Chapter 5

Taxonomy of Mobile Malware

Solutions in this chapter:

- **Infection Strategy**
- **Distribution**
- **Payload**

- ☑ **Summary**
- ☑ **Solutions Fast Track**
- ☑ **Frequently Asked Questions**

Introduction

With the increasing pervasiveness of computer viruses targeting mobile devices, taxonomy of known samples is needed to make some sense of what we have seen and what we may soon see. The taxonomy will be based on infection strategy, distribution, and payload. Each of these characteristics will be used to place each sample in the taxonomy for the purpose of illustrating which areas of mobile devices are most used by attackers to enter, control, and exploit the devices' systems. This can offer insight into future attacks and allow proper prevention by protecting areas highly used by current malicious code targeting mobile devices.

The current new virus wave targeting mobile devices has evolved at a much faster pace than viruses for desktop computers. The nature of a mobile device—ergo, its mobility—has required mobile malicious code (MM) to principally employ wireless and synchronization technologies to infect these devices. Bluetooth, e-mail, SMS, and Device-to-PC synchronization (D2P) have been the main tools used to infect and distribute MM into a device, between devices, and from devices to desktops. These infection strategies have rewritten the rules of how MM work and raised the bar on how to detect them.

Aside from infection, MM has also used both some old and some new tactics for distribution. Mainly, distribution amongst mobile devices has been the norm to date. Only a handful of MM, most being proof-of-concept code, have attempted distribution to other non-mobile devices. Principally Bluetooth, removable media, e-mail, D2P, and SMS have been the main tools to achieve this effort. One queasy effect of this is the problem of tracking wireless distribution to a source of initiation. Many a MM researcher has spent sleepless nights attempting to trace the distribution of these viruses due to the ease at which they can travel incognito across wireless channels. A more troubling issue is the bad actor using a mobile device to launch an MM and then destroying the device. This seemingly creates a faceless attacker that is never to be traced or identified. This form of attack with MM is predicted to increase in the coming years.

When viruses for desktops first appeared, the focus was mainly on infection and distribution with the payload being a sideshow. Since then, the evolution of malware in general has made payload the key factor, with infection and distribution becoming efficient B-52 bombers, attacking as many computers as possible and releasing their deadly payload at each stop. In the land of MM, payload has been a key component, being included in the very early pioneering samples, and today performs everything from file deletion to remote access to data farming. Of these, the collection of data for malicious use is the most troubling, given the high amount of sensitive information kept in mobile devices and the ease with which they can be attacked and exploited.

The taxonomy presented in this chapter is an initial attempt to bring order to what has already been achieved by MM and a glimpse of what is to come. The taxonomy is by default incomplete since the nature of MM and their authors is constantly evolving and delving into new yet unseen areas in the eternal pursuit of new and improved MM with innovative payloads and functionalities.

Infection Strategy

The initial introduction of a virus into a system is the essential step that must always succeed for the virus to do its dirty deeds. If a virus fails to infect the system, it cannot succeed within that system. In the world of MM, the means to which infection is achieved is spread across all the newly created and popular forms of communication. All the known wireless forms of communicating, including Bluetooth and MMS, plus removable storage such as memory cards, have all been used by MM authors to infect mobile devices. This critical step in the execution of MM is a key factor in analyzing how MM has infected mobile devices up to now and provides a glimpse of what could be next.

Creating a taxonomy based on infection strategies for viruses is not new. Previous malware taxonomies have all used infection as the main taxa of their systems and are well documented as to the hierarchy of types that exist in this area. MM introduces a hierarchy of taxa types that were previously grouped with many others but that now stand alone. Primarily, wireless forms of communications used by mobile devices along with removable storage media and Device-to-PC (D2P) synchronization are the main subtaxa in this hierarchy. This taxon is the root of a hierarchy that produces two subtaxa: wireless and wired. Each of these has a group of specific subtypes used by MM for infection of mobile devices. The balance of this section will focus on these subtypes, providing an explanation of their use by MM and the names of specific MM belonging to each.

Wireless Communication

Since the inception of the cell phone, wireless communication has become the mainstream form of communication for individuals around the world. The handheld device offers a cornucopia of wireless connectivity options from Wi-Fi to Bluetooth to infrared. Of course, as these technologies emerged and achieved widespread use, MM exploiting these connectivity options started emerging. Every wireless communication channel represents a possible entry of infection for MM onto the handheld device. Although the most common form of infection using wireless communication is into a handheld device, the real threat is in using wireless and a handheld to send an MM out. This form of use protects the bad actor, allowing invisibility while releasing dangerous malicious code into the wild. The following subtypes represent the novel wireless communications most commonly used by handheld devices today. For each subtype, the technology is briefly explained, followed by a list of the major known MM categorized in the subtype and a description of the MM's use of the technology.

MMS

An acronym for Multimedia Messaging Service, MMS is an enhancement to SMS (explained next), which allows the sending of multimedia objects such as images, video, audio, and enhanced text in addition to plain-text messages. Currently, with a camera and microphone installed in every modern mobile device, sending multimedia via MMS in mobile devices is

becoming a fast-growing phenomenon, slated to be the standard attachment to a text message. Infecting a mobile device using MMS has so far occurred in two specific ways: first by using the MMS to carry a copy of a MM to infect a device and second by the MMS itself containing code that exploits vulnerability in targeted devices. Both of these have been seen both in the wild and as zoo samples.

In 2005, the MM SymbOS.CommWarrior.A was discovered and labeled the first worm that propagated via MMS. It also propagated via Bluetooth. The MM targeted cell phones running the Symbian series 60 operating system. Originating in Russia, CommWarrior would attach a copy of itself to an MMS message as an infected Symbian archive file (SIS) attachment named commw.sis, which was sent to all contacts listed in the infected device's address book. The two other variants of CommWarrior—B and C—also propagated in the same manner. There was no payload, but the fear was the high speed at which the MM could spread using MMS. This propagation was similar to classic e-mail worms, which are known spread greatly in just a few minutes. Another worry spreading via MMS created was the reach ability of the MM. Using MMS, the worm could propagate to any device in the world, unlike other communication methods such as Bluetooth, which is limited to a region or local area for effective detection of other devices. A side effect of propagation via MMS was the cost to the device's owner. The worm spread silently as a background process and the owner in many cases never found out about the spreading until their cell phone bill showed up with several hundred (or thousands of) dollars in mysterious MMS messages sent out. The messages had one of several subject and text lines, as shown next:

```
Norton Antivirus Released now for mobile, install it!
3DGame 3DGame from me. It is FREE !
3DNow! 3DNow!(tm) mobile emulator for *GAMES*.
Audio driver Live3D driver with polyphonic virtual speakers!
CheckDisk *FREE* CheckDisk for SymbianOS released!MobiComm
Desktop manager Official Symbian desctop manager.
Display driver Real True Color mobile display driver!
Dr.Web New Dr.Web antivirus for Symbian OS. Try it!
Free SEX! Free *SEX* software for you!
Happy Birthday! Happy Birthday! It is present for you!
Internet Accelerator Internet accelerator, SSL security update #7.
Internet Cracker It is *EASY* to *CRACK* provider accounts!
MS-DOS MS-DOS emulator for SymbvianOS. Nokia series 60 only. Try it!
MatrixRemover Matrix has you. Remove matrix!
Nokia ringtoner Nokia RingtoneManager for all models.
PocketPCemu PocketPC *REAL* emulator for Symbvian OS! Nokia only.
Porno images Porno images collection with nice viewer!
PowerSave Inspector Save you battery and *MONEY*!
Security update #12 Significant security update. See www.symbian.coml
```

```
Symbian security update See security news at www.symbian.com
SymbianOS update OS service pack #1 from Symbian inc.
Virtual SEX Virtual SEX mobile engine from Russian hackers!
WWW Cracker Helps to *CRACK* WWW sites like hotmail.com
```

Notes from the Underground…

No Dummies!

