SMS Fuzzing – SIM Toolkit Attack

Bogdan Alecu

bogdalecu@gmail.com

www.m-sec.net

Abstract

In this paper I will show how to make a phone send an SMS message without the user's consent and how to make the phone not to receive any message. The method used works on any phone, no matter if it's a smartphone or not and also on any GSM/UMTS network. I will present how you can take advantage of sending a special crafted SIM Toolkit command message in order to achieve all that. Finally, I will present the results and their impact on the user and mobile networks security.

1 Introduction

SMS stands for Short Message Service and represents a way of communication via text between mobile phones and/or fixed lines, using a standardized protocol. It is an effective way of communication as the user just writes some text and it's almost instantly delivered to the destination.

SMS as used on modern handsets was originated from radio telegraphy in radio memo pagers using standardized phone protocols and later defined as part of the Global System for Mobile Communications (GSM) series of standards in 1985 as a means of sending messages of up to 160 characters, to and from GSM mobile handsets.[1] Since then a lot of things have changed regarding this service and now it can be used for multiple purposes: MMS – Multimedia Messaging Service, OTA – Over The Air – phone configuration, notification for

[1] http://en.wikipedia.org/wiki/SMS

voice mail, email, fax, micropayments – paying a very small sum of money for different services.

All these ways of using SMS can lead to security issues as their implementation isn't fully tested and more important because SMS is like an opened firewall: every phone has it implemented and the phone always receives the message. There have been discovered different errors, security issues related to the SMS: remote DoS for Nokia S60 phones[2], phone crashing, rebooting, remote executing EXE files, hijacking mobile data connections[3], etc.

Until now most of the SMS related security issues have been found by accident. This is also the case for the current security issue presented in the paper. I was experimenting with the binary message sending – multipart messages: sending the second part but the message had only one part, sending the 10000's part message, etc. and trying to configure the SMSC number stored by sending SIM Application Toolkit messages – when suddenly I've noticed that my phone started to send a message by itself. Later on, after playing more with the message that caused this behavior, my phone was not receiving any other messages. I tried putting the SIM on another phone, resetting the SMSC number but nothing helped.

In this paper I will show how you can achieve the above behavior, why it happens, what are the security implications and how you can protect.

But first, a little bit of theory…

2 SMS

The Point-to-Point Short Message Service (SMS) provides a means of sending messages of limited size to and from GSM mobiles. The provision of SMS makes use of a Service Centre, which acts as a store and forward centre for short messages.

Two different point-to-point services have been defined: mobile originated and mobile terminated. Mobile originated messages will be transported from an MS to a Service Centre (SC). These may be destined for other mobile users, or for subscribers on a fixed network. Mobile terminated messages will be transported from a Service Centre to an MS. These may be input to the Service Centre by other mobile users (via a mobile originated short message) or by a variety of other sources, e.g. speech, telex, or facsimile. The text messages to be transferred contain up to 140 octets.

"An active MS shall be able to receive a short message TPDU - Transfer protocol data unit - (SMS-DELIVER) at any time, independently of whether or not there is a speech or data call in progress. A report will always be returned to the SC; either confirming that the MS has

[2] http://berlin.ccc.de/~tobias/cos/s60-curse-of-silence-advisory.txt

[3] http://www.mseclab.com

received the short message, or informing the SC that it was impossible to deliver the short message TPDU to the MS, including the reason why."[4]

"An active MS shall be able to submit a short message TPDU (SMS-SUBMIT) at any time, independently of whether or not there is a speech or data call in progress. A report will always be returned to the MS; either confirming that the SC has received the short message TPDU, or informing the MS that it was impossible to deliver the short message TPDU to the SC, including the reason why."[5]

2.1 SMS-SUBMIT details

Here are the basic elements for SMS-SUBMIT type:

[4] ETSI TS 100 901 V7.5.0 (2001-12), page 13

[5] ETSI TS 100 901 V7.5.0 (2001-12), page 13

Abbr.	Reference	P1)	P2)	Description
TP-MTI	TP-Message-Type-Indicator	M	2b	Parameter describing the message type.
TP-RD	TP-Reject-Duplicates	M	b	Parameter indicating whether or not the SC shall accept an SMS-SUBMIT for an SM still held in the SC which has the same TP-MR and the same TP-DA as a previously submitted SM from the same OA
TP-VPF	TP-Validity-Period-Format	M	2b	Parameter indicating whether or not the TP-VP field is present.
TP-RP	TP-Reply-Path	M	b	Parameter indicating the request for Reply Path.
TP-UDHI	TP-User-Data-Header-Indicator	O	b	Parameter indicating that the TP-UD field contains a Header.
TP-SRR	TP-Status-Report-Request	O	b	Parameter indicating if the MS is requesting a status report.
TP-MR	TP-Message-Reference	M	I	Parameter identifying the SMS-SUBMIT.
TP-DA	TP-Destination-Address	M	2-12o	Address of the destination SME.
TP-PID	TP-Protocol-Identifier	M	o	Parameter identifying the above layer protocol, if any.
TP-DCS	TP-Data-Coding-Scheme	M	o	Parameter identifying the coding scheme within the TP-User-Data.
TP-VP	TP-Validity-Period	O	o/7o	Parameter identifying the time from where the message is no longer valid.
TP-UDL	TP-User-Data-Length	M	I	Parameter indicating the length of the TP-User-Data field to follow.
TP-UD	TP-User-Data	O	3)	

Table 1 - Basic elements of the SMS-SUBMIT type [6]

1) Provision; Mandatory (M) or Optional (O).

2) Representation; Integer (I), bit (b), 2 bits (2b), Octet (o), 7 octets (7o), 2-12 octets (2-12o)

3) Dependent on the TP-DCS

2.1.1 Example of SMS-SUBMIT

Octet(s)	Description
00	Info about SMSC – here the length is 0, which means that the SMSC stored in the phone should be used.

[6] ETSI TS 100 901 V7.5.0 (2001-12), page 42

01	First octet of the SMS-SUBMIT message. It indicates that there is no reply path, User Data Header, Status Report Request, Validity Period, Reject Duplicates. The message type is SMS-SUBMIT.
00	TP-Message-Reference. The "00" value here lets the phone set the message reference number itself.
0B	Address-Length. Length of phone number (11)
91	Type-of-Address. Here it is the international format of the phone number.
4421436587F9	The phone number in semi octets – 44123456789
00	TP-PID, none specified
00	TP-DCS, none specified
0B	TP-User-Data-Length. Length of message = length of septets = 11
E8329BFD06DDDF723619	TP-User-Data. These octets represent the message "hello world".

Table 2 – Details of how SMS-SUBMIT is composed

In order to send this message trough AT commands via a GSM modem, the following steps should be performed:

a) Set the modem in PDU mode: AT+CMGF=0

b) Check if modem is able to process SMS: AT+CSMS=0

c) Send the message: AT+CMGS=23 > 0001000B914421436587F900000B E8329BFD06DDDF723619

In order to better understand, see below some screenshots from WireShark used for capturing the debug mode of a Nokia 3310.

