GSM für die Lehre – Basisstation, IMSI-Catcher und Monitordevices aus Standardkomponenten selbst gebaut

Dirk von Suchodoletz, Dennis Wehrle, Holger Bertsch

dirk.von.suchodoletz@rz.uni-freiburg.de
dennis.wehrle@rz.uni-freiburg.de
holger.bertsch@gmx.de

Lehrstuhl für Kommunikationssysteme
Rechenzentrum der Universität Freiburg
Hermann-Herder-Str. 10
79104 Freiburg

Abstract: Für Demonstrationszwecke in Vorlesungen und für Sicherheitsuntersuchungen ist der Aufbau einer prototypischen GSM Base Tranceiver Station von Interesse. Ähnlich wie für Lehrveranstaltungen zu Netzwerken, die sinnvollerweise vielseitige praktische Demonstrationen bieten, sollte dieses auch für den Bereich Mobilfunk gelten. Durch den Betrieb einer eigenen BTS können viele Abläufe auf verschiedenen Layern analysiert und nachvollziehbar gemacht werden. Darüber hinaus lassen sich bestehende Sicherheitslücken gut mit Hilfe eines IMSI-Catchers illustrieren. Diesbezüglich wird gezeigt, wie Mobilfunkteilnehmer sich ohne ihr Wissen in den IMSI-Catcher einbuchen und überwacht werden können, ohne dass ihnen ihr Mobiltelefon das mitteilt. Die Kontrolle, der sie hierbei unterliegen, beinhaltet, abgesehen vom Auslesen der IMSI und IMEI, auch eine Auflistung aller aktuell geführten Gespräche und die Möglichkeit diese aufzuzeichnen.

1 Einleitung

Die Black-Box-Ära im Mobilfunk neigt sich ihrem Ende entgegen. Neue Hardwareentwicklungswerkzeuge und Open-Source-Mobilfunklösungen öffnen das Feld für den informierten Jedermann. Mobilfunknetze haben die heutige Lebenswelt komplett durchdrungen. Die mobile Telekommunikationslandschaft hat sich in den letzten 20 Jahren signifikant demokratisiert. Verfügte früher eine sehr überschaubare Elite aus Politik und Wirtschaft über die Technik in gewissem Rahmen mobil zu telefonieren, so ist die Zahl der registrierten SIMs in Deutschland höher als die der Einwohner. Weltweit nutzen mehr als zwei Milliarden Menschen GSM. Bisher fanden Sicherheitsdiskussionen und mögliche Angriffsszenarien auf diese Infrastruktur nur in kleinen Fachzirkeln statt. Die neuen Möglichkeiten werden diesen Zustand in den nächsten Jahren sicherlich verändern und neue Sicherheitsdiskussionen hervorrufen [PN09].
Dieses schafft zudem ganz neue Grundlagen, um aus rein theoretischen Vorlesungen zum

Thema eine deutlich interaktivere Veranstaltung mit praktischen Demonstrationselementen zum Nachbauen und Analysieren zu machen. Damit lassen sich aktuelle GSM-Infrastrukturen ähnlich gut präsentieren, wie diverse Internet-Protokolle.

2 Die selbstgebaute GSM-Zelle

Zum Aufbau eines kleinen GSM-Mobilfunknetzes muss nicht komplett die mehrtausendseitige Spezifikation umgesetzt werden. Es genügen die zentralen Komponenten des Radio und des Network Subsystems, um mit herkömmlichen Mobiltelefonen bereits Gespräche führen zu können oder SMS zu empfangen. Die Base Transceiver Station einer kleinen Zelle mit bis zu sieben Teilnehmern lässt sich mittels eines Universal Software Radio Peripheral (USRP) an einem Steuercomputer betreiben. Eine Übersicht und einen Vergleich zwischen der GSM-Infrastruktur und dem Setup mittels USRP gibt die Abbildung 1.

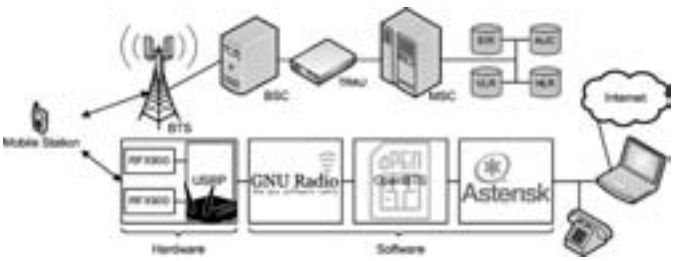

Abbildung 1: Systemüberblick der benötigten Hard- und Softwarekomponenten für den Aufbau einer kleinen GSM-Zelle.

