Chapter 4

Overview of Mobile Malware Families

Solutions in this chapter:

- **Cabir**
- **Skuller**
- **Doomboot**
- **Cardtrap**

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

Introduction

Since 2004, the genesis of MM, over 30 distinct families have appeared. The combined total of known original MM viruses and their variants since then have climbed to several hundred. These families and their variants have evolved to achieve the same goals as classic computer viruses. However, while computer viruses evolved over a period of a quarter century, MM met and surpassed the same evolution in just four short years. This lightning speed growth is not surprising, given the wealth of knowledge from 30 years of classic computer viruses. MM authors were well equipped with advanced infection, distribution, payload, and stealth techniques for their nefarious creations. What is surprising is the ease with which they were able to implement these on newly created mobile device platforms. This evolution clearly shows MM authors to be way ahead of the game. In the future of MM, new samples will inevitably include never before seen techniques that will prove to be difficult to analyze and mitigate.

It is important in the new MM era to analyze the families and variants that have come to light. Many of these families are truly original, showcasing what can be accomplished with mobile devices. Other families and variants are merely script kiddies modifying previous MM code to achieve little beyond what the original sample did. These families show that the authors behind them range from seasoned veterans, responsible for some of the totally original viruses, to new faces arising from the masses with the needed expertise to exploit this new MM frontier.

In the evolution of mobile malicious code (MM), four families—Cabir, Skuller, Doomboot, and Cardtrap—have risen to dominate the scene based on a large number of variants. These families are considered pioneers in this category. What follows in this chapter is an analysis of each of these families and their variants with a focus on their infection strategies, distribution, payloads, life cycle, novel contribution, and impact on the MM scene.

Cabir

Cabir is the virus that ignited the MM revolution. The first sample of the family was released in June 2004. The source code was released in 29A ezine and quickly produced 35 new known variants as a result. The original sample, Worm.SymbOS.Cabir, ran on the Symbian platform in Nokia phones. It spread via Bluetooth, which was a totally novel approach at the time for worm distribution.

Notes from the Underground…

Viva España!

The original Cabir.A MM was e-mailed to Kaspersky Labs by a famous virus collector from Spain name VirusBuster.

The worm would spread as a SIS archive file named caribe.sis, which arrived in the inbox of the target device. The user was required to give permission to install the file onto the device. Once the worm was installed, it would immediately start seeking other Bluetooth-enabled devices within range. When a device was located, Cabir would lock to that device and commence sending the SIS files multiple times in the hopes of successful infection. A bug in Cabir.A was that the lock to another Bluetooth device would continue even after the device went out of range. This resulted in continued attempts to send the SIS file to an unreachable device, which greatly lowered the propagation of the worm in the wild. Cabir.A would not search for other Bluetooth-enabled devices once it locked on to the fist discovered device. It was only capable of attempting replication to one other device each time it executed. Another side effect of Cabir.A that slowed down its propagation occurred when a newly infected phone started searching for other Bluetooth-enabled devices and discovered the original device that sent the worm to it. This would become a tennis match sending the worm back and forth between two phones. Cabir.A would propagate much better when the sender of the worm was out of range of the newly infected device. The following are the files included in the SIS file and the locations they were copied to when the worm infected a new device:

- caribe.app to \system\symbiansecuredata\caribesecuritymanager\
- caribe.rsc to \system\symbiansecuredata\caribesecuritymanager\
- flo.mdl to \system\recogs

The source code for this virus was released to the public in the #8 issue of the ezine published by the malware group 29A. The author's name is Vallez. The malware was written in the C/C++ languages specifically for Symbian series 60 platform. It was known to work on Nokia phones. The source code was quickly used by other MM authors, spurring a long list of variants. Even though Carbir.A was novel in being the first true mobile device MM

and the first to replicate via Bluetooth, it was only a proof-of-concept, and was never released in the wild. The biggest impact it had was firing up the engines for the MM revolution.