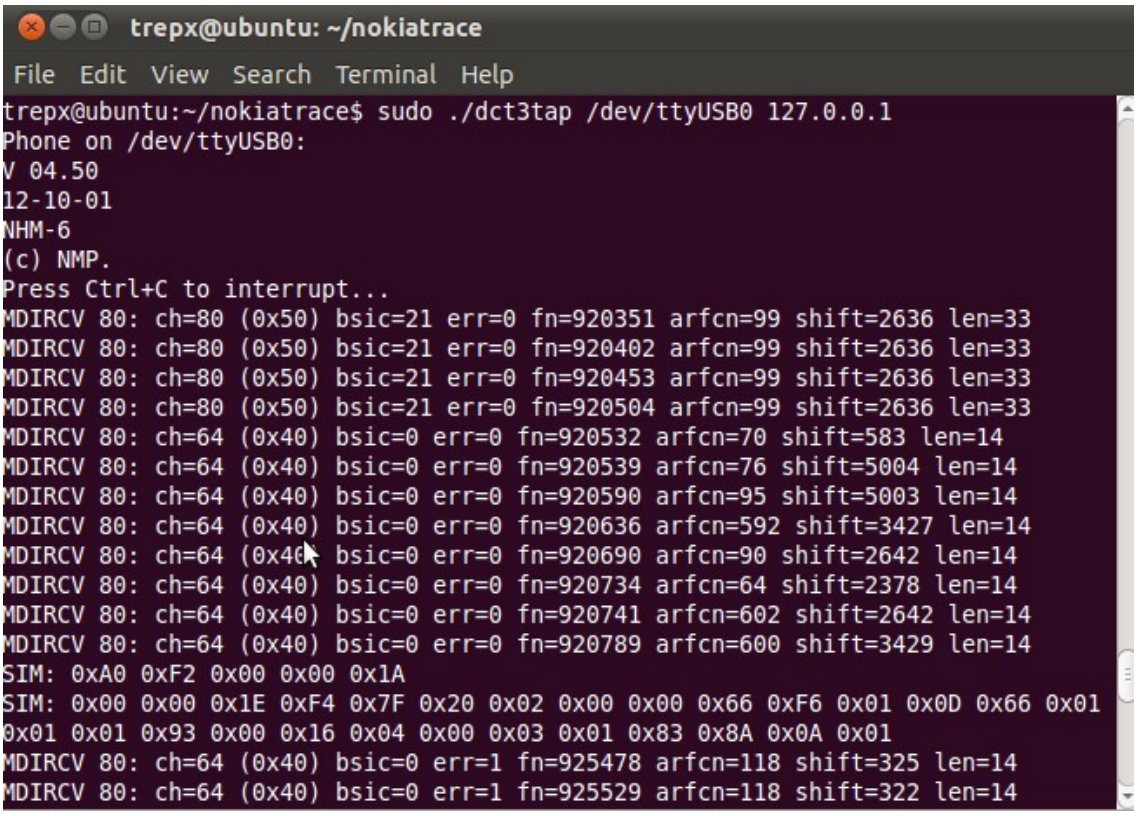
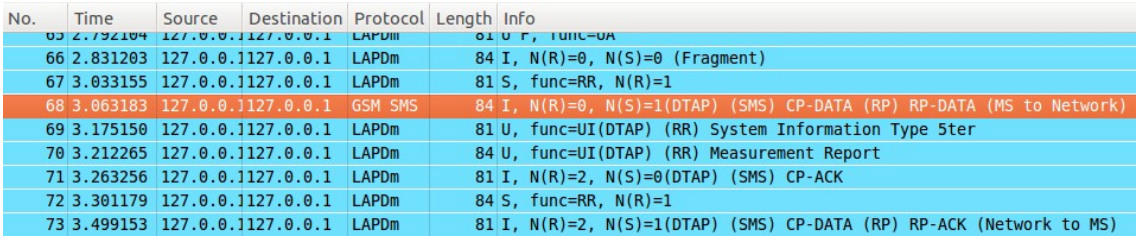


Figure 1 – Capture from dct3tap software [7]



No.	Time	Source	Destination	Protocol	Length	Info
66	2.831203	127.0.0.1	127.0.0.1	LAPDm	84	I, N(R)=0, N(S)=0 (Fragment)
67	3.033155	127.0.0.1	127.0.0.1	LAPDm	81	S, func=RR, N(R)=1
68	3.063183	127.0.0.1	127.0.0.1	GSM SMS	84	I, N(R)=0, N(S)=1(DTAP) (SMS) CP-DATA (RP) RP-DATA (MS to Network)
69	3.175150	127.0.0.1	127.0.0.1	LAPDm	81	U, func=UI(DTAP) (RR) System Information Type 5ter
70	3.212265	127.0.0.1	127.0.0.1	LAPDm	84	U, func=UI(DTAP) (RR) Measurement Report
71	3.263256	127.0.0.1	127.0.0.1	LAPDm	81	I, N(R)=2, N(S)=0(DTAP) (SMS) CP-ACK
72	3.301179	127.0.0.1	127.0.0.1	LAPDm	84	S, func=RR, N(R)=1
73	3.499153	127.0.0.1	127.0.0.1	LAPDm	81	I, N(R)=2, N(S)=1(DTAP) (SMS) CP-DATA (RP) RP-ACK (Network to MS)

Figure 2 – Capture from Wireshark compiled with GSMTAP showing an outgoing SMS

[7] http://bb.osmocom.org/trac/wiki/dct3-gsmtap

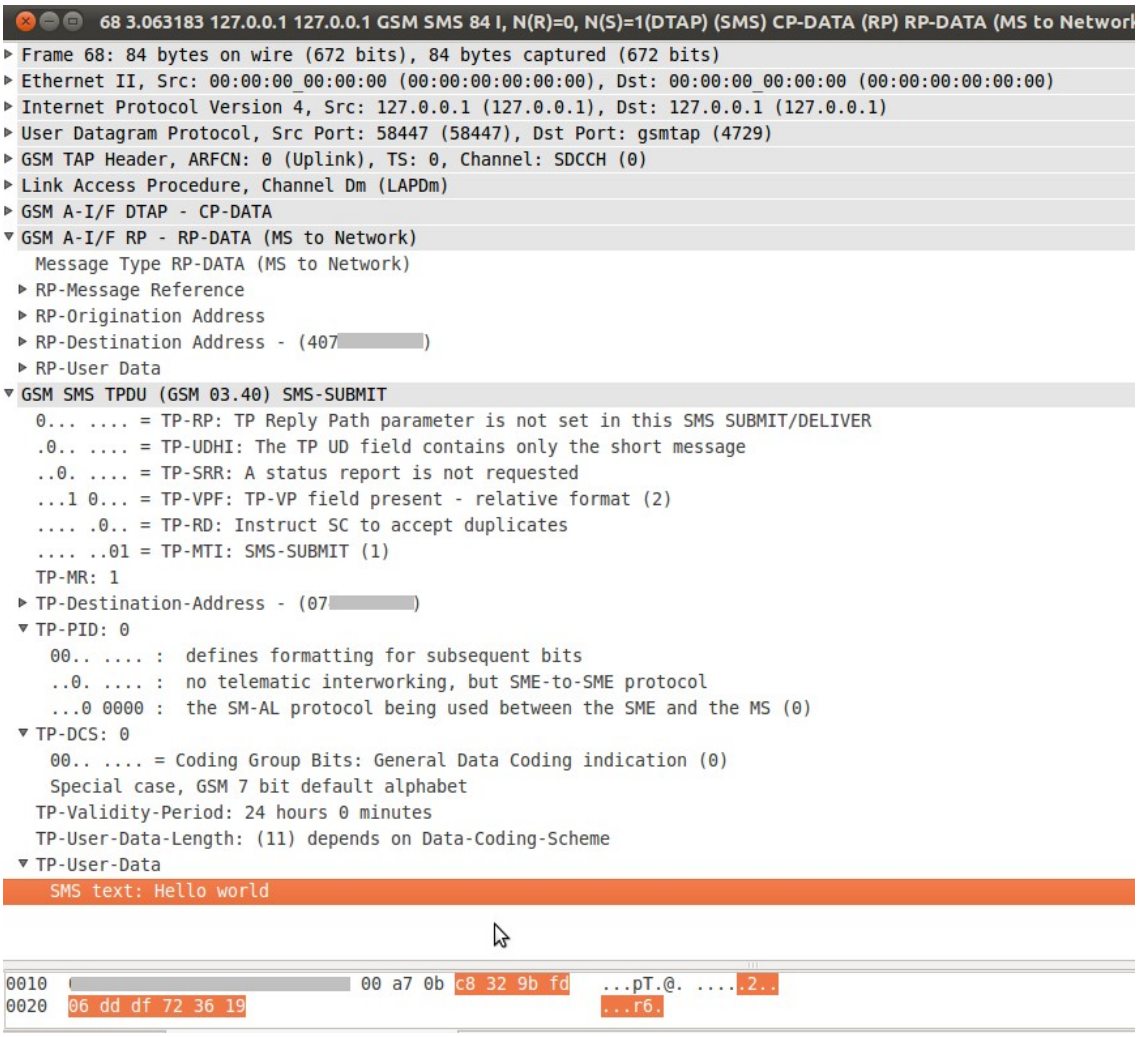


Figure 3 – Capture from Wireshark compiled with GSMTAP showing the SMS-SUBMIT packet

