VIRUS ANALYSIS

PARADISE LOST

Peter Ferrie and Frédéric Perriot
Symantec Security Response, USA

Eight months ago, Peter Ferrie and Péter Ször asked at the end of their article on SymbOS/Cabir: 'What will be next? A mass mailer using MMS?' (see *VB*, August 2004, p.4). The answer was yes, that is what came next.

SymbOS/Commwarrior.A is the first worm to use MMS (Multimedia Messaging Service) technology to spread on cellular phones. Following in the footsteps of Cabir, it also replicates using Bluetooth, though with some improvements in its implementation. This double-pronged approach to replication makes Commwarrior a more likely candidate to be seen in the wild – although, at the time of writing, no such reports have been received.

As with Cabir, Commwarrior replicates only on *Nokia Series 60*-compatible devices.

CONSULTING ON THE SUM OF THINGS

The worm begins by counting the number of copies of its process that are running. Generally, it will exit if another copy is running already (unless many copies start at the same time).

Next, the worm retrieves the machine identification number, and calculates an additive sum of the characters, to produce a unique value. This value might have been used by the worm's author during testing to avoid the infection of his own device, but now the result is simply discarded.

TO NONE ACCOUNTABLE

The worm walks the list of running processes, and renames itself to the name of the first process in the list (which is usually 'EKern', the system kernel), followed by some random numbers.

In addition, the worm changes its owner and type to those of the first process's owner and type. Finally, the worm protects its process to prevent any other process from changing the priority of, or terminating, its process. Any attempt to terminate the process, by using a tool such as 'Switcher', is simply ignored. The screenshots in Figure 1 show the process list, with the legitimate EKern at the top of the list, and the worm process at the bottom of the list. Notice the size difference in memory.

If the worm has not been run from the 'c:\system\updates\commwarrior.exe' path, it creates the directories 'c:\system\updates' and 'c:\system\recogs', then

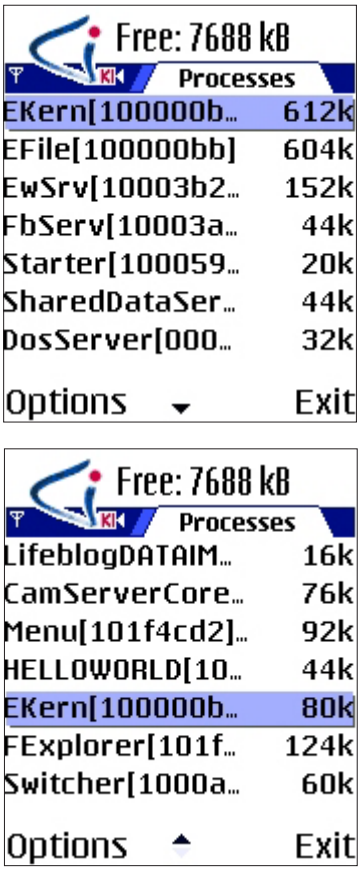

Figure 1. The legitimate EKern process at the top, and the worm process at the bottom of the list.

copies the 'commrec.mdl' file to the 'updates' and 'recogs' directories, and the 'commwarrior.exe' file to the 'updates' directory.

The 'commrec.mdl' file is a MIME recogniser file. It is intended to run the 'commwarrior.exe' file from the 'updates' directory whenever the phone starts, however on recent models of phones, such as the *Nokia 7610*, this does not work.

The worm creates a SIS file named 'c:\system\updates\commw.sis', by appending the 'commwarrior.exe' and 'commrec.mdl' files to the SIS header that is carried in its code. The SIS file uses the store method only – no compression is used – and the 'commwarrior.exe' file is marked to auto-execute on completion of the installation.

AS SOFT AS NOW SEVERE, OUR TEMPER CHANGED

The worm contains various texts, but the most amusing is the one that reads 'OTMOPO3KAM HET' (which translates

roughly as 'No to softheads!'). According to our colleague Sergei Shevchenko, this is Russian slang, and the word for 'softheads' identifies someone whose brain has frozen so that the person has lost his ability to think and control himself.

BOTH WHEN WE WAKE, AND WHEN WE SLEEP

The replication strategy of the worm is interesting because it adapts its infection vector according to the time of the day.

The worm uses Bluetooth during the normal waking hours of the phone's owner, when it is most likely to have other Bluetooth devices in range. It uses MMS during 'sleeping hours', and cleans up the sent-message logs carefully afterwards. In addition, the worm intentionally sets a lower priority to the replication threads, to make their activity less noticeable.

The overall scheduling of the worm's replication is accomplished by a single timer, which is set to trigger every ten seconds. Within the main timer callback, the worm checks for the payload condition, the time of day and the Bluetooth state, in order to pick a replication method.

The worm favours finishing any on-going Bluetooth replication cycle over sending MMS messages. Its schedule looks like this:

- 08:00am – 11:59pm Bluetooth replication
- 12:00am – 06:59am MMS replication
- 07:00am – 07:59am MMS queue cleanup

WHY HAS THOU ADDED THE SENSE OF ENDLESS WOES?

On the 14th day of any month, in the hour between midnight and 12:59am, Commwarrior's payload triggers. The worm's payload is to warm-boot the phone unconditionally.

