osmocom.org - FOSS for mobile comms

Free Software Tools for GSM Security Research

Harald Welte <laforge@gnumonks.org>

osmocom.org
sysmocom GmbH

July 20, HITCON 2013 / Taipei

Outline

1. Introduction
2. Researching communications systems
3. Bootstrapping Osmocom
4. The Osmocom project

About the speaker

- Linux Kernel / bootloader / driver / firmware development since 1999
- IT security expert, focus on network protocol security
- Former core developer of Linux packet filter netfilter/iptables
- Board-level Electrical Engineering
- Always looking for interesting protocols (RFID, DECT, GSM)
- OpenEXZ, OpenPCD, Openmoko, OpenBSC, OsmocomBB, OsmoSGSN

Legal Disclaimer

- GSM operates in licensed spectrum
- Operating any transmitter in the GSM frequency bands requires a license from the respective regulatory authority
- Interference with commercial cellular operators is often a fellony and punishable as a crime
- It is the users responsibility to configure OpenBSC and BTS equipment in a way that complies with the law

Telco vs. Internet-driven IT security

mobile industry today has security practieses and procedures of the 20th century

- no proper incident response on RAN/CN
- no procedures for quick roll-out of new sw releases
- no requirements for software-upgradeability
- no interaction with hacker community
- no packet filtering / DPI / IDS on signalling traffic
- active hostility towards operators who want to do pentesting
- attempts to use legal means to stop researchers from publishing their findings

this sounds like medieval times. We are in 2012 ?!?

Real-world quotes

The following slides indicate some quotes that I have heard over the last couple of years from my contacts inside the mobile industry. They are not made up!

Quote: Disclosure of Ki/K/OPC

"we are sending our IMSI+Key lists as CSV files to the SIM card supplier in China"

Quote: RRLP

"RRLP? What is that? We never heard about it!"

Quote: SIM OTA keys

"we have no clue what remote accessible (OTA) features our sim cards have or what kind of keys were used during provisioning"

Quote: Malformed

"we have never tried to intentionally send any malformed message to any of our equipment"

Quote: Roaming

"We are seeing TCAP/MAP related attacks/fraud from Operator XYZ in Pakistan. However, it is more important that European travellers can roam into their network than it is for Pakistanis to roam into our network. Can you see while the roaming agreement was only suspended for two days?"

Quote: SIGTRAN IPsec

"we are unable to mandate from our roaming partners that SIGTRAN links shall always go through IPsec - we don't even know how to facilitate safe distribution of certificates between operators"

Quote: NodeB / IPsec

"We mandated IPsec to be used for all of the (e)NodeB back-haul in our tender, the supplier still shipped equipment that didn't comply to it. Do you think the CEO is going to cancel the contract with them for that?"

Quote: Government / independent study

"Govt: We put out a tender for a study on overal operator network security in our country. Everyone who put in a bid is economically affiliated or dependent on one of the operators or equipment suppliers, so we knew the results were not worth much."

Quote: Technical Staff

"15 years ago we still had staff that understood all those details. But today, you know, those experts are expensive - we laid them off."

Quote: Baseband chip vendor

"We have no clue what version of our protocol stack with what modifications are shipped in which particular phones, or if/when the phone makers distribute updates to the actual phone population"

Research in TCP/IP/Ethernet

Assume you want to do some research in the TCP/IP/Ethernet communications area,

- you use off-the-shelf hardware (x86, Ethernet card)
- you start with the Linux / *BSD stack
- you add the instrumentation you need
- you make your proposed modifications
- you do some testing
- you write your paper and publish the results

Research in (mobile) communications

Assume it is before 2009 (before Osmocom) and you want to do some research in mobile comms

- there is no FOSS implementation of any of the protocols or functional entities
- almost no university has a test lab with the required equipment. And if they do, it is black boxes that you cannot modify according to your research requirements
- you turn away at that point, or you cannot work on really exciting stuff
- only chance is to partner with commercial company, who puts you under NDAs and who wants to profit from your research

GSM/3G vs. Internet

- Observation
  - Both GSM/3G and TCP/IP protocol specs are publicly available
  - The Internet protocol stack (Ethernet/Wifi/TCP/IP) receives lots of scrutiny
  - GSM networks are as widely deployed as the Internet
  - Yet, GSM/3G protocols receive no such scrutiny!
- There are reasons for that:
  - GSM industry is extremely closed (and closed-minded)
  - Only about 4 closed-source protocol stack implementations
  - GSM chipset makers never release any hardware documentation