Das von der Firma Ettus [Ett09] hergestellte Universal Software Radio Peripheral (USRP) erlaubt es, die gesamte Signalverarbeitung mittels Software zu realisieren. Im Gegensatz zu einer für ein bestimmtes Einsatzgebiet konstruierten Spezialhardware können mit dem Software-defined Radio unterschiedliche Modulationsarten, Multiplex- und Medienzugriffsverfahren für den Frequenzbereich von 1 MHz bis zu 5,9 GHz umgesetzt werden. [Ett] Das gewährleistet eine größtmögliche Flexibilität, die es erlaubt, eine Reihe verschiedener Anwendungen wie RFID-Lesegeräte, WLAN-Empfänger, FM- Radiostationen oder GSM-Netzwerkkomponenten für das Radio Subsystem aufzubauen. Es existieren zwei Varianten am Markt, wobei das ältere USRP1 vom OpenBTS-Projekt unterstützt wird und das neue sich für Frequenzbandanalysen eignet.
Das Time Divison Multiplexing von GSM ist zeitkritisch. Ausführliche Experimente zeigten, dass der eingebaute Zeitgeber des USRP für einen langzeit-stabilen Betrieb nicht ausreicht. Deshalb wurde das USRP so modifiziert, dass sich verschiedene externe Zeitgeber anschließen lassen. Eine Variante besteht im Einsatz des Bausatzes FA-SY1, der von Funkamateuren genutzt wird. [7309] Der Taktgeber lässt sich über USB auf eine Frequenz zwischen 10 bis 160 MHz einstellen und weist anfänglich eine Abweichung von $\pm$ 20 ppm auf, welche durch eine Kalibrierung allerdings verringert werden kann. Für eine tempe-

raturunabhängige Frequenzstabilität besitzt der Taktgeber einen Heiztransistor, der über einen Temperaturfühler geregelt wird.
Da das USRP lediglich das High-Level-Sampling übernimmt, muss ein Linux-PC mit GNU Radio, die Signalverarbeitung der niedrigen Abtastraten übernehmen. Dabei stellt es lediglich eine allgemeine Schnittstelle zum USRP bereit, die aus Bibliotheken zur Signalverarbeitung und einem USB-Kernelmodul für die Ansteuerung der Hardware besteht. Erst mit Hilfe des OpenBTS Projekts wird daraus eine GSM-Basisstation. OpenBTS bildet hierzu das Mindestmaß einer GSM-Infrastruktur nach, wozu es bespielsweise über eine Art integrierte Mini-VLR verfügt, in dem die TMSI's verwaltet werden. Ebenfalls benötigt OpenBTS einen Asterisk-Server, um eine ganze Reihe von Aufgaben, wie die Identifikation und Authentifizierung (HLR) der Teilnehmer sowie das Führen von Telefongesprächen innerhalb von OpenBTS in das allgemeine Telefonnetz (TRAU) zu realisieren.
Gestartet werden kann OpenBTS mit dem Befehl `./OpenBTS`. Wichtig hierbei ist, dass die Konfigurationsdatei *OpenBTS.config* zuvor bearbeitet und entsprechende Parameter eingestellt wurden. Die wohl wichtigsten Konfigurationsparameter sind im „GSM" Abschnitt der Datei *OpenBTS.config* zu finden. Durch die Variable *GSM.Band 900* kann zwischen GSM 900 und 1800 gewechselt werden. Mittels *GSM.ARFCN 29* wird die entsprechende Frequenz eingestellt, ARFCAN 29 entspricht dabei 940.8 MHz. Durch die Variablen *GSM.MCC 922*, *GSM.LAC 667*, *GSM.CI 10* sowie *GSM.ShortName „OpenBTS"* wird eine GSM-Zelle mit dem Namen OpenBTS, dem Ländercode 922, dem Location Area Code 667 und der CellID 10 gestartet. Ältere Mobiltelefone zeigen als Netzname in der Netzliste „922 55" oder „Nor 55" an. Die 55 entspricht dem festgelegten Mobile Network Code (*GSM.MNC 55*). Diese Einstellungen lassen sich verwenden, um „Original"-Zellen zu simulieren.