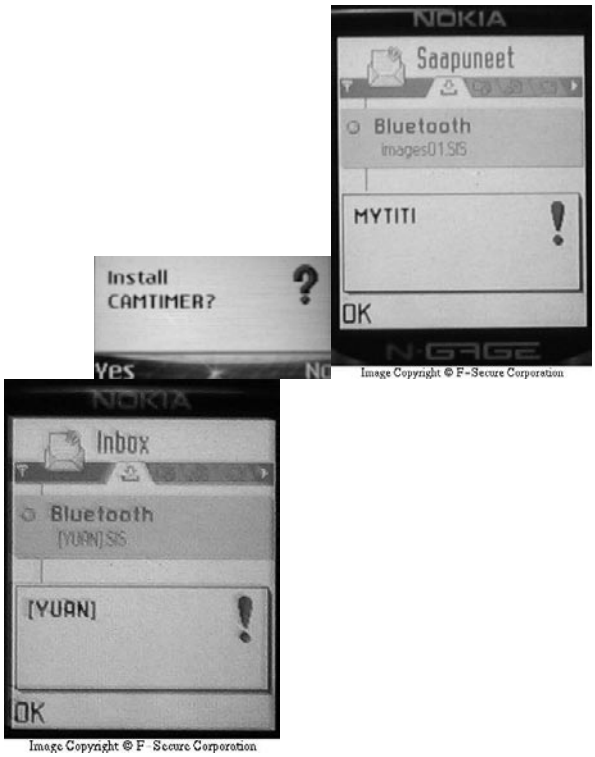
When Cabir.B was released the same year as its predecessor, the new variant had the identical functionality as the original MM. The only difference was Cabir.B would display the word *caribe* every time the device was restarted. It also would try to replicate to any Bluetooth-enabled device, including those not running the Symbian OS, the side effect of this was a rapid draining of the device's battery.

NOTE

In 2005, the computer security company F-secure used Cabir.B and Cabir.H to attempt infecting a Toyota Prius through its Bluetooth capability. Fortunately, the only problems that occurred were the result of a low battery. Successful infection by the MM was never achieved.

Cabir.C through Cabir.G are identical in functionality to Cabir.B, with the only difference being the name of the SIS archive file and the text displayed on the device when the MM is installed. It is suspected that these variants were just script kiddies making minor hexadecimal modifications to the source code of Cabir and releasing them to antivirus companies. But the word in the underground is that these variants were actually tests attempting to fix the bug that Cabir carried, which limited it to only infecting one other Bluetooth device per execution. The next batch of variants was the result of the testing. Figure 4.1 shows some screenshots of the different names displayed after infection was completed for these variants.

Figure 4.1 Screenshots of Cabir.C, .D, and .E



Are You Owned?

Bluetooth Openness

The majority of Bluetooth MM infects mobile devices only when the device is set to discoverable mode. By switching this option to hidden, you just protected yourself from several headaches. Is your Bluetooth-enabled phone in discoverable mode?

The next group of variants, Cabir.H through Cabir.J, had two distinct differences from their predecessors. First, they were recompiled versions of the original source code, which surprised many in the security world who were not aware the source code was floating around the underground, even though the group 29A had released the Cabir.A source code in their #8 issue. The second difference, and the most important, was that the bug limiting the propagation of Cabir had been fixed. This new incarnation of Cabir now had the capability to propagate via Bluetooth to several devices. When Cabir found a Bluetooth-enabled device, it would send a SIS file named velasco.sis repeatedly to the device until it accepted it or went out of range. Once the device went out of range, Cabir would immediately start searching for another Bluetooth-enabled device. This empowered Cabir by now having the ability to infect more than one device per execution. Luckily, no reports of it in the wild ever emerged. The author of the Cabir.H variant was Velasco, who posted the source code on a malware Web page. A smaller difference was that this variation did not display any text onscreen once installation was completed. It only showed the SIS name and nothing else. Figure 4.2 shows the display.

Figure 4.2 Display of Cabir.H after Completed Installation

The Cabir.K variant was also identical to Cabir.H but had an added functionality employing MMS as a new vector of infection. When installation started, the MM displayed the following text on the screen:

```
Caribe Version 2 - ValleZ/29a
```

After this MM installed on a device, it would automatically respond to every incoming SMS and MMS with a reply MMS that contained a copy of the SIS file that would install the worm on the sender's device, if the user accepted of course. At this point in its evolution, Cabir was able to propagate to multiple devices via Bluetooth and MMS. Cabir.L is functionally the same as Cabir.H, with the only difference being a different binary form being recompiled.

The variants Cabir.M through Cabir.AB and Cabir.AD were functionally identical to Cabir.B, with the only noticeable differences being a different name for the SIS file and different text displayed on the device's screen. Most of these variants were again due to script kiddies performing hexadecimal edits to the code of Cabir.B. The only other difference of interest was found in Cabir.AA: when the worm was executed, a text message would display on the screen, along with an image (as shown in Figure 4.3).

Figure 4.3 Message from Cabir.AA upon Execution

The variant Cabir.AC was a minor hexadecimal edit of Cabir.AA with the difference being a different filename for the SIS file and different text displayed on the device's screen upon execution. Cabir.AE was a variation of the original Cabir.A with a significant difference being a new bootstrap component used to install the SIS file to a target device. Cabir.AF was functionally equivalent to Cabir.A, but the file size was smaller by a few kilobytes and when installation completed there was no text displayed on the device's screen. Three more Cabir variants were discovered in 2006, each ending with Cabir.AI.

