New Security Threats Caused by IMS-based SMS Service in 4G LTE Networks

Guan-Hua Tu*
Michigan State University,
East Lansing, MI, USA
ghtu.msu@gmail.com

Chi-Yu Li*
National Chiao Tung University,
Hsinchu City, Taiwan
chiyuli@cs.nctu.edu.tw

Chunyi Peng
Ohio State University
Columbus, OH, USA
chunyi@cse.ohio-state.edu

Yuanjie Li
University of California,
Los Angeles, CA, USA
yuanjie.li@cs.ucla.edu

Songwu Lu
University of California,
Los Angeles, CA, USA
slu@cs.ucla.edu

ABSTRACT

SMS (Short Messaging Service) is a text messaging service for mobile users to exchange short text messages. It is also widely used to provide SMS-powered services (e.g., mobile banking). With the rapid deployment of all-IP 4G mobile networks, the underlying technology of SMS evolves from the legacy circuit-switched network to the IMS (IP Multimedia Subsystem) system over packet-switched network. In this work, we study the insecurity of the IMS-based SMS. We uncover its security vulnerabilities and exploit them to devise four SMS attacks: silent SMS abuse, SMS spoofing, SMS client DoS, and SMS spamming. We further discover that those SMS threats can propagate towards SMS-powered services, thereby leading to three malicious attacks: social network account hijacking, unauthorized donation, and unauthorized subscription. Our analysis reveals that the problems stem from the loose security regulations among mobile phones, carrier networks, and SMS-powered services. We finally propose remedies to the identified security issues.

Keywords

Mobile networks; LTE; IMS; SMS; attack; defense

1. INTRODUCTION

SMS (Short Message Service) is one of the fundamental services in mobile networks. It is supported by almost all cellular-connected mobile devices (7.4 billion devices in 2014). It is not only used for interpersonal communications, but also employed by SMS-powered services, which empower companies to reach or/and authenticate their customers via SMS. They have been used by various types of industries, such as social network (*e.g.*, Facebook, Twitter), grocery (*e.g.*, Walmart), airline (*e.g.*, American Airline), bank (*e.g.*, Chase), apparel (*e.g.*, A&F), courier (*e.g.*, Fedex, UPS) and instant messaging application (*e.g.*, Whatsapp), to name a few.

The success of the SMS-based approach stems from two reasons. First, the delivery of SMS messages within mobile networks protects confidentiality and integrity [21]. Though it has some security issues (*e.g.*, the unauthorized SMS messages sent by the phone-side malware or the spoofed SMS messages sent from the Internet), thanks to the efforts of research community and industry [8, 9, 15, 37, 43, 48, 49], they are well addressed in the 2G/3G networks, at least for the top four largest US carriers. Second, SMS, a fundamental service of mobile phones, is the most convenient way for service providers to reach billions of mobile users.

Since the 4G LTE network supports only packet-switched (PS) domain, the services of the conventional circuit-switched (CS) domain shall be migrated to the IP Multimedia Subsystem (IMS) [1] over the PS domain. The PS domain is used for the data plane of mobile networks, whereas the CS domain is mainly for the signaling messages on the control plane. Thus, the underlying technology of SMS has to shift from the control-plane CS to the data-plane IMS. At this point, the natural question is: *given the dramatic change of the SMS design in the 4G network, are mobile phones, carriers' SMS infrastructures and SMS-powered services as secure as usual?*

Unfortunately, our study yields a negative answer. Our results show that all of those three parties may suffer from the attacks caused by the change of underlying security semantics. In particular, 28 out of the 40 SMS-powered services (summarized in Table 2) which we choose from some big companies of various industries, are vulnerable to SMS security threats. Due to space limit and similarity, we do not list other studied SMS-powered services from USPS, Dollar Tree, ZipCar, Weather.com, *etc.*

Specifically, we devise four attacks based on the vulnerabilities of the IMS-based SMS: silent SMS abuse, SMS spoofing, SMS client DoS, and SMS spamming towards IMS. The victims of the first three attacks are mobile users, whereas those of the last attack are carriers. We further discover that those vulnerabilities can be exploited to launch three major attacks against SMS-powered services: social network account hijacking, unauthorized donation, and unauthorized subscription. Table 1 summarizes our findings. Note that our presumed attack model is relatively simple: the attacker uses only commodity smartphones and has no control of carrier networks. Moreover, we evaluate those attacks in a responsible and controlled manner (*i.e.*, the victims are only the participants of this project).

*The first two authors equally contribute to this work.

CCS'16, October 24-28, 2016, Vienna, Austria

DOI: http://dx.doi.org/10.1145/2976749.2978393

Category	Attack	Victim	Description and Threat	Main Vulnerability
IMS-based SMS (§4)	Silent SMS abuse	Mobile user	Adversary exploits the malware on the victim phone to abuse SMS silently.	V1: Leakage of SIP Session Information (§4.1.1) V2: Injection of Forged SIP Messages (§4.1.2) V3: Insufficient SMS Access Defense at Phone (§4.2.1)
	SMS client DoS	Mobile user	Adversary exploits the malware on the victim phone to exhaust the SMS client's resources.	
	SMS spamming towards IMS	Carrier	Adversary sends spam SMS messages to the IMS system.	
	SMS spoofing	Mobile user	Adversary sends SMS messages on behalf of the victim without his/her awareness.	V1, V2, V3 V4: Spoofable SMS Messages at IMS Server (§4.2.2)
SMS-powered services (§5)	Account hijacking	Mobile user	Adversary hijacks the victim's Facebook account and abuse it.	○ Phones & Carriers: V1, V2, V3 and V4 ○ Facebook: No runtime authentication (§5.1)
	Unauthorized donation	Mobile user	Adversary donates money to Red Cross from the victim's bill.	○ Phones & Carriers: V1, V2, V3 and V4 ○ Red Cross: weak authorization (§5.2)
	Unauthorized subscription	Service provider	Adversary makes mobile users subscribe to one service, the provider of which may receive the users' complaints of unauthorized subscription.	○ Phones & Carriers: V1, V2, V3 and V4 ○ Home Depot: weak authorization (§5.3)

Table 1: Summary of our main findings on IMS-based SMS vulnerabilities and proof-of-concept attacks.

The identified attacks root in the security vulnerabilities spanning mobile phones, carrier networks, and SMS-powered services. On the mobile phones, the SMS-related security mechanisms (*e.g.*, SMS permission control) remain invariant while the SMS technology evolves; thus, they are easily bypassed. For the carrier networks, though the standards provide several SMS security options, the flexibility of which helps carriers accommodate the diversified service demands of mobile users, they may expose both carriers and mobile users to the serious security threats. The SMS-powered service providers still rely on the existing defense, which is used to be against the legacy SMS threats. Therefore, the unprecedented SMS threats introduced by the IMS-based SMS may hurt the SMS-powered services.

In summary, we study the insecurity of the IMS-based SMS by systematically exploring all the parties involved: mobile phones, carrier networks, and SMS-powered services. The paper makes three major contributions.

1. We identify four vulnerabilities of the IMS-based SMS on mobile phones and the IMS system. They come from the security issues of its fundamental designs (*i.e.*, software-based client, flexible protocol, data-plane communication channel), and the security mechanisms stipulated by the standards.

2. We devise proof-of-concept attacks against mobile users, carriers, and SMS-powered service providers, by exploiting the identified vulnerabilities. We assess their impact in two major US carriers.

3. We point out root causes and propose recommended solutions. The lessons we learned not only help secure the global deployment of IMS-based SMS, but also benefit the mobile industry.

The rest of this paper is structured as follows. §2 introduces the background of SMS. §3 describes the potential security issues of IMS-based SMS, as well as threat model and methodology. In §4, we present four security vulnerabilities of IMS-based SMS, and sketch four proof-of-concept attacks. We then devise three major attacks against SMS-powered services in §5. We propose solutions and discuss several remaining issues in §6 and §7, respectively. §8 presents related work, and §9 concludes the paper.

2. BACKGROUND

Short Message Service (SMS) is a text messaging service for mobile users. Its underlying technology advances with the deployment of the IP Multimedia Subsystem (IMS), which is the designated solution of offering multimedia services in mobile networks. It shifts from the legacy circuit-switched (CS) technology to the IMS-based, packet-switched (PS) design. To empower the Internet users to communicate with mobile users via SMS, there exists the other Internet-based SMS. Based on this technology, many *SMS-powered services* are developed by companies to interact with their customers via SMS (*e.g.*, Uber contacts users via SMS).

Figure 1 shows the architecture of the Internet-based, CS-based (*i.e.*, the legacy), and IMS-based SMS services. Each of them has an SMS client at the end device. The client sends/receives SMS messages to/from a central controller called SMS Center. The SMS center is responsible for store-and-forward of SMS messages. We next elaborate each SMS service.

Internet-based SMS. The SMS client maintains a session with the server of the Internet SMS provider (*e.g.*, Twilio, Vibes, *etc.*). The server forwards SMS messages between the Internet client and the SMS center in the mobile network. The interface used between the server and the SMS center relies on SMPP (Short Message Peer-to-Peer) protocol.

CS-based SMS. It is mainly used in 2G/3G networks. The SMS client on the mobile phone relies on the CS gateway to forward SMS messages to/from the SMS center. The messages are carried by a particular control signaling through the control plane of the mobile network.

IMS-based SMS. Unlike the CS-based SMS, its messages are carried by particular data packets through the data plane, which is taken care of by the PS gateway. Specifically, it relies on the popular Session Initiation Protocol (SIP) [36] to control the SMS delivery. A SIP-based session is maintained between the phone's SMS client and the IMS server. The IMS server is responsible for bridging the SIP session and the SMS center [1].

