“SMS-o-Death: from analyzing to attacking mobile phones on a large scale”

CanSecWest 2011

Nico Golde & Collin Mulliner, Vancouver, March 9-11 2011
{nico,collin}@sec.t-labs.tu-berlin.de
About us

- Collin Mulliner
  - coder, hacker, security researcher, PhD student
  - Past:
    - p0wnd iPhone, Android, Windows Mobile with SMS
    - Bluetooth and NFC phone security
    - p0wnd Windows Mobile with MMS

- Nico Golde
  - (almost not anymore) student
Agenda

- Introduction
- SMS
- Fuzzing Setup
- Fuzzing Results
- Fun with the Network Operators
- Attacks
- Conclusions
Introduction

- Mobile phone security research is a really HOT topic right now

- Research areas
  - Protocol level attacks
  - Crypto (A5/1)
  - Application level attacks on smart phones
  - SMS-based attacks against smart phones

- > 4 billion mobile phone users
  - High attack surface
Introduction

- Mobile phone security research is a really HOT topic right now

- Research areas
  - Protocol level attacks
  - Crypto (A5/1)
  - Application level attacks on smart phones
  - SMS-based attacks against smart phones

- > 4 billion mobile phone users
  - High attack surface

- In this talk we will focus on feature phones
  - We will look at 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 (smart phones 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!
- Further: feature phones haven been ignored by previous work!
Feature Phone Platforms...

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

- Theory 1: since all phones are based on same platform
  - A bug found on phone A works on phones B, C, and D

- Theory 2: single CPU architecture
  - Application crash → phone crash → reboot
Manufacturer Selection

- Way too many mobile phone manufacturers
  - We can't go after all of them

- Select the few ones that have a good 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 go after all of them

- Select the few ones that have a good 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 are a good basis for targeted attacks
  - Say you want to attack mobile users in Germany you just look at the market shares for your target country and know what to attack ;-)

Data: ComScore (see references...)

---

### Market Share Data

**Germany, November 2009**

<table>
<thead>
<tr>
<th>Manufacturer</th>
<th>Market Share</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nokia</td>
<td>35.4%</td>
</tr>
<tr>
<td>Sony Ericsson</td>
<td>22.0%</td>
</tr>
<tr>
<td>Samsung</td>
<td>15.0%</td>
</tr>
<tr>
<td>Motorola</td>
<td>8.6%</td>
</tr>
<tr>
<td>Siemens</td>
<td>5.4%</td>
</tr>
</tbody>
</table>

**U.S.A., May 2010**

<table>
<thead>
<tr>
<th>Manufacturer</th>
<th>Market Share</th>
</tr>
</thead>
<tbody>
<tr>
<td>Samsung</td>
<td>22.4%</td>
</tr>
<tr>
<td>LG</td>
<td>21.5%</td>
</tr>
<tr>
<td>Motorola</td>
<td>21.2%</td>
</tr>
<tr>
<td>RIM</td>
<td>8.7%</td>
</tr>
<tr>
<td>Nokia</td>
<td>8.1%</td>
</tr>
</tbody>
</table>

**Europe, June 2010**

<table>
<thead>
<tr>
<th>Manufacturer</th>
<th>Market Share</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nokia</td>
<td>32.8%</td>
</tr>
<tr>
<td>Samsung</td>
<td>12.5%</td>
</tr>
<tr>
<td>LG</td>
<td>4.1%</td>
</tr>
<tr>
<td>Sony Ericsson</td>
<td>3.7%</td>
</tr>
<tr>
<td>Apple</td>
<td>3.0%</td>
</tr>
<tr>
<td>RIM</td>
<td>2.4%</td>
</tr>
<tr>
<td>Others</td>
<td>3.0%</td>
</tr>
</tbody>
</table>

