Exploiting MMS Vulnerabilities to Stealthily Exhaust Mobile Phone's Battery

Radmilo Racic
University of California, Davis
racic@cs.ucdavis.edu

Denys Ma
University of California, Davis
madl@cs.ucdavis.edu

Hao Chen
University of California, Davis
hchen@cs.ucdavis.edu

***Abstract*— As cellular data services and applications are being widely deployed, they become attractive targets for attackers, who could exploit unique vulnerabilities in cellular networks, mobile devices, and the interaction between cellular data networks and the Internet. In this paper, we demonstrate such an attack, which surreptitiously drains mobile devices' battery power up to 22 times faster and therefore could render these devices useless before the end of business hours. This attack targets a unique resource bottleneck in mobile devices (the battery power) by exploiting an insecure cellular data service (MMS) and the insecure interaction between cellular data networks and the Internet (PDP context retention and the paging channel). The attack proceeds in two stages. In the first stage, the attacker compiles a hit list of mobile devices — including their cellular numbers, IP addresses, and model information — by exploiting MMS notification messages. In the second stage, the attacker drains mobile devices' battery power by sending periodical UDP packets and exploiting PDP context retention and the paging channel. This attack is unique not only because it exploits vulnerable cellular services to target mobile devices but also because the victim mobile users are unaware when their batteries are being drained. Furthermore, we identify two key vulnerable components in cellular networks and propose mitigation strategies for protecting cellular devices from such attacks from the Internet.**

I. Introduction

Cellular networks are part of our critical information infrastructure. As mobile devices become more powerful, cellular companies are rapidly deploying broadband data services, such as High-Speed Downlink Packet Access (HSDPA) and Evolution-Data Optimized (EV-DO) as well as new applications, such as Multimedia Messaging Service (MMS). While these new services and applications enhance mobile computing experience, they also introduce serious security concerns. Besides launching typical Internet attacks — such as denial of service (DoS), malware, spamming and phishing — against mobile devices, an attacker can exploit emerging vulnerabilities in cellular networks, mobile devices, and the interaction between cellular data networks and the Internet.

In this paper, we demonstrate such an attack. The attack targets battery power, which is a critical but scarce resource on mobile devices. This attack would be devastating not only in critical situations, such as disasters, but also for industries relying on mobile communications. For example, professions like real estate agents and brokers rely on the ability to perform on-the-spot credit reports or provide instant quotes. Similarly, occupations such as network system administrators trust their cellular handset's availability in order to be reached. Moreover, the victim would not notice this attack until his or her phone's battery is completely drained, which is also likely the most inopportune time according to Murphy's law.

This attack exploits vulnerabilities in MMS (a cellular data service), PDP context retention (interactions between the Internet and cellular data networks), and the paging channel. Furthermore, this attack has unique features that (1) it is clandestine – victim mobile users will not notice when their batteries are being drained; (2) it is not limited to certain mobile device hardware or software; and (3) it targets individual mobile devices rather than the network, an attack that is often harder to detect and defend effectively by network operators.

We implemented this attack in two stages. In the first stage, we were able to build a fairly accurate "hit-list" of all the users with an active Internet connection by taking advantage of the insecure MMS protocol. In the second stage, we exploit the PDP context retention to surreptitiously drain a phone's battery up to 22 times faster than normal. This attack illustrates two key vulnerable components in the cellular data network, and we will propose mitigating strategies for securing these components.

The rest of this paper is organized as follows: Section II briefly describes cellular network architecture, in particular General Packet Radio Service (GPRS) and its mobility management scheme as well as MMS. Section III presents a battery draining attack, and Section IV describes our mitigation assessments. Section V presents related works on cellular security. Section VI describes our future work. Section VII concludes the paper.

II. BACKGROUND OVERVIEW

To help understand the vulnerabilities and attacks that we discovered, we present an overview of the relevant components in cellular networks: GSM, GPRS and MMS.

A. GSM

The key elements in GSM are: the Base Station Subsystem (BSS), which includes the Base Transceiver Station (BTS) and the Base Station Controller (BSC), and Mobile Switching Center (MSC) which is the core of the Network Sub System (NSS). Additionally, these GSM elements utilize databases like Home Location Register (HLR) and Visitor Location Register (VLR) for storing users' home as well as roaming information, respectively.

BTS provides the means to transmit and receive radio signals as well as encrypt and decrypt communication with the BSC. BSC provides network intelligence by allocating radio channels, controlling inter-BTS hand-offs and, most importantly, serving as a gateway to the MSC. MSC, on the other hand, sets up circuit-switched communications, takes care of mobility management and manages other databases.

A cellular network needs to keep track of the location of each Mobile Station (MS[1]) in order to deliver calls and data to the correct destination reliably. Typically, the network utilizes an event-based mechanism to collect mobile device's location. Events such as powering up, shutting down, and crossing into another location area are events that trigger the location update procedure.

A cellular network is partitioned into cells serviced by BTSs. Cells are then grouped together to optimize signaling and to facilitate tracking of mobile phones within the network. Each group, managed by one BSC, is identified by a location area code broadcast by each BTS at regular intervals. Two fundamental operations within the location area are *location update* and *paging*.