2.1 Current Threats and Defenses

The practical security issues of the IMS-based SMS are less explored, but those of both the CS-based and Internet-based SMS services have been well studied [8, 15, 26, 30, 39, 45]. There are two major SMS threats in those two SMS services: unauthorized SMS access and SMS spoofing. The former mainly happens on mobile phones, whereas the latter takes place on both mobile and the Internet devices. We describe each threat and its defenses below.

Unauthorized SMS access. A mobile application maliciously

[1]We here focus on the IMS-based SMS supported by the 4G network, though it can be also deployed in the 3G network.

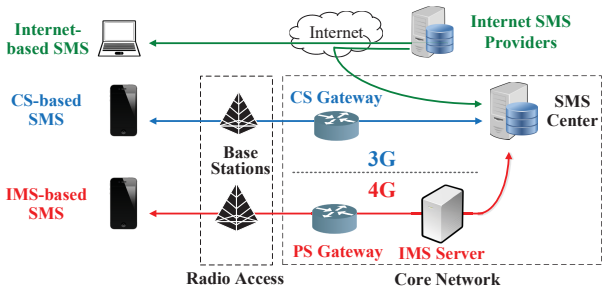


Figure 1: Architecture of the legacy (Internet-based and CS-based) and IMS-based SMS services.

sends out SMS messages without user consent. This threat has been largely prevented by most antivirus applications (*e.g.*, Kaspersky and McAfee) and mobile OS. Both of them monitor all the SMS activities from mobile applications, and then take actions when any malicious behaviors are detected (*e.g.*, users send SMS messages to the numbers that would cause monetary loss). Some actions have been taken by antivirus applications, such as stopping malicious SMS activities, bring them to users' attention, *etc.* The major action taken by mobile OS (*e.g.*, Android) is to halt each malicious SMS activity until the user permits it through a pop-up dialog.

SMS Spoofing. An attacker is able to set who an SMS message appears to come from by replacing the originator's phone number so that she can send out SMS messages to a recipient on behalf of another mobile user. This threat happens in both the CS-based and Internet-based SMS services. For the CS-based SMS, the threat comes from the vulnerability of 2G networks, no mutual authentication [11]. It does not require mobile phones to authenticate their serving networks, so they may be trapped into fake 2G networks. Through the fake networks, spoofed SMS messages can be easily sent to the phones. Though they can be detected by the law enforcement agencies [10], there are still no ultimate solutions that can eliminate all the fake networks or prevent devices from associating with the fake ones. Moreover, the 2G technology will not completely disappear in the near future. Some carriers [7] will retire 2G shortly, but other carriers and most devices will still maintain 2G backward compatibility for years.

The Internet-based SMS spoofing is based on the vulnerability that the Internet SMS providers do not restrict the originator number of each SMS message only to be the originator's. Their customers are thus allowed to maliciously send out the spoofed SMS messages on behalf of other mobile users, who are considered as victims. For example, by sending a spoofed SMS message on behalf of a victim to a charity organization, which provides an SMS-powered service to accept the donation, an attacker can force the victim to donate money. There were even some web sites [44, 46, 47] offering SMS spoofing services. This spoofing threat can be prevented by authenticating the originator numbers of SMS messages. However, this mechanism is not stipulated in the standard, but relies on various Internet-based SMS providers outside mobile networks. It is still uncertain whether all providers adopt the mechanism or do it right, so the threat may not be eliminated completely.

Note that these two threats may also happen for the IMS-based SMS, but the current defenses are not applicable to it. Different from the current SMS spoofing attacks, which are from either the fake 2G networks or the Internet, our identified SMS spoofing threat comes from malicious mobile users inside mobile networks.

3. NEW SECURITY ISSUES FROM IMS-BASED SMS

We investigate the security issues of the IMS-based SMS by systematically considering all the aspects where it differs from the legacy, CS-based SMS. There are three major dissimilarities: SMS client design (*i.e.*, software-based versus hardware-based), SMS protocol design (*i.e.*, different flexibility), and the communication channel between the SMS client and the core network (*i.e.*, data-plane versus control-plane). We also review the security mechanisms stipulated for the IMS system in the standards. In the following, we analyze the security issues which may happen from these four aspects, and present our threat model and methodology.

Software-based Client Design. The IMS-based SMS takes a software-based client design so that the SMS client is deployed as a mobile application (*e.g.*, an Android application). Different from the hardware-based client of the legacy SMS, it is so flexible that carriers can customize SMS to satisfy various user demands. However, it is more vulnerable to attacks, since abusing the software-based client is much easier than compromising the hardware-based client located at the phone modem. Once the mobile OS does not have any proper security protection for the SMS client, SMS is exposed to security threats. For example, a malicious user can hijack the SMS client to attack the IMS system, or the malware can send out forged SMS messages by pretending to be the SMS client.

Flexible Protocol Design. The SIP protocol on which the IMS-based SMS relies is more flexible than the legacy SMS. Specifically, the control information in the SMS message header can be specified on the device end, but the legacy SMS does not offer such flexibility. Without the strict security check of the SMS message header in the IMS server, this flexibility may allow malicious users or the malware to easily forge harmful SMS messages. For example, a malicious user can spoof the originator's number of a SMS message to launch the attack of SMS spoofing.

Data-plane Communication Channel. The IMS system shifts the communication channel of SMS, which is between the client and the core network, from the control plane of the legacy SMS to the data plane. All well-tested security mechanisms of the control-plane signaling (*e.g.*, identity authentication, message encryption, integrity protection, *etc.*) at the CS gateway (as shown in Figure 1) are not applicable to the PS-based IMS system. As a result, when the IMS system's security mechanisms are still at the initial stage, its offered SMS can be vulnerable to security threats, like VoLTE [24]. For example, the forged SMS messages can take effect without being detected.

Stipulated Security Mechanisms. The 3GPP and 3GPP2 [2] standards, which are two different telecommunication specifications, stipulate different IMS system designs and security mechanisms. They are funded by two different industry consortiums, and each of them has supporters. In the US, they are supported by AT&T/T-Mobile and Verizon/Sprint, respectively. In the 3GPP standard, IPSec-3GPP, where IPSec (Internet Protocol Security) is a protocol suite for secure IP communications, is the only and mandatory security mechanism for IMS. However, the 3GPP2 standard offers more freedom on the IMS security. Besides the IPSec-3GPP mechanism, it provides other four options: TLS (Transport Layer Security), DIGEST (Digest authentication), IPSec-IKE (IPSec Internet Key Exchange), and IPSec-Main. Such freedom may expose the IMS system to security threats. For example, one carrier may employ only the DIGEST mechanism without the end-to-end security, IPSec, so that it may suffer from eavesdropping or man-in-the-middle attacks.

3.1 Threat Model and Methodology

The victims can be mobile users, carriers (the IMS system is under attack), or SMS-powered service providers, whereas the presumed attacker is a mobile user. The attacker has a rooted commodity smartphone so that she can gain full control of the mobile OS for two purposes. First, she can crack the IPSec protection of SMS to launch attacks against the IMS system with IPSec (*i.e.*, SMS spamming). Second, she can collect the information that cannot be obtained without root access. It includes the SIP session information of SMS, which is required for the development of the malware used in the attacks. In the attack of SMS spoofing, the malware is not required to be deployed on the victim phone, but can be put on another phone that is used as a springboard to attack the victim. However, the other attacks requiring the malware need it to be on the victim phone. Note that the malware does not require root access or SMS permission (*i.e.*, allowed to use SMS API to send/receive SMS messages) in the mobile OS, but only network permission. In all cases, the attacker has no full control of the victim phones, the IMS system and the SMS-powered service servers.

We validate vulnerabilities and attacks in two top-tier US carriers, which are denoted as OP-I and OP-II for the privacy concern. They together take almost 50% of market share [29]. We conduct experiments by using three Android phone models that support the IMS-based SMS: Samsung Galaxy S5 with Android 4.4.4, Samsung Galaxy S6 with Android 5.0.2/6.0.1, LG G3 with Android 4.4.2. We here focus on the Android OS, but we believe that the identified issues are applicable to any other OS.

We bear in mind that some feasibility tests and attack evaluations might be harmful to mobile users, carriers, and SMS-based service providers. We thus conduct this study in a responsible manner through two measures. First, we use only our own phones as the victims. Second, we purchase unlimited SMS plans for all tested phones. We seek to disclose new security vulnerabilities of the IMS-based SMS and SMS-powered services, as well as effective attacks, but not to aggravate the damage.

4. NEW THREATS FROM IMS-BASED SMS

In this section, we introduce new threats from the IMS-based SMS. According to the security issues presented in Section 3, we discover four vulnerabilities ranging from the phone to the IMS server. Two vulnerabilities are in the SIP session of SMS, whereas the other two are on the phone and the server, respectively. They enable a malicious application without root access to fabricate legal SMS messages and deliver them successfully without the awareness of security applications, mobile OS or users. It results in the attack of silent SMS abuse. More threateningly, the originator phone numbers of the forged SMS messages can be spoofed, thereby leading to the SMS spoofing attack. The forged SMS messages can be further manipulated to launch the DoS and spamming attacks against the SMS client and the IMS system, respectively.

