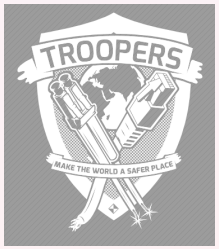
Dirty use of USSD codes in cellular networks

Ravishankar Borgaonkar

Security in Telecommunications, Technische Universität Berlin

TelcoSecDay, Heidelberg, 12th March 2013



Agenda

- USSD codes and services in mobile telephony
- Attacks in USSD network infrastructure
- Attacks on smartphones (end-users)

technology - features - applications

What is USSD in mobile telephony?

- a messaging service between mobile phones and an application server in the network
- but data is transferred in real time as a session (no SMSC-store and forward)
- faster than SMS and interactive service
- supported by all mobiles - feature phones to smartphones
- why USSD? to increase ARPU (Average Revenue Per User)

Services

Services based on USSD protocol:

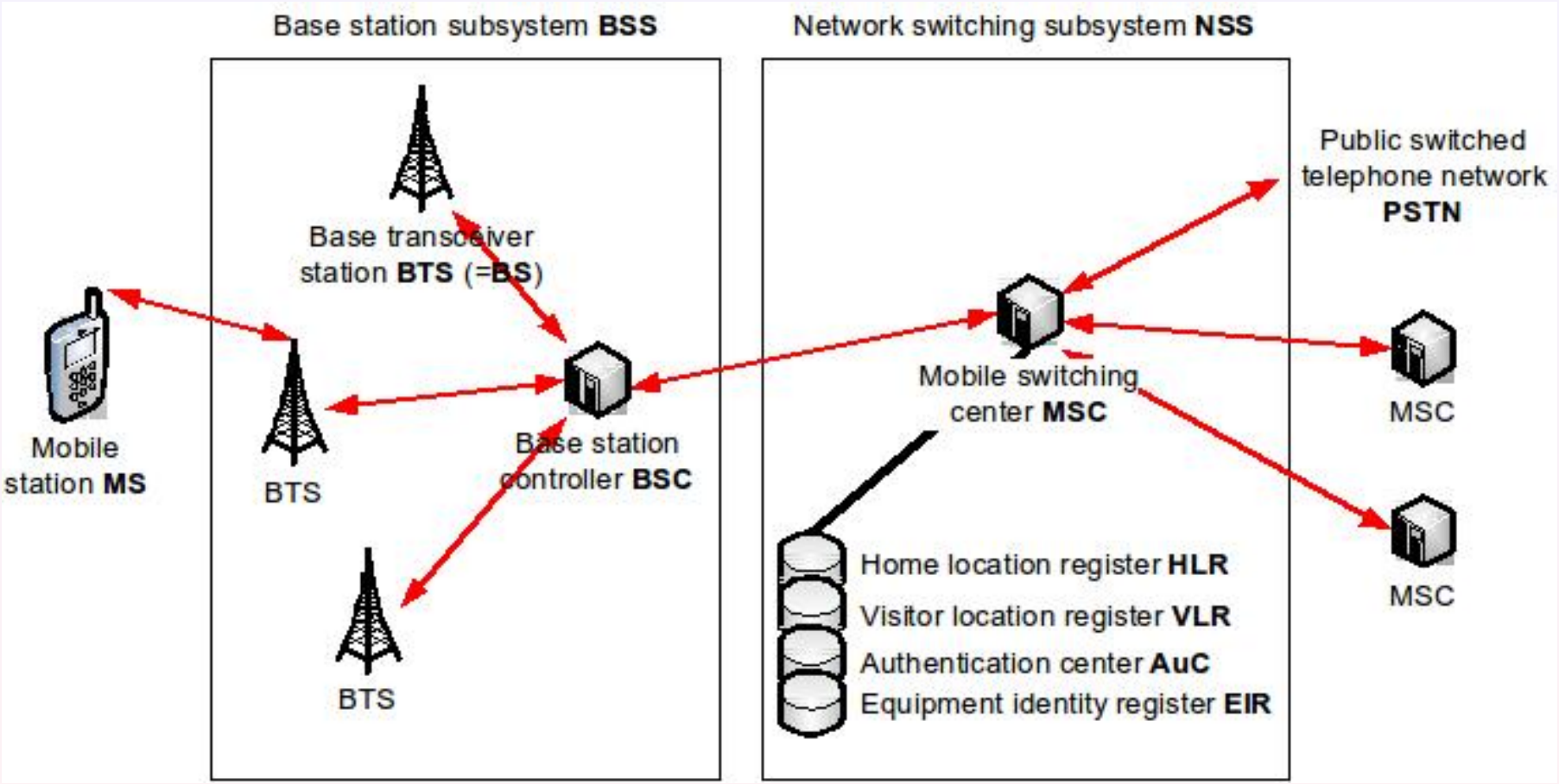
- interactive data services (News, Sports etc)
- pre-paid phone top-up and balance queries
- mobile banking and money services
- access to social services such as Twitter, Facebook

