Berlin Institute of Technology
FG Security in Telecommunications

SMS of Death: from analyzing to attacking mobile phones on a large scale

USENIX Security 2011

Collin Mulliner, Nico Golde, Jean-Pierre Seifert
{collin,nico,jpseifert}@sec.t-labs.tu-berlin.de

Introduction

- Mobile phone security is a hot topic, but...
  - Previous work only focused on smartphones
- We always got the question: can you “hack” my cheap phone?
  - Cheap phone → Feature Phone
- **This work targets feature phones**
  - We investigate the (in)security of SMS implementations

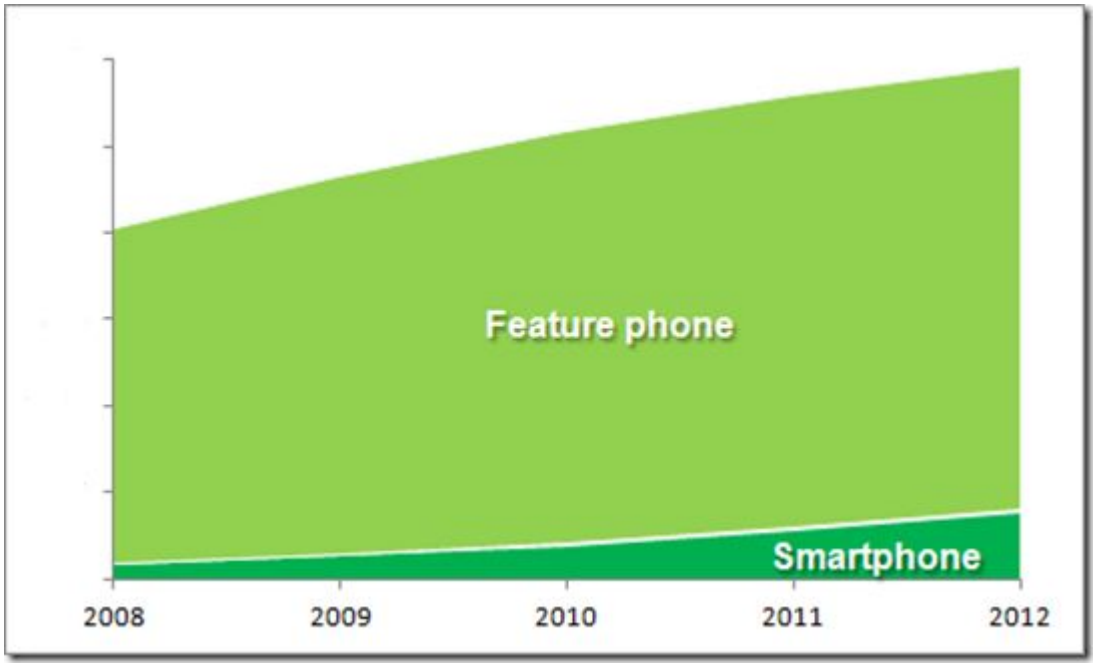
So what is a Feature Phone?

- Mobile phone with “additional features” → feature phone
  - Web browser, MP3 player, ...
- Single CPU device (smartphones normally have 2 CPUs)
  - Baseband and applications run on same processor
- 3rd party applications just J2ME, BREW, ...
  - No native code!
- Reasons why feature phones are still very popular
  - Price, battery run time, rugged case, ...



Why Feature Phones?

- World wide ~4.6 billion mobile phone users
- Only 16% of mobile phones in the world are smart phones!
  - A little more in the western world
- Therefore, feature phones → large impact!
- Feature phones haven been mostly ignored by other work.



Contributions

- **Vulnerability Analysis Framework for Feature Phones**
  - Novel method for crash monitoring
  - Analysis method based on a small GSM base station
- **Bugs Present in Most Feature Phone Platforms**
  - Bugs can be abused for Denial-of-Service attacks
- **Attack Impact**
  - Large scale attacks possible with only a few bugs
  - End users, manufacturers, operators

Feature Phone Platforms

- Manufacturer has one OS for their entire line of feature phones
  - Nokia **S40**, Sony Ericsson **OSE**, …

1) Since all phones are based on same platform
   - A bug found on phone *A* works on phones *B, C, D,* … *Z*

2) Single CPU architecture
   - Application crash → phone crash → reboot



Manufacturer Selection

- Way too many mobile phone manufacturers
  - We can't analyze after all of them
- Select the few ones that have a relevant market share
  - This makes sure that we have a global effect, remember our aim is “large scale”!

Manufacturer Selection

- Way too many mobile phone manufacturers
  - We can't analyze after all of them
- Select the few ones that have a relevant market share
  - This makes sure that we have a global effect, remember our aim is “**large scale**”!