4.1 SIP Session Vulnerabilities

The IMS-based SMS relies on a SIP session between the SMS client on the phone and the IMS server. On the phone, a new interface is created for the IMS-based services (*e.g.*, VoLTE, SMS, *etc.*). It is different from the network interface of mobile data services. We here call this new interface as IMS-specific interface. The SIP session is established over this IMS-specific interface by the SMS client. It remains active as long as the IMS-based SMS is on. We next present two vulnerabilities of this SIP session: (V1) leakage of SIP session information and (V2) injection of forged SIP messages.

4.1.1 (V1) Leakage of SIP Session Information

Once the SIP session information (*e.g.*, message format, session parameters, *etc.*) is disclosed, the attacker may be able to fabricate legal SIP messages to carry forged SMS messages based on it. Though the SIP session is protected by the security mechanisms stipulated in the standards, we discover that the session information can be leaked to the attacker for both OP-I and OP-II. For OP-I, the session is secured only by the DIGEST security mechanism, which provides only access authentication during the SIP registration procedure. Without data confidentiality, the SIP messages are thus in plain text. For OP-II, the session is protected by the IPSec-3GPP with data confidentially, but the encrypted SIP messages can still be decoded by the method presented in the work [51]. Note that the packets containing SIP messages can be captured by monitoring the IMS-specific interface. Though this packet capture requires root access, it can be done at the attacker phone beforehand. After getting the understanding of the session information, the attacker can develop the malware without root access for the attacks that are introduced later.

Validation. We use a mobile application called *Shark* to capture the SIP messages of SMS by monitoring the IMS-specific interface. For OP-I, the SIP messages are plain-text, so the session information can be easily obtained. For OP-II, we load the IPSec keys that are being used by the Android OS to the *WireShark* application to decrypt the SIP packets. Note that the IPSec keys can be fetched using the command `ip xfrm state`.

We here show how to get the SIP session information by considering a packet trace of OP-I as an example. Figure 2 shows an example packet trace of the SIP registration procedure. After registering to the IMS server, the SMS client receives a message with the status, `SIP 401 Unauthorized`. In this message, the authentication method is specified to be DIGEST, and several parameters are included for authentication (*e.g.*, challenge, algorithm, *etc.*). The SMS client then finishes the registration by sending another new message of `SIP REGISTER`, which includes the answer to the authentication challenge. Afterwards, no encryption of the received message with the status, `SIP 200 OK`, shows that the follow-up SIP messages will be plain-text. Figures 3 and 4 respectively show the SIP message header and body for an SMS message. As a result, the SIP message format can be easily learned.

Causes and lessons. The 3GPP2 standard, which is supported by OP-I, leaves the freedom of security mechanisms to carriers, but such freedom may expose the IMS system to security threats. Due to some operational concerns, carriers may prefer simple security methods, which are not secure enough. For example, one server with the IPSec support is much more expensive than another without it, given that they support the same bandwidth. Therefore, in order to secure the IMS system, the 3GPP2 standard should learn from 3GPP that the most secure mechanism, IPSec-3GPP, is set to be mandatory.

Though OP-II relies on the IPSec-3GPP mechanism by following the 3GPP standard, the SIP session information can still be leaked. It is because the IPSec security module (*e.g.*, `XFRM` in Android) leveraged by the SMS client can be abused by malicious users with root access. To be more secure, the SMS client may require another level of security protection on top of the IPSec.

4.1.2 (V2) Injection of Forged SIP Messages

We discover that for both OP-I and OP-II, the forged SIP messages can be injected into the SIP session between the SMS client and the IMS server. The IMS server accepts all the packets with the correct session identifier (*i.e.*, the destination address pair of the

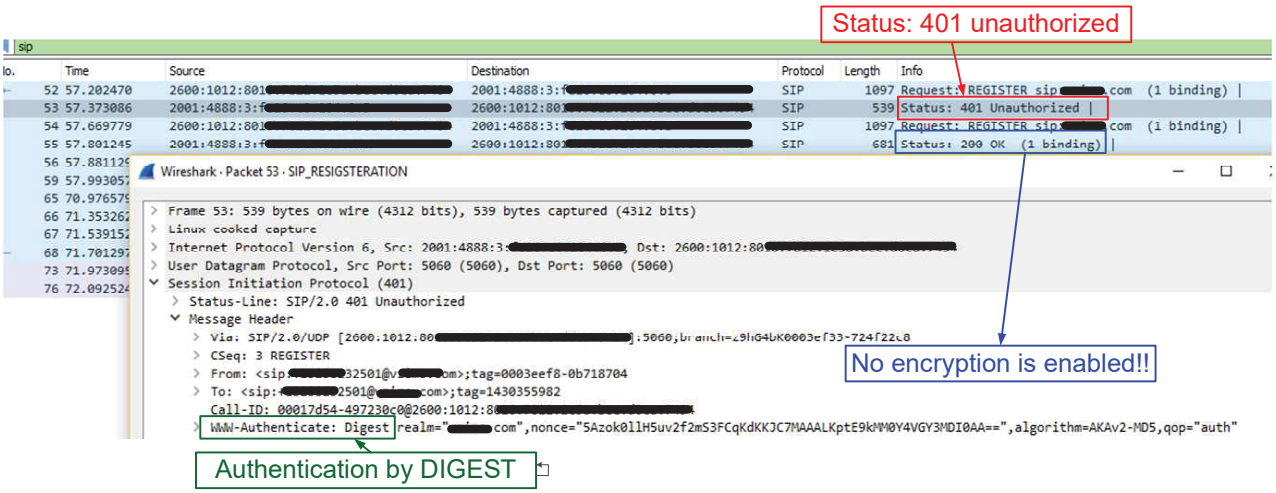


Figure 2: An example packet trace of the SIP registration procedure at OP-I.

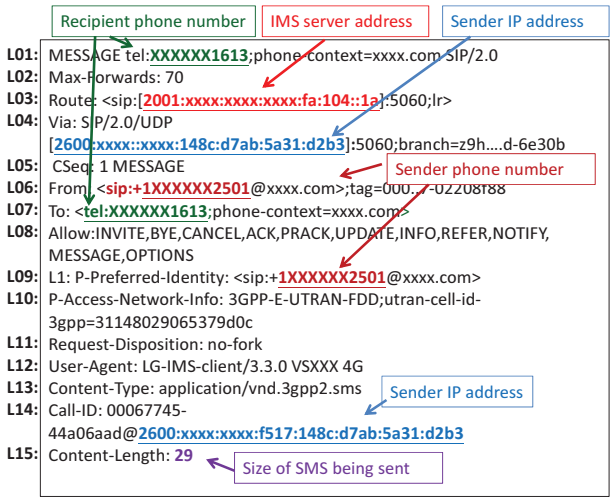


Figure 3: The SIP message header of an SMS message.

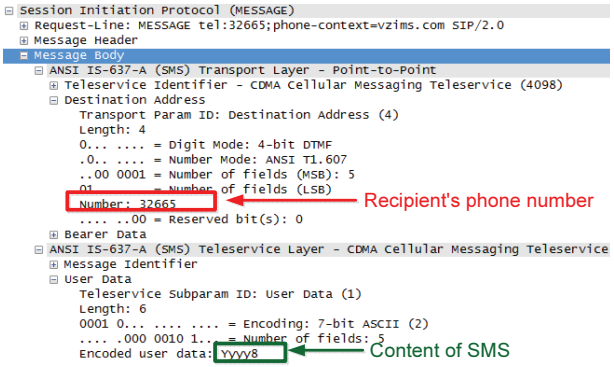


Figure 4: The SIP message body of an SMS message.

UDP-based SIP sessions, or the 5-tuple address of the TCP-based session) no matter where they come. For OP-I, the SIP session is built over UDP. Once a forged UDP packet with the SIP session's destination IP and port is sent via the IMS-specific interface, the IMS server can receive it and consider it to belong to the SIP session. For OP-II, though the SIP session is protected by the IPSec over TCP, a forged TCP packet can still be sneaked into the SIP session by using the security module, `XFRM`. Note that exploiting this vulnerability does not require root access at OP-I, but it is needed at OP-II.

Validation. We first validate this vulnerability for OP-I. We show that an application without root access can sneak a forged SIP message into the SIP session, and the recipient specified in the message can receive it. Based on the understanding of the SIP packet format from V1, the application can fabricate a SIP message with the header shown in Figure 3. It then uses a UDP socket to send it to the destination address of the ongoing SIP session via the IMS-specific interface. The successful delivery of the message is validated based on a response message with the status, `202 Accepted`, from the IMS server. Moreover, the recipient, `XXX-YYYY-1613` [2], as shown in Line 1, indeed receives it.

Note that though some information items are required to forge the header and send out the forged message, all of them can be obtained without root access. The major ones are the IMS server's IP address and the UDP socket's destination port number. The former can be fetched from the routing table and the latter is fixed for each carrier.

We next validate this vulnerability for OP-II, but root access is required. There are three major steps to do the message injection. First, we fetch the information of the IPSec security using the command, `ip xfrm state`. It includes the HMAC-SHA1/SHA2 keys for integrity protection and authenticity, and the SPI (Security Parameter Index) value of ESP (Encapsulating Security Payload). Second, we configure `XFRM` with the IPSec information and the SIP session identifier of TCP (*i.e.*, 5-tuple address). Third, we use `RAW SOCKET` to create an IP/TCP packet with the 5-tuple address to carry the forged SIP message, and then send it out through the IMS-specific interface. Before being sent out, the packet is automatically encapsulated into an IPSec packet due to the `XFRM` configuration. Afterwards, a response message with the same status `202 Accepted` is received.

Causes and lessons. Although the data-plane communication

[2]For the privacy concern, only the last four digits are shown.