1) Location update: The MS sends location update messages to its current BTS periodically in order to route all incoming calls or data appropriately. If the MS sends updates seldom, its location is unknown and the MS must be paged for each downlink packet (or call), thus degrading the quality of service. If, on the other hand, the MS sends frequent updates and its location is known, then data packets can be delivered without any additional paging delay.

2) Paging: To minimize the amount of updates, preserve MS's battery, and minimize bandwidth utilization, the network will page the MS over the Paging Channel (PCH) to determine its location. In other words, PCH is used for communication from BTS to MS when MS is not assigned a traffic channel; that is, the MS's location is unknown or out of date.

The paging bandwidth burden is relatively small in small location areas - less than 1% of the bandwidth allocated for voice channels. On the other hand, in an area with a large number (over 1000) of cells per location area, the paging bandwidth burden could be considerably higher. [1]

B. GPRS

GPRS [2] is integrated into the existing GSM infrastructure with a new class of network nodes called GPRS Support Nodes (GSNs). GSNs are responsible for the delivery and routing of data packets to and from the mobile network. There are two types of GSNs: Serving GPRS Support Node (SGNS) and Gateway GPRS Support Node (GGSN). SGSN is responsible for transferring and routing of data packets, mobility management, logical link control, authentication and billing services within its service area. GGSN acts as an interface between the GPRS backbone and external packet networks (primarily the Internet). Its primary function is to convert GPRS packets coming from the SGSN to IP packets and vice versa.

Before an MS can utilize GPRS services, it must register with an SGSN so all packets can be routed through it. During this procedure, called GPRS attach, a PDP (Packet Data Protocol) context is created. In particular, SGSN checks if the user is authorized, copies the user profile from the HLR to itself, assigns a Packet Temporary Mobile Subscriber Identity (P-TMSI)[2], maps it to an IP address, and assigns a GGSN that will serve as the gateway to the Internet. The PDP context, composed of the above mentioned information, is stored at the SGSN. GPRS detach, on the other hand, disconnects the MS from the GPRS network and deactivates the PDP context.

Location areas have been proven to be efficient in voice networks; however, the bursty nature of data traffic increases the number of paging messages per phone in each location area. Therefore, each location area is further subdivided into routing areas used by GPRS to decrease the penalty for locating an MS. GPRS phones utilize IDLE, STANDBY and READY states in increasing order of battery consumption. When an MS is in the READY state, SGSN is aware of the MS's location. In particular, the MS performs frequent location updates to provide the network with the actual cell ID so that no paging is necessary. When in the READY state, the MS can send and receive data. Furthermore, it will stay in the READY state until READY timer expires, at which it will transition to the STANDBY state. While

[1]MS and phone will be used interchangeably.

[2]The reasoning is to minimize use of IMSI (International Mobile Subscriber Identity) for security purposes.

in the STANDBY state, the MS has established the PDP context and it can receive calls or data. However, its location updates are more coarse, in the sense that it informs the SGSN of only routing area changes, but not cell changes. If SGSN needs to deliver data to the MS while the MS is in the STANDBY state, SGSN will send a page request in the routing area where the MS is located. When MS responds to the page, it will transition to the READY state. IDLE state is the lowest battery consumption state, in which the SGSN is not aware of the MS's location. The MS can transition out of IDLE state only if it performs a GPRS attach procedure. Alternatively, an MS could initiate a GPRS detach procedure to transition to the IDLE state. Figure 1 shows the state machine of the GPRS MS.

Upon completion of the communication, the MS will go into a STANDBY mode. The PDP context, on the other hand, *will remain allocated to the MS*. We conducted experiments to discover how long each handset retained its assigned PDP context and IP address. We found that addresses seemed to be relinquished in as short as 15 minutes to as long as several hours. The reason for not deactivating a PDP context is simple: a cellphone can be unavailable for a period of time due to radio link failure; deactivating and activating a new context would imply that the phone would need to recreate all TCP sessions, possibly restarting applications and requiring the user to re-enter all the passwords.

C. MMS

MMS has become a very popular cellular message service. The MMS architecture spans both the cellular network and the Internet and uses technologies in both networks, such as WAP, SMTP, and HTTP.

The MMS architecture consists mainly of the MMS Relay/Server (MMS R/S) and user agents. Several optional entities of the architecture – the billing server, the Home Location Register, and the User Database — may exist inside or outside MMS R/S. Figure 2 shows an overview of the MMS architecture.

The MMS R/S is responsible for all of the transactions of MMS. When a user transmits an email or an MMS message, the mobile phone formats these messages in Synchronized Multimedia Integration Language (SMIL) [3]. The MMS R/S translates (transcodes) the message to either email or different MMS formats depending on the provider. The message is then sent to the destination SMTP mail server or the destination MMS R/S using SMTP. Upon receiving the message, the destination MMS R/S then stores the message in the user's buffer while sending a notification message to the user via a SMS or WAP push message. The notification message contains the location of the message, usually specified as an HTTP address. User can configure their mobile phones either to automatically download the message upon receiving the notification or to manually download the message themselves.

III. Attacks

In this section, we present our findings on attacking the cellular network. We first investigated the MMS protocol and discovered several vulnerabilities through which we leveraged into the heavily protected cellular network. Then, by exploiting these vulnerabilities, we implemented a proof-of-concept attack on a scarce resource – the battery power – of mobile devices. The attack is stealthy, as it is noticeable to neither mobile users nor network operators. Our experiments demonstrate that unique threats against cellular networks and mobile devices exist and are exploitable. Finally, we discuss how to make this attack even more effective.