For two years, Cabir evolved in a few directions, some more significant than others. It is now viewed as the original MM that ignited a flood of interest in MM and led to the release of many other novel and somewhat dangerous MM samples both in the wild and the zoo. The most significant variants of this family are:

- **Cabir.A** The original Bluetooth MM
- **Cabir.H** Fixed the distribution bug of Cabir.A, leading to wider propagation

- **Cabir.K** Clearly the most powerful variant in this family, with the ability to propagate via Bluetooth and MMS

One other lesson learned form Cabir is a reaffirmation that many variants will be produced when source code is released to the general public. Much of the black hat underground is fueled by sharing of code, and Cabir was no exception. What is notable is that, of the 35 known variants, most were hex edits of binary code leading to changes of filenames and display text. The more significant changes appeared in only a small number of the variants, and as rumor has it, by the same authors. This hints to the lack of knowledge in programming for Symbian OS at the time Cabir first appeared. It actually served as a class to learn the Symbian platform for software development, and as more proficiency in the operating system increased, so did the number of new and novel MM for this platform. But it was Cabir that started it all.

Skuller

A Trojan for the Symbian platform, Skuller (a.k.a., Skulls) rendered the victim's device useless with only the ability to make phone calls while all other features were disabled. This Trojan had over 90 known variants. It infected the device due to one of several vulnerabilities in the Symbian OS. Its most recognizable feature was the skull and bones icon used to replace the icons of existing application files installed on the device. The base file for the MM named Trojan.Skuller.gen was made available online and many people quickly used it to create their own variants.

The original MM, named SymbOS.Skulls.A, appeared in late 2004. It was packed in a SIS archive file named Extended theme.sis. It masqueraded as a theme manager file for the Nokia 7610 Smartphone claiming to have new icons and wallpapers usable on the device. The MM author went by the name of Tee-222. It was designed to only infect Symbian series 60 platform but strangely enough it also infected the Symbian series 90 platform as well. The Trojan did not carry any malicious code per se. What it did was overwrite application files with its own versions, which were exact copies extracted from the ROM of the device. It turns out that Symbian had a flaw that rendered system application files useless when they were overwritten by the same file extracted from the ROM.

Another effect was that the icon AIF files were replaced with new AIF files, which replaced the original icon with a Skull and Bones icon. The latter did not allow the application to be accessed by its shortcut. The AIF file containing the Skull and Bones icon was the only one that could be considered malicious for blocking access to the application of the icon that Skuller replaced. The worst thing a victimized device's user could do was reboot the device, which would render it totally useless. None of the functions worked except the phone component. Skuller was the first MM to use flaws of the Symbian OS that allowed system files to be replaced by the MM's own files without approval from the user. This novel

contribution opened the flood gates for other MMs to emerge that also used flaws present in the Symbian operating system.

Soon after the release of Skulls.A, its first of many variants, Skulls.B, was discovered. This variant was functionally identical to Skulls.A but had a few significant differences. First, the SIS file was changed to Icons.sis. Second, this MM displayed no text when being installed. Third, and most importantly, Cabir.B was included in the SIS file. When the Trojan was executed, Skuller would copy the caribe.sis file to the device and an icon for it would appear. Cabir would not install automatically, but if the user tapped the icon, the Cabir would install and start seeking other victim devices in an attempt to propagate.

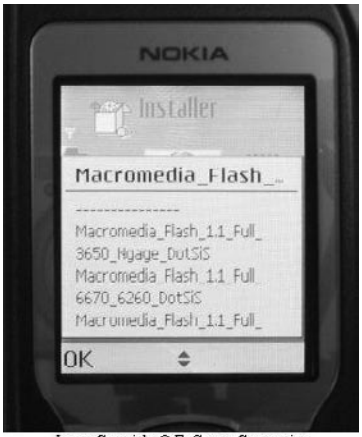
NOTE

Some virus companies reported a Cabir variant that carried the Skuller Trojan, even though it supposedly didn't work properly. It was one of the early MMs, along with Skulls.B, that became carriers of other MM.

Skulls.C was functionally equivalent to Skulls.A but had a few characteristics that were present in Skulls.B. It did not display text when installed. This MM carried Cabir.F in its SIS file and would copy it to the device. The MM Cabir.F would not run automatically, the user had to tap its icon and give permission to install its payload. The unique characteristic of Skulls.C was that it attempted to overwrite and disable the F-Secure antivirus software if it was installed on the device. This was the first time an MM specifically targeted a security application for disabling.