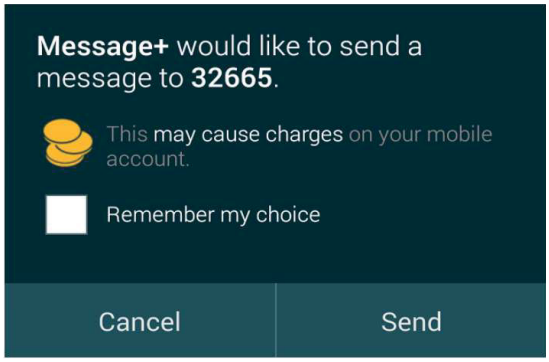


Figure 5: A pop-up confirmation dialog that is used when a non-SMS application sends out a SMS message that may cause charges.

offers more flexibility to the IMS-based SMS, the IMS server does not restrict the traffic carried by the SIP session to the SMS client only. For OP-I, no end-to-end security mechanism is used so that the injection of forged messages can be easily done without root access. For OP-II, even if the IPSec mechanism is employed, the message injection can still happen with root access. It may come from the IMS system's fundamental limitation that the IMS server is only able to authenticate the device but not software (*i.e.*, the SMS client), since the private keys used for the IMS service are installed in the device hardware (*i.e.*, SIM card). Once malicious users can leverage the keys no matter which way is used (*e.g.*, exploiting the security module `XFRM`), the IPSec security mechanism can be abused.

4.2 Phone and IMS Server Vulnerabilities

We identify other two vulnerabilities at the phone and the IMS server respectively: insufficient SMS access defense and spoofable SMS messages.

4.2.1 (V3) Insufficient SMS Access Defense at Phone

Based on the current SMS access defense on the phone, when a non-SMS application without root access wants to send SMS messages, it has to be granted the SMS permission during its installation (*e.g.*, SEND_SMS in Android). However, when SMS moves from the CS-based technology to the IMS-based design over PS network, mobile OS may not provide sufficient defense of the SMS access. We discover that for OP-I, an application without the SMS permission is allowed to send SMS messages by using network sockets. This application does not require root access but only the network permission. Such vulnerability empowers the malware to abuse SMS without awareness of security applications, mobile OS, or users. Note that this vulnerability is not feasible for OP-II, since sending out SMS messages from a non-SMS application requires to crack the IPSec security (as presented in the V2 validation) but the cracking needs root access.

Validation. We validate this vulnerability for OP-I by letting an application send an SMS message to a service number which causes charges on the sender's account. When the application with the SMS permission relies on the SMS APIs to do it, the Android OS will pop up a dialog to request user's confirmation before sending it out, as shown in Figure 5. Based on V2, we develop an application without root access or the SMS permission, and it is able to send out forged SMS messages by injecting the SIP messages to the SIP session. It is observed that there is not any pop-up confirmation dialog at the sender phone, after the application sends out an SMS message to the number.

Causes and lessons. The root cause is that the SMS permission control relies on the SMS API to monitor the SMS activities, but the way of using network sockets to send out SMS messages is neither monitored nor prevented. Though SMS has been shifted to the IMS-based SMS, the security mechanisms at the phone are not updated with its deployment. Moreover, the loopholes that the IMS-based SMS may open in the existing system are not carefully examined and addressed by mobile OS.

4.2.2 (V4) Spoofable SMS Messages at IMS Server

To ensure that the sender identifier specified in an SMS message belongs to its genuine sender, it is not an issue for the CS-based SMS. It is because the control information including the sender identifier is specified by the CS gateway in the core network. However, the control information of the IMS-based SMS is allowed to be specified at the phone, so the identifier may be spoofed by malicious users. Once the IMS server does not check the sender identifier of each incoming SIP message, the spoofed SMS messages can be delivered to their recipients.

Validation. We discover that this vulnerability works for OP-I but not for OP-II. In order to pass the integrity check at the IMS server of OP-I, we identify that there are eight header fields which have to be filled correctly in a forged SIP message. The values of the other fields do not affect the result of the integrity check. As shown in Figure 3, those eight fields are Request-Line at Line 1, Route at Line 3, Via at Line 4, From at Line 6, To at Line 7, P-Preferred-Identify at Line 9, Call-ID at Line 14, and Content-Length at Line 15. They respectively require the recipient's phone number, the IMS server's address, the sender's IP address, the sender's phone number, the recipient's phone number, the sender's phone number, the sender's IP address, and the message body size. For the message body, there are only two fields required to be filled: the recipient's phone number and the SMS message content, as shown in Figure 4. To build the message body, we can employ a set of classes (*e.g.*, CdmaSmsAddress and BearerData) in the library, *ITelephony* [12].

We validate this vulnerability by sending a spoofed message, where the specified sender number is not the sender's, to a recipient. The message can be received by the recipient, and it still carries the spoofed sender number. We want to note two things. First, all the required information items in the message header can be obtained without root access. For the IMS server's address, the IP address can be fetched from the phone's routing table, whereas the port number is always the same for each carrier. Second, the sender's IP address and phone number do not need to match. That is, a malicious user can spoof other phone numbers by keeping using her IP address.

Causes and lessons. The root cause is that there is no secure binding between the PS network identifier (*i.e.*, IP address) used to set up the SIP session, and the sender identifier (*i.e.*, phone number) specified in the SIP message. The IMS-based SMS offers more flexibility than the legacy from the protocol design, but does not prevent the abuse of the flexibility. OP-I should learn from OP-II that the secure binding is applied to SMS messages.

4.3 Proof-of-concept Attacks

We devise four proof-of-concept attacks: (1) silent SMS abuse; (2) SMS spoofing; (3) SMS Client DoS; (4) SMS spamming toward IMS. All the attacks can be launched in the OP-I network, whereas in the OP-II, only the forth one is feasible. For OP-I, the two at-

tacks, silent SMS abuse and SMS client DoS, require the malware at the victim phone. However, the other two can be launched from the attacker phone or any non-victim phone that is used to be a springboard and has the malware installed. Note that the malware in these attacks does not require root access. For OP-II, the SMS spamming attack requires root access at the attacker phone to crack the SIP session's IPSec.

Silent SMS abuse. Clearly, the discovered loopholes can be exploited to abuse SMS on a mobile phone silently. This silent SMS abuse can result in the victim's monetary loss. It works as follows. The malware without root access requires to be deployed at the victim phone, and sends out its forged SMS messages to the recipients who cause charges (*e.g.*, premium-rate text service [50]). With V1, the attacker can develop the malware which knows how to fabricate SIP/SMS messages. According to V2 and V3, the malware is able to send out the forged SIP/SMS messages via the IMS-specific interface without getting the victim's attention (*e.g.*, no pop-up confirmation dialog.)

SMS spoofing. The attacker can send SMS messages on behalf of another mobile user without his/her awareness or involvement. Such SMS spoofing attack may lead to the victim's monetary loss, and the hijacking of the victim's account, to name a few, when it targets the SMS-powered services of the victim. More details of the damage propagating towards the SMS-powered services are presented in Section 5. The attack works as follows. According to V2, the attacker can successfully send out the forged SIP/SMS messages, where the originating number is set to the victim's phone number, to the IMS server. The spoofed SMS messages can be then delivered to the recipient due to V4. From the recipient's point of view, those SMS messages are sent by the victim.

Moreover, this SMS spoofing attack can be launched from other phones to prevent the attacker from being traced back. This attack can be done by the malware without root access at those springboard phones. Though there exists a risk that the malware can be detected, thereby impeding the attack or possibly tracing back to the attacker, the risk is very low. It is because the current defenses (*e.g.*, the security mechanisms from two research studies [3, 19], the confirmation dialog of the Android OS, and other mechanisms from antivirus applications) against the SMS malware all focus on whether the applications with the SMS permission would abuse SMS or not. However, the malware does not require the SMS permission.

SMS client DoS. The malware on the victim phone can send a large amount of SIP/SMS messages to the local phone's SMS client, thereby exhausting the client's resources to result in its DoS. In order to send a SIP message to the local SMS client, the malware requires to configure its destination address with the IP address of the local IMS-specific interface and the port number used for SMS (*i.e.*, 5060 in OP-I). Its source address can be assigned any arbitrary IP address and port number. Note that the IMS-specific interface's IP address can be obtained from the system's network information without root access. Due to different implementations of the SMS clients from phone companies, different results are observed. We here examine two different SMS clients from Samsung and LG, and test two phone models, Samsung S5 and LG G3, respectively.

On the phone S5, the DoS attack prevents the SMS client from receiving any SIP/SMS messages. It is because the SMS client, the process of which is named as `com.sec.ims.android`, cannot handle any incoming SIP messages once the client's CPU usage is equal to or higher than 25% on the tested phone. We observe that when the malware sends SIP messages to the SMS client with the speed at least 3,825 messages (1 KB each) per second, the client would suffer from DoS with the CPU usage at least 25%. The effect can last if the attack does not stop. Note that the threshold of the SMS client's CPU usage may vary with different clients, systems, or/and other ongoing services.

More severely for the phone G3, the DoS attack crashes the SMS client on the victim phone. With the crash, the mobile OS would also stop responding or slowly respond to user input for a period of time. Though the SMS client can be automatically recovered within 150 seconds after being crashed, the attack can be repeatedly launched to crash it whenever its recovery completes. The root cause is that the SMS client, the process of which is named as `com.lge.ims`, has the vulnerability of memory leak. We discover that the process would buffer part of the incoming SIP/SMS messages that have new recipient phone numbers into its memory. When a SIP message, which has the large size of 7.5 KB and a new recipient number, is sent to the SMS client, the client's memory usage would increase by 4.5 KB. We thus develop the malware to launch the attack by continually sending the SIP messages that have the large size and different recipient numbers, to the SMS client on the victim phone. It is observed that the attack takes around 200-250 seconds to crash the client's process with more than 140K forged SIP messages, and its memory usage reaches 128 MB right before the crash.

