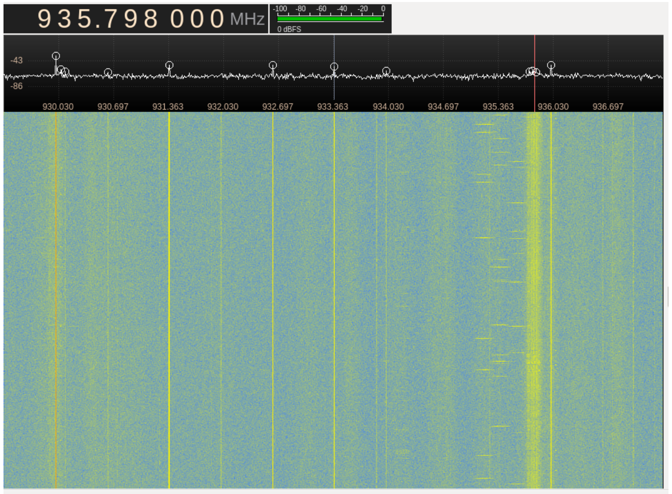


Figure 4: Waterfall display of GSM 900, captured with Gqrx.

C. Setting up the base station

The fake base station was set up with YateBTS, which is a fork project of the OpenBTS project. YateBTS can run either in "Network in a Box" mode or in roaming mode. In roaming mode, it can be interconnected with other YateBTS stations to form a cellular network. To get YateBTS to work with the Blade-RF, specific versions were needed. For the proof of concept, libbladeRF 1.6.1 had to be compiled on the host system. The firmware on the Blade-RF has to be flashed to version 1.6.1. A technical how-to on getting YateBTS to run with the Blade-RF can be found in Appendix A.

The base station tests for the proof of concept where done on the spectrum of GSM 900. In figure 5 the base station can be seen by the HackRF.

D. Jamming

A jammer was (attempted to) set up to test whether a phone would jump to the fake base station if 3G were to be jammed. In GNU radio Companion a noise generator was connected to a band pass filter and a transmission sink. All jamming tests were conducted inside a Faraday cage using the HackRF.

It was possible to interfere with the channel of the fake base station which was transmitting on GSM 900. Both phones lost the ongoing phone conversation with the YateBTS test number. The HTC phone conversation would initially drop, but when a new conversation was created it somehow overcame the interference created by the HackRF. Increasing the transmission gains or jamming the uplink channel had no effect on this. Also, when a higher frequency was used such as GSM 1800, it was not possible to interfere even when putting the phone 10 cm away from the HackRF. The jamming only worked consistently against the old Nokia phone on the 900 spectrum.

Jamming 3G was also attempted but not possible with the HackRF. The strongest 2100 MHz 3G signal the HackRF produced (inside the cage) was more weak than an actual 3G signal. On top of this, 3G UMTS uses the spread spectrum technique, which is a radio transmission technique designed to overcome interference and jamming.

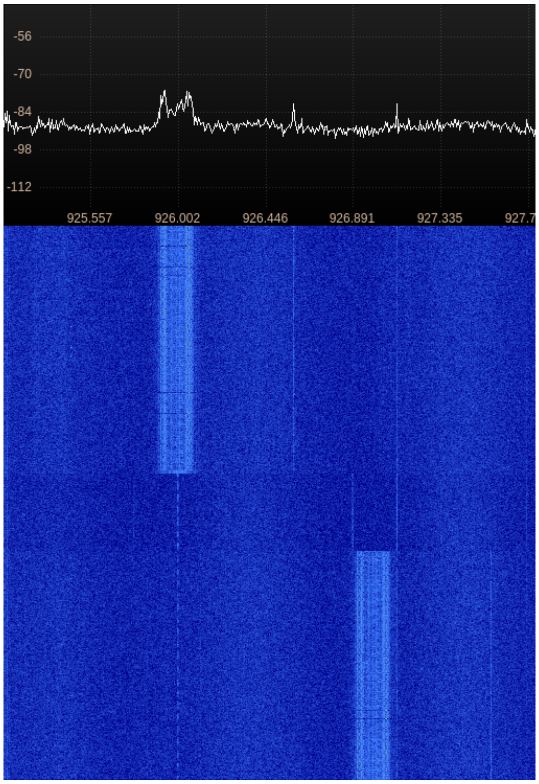


Figure 5: YateBTS switching between 926 and 927MHz seen in Gqrx.