Skulls.D was a mix of both Skulls.A and Skulls.B. This MM was found both as a standalone SIS file and masquerading as a Macromedia Flash player for the Symbian series 60 platform. Figure 4.4 shows the masquerade in action.

Figure 4.4 Skulls.D Masquerading as Macromedia Flash Player



Skulls.D also carried Carib.M and copied it to the device. To install Carib.M, the user had to give permission since it would not install automatically. This MM only overwrote files related to security products and Bluetooth capabilities. Unlike previous versions of Skuller, this one specifically targeted overwriting the needed files to disinfect the device of the MM. Most interestingly, Skulls.D installed a third-party application that ran a new background image on the display screen that persisted regardless of which application was running at any given time. The new background image was a rather disturbing animated rendering of a skull that fills the whole display screen. Figure 4.5 shows a screen capture of the background image.

Figure 4.5 Background Image from Skulls.D

Notes from the Underground…

Black Hat Humor a la Geek

Skulls.D stored the background image used for display in the folder:

\nokia\images\nokias\malaysia\johor\pj\pj\pj\jb\jb\jb\imos\yuan\yuan\yuanyuan\blue\a-team\terence\ownpda\fuyuan.gif

If you stop and notice the folder names you see the country of origin "Malaysia," the possible name of the author "yuan," the words "a-team," "blue," and "imos." Most importantly, you see the intent of this MM "ownpda," related to the authors who call themselves Ownpda. Often, MM authors use descriptive folders to get their messages across to fellow authors and security experts, knowing that only they would stop and notice details like folder names—proving a Geek factor of 10 out 10.

At this point in its evolution, Skuller has proven to be capable of evolving into new variants that have very unique and novel characteristics, making Skulls.A through Skulls.D unique in its own way. Skulls.E was a minor variation of Skulls.C, only changing the name of the SIS file. It also copied to the device a slightly modified version of Cabir.F; the modifications were never made clear.

Skulls.F was a variant of Skull.D, but with a bigger payload. This MM copied to the device the MM Locknut.B and several of the early variants of the Cabir worm. None of this MM was automatically installed on the device once copied there by Skuller. Each one had to be individually executed and given permission by the device's user to install.

Skulls.G through Skulls.H were modified versions of Skulls.D. Skulls.H spread as NokiaGuard.sis and ScreenSaver.sis and also carried the MM Locknut.B and several Cabir variants. Skulls.I functions the same as Skulls.D but also carried Skulls.D in its SIS file along with a few Cabir variants. It is interesting to note that this was the first MM to carry an earlier variant of itself and so copied itself to infected devices. The potential of this was a device infected by multiple versions of the MM that initially infected it; this had not previously been seen.

A very weird variant appeared with the release of Skulls.J. This was a modified version of Skulls.D but had some significant differences. First, it did not carry any versions of Cabir or its own earlier variations. Instead, it carried the MM SymbOS.AppDisabler.A. Second, the display background image of a Skull was modified to appear all in black and was not animated. Most interestingly, however, Skulls.J did not carry the needed instructions to set the Skulls image as the background image. This code was found in AppDisabler.A, which also carried in its payload Cabir.Y and Locknut.B. The twist was that AppDisabler.A could not place the startup code for the Skulls background screen to appear. This is because Locknut.B, which

Appdisabler.A copied to the device, would block the attempt to place the startup code on the device. Thus, the Skulls image never appeared.

Skulls.K was a minor variation of Skulls.C that carried Cabir.M and the Skulls background image of Skulls.D. F-secure got a small scare when Skulls.L came out. It, too, was a minor variation of Skulls.C, carrying with it Cabir.F and Cabir.G. What caught people off guard was that this MM masqueraded as a pirated version of F-secure's antivirus software for mobile devices, with the name of the SIS file as F-secure_Antivirus_OS7.sis. Unsuspecting users were installing it thinking they were getting a fully working copy of the software without having to pay for it!!!!!! In an unexpected move, this MM taught users that piracy does not pay. Figure 4.6 shows some screen captures of this MM during installation and after infection.