A. MMS security analysis

To test how cellular providers implement MMS and gain insight into their interface designs, we setup our own MMS R/S, based on an open-source project [4]. We discovered several vulnerabilities that a wily attacker could exploit, as described in the following sections.

1) Unencrypted and unauthenticated MMS messages: We confirmed that MMS messages and MMS notification messages, composed of headers and content sections, were sent in plain-text. In addition to the SMIL headers, the packet also included an HTTP POST header containing the source and destination IP address, the profile of the user agent, the content type and size, and the user agent name.

2) Unauthenticated MMS R/S: To mitigate the problem of unencrypted messages, cellular providers hide their own MMS R/S's IP addresses in the phones, hoping that cellular users cannot read or overwrite them. Unsurprisingly, we discovered that this attempt at *security by obscurity* is broken.

In order to inspect the MMS message raw format, we modified a phone's firmware to route all MMS messages through our MMS R/S. The MMS R/S setting is well hidden in our phone's firmware, which suggests that providers do not intend to allow users to modify the setting. After modifying the MMS R/S entry in our phone, we discovered that the phone had no security mechanism to alert the new, unauthorized MMS R/S. Furthermore, MSs also do not authenticate MMS notification messages and MMS messages sent from the network. MSs will accept any MMS messages as long as the format is correct. Consequently, we were able to send unlimited MMS messages for free, without alarming the cellular provider.

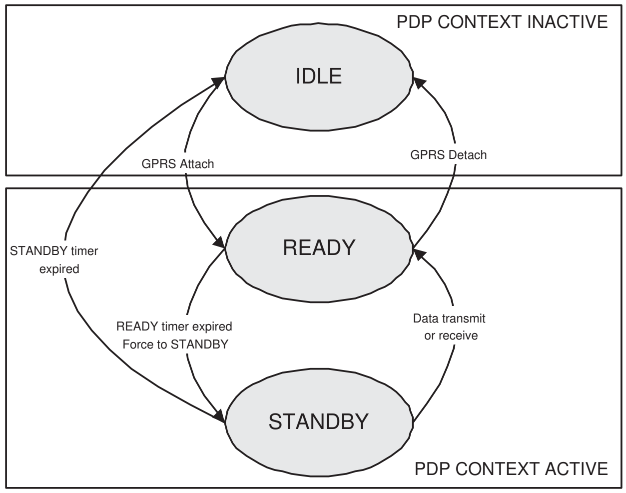

Fig. 1. The GPRS mobile station state machine

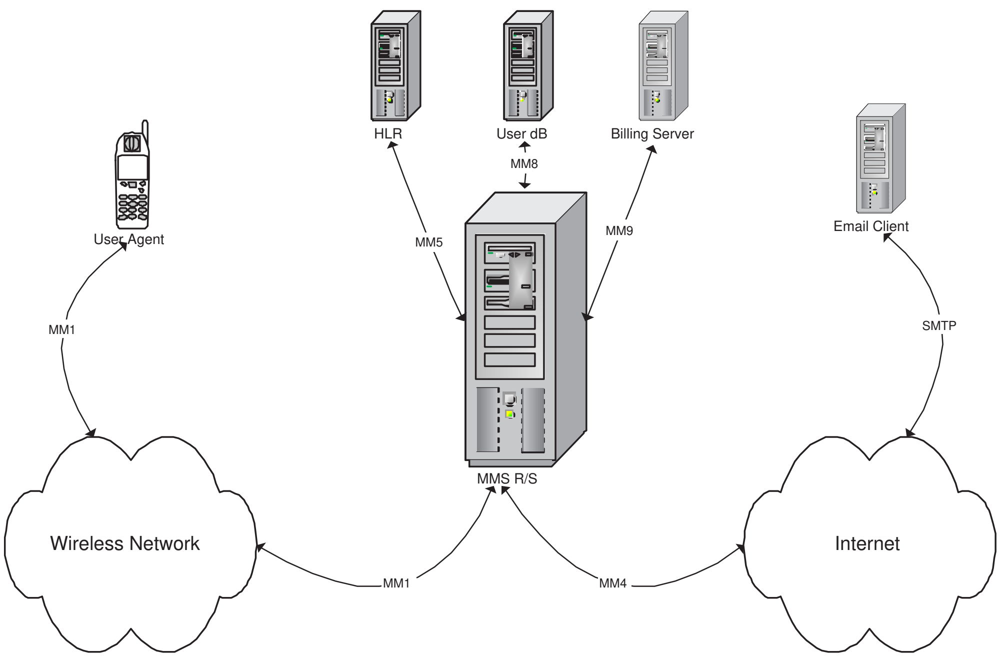

Fig. 2. MMS Infrastructure

3) Critical phone information disclosure: We discovered that handsets include pertinent user agent platform information whenever they communicate over HTTP. Accordingly, we set up a web server running *ethereal* to capture HTTP requests from various handsets on different networks. We found that every phone disclosed either its full profile or information that included one or more of the following: hardware platform description, display capabilities, and the current and compatible software. An attacker could write a script that extracts the model number of each handset very easily.