2.2 SMS-DELIVER details

Here are the basic elements for SMS-DELIVER type:

Abbr.	Reference	P[1)]	R[2)]	Description
TP-MTI	TP-Message-Type-Indicator	M	2b	Parameter describing the message type.
TP-MMS	TP-More-Messages-to-Send	M	b	Parameter indicating whether or not there are more messages to send
TP-RP	TP-Reply-Path	M	b	Parameter indicating that Reply Path exists.
TP-UDHI	TP-User-Data-Header-Indicator	O	b	Parameter indicating that the TP-UD field contains a Header
TP-SRI	TP-Status-Report-Indication	O	b	Parameter indicating if the SME has requested a status report.
TP-OA	TP-Originating-Address	M	2-12o	Address of the originating SME.
TP-PID	TP-Protocol-Identifier	M	o	Parameter identifying the above layer protocol, if any.
TP-DCS	TP-Data-Coding-Scheme	M	o	Parameter identifying the coding scheme within the TP-User-Data.
TP-SCTS	TP-Service-Centre-Time-Stamp	M	7o	Parameter identifying time when the SC received the message.
TP-UDL	TP-User-Data-Length	M	I	Parameter indicating the length of the TP-User-Data field to follow.
TP-UD	TP-User-Data	O	3)	

Table 3 - Basic elements of the SMS-DELIVER type [8]

1) Provision; Mandatory (M) or Optional (O)

2) Representation; Integer (I), bit (b), 2 bits (2b), Octet (o), 7 octets (7o), 2-12 octets (2-12o)

3) Dependent on the TP-DCS

In order to better understand how the previous message was received by the phone, I will attach some screenshots from WireShark used for capturing the debug mode of a Nokia 3310.

[8] ETSI TS 100 901 V7.5.0 (2001-12), page 38

69	12.267937	127.0.0.1	127.0.0.1	LAPDm	84 S, func=RR, N(R)=2
70	12.466978	127.0.0.1	127.0.0.1	GSM SMS	81 I, N(R)=0, N(S)=2(DTAP) (SMS) CP-DATA (RP) RP-DATA (Network to MS)
71	12.482979	127.0.0.1	127.0.0.1	GSMTAP	65 GSM SELECT File DF.TELECOM : 7f10
72	12.504096	127.0.0.1	127.0.0.1	LAPDm	84 S, func=RR, N(R)=3
73	12.554978	127.0.0.1	127.0.0.1	GSMTAP	58 [Malformed Packet]
74	12.562972	127.0.0.1	127.0.0.1	GSMTAP	65 GSM SELECT File EF.SMS : Technical problem with no diacnostic
75	12.614917	127.0.0.1	127.0.0.1	LAPDm	81 U, func=UI(DTAP) (RR) System Information Type 6
76	12.628144	127.0.0.1	127.0.0.1	GSMTAP	58 [Malformed Packet]
77	12.661022	127.0.0.1	127.0.0.1	LAPDm	84 U, func=UI(DTAP) (RR) Measurement Report
78	12.711928	127.0.0.1	127.0.0.1	LAPDm	81 U, func=UI
79	12.742908	127.0.0.1	127.0.0.1	LAPDm	84 I, N(R)=3, N(S)=0(DTAP) (SMS) CP-ACK
80	12.858951	127.0.0.1	127.0.0.1	GSMTAP	58 [Malformed Packet]
81	12.938932	127.0.0.1	127.0.0.1	LAPDm	81 S, func=RR, N(R)=1
82	12.976120	127.0.0.1	127.0.0.1	LAPDm	84 I, N(R)=3, N(S)=1(DTAP) (SMS) CP-DATA (RP) RP-ACK (MS to Network)

Figure 4 – Capture Wireshark compiled with GSMTAP showing an incoming message

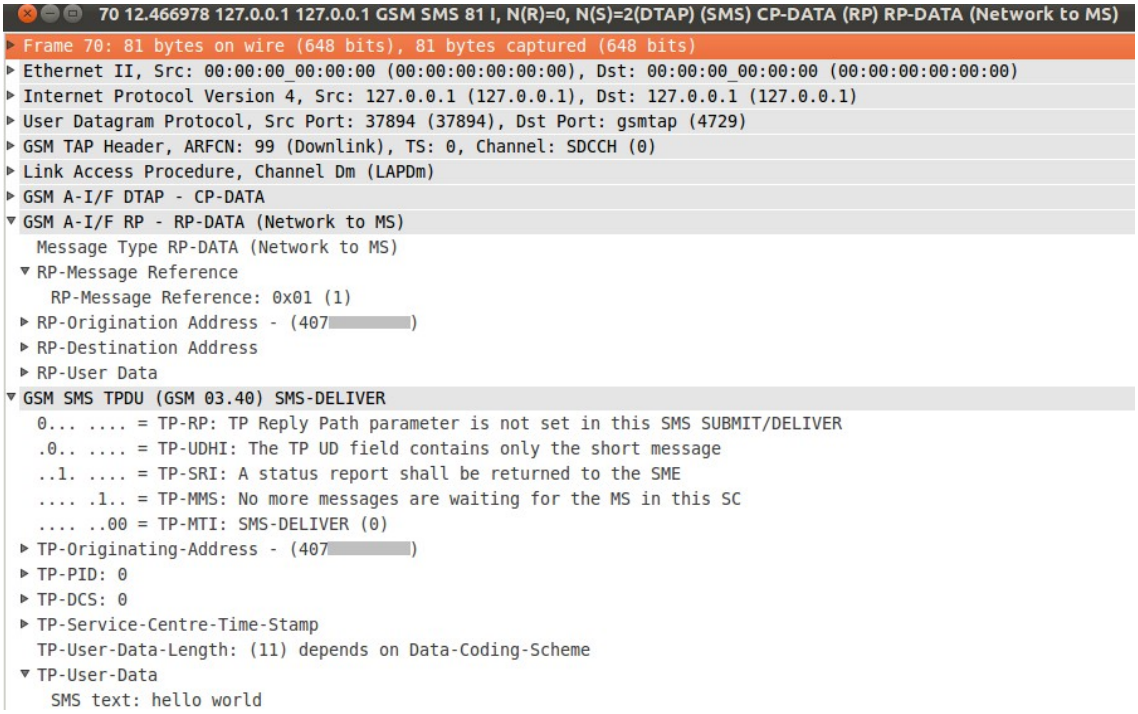


Figure 5 – Capture Wireshark compiled with GSMTAP showing details of SMS-DELIVER packet

2.3 User Data Header (UDH)

The User Data Header contains octets that are added to the beginning of the user data part. UDH provides value added services, creating a smart messaging.

Field	Length
Length of User Data Header	1 octet

Information-Element-Identifier "A" (IEI)	1 octet
Length of Information-Element "A" (IEDL)	1 octet
Information-Element "A" Data (IED)	n octets, based on IEDL

UDH can be used for:

- Ringtone
- WAP Push
- Operator logo
- VCARD
- Concatenation of messages
- SIM Toolkit Security headers

2.3.1 SIM Toolkit Security headers

There are two types of secure commands in the user data:

- Command Packet - a secured packet transmitted by sending entity to the receiving entity, containing secured application message

- Response Packet - secured packet transmitted by receiving entity to the sending entity, containing secured response and possibly application data

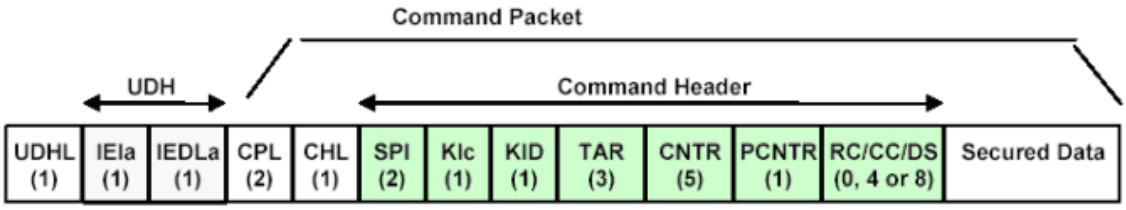


Figure 6 – Structure of the command packet according to GSM 03.48 [9]

Command Packet Length (CPL) - shall indicate the number of octets from and including the Command Header Identifier to the end of the Secured Data, including any padding octets required for ciphering.