Figure 4.6 Screenshots of the Effects of Skulls.L

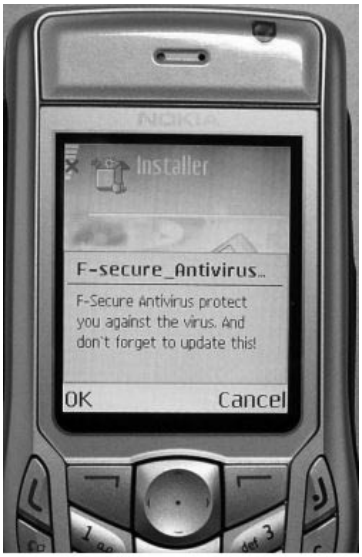


Image Copyright F-Secure Corporation

Image Copyright F-Secure Corporation

Skulls.M was a variant of the original Skulls.A, with a different Skull and Bones icon. Skulls.N through Skulls.O were a variation of Skulls.D. The MM Fontal.A and CommWarrior.B were carried by Skulls.O. The following variations, Skulls.P through Skulls.R, were a cornucopia of several earlier versions, with Skulls.D and Skulls.N being the most prominent. Skulls.P carried several other MM, including SymbOS.Mabir.A, Cabir variants, and parts of Fontal and Doomboot. A vicious part of the payload resulted from Doomboot, which did not allow the phone to be rebooted. The only way to disinfect Skulls.P from a device was with the use of a memory card.

After the release of Skulls.P, several other variants—the last named, Skulls.CL—were released in May 2006. All of the later variants were modified versions of earlier ones. One specific variant, Skulls.AG, carried in its payload the MM FlexiSpy.A, which is a known spying Trojan that records information on phone calls and test messages. A total of 90 known variants were documented for the Skuller family, making it one of the largest known MM families. Of all the variants, the following are the most notable:

- **Skulls.A** Overwrote system files without user knowledge and replaced icons, rendering their shortcuts useless.
- **Skulls.B** One of the first MM to carry another MM, Cabir.B, in its payload.
- **Skulls.D** Masqueraded as Macromedia Flash Player, and was installed more easily due to effective social engineering.

One of the biggest lessons learned from the Skuller family is the ease with which multiple MMs can be added to one MM and then copied to an infected device. Skuller had the potential, in some of its variants, to infect a device with up to six or more MM that literally could convert the device into an expensive paperweight.

Doomboot

The Doomboot Trojan first appeared in 2005 as Trojan.SymbOS.Doomboot. This family grew to have 25 known variants. The original sample carried as its payload the CommWarrior.B worm. It infected the Symbian OS based on one of the several vulnerabilities existing on that platform. This first version of Doomboot is what we like to call a double whammy. First, CommWarrior.B starts spreading immediately after being installed and runs as an invisible process on the device. This results in the user being unaware that the MM is executing, and most importantly, the battery is drained quickly. That's the fist whammy. Doomboot then installs corrupted system files in the device. These corrupted files will be loaded when the device is rebooted but will immediately cause the device to crash and not boot again. Combine this with the quick battery depletion, and you got your double whammy. The Trojan would arrive as the SIS file entitled Doom_2_wad_cracked_by_DFT_S60_v1.0.sis and masquerade as a cracked version of a popular game called Doom 2.

This minor social engineering is all that was used to trick the user into approving the installation. Once the installation finished, no display messages appeared on the screen and no new icons were added to the device's menus. Figure 4.7 shows Doomboot asking permission to install.

Figure 4.7 Doomboot.A ,Masquerading as the Game Doom 2, Asking Permission to Install

Soon after the original MM was released, its first variant, DoomBoot.B, appeared. This version was functionally identical to the original version, with the difference of not carrying any other MM in its payload. Instead, it carried an application that would cause the device to reboot, and due to some included corrupted files, the device would not be able to successfully reboot. It masqueraded as a utility named Restart_20.sis, which supposedly reboots the phone in the proper manner. Doomboot.C was equivalent to Doomboot.B, with the one difference being it masqueraded as a set of fancy effects for Nokia phones and used the file name: Nokia Camera Effects v1.05 by Dj 6230.sis.

The D version was also a minor variant of C with a twist. This MM masqueraded as a collection of images of actress Angelina Jolie, and surprisingly it actually did contain the images, a rarity for Trojans of this type. It used the name *Angelina Jolie Theme(Universal Theme).sis*. Once installed it would replace the background image with one of Jolie. Doomboot.E was exactly the same as the D version, but their model of choice was Jennifer Lopez, with the filename *Jennifer Lopez Theme++ by Dj Hardcore.sis*.

Doomboot.F follows in the path of Doomboot.D, with the added bonus of having Fontal.A and CommWarrior.B in its payload. Doomboot.G through Doomboot.N are all variants of earlier versions, each one carrying corrupt files to install on the device. They also carried portions of other known MM, and all had the capability of crashing the device by not allowing a reboot to occur. The message displayed by Doomboot.L after installation is shown in Figure 4.8.