IV. RESULTS

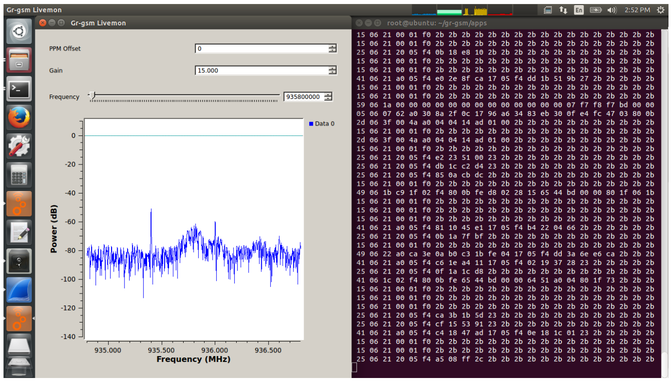
A. Decoding GSM

Captured GSM traffic first needs to be decoded before being fed to Wireshark. Note that "decoding" in this context means converting the radio signals back to data (no cryptography). This can be done using the "grgsm" tool. Grgsm makes use of the GNU Radio Companion (GRC) tool to process GSM signals. GNU Radio Companion is a popular tool for programming an SDR. Using a visual interface, different components (which are called blocks) can be tied together, much like a real hardware board. Grgsm offers a collection of GRC blocks and scripts that can be used to decode GSM signals.

Running `/gr-gsm/apps/grgsm_livemon.grc` opens an interface where the GSM frequency can be set. If the decoder receives valid GSM signals the received I/Q values can be seen in another terminal window. The fully decoded GSM network traffic is send over the `localhost` interface were the traffic can be analyzed. Figure 6 shows the process of decoding.

Wireshark can be used to inspect the decoded GSM traffic. As stated in the background, this traffic content is encrypted. Figure 7 shows the decoded GSM data that was captured using the Hack-RF.

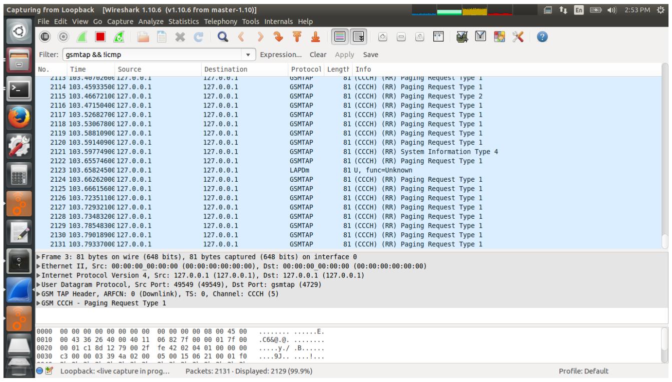
1) A5/1: Even though it is known for years that A5/1 is vulnerable, it turned out that A5/1 is actually still used about 44% of the time according to gsmmap [23]. Table II shows the percentage of implemented protection measures.



PPM Offset: 0
Gain: 15.000
Frequency: 935800000

```
15 06 21 00 01 f0 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b
15 06 21 00 01 f0 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b
25 06 21 20 05 f4 0b 18 e8 10 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b
15 06 21 00 01 f0 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b
15 06 21 00 01 f0 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b
41 06 21 a0 05 f4 e0 2e 8f ca 17 05 f4 dd 1b 51 9b 27 2b 2b 2b 2b 2b
15 06 21 00 01 f0 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b
15 06 21 00 01 f0 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b
59 06 1a 00 00 00 00 00 00 00 00 00 00 00 00 00 07 f7 f8 f7 bd 00 00
05 06 07 62 a0 30 8a 2f 0c 17 96 a6 34 83 eb 30 0f e4 fc 47 03 80 0b
2d 06 3f 00 4a a0 04 04 14 ad 01 00 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b
15 06 21 00 01 f0 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b
2d 06 3f 00 4a a0 04 04 14 ad 01 00 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b
15 06 21 00 01 f0 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b
25 06 21 20 05 f4 e2 23 51 00 23 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b
25 06 21 20 05 f4 db 1c c2 d4 23 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b
25 06 21 20 05 f4 85 0a cb dc 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b
15 06 21 00 01 f0 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b
49 06 1b c9 1f 02 f4 80 0b fe d8 02 28 15 65 44 bd 00 00 80 1f 06 1b
15 06 21 00 01 f0 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b
15 06 21 00 01 f0 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b
15 06 21 00 01 f0 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b
41 06 21 a0 05 f4 81 10 45 e1 17 05 f4 b4 22 04 66 2b 2b 2b 2b 2b 2b
25 06 21 20 05 f4 0b 1a 7f bf 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b
15 06 21 00 01 f0 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b
49 06 22 a0 ca 3e 0a b0 c3 1b fe 04 17 05 f4 dd 3a 6e e6 ca 2b 2b 2b
41 06 21 a0 05 f4 c6 1e a4 11 17 05 f4 02 19 37 28 23 2b 2b 2b 2b 2b
25 06 21 20 05 f4 0f 1a 1c d8 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b
41 06 1c 02 f4 80 0b fe 65 44 bd 00 00 64 51 a0 04 80 1f 73 2b 2b 2b
15 06 21 00 01 f0 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b
15 06 21 00 01 f0 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b
15 06 21 00 01 f0 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b
25 06 21 20 05 f4 ca 3b 1b 5d 23 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b
25 06 21 20 05 f4 cf 15 53 91 23 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b
41 06 21 a0 05 f4 c4 18 47 ad 17 05 f4 0e 18 1c 01 23 2b 2b 2b 2b 2b
15 06 21 00 01 f0 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b
15 06 21 00 01 f0 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b
25 06 21 20 05 f4 a5 08 ff 2c 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b 2b
```

Figure 6: Decoding GSM radio signals using grgsm.