The body of the CommWarrior MMS message contained the following text:

CommWarrior v1.0b (c) 2005 by e10d0r

CommWarrior is freeware product. You may freely distribute it in its original unmodified form.

OTMOP03KAM HET!

The last line reportedly translates to English as: "No to Stupid People!"

Once the MMS arrived, the worm was included as an infected SIS file. The user had to execute the SIS file, which would then install the worm. During the process, the user was asked several times to give permission to install CommWarrior and had many accompanying text messages, as shown in Figures 5.1.

Figure 5.1 CommWarrior Asking Permission to Install

Image Copyright F-Secure Corporation

In 2007, a proof of concept virus was presented by Collin Mulliner, exploiting an MMS vulnerability to infect mobile devices named Exploit/MMS.A. The exploit and MM was presented at the 2006 Chaos Communication Congress in Berlin, Germany. This proof of

concept MM was a zoo sample and never released in the wild. The vulnerability was discovered in the Synchronized Multimedia Integration Language (SMIL) used to format the embedded multimedia objects in an MMS message. SMIL is an XML markup language used to describe and present various multimedia objects. A malformed MMS message caused a buffer overflow, allowing for execution of arbitrary code. This allowed an attacker to explore and control the device. This MM was device-specific, working only on Windows Mobile operating systems using the ArcSoft MMS composer with release dates prior to August 2006. The only noticeable payload was the MMS reader crashing. Figure 5.2 is a portion of the exploit announcement from 2006 detailing the SMIL vulnerabilities.

Figure 5.2 The SMIL Exploit Portion of Exploit/MMS.A Vulnerability Report

```
Parser for SMIL (Message display function)

Transported in: M-Retrieve.conf body content

Buffer overflows in handlers for the following parameters:

 1)  ID parameter of REGION tag
 ID=”CONTENT” CONTENT is copied into stack-based variable, CONTENT
 can be arbitrary long.

 2)  REGION parameter of TEXT tag
 REGION=”CONTENT” CONTENT is copied into stack-based variable,
 CONTENT can be arbitrary long.

Both overflows allow one to overwrite the return address on the stack. Both are
exploitable and we were able to create a proof-of-concept exploit. The exploit is
triggered by viewing the malicious MMS message (this is different from other
exploits that require substantial user interaction – e.g., to install a program).

Overflow happens after 300 bytes in version 1.5.5.6 and after 400 bytes in version
2.0.0.13.

Categorization: CRITICAL (REMOTE CODE EXECUTION)

Exploit: Proof-of-Concept available (code execution)
```

NOTE

Software vendors had been advised of this exploit by Mulliner six months earlier but no one paid much attention to it! The decision was made to go public to get everyone’s attention.

Two specific areas of the SMIL were found to be vulnerable. The first was the ID parameter of a region tag. This tag held an ID in double quotes that could be given an excessively long content, causing the return address to be overwritten when the parameter

was placed on the stack. The second was the region parameter of a text tag that carried between double quotes text of arbitrary length. This could be excessively written to overflow the stack and cause the return address to be rewritten. The exploits opened the device to Denial-of-Service attacks and remote code injection and execution. A user only had to view the MMS message for the exploit to occur. Once the device was infected, a windowed message appeared with the following statement: "MMS g0t YOu OWnD!!."

Bluetooth

A wireless protocol facilitating data transfer between mobile and fixed devices across short ranges, Bluetooth is one of the most highly used forms of wireless communications around the globe. Devices using Bluetooth range from digital cameras to GPS systems to mobile devices to laptops and gaming devices. This technology has a long record of documented security concerns and has been extensively exploited by MM authors to both infect devices and distribute their payload among potential victims. The most appealing aspect of Bluetooth to MM authors is the ability to use it silently on the device without calling attention to itself. The downside is that Bluetooth only works in short distances of about ten meters. Therefore, it is best employed in heavily populated commercial urban areas with a high Bluetooth device presence. This is needed to maximize discovery of potential victims.

In 2004, the first Bluetooth MM appeared on the scene. A worm named SymbOS.Cabir.a was found spreading across mobile devices running the Symbian operating system with the series 60 platform. The worm arrived to a device in the inbox with the filename caribe.sis. The user was prompted to install the file, and once accomplished, the MM immediately started scanning for other Bluetooth devices within range. Once a device was identified, the MM would commence sending several infected SIS files to the device, attempting to infect it. The infected SIS archive file contained three files:

- The main worm executable file caribe.app
- System recognizer flo.mdl
- The resource file caribe.rsc

The SIS file also contained *autostart* commands that would install the worm on the device once the user agreed.

NOTE

Cabir would only infect mobile phones equipped with Bluetooth and were set to discoverable mode. Setting a mobile device to non-discoverable mode (also called hidden) would prevent Cabir from infecting that device.

A known bug in this MM caused it to lock to a Bluetooth device and only send infected SIS files to that one device. This meant that every time the infected device was rebooted or activated, Cabir would scan for other Bluetooth devices, and upon discovering one would lock to that device, sending it infected SIS files and not search for any other Bluetooth devices. This limited the spread of Cabir to a one-to-one propagation, resulting in slow infection and preventing a widespread epidemic. During the infection process of Cabir on a mobile device, the following messages appeared:

Receive message via Bluetooth from Unnamed device?

Install caribe?

Caribe-V2/29a!

In 2005, a new Bluetooth worm very similar to Cabir was discovered. Named SymbOS. Lasco.A, this MM used the same source code as a variant MM, Cabir.H. It spread via Bluetooth in a fashion similar to Cabir but with one improvement: When a device fell out of range, Lasco would search for other Bluetooth devices to infect. This, in contrast to Cabir, created a scenario where Lasco could spread rapidly in the wild. The infected file sent via Bluetooth was named velasco.sis. The user was asked permission to install it, as shown in the screen capture in Figure 5.3.