B. Attack implementation

Based on our MMS security evaluation, we implemented a battery draining attack utilizing a hit-list built using superfluous but pertinent information disclosed during MMS exchanges. Figure 3 illustrates the attack.

1) Building target hit-list: To launch effective, large scale attacks, an attacker needs to build a hit-list that contains important information about the network and end users. One way to obtain such information is by asking the mobile phones.

An attacker can send MMS notification messages, whose content address is at a malicious web server, to

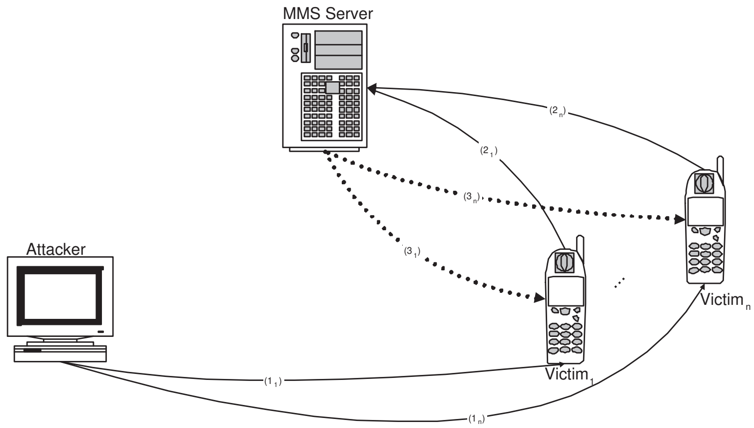

Fig. 3. A two-step attack on cellular devices. In Step 1, the attacker builds a hit list using MMS message notifications (Messages (1)s), and captures information about mobile users from the HTTP requests from mobile users (Messages (2)s). In step 2, the attacker drains the batteries of cellular devices on the hit-list surreptitiously by sending UDP packets (Messages (3)s) periodically to the cellular devices.

numerous recipients. The target phone numbers can be generated automatically using known area codes and prefixes for cellular phone numbers. The MMS notification messages can be sent using SMS or WAP push. There are many free SMS messaging websites, including those offered by cellular providers.

Once MMS notification messages are sent, the attacker waits for HTTP request messages at his web server, which has stated its location in the MMS notification message. Since many cell phones are configured to download MMS messages automatically upon receiving notification, they will make HTTP requests to the attacker's web server. The HTTP requests often contain the profiles and IP addresses of the phones, and even file extensions that the phones are able to process. By sending a slightly different URL to each phone, the attacker can build a hit list that maps each phone number to a profile of its cellular device. More importantly, the phone's response to the MMS notification message activates a PDP context, making our attack easy and simple to execute even in the presence of NAT and firewalls.

2) Draining batteries: Using the hit-list generated from MMS notification messages, an attacker can target the cellular network and cellular devices more precisely and effectively. Apropos, we implemented a battery draining attack that focuses on the end hosts instead of the network. We implemented our attack using UDP packets (we will explain an improved technique later.)

The key to maximizing a cell phone's battery life is to use its transceiver sparingly. In fact, when a cellular phone is turned on, its transceiver is active less than 3% of the time. As a reference, in wireless sensor nodes, transmitting one bit of information consumes 1500 to 2700 times as much energy as executing one instruction [5]. Thus, if a packet is sent to a phone, the SGSN will deliver the packet if the phone's location is known, or attempt to locate the phone by sending a page request to it. However, since cellular phones spend most of their time in the STANDBY mode (or other dormant modes), the page on the paging channel will awaken the phone to the READY state and force it to perform a location update. The sine qua non of this attack is to keep the phone in the READY state (high battery consumption), therefore disabling its ability to preserve battery life, or to let the phone temporarily go into the STANDBY state only to be immediately awakened with a page and forced to perform a location update; both of these actions consume much energy.

3) Theoretical impact: To investigate the severity of the aforementioned attack, we estimate the damage that an attacker with a home DSL Internet connection can inflict. A typical DSL upload speed ranges from 256kbps to 416kbps. We use the medium speed, B = 384kbps, for the upload bandwidth as an estimate. Each UDP packet consists of a character in the data segment, which might be padded to 4 bytes depending on the provider's DSL modem. The UDP packet header has 8 bytes, and the IP header has 20 bytes. In the pessimistic estimate where our data is padded, the total size of the packet is S = 32 bytes. Therefore, the maximum number of UDP packets per second that an attacker may send is $(B/8)/S = 1500$.

To attack a phone effectively, an attacker must send one UDP packet to the phone every T seconds. In this case, the maximum number of phones that the attacker can attack simultaneously is $(B/8) * T/S$. We estimated the time T by trial and error using different test configurations. For our experiment, we chose 3.75 seconds for the GSM-based network and 5 seconds for the CDMA-based network. Using our equation, we calculated that an attacker can attack about 5625 phones using a standard ADSL line for a GSM-based network and around 7000 phones for a CDMA-based network.

C. Attack experiment results

We successfully drained our test phones' batteries considerably faster than our average usage. We conducted six test runs on a high-end Nokia smart phone and completely drained its battery in an average of 7 hours, instead of 156 hours in normal usage with bluetooth switched off most of the time. We also observed severe battery exhaustion in our Sony Ericsson test phone, where the battery was drained down to 20% within less than 7 hours without talking and with bluetooth switched off. If a phone is connected to the Internet continuously (for example, to use the instant messaging service), its battery life would be reduced much faster. To test this hypothesis, we attacked our Motorola test phone while connecting it to the Internet continuously. Our test completely drained its battery within 2 hours. Table I summarizes the results of our attack.