Capturing from Loopback [Wireshark 1.10.6 (v1.10.6 from master-1.10)]
2:53 PM
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Expression... Clear Apply Save

Filter: gsmtap && !icmp

```
No.   Time        Source       Destination  Protocol Length Info
2113 103.4070260 127.0.0.1    127.0.0.1    GSMTAP    81 (CCCH) (RR) Paging Request Type 1
2114 103.4593350 127.0.0.1    127.0.0.1    GSMTAP    81 (CCCH) (RR) Paging Request Type 1
2115 103.4667210 127.0.0.1    127.0.0.1    GSMTAP    81 (CCCH) (RR) Paging Request Type 2
2116 103.4715040 127.0.0.1    127.0.0.1    GSMTAP    81 (CCCH) (RR) Paging Request Type 1
2117 103.5268270 127.0.0.1    127.0.0.1    GSMTAP    81 (CCCH) (RR) Paging Request Type 1
2118 103.5306780 127.0.0.1    127.0.0.1    GSMTAP    81 (CCCH) (RR) Paging Request Type 1
2119 103.5881090 127.0.0.1    127.0.0.1    GSMTAP    81 (CCCH) (RR) Paging Request Type 1
2120 103.5914090 127.0.0.1    127.0.0.1    GSMTAP    81 (CCCH) (RR) Paging Request Type 1
2121 103.5977490 127.0.0.1    127.0.0.1    GSMTAP    81 (CCCH) (RR) System Information Type 4
2122 103.6557460 127.0.0.1    127.0.0.1    GSMTAP    81 (CCCH) (RR) Paging Request Type 1
2123 103.6582450 127.0.0.1    127.0.0.1    LAPDm     81 U, func=Unknown
2124 103.6626200 127.0.0.1    127.0.0.1    GSMTAP    81 (CCCH) (RR) Paging Request Type 1
2125 103.6661560 127.0.0.1    127.0.0.1    GSMTAP    81 (CCCH) (RR) Paging Request Type 1
2126 103.7235110 127.0.0.1    127.0.0.1    GSMTAP    81 (CCCH) (RR) Paging Request Type 1
2127 103.7293210 127.0.0.1    127.0.0.1    GSMTAP    81 (CCCH) (RR) Paging Request Type 1
2128 103.7348320 127.0.0.1    127.0.0.1    GSMTAP    81 (CCCH) (RR) Paging Request Type 1
2129 103.7854830 127.0.0.1    127.0.0.1    GSMTAP    81 (CCCH) (RR) Paging Request Type 1
2130 103.7901890 127.0.0.1    127.0.0.1    GSMTAP    81 (CCCH) (RR) Paging Request Type 1
2131 103.7933700 127.0.0.1    127.0.0.1    GSMTAP    81 (CCCH) (RR) Paging Request Type 1

▶Frame 3: 81 bytes on wire (648 bits), 81 bytes captured (648 bits) on interface 0
▶Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)
▶Internet Protocol Version 4, Src: 127.0.0.1 (127.0.0.1), Dst: 127.0.0.1 (127.0.0.1)
▶User Datagram Protocol, Src Port: 49549 (49549), Dst Port: gsmtap (4729)
▶GSM TAP Header, ARFCN: 0 (Downlink), TS: 0, Channel: CCCH (5)
▶GSM CCCH - Paging Request Type 1

0000  00 00 00 00 00 00 00 00  00 00 00 00 08 00 45 00   ........ ......E.
0010  00 43 36 26 40 00 40 11  06 82 7f 00 00 01 7f 00   .C6&@.@. ........
0020  00 01 c1 8d 12 79 00 2f  fe 42 02 04 01 00 00 00   .....y./ .B......
0030  c3 00 00 03 39 4a 02 00  05 00 15 06 21 00 01 f0   ....9J.. ....!...
```

Loopback: <live capture in prog... Packets: 2131 · Displayed: 2129 (99.9%) Profile: Default

Figure 7: GSM network traffic viewed in wireshark.

A5/3 is slowly being implemented and for most providers it has become the standard, but there are many instances where the vulnerable A5/1 cypher is still being used.

As described in the background section, A5/1 can be practically cracked in seconds using a rainbow table. The cracking tool used for this is kraken. Examples on how to use this tool can be found on the internet [13] [3] [2]. Unfortunately due to time constrains this could not be tested.