**World, for the year 2009**

<table>
<thead>
<tr>
<th>Manufacturer</th>
<th>Market Share</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nokia</td>
<td>38%</td>
</tr>
<tr>
<td>Samsung</td>
<td>20%</td>
</tr>
<tr>
<td>LG</td>
<td>10%</td>
</tr>
<tr>
<td>Sony Ericsson</td>
<td>5%</td>
</tr>
<tr>
<td>Motorola</td>
<td>5%</td>
</tr>
<tr>
<td>ZTE</td>
<td>4.5%</td>
</tr>
<tr>
<td>Kyocera</td>
<td>4%</td>
</tr>
<tr>
<td>RIM</td>
<td>3.5%</td>
</tr>
<tr>
<td>Sharp</td>
<td>2.6%</td>
</tr>
<tr>
<td>Apple</td>
<td>2.2%</td>
</tr>
<tr>
<td>Others</td>
<td>5%</td>
</tr>
</tbody>
</table>

Data: ComScore (see references...
Acquiring Phones

- We need a phone 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 everywhere
  - 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 Problem

- 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
  - Don't want to hook up JTAG +10 different phones

- Reverse Engineering is a lot of work
  - Multiple platforms make it even worse
The Solution...

- **Use own GSM network for analysis**
  - SMS messages for free!
  - Don't interfere with operator's network
  - Speed improvement over real operator network
  - Full control over everything
  - 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

- But 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, AUC
  - Supports nanoBTS and BS11
The 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 telnet interface for text 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: 41000491311300f1880500034affdb40404040404040404...
Monitoring the Phones

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

- Finding more than crashes
  - State fuckups → 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)
**SMS_SUBMIT**

- “Hello World” SMS to 1234 in PDU format

```
01000491214300000BE8329BFD06DDDF723619
```