Figure 5.3 The Lasco Worm Asking Permission to Install

A secondary form of infection, not related to Bluetooth was file infection done by Lasco. It would search an infected device for SIS archive files and attempt to infect the file in the hopes that file would be copied to some other device. In this case, Lasco would automatically

attempt to infect the new device and commence propagation. Lasco had no payload but its potential to spread quickly made it a very worrisome worm.

Notes from the Underground…

One Author, Two MM

Both Lasco.A and Cabir.H were written by the same MM author. It appears Lasco was created to fix the bug in Cabir, allowing Lasco to detect multiple Bluetooth devices, which Cabir could not do. This let Lasco quickly propagate across Bluetooth devices.

In 2006, Mac users got a taste of a Bluetooth worm with the release of the zoo MM, Inqtana.A, a Java-based worm that targeted OSX 4.0 Tiger systems lacking a patch for vulnerability CAN-2005-1333. This proof-of-concept worm replicated via Bluetooth to devices by attempting to copy three files to that device using an OBEX push request that required the user to accept the data transfer. The worm was set to not function after February 2006 and was never seen in the wild, yet the novelty of using Bluetooth to replicate to any enabled mobile device showed the capability of mass chaos that Bluetooth MM can cause in the future.

In December 2007, a new Symbian worm appeared that was strikingly similar to CommWarrior. Titled SymbianOS.Beselo.A, this worm spread across MMS and Bluetooth by replicating the worm body and sending itself to other Bluetooth-enabled devices. It functions in primarily the same way as CommWarrior, with one novel difference: The file extensions were changes from SIS to popular ones such as JPG, MP3, and RM. This social engineering tricked people into feeling comfortable and allowing the installation of the SIS file while thinking they were going to enjoy a picture, video, or audio clip. Figure 5.4 is a screenshot of one filename used by Beselo.A:

Figure 5.4 SIS Beselo Infected Using a Fake Filename to Trick Users into Installing the Worm

E-mail

In classic malware, e-mail has long been used as a vector of infection for several worms. Typically, they all work the same way: search for addresses and an SMTP, and create e-mails with the malware attached to the message. Once sent out, the recipient is tricked through social engineering into running the attachment, and thus infection is achieved. In the world of mobile devices, e-mail is the second biggest task performed, with text messaging in first place. Currently, not too many MM have been seen using e-mail for infection, but one notable sample has arisen, setting the stage for future MM.

In 2006, an e-mail worm named MSIL.Letum.A@mm arrived on the scene. This mass mailing worm was written on the Microsoft .NET platform and was built in the MSIL specification. Letum spread by e-mailing itself through any SMTP found on the victim's machine as an attachment to addresses found on a fixed computer. It infected all the known versions of Microsoft Windows, but what was later discovered was that Letum was actually built in the .NET CF platform, which is specifically created to run on Windows Mobile. The result was an e-mail mass mailing worm that infected any Windows platform having .NET or .NET CF installed. The worm also spread via newsgroups through NNTP. A typical e-mail, with the worm in the attached file test.exe, is identified in Figure 5.5.

Figure 5.5 A Letum E-mail with test.exe as a Copy of the Worm

```
From: Symantec Security Response [pete{BLOCKED}rrie@symantec.com]

Subject: (any of the following)

■ Warning
■ Virus Alert!
■ Customer Support
■ Re:
■ Re:Warning
■ Security Response
■ Virus Alert
■ Letum
■ Virus Report
■ Warning!

Message Body:

Dear User,

Due to the high increase of the Letum worm, we have upgraded it to Category B.
Please use our attached removal tool to scan and disinfect your computer from
the malware.

If you have any comments or questions about this, then please contact us.

Regards
```

```
OR

'Hiya,

I've found this tool a couple of weeks ago, and after using it i was surprised on
how good it was on squashing viruses. I wonder if avers know about this? ;)'

OR

'Maybe not but try this, i'm sure it will help you in your fight against malware.
The engine it uses isnt to bad, but the searching speed is very fast for such a
small size '

Pete{BLOCKED}rrie

Senior Anti-Virus Researcher / Senior Principal Software Engineer

©1995 - 2006 Symantec Corporation All rights reserved.

Attachment: test.exe
```

Wired Communication

It almost seems that today's mobile devices have no need to connect to anything via a wire. In the near future, that may be true, but for now there are still a few necessities that are best accomplished with the use of a wired connection. Mostly mobile devices get wired to perform system backups, updates, and synchronizations of data. Most mobile devices have ports for removable media to ease the transfer of photos, video, audio, and other important files. This is usually done with memory cards, which can be used with almost all mobile devices on the planet, barring a few exceptions—like the iPhone, for example.

A respectable amount of MM samples have used both synchronization and memory cards to spread. Each has used the development tools available to create MM to infect across these vectors with little or no problem. These vectors have proven to be very reliable, causing little to no side effects that prevented MM from spreading. Therefore, they can be viewed as very reliable for use by future MM.

Removable Storage

Memory cards, flash memory, memory sticks, SD cards, and so on… All these represent little plastic wafers of technology capable of holding enormous amounts of data that can be carried in your pocket, wallet, or false shoe bottom without hassle. Practically every device from cameras to printers to laptops to mobile devices come equipped with insertion ports, allowing the full use of these cards to store and transfer data. MM authors have been quick to figure out how to use memory cards to expand the horizons of their infections. Using these cards, an MM can potentially infect not only other mobile devices, but any device equipped to read the card. This opens many new possibilities by creating MM that will run

on more than one platform. These multiplatform MM are in the growing stages now but stand to become more sophisticated in future MM.

In 2005, an MM named SymbOS.Cardtrap.A (Cardtrap) was discovered in the wild. This MM affected devices running the Symbian OS with the series 60 platform. When the MM was installed on a mobile device, the payload would copy the following three MM files to any currently present memory card:

- Fsb.exe – W32 backdoor BKDR_BERBEW.A
- Caribe.sis – MM SYMBOS_CABIR.A
- System.exe – W32 memory resident WORM_WUKILL.B

Each of these files was previously discovered malware and the intent to attempt infection again was clear. The Cabir MM was also installed on the device, not just copied to the memory card. Along with these three MM files, Cardtrap also created an autorun file on the memory card. The autorun attempted to install BKDR_BERBEW.A on a system once the memory card was inserted into a card reader. This was a novel concept that had not been seen in any other MM to this point. Using the memory card to infect other systems, principally a PC, was the first of its kind. By attempting to install a backdoor on W32 systems, Cardtrap was giving its shadow masters access to both mobile devices and fixed computers that could later be used to accomplish anything from data stealing to Denial-of-Service attacks. Cardtrap also rewrote application files on the device, rendering them useless.