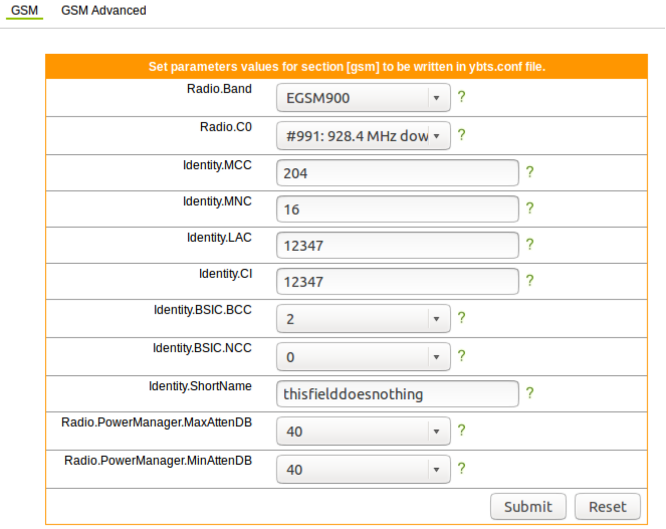
Table II: 2G over-the-air protection, retrieved from the GSM Map Project [23].

	KPN	T-Mobile	Vodafone
A5/0	2%	0%	0%
A5/1	45%	**59%**	23%
A5/3	**53%**	41%	**77%**

B. Man in the Middle

1) Base station configuration: Once YateBTS was set up, the spoofing experiment could begin. Figure 8 shows the configuration used in the Proof of Concept. The goal here was to see if a phone can be spoofed into connecting with the fake base station.

- First an empty channel was chosen. In this case channel 991 which broadcasts at 928.4 MHz. This is to prevent collisions and make the phone believe that it has moved to a neighboring antenna.

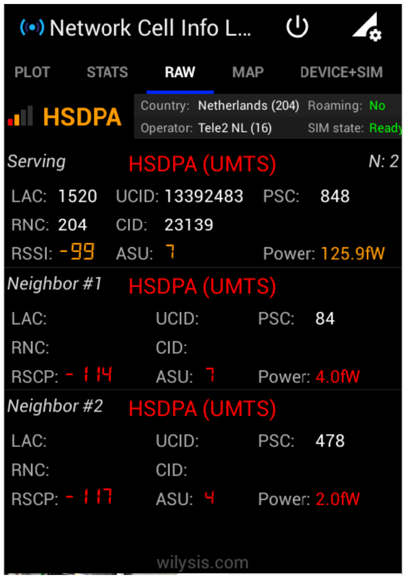


Set parameters values for section [gsm] to be written in ybts.conf file.

Radio.Band	EGSM900 ?
Radio.C0	#991: 928.4 MHz dow ?
Identity.MCC	204 ?
Identity.MNC	16 ?
Identity.LAC	12347 ?
Identity.CI	12347 ?
Identity.BSIC.BCC	2 ?
Identity.BSIC.NCC	0 ?
Identity.ShortName	thisfielddoesnothing ?
Radio.PowerManager.MaxAttenDB	40 ?
Radio.PowerManager.MinAttenDB	40 ?

Figure 8: YateBTS settings viewed in the web config.

- To spoof an existing cellular network one has to alter the Mobile County Code (MCC) and the Mobile Network Code (MNC). On http://www.mcc-mnc.com/ a list can be found of all the codes for every country and its networks. Since Tele2 makes use of the T-Mobile network, the settings became MCC: 204 and MNC: 16 (which is the MNC of T-mobile).
- The Local Area Code (LAC) and Cell ID (CI) are what identifies an individual base station in a network. Any value for these will suffice. An attempt has been made to make these identical to the original base station by extracting it from the phone or using websites such as OpenCellID and Antenneregister to find out the real LAC and CI of existing networks. However, this had no effect on the spoofing.
  Changing the LAC frequently should make the mobile believe it has moved to a different location. The frequency of changing the LAC and the CI had no effect on the spoofing.
- The Base Station Identity Code (BSIC) is a combination of the 3-bit Network Color Code (NCC) and the 3-bit Base station Color Code (BCC). The NCC bits are implemented to make sure mobile stations that sit on country borders do not conflict with each-other if they happen to have conflicting parameters. For the Netherlands the NCC bits are 0 and 4. Setting the NCC to 0 or 4 and/or setting the BCC to 2 or 3 had no effect on the effectiveness of the spoof.
- The shortname can only be set in the commercial version of YateBTS. This functionality has been removed from the source code but is still available in OpenBTS. However, setting a shortname is not necessary for spoofing, as will be demonstrated further on.
- The power output level for the ouput power control loop is set to 40 dB. This is just enough to provide a stable connection up to 2 meters.



HSDPA
Country: Netherlands (204) Roaming: No
Operator: Tele2 NL (16) SIM state: Ready

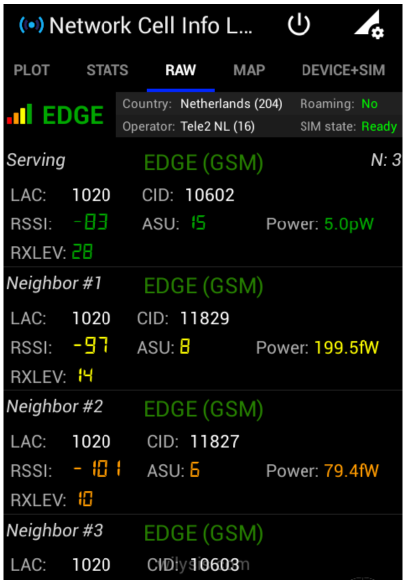
Serving HSDPA (UMTS) N: 2
LAC: 1520 UCID: 13392483 PSC: 848
RNC: 204 CID: 23139
RSSI: -99 ASU: 7 Power: 125.9fW

Neighbor #1 HSDPA (UMTS)
LAC: UCID: PSC: 84
RNC: CID:
RSCP: -114 ASU: 7 Power: 4.0fW

Neighbor #2 HSDPA (UMTS)
LAC: UCID: PSC: 478
RNC: CID:
RSCP: -117 ASU: 4 Power: 2.0fW

wilysis.com

Figure 9: Phone remains connected to the weaker 3G network. Viewed in the Network Cell Info app.