<table>
<thead>
<tr>
<th>Field</th>
<th>Size</th>
<th>Bytes (Hex)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUBMIT</td>
<td>1</td>
<td>01</td>
</tr>
<tr>
<td>TP-MR</td>
<td>1</td>
<td>00</td>
</tr>
<tr>
<td>Destination</td>
<td>5</td>
<td>04 91 2143</td>
</tr>
<tr>
<td>TP-PID</td>
<td>1</td>
<td>00</td>
</tr>
<tr>
<td>TP-DCS</td>
<td>1</td>
<td>00</td>
</tr>
<tr>
<td>TP-VP</td>
<td>variable</td>
<td>00</td>
</tr>
<tr>
<td>TP-UDL</td>
<td>1</td>
<td>0B</td>
</tr>
<tr>
<td>TP-UD</td>
<td>variable</td>
<td>E8329BFD06DDDF723619</td>
</tr>
</tbody>
</table>

```

"Hello World" SMS to 1234 in PDU format

01000491214300000BE8329BFD06DDDF723619

<table>
<thead>
<tr>
<th>Field</th>
<th>Size</th>
<th>Bytes (Hex)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUBMIT</td>
<td>1</td>
<td>01</td>
</tr>
<tr>
<td>TP-MR</td>
<td>1</td>
<td>00</td>
</tr>
<tr>
<td>Destination</td>
<td>5</td>
<td>04 91 2143</td>
</tr>
<tr>
<td>TP-PID</td>
<td>1</td>
<td>00</td>
</tr>
<tr>
<td>TP-DCS</td>
<td>1</td>
<td>00</td>
</tr>
<tr>
<td>TP-VP</td>
<td>variable</td>
<td>00</td>
</tr>
<tr>
<td>TP-UDL</td>
<td>1</td>
<td>0B</td>
</tr>
<tr>
<td>TP-UD</td>
<td>variable</td>
<td>E8329BFD06DDDF723619</td>
</tr>
</tbody>
</table>
More...

- simple text message
- Messages can carry binary payload
- Additional features added by UDH chunks
  - Part of TP-UD

<table>
<thead>
<tr>
<th>Field</th>
<th>Size</th>
<th>Bytes (HEX)</th>
</tr>
</thead>
<tbody>
<tr>
<td>IET</td>
<td>1</td>
<td>05</td>
</tr>
<tr>
<td>IEDL</td>
<td>1</td>
<td>04</td>
</tr>
<tr>
<td>IED</td>
<td>4</td>
<td>0B8423F0</td>
</tr>
</tbody>
</table>

16 bit port addressing, dst: 2948 src: 9200
UDH features

- Concatenated messages
- Port addressing (8 and 16 bit)
  - WAP-push
  - MMS notification
  - iPhone visual voicemail
- Rich text formatting (EMS)
- RFC 822 Email header
- (U)SIM Toolkit
- Sound
- Lots of others...
- Can be combined
SMS/UDH example (MMS notification)
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 tons 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
Our Faraday Cage ... so we can do what ever...

- Don't need a GSM license if you have one of these :)

![Faraday Cage Image]
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)
Samsung 1/2

- Test phones: S5230 Star, B5310 CorbyPro

- BUG
  - Multipart: chunk id madness

- Impact
  - Displayed message size huuuuge
  - Phone crashes on opening message
  - Network disconnect
  - User interaction required :-(/
Samsung 2/2

- Test phones: S5230 Star, B5310 CorbyPro

- BUG
  - Modified version of the payload

- Impact
  - Phone denies every SMS with Protocol error (*wink* Curse of Silence)
  - One silent message (no user interaction)
  - SMS application won't open again (Messages loading...)
  - Phone application won't open again
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
Demo Video
Notifying Vendors

- Nokia
  - no problem, got contacts from the past
- Sony Ericsson
  - email was #fail, but I ran into one of them at a con #win
- Motorola
  - security@motorola.com does not really work that well
- Samsung
  - Got contacted in Jan 2011 after initial presentation
- LG
  - Haven't found a security contact
- Micromax
  - Haven't found a security contact
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
  - Let's see how often and in what interval this happens...
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 tried 20/24 hours after last try shown in graph
Attacks

- Clearly we can (ab)use our bugs for attacks
- Disconnect calls
  - With just 1 SMS, to either side of the call (if both are mobile)
- Make sure you are not reachable
  - Send you an SMS every few seconds
  - Maybe costs a lot, but maybe you are worth it?
  - If we get your phone to switch off it will be cheap (Nokia)
Large Scale Attacks... possible

- **Mobile Network Operator** (MNO) → disconnect his customers
  - Make him look bad (fun)
  - Extort him (organized crime)
    (customers might claim their phone to be broken)
  - **Will 10,000 reconnecting phones kill the operators infrastructure?**

- **Manufacturer** → attack random people owning specific brand
  - Make them look bad (fun)
  - 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?
    (okay ... will become expensive)
Sending large Quantities of SMS Messages

- Using a few normal phones won't work
  - Very slow, pricey, easily traceable, ...

- Bulk SMS operators (the guys you go to for SMS spam)
  - Cheap, no-questions asked, high injection rate (fun!!)
    (our favorites: HSL, Clickatell, Routomessaging, ...) 

- 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, know somebody...)
Feature Phones and Firmware Updates

- **Price**
  - Phones are quite cheap → manufacturers don't offer updates

- **Branding**
  - Phones are branded by operators → firmware can only be updated with branded firmware image

- **Net-Lock**
  - Phones can often not be updated → updates can be used to remove the net-lock

- **Installing the Update**
  - How do you know there is one? Your phone doesn't tell you
  - Need a desktop computer? Or even go to a special store
Counter measures: SMS filtering by MNOs

- Mobile Network Operators can obviously filter SMS messages

- Filter software seems not well prepared for binary
  - Mostly designed to fight sms spam and filter political content

- How to configure filters?
  - We don't want to publish payloads (deal with manufacturers!)
  - We compiled a white paper that tells you what to filter
  - White paper will be 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 totally possible with this bug arsenal
- **SMS re-transmit by operator helps** you with attacks
- Attack against users **possibly** leads to attack against operator
- Manufacturers need to provide **updates for feature phones**
The End: Q & A

Thank you for listening!
Question?

Contact:
- nico@sec.t-labs.tu-berlin.de
- collin@sec.t-labs.tu-berlin.de Twitter: @collinrm
- http://www.sec.t-labs.tu-berlin.de
Thanks and Greez

- Special Thanks
  - Ravii and Simon Schoar for buy / borrowing us phones!

- Greez (in no particular order)
  - Harald Welte
  - Dieter Spaar
  - ak
  - FX
  - Joernchen
  - Mumpi
  - scusi
  - ths
  - shadow
  - Charlie Miller
  - Martin Herfurt
References