SMS spamming towards IMS. The attacker can also launch SMS spamming attack towards the IMS system to downgrade its performance. This attack, which relies on V2, requires root access to crack IPSec at the attacker phone for OP-II, but it is not needed for OP-I. The victims are the IMS systems of the carrier networks. The attack aims to cause the heavy computation load (*e.g.*, decrypting plenty of IPSec packets, handling lots of SIP messages with large sizes, *etc.*) in the IMS system by sending a large amount of SIP/SMS messages to the IMS server. However, due to the legal concerns, we did not conduct this attack against the IMS systems of OP-I and OP-II.

5. THREAT PROPAGATION TOWARDS SMS-POWERED SERVICES

In this section, we examine how the threats caused by the IMS-based SMS menace SMS-powered services. We study 40 SMS-powered services, which are summarized in Table 2, in the US. With the threat of the SMS spoofing, an attacker can send SMS messages to use the SMS-powered services on behalf of a victim without his/her awareness. Together with the vulnerabilities of these SMS-powered services, the SMS threat can be manipulated to launch attacks against them. It can lead to three major types of attacks: account hijacking, unauthorized donation, and unauthorized subscription. We below consider one representative service of each attack type as an example to illustrate service vulnerabilities, attack methods, and negative impacts. The SMS-powered services corresponding to those three attack types are offered by Facebook (a social network company), American Red Cross (a charity organization), The Home Depot (a home improvement retailer), respectively. Note that these attacks are feasible only for OP-I, but not OP-II.

5.1 Facebook: Account Hijacking

A Facebook user is allowed to use SMS to manage his/her account (*e.g.*, posting status, adding a friend, poking someone, liking a page, *etc.*) with the service of Facebook Text [16]. For example, for the action of liking the Facebook page, Lakers Nation, a user can send an SMS message with the text, `Like LakersNation`, to the number `32665`. To use this Facebook Text service, the user

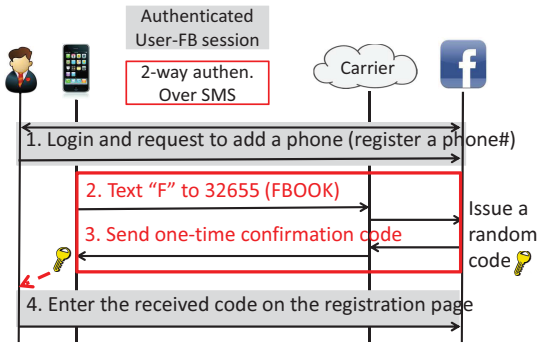


Figure 6: Authentication procedure of the phone number registration on Facebook.

is required to securely bind his/her phone number to the account beforehand, as shown in Figure 6. During the phone number registration, Facebook relies on a one-time confirmation code to authenticate the number. However, the Facebook Text service can be abused to launch the account hijacking attack due to its vulnerabilities and the SMS spoofing threat.

Vulnerabilities. We identify two security vulnerabilities of the Facebook Text service: *no runtime authentication* and *inappropriate binding of phone number registration and Facebook Text service*. The first vulnerability is that a user can keep using the registered phone number to manipulate his/her account via SMS without any runtime authentication. The initial authentication of the phone number is the only security mechanism used for the Facebook Text service. It is validated in our 28-day experiment. As a result, once obtaining the victim's phone number, the attacker is able to manipulate the victim's Facebook account by sending out the spoofed SMS messages.

The second one is that once a user registers his/her phone number, his/her Facebook Text service is automatically enabled. However, the user's phone number registration may be used for only security purpose (*e.g.*, password recovery). This inappropriate binding may expose the user to the SMS threats, but s(he) does not know it. Moreover, the number of this kind of users is not small, because Facebook encourages users to register their numbers by continuously showing a reminder dialog, as shown in Figure 7.

Attack and negative impacts. By leveraging the SMS spoofing threat, the attacker can hijack the victim's Facebook account via SMS without the victim's awareness. Since there is no runtime authentication, the victim would not get involved in the attack or receive any confirmation messages. We develop an Android application, which is named as *HackFacebook*, to launch this attack. It does not require root access, but only the network permission. It can be also converted to be the malware that is used to launch attacks from the non-attacker phones.

We validate the attack feasibility by using *HackFacebook* to attack the victim phone which has the number, `XXX-YYYY-4347`, and the associated Facebook account, ResearchOne. We employ *HackFacebook* on one non-victim phone to update status, add a new friend, and like a page for the victim account, as shown in Figures 8(a), 8(b), and 8(c), respectively. The snapshot of the victim account's activity logs, as shown in Figure 8(d), confirms that those three attack actions are successful. This attack can be further employed to disclose the victim's private information (*e.g.*, friends, family members, photo, *etc.*), because the attacker can add one fake account to be one of the victim's friends without the victim's consent. The attacker can thus obtain all the information that the victim shares with his/her friends on Facebook.

5.2 American Red Cross: Unauthorized Donation

American Red Cross (ARC), a non-profit humanitarian organization, allows a mobile user to make a donation from his/her monthly bill via SMS. This service is called Mobile Giving. For example, to make a $10 donation, a mobile user can send an SMS message with the text, `REDCROSS`, to the number, `90999`. Afterwards, s(he) receives an SMS message with a confirmation request, and then needs to reply it by sending another message with the text, `YES`[3]. The donation will be charged in the user's monthly bill. Though the Mobile Giving service offers mobile users a very convenient way to make donations, its vulnerabilities can be abused to launch the attack of unauthorized donation.

Vulnerabilities. We discover two security vulnerabilities of the Mobile Giving service: *weak authorization* and *automatic enrollment*. The service's authorization mechanism, which relies on only a static response text (*i.e.*, `YES`), is too weak to defend against malicious attacks. The service can be easily abused, if the attacker is able to manipulate the victim's SMS. The second vulnerability is that most US carriers including AT&T, Verizon and T-Mobile, automatically enroll their mobile users to the Mobile Giving service. Some users may not be aware of it, since there is no explicit notification of this automatic enrollment. It is dangerous especially when this service is related to the users' monetary expense. It also provides an opportunity for the attacker to cause the victim's monetary loss by abusing SMS.

Attack and negative impacts. The attacker can make an unauthorized donation from the victim's mobile bill by launching the SMS spoofing attack against the Mobile Giving service. Due to the service's weak authorization, the attack can be done without the victim's involvement. To launch this attack, we develop an application, *DonateARC*, based on *HackFacebook*. Different from *HackFacebook*, it requires to send two consecutive SMS messages to the number, `90999`. The text in the first message is `REDCROSS`, whereas that in the second one is `YES`. Since the second message cannot be sent out until the first one is successfully delivered, a time interval is set between these two messages' deliveries. We set the interval value to be 5 seconds, because our experiments show that for OP-I, 95% SMS messages can be successfully delivered to their recipients within 5 seconds.

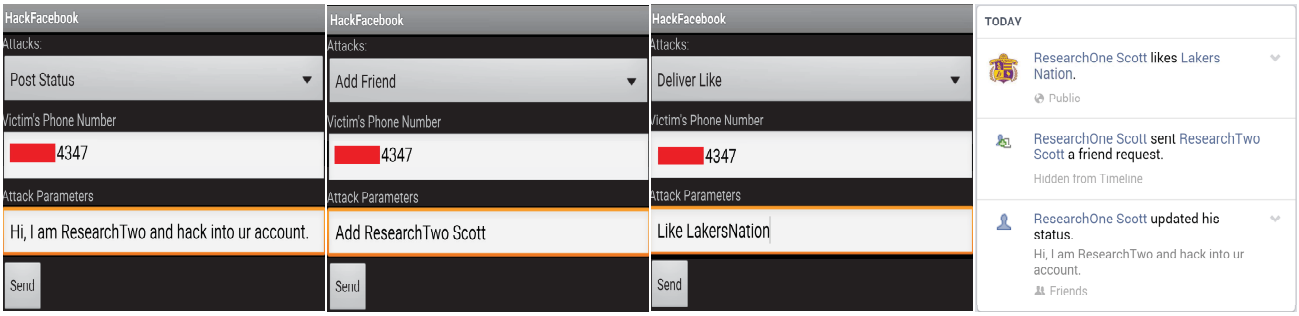
To validate the attack feasibility, we employ *DonateARC* to make an unauthorized $10 donation to ARC from the victim's mobile bill. After performing this attack, the victim's mobile bill, where $10 has been donated to ARC, confirms the validation. We want to note two things. First, the victim is one of our lab members. Second, the victim can be notified of the unauthorized donation by the SMS message of the confirmation request, but s(he) has no way of stopping the ongoing attack.

5.3 The Home Depot: Unauthorized Subscription

The Home Depot, an American retailer of home improvement needs, provides SMS advertising that sends advertisement (*e.g.*, exclusive offers, discounts, *etc.*) to customers via SMS. To receive the advertisement, a customer needs to subscribe to SMS advertising on the website of The Home Depot with his/her phone number and email address. After doing subscription, the customer would

[3]The text in reply to the confirmation may vary with different carriers.

Figure 7: The reminder dialog used by Facebook to encourage users to register their phone numbers for security.