3 GSM-Monitoring mit Wireshark und USRP oder Mobiltelefon

Wie für das Verständnis von TCP/IP auch, ist es hilfreich die verschiedenen Netzwerkprotokolle auf unterschiedlichen Layern der Protokoll-Stacks analysieren zu können. Das fängt auf der physikalischen Schicht mit der Ermittlung von Funkzellen an und setzt sich durch höhere Schichten und die Interpretation der Kanäle bis hin zu den Rahmenstrukturen fort. Auf diese Weise lässt sich beispielsweise die Frequenzverteilung in einem bestimmten Gebiet sichtbarmachen, zeigen wie das Einbuchen eines Mobiltelefons ins GSM-Netz erfolgt oder wie während geführter Telefonate die Qualität der Verbindung überwacht und bei Bedarf ein Handover eingeleitet wird. Diese Untersuchungen lassen sich sowohl im echten Mobilfunknetz als auch mit der selbstgebauten Basisstation vornehmen.

Nokia Netzmonitor Einige ältere Nokia-Mobiltelefone verfügen über einen Netzwerkmonitor, der über das gängige Menü normalerweise nicht zugänglich ist und erst mittels spezieller Software freigeschaltet werden muss.[1] Mittels dieses Monitors lassen sich Parameter wie Kanalzuteilung (CH), Leistungsregelung, Cell-ID (CID), Informationen über

[1] Die Vorgehensweise unterscheidet sich von Gerät zu Gerät. Eine ausführliche Anleitung für verschiedene Nokia Modelle findet sich auf der Homepage nokiaport.de.

Nachbarzellen (Display 03; erste Spalte ist der Kanal, dritte Spalte die Empfangsstärke) und Handover ermitteln. Vier dieser Netzmonitor-Displays eines Nokia 3310 sind in Abbildung 2 dargestellt.

Abbildung 2: Der Netzmonitor im Menü eines Nokia 3310 mit Informationen zu den einzelnen empfangenen Basisstationen.

USRP und GNU Radio-Spektrumsanalyse GNU Radio enhält eine Vielzahl an Beispielprogrammen, wie ein Softwareoszilloskop oder einen Spektrumsanalysator. Letzteres ist ein Python-Skript (*usrp_fft.py*), das sich für Untersuchungen und Experimente der GSM-Frequenzbänder nutzen lässt. Mit Hilfe dieses Analysators lassen sich Base Transciever Stations aufspüren. Mit folgendem Befehl kann ein Scanvorgang begonnen werden: `usrp_fft.py -R A -d 8 -g 47 -f 928M`. Dieser Befehl sorgt dafür, dass um die Frequenz von 928 MHz ($\pm$ 4 MHz) nach BTS gescannt wird. Das Ergebnis und die Einstellungen, wie Average, können der Abbildung 3 entnommen werden.

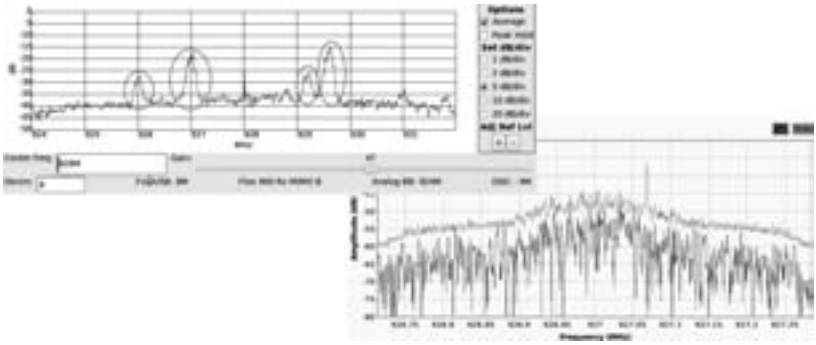

Abbildung 3: Spektrumsanalyse im Bereich von 924 - 932 MHz (oben) bzw. genauere Betrachtung (unten) eines BTS-Kanals bei 927 MHz (Decim=112) mittels GNU Radio-Skript: *usrp_fft.py*