Since the reboot does not cause the phone to switch itself off, and because the worm is part of the boot cycle, the phone could continue to reboot until the payload time ends.

DIM ECLIPSE, DISASTROUS TWILIGHT

The Bluetooth replication code differs from that seen in Cabir, in that it enumerates all the devices in range, whereas Cabir attempted to infect only the first device in range.

When the Bluetooth replication cycle starts, the worm enumerates all devices in range and builds a list of present devices. It queries each device for the availability of the 'Obex Push' service which is necessary to upload files. Devices meeting this condition are sent a copy of the SIS file of the worm, renamed to a random string which is eight characters in length, and consists of lower case letters and digits.

Once the worm has attempted replication to all devices that were found, it tears up all the connections and a new Bluetooth cycle can start. The first Bluetooth cycle does not start until 50 seconds after the worm process starts, in order to let the phone boot completely. A new Bluetooth cycle can, in theory, be triggered every 50 seconds, but if there are many devices within range, it may be slower than that.

RECEIVE THY NEW POSSESSOR

The worm's MMS functionality can be considered the equivalent of mass-mailing used by viruses on the *Windows* platform. This makes us very afraid that it will become the replication method of choice among any future self-replicating malware for cellular phones.

Commwarrior sends one MMS message at a time (i.e. one new MMS message at most every 10 seconds, since a message might take more than one cycle to complete). The recipients are picked randomly from the phone book.

On each cycle, one contact is picked at random, then Commwarrior enumerates the information fields of this contact, and selects from there the fields that correspond to mobile numbers only. This means that home numbers and work numbers (i.e. land line numbers) are ignored, in an attempt to maximise the chance of hitting other compatible cellular phones.

If a contact entry in the phone book contains several mobile numbers, then the MMS message is sent to all of those numbers.

From the debugging messages and code snippets in the Commwarrior code itself, it is possible to determine the origin of much of the MMS code used in the worm. Most of it was copied from a developer's page on a website, and altered slightly to add support for binary attachments (in fact, most of the rest of the code was copied too, from the *Symbian* SDK samples).

IN THIS PERFIDIOUS FRAUD, CONTAGION SPREAD

For each of the MMS messages that Commwarrior sends, the subject and message body are chosen randomly from the following list:

Norton AntiVirus
Released now for mobile, install it!
Dr.Web
New Dr.Web antivirus for Symbian OS. Try it!
MatrixRemover
Matrix has you. Remove matrix!
3DGame
3DGame from me. It is FREE !
MS-DOS
MS-DOS emulator for SymbvianOS. Nokia series 60 only. Try it!
PocketPCemu
PocketPC *REAL* emulator for Symbvian OS! Nokia only.
Nokia ringtoner
Nokia RingtoneManager for all models.
Security update #12
Significant security update. See www.symbian.com
Display driver
Real True Color mobile display driver!
Audio driver
Live3D driver with polyphonic virtual speakers!
Symbian security update
See security news at www.symbian.com
SymbianOS update
OS service pack #1 from Symbian inc.
Happy Birthday!
Happy Birthday! It is present for you!
Free SEX!
Free *SEX* software for you!
Virtual SEX
Virtual SEX mobile engine from Russian hackers!
Porno images
Porno images collection with nice viewer!
Internet Accelerator
Internet accelerator, SSL security update #7.
WWW Cracker
Helps to *CRACK* WWW sites like hotmail.com
Internet Cracker
It is *EASY* to *CRACK* provider accounts!
PowerSave Inspector
Save you battery and *MONEY*!
3DNow!
3DNow!(tm) mobile emulator for *GAMES*.
Desktop manager
Official Symbian desctop manager.
CheckDisk
FREE CheckDisk for SymbianOS released!MobiComm

(Due to what appears to be a missing terminating character, the last message body appears to contain the subject ['MobiComm'] for the next message body ['MobiComm, Mobile communications inspector. Try it!'] which is never referenced.)

This worm's use of social engineering is very similar to that seen in many email worms, and has proven very successful in the past. The MMS messages contain an attachment whose name is always 'commw.sis'. The attachment is the worm installer, and its MIME type is set explicitly to 'application/vnd.symbian.install'.

The worm maintains a list in memory of all of the recipients of its MMS messages, and uses the list to avoid sending multiple messages to any recipient. In the event that the phone is switched off (or the payload executes), the list will be lost, and recipients will receive additional messages if the worm process is executed again.

In the early hours of the morning, the worm cleans up the MMS queue. This means that the user will not be alarmed by any worm messages in the 'Sent' box.

JOURNEYED ON, PENSIVE AND SLOW

One mitigating factor to the success of the MMS replication method is that the phone operator interoperability seems to be very limited. Indeed, during our attempts to send our own test messages, we experienced many failures to send messages at all between different providers, and long delivery times.

It should be noted that, upon receipt of the SIS file, whether by Bluetooth or MMS, the user must agree explicitly to its installation via several dialog boxes. If, at any point, the user cancels the installation, the worm does not execute.

CONCLUSION

Due to its openness and the ready availability of development tools, the *Symbian* platform appears to be a fertile ground for new malware, and will become a required area of expertise for current and future anti-virus researchers. The fact that the *Symbian* OS is designed to run on embedded platforms, whose resources are limited, and that its core APIs are based on C++, can throw off reverse engineers who are used to the PC platform.