In 2006, the MM W32.Mobler.A worm was discovered by F-Secure. This MM was written to run on the Windows platform but also had in its payload malware to infect Symbian OS mobile devices. The cross infection occurred by propagation through the memory card. On the Windows side, Mobler would hide several folders and copy itself to all available folders, USB drives, and memory cards. Mobler was very destructive on the Windows side, but on the Symbian side it only attempted to infect memory cards with its payload of Windows malware in hopes the user would insert the memory card to a PC card reader, allowing the MM to infect further. The files it carried in the payload were:

- **autorun.inf** An autostarter file for system.exe
- **black.app** A text file
- **black.html** An HTML file with a short message from the author
- **black.ico** An icon file
- **black.jpg** An image file
- **black.txt** A text file
- **makesis.exe** A clean utility that creates SIS archives

- **Black_Symbian.sis** An archive of the worm and other files to run on Symbian
- **Black_Symbian.pkg** A list of files in the SIS archive
- **system.exe** A copy of the worm

Device-to-PC (D2P) Synchronization

Every mobile device has the ability to connect with a fixed computer for the purpose of synchronizing data on both machines. This is commonly done with contacts, e-mails, notes and specified folder contents. Synchronization is also used to back up the complete mobile device system and apply operating system updates and patches. The connection created between a fixed and mobile computer is a perfect, stable, and easy way for an MM to infect a mobile device from a fixed computer. Only one novel MM achieved this goal, but as computer connectivity becomes more ubiquitous, this form of multiplatform malware will soon be on the rise.

In 2006, a proof-of-concept worm named MSIL.Cxover.A was announced by a group of mobile device researchers named MARA. The worm was written in C# for any Windows operating systems running the .NET and .NETCF platforms, including Windows Mobile. The MM infected mobile devices using the ActiveSync connection to propagate from the PC to the mobile device. Once installed on the mobile device, CxOver would erase all files in the My Documents directory and install itself to run on each reboot of the machine. On the PC side, the MM would silently run in the background, waiting for an ActiveSync connection to be established, at which point propagation would commence. It was the first MM to infect the mobile device from a PC automatically without the need of user interaction to approve the installation. The MM was a zoo sample and never released in the wild. The MM did raise concerns since it showed the viability of cross-platform malware further complicating what could be expected in future MM.

Notes from the Underground…

A Malware with Four Names

Cxover was originally named Crossover by the anonymous author. Through naming conventions used by antivirus companies, it was also named CxOver, Xover, and OverCross, resulting in four names for one MM.

Other Infection Strategies

In this part of the taxonomy, we examine infection strategies that have not been used to a great extent by MM but have great potential for future abuse. These infection vectors are currently in the R&D states for MM authors, and it is only a matter of time before bad actors and shadow masters employ these vectors in MM. It is important to understand these vectors now and adequately build defenses for them before they emerge from the hands of a shadow master.

SMS

An acronym for Short Message Service, SMS is the key communications protocol used in sending and receiving text messages on handheld devices. Text messaging has surpassed e-mail as the number one form of communication between individuals around the world, with an average of 3 billion active global users. SMS allows messages to be sent as plain text across communication networks. What most people don't see in a SMS message is the portion that instructs the device to take certain actions. Each SMS message is accompanied by a list of commands that are read and executed by the device to process the text message properly. It is in this area where the SMS becomes a vector of infection for mobile devices. Currently, no major MM has appeared that exploits SMS to infect mobile devices. However, vulnerabilities have been discovered and SMS could be an infection vector for future MM.

In 2000, WebtoWap AS identified an SMS vulnerability in SMS-enabled Nokia phones. This vulnerability was exploitable by sending a specifically formatted SMS text message. The message could cause the phone to freeze, disable function buttons, and create other minor forms of havoc. The phone battery had to be removed and returned to set the phone back to normal working status. Fortunately, MM using this never emerged since it required special hardware knowledge, plus access to sophisticated tools not available to the general public, and the author had to be a skilled software developer. Nonetheless, this exploit shows potential for future privately discovered exploits to appear in MM.

In 2002, another SMS vulnerability was discovered by Job de Haas, a researcher for the Dutch security firm ITSX. Similar to the 2000 vulnerability, this one allowed a malformed text message to cause the mobile device to crash and even render some devices useless. The exploit worked in Nokia phones. At the time of its discovery, the vulnerability was played down and did not garner too much attention. Nokia later remedied the vulnerability to avoid the exploit from occurring in the future.

Wi-Fi

The potential of a widespread Wi-Fi MM epidemic has been greatly theorized and feared for some time now. Yet this form of infection by an MM has yet to be realized, though many believe it is on the horizon and poses a major threat to both mobile devices and fixed computers. In late 2007, a research team from Indiana University conducted simulations of a hypothetical Wi-Fi worm outbreak in a densely populated area. The testing simulated attacks in seven American cities, which resulted in several thousand wireless routers being infected within 24 hours of the initial launch. The worm jumped from router to router turning each one into a little spy that could monitor information flowing from devices connected to it. Though the researchers did not address the impact on mobile devices, it is clear to see how the data stored on them could easily be stolen and abused. More interestingly is the use of a mobile device as the initial launch point of the Wi-Fi attack, leaving no evidence with which to uncover the bad actor responsible for the epidemic. The conclusion of the simulation was that a Wi-Fi epidemic could spread wirelessly, jumping from router to router similar to how an airborne human virus spreads. The payload of such an attack on a dense urban city is only limited by the reader's imagination.

OS Vulnerabilities

Many classic malware infect a computer by exploiting a vulnerability in the operating system of that computer. MM is no exception to this rule, with several known samples succeeding in infecting a mobile device by exploiting a vulnerability in the OS. What is of interest is that in almost every case the vulnerable operating system was the Symbian OS, with buffer overflows and return address modification leading the pack. This is not to say that other mobile device operating systems do not have their flaws, but up to now the majority of mobile devices in use run Symbian OS, so it was a clear target for MM authors. As the landscape changes and more devices come into use using Java JRM, Windows Mobile, and iPhone/iPod it is almost certain that MM authors will focus on exploiting these platforms as well. Known MM samples using OS exploits to infect are too numerous to describe, instead a list of names is provided in Figure 5.6, and encouragement is given to the reader to find the details of each.

Figure 5.6 A List of MM Infecting via an OS Vulnerability

Worm.SymbOS.Mobler.a	Trojan.SymbOS.Singlejump	Trojan.SymbOS.Hobble
Trojan.SymbOS.Locknut	Trojan.SymbOS.Dampig	Trojan-Dropper.SymbOS.Agent
Trojan.SymbOS.Bootton	Trojan.SymbOS.Romride	Trojan.SymbOS.Skuller
Trojan.SymbOS.Appdisabler	Trojan.SymbOS.Drever	Trojan.SymbOS.Skudoo
Trojan.SymbOS.Cardblock	Trojan.SymbOS.Cardtrap	Trojan.SymbOS.Fontal
Trojan.SymbOS.Blankfont	Trojan.SymbOS.Doombot	Trojan.SymbOS.Rommwar

Distribution

Malware has always attempted to attack as many vulnerable systems as possible. In the history of malware, some of the most malicious were able to spread to thousands if not millions of computers worldwide, causing enormous damage, and costing millions (in some cases, billions) of dollars. In the era of MM, the capacity to distribute amongst mobile devices grows exponentially and the threat of potential damage grows in parallel. In today's world, for every person with a desktop or laptop there are a hundred others with a cell phone, a PDA, or a portable music player. All of these are equipped with the infrastructure necessary to be a target of an MM when it commences distribution to attack other potential victims. The result of today's use of mobile devices in every hand is a much bigger pool of potential victims, who could become part of a catastrophic MM attack causing damages in the billions (maybe hundreds of billions) of dollars worldwide.