Command Header Length (CHL) - the number of octets from and including the SPI to the end of the RC/CC/DS

[9] http://adywicaksono.wordpress.com/2008/05/21/

Security Parameter Indicator (SPI) - defines the security level applied to the input and output message

Ciphering Key Identifier (KIc) - Key and algorithm Identifier for ciphering

Key Identifier (KID) - Key and algorithm Identifier for Redundancy Check (RC) / Cryptographic Checksum (CC) / Digital Signature (DS)

Toolkit Application Reference (TAR) - is part of the 23.048 header that identifies and triggers the Over The Air (OTA) feature, which is an application on the SIM

Counter (CNTR) - Replay detection and Sequence Integrity counter

Padding counter (PCNTR) - indicates the number of padding octets used for ciphering at the end of the secured data

3 About mobile phone

Currently there are two types of phones: feature phones and smartphones. Feature phones run the GSM stack and other applications on a proprietary firmware, with no operating system, by using a single processor – baseband processor. They also have a USB port which is used for connecting to a computer, thus acting as a terminal adapter from which the user sends AT commands. Smartphones have two processors: one is the baseband and the other is the application processor for the applications and the user interface. Each processor has its own memory allocation, no matter if there is a separate memory for each or a shared one.

4 Test case

Like I specified before, this security issue has been discovered by a mistake, when playing with different binary messages. In order to make it easy for me to compose these binary messages, a few tools have been used. Also since I didn't have any hardware available for using the OpenBSC or OpenBTS, I just used the live networks. Since I wanted to keep the spending to minimum, I just chose a pay as you go plan for 5 EUR which has unlimited texting in the same network.

4.1 Tools used

Here is the software and hardware that I used:

- PDUspy – for better understating the incoming message and building my own crafted message (available at http://www.nobbi.com/pduspy.html)

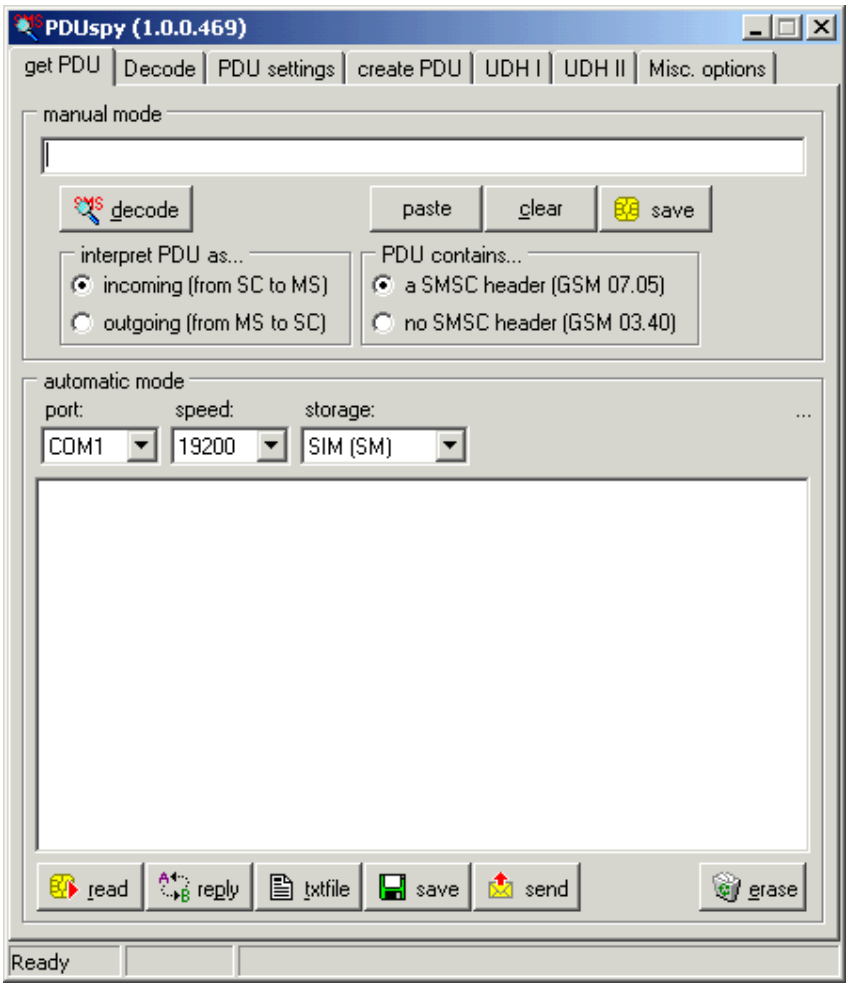


Figure 7 – Overview of PDUSpy

- Nokia 3310 with F-BUS USB cable – I bought the cable on E-Bay

Figure 8 – Nokia 3310 with F-BUS cable attached

- dct3tap command line utility (Linux) to capture the GSM Um and SIM-ME interfaces from a Nokia DCT3 phone (eg. 3310) and forward via GSMTAP to the Wireshark protocol analyzer. This tool has been created by Duncan Salerno and is available on http://bb.osmocom.org/trac/wiki/dct3-gsmtap

- Wireshark development release 1.6.0.rc2 compiled and patched with GSMTAP and SIMCARD in order to decode GSM traffic and SIM access. Instructions on how to patch it can be found at http://bb.osmocom.org/trac/wiki/dct3-gsmtap

- NowSMS Gateway for an easy way of sending messages and connection to an SMS provider by SMPP - http://www.nowsms.com/download-free-trial

Figure 9 – Image from NowSMS showing the available connection types

- Gemalto GemPC Twin reader for accessing the SIM

- SIMinfo Python script for reading the SIM files (available at https://gsm.tsaitgaist.info/doku.php?id=siminfo)

4.2 The attack

I will not give the exact binary message that was sent, but describe why the issue is possible.

First of all, it is important that the SIM to have the service "data download via SMS Point-to-Point allocated and active. Also the SIM must have a SIM Toolkit Application on it

in order to work. Below you will find a table with the results from reading the SIM files with SIMinfo script.

File readed	result
card reader	Gemplus GemPC Twin 00 00
card ATR	3B 9F 95 80 1F C3 80 31 A0 73 BE 21 …
ICCID	89490240001381900000
CVH1	3 tries left (10 to unblock)
CVH2	3 tries left (10 to unblock)
number of CHV/UNBLOCK CHV/ADM	4
CHV1/PIN is disabled	
IMSI	262011910185216
Kc [seq.]	3E104356638C70D0 [2]
PLMN selector (user priority)	
- 222 03	
- 222 06	
- 222 10	
- 211 30	
forbidden PLMN	
- 266 02	
- 222 01	
- 266 07	
- 266 03	
user controlled PLMN	
operator controlled PLMN	
- 222 03	
- 000 22	
phase	2 and PROFILE DOWNLOAD required
SIM service table	
- 1 CHV1 disable function	allocated, activated
- 2 Abbreviated Dialling Numbers (ADN)	allocated, activated
- 3 Fixed Dialling Numbers (FDN)	allocated, activated
- 4 Short Message Storage (SMS)	allocated, activated
- 5 Advice of Charge (AoC)	not allocated, not activated
- 6 Capability Configuration Parameters (CCP)	not allocated, not activated
- 7 PLMN selector	allocated, activated
- 8 RFU	not allocated, not activated
- 9 MSISDN	allocated, activated
- 10 Extension1	not allocated, not activated
- 11 Extension2	not allocated, not activated
- 12 SMS Parameters	allocated, activated
- 13 Last Number Dialled (LND)	allocated, activated
- 14 Cell Broadcast Message Identifier	not allocated, not activated