Selected Manufacturers

- **Nokia, Samsung, Sony Ericsson, LG, Motorola,** and **Micromax**
  - Micromax is a very popular brand in India

- Market shares provide a good basis for targeted attacks
  - Say you want to attack mobile users in *Germany* you just look at the market shares for Germany and know what device(s) to target

(a) Germany, November 2009

Manufacturer	Market Share
Nokia	35.4%
Sony Ericsson	22.0%
Samsung	15.0%
Motorola	8.6%
Siemens	5.4%

(b) U.S.A., May 2010

Manufacturer	Market Share
Samsung	22.4%
LG	21.5%
Motorola	21.2%
RIM	8.7%
Nokia	8.1%

(c) Europe, June 2010

Manufacturer	Market Share
Nokia	32.8%
Samsung	12.5%
LG	4.1%
Sony Ericsson	3.7%
Apple	3.0%
RIM	2.4%
Others	3.0%

(d) World, for the year 2009

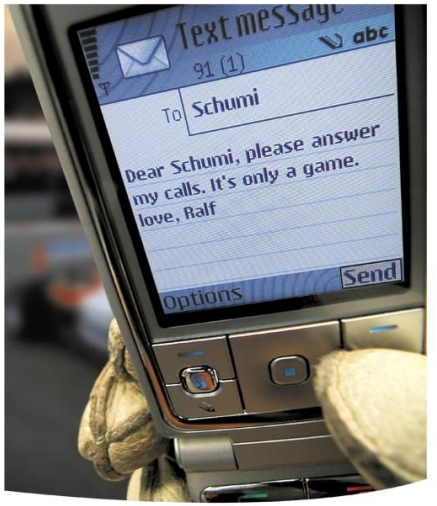
Manufacturer	Market Share
Nokia	38%
Samsung	20%
LG	10%
Sony Ericsson	5%
Motorola	5%
ZTE	4.5%
Kyocera	4%
RIM	3.5%
Sharp	2.6%
Apple	2.2%
Others	5%

Acquiring Phones

- We need a phones from all our selected manufacturers
  - We selected 6 manufacturers...
- Buying them new is no option, since this becomes expensive
  - About 150 Euro per phone
- eBay is our friend ;)
  - Decent feature phones are still expensive
  - We bought many “half broken” phones (5...30 Euro)

- Phones from eBay are always fun...
  - Many phones don't really allow a “hard reset”
    Phones still have: SMS, appointments, and pictures...

Why SMS (Short Message Service)?



- Supported by every mobile phone
  - ...and of course by every mobile operator
- Works everywhere in the world
  - Attacker can be located anywhere
  - No proximity required
- A ton of features
  - Flash SMS, VCard, MMS notification, multipart, port addressing, SIM toolkit, ...
  - Many implemented but rarely used (untested code!)
- Mostly no user interaction required
  - True remote bugs!

Analyzing Feature Phones … (the challenges)

- Completely closed system
  - Too many platforms
- No native 3rd party applications
  - No SDK and no debugger
- JTAG is no solution
  - Need detailed platform knowledge to use JTAG for serious work
  - Infeasible to hook up JTAG on 10+ different phones
- Reverse Engineering is a lot of work
  - Multiple platforms make it even worse
- Further: sending a lot of SMS messages is pricey

Our Solution

- **Use our own GSM network for analysis**
  - SMS messages for free
  - Speed improvement over real operator network
  - Full control over the entire environment
  - Use phone ↔ BTS communication for analysis

- Fuzzing-based testing
  - No source code no reverse engineering required
  - Make test cases once … use them for all phones

- Fuzzing requires monitoring
  - Without monitoring fuzzing is useless!

GSM Network Equipment

- Industry traditionally very closed
  - Protocol specs exist (>1k PDFs)
  - No public documentation of GSM equipment