Es ist deutlich zu erkennen, dass es mehrere Ausschläge im Bereich um 928 MHz gibt (rot markiert). Hierbei handelt es sich um 200 kHz breite Kanäle. Für eine genauere Betrachtung wurde mit einer feineren Auflösung der Frequenz (=Decim) der größte Ausschlag (BTS mit bester Sende- bzw. Empfangsstärke) bei 927.0 MHz gewählt. Dies erfolgt mit dem Befehl: `usrp_fft.py -R A -d 112 -g 32 -f 927M`. Außerdem wurde in Abbildung 3 die Option Peek Hold verwendet. Diese sorgt dafür, dass der größte Ausschlag gespeichert wird (grüne Linie). Genau +67,7 kHz von der Kanalmitte entfernt ist deutlich ein Ausschlag (Peek) zu erkennen. Hierbei handelt es sich um ein FCCH-Paket,

das periodisch alle zehn Pakete im Zeitschlitz 0 übertragen wird. Es dient der Frequenzkorrektur und zur Auswahl der Zelle des BTS mit der besten Empfangsfrequenz.

AirProbe und GSSM mit USRP Bei der Softwaresammlung AirProbe handelt es sich um ein GSM-Sniffer-Projekt des Chaos Computer Clubs. [Clu09] Ziel dieses Projekts ist es, ein Analyse-Tool zu schaffen, das in der Lage ist, GSM-Daten auf dem Air-Interface zu analysieren. Der dritte und vielleicht wichtigste Aspekt des Projekts ist es, die Sicherheitslücken des GSM-Standards zu demonstrieren. AirProbe gliedert sich in drei Hauptteile: Erfassung von Daten, Demodulation und Analyse.
Für sämtliche Tests kam die Mitte 2009 aktuelle AirProbe-Version aus dem Git-Repository des Chaos Computer Clubs zum Einsatz, die durch einen weiterentwickelten GSM-Receiver gepatcht wurde.[2] Der Ordner *gsmdecode/src/python* des AirProbe Verzeichnisses enthält zwei Skripte: *capture.sh* und *go.sh*. Ersteres dient dazu, mittels USRP und GNU Radio Daten aufzuzeichnen, das zweite, diese zu dekodieren. Vorher ist eine aktive Frequenz zu ermitteln, auf der eine Funkzelle sendet. So liess sich durch mehrere Tests auf verschiedenen Frequenzen mittels *go.sh* eine E-Plus Funkzelle auf 927.0 MHz identifizieren. Diverse Parameter wie Mobile Country und Network Code, Ordinary Subscribers sowie Emergency Call ließen sich darüber hinaus sichtbar machen.

GSSM Das Softwarepaket GSSM kann GSM Base Station Control Channels überwachen. Die Analyse der gesammelten Pakete erfolgt in einem gepatchten Wireshark mittels eines virtuellen TUN-Interfaces. GNU Radio übernimmt dabei die Demodulation und Dekodierung der einzelnen GSM-Pakete. Folgende Kontrollkanäle (zwischen BTS und MS) können mittels GSSM v.0.1.1.1a decodiert werden: FCCH, SCH, BCCH, PCH (nur Downlink), AGCH (nur Downlink), SACCH, SDCCH
Um die live mitgeschnittenen Daten sichtbar zu machen, muss Wireshark gestartet und das erstellte GSM-Interface zur Überwachung ausgewählt werden. Einen Ausschnitt der ermittelten GSM-Pakete in Wireshark wird in Abbildung 4 dargestellt.
Bisher ist GSSM lediglich in der Lage, GSM-Pakete von der BTS zur MS zu überwachen, aber nicht umgekehrt. Die Um-Schnittstelle wurde nur teilweise implementiert und viele Pakete kann Wireshark nicht korrekt interpretieren, da sie über eine abweichende Protokoll-Beschreibung verfügen. Die Identifizierung kann beispielsweise durch einen unterschiedlichen Frame-Type fehlschlagen.

GSM Dekodierung durch Nokia 3310 und Wireshark Das Mobiltelefon Nokia 3310 ist in der Lage, GSM-Nachrichten aus einem Gammu Trace Log zu dekodieren. Es ist möglich, Signalisierungsprozesse auf Layer 2 (LAPDm) in Sende- und Empfangsrichtung sichtbar zu machen. Diese Tatsache beruht darauf, dass die Entwickler eine Loggingfunktion eingebaut hatten. Die generierten XML-Dateien können mit Wireshark geöffnet und analysiert werden. Alternativ kann zur Analyse der aufgezeichneten Dateien auch das Programm *Gsmdecode* des Chaos Computer Clubs verwendet werden. Es ist genau wie Wire-

[2] Genauere Informationen auf der Webseite: AirProbe Git-Repository, `git://svn.berlin.ccc.de/`. Patch von Piot Krysik unter `http://home.elka.pw.edu.pl/_pkrysik/GSM`.