Toilet thinking

Motivation stories

- Airtel Money in India, really?
- An interesting document
- playing with NFC protocol on Android with Collin

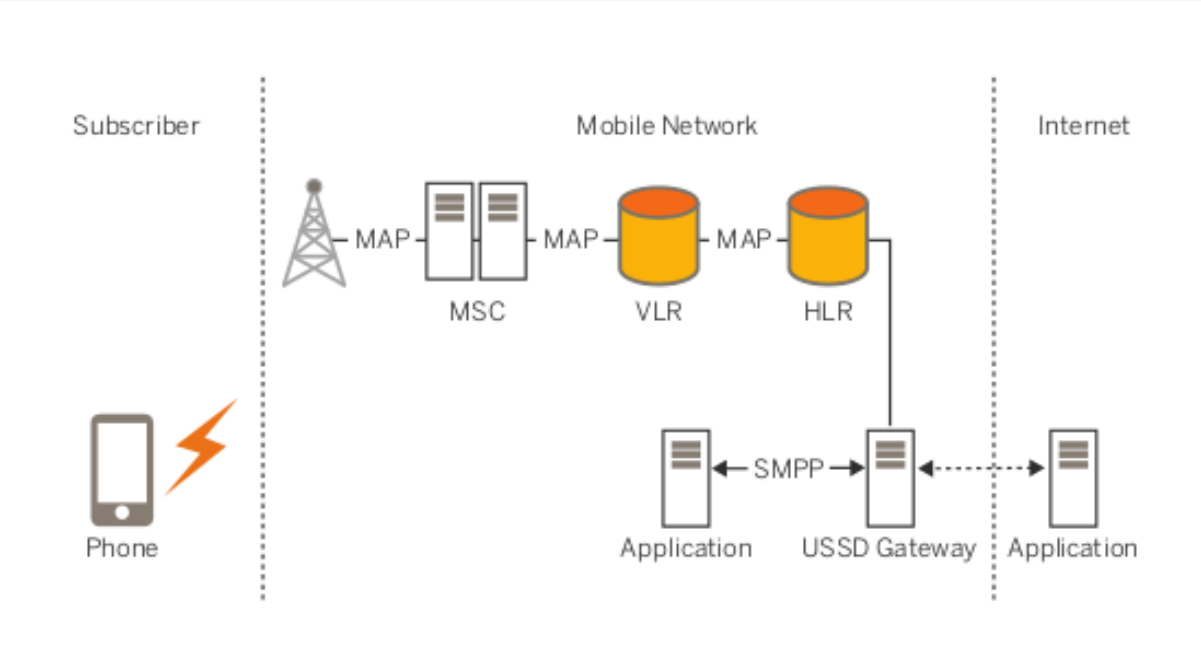
GSM cellular architecture



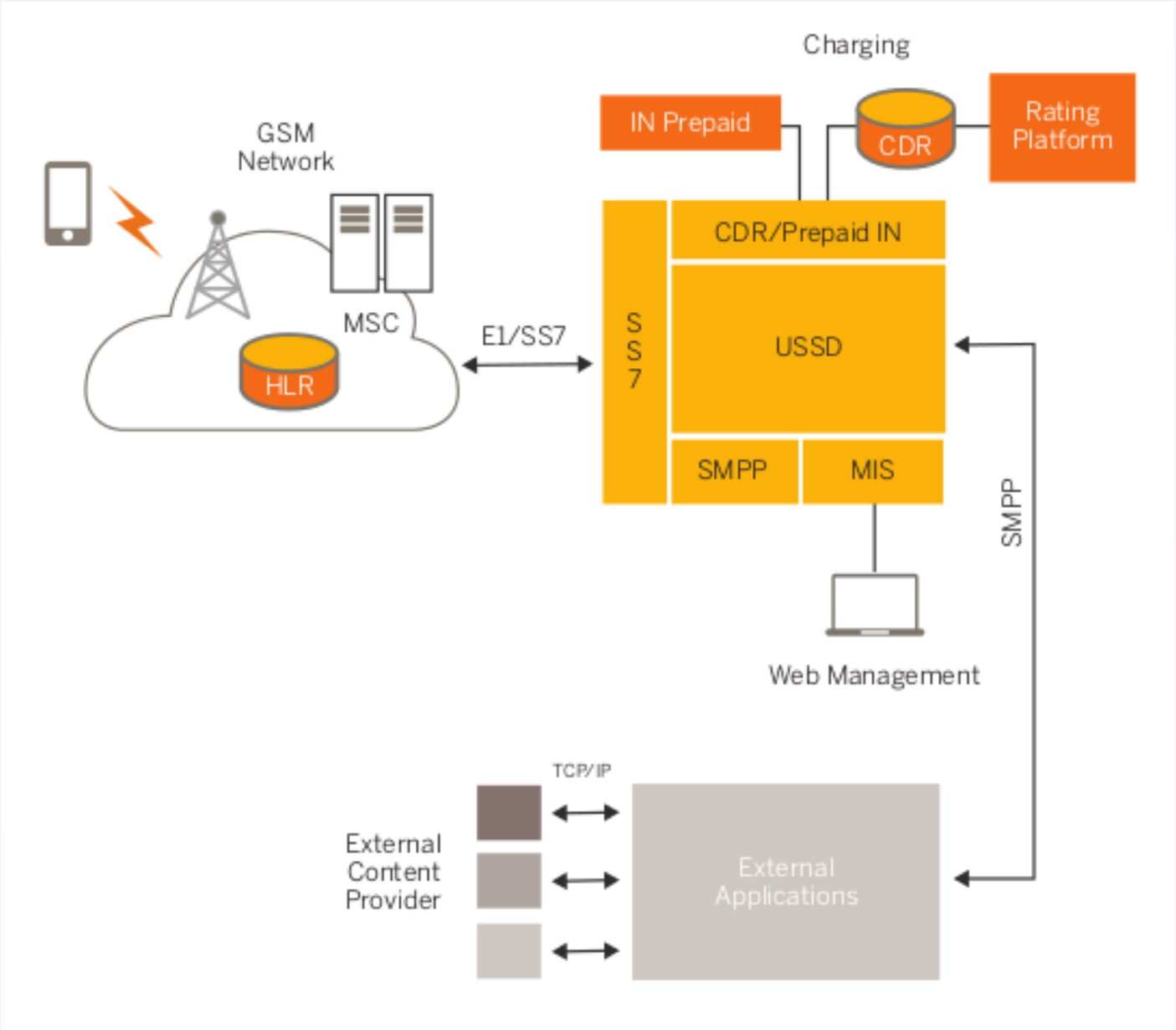
USSD Architecture

Architectural components:

- MSC (Mobile Switching Center),VLR (Visitor Location Register)
- USSD Gateway
- USSD application/server
- Simple Messaging Peer-Peer interface



Application Flow Example



Service Example

Mobile Banking:

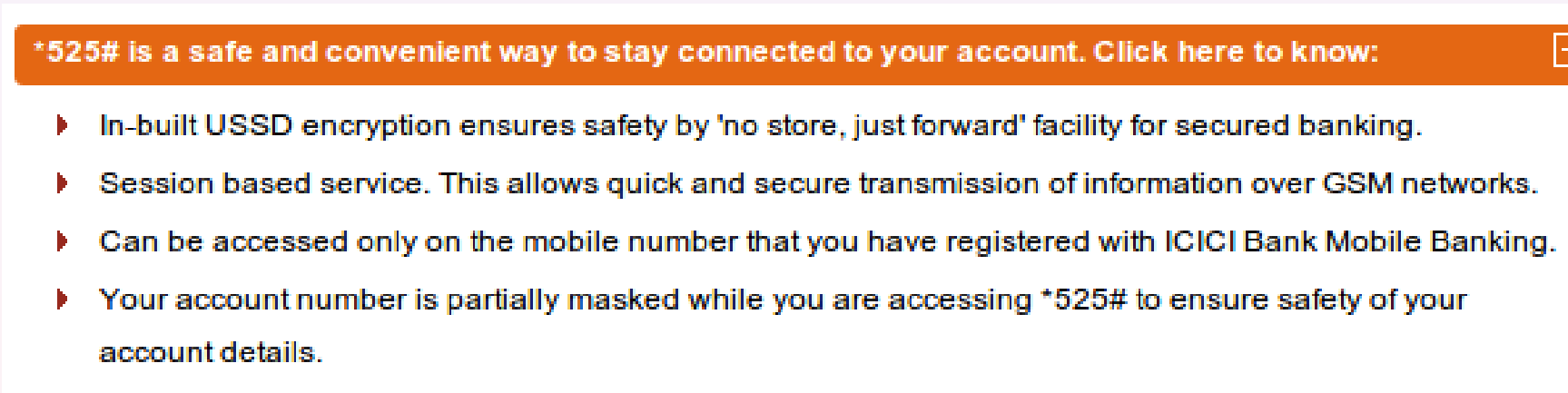
- Register your number to the Bank
- Get user id and MPIN (mobile pin)
- dial ussd code to access your account

Twitter:

- Register for the service by sending SMS (optional)
- dial USSD codes
- type username and password to access

Security in USSD services

- **Completely relies on security provided by cellular network**
- **The biggest bank in India claims:**



- **However in reality.. ☺ ☺**

2. Attacks in USSD network infrastructure

Information needed for an attacker

- USSD codes
- User ID to access the service
- password or MPIN
- tools to access the service on behalf of victim
- weaknesses of the cellular network

Issues in Cellular Networks (GSM)

- No mutual authentication between mobile and base station
- fake base station attacks ☺
- Base station decides when to turn on encryption
- Some networks do not use encryption ☺
- IMSI sent when requested by base station ☺

Phishing attack

Goal: Recover user id, password, MPIN

- set up a fake base station with OpenBSC
- openBSC have basic USSD support
- possible to build bank application
- base station can initiate USSD communication
- collect user ID, password, MPIN
- drawback: attack works in 200m range

Tools to exploit -1

Using a compromised femtocell:

- femtocell: a small access point, connects the mobile phone to the 3G/UMTS network
- blackhat 2012 talk by Nico, Kevin and me
- compromised femtocell can be used for MiTM
- set-up allows to intercept/inject messages
- drawback: attacking range is 50m

$\rightarrow$ **It is difficult for the victim user to recognize this attack**

Tools to exploit -2

Using OsmocomBB phone:

- using a phone supported by OsmocomBB
- the attack depends on the weaknesses in the cellular network
- Nullcon 2011 talk "Your Phone is Your Phone But Your Calls are My Calls" by Akib Sayyed et al.
  $\rightarrow$ authentication bypass
  $\rightarrow$ by using victim's IMSI/TMSI
- the same method can be used for replaying USSD messages

Issues with cellular networks

When mobile sends SMS/USSD message:

4092	290.580003	127.0.0.1	127.0.0.1	LAPDm	U F, func=UA(DTAP) (MM) CM Service Request
4093	290.815906	127.0.0.1	127.0.0.1	LAPDm	I, N(R)=0, N(S)=0(DTAP) (RR) Ciphering Mode Command
4094	290.815977	127.0.0.1	127.0.0.1	LAPDm	S, func=RR, N(R)=1
4095	290.816026	127.0.0.1	127.0.0.1	LAPDm	I, N(R)=1, N(S)=0(DTAP) (RR) Ciphering Mode Complete
4096	290.963265	127.0.0.1	127.0.0.1	LAPDm	U, func=UI(DTAP) (RR) System Information Type 6
4097	291.051085	127.0.0.1	127.0.0.1	LAPDm	U, func=UI
4098	291.257841	127.0.0.1	127.0.0.1	LAPDm	U, func=UI(DTAP) (RR) Measurement Report
4099	291.285992	127.0.0.1	127.0.0.1	LAPDm	S, func=RR, N(R)=1
4100	291.286029	127.0.0.1	127.0.0.1	LAPDm	I, N(R)=1, N(S)=1 (Fragment)
4101	291.434999	127.0.0.1	127.0.0.1	LAPDm	U, func=UI(DTAP) (RR) System Information Type 5
4102	291.522102	127.0.0.1	127.0.0.1	LAPDm	S, func=RR, N(R)=2
4103	291.522192	127.0.0.1	127.0.0.1	LAPDm/GS	I, N(R)=1, N(S)=2(DTAP) (SS) Register (GSM MAP) invoke process

```
▷ Frame 4092: 81 bytes on wire (648 bits), 81 bytes captured (648 bits)
▷ 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: 51792 (51792), Dst Port: gsmtap (4729)
▷ GSM TAP Header, ARFCN: 660 (Downlink), TS: 1, Channel: SDCCH/8 (1)
▷ Link Access Procedure, Channel Dm (LAPDm)
▽ GSM A-I/F DTAP - CM Service Request
  ▷ Protocol Discriminator: Mobility Management messages
    00.. .... = Sequence number: 0
    ..10 0100 = DTAP Mobility Management Message Type: CM Service Request (0x24)
  ▷ Ciphering Key Sequence Number
```

Issues with cellular networks

"*Operators turn off encryption/authentication to reduce load on the base station.*"

```
1143 30.308537000  127.0.0.1  127.0.0.1  LAPDm    U F, func=UA(DTAP) (MM) CM Service Request
1146 30.383750000  127.0.0.1  127.0.0.1  LAPDm    U, func=UI(DTAP) (RR) System Information Type 6
1151 30.544438000  127.0.0.1  127.0.0.1  LAPDm    I, N(R)=0, N(S)=0(DTAP) (MM) CM Service Accept
1152 30.544628000  127.0.0.1  127.0.0.1  LAPDm    S, func=RR, N(R)=1
1153 30.544724000  127.0.0.1  127.0.0.1  LAPDm    I, N(R)=1, N(S)=0 (Fragment)
1158 30.779628000  127.0.0.1  127.0.0.1  LAPDm    U, func=UI
1161 30.849170000  127.0.0.1  127.0.0.1  LAPDm    U, func=UI(DTAP) (RR) Measurement Report
1162 30.855868000  127.0.0.1  127.0.0.1  LAPDm    U, func=UI(DTAP) (RR) System Information Type 5
1167 31.015708000  127.0.0.1  127.0.0.1  LAPDm    S, func=RR, N(R)=1
1168 31.015871000  127.0.0.1  127.0.0.1  LAPDm/GS I, N(R)=1, N(S)=1(DTAP) (SS) Register (GSM MAP) invoke processUnstructu

▷ Frame 1151: 81 bytes on wire (648 bits), 81 bytes captured (648 bits)
▷ 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: 40119 (40119), Dst Port: gsmtap (4729)
▷ GSM TAP Header, ARFCN: 24 (Downlink), TS: 0, Channel: SDCCH/8 (5)
▷ Link Access Procedure, Channel Dm (LAPDm)
▽ GSM A-I/F DTAP - CM Service Accept
  ▽ Protocol Discriminator: Mobility Management messages
      0000 .... = Skip Indicator: 0
      .... 0101 = Protocol discriminator: Mobility Management messages (5)
    00.. .... = Sequence number: 0
    ..10 0001 = DTAP Mobility Management Message Type: CM Service Accept (0x21)
```

3. Attacks on smartphones (Andriod)

USSD on smartphones

USSD (Unstructured Service Supplementary Data):