We successfully conducted our attack on two major cellular service providers without triggering any alarms. Our test machine's IP was not blocked, our phones were fully operational after the attacks, and no notifications or warnings were sent to us regarding this issue. Moreover, during the attack the phone appeared to be operating normally and no additional Internet application was started, so the victim user would not notice the attack, until his/her battery died unexpectedly.

D. Attack improvement

There are several optimizations that could be done to improve our attack. Currently, we empirically determined a fixed interval between each UDP packet by trial and error. However, by using Qualcomm's CAIT software or knowing the implementation of a particular cellular network, we could obtain more accurate wait-time and thereby improve efficiency of our attacks. Also, knowing which IP addresses are vacant would increase the efficacy of our hit-list creation. We are currently in the midst of testing the following improvements to our attack.

1) Attack using TCP ACK packets: To force a phone to send as well as receive useless data, an attacker can periodically send TCP ACK packets to the phone's IP address. In accordance with RFC793, if the connection is reset or in half-open state, the receiver of an out-of-order ACK packet will send an RST packet. If, on the other hand, the connection is open, the receiver of an out-of-order ACK packet will reply with an empty packet. Either way, an attacker will force a phone that implements a full TCP stack to receive as well as send packets, thereby exacerbating the power consumption.

2) Attack using packets with maximum-sized payload: In implementing our previous attack, we used UDP packets with no payload in order to maximize the number of UDP packets an attacker can send per computer. However, this is not the most efficient method of draining a cellular phone's battery, since the whole packet must be downloaded to the mobile phone before the phone can discard the packet. Therefore, with an accurate hit-list collected using MMS above, the attacker can sacrifice the number of targets per his/her computer to deliver an even more efficient attack using a maximum-sized payload.

Using the original attack implemented with UDP, the attacker can send a maximum theoretical UDP data packet of 64Kb due to its 2 byte total length field. In the TCP variety of the attack, ACK messages "piggyback" onto the existing payload with a maximum size of 1500 bytes. Besides causing additional unnecessary downloads for the mobile agent, the attack could possibly be even more efficient due to packet fragmentation. This exacerbates the attack so that the attacker would only need to send a single packet that becomes multiple packets at the mobile agent.

3) NAT and firewall: Through field experimentation, we have determined that most providers who utilize NAT also implement Network Address and Port Translation (NAPT.) NAPT provides dynamic *(privateIP, privatePORT)* to *(publicIP, publicPORT)* translation. For example, the inside interface tuple $(10.0.0.5, 3000)$ could be mapped to the outside interface tuple $(199.156.3.4, 6000)$.

However, there are certain issues with network-wide NAT deployment. For example, it often hinders application deployment. Additionally, certain security protocols such as IPSec and Kerberos are affected – NAT changes the address in the IP header, causing loss of integrity. For these reasons, operators choose to implement NAT only on certain subnets affecting a selected customer base. In other words, most operators offer both private and public IP plans.

It would seem that our attack could be mitigated with NAT and firewall placement. However, a very simple restriction to the attack could yield the same result. The crux of the change would be an observation that each inside IP address maps to a port on the outside interface because the *publicIP* is the public IP address of NAT system. Thus, targeting an inside IP address reduces to targeting a certain port of the outside interface. Since NAPT does address and port translation dynamically, the IP address and port mappings are only alive during active PDP contexts. Thus, the attack must be delivered within an active session window. Since phones automatically create an outbound connection to connect to a malicious HTTP server, the server itself must deliver the attack,

Phone	Battery Life Without Attack		Battery Life Under Attack	
	Normal Use (hours)	Standby (hours)	Normal Use (hours)	Reduction
Nokia 6620	156	200	7	22.3:1
Sony-Ericsson T610	60	315	7	8.6:1
Motorola v710	36	150	2[a]	18.0:1

[a]Used as a wireless modem.

TABLE I
REDUCTION OF BATTERY LIFE DUE TO OUR ATTACK

thus prolonging the connection. The firewall would consider this connection valid as it is internally initiated over allowed ports, and NAT would continue the address and port translation for the duration of the attack.

IV. Mitigation strategies

Our attack uncovered two vulnerable components in cellular networks.

- *PDP Context is retained.* We observed that a mobile user's PDP context is kept alive even after the user has completed his/her data session. The PDP context may be kept active from 15 minutes to several hours, depending on the service provider. This active PDP context allowed us to send unwanted IP packets to the victim's mobile phone to drain its battery.
- *Attack packets are not in any active session.* Our attack periodically sends packets to mobile user without an active connection. A mobile user must initiate active connections before he receives data. Since the GGSN records the connection states, it can distinguish attack packets from normal packets that belong to active connections, unless the attacker can guess the correct sequence number, destination IP address and port number of an active connection.