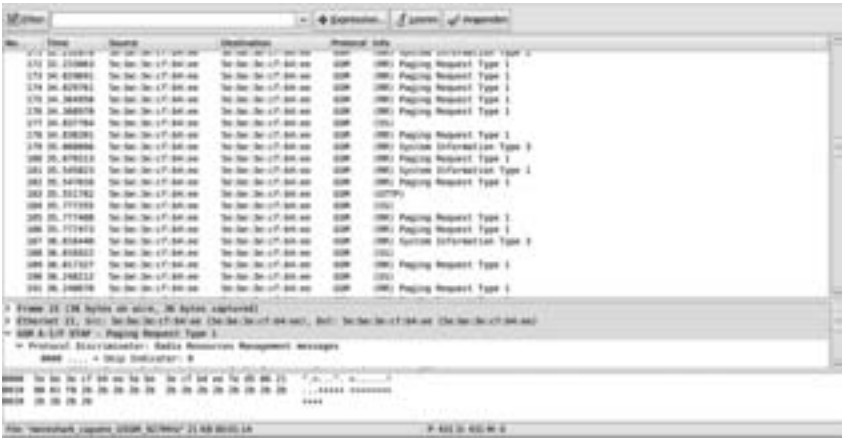

Abbildung 4: Ausschnitt eines GSM-Paketes in Wireshark

shark in der Lage, GSM-Nachrichten zu dekodieren.[3] Hardwareseitig wird ein Nokia 3310 Mobiltelefon und ein spezielles MBUS NK-33 Datenkabel benötigt.[4] Eine genaue Installationsanleitung für die Software kann dem AirProbe Wiki des CCC entnommen werden.[5] Es werden die Pakete *gammu*,[6] *Wireshark* (Version 1.2.1) und *dialog* benötigt.

Abbildung 5 zeigt einen Gesprächsaufbau des Nokia 3310 über das T-Mobile Netz im Wireshark. Die entsprechenden Pakete ließen sich parallel mittels Gammu-Software aufzeichnen und das Log-File danach mit Wireshark betrachten. Darüber hinaus wurde eine SMS-Nachricht mittels Nokia E71 an das Nokia 3310 gesendet und mittels Gammu-Tracelog mitgeschnitten. Der Nachrichteninhalt der SMS konnte auf Grund der eingeschalteten Verschlüsselung nicht eingesehen werden. Bei einem Gesprächsaufbau eines iPhone 2Gs zu einem Nokia 3310 über OpenBTS wird im Vergleich deutlich, dass in OpenBTS keine Verschlüsselung verwendet wird. Dasselbe gilt für SMS. Dazu wurde mittels der OpenBTS Konsole an das Nokia 3310 eine SMS-Nachricht geschickt, um die dabei ablaufenden Signalisierungsprozesse aufzuzeichnen. Der Inhalt dieser Nachricht „Das ist ein Test“ konnte in Wireshark unverschlüsselt mitgelesen werden.

4 Der IMSI-Catcher aus dem Elektronikmarkt

Der erste IMSI-Catcher mit dem Namen GA090 wurde von der deutschen Firma Rohde & Schwarz 1996 in München vorgestellt. Er wurde ursprünglich als Test- und Messsystem

[3] `https://svn.berlin.ccc.de/projects/airprobe/attachment/wiki/tracelog/gsmdecode-0.7bis.tar.gz`

[4] N-33 Nokia Cable 3310, 3330, 3390 with MBUS Interface (compatible), `http://ucables.com/ref/NK-33`

[5] Tracelog - AirProbe, `https://svn.berlin.ccc.de/projects/airprobe/wiki/tracelog`