- all smartphones including feature phones supports USSD as per 3GPP standards.
- technically referred as MMI (Man-Machine Interface) on the mobile device
- MMI commands and format:
  → activation: *SC*SI# ,deactivation: #SC*SI#
  → for more details read TS 122.030
  → Example: * 31 # <called number> SEND
- Codes are executed via "Call Settings" menu option usually

USSD on smartphones

USSD (Unstructured Service Supplementary Data):

- all smartphones including feature phones supports USSD as per 3GPP standard.
- technically referred as MMI (Man-Machine Interface) on the mobile device
- MMI commands and format:
  → activation: *SC*SI# ,deactivation: #SC*SI# (TS 122.030)
  → Example: * 31 # <called number> SEND
- Codes are executed via Call "Menu option" usually

USSD on Android

Vulnerability in Android :

- Dialer in Android
- invoking TEL:123 intent via any Android app put number 123 on the dialer to call
- however, Android dialer fails to differentiate between phone number and USSD codes
- $\rightarrow$ this failure allows to execute USSD codes
- affects versions: ICS, Jelly Bean and older versions too

Let's try some dirty USSD codes ☺ ☺

Affected Devices

Almost every Android device (JellyBean, ICS and older versions too)

- Google Nexsus series
- HTC One series, HTC Sensation
- Samsung Galaxy SI, SII, SIII
- Motorola Driod series
- Sony Ericsson
- other vendors might be (not tested)

SIM attacks

Locking SIM card:

- Every SIM card has PIN code
- however there are only 3 valid attempts SIM
- 3 wrong pins $\rightarrow$ card gets locked and ask PUK code
- PUK code is on smart card

Solution: SIM card works after entering PUK code ..dammm..**less impact** :(

SIM attacks

Killing SIM card:

- Instead of changing PIN code, change PUK code
- 10 wrong PUK code $\rightarrow$ SIM is unusable
- for this attack, it does not matter you set up PIN on SIM card or not

Solution: Go to shop and buy new SIM card. ☺

Dirty codes and methods

USSD Codes:

- **05*1234545*1234*1234# - Change PIN code
- *#06# - Show IMEI number
- *#7780# - factory reset, different for every handset

Method: everybody loves iframes (Reasons?)

Attacking method

1. From a malicious website
   - visiting a link kills your SIM permanently
   - can be invoked via any Android app having permission to call phone
   - attack works in all Android devices

Attacking method

From QR code

- QR Droid (popular barcode scanner app)
  $\rightarrow$ 10,000,000+ downloads in Google Play
- it opens website directly by default
- Not all barcode apps tested
- attack works in all Android devices

Solution: Remove QR Droid from your phone

Attacking method

By sending a WAP Push SMS

- WAP Push SMS (need a special application to send such SMS)
- discovered by c0rnholio @ http://www.silentservices.de/
- thanks Nico (@imnion) for informing
- I extended the above attack with USSD exploit code
- however, this attacks works on Samsung devices only so far

Solution: Turn off "Service Loading" feature

Attacking method

From NFC tag

- few NFC tag readers open URL directly by default
- it was showed earlier but still developers fail to implement basics of security
- works in NFC based Android devices

Wiping out Samsung phones

Samsung tragedy

- there is a USSD code for factory reset settings on Samsung devices
- send a SMS or a link and wipe out the device
- victim can only see the show, cant stop it ;)
- on **Galaxy SIII**, vulnerability can be exploited via **NFC**

Attack can be combined: Kill SIM card and Wipe the phone in 3 sec

Vulnerability Impact

Mobile users:

- Loss of valuable data (if there is no backup)
- disconnects from the cellular network services until getting new SIM
- Financial loss- buy a new SIM card

Network operators and vendors:

- loss in service -> money loss for operators
- issue new SIM cards if affected
- cost of updating

Fixing the vulnerability

- informed to the involved parties
- it has been patched but Android fails always in updating the devices
- issues with Android devices on operator's contract
- update your device

Test your Android device at :
www.isk.kth.se/řbbo/testussd.html

thanks (in no particular order)

- Jean-Pierre Seifert
- Collin Mulliner
- Nico Golde

the end

thank you for your attention

questions?

on tweet : @raviborgaonkar