- 15 Group Identifier Level 1	allocated, activated
- 16 Group Identifier Level 2	allocated, activated
- 17 Service Provider Name	allocated, activated
- 18 Service Dialling Numbers (SDN)	not allocated, not activated
- 19 Extension3	not allocated, not activated
- 20 RFU	not allocated, not activated
- 21 VGCS Group Identifier List (EFVGCS and EFVGCSS)	not allocated, not activated
- 22 VBS Group Identifier List (EFVBS and EFVBSS)	not allocated, not activated
- 23 enhanced Multi-Level Precedence and Pre-emption Service	not allocated, not activated
- 24 Automatic Answer for eMLPP	not allocated, not activated
- 25 Data download via SMS-CB	not allocated, not activated
- 26 Data download via SMS-PP	**allocated, activated**
- 27 Menu selection	allocated, activated
- 28 Call control	not allocated, not activated
- 29 Proactive SIM	allocated, activated
administration data	
MS operation mode	normal operation
OFM (Operational Feature Monitor)	enabled
length of MNC in the IMSI	2

Table 4 – Result of file reading on SIM where Data download via SMS-PP is present

The type of message sent is addressed directly to the SIM, by setting the PID to 0x7F, corresponding to USIM Data Download, as you will see below. Also the DCS has to be a class 2 message type. According to GSM 11.14 here is what happens when these are set[10]:

If the service "data download via SMS Point-to-point" is allocated and activated in the SIM Service Table, then the ME shall follow the procedure below:

- When the ME receives a Short Message with protocol identifier = SIM data download, and data coding scheme = class 2 message, then the ME shall pass the message transparently to the SIM using the ENVELOPE (SMS-PP DOWNLOAD) command.

- The ME shall not display the message, or alert the user of a short message waiting.

In other words, the phone will not display anything and the user will not be aware of this attack.

Let's have a look at the secure command SMS header. One of its components is the **Security Parameter Indicator (SPI).** SPI is 2 octets long and it has the following structure:

[10] ETSI GSM 11.14, December 1996, Version 5.2.0, page 33

First Byte:

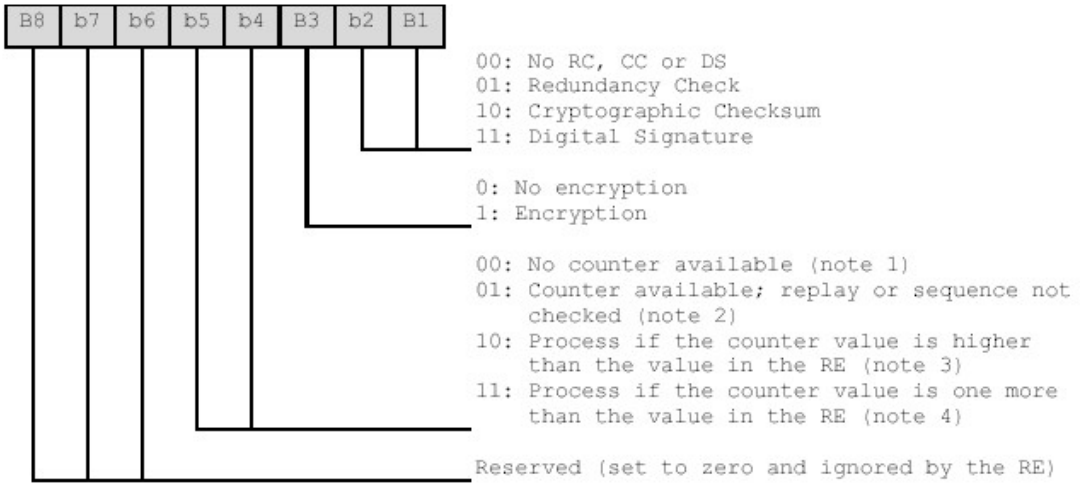


Figure 10 – Coding of the first SPI octet [11]

Second Byte:

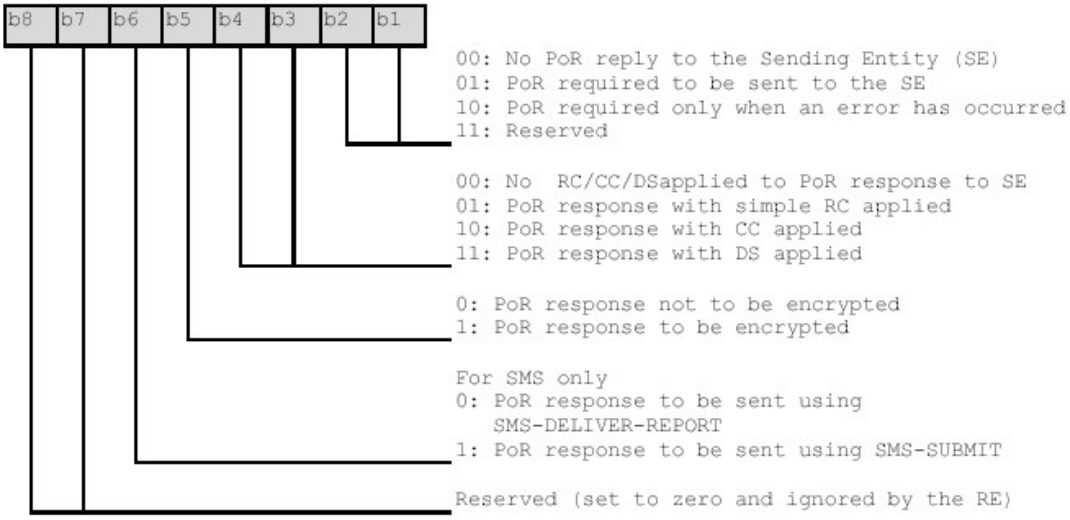


Figure 11 – Coding of the second SPI octet [12]

The vulnerability is possible due to the second byte: here you can set how the proof of receipt (PoR) to be sent – via SMS-DELIVER-REPORT or SMS-SUBMIT. When is set to be on SMS-SUBMIT the phone will try to send back a reply to the originated sender.

If we set it to acknowledge the receipt via DELIVER REPORT, the phone will report to the network the status of the message. Since we don't have valid entries for the KIc, KID,

[11] ETSI TS 101 181 V8.9.0 (2005-06), page 13

[12] ETSI TS 101 181 V8.9.0 (2005-06), page 13

TAR, the result of the STK command is an error so the report will be an error. The sending SMSC then thinks that the phone hasn't received the message and it will try again to send the message, putting on hold any other future messages that are supposed to be delivered, until the initial message expires.

Below you will find a list of images taken from Wireshark indicating how the phone behaves when the special crafted message with PoR set to SMS-SUBMIT is sent.