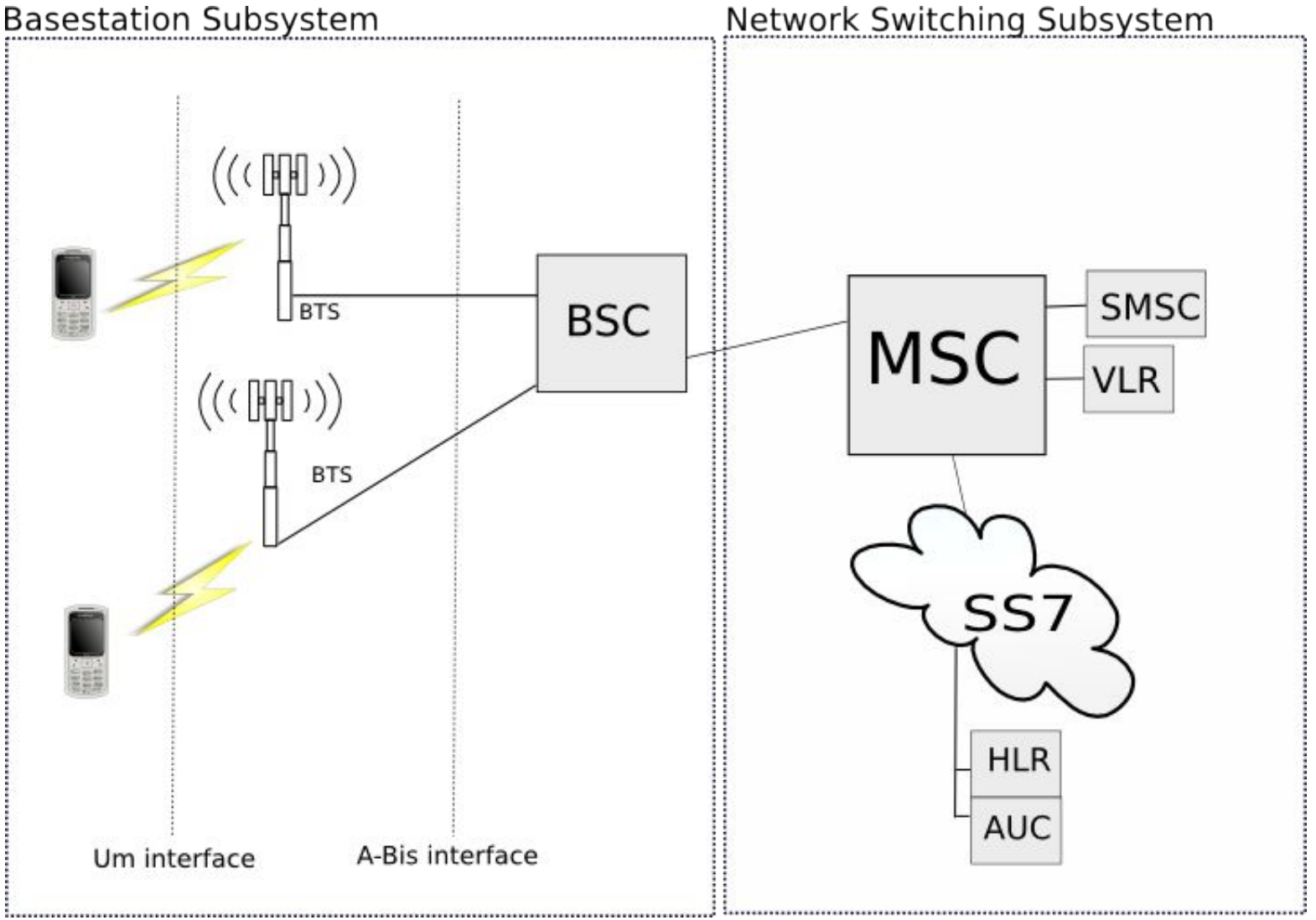
→ OpenBSC, OpenBTS, OsmocomBB are game changers

- OpenBSC:
  - Free Software implementing A-bis over IP
  - Minimal subset of HLR, MSC, SMSC, BSC, and AUC
  - Supports a number of different base transceiver stations