Figure 4.8 Message Displayed by Doomboot.L after Installation

```
This installation was created with KVT Symbian Installer. Get it free
from:
<domain>
by Kheng Vantha
---------------
This will incrase the speed!
Enjoy, regards DFT!
```

The variant Doomboot.O was a very simplified variant of earlier versions. In fact, it did not perform many of the malicious acts of its predecessors. Instead, it carried three known malicious MM in its payload and copied them to the victimized device. In addition, it corrupted system files causing the device to fail on reboot. The three MM carried in the payload were:

- SymbOS/Cabir.B
- SymbOS/CommWarrior.B
- SymbOS/Cdropper.H

This version of Doomboot stands out from the others for breaking the pattern of being a modified version of an earlier variant. It can be labeled an early "B-52 Bomber" of this MM family. It is definitely not the biggest carrier of other MM as we shall see next. Several more variants of this family arose, all of which were similar in carrying other MM in their payload and rendering the device useless by causing a system crash on reboot. Of these later variants, two stand out from the rest. Doomboot. P carried in its payload the following files:

- \system\RECOGS\flo.mdl – SymbOS.Cabir
- \system\symbiansecuredata\caribesecuritymanager\sexxxy.sis – SymbOS.Cabir
- \system\apps\OIDI500\OIDI500.mdl – SymbOS.Cabir
- \system\apps\OIDI500\OIDI500.app – SymbOS.Cabir
- \system\apps\caribe\flo.mdl – SymbOS.Cabir
- \system\apps\caribe\caribe.app – SymbOS.Cabir.B

- \system\CARIBESECURITYMANAGER\caribe.app – SymbOS.Cabir.B
- \system\apps\gavno\gavno.app – SymbOS.Locknut.A
- \system\apps\AppMngr\AppMngr.aif – SymbOS.Skulls.C
- \system\apps\Menu\menu.aif – SymbOS.Skulls.C
- \System\Apps\Phone\Phone.aif – SymbOS.Skulls.C

The files carried in the payload were in fact four previously discovered MM, all of which were copied to the victim's device. These copied MM did not automatically install on the system. They each had to be run and given permission by the device's user to successfully infect. This MM also replaced icons on the display menu with its own customized icon that rendered the shortcut to the original icon's application useless. This was reminiscent of the Skulls family, which made icon replacement popular amongst MM authors. This MM also carried corrupted system files, causing the device to crash on reboot.

The super "B-52 bomber" of this family is without question Doomboot.S. This variant carried ten known MM in its payload, making it the biggest carrier of other known MM in this family. It also had the distinctive trademark of copying corrupted system files onto the device, causing it to crash on reboot. The ten MM it carried were as follows:

- SymbOS.Blankfont.A
- SymbOS.Cabir
- SymbOS.Cabir.C
- SymbOS.Cardblock.A
- SymbOS.CommWarrior.A
- SymbOS.Fontal.A
- SymbOS.Mabir.A
- Trojan.Mos
- SymbOS.Pbstealer.A
- SymbOS.Sendtool.A

The variants for this family totaled 25 known, with the last one, DoomBoot.y, appearing in mid-2006. Of all the variants, five stand out:

- **Doomboot.D** Replaced background image with Angelina Jolie, good use of social engineering
- **Doomboot.E** Replaced background image with Jennifer Lopez, good use of social engineering

- **Doomboot.O** Early variant carrying several known MM in its payload
- **Doomboot.P** Modified display icons; reminiscent of the Skuller family
- **Doomboot.S** Carried ten known MM in its payload, more than any other Doomboot variant

This family's contribution to MM is twofold. First, all of its variants kept the same basic payload active, which was to install corrupt system files that always caused the device to crash on reboot. This portion of the payload was never absent from any of the family members. This could be the result of the same authors creating all the variants or of script kiddies that were not able to hex edit the portion of the original Trojan that carried this part of the payload. In either case, the whole family carried the same payload portion to cause a system crash on reboot. The second contribution from this family is its insatiable thirst for being a carrier of other known MM. Practically every variant carried at least one other known MM in its payload. This trend of carrying other MM in the payload was started with the Skuller family and possibly Cabir. But it was Doomboot that really brought an MM carrying payload to the main stage of the malware world.

Cardtrap

Yet another Trojan for the Symbian platform, the Cardtrap family has 38 known variants and a multicomponent payload. It first appeared in September 2005, infecting Nokia phones running the Symbian OS via one of the many known vulnerabilities existent in that platform. The payload of Cardtrap did the following: deleted antivirus files; rendered installed applications useless while installing other dummy applications; and, most interestingly, installed the Win32/Padobot.z and Win32/Rays viruses to any memory card installed on the device. When the memory card was installed in a PC, the two viruses would attempt execution and infection of the PC. Cardtrap was the first cross-platform MM employing memory cards to distribute W32 malware to windows systems in an attempt to infect those platforms. It was the first MM attempting to infect two distinct operating systems: Symbian *and* Windows.