(a) Update status (b) Add a friend (c) Like a page (d) Victim's activity logs

Figure 8: HackFacebook launches three attack actions against the victim, ResearchOne: update status, add a friend, and like a page.

receive an SMS message with a confirmation request. s(he) is then required to send another SMS message with the text, `Y`, in reply to it. Similar to the attack of unauthorized donation, the subscription can be abused to be without authorization.

Vulnerability. The vulnerability of the SMS advertising is the *weak authorization* procedure of the subscription. Similar to the Mobile Giving service, the subscription confirmation relies on a static response text (*i.e.*, `Y`). As a result, by leveraging the SMS spoofing, the attacker can let the victim subscribe to the SMS advertising without the victim's consent.

Attack and negative impacts. The attacker can make an unauthorized subscription to the victim so that the victim would keep receiving unwanted SMS advertisements and may feel annoyed. The way to launch this attack is similar to that of the unauthorized donation attack. Both of them have two steps, requesting donation or subscription for the victim and sending an SMS message to do confirmation. The major difference is that the first step of this attack needs to be done at the website but not via the delivery of an SMS message. We validate this attack by subscribing a victim to the SMS advertising of The Home Depot using the SMS spoofing. Our experiments show that it causes the victim to receive up to 10 SMS messages per month. When this attack is applied to a large number of mobile users, many complaints could be made from the victims. The goodwill of The Home Depot may be thus impaired. We will examine the feasibility of launching large-scale attacks in the next section.

5.4 Feasibility Study of Large-scale Attacks

We further study the feasibility of launching large-scale attacks for OP-I. In order to impede traceback, it is better for the attacker to launch our discovered attacks from the malware at other phones. So, we examine whether there is any limitation of sending a large number of forged SMS messages from the malware in that case. In our experiments, we use the *HackFacebook* application at one phone to send as many forged SMS messages as possible to another phone within a time period. We also test the default messaging application for the comparison by using it to send SMS messages as fast as possible. The test time is 30 minutes. Note that we purchase an unlimited SMS plan for each tested phone to avoid legal issues.

The result shows that in terms of the speed of the SMS message

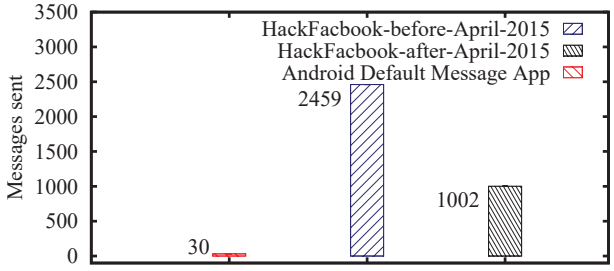


Figure 9: The number of the SMS messages that are successfully delivered within 30 minutes.

delivery, *HackFacebook* can be 33x faster than the default messaging application, as shown in Figure 9. It is observed that the default messaging application is unable to send more than 30 SMS messages within 30 minutes. However, the *HackFacebook* application can successfully deliver 2459 and 1002 SMS messages before April 2015 and after April 2015, respectively. The malware can thus launch large-scale attacks with the speed at least 33 SMS messages per minute. We have two other observations. First, there is a cap of 30 SMS messages per 30-minute by using the Android SMS API to send SMS messages. But, *HackFacebook*, which does not rely on the API, can bypass the cap. Second, it seems that OP-I deploys a network-based SMS control which limits the speed of the SMS message delivery after April 2015.

Note that it would be interesting to exhaust the IMS server's resources by launching large-scale attacks against SMS-powered services, and then cause SMS DoS. For example, the attacker uses a large number of the malware applications to keep subscribing to different SMS-powered services, thereby overloading the IMS server. However, we did not do it due to legal issues.

5.5 Lessons Learned

Most SMS-powered service providers rely on mobile networks to authenticate mobile users. They may assume that the users' phone numbers cannot be spoofed, so only weak security mechanisms or none are used for the SMS-powered services. When the mobile networks suffer from the SMS spoofing attacks, most

SMS-powered services are exposed to security threats. Moreover, there are various mobile networks around the world, so it is difficult for the service providers to ensure their security based on them. Therefore, the SMS-powered services should have their own authentication mechanisms. Moreover, not only the users of the SMS-powered services but also other mobile users may be attacked, because some service providers or carriers may automatically enroll their users to their SMS-powered services without the users' awareness. It is unfair for those users who do not subscribe to the SMS-powered services to bear the security risks.

6. RECOMMENDED SOLUTIONS

We propose recommended fixes to the IMS-based SMS threats. We seek to address the vulnerabilities at all the involved parties including the mobile phone, the mobile network, and the SMS-powered service provider. The proposed fixes consider not only feasibility, but also standard compatibility, deployment cost and the feedback from industry (*e.g.*, carriers and SMS-powered service providers).

Mobile phone. We recommend two remedies to two main vulnerabilities, which are insufficient SMS access defense (V3) and the IPSec crack (V1 and V2), on the phone. First, mobile OS should prevent all the ways other than using the SMS API from being leveraged to access SMS. That is, applications should be forbidden from using network sockets to send/receive SMS messages. To achieve it, the mobile OS should restrict the IMS-specific interface to the IMS-related applications only (*e.g.*, the SMS client). The advantage of our approach is that both the current SMS permission control of mobile OS and the existing permission-based malware detection methods [27, 38] can still be effective without any modifications. Certainly, this vulnerability can be resolved when the SIP session is protected by IPSec, like that of OP-II. However, enabling the IPSec protection requires upgrades of both the SMS client and the IMS system, thereby possibly resulting in too much overhead or deployment cost. The SIP session could still suffer when the IPSec can be abused on a rooted phone.

Second, the SMS client should have another level of security by itself in addition to IPSec, because the `XFRM` module on which the client relies can be exploited to abuse IPSec by the applications with root access. The additional security level can be done by hiding the destination address of the SMS service (*e.g.*, dynamic assignment of service ports, avoiding the leakage of the IMS server's IP address in the routing table, *etc.*), or adding a security method (*e.g.*, security challenges, DIGEST, *etc.*). In the former mechanism, hiding the destination address can increase the difficulty of knowing where the forged SMS messages should be sent. The latter security method can prevent the attacker from forging legal SIP/SMS messages. Note that both mechanisms require only the upgrade of the SMS client without modifying the mobile OS, and the former has been adopted by OP-I.

Mobile network. On the network side, we suggest a remedy to prevent the SMS spoofing. The network should provide a secure binding between the client's IP address used to set up the SIP session and the originator identifier (*i.e.*, phone number) specified in the SIP/SMS message. When a SIP session is initialized and authenticated successfully, the IMS server should bind the originator phone number specified in the initial messages to the client's IP address. Then, there are two ways to carry out the secure binding for the following messages within the session. First, the IMS server can verify the originator phone number in each SIP message, and drop it when the number is spoofed. Second, learned from the CS-based SMS, the originator information of the SIP message needs to be specified by the network (*i.e.*, IMS server) instead of the phone. Therefore, the IMS server would skip the originator information specified in each incoming SIP message and fill in the related fields with the early binding of the SIP session.

Note that one possible remedy is to add a secret (*e.g.*, DIGEST) to the SIP message so that the originator identifier can be verified in each message. It can prevent the attacker from fabricating legal SIP messages. However, this security mechanism requires extra efforts to verify the secret carried in each SMS. Besides, this approach, which is located at the core function of SMS, is not stipulated by the standards. To be compatible with the standards and avoid the possible interoperability issues, carriers do have concerns to carry out it.

SMS-powered service provider. We propose two remedies to strengthen the security of the SMS-powered service. First, one user's subscription to the SMS-powered service should be confirmed by the user. To some extent, it can prevent the attacker from successfully launching the SMS spoofing attack against the SMS-powered service. The attacks against the users without the service subscription would fail, and the users do not need to bear the security risks of the services to which they do not subscribe. Second, the SMS-powered service provider should authenticate the mobile user by itself for each service request. The reason a service relies on SMS is that SMS is a convenient tool to be used, so the authentication mechanism has to be simple and does not require any additional application to be installed. We suggest that one-way request without confirmation (*e.g.*, the Facebook service) should be issued with a secret code, and the confirmation request/reply (*e.g.*, the services of ARC and The Home Depot) should contain a dynamic short code. For example, to like a page (say, LakersNation) on Facebook via SMS, the text of the SMS request message should be `Like LakersNation, 3847`, where 3847 is the sender's secret code. To do the confirmation of a donation, the user should reply the confirmation request which includes a short code instead of the message with a static text. As a result, even if the SMS spoofing attack is feasible, the SMS-powered services are protected against it with the secret code or the short code. This approach has been adopted by Facebook as MobilePIN.

Note that though there have been several proposals for an SMS security framework [39, 39] and end-to-end SMS encryption [26, 33] to secure SMS communication. However, the lessons we learned from industry is that they either are too complex to be deployed or require another application (*e.g.*, Pushbullet or SilenceIM) to be installed. Relying on them to provide security may destroy the convenience of SMS, thereby discouraging people from using SMS. That might be the reason why Facebook chose to implement its MobilePIN (*i.e.*, appending a secret code to the SMS request), instead of those aforementioned approaches.

7. DISCUSSION

We next clarify several remaining issues.

How about other carriers' IMS-based SMS? We further examine whether the discovered vulnerabilities exist at other two major US carriers, which are denoted by OP-III and OP-IV for the privacy concern. They together take more than 45% of market share [29]. Our experimental results show that similar to OP-II, OP-III uses IPSec to secure the SIP session. At OP-III, Vulnerabilities V1, V2 and V3 exist, but V4 does not. We further discover that the IMS server always replaces the originator phone number with the sender's number, no matter what number is assigned to it on the phone. As for OP-IV, we do not observe that the IMS-based SMS is supported on our test phones.