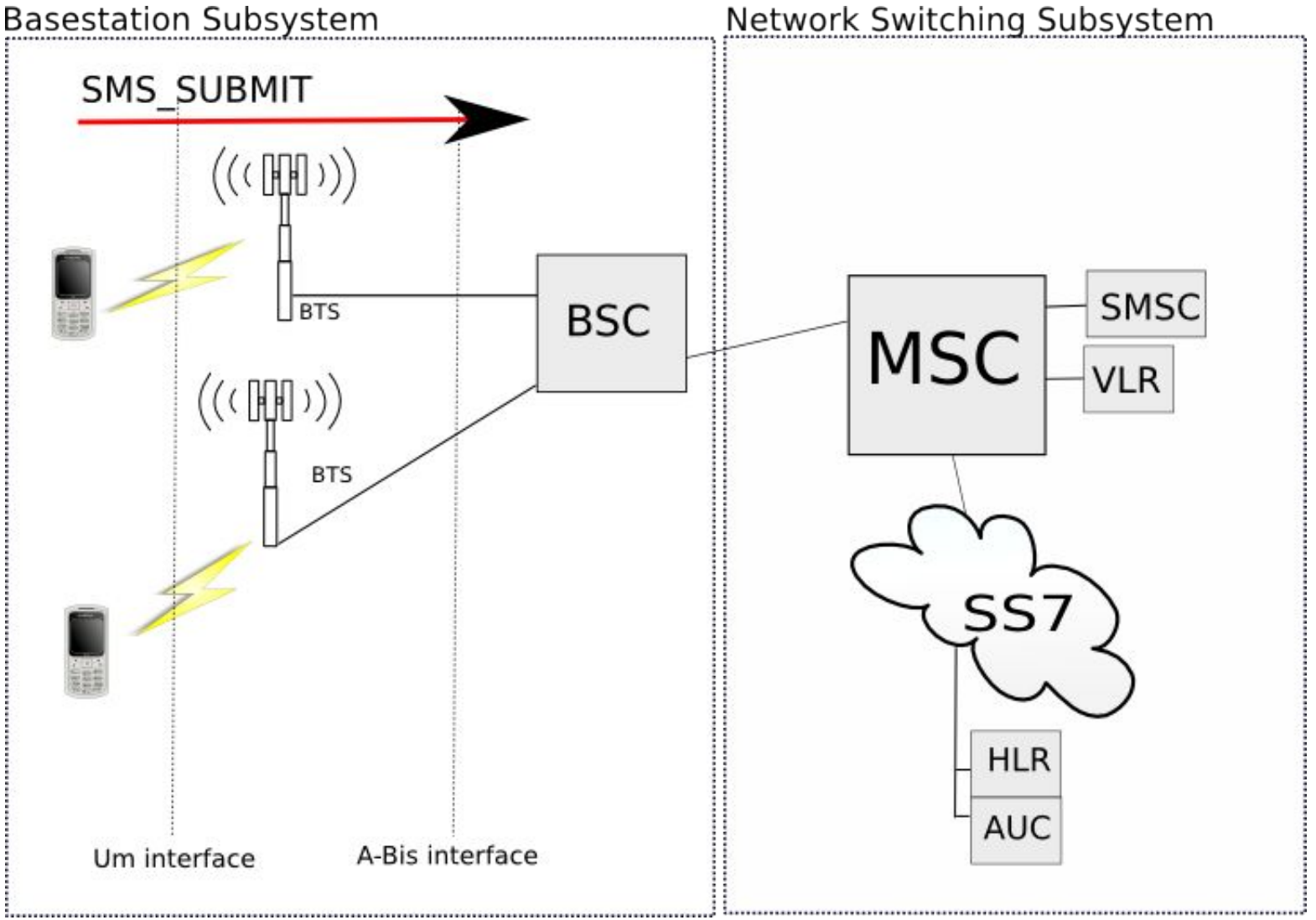
Our Setup

Laptop (running OpenBSC), nanoBTS, and some phones

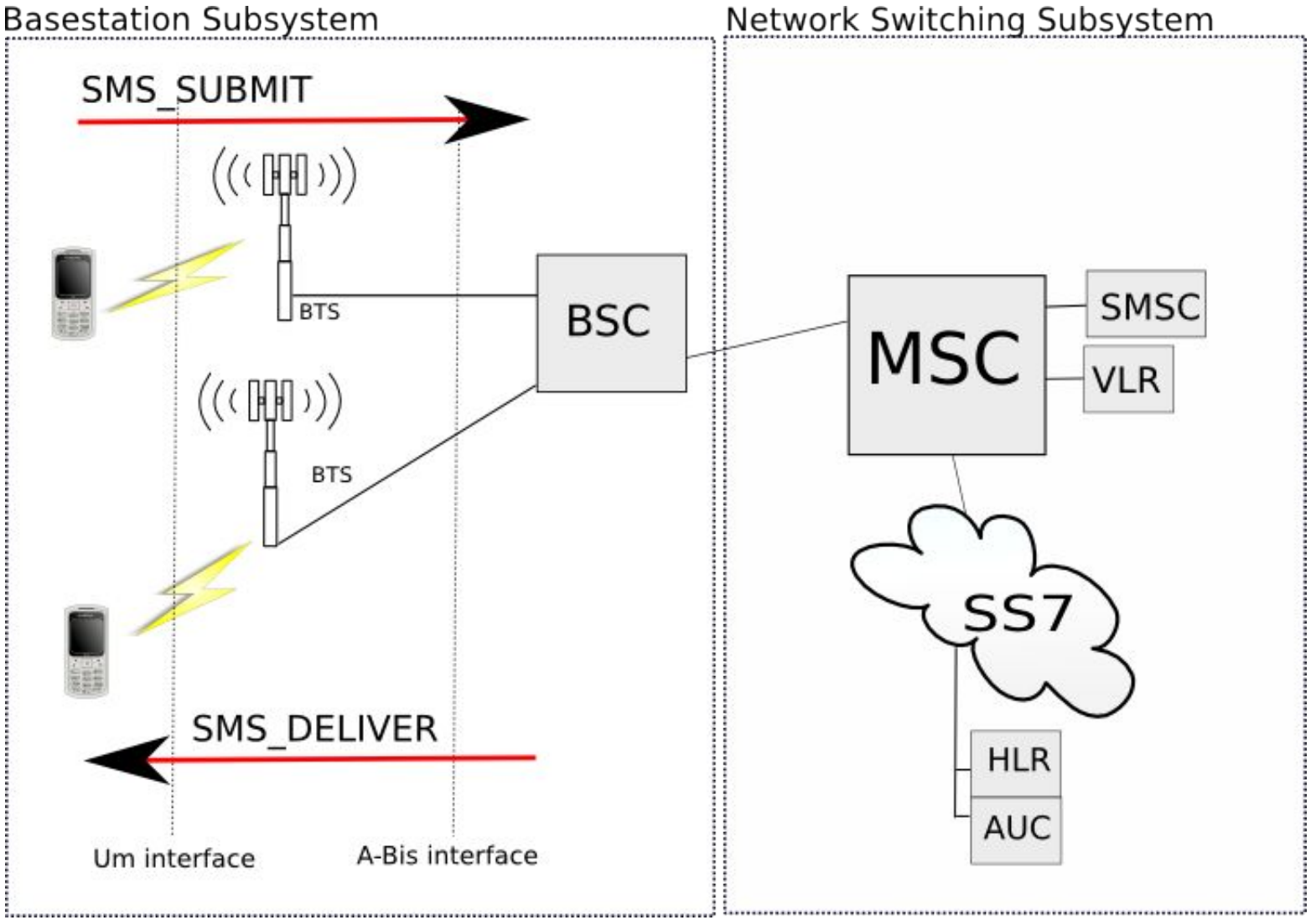
A typical GSM network (simplified)