The closed GSM industry

Handset manufacturing side

- Only very few companies build GSM/3.5G baseband chips today
  - Those companies buy the operating system kernel and the protocol stack from third parties
- Only very few handset makers are large enough to become a customer
  - Even they only get limited access to hardware documentation
  - Even they never really get access to the firmware source

The closed GSM industry

Network manufacturing side

- Only very few companies build GSM network equipment
  - Basically only Ericsson, Nokia-Siemens, Alcatel-Lucent and Huawei
  - Exception: Small equipment manufacturers for picocell / nanocell / femtocells / measurement devices and law enforcement equipment
- Only operators buy equipment from them
- Since the quantities are low, the prices are extremely high
  - e.g. for a BTS, easily 10-40k EUR

The closed GSM industry

Operator side

- Operators are mainly banks today
- Typical operator outsources
  - Network planning / deployment / servicing
  - Even Billing!
- Operator just knows the closed equipment as shipped by manufacturer
- Very few people at an operator have knowledge of the protocol beyond what's needed for operations and maintenance

GSM is more than phone calls

Listening to phone calls is boring...

- Machine-to-Machine (M2M) communication
  - BMW can unlock/open your car via GSM
  - Alarm systems often report via GSM
  - Smart Metering (Utility companies)
  - GSM-R / European Train Control System
  - Vending machines report that their cash box is full
  - Control if wind-mills supply power into the grid
  - Transaction numbers for electronic banking

The closed GSM industry

Security implications

The security implications of the closed GSM industry are:

- Almost no people who have detailed technical knowledge outside the protocol stack or GSM network equipment manufacturers
- No independent research on protocol-level security
  - If there's security research at all, then only theoretical (like the A5/2 and A5/1 cryptanalysis)
  - Or on application level (e.g. mobile malware)
- No open source protocol implementations
  - which are key for making more people learn about the protocols
  - which enable quick prototyping/testing by modifying existing code

The closed GSM industry

My self-proclaimed mission

Mission: Bring TCP/IP/Internet security knowledge to GSM

- Create tools to enable independent/public IT Security community to examine GSM
- Try to close the estimated 10 year gap between the state of security technology on the Internet vs. GSM networks
  - Industry thinks in terms of *walled garden* and *phones behaving like specified*
  - No proper incident response strategies!
  - No packet filters, firewalls, intrusion detection on GSM protocol level
  - General public assumes GSM networks are safer than Internet

To actually do research on GSM, we need

- detailed knowledge on the architecture and protocol stack
- suitable hardware (there's no PHY/MAC only device like Ethernet MAC)
- a Free / Open Source Software implementation of at least parts of the protocol stack

Bootstrapping GSM Research

How would you get started?

If you were to start with GSM protocol level security analysis, where and how would you start?

- On the handset side?
    - Difficult since GSM firmware and protocol stacks are closed and proprietary
    - Even if you want to write your own protocol stack, the layer 1 hardware and signal processing is closed and undocumented, too
    - Publicly known attempts
        - The TSM30 project as part of the THC GSM project
        - mados, an alternative OS for Nokia DTC3 phones
    - none of those projects successful so far

Bootstrapping GSM research

How would you get started?

If you were to start with GSM protocol level security analysis, where and how would you start?

- On the network side?
  - Difficult since equipment is not easily available and normally extremely expensive
  - However, network is very modular and has many standardized/documented interfaces
  - Thus, if BTS equipment is available, much easier/faster progress

Bootstrapping GSM research

The bootstrapping process

- Read GSM specs (> 1000 PDF documents, each hundreds of pages)
- Gradually grow knowledge about the protocols
- Obtain actual GSM network equipment (BTS)
- Try to get actual protocol traces as examples
- Start a complete protocol stack implementation from scratch
- Finally, go and play with GSM protocol security