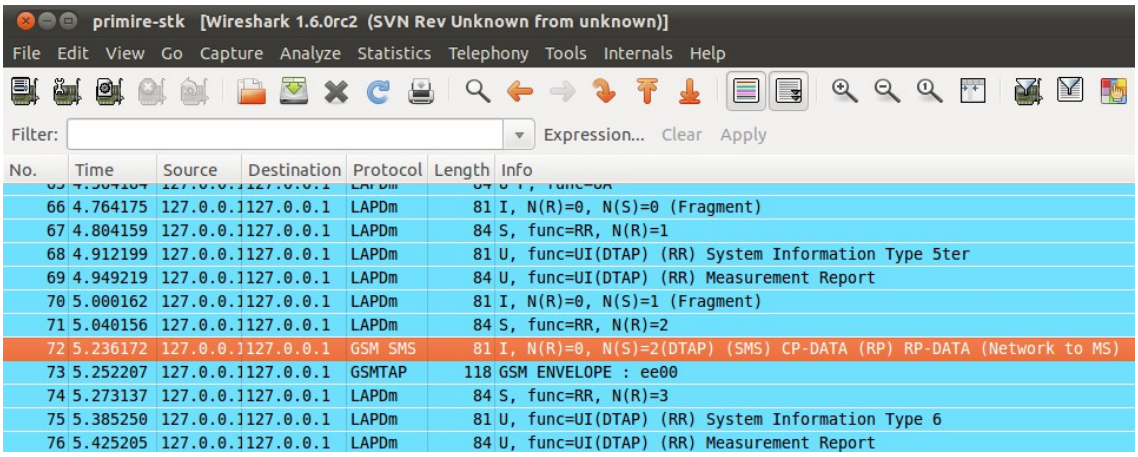


Figure 12 – Capture from Wireshark showing the receipt of the STK message

First, the message is received from the network

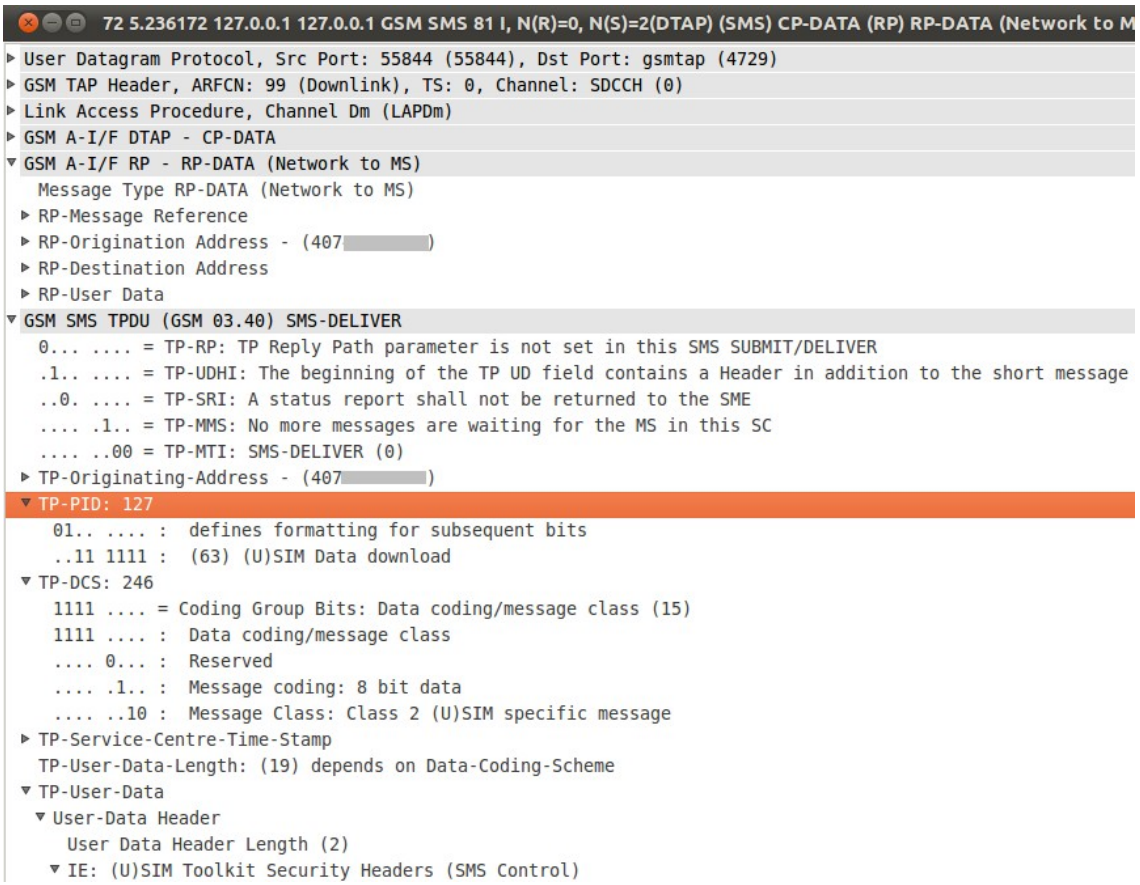
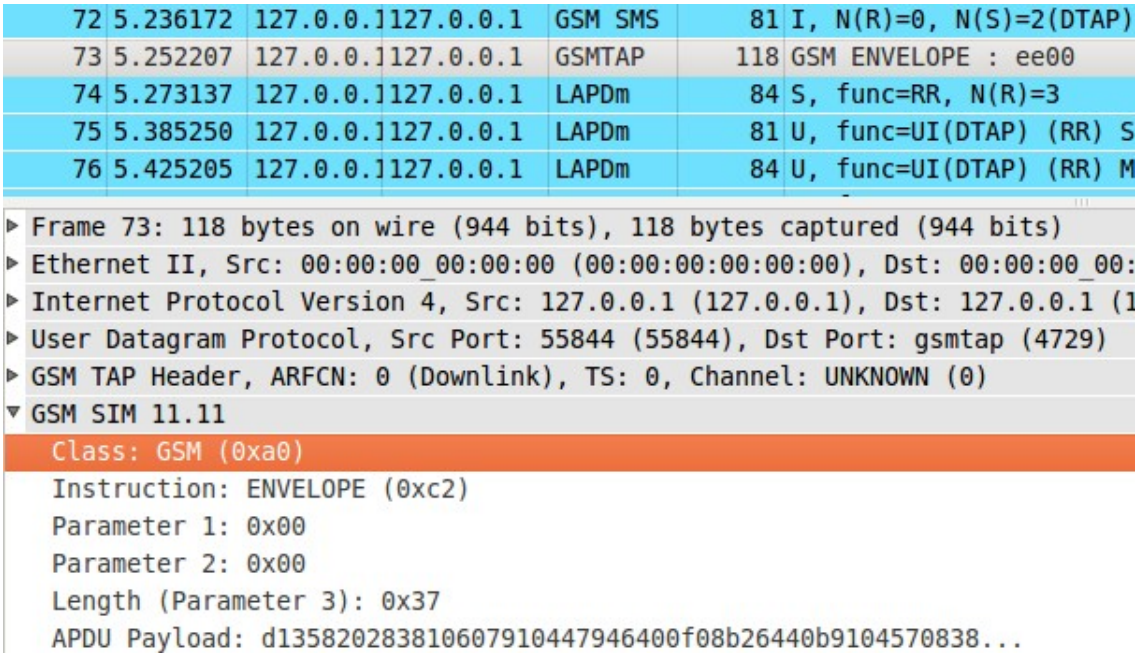


Figure 13 - Capture from Wireshark showing the receipt of the STK message

As you can see, the PID is set to “SIM Data download”, DCS to “SIM specific message” class 2 and the Information Element has the STK headers.



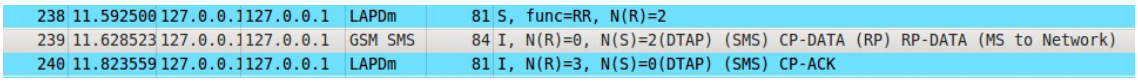


Figure 14 - Capture from Wireshark showing GSM SIM packet

Next the SIM prepares to send a reply message – GSM ENVELOPE packet – and sends it to the network.