SMS submission



SMS delivery



OpenBSC and SMS

- Supports SMS from phone → phone
- Provides text-based interface for text-only SMS messages
  → by default not fuzzing friendly
  - Only text
  - Very slow/for attached subscribers
  - Stored message sent to all subscribers

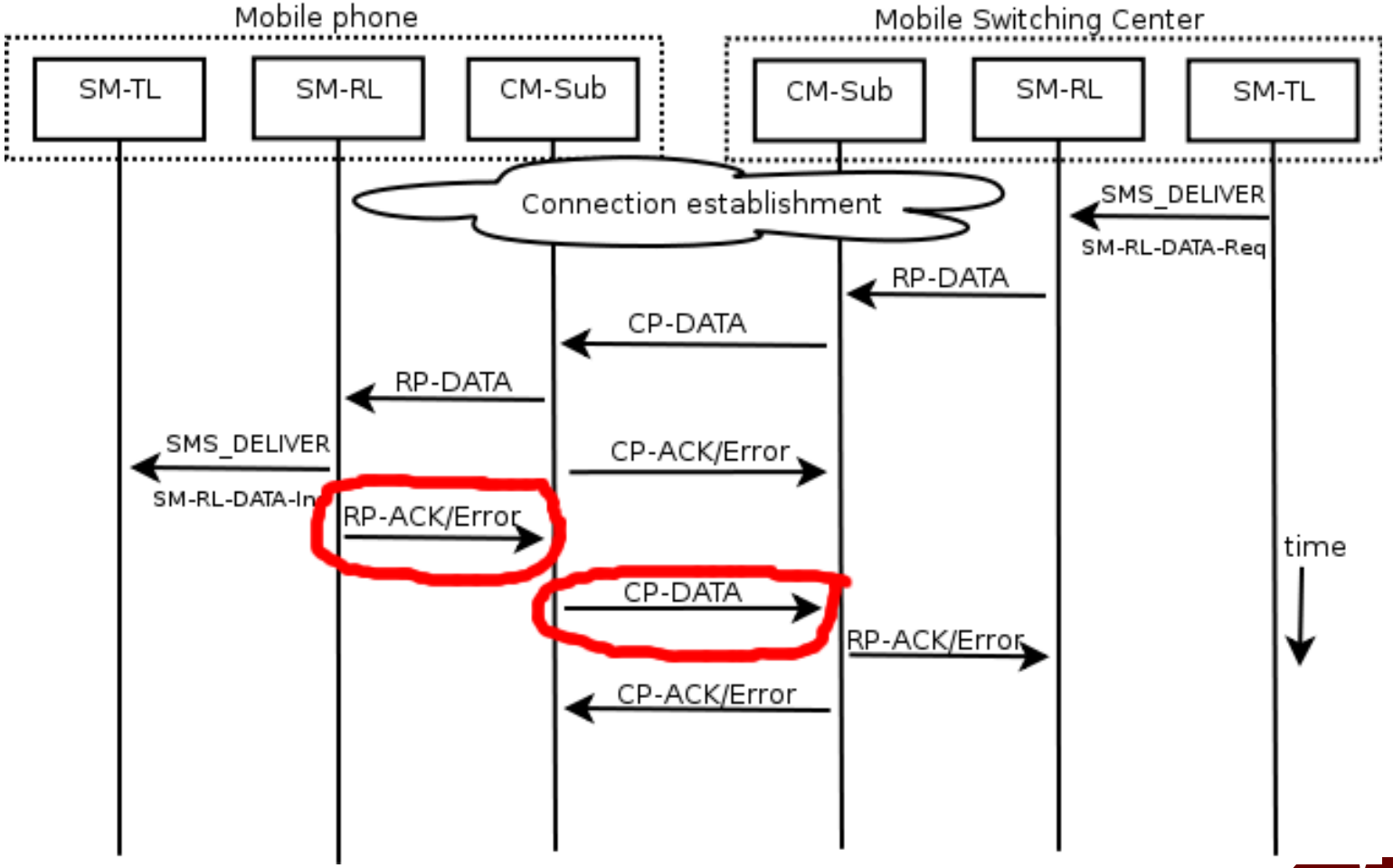
OpenBSC Modifications

- Injection of pre-encoded SMS in PDU format (SMS_SUBMIT)
- Relaxed message checking
  - Allow fuzzed/unsupported message types
- Logging
  - Phone feedback: Memory full, Protocol errors, ...
  - Channel release states (break downs)
- Event → message mapping

```
phone (1331) went offline at 2010-10-29 14:28:37,
checking last sms...
the error was very likely caused by the following sms:
41000491311300f1880500034affdb404040404040404....
```

