



Codenomicon whitepaper:
SMS Fuzzing

- Miia Vuontisjärvi and Tero Rontti -



- 1** Introduction
- 2** SMS vulnerabilities
- 3** Fuzz testing background
- 4** Codenomicon SMS test tools

CODENOMICON Ltd. | info@codenomicon.com | www.codenomicon.com

Tutkijantie 4E | FIN-90590 OULU | FINLAND | +358 424 7431
12930 Saratoga Avenue, Suite B-1 | Saratoga, CA 95070 | UNITED STATES | +1 408-414-7650

PREEMPTIVE SECURITY AND ROBUSTNESS TESTING SOLUTIONS

1 Introduction

Short message service (SMS) is a communication service that allows sending text messages from one mobile phone to another or between mobile phones and other short message entities such as mobile banking, e-mail gateway or notification applications. Mobile originated messages are transported from a mobile phone to a Short Message Service Center (SMSC). They may be destined for other mobile users, or for subscribers on a fixed network. Mobile terminated messages are transported from a SMSC to a mobile. These messages can be input to SMSC by other mobile users (via a mobile originated short message) or by a variety of other sources, e.g. speech, telex, or facsimile. Since its deployment in 1993, SMS has become widely used, with more than 6.1 trillion text messages sent and received in 2010.

Lately, there have been discussions about SMS security and the possibility of exploits. The fact is, SMS makes an ideal attack vector. The SMS feature is always on: practically every cell phone supports SMS, and it cannot be turned off. When mobile is connected to the network, it can send and receive text messages. Also, SMS is used for a variety of services, such as vcard, which adds to the complexity of the service. Complex services are more likely to contain exploitable vulnerabilities. These two traits combined with the number of users surely give a reason for concern and should motivate security testing of SMS.

Fuzzing is a robustness testing method that helps finding vulnerabilities proactively, before they are exploited or cause robustness problems. It can be used for finding SMS vulnerabilities both from mobile phones and network elements.

2 SMS Vulnerabilities

SMS, like practically all software, contains vulnerabilities. Vulnerability is a coding mistake, a bug in a software that may cause problems in the system operation and offer a point of entry for an outside attacker. The most famous SMS vulnerability today is undoubtedly the iPhone SMS vulnerability discovered by researchers Colin Mulliner and Charlie Miller in 2009, but it would be naive to think that it was one of its kind.

In case SMS vulnerability is triggered, what are the consequences?

From a SMS terminal point of view, terminal has to display or act based on the received SM. This means that parsing and handling is required. A malformed message may cause a Denial-of-service condition in terminal: Reboot, busy loop, flash corruption. Running exploit code in the terminal allows the attacker to take over the terminal and install malware, make calls, send spam, basically do anything the phone's owner can do. Also, if SMS protocol PDU handling is performed at chipset level, it may result in a failure in chipset that is hard, even impossible to fix after deployment.

Apart from the terminal, other networks elements may also contain vulnerabilities. Particularly challenging is handling the text messages traveling from TCP/IP networks to mobile networks and vice versa. As the two networks do not speak the same language, there is a lot of logic involved, and sometimes reorganizing or recoding some of the data is necessary. There are several possibilities for vulnerabilities which degrade the service. Messages delivered over short message peer-to-peer (SMPP) protocol have to be reassembled into SMS PDUs (Deliver, Submit) before transmission to final endpoint. SMSC may perform firewalling for SMS. Error conditions include crash, increased CPU load, and running exploit code, among others.

3 Fuzz Testing Background

Vulnerabilities enable exploits and cause service disruptions, and proactive security testing aims to find and fix them before they cause problems. Fuzz testing has proved to be an excellent method for finding vulnerabilities in software.

In fuzzing, unexpected data in the form of modified protocol messages and message sequences are fed to the system under test. When creating fuzz test tools, at first valid messages are studied. Then, the valid message structure is slightly altered to create invalid messages that target the areas that are most likely to contain vulnerabilities. Resulting anomalous message sequences are used as an input to test the system robustness, and the behavior of the system is monitored with various instrumentation techniques to detect failures. Abnormal response to an invalid message indicates that there is a vulnerability in the system, and that the vulnerability may be exploitable.

Here are some examples of SMS message structures that can be anomalized.

- **SMS User Data Header.** In addition to delivering text contents, the message payload (User Data) may contain binary header (User Data Header) used for different purposes: Binary data delivery, addressing to deliver application data, and message concatenation indicators.
- **SM-DATA.** Binary data can also be present in User Data SM-DATA field (Short Message Data). Typically when binary data is delivered in SM-DATA, also port number IEs (Information Elements) are used in User Data Header. In SM-DATA field, textual data and different payload types (smart messages or enhanced messaging for example) can be anomalized.

- **Message concatenation.**

Short message is, by definition, short: it allows 140 octet contents per message, which means up to 160 characters per message, depending on character set. Concatenated messages can be longer than 160 characters. A concatenated message is divided into parts which are sent as individual short messages. Each part has a user data header that contains a concatenation indicator. The indicator consists of a common identifier, sequence number of the part and total number of parts for the message. Once the destination has received all parts of the message, the message is reconstructed based on the sequence numbers.