The Cardtrap.A Trojan spread in a SIS archive file named Black_Symbian v0.10.sis. The MM would corrupt several system files and third-party applications by overwriting their main executable files. It would also check for the presence of a memory card. If one was found, it would install the viruses W32.Padobot.Z and W32.Rays to the card, along with an autostart file. These two malware infect the Windows platform, not Symbian. If the memory card is placed in a Windows system, the startup file attempts to infect that system with the two Windows payloads.

Cardtrap.B functioned the same as the A version, but also carried components of the MM Doomboot.A, which would cause the device to crash on reboot. Cardtrap.C follows its predecessor but does not carry any Windows malware. Instead, it has components of SymbOS.Lasco.A MM. This was copied to the memory card, and if inserted into a Windows

system would attempt infection of all SIS files found in the Windows system. Testing showed this failed due to mission or corrupted files needed by Lasco to function properly. Both Cardtrap.D and Cardtrap.E are minor variants of Cardtrap.B with the one difference that these two variants corrupt a smaller number of the device's applications than Cardtrap.B.

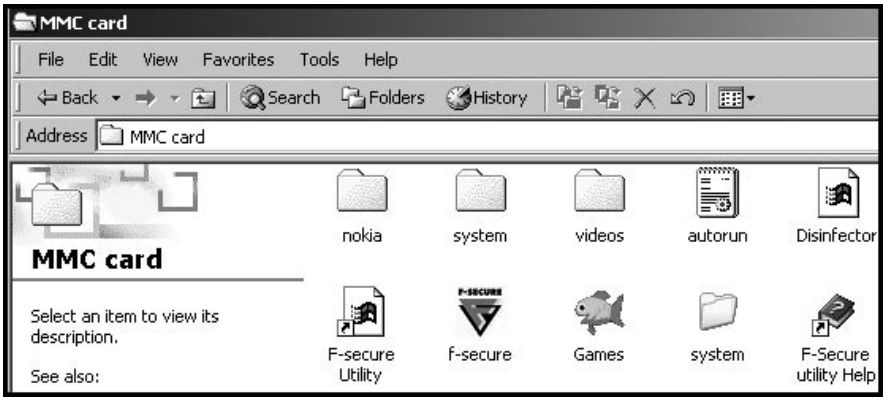
Both Cardtrap.F and Cardtrap.G execute the same as earlier versions but carried three Windows malware:

- W32.Rays
- W32.Padobot.Z (a.k.a., Korgo family)
- W32.Cydog.B

Each of these viruses were installed to the memory card with an auto start file. If the memory card was installed in a Windows machine card reader, all three would attempt infection. Cardtrap.H through Cardtrap.L similarly carried W32 malware in the payload to copy to any present memory cards on the victimized mobile device. Some security companies claimed Cardtrap.L did not function properly… yet it still executed its entire payload and rendered the phone useless on reboot—so that doesn't exactly sound like a nonfunctioning MM to us.

Cardtrap.M and Cardtrap.N carried several Windows and Symbian malware. They used heavy social engineering to trick users into installing the malware carried in its payload. This MM would use icons of applications such as F-Secure to trick Windows users into installing the W32 malware from the memory card to the windows system. As expected, F-Secure was up in arms about this, seeing it as a valid threat to their reputation, and rightfully so. Figure 4.9 is a screen capture of an infected memory card with the misleading icons.

Figure 4.9 Misleading Icons on a Cardtrap.M- and Cardtrap.N-Infected Memory Card



The Windows malware carried by Cardtrap.M and Cardtrap.N were the following:

- Virus.Win32.Kangen.a
- E-mail-Worm.Win32.Brontok.c
- VBS.Starer.A
- VBS.Soraci.A
- Trojan.Win32.VB

This MM also carry the following Symbian MM, which would masquerade as benign applications to trick users into installing them on the mobile device:

- SymbOS/Doomboot.K
- SymbOS/Cabir.AB
- Symbian dropper for Win32/Istbar.IS

Cardtrap.O through Cardtrap.AL, this family's last known variant, were all similar to Cardtrap.N, with the only difference being the types of MM carried in their respective payloads. The last variant of this family, Cardtrap.AL, was discovered in September 2007. The variants of this family that made the most novel contributions were:

- **Cardtrap.A** The first cross-platform MM using a memory card to propagate
- **Cardtrap.F** Contained multiple Windows malware in its payload
- **Cardtrap.M** Held several Windows and Symbian malware; implemented through effective social engineering

This family, with its 38 variants, tied together some of the characteristics of previous MM. It really made the most of carrying other MM, a characteristic found in both Skuller and Doomboot. But it was the first MM to attempt infecting two separate operating systems, thus establishing itself as an early cross-platform MM. Its one drawback was that the Windows malware had to be placed on a memory card. This memory card then had to be inserted in a Windows system card reader. In some cases, once this happened the malware would automatically infect the device, but in others the user had to run the executable for infection to occur. This series of steps held back propagation and resulted in a less effective MM.