Monitoring the Phones

- Messages sent over SDCCH/SACCH
  - Monitor feedback and channel tear down



Additional monitoring

- Finding more than crashes
  - State “mess up” → swallowed messages
- Health monitoring with “echo server” on the phone
  - Binds to SMS port
  - Receives incoming message
  - Replies with message to “special” number
  - Implemented in J2ME

- Inject “echo” SMS every *N* messages
  - Check message counter in SMSC database (OpenBSC)

Test cases

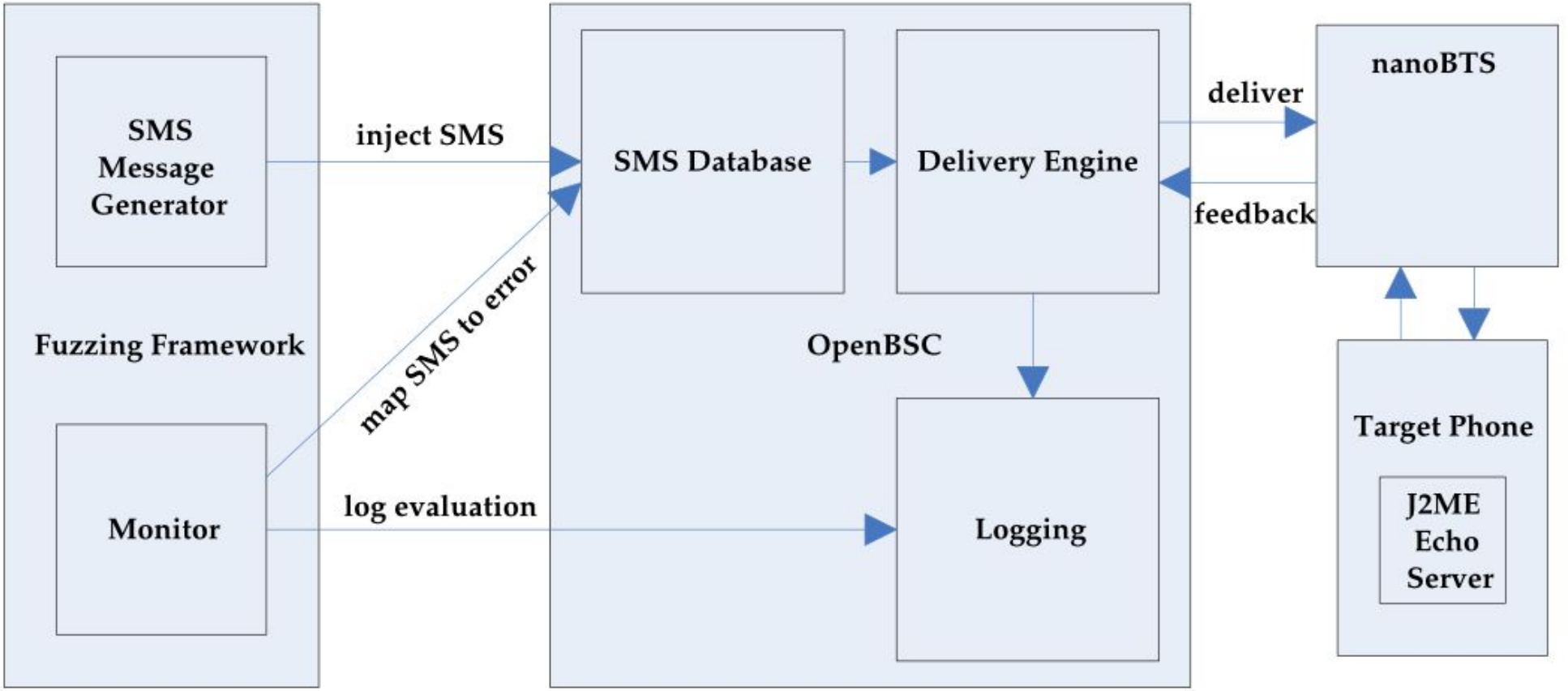
- Multipart
  - UDH (reference, parts, current part)
- MMS notification
  - Various variable length strings
- Simple text
  - Invalid alphabet encoding (array out of bounds)
- Flash SMS
  - Separated code paths
  - Multipart
- TP-PID/TP-DCS combinations
  - In combination with UD payload

- ~120k messages

Fuzzing trial

- Python library for SMS generation
- Submit ~1000 of messages to OpenBSC
  - Stored in SMSC database
- Send message to fuzz-phone(s)
  - **Open channel**
  - **Send message 1...n**
  - **Close channel**
- Script evaluating added logging
  - Flag invalid messages
  - Monitor channel breakdown → SMS

The Complete (logical) Setup



Results

- Fuzzed for quite some time
  - Took a lot of work
- A lot of automation but you still have to...
  - Delete messages by hand
  - Get phones out of the “totally stuck” mode → “hard reset”
- We were mostly looking for crashes that...
  - Disconnect phone from network
  - Reboot the phone
- Here are some interesting bugs we found!

Nokia S40

- The world wide market leader!
- S40 → Nokia's feature phone platform
  - Our test phones: 3110c, 6300, 6233, 6131 NFC,...