EDGE
Country: Netherlands (204) Roaming: No
Operator: Tele2 NL (16) SIM state: Ready

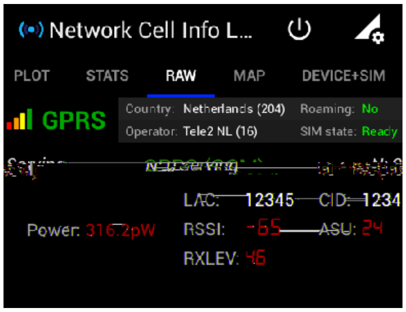
Serving EDGE (GSM) N: 3
LAC: 1020 CID: 10602
RSSI: -83 ASU: 15 Power: 5.0pW
RXLEV: 28

Neighbor #1 EDGE (GSM)
LAC: 1020 CID: 11829
RSSI: -97 ASU: 8 Power: 199.5fW
RXLEV: 14

Neighbor #2 EDGE (GSM)
LAC: 1020 CID: 11827
RSSI: -101 ASU: 6 Power: 79.4fW
RXLEV: 10

Neighbor #3 EDGE (GSM)
LAC: 1020 CID: 10603

Figure 10: Phone connected to EDGE (2.75G) instead of the fake base station.



GPRS
Country: Netherlands (204) Roaming: No
Operator: Tele2 NL (16) SIM state: Ready

Serving
LAC: 12345 CID: 1234
Power: 316.2pW RSSI: -65 ASU: 24
RXLEV: 46

Figure 11: Phone connected to the fake base station.

2) Spoofing results: Figure 9 shows a screenshot of the "Network Cell Info Lite" app. It can be seen that the phone is connected to an HSDPA (3.5G) network. This is the same base-station that the phone was connected to **prior** and **after** the fake base station was enabled after 4 hours. So even though there is a base station that has a way stronger signal, the phone will remain connected to this 3.5G network.

The only chance of making the phone connect to the fake base station is to switch the phone to only use 2G networks. However, whether the phone connects to the fake base station is still depended on the location of the phone. Figure 10 shows the EDGE network the phone would connect to, even though the base station sat right next to the phone. The phone always prefers the faster base station if within a certain signal threshold. Since YateBTS is essentially a 2.5G base station (GPRS) the phone prefers to connect to an EDGE (2.75G) base station, even though the 2.5G station has a much stronger signal. Rebooting the phone has no effect.

In the System and Network engineering lab the phone will connect to the fake base station if the weather is poor. This is also the case in the auditorium. If the weather is good, or the test is done on a different location in Amsterdam, the phone again connects to an EDGE base station. Figure 11 shows the network information of the fake base station when it is able to connect. It can be seen that the operator name is `Tele2 NL (16)` and that there are no neighbors being advertised.

If the phone is connected to the fake base station and the base station goes down, the phone will automatically switch to the real network. When the fake base station goes up again it will remain connected to the real network.

The spoofing has also been tested on the Nokia 6021 phone, which does not support 3G. The phone did not connect with the fake base station if it was already connected to a real network. When analyzing the radio traffic with the Hack-RF, it turned out the phone was connected to a GSM 1800 tower. Setting the fake base station to a GSM 1800 channel did not make the phone connect either.

What did work however, was having the fake base station running **before** the phone was booted. In this case, the phone connects to the base station automatically after the phone is booted. Figure 12 shows the display when connected to the fake base station. It can be seen that the network name displayed on the screen is identical to the real network (Tele2 NL).

3) Jamming: Jamming works with a HackRF, but only against 2G on a 900MHz band. In the experiment with the fake base station, the phone was switched to 2G mode. This could simulate a situation where 3G were to be jammed. Unfortunately, no real evidence can be generated since jamming 3G was not possible with the HackRF.