Based on these observations, we propose additional security mechanisms in GPRS and MMS. Firewalls and IDSs are common mechanisms for defending against malicious behavior from the Internet, but they have several disadvantages: (1) firewalls and IDSs become the single point of failure, (2) they are external entities, and they usually do not protect against insider attacks, (3) they are not flexible enough to dynamically adapt to traffic conditions without system administrators – they require knowledgeable administration staff, (4) they are not suitable for monitoring peer-to-peer (such as Bluetooth) communication, and (5) they cannot protect against attacks exploiting insecure protocols whose action is seemingly valid – they either allow or deny a connection. To mitigate threats against MMS, we propose a redesign by incorporating security mechanisms into the protocol.

- *Message and server authentication*. To avoid man-in-the-middle attacks, we should authenticate MMS messages and R/Ss, using PKI for instance.
- *Information hiding at WAP gateway*. WAP gateway should prevent outside web servers from obtaining critical information about mobile devices, such as their IP addresses, and hardware and software profiles. Since profiles are used only by the WAP gateway for converting web contents, the WAP gateway should filter out all but essential information about the user agent in HTTP requests.
- *MMS message filtering*. Service providers typically hard-code their approved MMS R/S into mobile devices' OS or firmware to prevent users from choosing alternative MMS R/Ss. However, sophisticated users can modify their OS or firmware to defeat this protection. A more reliable approach for service providers is to filter MMS messages, since all MMS packets must traverse the provider's network. The filter can scan MMS message headers to ensure that the destination IP address is one of the MMS R/S or accredited third party Value Added Service (VAS) providers. The filter should not be implemented at the WAP gateway, but rather at the SGSN or GGSN, since users can easily modify the phone's settings and bypass the cellular provider's WAP gateway.
- *Improved PDP context management*. To detect and mitigate attacks that could stealthily bypass current security mechanisms in GPRS, we suggest a defense framework that could avoid the shortcomings of external firewalls and IDSs mentioned above by supplementing these protection mechanisms:
  - This defense mechanism can also serve as an event detector for IDSs already in place to monitor the internal network.
  - It must also be effective against insider attacks, where malicious users are connected using the cellular network instead of the Internet.
  - It should be designed with the goal of being non-intrusive so that it does not require ancillary network infrastructure; it should utilize existing GPRS mechanisms to provide an ad-

ditional layer of protection.

Such defense strategy is best implemented on GGSN. Since service providers already perform some proprietary PDP management scheme, as we discovered empirically, implementing a defense scheme would be straightforward. Furthermore, as most of the functions needed have already been implemented (such as the gateway assisted PDP context modification function), we expect light additional implementation work.

We propose to modify the PDP context management scheme. The implementation of this scheme can be transparent to mobile devices, and the mapping can be done entirely at GGSN. Since GGSNs are already stateful, a simple change in IP address assignment would not be difficult. Furthermore, a modification to the PDP context would also provide a NAT-like behavior, as each IP address can be assigned multiple times using different ports. The scheme should not require any user interactions nor any infrastructure alterations. The defense mechanism automatically adjusts to each user's PDP context management algorithm based on their usage. We hope that this improved scheme will lay the foundation for a comprehensive defense system, which can incorporate more detection and response strategies.

V. Related work

In recent years, significant amount of research efforts have been focused on security requirements and threat model evaluation on current and emerging cellular technologies, including GSM [6]–[8], GPRS [9]–[12], CDMA [13], SMS [14], MMS [15], and EVDO [16]–[18]. These works identify the following key security requirements in cellular networks: subscriber confidentiality, authentication, privacy, cloning prevention, integrity of information as well as billing, fraud detection, and safe key management. These works also address security threats such as eavesdropping, impersonation of a user and network, denial of service, man-in-the-middle attacks, hijacking services, and compromising authentication vectors. Apropos, researchers evaluated the risk levels of each of these threats as well. Our work is complementary to these previous efforts to secure cellular networks. In fact, we focus in two new directions: the end user devices (i.e., power-depletion attack and defense) and the security interactions between different cellular applications (i.e., the merging of cellular network and the Internet). We also propose new defense mechanisms based on existing cellular infrastructure.

Extensive research has been conducted on the cryptography technologies [19]–[21]. For instance, studies like [19], [20] suggest the use of a PKI scheme in the GSM/UMTS network while [21] proposes the use of a SIM card for authentication and payment of web services by mobile users.

Cryptographic solutions, while efficiently and elegantly mitigating some principal concerns in cellular networks, cannot defend against some unique threats to end users, such as a DoS attack and resource starvation attacks. Our work complements the existing cryptography mechanisms in order to alleviate additional non-conventional threats unique to emerging cellular data technologies and applications.

In addition to cryptographic solutions, schemes are also developed to defend against cloning and fraud, such as device and user fingerprinting [22], mobility pattern recognition [23], and usage pattern recognition [24], [25]. These research studies propose new security mechanisms strictly for cellular networks. However, most studies stipulate fundamental changes in either architecture or end user equipment. In order to minimize disturbance of current implementation of cellular networks, our research will focus on utilizing existing security mechanisms, such as PDP context modification, to mitigate new attacks that were not discovered or considered.

Despite the significant efforts on threat assessments in cellular networks, many attacks on cellular architecture were discovered. Guo et al. discussed possibilities of mounting attacks on smart-phones after they have been compromised [26]. They envisioned attacks such as DoS, spamming, identity theft, and wiretapping, and sketched several defense strategies for mitigating these attacks. Inspired by their vision, we examined the implementation details of MMS, discovered exploitable vulnerabilities, and launched real attacks. Furthermore, our attacks apply to commodity phones with MMS capabilities while the authors focused on smart phones only.