Could TrustZone be a solution? TrustZone [5], a hardware-based security technology of ARM processors, could be a candidate solution for the IMS-based SMS issues. It partitions hardware into two worlds, trusted and non-trusted, thereby separating trusted software, data and hardware from the non-trusted world. In the trusted world, a secure network communication [25], such as the device's communication with the IMS server, can be built, or the SMS client can be placed. As a result, neither is the attacker able to get the SIP session information of SMS, nor does the malware have a chance to sneak into the SIP session. However, it has three concerns. First, the feasibility of TrustZone is processor-dependent, so it does not work for all the devices. Moreover, enabling it in the existing devices requires the firmware update, which not all users are willing to do. Second, there exists an overhead of the switch between two worlds, so whether to employ TrustZone should depend on the overhead, which should be evaluated by considering the SMS client's behaviors. Third, an outbound authentication issue [42] can also exist. Even if TrustZone can deal with security functions and/or secret keys, it is hard for TrustZone to authenticate all the requests coming from the non-trusted world. We will consider these concerns in our future work.

Premium SMS spoofing attack The premium SMS [32] is to provide third-party providers' services (*e.g.*, charitable donations, TV voting, financial/stock information, *etc.*) to users via SMS, and carriers charge them at the prices higher than normal SMS. Therefore, launching the attack of premium SMS spoofing can increase the victim's bill, thereby resulting in his/her monetary loss. The attack of unauthorized ARC donation presented in Section 5.2 is one of the premium SMS spoofing attacks. However, this type of attack may not work for all the carriers, since several major US carriers have stopped charging for most premium text messages [6].

Similar to MMS spoofing threat? People may think that the MMS (Multimedia Messaging Service) spoofing threat identified in early works is similar to the SMS spoofing introduced in this work. However, they are totally different, since they root in different protocols and security mechanisms.

8. RELATED WORK

In this section, we present related work in the security areas of LTE network, VoLTE (voice over LTE, an IMS-based voice service), and SMS. Several previous works [13, 22, 28, 41] have examined the security issues of LTE network. Shaik *et al.* [41] exploited the vulnerabilities of LTE access network to expose the mobile user's location. Dabrowski *et al.* showed that once the phone's IMSI (International Mobile Subscriber Identity) is exposed to the rogue base station the phone can be tracked, and proposed solutions to secure the access network. However, our work focuses on the security vulnerabilities of the core network (*i.e.*, IMS system) and the mobile software (*i.e.*, mobile OS and IMS client), rather than those of the radio access network. Two recent surveys [22, 28] focus on the security vulnerabilities that exist in the LTE network, but our discovered vulnerabilities and attacks are not presented in them.

There have been several works [23, 24] which study the VoLTE security issues. They identified several vulnerabilities of the VoLTE device and infrastructure, and further showed that the adversary can gain free data service by delivering data packets through the signaling or voice channel of VoLTE, and launch the DoS attack against VoLTE. There are three major differences from our work. First, our work looks into IMS-based SMS service, but not IMS-based voice service (*i.e.*, VoLTE). Second, our work focuses on the vulnerabilities of IMS signaling protocol, rather than those of IMS control/data channels, which are the focuses of those two VoLTE works. Third, the discovered vulnerabilities in our work can be exploited to launch large-scale attacks, which could cause monetary loss or privacy leakage to a large number of people, but not only individual attacks they focus. To the best of our knowledge, this is the first work that studies the security vulnerabilities of IMS-based SMS service in the operational networks.

The security issues of SMS service are hot research topics in recent years. There are several works [14, 31, 40] which focus on the (in)security of mobile two-factor authentication via SMS or other channels. The authors in the work [35] study the security practices of benign SMS services, and the malicious misuse of the SMS ecosystem. Other research studies include defending against the threats of user privacy leakage from the SMS [34], faking SMS configurations [37], launching attacks against the mobile device from fake network infrastructure [45], launching DoS attack against the mobile network by sending a large number of SMS messages from the Internet [15,49], devising the malware to abuse SMS [4, 19, 27], embedding malware/virus into the SMS message [8, 9] and man-in-the-middle attacks via SMS [20, 26, 30, 33, 39], to name a few. Different from them, our work focuses on the IMS-based SMS, but not the CS-based SMS or the Internet-based SMS.

9. CONCLUSION

In this work, we examine the security implications of IMS-based SMS. Several new vulnerabilities are discovered from IMS system and mobile OS. Though there exist security shields deemed effective for the legacy SMS, they hardly protect the IMS-based SMS. We show that the vulnerabilities can be exploited to launch attacks against mobile users and carrier networks. The users may suffer from the attacks of SMS spoofing and DoS, whereas the carrier networks may be under the spamming attack.

Moreover, the SMS threat can propagate to SMS-powered services. Most service providers rely on SMS to authenticate mobile users so that they may assume SMS messages cannot be abused. The services may thus have no runtime authentication or weak authorization, thereby being easily abused by the SMS threat. The users may suffer from account hijacking and unauthorized donation, whereas the providers may bear the risks of goodwill impairment. We show that no sophisticated attacks are needed, and simple attacks may work in practice. The solution calls for the concerted efforts among all parties involved. Any of parties which do not timely enforce the necessary security mechanisms will cause security threats and unexpected damages.

10. UPDATES

We have reported the identified issues to OP-I, Facebook, LG and Samsung, and worked with OP-I and Facebook to address their issues. According to the OP-I security patches at the phone, there are two remedies. First, the IMS server's IP address is hidden from the routing table, so the malware without root access is unable to know where the forge SIP messages should be sent. Second, OP-I adopts the locked bootloader [18] to prevent the phone from being rooted so that the IPSec of the SIP session cannot be abused. Besides, OP-I plans to deploy the network-based solution, which is the secure binding of the originator's phone number and IP address, in October 2016. As for Facebook, the user is allowed to specify a secret code for the Facebook Text service, but it is disabled by default. Moreover, Facebook removes the features of add-a-friend and like-a-page from the service. Note that the security team of LG and Samsung are currently investigating the security issues we reported.

11. ACKNOWLEDGMENTS

We greatly appreciate our shepherd, Prof. William Enck, and the anonymous reviewers for their valuable feedback. This work is supported in part by the National Science Foundation under Grants No. CNS-1421933, CNS-1422835, CNS-1528122 and CNS-1527613. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors only and do not necessarily reflect those of the National Science Foundation.