- BUG
  - 8 bit class 0 (Flash SMS) with certain TP-UD payload
- Impact
  - “Nokia White Screen of Death”
  - Interface reboot
  - Disconnect phone from network (interrupting call)
  - Message ACK never reaches network (more on that later...)
  - Message not visible on the phone
  - Watchdog shuts down phone after repeated crashes

Sony Ericsson

- Very common in Germany (22% market share)
- Test phones: w800i, w810i, w890i, Aino (May 2010)

- BUG
  - Certain (reserved) TP-PID value & >= certain length TP-UD

- Impact
  - Complete phone reboot
  - Disconnect phone from network (interrupting call)
  - Message ACK never reaches network (again, later...)
  - Message not visible on phone
  - Sometimes also completely freezes
  - Errm, one test phone bricked

LG Electronics

- Test phone: LG GM360, likely more phones affected
- BUG
  - Classic buffer overflow in various MMS notification fields
- Impact
  - Phone reboots
  - If PIN set → phone locked (permanently offline)
  - Disconnects from network (interrupting calls)
  - Same happens on opening the message
- Good target for future work (reversing/code execution)

Motorola

- Test Phones: Razr, Rokr, SVLR L7

- BUG
  - Internet Electronic Mail interworking (0x32)
    + certain payload
- Impact
  - Flashing white screen
  - User interface restart
  - Network disconnect (interrupt calls)

- Rather fragile devices, couldn't test in-depth due full memory, weird behavior...

Micromax

- Number three (3) manufacturer in India!
- Test phone: X114 (tested briefly, last arrived phone)

- BUG
  - Multipart assembly madness again (this time Flash)
  - Reference id has to be unused (no problem)

- Impact
  - Few seconds after receipt → black screen
  - Network disconnect (interrupt calls)
  - Message is silent

Notifying Vendors

- Nokia
  - no problem, got contacts from the past
- Sony Ericsson
  - Painful, but we met some guy at a security conference ;-)
- Motorola
  - security@motorola.com does not really work that well
- Samsung
  - Contacted
- LG
  - Haven't found a security contact, but contacted through GSMA
- Micromax
  - Haven't found a security contact, contacted through GSMA

The Special “early” Crash

- Some bugs crash the phone before ACKing the SMS to the net
  - Nokia + Sony Ericsson
- Results: Network believes SMS was not received
- Action: SMSC tries to re-transmit message
  - Phone crashes again
  - Repeat...
  - Fix: move SIM card to non affected phone

The Special “early” Crash

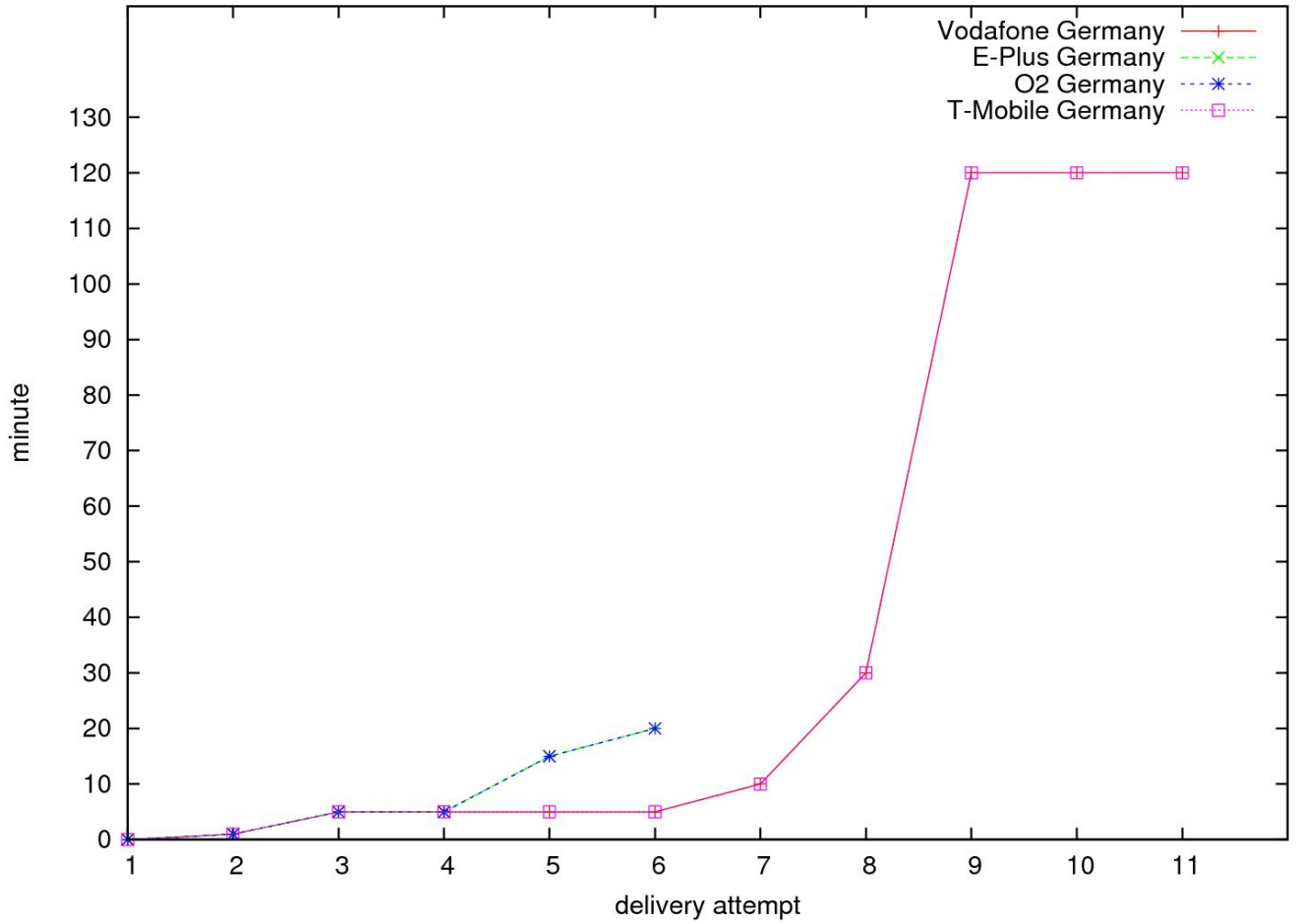
- Some bugs crash the phone before ACKing the SMS to the net
  - Nokia + Sony Ericsson
- Results: Network believes SMS was not received
- Action: SMSC tries to re-transmit message
  - Phone crashes again
  - Repeat...
  - Fix: move SIM card to non affected phone
- **Conclusion: Abuse behavior for attack amplification**
  - Send one message → network makes phone crash multiple times
  - How often and in what interval is this happening?