[6] `http://www.gammu.org/wiki/index.php?title=Download` - Version: 1.26.1

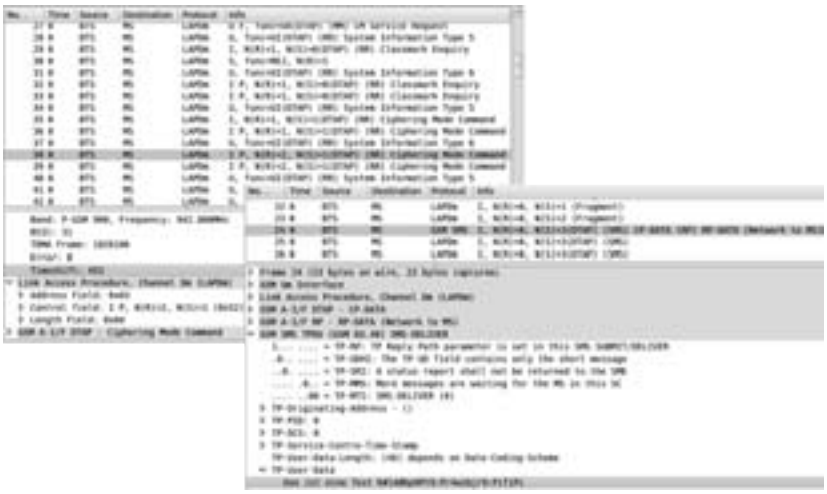

Abbildung 5: Wireshark-Mitschnitt: Gesprächsaufbau mit Verschlüsselungskommando im T-Mobile Netz (oben) und Empfang einer SMS von OpenBTS an Nokia 3310 (unten)

konstruiert und später zur Bestimmung der Endgerätekennung von Mobile Stations weiterentwickelt. [Han]
Auf Basis der USRP-Hardware und OpenBTS konnte ein eigener IMSI-Catcher entwickelt werden, der sowohl die IMSI als auch die IMEI auslesen, sowie alle aktuellen Gespräche mit Zielrufnummer anzeigen und aufzeichnen kann. [Weh09] Die Funktionsweise des Open Source IMSI-Catchers (Abbildung 6) unterscheidet sich vom Standard-IMSI-Catcher dahingehend, dass für die Weiterleitung der Daten keine an den IMSI-Catcher angeschlossene MS verwendet werden kann. Diese Beschränkung ergibt sich auf Grund der verwendeten Hard- und Softwarekomponenten. Daher ist der Aufbau des IMSI-Catchers und die Authentifizierung gegenüber dem Standard IMSI-Catcher leicht verändert. Der unverschlüsselte Datenverkehr wird nicht mehr über eine MS, sondern über den Computer und darin installiertem Asterisk-Server weitergeleitet. Dieser Server ermöglicht es, Festnetz- und Mobilfunkgespräche zu führen. Wie beim Standard IMSI-Catcher wird jedoch nicht die Identität des Teilnehmers vorgetäsucht. Aus diesem Grund muss die Rufnummerübermittlung deaktiviert werden. Der angerufene Teilnehmer bekommt somit einen Anruf von einem „unbekannten Teilnehmer“. Damit die MS die vom IMSI-Catcher simulierte Zelle, nicht von einem realen Netz unterscheiden kann, muss der IMSI-Catcher ein entsprechendes Netz des gewünschten Anbieters vortäuschen. Hierfür werden diverse Konfigurationseinstellungen in der *OpenBTS.conf* vorgenommen. Wichtig für das Simulieren sind lediglich der richtige Country Code (*MCC*), der vorzutäuschende Netzanbieter Code (*MNC*), die Frequenz sowie der Name der Funkzelle (*Shortname*), der identisch dem Netzanbieter sein muss. Allerdings darf die Frequenz nicht die selbe sein. Die notwendigen Informationen über aktuelle Basisstationen beschafft man mit Hilfe der bereits gezeigten Frequenzanalyse. Alle anderen Parameter sind irrelevant, da diese Informationen der MS lediglich dazu dienen, ihren Standort (LAC und CID) zu bestimmen.

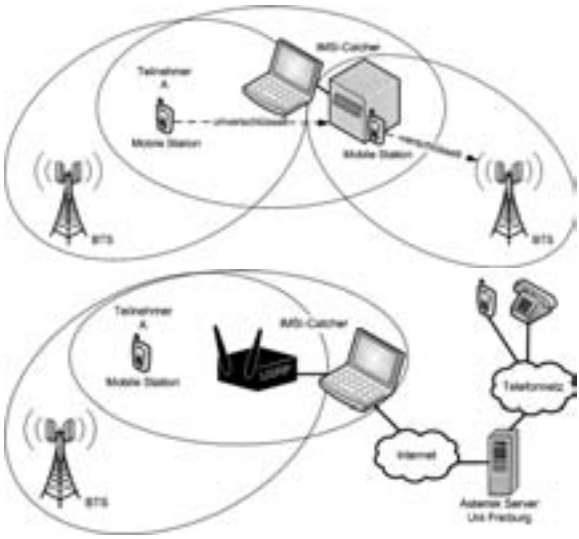

Abbildung 6: Funktionsweise eines Standard IMSI-Catcher (oben) und Open Source IMSI-Catcher mit USRP und Asterisk-Server (unten)