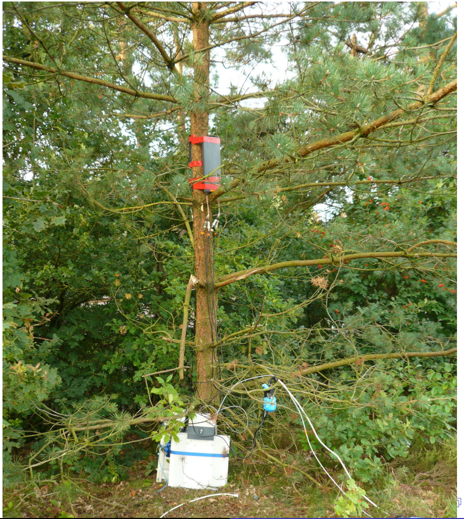
Osmocom / osmocom.org

- Osmocom == Open Soruce Mobile Communications
- Classic collaborative, community-driven FOSS project
- Gathers creative people who want to explore this industry-dominated closed mobile communications world
- communication via mailing lists, IRC
- soure code in git, information in trac/wiki
- http://osmocom.org/

OpenBSC

- first Osmocom project
- Implements GSM A-bis interface towards BTS
- Supports Siemens, ip.access, Ericsson, Nokia and sysmocom BTS
- can implement only BSC function (osmo-bsc) or a fully autonomous self-contained GSM network (osmo-nitb) that requires no external MSC/VLR/AUC/HLR/EIR
- deployed in > 200 installations world-wide, commercial and research

OpenBSC test installation

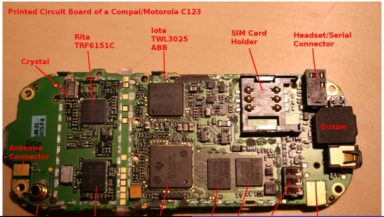


OsmoSGSN / OpenGGSN

- extends the OpenBSC based network from GSM to GPRS/EDGE by implementing the classic SGSN and GGSN functional entities
- OpenGGSN existed already, but was abandoned by original author
- Works only with BTSs that provides Gb interface, like ip.access nanoBTS
- Suitable for research only, not production ready

OsmocomBB

- Full baseband processor firmware implementation of a mobile phone (MS)
- We re-use existing phone hardware and re-wrote the L1, L2, L3 and higher level logic
- Higher layers reuse code from OpenBSC wherever possible
- Used in a number of universities and other research contexts



OsmocomTETRA

- SDR implementation of a TETRA radio-modem (PHY/MAC)
- Rx is fully implemented, Tx only partial
- Can be used for air interface interception
- Accompanied by wireshark dissectors for the TETRA protocol stack

OsmocomGMR

- ETSI GMR (Geo Mobile Radio) is "GSM for satellites"
- GMR-1 used by Thuraya satellite network
- OsmocomGMR implements SDR based radiomodem + PHY/MAC (Rx)
- Partial wireshark dissectors for the protocol stack
- Reverse engineered implementation of GMR-A5 crypto
- Speech codec is proprietary, still needs reverse engineering

OsmocomDECT

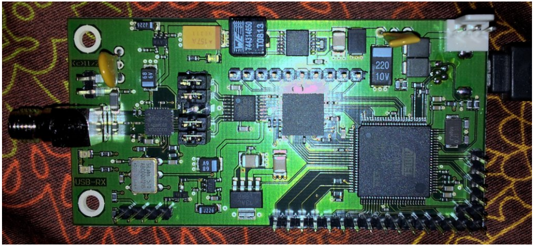
- ETSI DECT (Digital European Cordless Telephony) is used in millions of cordless phones
- deDECTed.org project started with open source protocol analyzers and demonstrated many vulnerabilities
- OsmocomDECT is an implementation of the DECT hardware drivers and protocols for the Linux kernel
- Integrates with Asterisk

OsmocomOP25

- APCO25 is Professional PMR system used in the US
- Can be compared to TETRA in Europe
- OsmocomOP25 is again SDR receiver + protocol analyzer

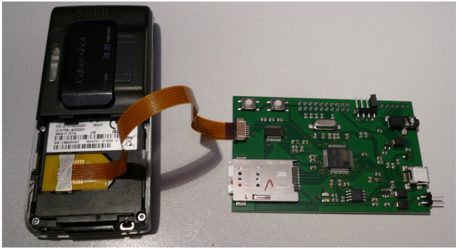
OsmoSDR

- small, low-power / low-cost USB SDR hardware
- higher bandwidth than FunCubeDonglePro
- much lower cost than USRP
- Open Hardware
- Board available to developers only (Firmware not finished)