C. *Interception possibilities*

1) GPRS: General Packet Radio Service (GPRS) is the packet transfer standard that goes with YateBTS. It has to be noted that GPRS is one of the slowest mobile internet technologies and is unusable for media such as browsing and

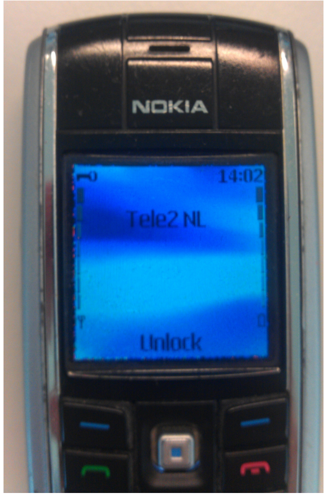
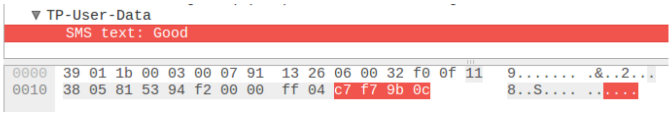


Figure 12: Tele2 NL spoofed network display viewed from a Nokia 6021.



```
▼ TP-User-Data
    SMS text: Good

0000  39 01 1b 00 03 00 07 91  13 26 06 00 32 f0 0f 11   9....... .&..2...
0010  38 05 81 53 94 f2 00 00  ff 04 c7 f7 9b 0c          8..S.... ......
```

Figure 13: Plaintext SMS viewed in Wireshark.

streaming. However, the speed of GPRS is enough to process email fetches, Whatsapp messages, Telegram messages and queries made by apps.

YateBTS creates a NAT interface called `sgnstun`. The technical details for setting up GPRS with YateBTS can be found in Appendix B. All the mobile data will flow through this `sgnstun` NAT interface. One can analyse all the traffic by letting Wireshark listen on this interface.

2) SMS: SMS can be easily intercepted (figure 13). In YateBTS, traffic tapping can be enabled which let GSM traffic be dumped to the `localhost` interface (port 4729). The same traffic can be observed as the decoded GSM traffic captured with the HackRF in chapter IV. The plain text SMS can be uncovered by using Wireshark in the following way:

1) Run `tshark -i lo -f "port 4729" -w /tmp/sms2.cap` to capture traffic coming in at port 4729. This is necessary to prevent ICMP packets to corrupt the GSM data. This bug should be fixed in the latest version of Wireshark [22].
2) Filter at Protocol and look for "GSM SMS"
   - > GSM SMS TPDU (GSM 03.40) SMS-DELIVER
     - > TP-User-Data
       - > SMS text: `plaintext`

3) Other features: With YateBTS it is possible to forward outgoing phone calls over a SIP or AIX trunk. This means calls can be transported over VOIP.

This voice traffic can then either be sniffed from the GSMTAP interface (`localhost` port 4729) or from the WAN interface were the VOIP traffic is being transported.

V. Discussion

Spoofing 2G turns out to be ineffective so is it possible to set up a 3G base station instead? There is a UMTS version of OpenBTS called OpenBTS-UMTS. This is unfortunately not compatible with the BladeRF. It is compatible with some Ettus Research USRP devices, which is a slightly more expensive alternative of the BladeRF. But setting up a UMTS base station will not guarantee that all phones will connect. UMTS is a 3.0G technology and during the experiment, the phone was at first connected to HSDPA which is a 3.5G technology. When having a 3G fake base station set up, it is likely that the phone will (again) prefer the faster base station. A 3G-only base station will also not be compatible with pre-3G devices like the Nokia 6021.

In the spoof experiment with the Nokia phone, the phone remained connected with the current base station unless it reboots. It is hard to prove why this is happening. Since the Nokia 6210 supports 2.75G (EDGE), it could have preferred the faster base station. Perhaps when the device boots, the neighbor list is empty and the phone will simply connect to the strongest base station that is advertising the right MNC and MCC codes. But this remains speculation.

VI. Conclusion

The general conclusion is that an IMSI catcher isn't as effective in capturing traffic as thought. While this attack worked well 15 years ago, with the introduction of 3G and 4G it is not effective anymore since phone prefer to use these faster technologies.

In order for an IMSI catcher to work, it must be able to let phones connect to the fake base station. The problem is that modern phones use 3G and 4G and prefer to use these faster technologies. Even when forcing phones to only use 2G, it was observed that catching a phone will be tricky at best and relies on how strong the nearest 2.75G (EDGE) base station is among other factors. If 3G and 4G were to be disabled or jammed then a 2G IMSI catcher can have merit. Unlawful people might find this useful, but for a commercial party like Deloitte getting permission to jam GSM signals would be a tough legal issue [16]. In any case, if a phone does connect with the IMSI catcher, all internet traffic and outgoing SMS can be intercepted.

A. *Future work*

The following subjects can be looked into for further research:

- Create a proof of concept for cracking sniffed A5/1 traffic. Also automate this process. The advantage of this method is that it does not need setting up a fake base station. But since probably all smartphones make use of 3G or 4G, not much traffic may be extracted (unless combined with some downgrade attack).
- Since phones have the preference to connect to faster base stations, further research can be done on other techniques that force phones to either downgrade or connect to a fake base station. Perhaps certain messages can spoof the phone or force it to drop to 3G. Another way that might work is by jamming all

2G channels except the channel that is used by the fake base station. So one will have to selectively jam parts of the GSM spectrum. Naturally, there are some ethical issues involved with this kind of research, since even being in possession of a jamming device is illegal in the Netherlands [16].