How big can an MM attack be based on distribution? Consider downtown in any urban city around the world. It's 8 A.M.… People are going to work and are roaming about with their mobile devices in hand. A bad actor arises from the masses, retrieves a mobile device and presses **Enter**. An MM using privately discovered zero-day vulnerability is released and starts scanning for potential victims via Wi-Fi. In a matter of seconds 98 percent of the mobile devices in a three-mile radius become totally inoperable. Twenty minutes later, news reports come in from urban centers all over the world of an unexplained phenomenon of mobile device failures. Within two hours, 90 percent of all active mobile devices around the world have been rendered useless. All this is the result of one bad actor—or in this case, a shadow master—in one downtown urban center, releasing one MM with a zero-day exploit. Three hours after its initial release, panic is raging worldwide as persons unable to use their mobile devices don't know what to do or how to function, chaos ensues with unforeseen consequences…. And the bad actor? Back at home watching a pirated DVD while eating pizza and realizing the just accomplished destruction of the mobile device used to launch the attack ensures no positive identification and the possibility of a repeat attack at a future date.

When considering taxonomy based on distribution, one must focus on what is available for use by an MM. To make this conclusion, an analysis of the current mobile devices is needed. One can quickly conclude that every form of known communications available to computers is also found in any given mobile device. But within this cornucopia is a subset that is most often used by known MM. Of this subset, three which have proven to prevail, will be the focus of this taxonomy based on distribution. The three taxa are as follows:

Bluetooth, SMS, and memory cards. The new taxa will again be subtypes of the main taxon: wired and wireless. Since some of the technologies presented in this section have already been explained, we will only present here their relation to distribution, along with a MM sample's usage of the technology.

Wireless Communication

Clearly, from the known MM samples, distribution via wireless is king. With just a few exceptions, the vast majority of known MM used one or more wireless communication technologies to spread their nasty payloads in search of other victims. The taxa presented here are, up to now, the most commonly used. As we move forward, we suspect Wi-Fi to become a bigger player in MM distribution. Along with Bluetooth, these represent the fastest vectors so far for a bad actor to quietly spread MM without causing fear or calling attention to itself. Yet there are other technologies on the horizon, like 3G, tha t will prove to be kings of the next round of most commonly used MM distribution vectors.

SMS

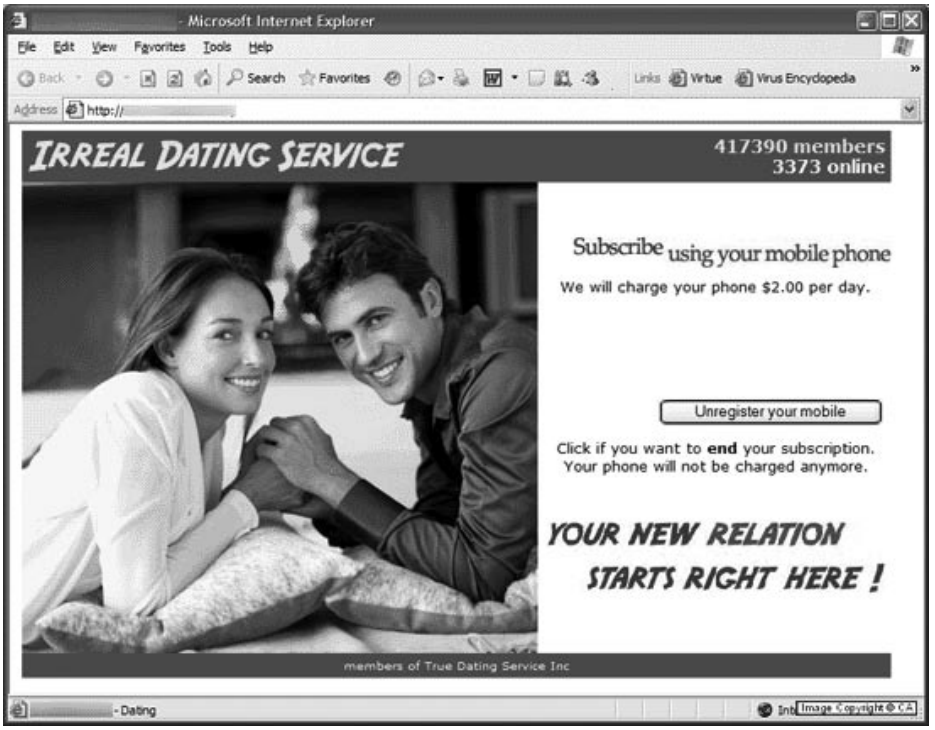
Unlike MMS which has been used more for infection, SMS has been a tool of MM distribution for some time now. With billions of text messages going out every day around the world, SMS has proven a speedy distribution tool for bad actors. Add to that the ability to send SMS to a mobile device from almost anywhere—and with strong anonymity—and it becomes a logical starting point of release and distribution for new MM being let loose by a shadow master into the wild. As long as SMS can be used in an anonymous nontraceable fashion, it will continue to distribute MM for shadows while they are granted "diplomatic immunity."

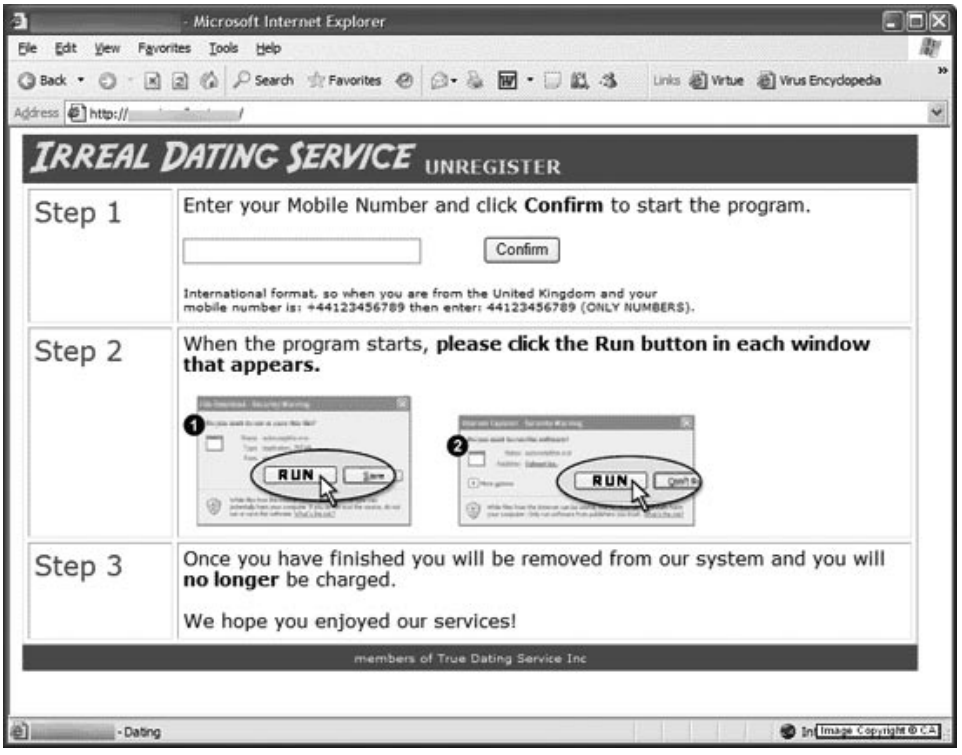
In 2006, a W32 Trojan named Bambo.CF was luring people to a dating Web site in the hopes of downloading the MM to their mobile devices. The MM was distributed by sending SMS messages to mobile devices with text similar to the following:

```
Thanks for subscribing to *****.com dating service. If you don't unsubscribe you
will be charged $2 per day.
```