Up to 255 parts may be delivered, giving a technical limit of around 32kB of data delivered in a concatenated message. The actual limit depends on the number of user data headers present in messages: Each UDH element consumes some space from the payload data.

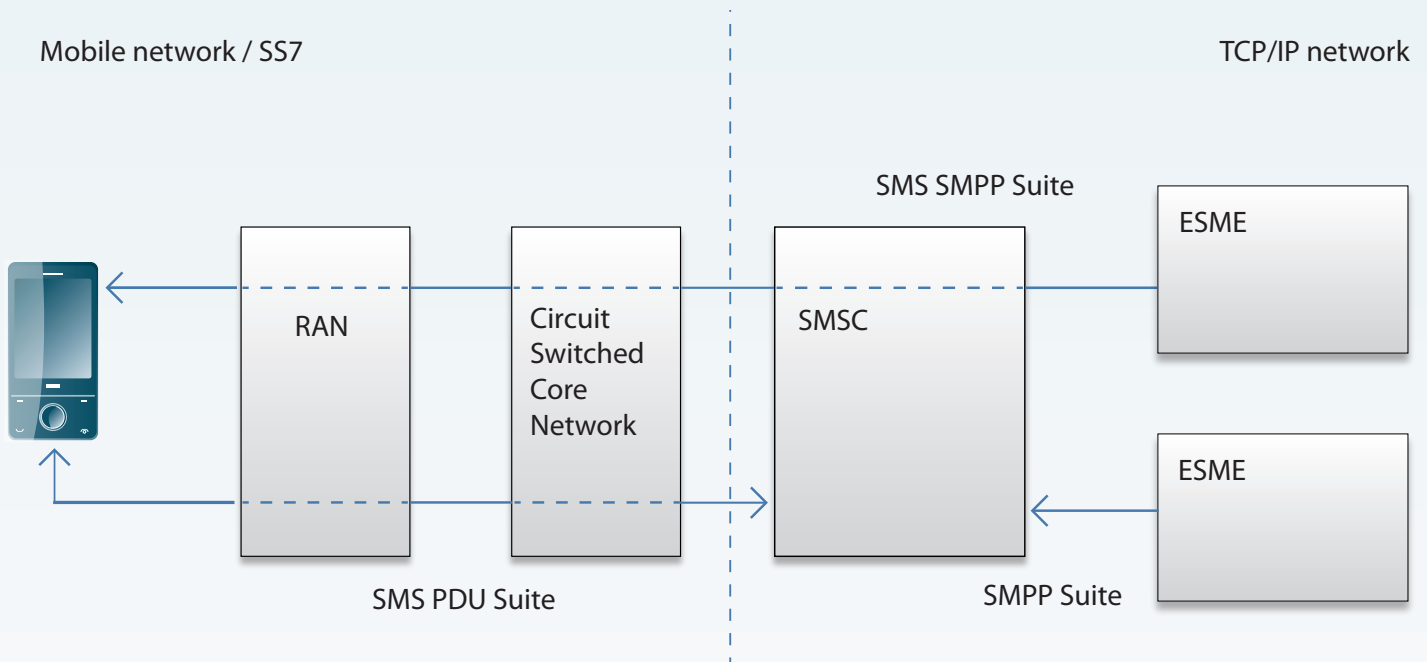
- **SMS PDU structure.** There are two ways of sending and receiving SMS messages: by text mode and by protocol description unit (PDU) mode.

SMS PDU Deliver and Submit message headers contain addressing and other delivery information (such as time to live at SMSC for delivery, if delivery notification is requested, and so on). User Data contains the actual payload, and this data can be delivered also in SMPP messages.

4 Codenomicon SMS Test Tools

Codenomicon Defensics offers three test suites for testing SMS.

- SMPP test suite for testing the SMPP protocol used for delivering short messages from services to SMS gateways and vice versa
- SMS over SMPP test suite for testing the delivery of SMS user data over SMPP
- SMS PDU test suite, SMS Submit and Deliver PDUs and SMS User Data file based injection



RAN = Radio Access Network
 SMSC = Short Message Service Center
 ESME = External Short Message Entity
 PDU = Protocol Description Unit
 SMPP = Short Message Peer to Peer

Short message peer-to-peer test suites

The Short Message Peer to Peer (SMPP) protocol is an open, industry standard protocol designed to provide a flexible data communications interface for transfer of short message data between a Message Center, such as a Short Message Service Centre (SMSC), GSM Unstructured Supplementary Services Data (USSD) Server or other type of Message Center and a SMS application system, such as a WAP Proxy Server, EMail Gateway or other Messaging Gateway.

SMPP test suite anomalizes SMPP protocol elements and dialogs, and tests the SMSC robustness. The test suite does not contain extensive tests for SMS user data delivered, for that purpose SMS over SMPP suite is used.

SMS over SMPP test suite anomalizes the structure of Short Messages delivered over Short Message Peer to Peer (SMPP) protocol to test the robustness of all the network elements that handle the anomalized message, including the terminal.

SMS Protocol Description Unit test suite

SMS PDU test suite generates test cases for SMS testing. SMS PDUs (both mobile originated and SC originated PDUs) as well as raw SMS user data elements may be generated as files or delivered by a defined TCP interface to the BTS. The suite tests both terminals and network elements handling the anomalized PDU.