- For this research only a half IMSI catcher was build. Perhaps for future research, a full IMSI Catcher can be build. However, just as the half IMSI catchers, the full IMSI catcher also has to be able to successfully spoof a base station.

APPENDIX A
GETTING YATEBTS TO WORK WITH THE BLADERF X40

1)
```
sudo apt-get install git telnet
apache2 php5 libusb-1.0-0
libusb-1.0-0-dbg libusb-1.0-0-dev
libgsm1 libgsm1-dev cmake automake
build-essential libncurses5-dev
libtecla1 libtecla-dev pkg-config
```
2) Connect the BladeRF and verify with `dmesg`
3) Download and compile libbladeRF v1.6.1: https://github.com/Nuand/bladeRF/releases/tag/libbladeRF_v1.6.1
```
sudo -i
cd bladeRF-libbladeRF_v1.6.1
cd host
mkdir build
cd build
cmake -DCMAKE_BUILD_TYPE=Release
-DCMAKE_INSTALL_PREFIX=/usr/local
-DINSTALL_UDEV_RULES=ON ../
make -j4
make install > install.log
ldconfig
```
4) Firmware 1.6.1-git-053fb13-buildomatic needs to be flashed to the BladeRF. This version can be downloaded from NUAND. The firmware can be flashed by running:
```
bladeRF-cli -f *****.img -v verbose
```
The FPGA can remain unloaded. YateBTS ships with bladeRF FPGA's and loads the right one automatically.
5) A specific YateBTS version is needed. The evilsocked code is used for this [14].
```
sudo -i
git clone
https://github.com/evilsocket/evilbts.git
cd evilbts
cd yate
./autogen.sh
./configure - -prefix=/usr/local
make -j4
sudo make install
sudo ldconfig
cd ..

cd yatebts
./autogen.sh
./configure - -prefix=/usr/local
make -j4
sudo make install
sudo ldconfig
```
6) Minimal configuration:
```
cd /var/www/html/
```
`sudo ln -s /usr/local/share/yate/nib_web nib` links nib to apache folder.
`sudo chmod -R a+w /usr/local/etc/yate` allows the web GUI to write the config.

7) Follow the tutorial at [26] for configuring yate (skip GPRS) and use the following settings:
    - In `sudo nano /usr/local/etc/yate/subscribers.conf` : Country code 31 (NL)
    - Regexp: `.*`
    - 50dB
    - EGSM900 - channel 75
    - `Radio.PowerManager.MaxAttenDB + Radio.PowerManager.MinAttenDB`: 50dB (lower value means stronger signal). 50dB is effective within 1-2 meters.
8) Run `sudo yate -s -vv` to start the base station.
9) Then manually look for the AP with your phone. It should be called "TestSIM" depending on the MCC.

Sources: [9] [26] [14]

Appendix B
Getting GPRS to work in YateBTS

1) First tail the GGSN log:
```
tail -f /usr/local/lib/yate/service/bts/ggsn.log
```
2) Make sure Voice echo and the SMS bot works:
    - In `/usr/local/etc/yate/javascript.conf`:

      ```
      route=welcome.js
      ```
    - Then call `32843` and sms to `35492` (sms bot).
3) Make sure NAT works:
```
sudo iptables -t nat -A POSTROUTING
-o wlan0 -j MASQUERADE
sudo iptables -list -t nat -v
```

    In `/etc/sysctl.conf` uncomment: #net.ipv4.ip_forward=1

    Test NAT by pinging `8.8.8.8` from the phone. Once NAT works:
```
sudo apt-get install
iptables-persistent
```
    (or run `dpkg-reconfigure iptables-persistent` if ran before)
4) Configure DNS in NIB. `8.8.8.8` usually works. Test DNS by performing `ping` and `nslookup` on the phone. If DNS does not work use the ip adress of the DNS server in the network.
5) The GPRS tap does not have to be enabled in NIB. This is because all traffic is already flowing trough `sgstun`. Simply let Wireshark listen on this interface to inspect the traffic.

References

[1] 3GPP. Prohibiting a5/2 in mobile stations and other clarifications regarding a5 algorithm support. http://www.3gpp.org/ftp/tsg_sa/TSG_SA/TSGS_37/Docs/SP-070671.zip. Accessed: 2016-06-20.

[2] Unknown author. Gsm cracking a5 encryption and sms sniffing with rtl-sdr. https://www.youtube.com/watch?v=TOl4Q4lyJTI, 2015.

[3] Unknown author. Gsm hacking: Topguw. https://n0where.net/gsm-hacking-topguw/, 2015.

[4] Alex Biryukov, Adi Shamir, and David Wagner. Real time cryptanalysis of a5/1 on a pc. In *International Workshop on Fast Software Encryption*, pages 1–18. Springer, 2000.

[5] Ars Technica Cyrus Farivar. Non disclosure agreement. https://www.documentcloud.org/documents/1727748-non-disclosure-agreement.html#document/p3/a212440. Accessed: 2016-06-20.