The message was a good piece of social engineering, luring the reader to the malicious Web site in hopes of avoiding unwanted charges. The link led to a fake dating Web site where the user was enticed to enter their phone number and then click a button labeled Unregister Your Mobile. Once the button was clicked, the Trojan was installed on the mobile device. Figure 5.7 shows screen captures of the false dating service Web site.

Figure 5.7 Malicious Web Site Used to Install Bambo.CF on Mobile Devices





Another MM released in 2006 also used SMS to lure victims to download the malware to their devices. The name of this MM was VBS.Eliles.A, written in Visual Basic script, it is classified as an e-mail mass mailing worm. As a secondary form of distribution it would send out SMS messages to mobile devices containing a link to download the MM. The phone numbers used to send SMS were calculated with a built-in routine that generated random phone numbers for two mobile phone service providers in Spain. The user received an SMS claiming to be from the service provider offering to download antivirus software. The link would instead download a SIS file containing the MM. It is interesting to note at the time of release that no mobile device was equipped to run Visual Basic scripts. That made it clear this MM was targeting Symbian phones but had a separate MM wrapped in a SIS file for infection. The body of the SMS message was similar to the one in Figure 5.8.

Figure 5.8 SMS Text of the Eliles Worm

```
Subject: Msj Operador: Proteja su movil
Body:
Descarguese gratis el Antivirus para Nokias Series 60.
(6630,6680,7610,7650,N70,N90), totalmente gratuito.
        http://f1.grp.yahoofs.com/v1/ oHDmRCSTUJ2I3kbX4Kr8GMzmLAO7taS5yJIVcWx2F_
        6NWlo_LBonXVhAfgMBbxzzC4LoS8XSwl_-YO7ZMH01Sw/Antivirus.sis
```

In 2007, researchers from the University of California at Santa Barbara released a zoo sample of a proof-of-concept worm named SymbOS.Feak (also known as SymbOS.Keaf). The worm distributed by sending out SMS messages from the infected mobile device. The text of the message contained a link to an Internet site that would download the worm and infect the device. This MM consisted of the following two files:

- **feakk.exe** The worm executable
- **feakk.mdl** An installer file for the worm executable

When the device was started or rebooted, feakk.mdl would execute feakk.exe. Once installed, the MM would search in the list of contacts for a trigger entry named HACKME. This was done to control distribution of the zoo sample to only test devices. If the entry was found, the MM would commence sending out messages to all the contacts found on the device. Once a target device received the message, the link would be followed to download the UCSB hosted worm. The body of the message was as follows:

```
hey check this link out http://www.cs.ucsb.edu/%7efeakk/feakk.zip bye!
```

Notes from the Underground…

A Pile of Feak?

The word Feak is defined as slang for fecal matter, butt residue, small granules of poop, or the invisible smell left on the hands after taking a poop. You can't see it but you can definitely smell it. Now, is that an appropriate name for a POC MM?

Bluetooth

For distribution purposes, Bluetooth serves as a direct way of spreading MM to other Bluetooth-enabled devices. This approach allows the MM to be sent aggressively to other devices in a direct and aggressive manner. Only an acceptance from the device user is needed for the MM to enter the device and cause havoc. This is a very appealing approach, simply because every mobile device is Bluetooth-enabled, and in some cases an MM can install without user interaction after being distributed through Bluetooth. It is a standard distribution approach for MM that is not going away anytime soon.

In 2007, an SMS Trojan named SymbOS.Viver.A began doing the rounds, being distributed through the Internet and Bluetooth. The Trojan itself was a SIS file designed to run on Symbian-enabled mobile devices. The Trojan carried two SIS files:

- RulesViver.sis (42,962 bytes)
- NetCompressor.sis (10,624 bytes)

When the Trojan arrived via Bluetooth to a mobile device, the user had to give permission for the installation to occur. The Trojan masqueraded as a standard application to trick the user into approving installation. Once installed, the malicious payload would cause the phone to dial premium rate numbers. The result was the owner being charged for the calls, with a portion of the moneys ending up in the shadow master's pocket since he/she had rented the premium phone numbers being dialed.

Another interesting Trojan horse released in 2007 targeting Symbian-enabled phones was SymbOS.Stealwar.A. This Trojan did not use Bluetooth to distribute itself. Instead, it used Bluetooth to distribute other known MM to enabled mobile devices within range. The Trojan came as a SIS file that, once installed, placed the following MM on the device:

- SymbOS/Cabir.A
- SymbOS/Lasco.A
- SymbOS/CommWarrior.A
- SymbOS/Pbstealer.A

Once these MM were installed on the mobile device, they would each start distributing and infecting other mobile devices via Bluetooth. This created heavy Bluetooth traffic on the device, which had the side effect of depleting the battery very quickly.

Wired Communication

Given the advantage of wireless communications in mobile devices, it is not surprising that few MM used wired technologies to distribute themselves. For infection, several novel MM have appeared, using wired communications, as explained earlier in this chapter, but for distribution it is a dying art form. The only noticeable wired technology used for distribution has been memory cards. Along with infection they are very convenient in distributing MM from one device to another, and one platform to another. Moving forward as long as memory cards remain open for free reading and writing and have the ability to execute an autostart file, they will be employed by bad actors to distribute MM. As for other forms of wired communication, they will be left behind, only used for direct MM infection and not much else. As a vector of distribution, they may eventually be pushed to the side in favor of faster wireless technologies that provide speed, widespread reach, and most importantly to the shadow master, anonymity.