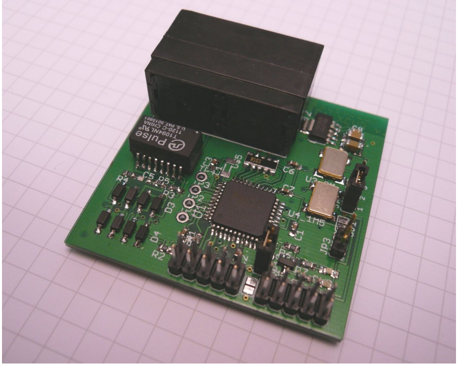
OsmocomSIMTRACE

- Hardware protocol tracer for SIM - phone interface
- Wireshark protocol dissector for SIM-ME protocol (TS 11.11)
- Can be used for SIM Application development / analysis
- Also capable of SIM card emulation and man-in-the-middle attacks



osmo-e1-xcvr

- Open hardware project for interfacing E1 lines with microcontrollers
- So far no software/firmware yet, stay tuned!



osmo-bts-amp

- Open hardware project for a 2W PA, LNA and ceramic duplexer to amplify small BTSs like ip.access nanoBTS
- 2W may sound little, but from 200mW it's a factor of 10
- Still much less than a regular macro cell, but more than a picocell for indoor coverage
- Scheamtics and Gerber files for the hardware available openly
- small and compact form factor compared to large/bulky cavity duplexers

OsmoCOS

- Smartcards such as SIM/USIM cards, but actually any type of chip/smartcards you can normally buy are proprietary and closed, as chip makers never release manuals
- Even if you write your own Card Operating System (COS), normally you would have to put it in mask ROM, requiring six or seven digit quantities as it basically would be your own version of the silicon.
- Thus, so far, all Smart Cards (even the OpenPGP Smart Card) run proprietary software inside

OsmoCOS

- We found a Chinese smarcard chip maker (ChipCity) that provided the programming manual to their chip without NDA. It has no ROM, but 256 kByte Flash and a known ARM7TDMI core.
- We started to write some low-level code like hardware drivers and can now work on our own Card Operating System
- Progress is slow, due to many other projects and few contributors

osmo_ss7, osmo_map, signerl

- Erlang-language SS7 implementation (MTP3, SCCP, TCAP, MAP)
- Sigtran variants (M2PA, M2UA, M3UA and SUA)
- Enables us to interface with GSM/UMTS inter-operator core network
- Already used in production in some really nasty special-purpose protocol translators (think of NAT for SS7)

The OpenBTS Um - SIP bridge

- OpenBTS is a SDR implementation of GSM Um radio interface
- directly bridges to SIP/RTP, no A-bis/BSC/A/MSC
- suitable for research on air interface, but very different from traditional GSM networks
- work is being done to make it interoperable with OpenBSC

airprobe.org

- SDR implementation of Um sniffer
- suitable for receiving GSM Um downlink and uplink
- predates all of the other projects
- more or less abandoned at this point

sysmocom GmbH

systems for mobile communications

- small company, started by two Osmocom developers in Berlin
- provides commercial R&d and support for professional users of Osmocom software
- develops its own producst like sysmoBTS (inexpensive, small-form-factor, OpenBSC compatible BTS)
- runs a small webshop for Osmocom related hardware like OsmocomBB compatible phones, SIMtrace, etc.

Where do we go from here?

- Dieter Spaar has been working with 3G NodeBs (Ericsson, Nokia) to be able to run our own RNC
- Research into intercepting microwave back-haul links
- Research into GPS simulation / transmission / faking
- Port of OsmocomBB to other baseband chips
- Low-level control from Free Software on a 3G/3.5G phone
- Re-using femtocells in creative ways
- Proprietary PMR systems

Call for contributions

- Don’t you agree that classic Internet/TCP/IP is boring and has been researched to death?
- There are many more communications systems out there
- Never trust the industry, they only care about selling their stuff
- Lets democratize access to those communication systems
- Become a contributor or developer today!
- Join our mailing lists, use/improve our code
- for OsmocomBB you only need a EUR 20 phone to start

Thanks

I'd like to thank the many Osmocom developers and contributors, especially

- Dieter Spaar
- Holger Freyther
- Andreas Eversberg
- Sylvain Munaut
- On-Waves e.h.f
- NETZING AG

Thanks

Thanks for your attention. I hope we have time for Q&A.