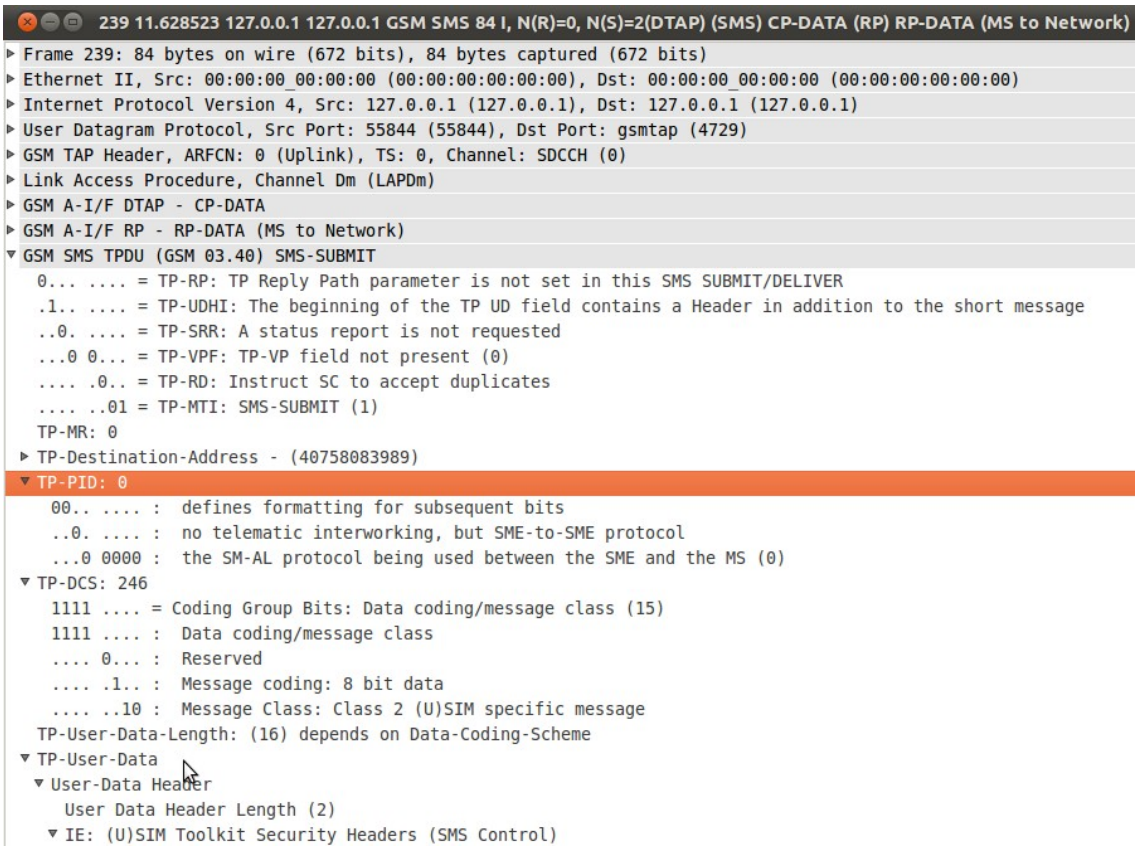
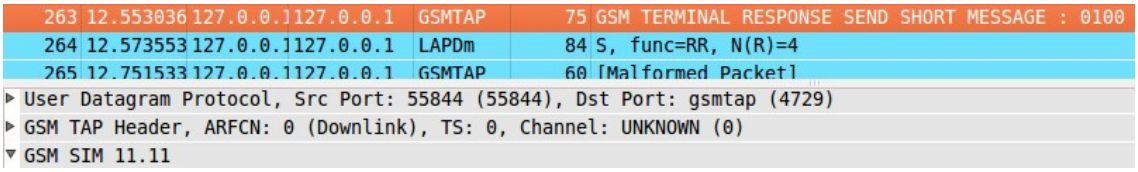


Figure 15 - Capture from Wireshark showing the automated SMS-SUBMIT

Again, note the SIM Toolkit Security Headers.



```
263 12.553036 127.0.0.1 127.0.0.1 GSMTAP 75 GSM TERMINAL RESPONSE SEND SHORT MESSAGE : 0100
▸ Frame 263: 75 bytes on wire (600 bits), 75 bytes captured (600 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: 55844 (55844), Dst Port: gsmtap (4729)
▸ GSM TAP Header, ARFCN: 0 (Downlink), TS: 0, Channel: UNKNOWN (0)
▾ GSM SIM 11.11
    Class: GSM (0xa0)
    Instruction: TERMINAL RESPONSE (0x14)
  ▾ Card Application Tookit ETSI TS 102.223
    ▾ Command details: 011300
        Command Number: 0x01
        Command Type: SEND SHORT MESSAGE (0x13)
        Command Qualifier: 0x00
    ▾ Device identity: 8281
        Source Device ID: Terminal (Card Reader) (0x82)
        Destination Device ID: SIM / USIM / UICC (0x81)
    ▾ Result: 00
        Result: Command performed successfully (0x00)
    Status Word: 0100
```

Figure 16 - Capture from Wireshark showing GSM SIM packet with STK reply

The most important proof: the SMS was sent automatically due to a SIM Toolkit operation and not due to a human intervention.

SIM Application Toolkit provides Value Added Services for the mobile operators. Basically is a set of commands written on the SIM card which helps the card to communicate with the mobile device, making it possible to initiate commands independently of the network or handset.

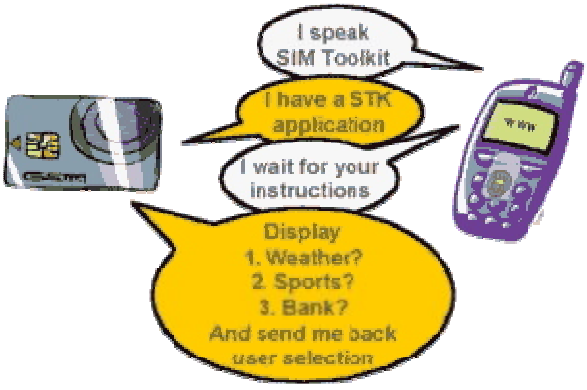


Figure 17 – Communication between the phone and the SIM [13]

Starting with the future phones, all of them are able to communicate via SIM Toolkit.

[13] http://www.gemalto.com/techno/stk/

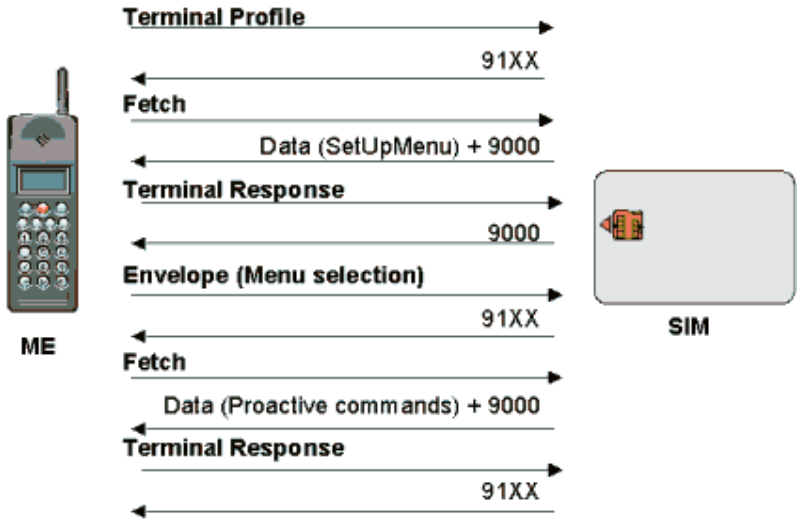


Figure 18 – Communication between the phone and the SIM [14]

One of the best examples of the STK usage is the extra-menu that you see on your phone, from which you can find details about weather, recharge account, your bank account details, etc.