DoS attacks also attracted a lot of attention because resources in cellular networks are much more limited than on the Internet. Furthermore, control channels are particularly vulnerable. Agarwal et al. [27] conducted a capacity analysis of shared control channels used for SMS delivery. They concluded that increasing volume and message sizes can significantly affect network performance. Apropos, Enck et al. [28] presented a DoS attack based on that idea by sending a sufficient number of SMS messages per second to a range of cellular phones in the same area. An attacker would need only a single computer with a broadband network access in order to disrupt a network in a major city by saturating control channels shared between voice calls and SMSs. Additionally, Mutaf et al. warns that paging channel is another scarce resource that an attacker on the Internet can overwhelm and cause a DoS attack [29]. While their work disrupts network availability, our work focuses on attacking the poorly protected end hosts. Furthermore, our battery attack (Section III-B) is clandestine in stark contrast to the SMS attack, in which users were well aware of the SMSs being received.

In addition to DoS attacks, spam is another well-known problem in the SMS network [14]. Network providers allow email and web-based interfaces to send SMS messages to individual or multiple handsets directly. Spammers can also employ phishing [30] to trick users into divulging private personal information. SMS-based phishing has already been discovered in a small German cellular provider [31], where users are tricked into sending a reply SMS to a value-added service's SMS number, charging a small fee per user. Our approach in Section III-B of building a hit-list of phone IP addresses and model information was inspired by phishing; however, our approach does not need the user's participation or even attention, because such information is reported to our server automatically by most phones.

Computer worms that target cellular networks have also appeared in recent years. Timifonica worm [32] spreads itself via email attachments. Upon infection, a computer sends SMS messages to random cell phone numbers belonging to a service provider, Movistar, and thus attempts to cause a DoS attack. A proof of concept worm was developed in early 2005 demonstrating the effects of a worm outbreak on cellular phone platforms. The Cabir [33] worm, spreading via Bluetooth on Nokia series 60 handsets running Symbian OS, changes the operating system and searches for other handsets to infect. An epidemic worm spreading model in mobile environments was proposed by Mickens et al [34]. Our work is an extension to these previous works. Using a hitlist of phone numbers, IP addresses, and model information gathered in our attack described in Section III-B, worm designers could write better worms by tailoring to different platforms.

VI. Future work

We understand that there is much to be done on securing the cellular network. Securing the current messaging systems (SMS and MMS) and developing a new secure messaging system will be our primary focus. In particular, we will develop techniques to secure these messaging systems to prevent attackers from leveraging into the cellular networks from the Internet.

Many refinements and validation of our attacks are available. An attractive target for the attacker is the billing server that resides in the MMS R/S. We have found a potential vulnerability that can not only circumvent the billing server, but also poison the billing records as well. We are investigating additional attacks on MMS messages, such as identity theft, spam, and phishing. Also, additional testing is needed to accurately determine the time period between each UDP packet in order to maximize damages caused by our attack. We plan to conduct more research using Qualcomm's CAIT software and an aircard, which can provide us with more insights into the cellular network. Finally, we plan to conduct more experiments with our malicious MMS R/S to discover vulnerabilities in the MMS application on handsets. We will also investigate a possible worm deployment targeting vulnerabilities found on handset applications using MMS. Furthermore, host-initiated battery draining attacks are also possible. When a phone is in a low reception area it boosts its transceiver signal strength to its maximum in attempt to get reception. A modification to the phone's OS to always run the transceiver at maximum signal strength would be disastrous for the battery life of the phone. This action can be automated with a worm.

Finally, a battery draining attack is one of the many unique threats to the cellular networks. New mechanisms and improvements to the current state of firewalls and IDSs should be developed to enhance detection of these new attacks. Our assessments on the current implementation of the cellular system against new threats should serve as a start in re-designing or improving these new mechanisms to detect and remove these new threats.

VII. Conclusion

While cellular users embrace new broadband data services and applications, attackers get opportunities to exploit emerging vulnerabilities in mobile devices, cellular data networks, and the interaction between cellular networks and the Internet. We demonstrated such an attack, which is able to drain mobile devices' battery power as much as 22 times faster. This attack proceeds in two stages. First, the attack exploits vulnerabilities in MMS to build a hit list of mobile devices. Then, the attack exploits PDP content retention and the paging channel to drain mobile devices' battery power. We were able to drain batteries without alerting either the mobile user victims or the cellular network operators. Our analysis shows that an attacker would need only several home DSL Internet connections to mount a large scale attack against a large number of cellular phones. We identified key components in cellular networks that enable this attack and proposed corresponding mitigating solutions.

Due to the complex interaction between mobile devices, cellular data networks, and the Internet, we conjecture that our discovered attack may be just the tip of an iceberg. We hope that our work will bring attention to this emerging threat and will inspire future research for securing cellular data services and applications.

Acknowledgment

We thank Xin Liu at UC Davis, and Hui Zang, Tao Ye, and Jean Bolot at Sprint Labs for their valuable comments and discussions on this research.

References

[1] C. R. C.U. Saraydar, "Minimizing the paging channel bandwidth for cellular traffic," in *IEEE ICUPC*, 1996.

[2] P. McGuiggan, *GPRS In Practice: A companion to the specification*. John Willey & Sons, 2004.