Removable Storage

Of all the known MM that employ removable storage in some fashion, the majority use it as a vector of infection. But there is one known MM variant that used memory cards more for distribution then infection, though admittedly the argument can go both ways. The name of the MM is SymbOS.Beselo.B. This worm infected mobile devices running the Symbian operating system. It primarily distributed via MMS and Bluetooth. As a third form

of distribution, the MM used memory cards to spread to other Symbian mobile devices. Beselo listens for the insertion of a memory card into the infected phone. If a card is inserted, it copies itself to the card and bootstraps it. The bootstrap will run and install a file that places the worm into another mobile device. Beselo copies the following two files to the memory card:

- **qsnpwsg.exe** The worm executable
- **gsnp.mdl** An install file for the worm executable

Payload

The payload is normally the damage inflicting component of malware. It is only limited by the imagination and devious nature of the malware author. Typically, payload consists of two types: nuisance and devious. Nuisance payloads are normally not catastrophic, not a breach of security, or an invasion of privacy. They tend to be recoverable and are done just to upset the victim of the target. Examples of nuisance are: file deletions, e-mail deletions, disabling Internet connections, defacing your background picture and icons, and uninstalling software. Devious payloads, on the other hand, are used with more sinister goals in mind. These payloads are meant to exploit the information stored in a target for financial gain, further distribution, identity theft, or use in other malicious deeds or crimes. Some examples of devious payloads are unauthorized access, stealing of sensitive data, invasion of privacy, and identity theft.

With the advent of MM, new forms of payloads have emerged that are potentially more dangerous than any seen previously. The most dangerous of all is the bad actor accessing a victimized mobile device to launch an MM attack and thus hide the identity of the real attacker. Other devious MM payloads include: unauthorized viewing through a built-in webcam; listening via the device's speakers; and taking pictures that are then sent to the bad actor. Some new nuisance payloads not heavily used or seen are: running a process to purposely deplete the device's battery, and dialing random phone numbers for an infinite period of time.

The taxon used for payload will include subtypes that have not yet occurred. These subtypes will be discussed in a hypothetical sense to give some direction of what to expect in future MM releases. For each specific payload discussed, a label of nuisance or devious will be given.

Communications Component

This component represents all the connectivity aspects of a mobile device minus the phone. This includes e-mails, Bluetooth, SMS, MMS, and others… These components have been used heavily by MM for many different reasons, as we have already seen. They are not used as much for payload purposes, but the use they do have is very precise and can be very costly.

Sending SMS Messages: Nuisance

In 2000, an early form of MM appeared called Timfonica. Its claim to fame was its ability to send SMS messages to randomly created numbers belonging to a service provider in Spain. At the time, SMS was not known and the MM was not paid attention to much. In reality, it was a forerunner of things to come.

In 2004, a Trojan name SymbOS.Mosquit was discovered. This Trojan had a payload that sent SMS messages to premium-rated services without the owner's knowledge. The list of numbers used for the SMS were hard-coded into the MM. It entered the devices by people downloading it from P2P networks where it masqueraded as a pirated version of a popular game called Mosquitos. The result of these SMS being sent out was a big bill for some owners at the end of the month.

File System

This type of payload has been very common in several classic viruses. Many examples exist, with payloads that delete files, uninstall applications, block access to hard drives, destroy boot sectors, and so on…. With the advent of MM, these classic payloads have not been ignored due primarily to the weak security mobile devices carry, which allows open access to the device's entire file system, thus giving the bad actor plenty of malicious options to execute.

Infecting Files: Nuisance

Most viruses infect files to replicate, and this destroys in many cases the targeted files, leaving them unable to be restored to their pre-infection state. This is a major pain in the neck to come back from, especially if you don't have a backup.

In 2004, the Wince.Duts.A virus was released by the virus writing group 29A. It was written by one of its members named Ratter. The code would infect the Windows Mobile platform and once installed would erase several files on the system. It was released as a proof-of-concept zoo sample and the user had to give permission for it to run.

Overwriting Files: Nuisance

Just like infecting files, overwriting them with garbage renders them useless. What is worse is overwriting applications and leaving your device as a great paperweight. Given that most mobile devices are not that easy to restore to their customized pre-infection state, having an MM overwrite files and applications is a major nuisance.

The Trojan SymbOS.Skuller.A, released in 2004 overwrote applications by creating new files with the same names in the same folders as the originals. No malicious code was included in these overwritten files. All the files that were overwritten were applications, and after overwriting they were rendered useless. The Trojan also created Skull icons that replaced the application's original icon and blocked access to that application. A bigger problem occurred when the device was turned off and then on again: It was rendered useless.

Multimedia Components

Any part of a mobile device that interacts with a human user can be considered a multimedia component. These include: webcams, microphones, music players, device buttons, touch screen buttons, voice recorders, styluses, and others. Up to this point, MM has not made too much use of these components in their payload, but some recent MM indicate they are starting to become more popular and can be considered payload targets in future MM. It is clear that the operating systems running on devices today provide the development tools to generate applications that give full access to a phone's multimedia components. This open access is what will eventually allow bad actors to create MM that employ these components in their payload.

Taking Photos: Devious

An MM employing this payload has not yet arisen. The idea though is not far from realization. An MM capable of taking photos by accessing the device's webcam component can be disastrous if, and only if, the right photos are taken. Blackmail comes to mind, along with character assassination. One requirement, of course, is that the photos must be sent to a shadow master quietly, leaving no trace in the device of the photo's existence. Another trivial challenge is to disable that annoying sound most devices make when a photo is taken.

Recording Voices: Devious

Not just recording the input sound of the device's microphone, but recording entire phone conversations could prove very damaging if placed in the wrong hands. A shadow master could do a lot of damage if the right words were recorded. One big problem for the bad actor is to keep from making an audio file of enormous size. This could cause alerts to appear on the device regarding low memory, and could make the transfer of the file back to the shadow master very slow or even impossible. Fortunately, this type of MM has not yet occurred.

Clandestine Video Recorder: Devious

Accessing the full capability of a device's recording components can lead to acquisition of full video with sound. If naughty acts captured on camera without knowledge of the device's owner were accessed it could land them in a lot of trouble. On the lighter side, capturing the right moments in life without the user knowing it can make for a great video to post on the Internet. In the future, it would not be a surprise at all to see an MM capable of clandestine video recording.

Playback: Devious

The three payloads previously described all relate to taking audio, video, and pictures from a device and placing them into the hands of a shadow master who then uses this for malicious purposes. A more frightening idea is to turn this around and have the shadow master send audio, video, and pictures to the user's device. Imagine hearing a voice suddenly talking to you on your device, or a media player that starts showing live shots from your home or office when you're not there. The emotional trauma caused by this could be devastating. This type of payload found in an MM can be some of the worst MM we may ever see, simply because it plays with a person's deepest emotions: fear and despair. Fortunately, this has not yet occurred, but moving forward it could become an uncomfortable reality.

Telephone Component

Clearly, the telephone functionality of a mobile device could also be used for mischief. This is an interesting area to exploit as part of a payload. One would think that a nuisance payload would be to start dialing phone numbers that are very costly. Or use the phone as a relay to talk to others while not being charged for it. These are just some of the payloads that can occur here, but that have not yet been seen. Today's development tools allow any developer to create applications that have full control of the telephone on a mobile device. This will eventually be blended into an MM, and from there the maliciousness will begin.