Testing SMS Re-Transmits Timings

- Linux PC with Bluetooth dongle + Sony Ericsson phone
- Monitor phone using Bluetooth RFCOMM link
  - Connect to “Dialup Networking Service”
  - Wait until Bluetooth link gets disconnected (phone reboots!)
- Attack phone, count reboots
  - Let it run for a few days (swap SIM cards in between)

SMS Re-Transmit Timings for German MNOs

Additional delivery attempt 20/24 hours after last attempt shown in graph



Attacks

- Clearly bugs can be used for attacks
- Disconnect calls
  - With just 1 SMS, to either side of the call (if both are mobile)
- Make sure a “specific” person is not reachable
  - Send an SMS every few seconds
  - Costs a lot, but maybe its worth it
  - If the phone switches off it will be cheap (Nokia)

Large Scale Attacks... possible

- **Mobile Network Operator** (MNO) → disconnect his customers
  - Make him look bad
  - Extort him (organized crime)
    (customers might claim their phone to be broken)
  - Smaller operator will likely have issues with a massive number of reconnecting phones

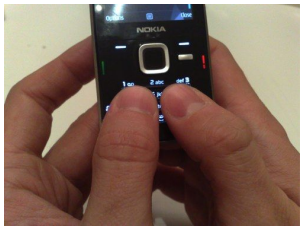
- **Manufacturer** → attack random people owning specific brand
  - Make them look bad
  - Extort him (organized crime)

- **Public Distress** → disconnect a lot of people
  - Next big outdoor event (protest, festival, etc...)
  - Police often relies on mobile phones
  - Remember Estonia 2007?
    (...will become expensive)

Sending large Quantities of SMS Messages

- Using a few normal phones wont work
  - Very slow, pricey, easily traceable, ...
- Bulk SMS operators (the guys you go to for SMS spam)
  - Cheap, no-questions asked, high injection rate

- Smart/mobile phone botnets
  - Cheap (free!), fast if you have a large botnet (remember all those jailbroken iPhones with SSH and default root password?)



- SS7 Access
  - SPEED, good price, hard to trace, no content limitations (you are/know an operator)

Countermeasures: SMS filtering by MNOs

- Mobile Network Operators can filter SMS messages
- But filter software seems not well prepared for binary
  - Mostly designed to fight SMS spam and filter political content
- How to configure filters? (work done after this paper was finished)
  - We don't want to publish payloads (deal with manufacturers!)
  - We compiled a white paper that tells you what to filter
  - White paper is available from:

    http://tinyurl.com/smssecurity/

Conclusions

- With openness on the GSM network side one can find bugs in the “closed” mobile phones
- Bugs in all major feature phone platforms!
- Large scale attacks are possible with this bug arsenal
- SMS re-transmit by operator amplifies the attacks
- Attack against users possibly can lead to attack against operator
- Manufacturers need to provide updates for feature phones

Q & A

Thank you for your attention!

Question?

Demo Video


Feature Phone SMS Security Analysis
Nico Golde, Collin Mulliner
SECT Security
IN TELECOMMUNICATIONS
NOKIA
LG
Eingang
Keine Nachrichten
Musik-Player
Radio
Favoriten
Menü
Namen
Sony Ericsson
MOTOROLA
NOKIA