[6] Orr Dunkelman, Nathan Keller, and Adi Shamir. A practical-time attack on the a5/3 cryptosystem used in third generation gsm telephony. *IACR Cryptology ePrint Archive*, 2010:13, 2010.

[7] Great Scott Gadgets. Where to buy hardware from great scott gadgets. https://greatscottgadgets.com/wheretobuy/, 2016.

[8] Ryan Gallagher. Meet the machines that steal your phone's data. http://arstechnica.com/tech-policy/2013/09/meet-the-machines-that-steal-your-phones-data/. Accessed: 2016-06-2.

[9] Nuand github. Getting started: Linux. https://github.com/Nuand/bladeRF/wiki/Getting-Started%3A-Linux#Debianbased_distros_eg_Ubuntu, 2016.

[10] hackingprojects.net. Secrets of the sim. http://www.hackingprojects.net/2013/04/secrets-of-sim.html. Accessed: 2016-06-20.

[11] Wireless Intelligence. Subscriber statistics end q1 2007. http://web.archive.org/web/20070927162249/http://www.gsmworld.com/news/statistics/pdf/gsma_stats_q1_07.pdf. Accessed: 2016-06-20.

[12] ITU-R. Requirements related to technical performance for imt-advanced radio interface(s). http://www.itu.int/dms_pub/itu-r/opb/rep/R-REP-M.2134-2008-PDF-E.pdf. Accessed: 2016-06-20.

[13] Sylvain Munaut Karsten Nohl. Wideband gsm sniffing [27c3]. https://www.youtube.com/watch?v=ZrbatnnRxFc. Accessed: 2016-05-30.

[14] Simone Margaritelli. How to build your own rogue gsm bts for fun and profit. https://evilsocket.net/2016/03/31/how-to-build-your-own-rogue-gsm-bts-for-fun-and-profit/, 2016.

[15] Ulrike Meyer and Susanne Wetzel. On the impact of gsm encryption and man-in-the-middle attacks on the security of interoperating gsm/umts networks. In *Personal, Indoor and Mobile Radio Communications, 2004. PIMRC 2004. 15th IEEE International Symposium on*, volume 4, pages 2876–2883. IEEE, 2004.

[16] Openbaar Ministerie. Richtlijn voor strafvordering telecommunicatiewet (2016r004). https://www.om.nl/organisatie/beleidsregels/overzicht-0/specialistisch/@94065/richtlijn-9/#hoofdstuk_417460, 2016.

[17] ngmn. Ngmn 5g white paper. https://www.ngmn.org/uploads/media/NGMN_5G_White_Paper_V1_0.pdf. Accessed: 2016-06-20.

[18] Nuand. bladerf x40 | nuand -. https://www.nuand.com/blog/product/bladerf-x40/, 2016.

[19] Joseph Ooi. Imsi catchers and mobile security, 2015.

[20] Chris Paget. Def con 18 - chris paget - practical cellphone spying. https://www.youtube.com/watch?v=fQSu9cBaojc. Accessed: 2016-05-30.

[21] Jose Pagliery. Fbi lets suspects go to protect 'stingray' secrets. http://money.cnn.com/2015/03/18/technology/security/police-stingray-phone-tracker/, 2015.

[22] Pascal Quantin. Change 15281. https://code.wireshark.org/review/#/c/15281, 2016.

[23] Berlin Security Research Labs. Mobile network security report: Netherlands. Accessed: 2016-07-1.

[24] Yubo Song, Kan Zhou, and Xi Chen. Fake bts attacks of gsm system on software radio platform. *Journal of Networks*, 7(2):275–281, 2012.

[25] SRLabs. Imsi catcher score. https://opensource.srlabs.de/projects/snoopsnitch/wiki/IMSI_Catcher_Score. Accessed: 2016-06-20.

[26] strcpy. Building a portable gsm bts using the nuand bladerf, raspberry pi and yatebts (the definitive and step by step guide). https://blog.strcpy.info/2016/04/21/building-a-portable-gsm-bts-using-bladerf-raspberry-and-yatebts-the-definitive-guide/, 2016.

[27] Daehyun Strobel. Imsi catcher. *Chair for Communication Security, Ruhr-Universität Bochum*, page 14, 2007.

[28] Craig Timberg. Feds to study illegal use of spy gear. https://www.washingtonpost.com/blogs/the-switch/wp/2014/08/11/feds-to-study-illegal-use-of-spy-gear, 2014.

[29] Masha Volynsky. Spy games turn real as eavesdropping technology spreads. http://www.radio.cz/en/section/curraffrs/spy-games-turn-real-as-eavesdropping-technology-spreads, 2012.

[30] D Wagner et al. The real-time cryptanalysis of a5/2//crypto'99 (santa barbara, august 15-19, 1999): Proc. http://www.cs.berkeley.edu/~daw/tmp/a52-slides.ps, 1999.

[31] Arnoud Wokke. Providers verbeteren bereik 3g-netwerken door gebruik lagere frequentie. https://tweakers.net/nieuws/90002/providers-verbeteren-bereik-3g-netwerken-door-gebruik-lagere-frequentie.html, 2013.