Summary

This chapter examined some of the largest known MM families, namely Cabir, Skuller, Doomboot, and Cardtrap. Each one offered several novel contributions to the world of MM. Several lessons were learned from analyzing these families. Source code released to the public led to several variants producing distinctly different variants with very unusual effects. This further shows the danger of releasing source code to the general public, even though it's a double-edged sword. Security researchers can use the same source code of analysis and antivirus solutions. Technologies such as Bluetooth and memory cards on mobile devices were shown to be very effective vectors of infection and distribution highly used by some or all of these families. It is always interesting to see how authors change variants within the same family. Even script kiddies doing hexadecimal edits are able to accomplish a lot, such as create payloads carrying other MM, text displays to show off their names and boost their egos, display images on mobile device backgrounds, and more.

As we move forward in the evolution of MM, new families will arise, showing similar traits in their variations, just as these families have. They will be closely related to each other, making detection much easier, both from a signature and behavior point of view. The variations will differ in key areas, usually those dealing with payload and infection. As we saw with Cabir, a major difference in one variation was fixing the Bluetooth bug. In Skuller, Doomboot, and Cardtrap, the payloads changed by carrying different numbers and samples of known MM. What is clear is that the functionality of these variants will likely not change significantly. The core components of the families seen here were never highly modified. This only occurred to fix flaws in the logic of the code. One other interesting observation that we should see is when something works well, there's no need to change it except to maybe improve it. The Doomboot family all installed corrupted system files to cause the device to crash on reboot. Even though the variants changed in other parts of the MM, including the payload, this portion was never changed or removed, only improved in some cases.

Future MM families have a great set of foundation samples to learn from and build upon. Their novel contributions will likely use parts of the mobile device not seen in these families but will remain consistently used in their variants. They will have faster distributions and scarier payloads then have been seen so far, but their family evolution will foundationally be the same as the families analyzed here.

Solutions Fast Track

Cabir

- ☑ Cabir was the first Bluetooth MM, with 35 variants.
- ☑ Cabir variants fixed Bluetooth distribution flaws and added MMS distribution.
- ☑ Of the 35 known variants of Cabir, most were hex edits of binary code leading to changes of filenames and display text.

Skuller

- ☑ Skuller was an early carrier of other MM in its payload, with 90 variants.
- ☑ Skuller increased payloads by carrying other MM and modifying display text and images.
- ☑ One of the biggest lessons learned from the Skuller family is the ease with which multiple MMs can be added to one MM and then copied to an infected device.

Doomboot

- ☑ The Doomboot Trojan first appeared in 2005 as Trojan.SymbOS.Doomboot.
- ☑ Doomboot added several known MM to its payload.
- ☑ Doomboot "B-52 Bomber" of Symbian MM had 25 variants.

Cardtrap

- ☑ Cardtrap first appeared in September 2005, infecting Nokia phones running the Symbian OS via one of the many known vulnerabilities existent in that platform.
- ☑ Cardtrap was the first cross-platform MM using memory card to propagate with 38 variants.
- ☑ Cardtrap variants were packed with increasing numbers of Symbian and Windows malware.

Frequently Asked Questions

Q: Are any MM reported in this chapter still a threat?

A: Yes and no. Most of these never went into the wild; the ones that did may still be roaming around and can infect mobile devices not equipped with antivirus software. If you harden your device against attack, such as setting Bluetooth to "hidden" and not discoverable, the chance of infection is negligible.

Q: Why are there so many variants of these families?

A: This may be due to a few reasons. Source code made available to the public allowed other MM authors to create new and better variations. Script kiddies can perform hex edits to the executable files, creating variations with minor changes. Some MM carry other files with them. These files can be readily changed since the filenames are not hardwired into the MM code.

Q: Should we expect future MM families to contain as many or more variants as Skuller?

A: Absolutely! It's a given that future MM will leak source code out and script kiddies will continue performing hex edits to create new variants. The real issue is if a particularly destructive and hard-to-detect MM produces many variants, some damage may be incurred before it is contained.

Q: What impact can these families have on future MM malware?

A: Just like other early MM samples, they serve as examples of what can be done with mobile devices and help stir the imagination of what can be done next.