Dialing Other Phone: Nuisance

An MM is installed on your phone and its payload is to repeatedly dial every number in your phone contacts. Just imagine how many people will become worried, upset, and furious. Once you explain to them what happened it will settle down, but the charges to their phone bill the following month will not make them recall you fondly. This payload has yet to be realized.

Dialing Your Own Phone: Nuisance

Take the previous scenario and flip it around: an MM that enters an infinite loop where the payload is to dial your own number in such a fashion that it rings and you get the busy signal at the same time. This is actually not difficult to build since every device with a phone has recorded within it the phone's telephone number. Normally, this is placed in the ROM when the phone is activated. This also has not yet been realized as a payload.

Using the Phone to Cover Your Tracks: Devious

A very devious use of the phone is to convert it as a relay to dial another number and have a conversation without the knowledge of the device's owner. The phone becomes a gateway connecting two other phones and provides them with unlimited connectivity to talk as long as they want. The advantage of this is that for the one placing the call there is no possibility of tracing the number, instead the number of the victim's phone appears as the source of the call. This application is very similar in functionality to a backdoor; the bad actor can come in at will and use the phone with no blockages. This also is a payload that may appear in future MM.

Data Farming

Data farming is the reading of data for the collection of specific information useful in some form. Bad actors that perform data farming on a mobile device have two principal motivations: financial gain and MM distribution. In the first scenario, the data can be used for identity theft or purchases made with someone else's credit card! In the second scenario, the bad actor uses the information to strike at new potential victims, with the MM spreading the malware further.

Stealing Contacts: Devious

In 2005, a Trojan named SymbOS.PBStealer spread on mobile devices running the Symbian operating system. This Trojan arrived in the SIS file PBEXPLORER.sis and masqueraded as an application that would compact your phone contact's database. In reality, the Trojan read the contacts database, wrote all the data to a text file named PHONEBOOK.TXT and then sent the text file to the first Bluetooth-enabled device it detected. The MM would continue passing requests to the device to accept the text file for one minute. If the target device never accepted, the Trojan ceased. Though stealing contacts is an invasion of privacy and could cause tremendous damage, this MM failed in sending the information to the bad actor (the MM author is clearly not a shadow master). Instead, it could potentially be sent to a random stranger who would ignore the requests and thus no damage is done. This MM highlights how easily data can be stolen from a mobile device and should be seen as a significant threat in future MM.

In 2006, a spyware application was released with the marketing campaign of "Catch Your Cheating Spouse." The application was a Trojan named SymbOS.FlexiSpy.A, which

ran on Symbian-enabled mobile devices. When the application installed on the device, it did not give a formal title or name. Once installation was complete, the MM would hide and lock all its files, thus avoiding being uninstalled. The application interface was only accessible through a password entered by the bad actor. The MM allowed for tracing of information of SMS messages and voice calls to and from the victim device. An option was also placed to choose when the tracing should occur. FlexiSPY recorded the following from voice mails:

- IMEI
- Client time
- Server time
- Direction
- Duration
- Phone number
- Contact name in the victim's phonebook

As for SMS, the following information was recorded:

- IMEI
- Client time
- Server time
- Direction
- Duration
- Phone number
- Contact name in the victim's phonebook
- Contents of SMS messages

The information was stored on a Web site accessible through a password. Figure 5.9 shows a screenshot of the Web site.

Figure 5.9 The FlexiSPY Web Site

Summary

This chapter has presented three taxonomies for mobile malicious code. The taxonomies were based on infection strategies, distribution, and payload. The taxonomies include taxa that highlight what has already been seen in known MM samples. It is clear that MM has borrowed heavily from classic viruses, using them as lessons learned. Also, the known MM samples have shown novel approaches that are only possible now with the technologies made available with mobile devices. Bluetooth, SMS, and MMS are all new vectors unique to mobile devices that are being heavily used by MM. The taxonomies have also shown potential approaches that have yet to occur but that carry a high probability of appearing in the future. The overall lesson here is that mobile devices will be a singular target of several future MM, and steps to avoid these potential epidemics and headaches must be taken; otherwise, the result could be nothing less than disastrous.

Solutions Fast Track

Infection Strategy

- ☑ The most common vectors of infection are Bluetooth, MMS, e-mail, synchronization, and memory cards.
  - ☑ CommWarrior spread in 2005 via Bluetooth and also MMS, creating a global MM threat for the first time in computing history.
- ☑ Distribution is accomplished mostly with SMS, Bluetooth, and memory cards, with Bluetooth as the most common MM vector to date.
  - ☑ The most common method for infections in the wild to date is via user interaction, accepting hostile files.
- ☑ The most common payloads are file system modifications and sending out SMS.
  - ☑ The most common indirect payload is the draining of a battery on a mobile device as worms attempt to spread over Bluetooth.

Distribution

- ☑ Millions of mobile devices results in millions of MM opportunities.
- ☑ As people learn to trust and depend on mobile devices and assets mature within the mobile medium, such as mBanking, risk increases.
- ☑ Exploitation of devices through a zero-day vulnerability has tremendous opportunity in the mobile medium.

Payload

- ☑ Phone components, webcams, and microphones are potential targets of future MM payloads.
- ☑ Wi-Fi MM exists in theory and can be realized. Simulations showed catastrophic epidemics using this vector.
- ☑ Blended MM using several vectors for infection, distribution, and payload are the next step in the evolution of malware.
- ☑ Using technologies in mobile devices that provide anonymity will play key roles in future generations of MM.

Frequently Asked Questions

Q: Which taxonomy is the most important of the three presented here?

A: They are all equally important since they each take a different viewpoint on categorizing MM.

Q: If you had to choose a taxonomy to address first, which one would it be?

A: My immediate concern would be protecting the vulnerabilities shown in the payload taxonomy. This taxonomy shows what can be done when an MM epidemic occurs. Thus, it should be remedied first.

Q: How can these taxonomies be modified to accommodate yet-to-be-seen aspects of MM?

A: The taxonomies should be created in a broad enough hierarchy where new taxa can be added to incorporate future MM components and approaches.

Q: Are all known samples described in the taxonomy?

A: No. For each taxa listed, we gave a few samples of known MM to illustrate the various forms in which the taxa has been used up to now. For each taxa there are many other MM samples that incorporate them. These are not presented here, however.

Q: What is a bad actor and shadow master?

A: A bad actor is a successful black hat malware author that works in anonymity. A shadow master is a legendary attacker at the top of his/her game that is usually sought out by others to do "complicated" jobs. They are collectively referred to as shadows or shadow (singular). A shadow actor has created successful malware with known technologies. Shadow masters have the same accomplishments as an attacker, plus proof-of-concept code that spearheads malware into new areas of emerging technologies.