12. REFERENCES

[1] 3GPP. TS23.228: IP Multimedia Subsystem (IMS);Stage 2, 2012.
[2] 3GPP2. IMS Security Framework.
[3] S. B. Almina and M. Chatterjee. A novel approach to detect android malware. In *ELSEVIER ICACTA*, 2015.
[4] A. J. Alzahrani and A. A. Ghorbani. Sms mobile botnet detection using a multi-agent system: Research in progress. In *ACM ACySe*, 2014.
[5] Arm inc.: Trustzone, 2016. http://www.arm.com/products/processors/technologies/trustzone/.
[6] AT&T, T-Mobile, Sprint to stop charging for most premium text messages. http://www.computerworld.com/article/2486212.
[7] AT&T to Retire 2G - GSM Sunset. http://www.sine-wave.com/blog/2g-sunset-retiring#.VmDW-narRaQ.
[8] A. Bose, X. Hu, K. G. Shin, and T. Park. Behavioral detection of malware on mobile handsets. In *ACM Mobisys*, 2008.
[9] A. Bose and K. G. Shin. On mobile viruses exploiting messaging and bluetooth services. In *IEEE Securecomm and Workshops*, 2006.
[10] China arrests 1500 people sending TEXT from fake base stations. http://www.ibtimes.co.uk/china-arrests-1500-people-sending-spam-text-messages-fake-mobile-base-stations-1442099.
[11] China spammers' latest weapon: fake base stations. http://www.electricspeech.com/journal/2013/12/6/china-spammers-latest-weapon-fake-base-stations.html.
[12] com.android.internal.telephony.itelephony. http://grepcode.com.
[13] A. Dabrowski, N. Pianta, T. Klepp, M. Mulazzani, and E. Weippl. Imsi-catch me if you can: Imsi-catcher-catchers. In *ACM ACSAC*, Dec. 2014.
[14] A. Dmitrienko, C. Liebchen, C. Rossow, and A.-R. Sadeghi. On the (in)security of mobile two-factor authentication. In *FC*, 2014.
[15] W. Enck, P. Traynor, P. McDaniel, and T. La Porta. Exploiting open functionality in sms-capable cellular networks. In *ACM CCS*, 2005.
[16] Facebook texts. https://www.facebook.com/help/170960386370271/.
[17] Fortune 500: Top 1000 companies. http://fortune.com/fortune500/.
[18] Galaxy S7 Locked Bootloader May Stay Locked. http://www.androidheadlines.com/2016/03/galaxy-s7-locked-bootloader-may-stay-locked.html.
[19] K. Hamandi, A. Chehab, I. H. Elhajj, and A. Kayssi. Android sms malware: Vulnerability and mitigation. In *IEEE WAINA*, 2013.
[20] R. He, G. Zhao, C. Chang, H. Xie, X. Qin, and Z. Qin. A pk-sim card based end-to-end security framework for sms. *Computer Standards & Interfaces*, 31(4):629–641, 2009.
[21] H. Holma and A. Toskala. *WCDMA for UMTS - HSPA Evolution and LTE*. Wiley, 2007.
[22] R. Jover. Security attacks against the availability of lte mobility networks: Overview and research directions. In *WPMC*, 2013.
[23] H. Kim, D. Kim, M. Kwon, H. Han, Y. Jang, D. Han, T. Kim, and Y. Kim. Breaking and fixing volte: Exploiting hidden data channels and mis-implementations. In *ACM CCS*, Oct. 2015.
[24] C.-Y. Li, G.-H. Tu, C. Peng, Z. Yuan, Y. Li, S. Lu, and X. Wang. Insecurity of voice solution volte in lte mobile networks. In *ACM CCS*, 2015.
[25] X. Li, H. Hu, G. Bai, Y. Jia, Z. Liang, and P. Saxena. Droidvault: A trusted data vault for android devices. In *IEEE ICECCS*, 2014.
[26] J. L.-C. Lo, J. Bishop, and J. H. P. Eloff. Smssec: An end-to-end protocol for secure sms. In *Computers & Security*, 2008.
[27] W. Luo, S. Xu, and X. Jiang. Real-time detection and prevention of android sms permission abuses. In *ACM SESP*, 2013.
[28] M. Ma. Security investigation in 4g lte networks. In *IEEE GLOBECOM*, 2012.
[29] Market share of wireless subscriptions held by carriers in the U.S. http://www.statista.com/statistics/199359/market-share-of-wireless-carriers-in-the-us-by-subscriptions/.
[30] U. Meyer and S. Wetzel. On the impact of gsm encryption and man-in-the-middle attacks on the security of interoperating gsm/umts networks. In *IEEE PIMRC*, 2004.
[31] C. Mulliner, R. Borgaonkar, P. Stewin, and J.-P. Seifert. Sms-based one-time passwords: Attacks and defense. In *DIMVA*, 2013.
[32] Premium SMS. http://vodafone.intelliresponse.com.
[33] Pushbullet. http://www.androidcentral.com/pushbullet-adds-end-end-encryption-sms-notification-mirroring-and-more.
[34] R. Racic, D. Ma, and H. Chen. Exploiting mms vulnerabilities to stealthily exhaust mobile phone's battery. In *IEEE Securecomm and Workshops*, 2006.
[35] B. Reaves, N. Scaife, D. Tian, L. Blue, P. Traynor, and R. K. Butler. Sending out an sms: Characterizing the security of the sms ecosystem with public gateways. In *IEEE S&P*, May 2016.
[36] RFC3261: SIP: Session Initiation Protocol, 2002.
[37] Rooting SIM cards with SMS OTA. https://srlabs.de/rooting-sim-cards/.
[38] P. Rovelli and Ý. Vigfússon. Pmds: Permission-based malware detection system. In *ICISS*, 2014.
[39] N. Saxena and N. Chaudhari. Easysms: A protocol for end-to-end secure transmission of sms. In *IEEE Transactions on Information Forensics and Security*, 2014.
[40] E. Shablygin and S. Bratus. How to count to two: What "two factor authentication" misses. Feb. 2015.
[41] A. Shaik, R. Borgaonkar, N. Asokan, V. Niemi, and J.-P. Seifert. Practical attacks against privacy and availability in 4g/lte mobile communication systems. In *IEEE NDSS*, Feb. 2016.
[42] S. W. Smith. Outbound Authentication for Programmable Secure Coprocessors. In *ESORICS*, 2002.
[43] Sms phishing. http://en.wikipedia.org/wiki/SMS_phishing.
[44] Smsspoofing. http://www.smsspoofing.com/.
[45] Y. Song, K. Zhou, and X. Chen. Fake bts attacks of gsm system on software radio platform. *JOURNAL OF NETWORKS*, 7(2):275–281, 2012.
[46] Spoofcard. http://www.spoofcard.com.
[47] Spooftexting. http://www.spooftexting.com.
[48] M. Toorani and A. Beheshti. Solutions to the GSM security weaknesses. In *IEEE NGMAST*, 2008.
[49] P. Traynor, W. Enck, P. McDaniel, and T. La Porta. Mitigating attacks on open functionality in sms-capable cellular networks. *IEEE/ACM Transactions on Networking*, 17(1):40–53, 2009.
[50] Trojan Sends Premium-rate SMS Messages, Aims at European and Canadian Android Users. http://www.pcworld.com/article/245021.
[51] G.-H. Tu, C.-Y. Li, C. Peng, and S. Lu. How voice call technology poses security threats in 4g lte networks. In *IEEE CNS*, 2015.

APPENDIX

We study 40 popular SMS-powered services, which cover 17 distinct industries including grocery, bank, social network, retailing, *etc.*, at OP-I in the US. We discover that 28 out of 40 services (*e.g.*, Walmart, Target, Facebook, Fedex, *etc.*) are vulnerable to the IMS-based SMS attacks due to no runtime authentication (§5.1) or weak authorization (§5.2,§5.3). The detailed results are summarized in Table 2. The vulnerable services can be classified into two categories: notification-based and request-based. The notification-based service (*e.g.*, SMS advertising, flight status SMS notification, *etc.*) may suffer from the attack of unauthorized subscription (*e.g.*, the SMS advertising of The Home Depot in §5.3). The provider's goodwill may thus be impaired according to plenty of the victims' complaints. Second, the request-based service (*e.g.*, account manipulation request, donation request, *etc.*), which provides users with the ability to request service actions via SMS, may suffer from the attack of account hijacking (*e.g.*, Facebook account hijacking in §5.1), unauthorized donation (*e.g.*, unauthorized ARC donation in §5.2) or any other unauthorized use of the service.

No.	Provider	Industry	Short code	Service	W	S	Threat
1	Walmart	Grocery	63257	Notification	No	Yes	Unauthorized Subscription
10	CVS Pharmacy	Pharmacy	35437	Notification	No	Yes	Unauthorized Subscription
18	Costco	Grocery	71034	Notification	No	Yes	Unauthorized Subscription
21	JP Morgan Chase	Bank	24273	Request	Yes[a]	No	No
23	Bank of America	Bank	692632	Request	Yes[a]	No	No
28	Citi Bank	Bank	692484	Request	Yes[a]	No	No
30	Wells Fargo	Bank	93557	Request	Yes[a]	No	No
33	**The Home Depot** (§5.3)	Retailing	65624	Notification	Yes	No	**Unauthorized Subscription**
36	Target Baby	Grocery	827438	Notification	Yes	No	Unauthorized Subscription
36	Target Store	Grocery	827438	Notification	Yes	No	Unauthorized Subscription
41	State Farm	Insurance	78836	Notification	No	No	No∗
47	UPS	Courier	69877	Notification	Yes	No	Unauthorized Subscription
50	Lowes	Retailing	656937	Notification	No	Yes	Unauthorized Subscription
65	Fedex	Courier	48773	Notification	No	Yes	Unauthorized Subscription
70	American Airline	Airline	35922	Notification	No	Yes	Unauthorized Subscription
84	Safeway	Grocery	25374	Notification	No	No	No∗
88	American Express	Bank	692639	Request	Yes[a]	No	No
104	TimeWarner Cable	ISP	789789	Notification	No	Yes	Unauthorized Subscription
105	Macy	Store	62442	Notification	Yes[a]	No	No
134	Staple	Grocery	555444	Notification	No	Yes	Unauthorized Subscription
138	US Bank	Bank	872265	Request	Yes[a]	No	No
157	KOHL's	Grocery	56457	Notification	No	Yes	Unauthorized Subscription
161	SouthWest Airline	Airline	72743	Notification	Yes	No	Unauthorized Subscription
187	Starbucks	Retailing	22122	Notification	No	Yes	Unauthorized Subscription
194	Office Depot	Grocery	33768	Notification	Yes	No	Unauthorized Subscription
221	Marriott	Hotel	58682	Notification	Yes	No	No
242	**Facebook** (§5.1)	Social Network	32665	Request	Yes[a]	No	**Account Hijacking**
245	Toys R US	Toy	78697	Notification	No	Yes	Unauthorized Subscription
250	JC Penny	Store	527365	Notification	No	Yes	Unauthorized Subscription
260	Bed Bath Beyond	Grocery	239663	Notification	No	Yes	Unauthorized Subscription
303	Discover	Bank	347268	Notification	Yes[a]	No	No
648	A&F	Apparel	231892	Notification	No	Yes	Unauthorized Subscription
648	Abercrombie kids	Apparel	34824	Notification	No	Yes	Unauthorized Subscription
648	Hollister Co	Apparel	743722	Notification	No	Yes	Unauthorized Subscription
NA	Southern Class	Apparel	313131	Notification	Yes	No	Unauthorized Subscription
NA	Twitter	Social Network	40404	Request	Yes[a]	No	Account Hijacking
NA	Domino Pizza	Fast Food	366466	Notification	Yes	No	Unauthorized Subscription
NA	Paypal	ePayment	729725	Request	Yes[a]	No	No
NA	Papa John	Fast Food	47272	Notification	No	Yes	Unauthorized Subscription
NA	**Red Cross** (§5.2)	HumanAid	90999	Request	No	No[b]	**Unauthorized Donation**
Total	40	17					**28/40**

Table 2: Summary of 40 SMS-powered services and the threats which they face at OP-I. No. represents the ranking of Fortune 500 companies [17]. The columns of W and S represent web enrollment and SMS enrollment, respectively.

[a]The web enrollment requires users to login the website with their passwords.
[b]The service is automatically enrolled by carriers.