Insgesamt existieren drei Möglichkeiten, wie sich eine MS in ein vorgetäuschtes Netz des IMSI-Catchers einbuchen kann. Wie sich durch diverse Versuche herausgestellt hat, ist es dabei unerheblich, ob der Teilnehmer eine manuelle oder automatische Netzauswahl eingestellt hat. Da die MS das Netz des IMSI-Catchers nicht von einem Originalnetz unterscheiden kann, ist dieses Verhalten leicht zu erklären.

1. Fall - Zielnetz nicht vorhanden
   Die MS besitzt keine Konnektivität und befindet sich im Suchmodus (Normal Cell Selection), wodurch nacheinander verschiedene Frequenzen nach einer aktiven BTS gescannt werden. In diesem Fall muss der IMSI-Catcher lediglich eine beliebige Frequenz, den Ländercode 262 als auch den benötigten Netzanbietercode und Shortname einstellen. Sobald die MS diese Frequenz scannt, versucht sie sich einzubuchen.

2. Fall - Zielnetz vorhanden
   Schwieriger ist der Fall eines vorhandenen aktiven Netzes, in dem die MS eingebucht ist. Hier existieren zwei Möglichkeiten, eine MS dazu zu bringen, sich beim IMSI-Catcher anzumelden. Eine Variante besteht darin, einen manuellen Zellwechsel anzustoßen. Eine weitere Variante, die Frequenz zu stören, auf der die MS im Netz des Anbieters eingebucht ist, damit die MS sich in den IMSI-Catcher einbucht.

   (a) **Erzwungener Zellwechsel:** Die Grundidee bei diesem Szenario ist, auf Grund der Nachbarschaftsliste einen Zellwechsel der MS in den IMSI-Catcher zu erzwingen.

(b) **Jammer:** Ein GSM-Jammer ist ein Störsender mit dem Ziel, die Frequenz, in der die MS eingebucht ist, zu verrauschen. Mit Hilfe des Jammers verliert die MS die Verbindung zur aktuellen BTS. Sie wechselt in den Frequenzsuchmodus mit dem Ziel, dass der IMSI-Catcher anschließend als Netz ausgewählt wird.

Gegenwärtig bietet GSM keinen ausreichenden Schutz vor einem IMSI-Catcher. Die minimale Schutzfunktion, unverschlüsselte Verbindungen anzuzeigen, wird typischerweise durch die Provider auf der SIM abgeschaltet. Selbst das aktuellere UMTS bietet nicht den notwendigen Schutz, da mit einem Jammer die Frequenzen von Basisstationen gestört werden können. Dem Mobiltelefon wird mit Hilfe einer „fallback"-Funktion die Nutzung des GSM-Netzes ermöglicht, falls kein UMTS-Netz verfügbar ist. Somit muss das Mobiltelefon lediglich dazu gebracht werden GSM zu nutzen. Wird allerdings ausschließlich das UMTS-Netz verwendet, muss ein anderer IMSI-Catcher entwickelt werden, der sich an dem Standard IMSI-Catcher orientiert und Daten an das Originalnetz weiterleitet. Die Funktionsweise eines UMTS-IMSI-Catchers (Abbildung 7) ist relativ ähnlich und läuft in drei Phasen ab: [MW04]

1. Die IMSI bzw. TMSI der Mobile Station muss beim initialen Registrierungsprozess gespeichert werden.

2. Der IMSI-Catcher überträgt die gespeicherte IMSI zum Originalnetz und bekommt die Zufallszahl RAND und das Authentication Token (AUTN) als Antwort. Der IMSI-Catcher trennt dann die Verbindung und speichert RAND und AUTN.

3. Der IMSI-Catcher überträgt anschließend die RAND und das AUTN an die Mobile Station, die die Korrektheit von AUTN anerkennt, da das Token aktuell ist. Die Mobile Station bucht sich anschließend in den IMSI-Catcher ein, der wiederum die Verschlüsselung ausschaltet.