[14] http://www.gemalto.com/techno/stk/

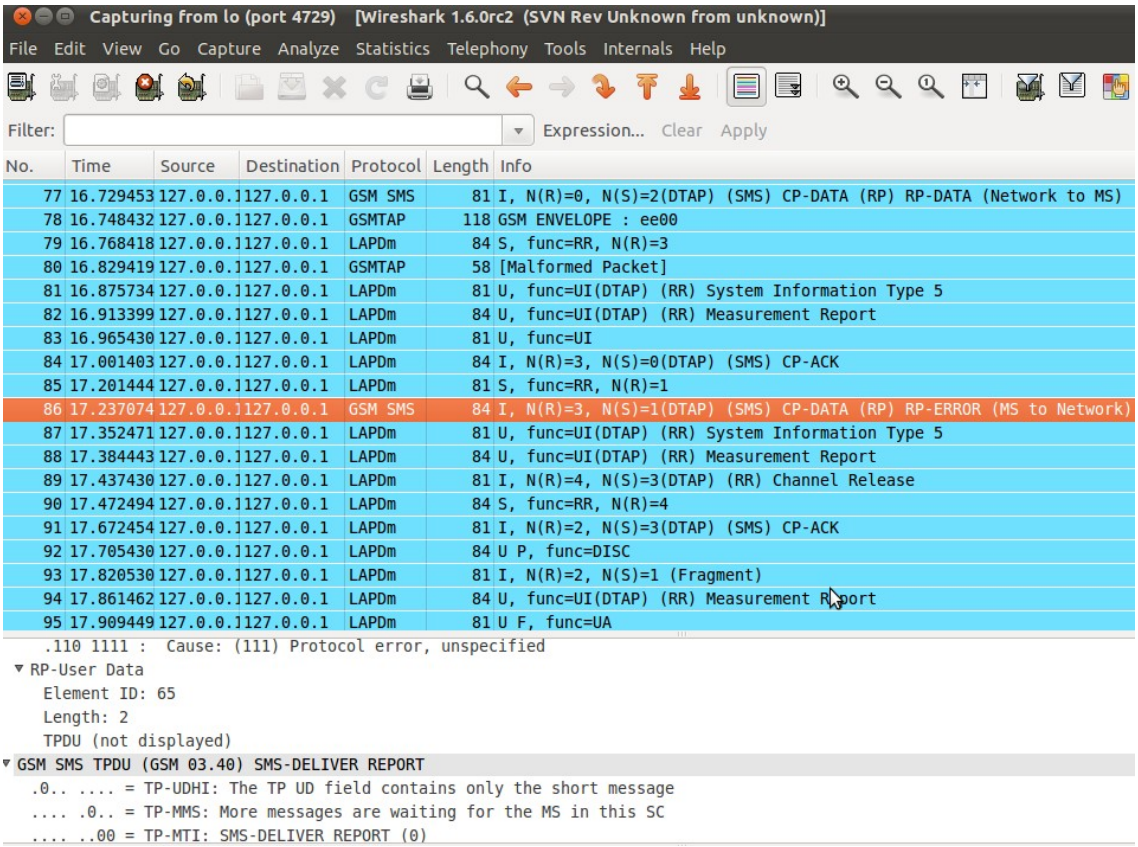


Figure 19 - Capture from Wireshark showing report error

In the above capture you can see what happens when the messages comes with the SPI byte having the POR set to DELIVER REPORT. First the phone receives the message (Network to MS) and reports to the network that there was an issue (CP-DATA (RP) RP-ERROR). Further the network believes that the phone was not able to properly receive the message and it will try over and over to send it, until it has expired.

5 Results and impact

As stated before, by sending this crafted STK message we can make the phone send automatically a reply to the originated sender. Another behavior is that the phone will not be able to receive any messages until the malformed one expires.

I first discovered the vulnerability somewhere in June 2010 and worked on better understanding it. Meanwhile on August 26 2010 I have reported the vulnerability to CERT (Computer Emergency Response Team) and they have assigned a CVE but it was not published yet. Details about this will be published on http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=2010-3612. At the time of this writing I am still waiting for a reply from CERT regarding the date they will disclose it.

An important thing to note is that the issue only works for the SIM cards that have SIM Application Toolkit (STK) / USIM Application Toolkit (USAT) implemented. Now more and

more operators have these kinds of cards due to the continuous development of the mobile banking, so it won't be so hard for an attacker to succeed. **However, even with this limitation, the attack works no matter the phone you use or the mobile network – GSM / UMTS.** Tests have been made on different feature phones and smartphone and they all succeeded.

Another thing that should be taken into consideration is that the type of message is addressed directly to the SIM. This means that the operator should allow the users to send this type of SMS, which normally only the mobile operator would send. In practice not many networks firewall the SMS, making it the perfect environment for an attacker since the SMS is always on. However, even if the operator would filter such messages, it would be possible to send the SMS if you have access to a SMSC. This brings us to a developed method of attack.

When sending the message between different networks or the same network it doesn't have such a great financial impact. First of all, if you send the message to someone registered to the same operator as yours, most probably that person won't be charged extra due to the billing plan which most of the time offers you some messages within the network. If you send to someone registered on a different network, even internationally, it will cost you too much. The question is how can we create a considerable financial impact? The answer is to use a SMSC provider.

There are plenty such providers over the Internet to which you can connect in different ways: by email, HTTP, SMPP. The issue is that not all forward correctly the APDU packets to all or some mobile operators. During my research, after contacting about 10 different providers I found two that correctly forwarded the messages to the destination networks tested. It will take you some time, but in the end you'll find for sure one that works. Most of these providers allow you to change the sender number (address) to either numeric or alphanumeric and they have really good rates for SMS – somewhere in the low euro cent area – and even cheaper than the rates you have on your own operator.

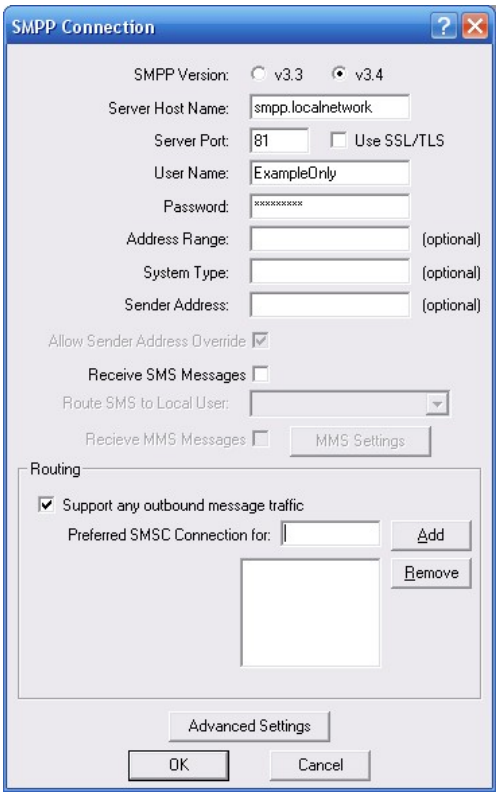


Figure 20 – Capture from NowSMS showing the options from SMPP connection

Given all these, now we can send a spoofed binary message coming from Thailand. Since the command message sends the reply to the originated number, it will send the reply in Thailand, thus paying much more and making the user get a pretty expensive bill. Even so, the attacker won't gain any money. To overcome this, the attacker can easily get a premium rate number which will charge 10 EUR per message received. All that is left to do is to spoof the sender with the premium rate number he got. Now for only one message sent – which is a few cents – he gains 10 times more. This has a great impact on the users and it's not much they can do.

How to protect from such attacks? There are a few ways:

- Mobile operators could filter command messages that are not coming from themselves. Even if it's a network protection, users not being protected if not all of the operators implement such security or in case someone comes with their own built network like OpenBTS / OpenBSC, I still consider it would be the best protection especially if we think about the premium rate numbers attack.

- Some mobile devices have the option to ask the user about SIM actions. If the option is set, when the phone will try to send the message it will ask to allow this. See some images about this configuration and how the phone behaves.

- Use a SIM card that has the service "data download via SMS Point-to-point" deactivated or one that doesn't have any Toolkit Application on it.

➢ Use a Nokia DCT3 phone and stay always connected with the F-BUS cable and Wireshark opened (hard to make it always).

However it's hard to answer such an ambiguous question as we don't have any information about the type of message, the content, the destination. Also some SIM cards are configured to automatically send a message to the network operator in order to receive settings for MMS and Internet access when they are put in a new device. Most probably the users will allow the phone to send the message.

References

ETSI TS 100 901 V7.5.0 (2001-12), page 13

ETSI TS 100 901 V7.5.0 (2001-12), page 38

ETSI GSM 11.14, December 1996, Version 5.2.0, page 33

ETSI TS 100 901 V7.5.0 (2001-12), page 42

ETSI TS 101 181 V8.9.0 (2005-06), page 13

ETSI TS 101 181 V8.9.0 (2005-06), page 13

http://adywicaksono.wordpress.com/2008/05/21/

http://bb.osmocom.org/trac/wiki/dct3-gsmtap

http://berlin.ccc.de/~tobias/cos/s60-curse-of-silence-advisory.txt

http://en.wikipedia.org/wiki/SMS

http://www.gemalto.com/techno/stk/

http://www.mseclab.com

http://www.nobbi.com/pduspy.html

https://gsm.tsaitgaist.info/doku.php?id=siminfo