[3] W. P. Recommendation, "Synchronized multimedia integration language (SMIL2.1)," http://www.w3.org/TR/SMIL2/.

[4] Humpa, "MMS pic server," http://www.humpa.com.

[5] C. S. Vijay Raghunathan, Saurabh Ganeriwal and M. Srivastava, "WFQ: An energy efficient fair scheduling policy for wireless systems," in *ISLPED*, 2002.

[6] C. Brookson, "GSM (and PCN) security and encryption," http://www.brookson.com/gsm/gsmdoc.htm.

[7] P. Yousef, "GSM-security: a survey and evaluation of the current situation," Master's thesis, Linkoping Institute of Technology, 2004.

[8] C. Peng, "GSM and GPRS security," in *HUT TML*, 2000.

[9] A. Bavosa, "GPRS security threats and solution recommendations," http://www.juniper.net/solutions/literature/white_papers/200074.pdf.

[10] C. Brookson, "GPRS security," http://www.brookson.com/gsm/gprs.pdf.

[11] O. Whitehouse, "GPRS security: Not ready for prime time," http://www.securitymanagement.com/library/wireless_tech0902.pdf.

[12] S. piot, "Security over GPRS," Master's thesis, University College London, 1998.

[13] C. Wingert and M. Naidu, "CDMA 1XRTT security overview," http://www.cdg.org/technology/cdma_technology/white_papers/cdma_1x_security_overview.pdf.

[14] A. A. Khan, "Security and vulnerability analysis of wireless messaging protocols and applications," in *Pak Con*, 2004.

[15] S. Andersson, "MMS security considerations," in *3GPP TSG SA WG3 Security*, 2003.

[16] R. Safavi-Naini, W. Susilo, and G. Taban, "Towards securing 3G mobile phones," in *The 9th IEEE International Conference on Network (ICON 2001)*, 2001.

[17] 3rd Generation Partnership Project, "3G security: Security threats and requirements," ftp://ftp.3gpp.org/Specs/2000-12/R1999/21_series/21133-310.zip.

[18] O. Whitehouse and G. Murphy, "Attacks and counter measures in 2.5G and 3G cellular IP networks," http://www.atstake.com/research/reports/acrobat/atstake_cellular_networks.pdf.

[19] S. I. M. Constantinos F. Grecas and I. S. Venieris, "Introduction of the asymmetric cryptography in GSM, GPRS, UMTS, and its public key infrastructure integration," in *Mobile Network and Applications*, 2003.

[20] C.-C. Lo and Y.-J. Chen, "A secure communication architecture for GSM networks," in *IEEE Transactions on Consumer Electronics*, 1999.

[21] J. A. MacDonald and C. J. Mitchell, "Using the GSM/UMTS SIM to secure web services," in *the 2nd Workshop on Mobile Commerce and Services WMCS*, 2005.

[22] M. J. Riezenman, "Cellular security: better, but foes still lurk," *IEEE Spectrum*, vol. 37, no. 6, 2000.

[23] B. Sun, F. Yu, K. Wu, and V. Leung, "Mobility-based anomaly detection in cellular mobile networks," in *2004 ACM workshop on Wireless security*, 2004.

[24] M. S. M. A. Notare, F. A. da Silva Cruz, B. G. Riso, and C. B. Westphall, "Security management against cloned cellular telephones," in *IEEE International Conference on Networks*. Washington, DC, USA: IEEE Computer Society, 1999, p. 356.

[25] A. Boukerche and M. S. M. A. Notare, "Behavior-based intrusion detection in mobile phone systems," *Parallel and Distributed Computing*, vol. 62, no. 9, pp. 1476 – 1490, 2002.

[26] C. Guo, H. J. Wang, and W. Zhu, "Smart-phone attacks and defenses," in *HotNets III*, Nov. 2004.

[27] N. Agarwal, L. Chandran-Wadia, and V. Apte, "Capacity analysis of the GSM short message service," in *National Conference on Communications*, 2004.

[28] W. Enck, P. Traynor, P. McDaniel, and T. L. Porta, "Exploiting open functionality in sms-capable cellular networks," in *12th ACM Conference on Computer and Communications Security (CCS'05)*, Nov. 7-11, 2005.

[29] P. Mutaf and C. Castelluccia, "Insecurity of the paging channel in the wireless internet: A denial-of-service attack that exploits dormant mobile ip hosts," in *3rd Workshop on Applications and Services in Wireless Networks*, 2003.

[30] A. P. W. Group, "What is phishing and pharming?" http://www.antiphishing.org/.

[31] Redteam, "Advisory: o2 germany promotes SMS-phishing," http://www.redteam-pentesting.de/advisories/rt-sa-2005-009.txt.

[32] B. Fonseca, "Worm calling," http://www.infoworld.com/articles/hn/xml/00/06/06/000606hnphoneworm.html.

[33] D. Ilett and M. Hines, "Skulls program carries cabir worm into phones," http://news.com.com/Skulls+program+carries+Cabir+worm+into+phones/2100-7349_3-5469691.html.

[34] J. W. Mickens and B. D. Noble, "Modeling epidemic spreading in mobile environments," in *WiSe '05: Proceedings of the 4th ACM workshop on Wireless security*. New York, NY, USA: ACM Press, 2005, pp. 77–86.