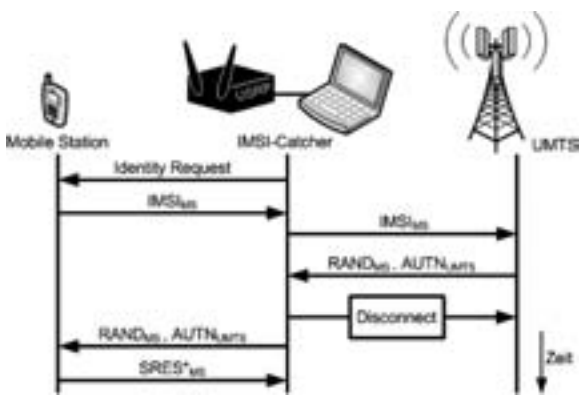

Abbildung 7: Funktionsweise eines UMTS-IMSI-Catchers

5 Fazit

Der Sicherheitsstandard von GSM lässt sich mit dem des Internets von vor 15 Jahren vergleichen. Insofern ist sicherlich in der nächsten Zeit mit weiteren Angriffen zu rechnen, beispielsweise mit einer Denial-of-Server, die auf dem 26C3 gezeigt wurde ([Spa09], [PN09]). Zunehmender SMS-SPAM oder der Versuch der Config-over-the-Air könnten weitere Problemvektoren darstellen: Gelingt es, verschiedene Teilnehmer in ein durch OpenBTS simuliertes Netz zu locken, können diese mit beliebig vielen SMS-Nachrichten überflutet werden. Anders als beim weit verbreiteten E-Mail Spam, der vorhandene Empfängeradressen voraussetzt, benötigt man keine gültigen Mobilfunknummern. Die MS müssen sich lediglich im Sende- bzw. Empfangsradius einer OpenBTS-Zelle befinden und sich einbuchen. Weitaus gefährlicher könnte es werden, Mobiltelefone per SMS zu manipulieren. Hierzu könnte beispielsweise der Austausch des WAP-Gateways, Internet oder SMS-Gateways oder auch die Manipulation der Mobiltelefon Firmware gehören.
Ein wesentlicher Nachteil gegenüber den Untersuchungen der Internet-Protokolle bleibt bestehen: GSM lässt sich nur mit einer Testlizenz bei Tageslicht benutzen. Andernfalls bleibt nur der abgeschirmte Tiefkeller.

Literatur

[7309] Box 73. *Box 73 Amateurfunkservice GmbH*. WWW-Dokument, `http://www.box73.de/catalog/`, 09 2009.

[Clu09] Chaos Computer Club. *airprobe*. WWW-Dokument, `https://svn.berlin.ccc.de/projects/airprobe/`, 09 2009.

[Ett] Ettus. *Brochure for the entire USRP product family*. PDF-Dokument, `http://www.ettus.com/downloads/er_broch_trifold_v5b.pdf`.

[Ett09] Ettus. *Ettus Research LLC*. WWW-Dokument, `http://www.ettus.com/`, 10 2009.

[Han] Uni Hannover. *IMSI-Catcher - Wanzen fuer Handys*. WWW-Dokument, `http://www.iwi.uni-hannover.de/lv/ucc_ws04_05/riemer/literatur/imsi-catcher.htm`.

[MW04] Ulrike Meyer and Susanne Wetzel. A man-in-the-middle attack on UMTS. In *WiSe '04: Proceedings of the 3rd ACM workshop on Wireless security*, pages 90–97, New York, NY, USA, 2004. ACM.

[PN09] Chris Paget and Karsten Nohl. *GSM: SRSLY?* WWW-Dokument, `http://events.ccc.de/congress/2009/Fahrplan/attachments/1519_26C3.Karsten.Nohl.GSM.pdf`, 12 2009.

[Spa09] Dieter Spaar. *Playing with the GSM RF Interface*. PDF-Dokument, `http://events.ccc.de/congress/2009/Fahrplan/attachments/1507_Playing_with_the_GSM_RF_Interface.pdf`, 2009.

[Weh09] Dennis Wehrle. *Open Source IMSI-Catcher*. PDF-Dokument, `http://www.ks.uni-freiburg.de/php_arbeitdet.php?id=166`, 10 2009.