# VIRUS ANALYSES

## Poetry in Motion

*Péter Ször*
*Symantec, USA*

This month Péter Ször takes a look at three *Windows* viruses, the first two of which are recently released mass-mailing worms. The third analysis is concerned with a potentially dangerous variant of WinNT/Infis.4608, now updated for *Windows 2000*.

### 1. Win95/Haiku

The number of mass-mailing email worms is rising very rapidly. At least one third of the 32-bit *Windows* virus variants written this year alone can be classified as mass-mailing worms. Win95/Haiku.16384 was created by a longtime 'retired' founder member of the 29A group who calls himself 'Mr Sandman'. His creations include the infamous Esperanto *Windows* virus released some years ago (see *VB*, December 1997, p.3). Since then he has been very quiet and nowadays he is no longer part of 29A. He is very interested in languages, claiming to speak several, and works as a professional translator.

Win95/Haiku is his first mass-mailing worm. The really interesting thing is not its mailing routine, but its functionality. Haiku is capable of creating small poems, so-called 'haiku'. The worm propagates itself by sending emails with an attachment called HAIKU.EXE.

### The Story of Haiku

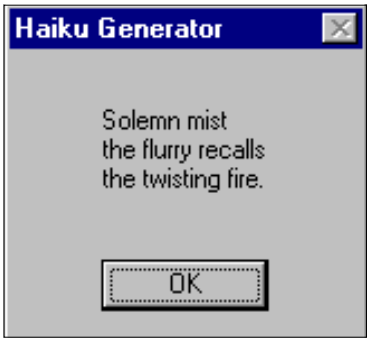
The subject of the email is 'Fw: Compose your own haikus!', so it looks like a forwarded message. The body contains a small introduction to the haiku form:

```
:))
—— Original Message ——
>"Old pond...
> a frog leaps in
> water's sound."
>- Matsuo Basho.
>
>DO YOU WANT TO COMPOSE YOUR OWN HAIKUS?
```

A haiku is a small, oriental-metric poem that first appeared in the sixteenth century. It is popular mainly in Japan and the USA. Apparently, its form transcends the limitations imposed by language structure and the scientific philosophy which treats nature and the human being as machines.

The poem usually consists of three lines and 17 syllables, distributed in five, seven and five format. It must register or indicate a movement, sensation, impression or drama of a specific fact of nature, rather like a photograph. More than inspiration, what you need in order to compose a real haiku is meditation, effort and perception.

### Initialization




When the attached HAIKU.EXE is executed it installs itself on the system by copying itself to the *Windows* directory as HAIKUG.EXE (Haiku Generator). Then it modifies the RUN field in WIN.INI under the section in order to execute HAIKUG.EXE at each system start from then on. Then the worm displays a haiku message box. Win95/Haiku randomly selects words from a word table. Some words may have different endings using 's' and 'es', respectively. The first few words in the table are: 'bridge light sea fish butterfly foghorn day moon evening spring sunset boat petal blossom stone mist passage darkness dolphin ant shadow star frost… '.

### Mail Propagation

The worm searches on the local hard disk for .DOC, .EML, .HTM, .RTF and .TXT files, opens them and checks if they contain any email addresses. Thus, Haiku is more like a spam generator – it does not determine emails on the fly.

Then the worm connects to IP address 194.106.68.104 and uses port 25 (mail). This server appears to be opened for anonymous usage. Anybody can log in and instruct the mail server to send emails. This is a very common security problem that is used by spam authors often. This will, of course, limit the worm's lifetime to the period when the mail server is open for anybody. The worm's mail engine uses the SMTP protocol to send emails.

First, it introduces itself to the server 'HELO haiku.com'. It then sends the email: 'MAIL FROM: haiku@haiku.com'. After sending the email the virus leaves the server with the 'QUIT' message. Haiku uses MIME encoding for the attachment. During propagation the worm may display a message box with the following encrypted text:

```
[ I-Worm.Haiku, by Mister Sandman ]
 The smallest box may hold
 The biggest treasure?
```

The Win95/Haiku worm also connects to 206.132.185.167 (http://www.xoom.com) and uses the GET command to download a *Windows* WAV file (…/HAIKU_WAV/HAIKU.WAV). It creates C:\HAIKU.WAV and plays the WAV. Finally, it deletes the WAV file. The header of the WAV file contains the copyright message: (c) Mister Sandman, 2-2000. The worm's propagation is speeded up because Win95/Haiku's code does not have to carry the 56 KB WAV file.

### 2. Win95/Fix2001

This is a relatively 'old' worm which was created during the autumn of 1999. At that time it was not particularly widespread in the wild. It took a few months for Fix2001 to get any real attention. Several companies in the US were hit by it in December and in January the number of submissions to *SARC* showed that Fix2001 is really out there, all around the globe.

Win95/Fix2001 is an Internet chain-letter worm that will secretly steal dial-up information (including the password) and send it out via email to the hacker. This capability makes it really dangerous, since a hacker can use the information to hack into previously infected networks unless the passwords are changed. For a few weeks the worm's mechanism was unknown to all major anti-virus vendors. This is because it uses a very sophisticated method to access the *Windows 9x* dial-up passwords. It gets this information from the active RASAPI32.DLL in memory.

The worm arrives via email as a MIME-encoded attachment named Fix2001.EXE. The subject of the email is 'Internet problem year 2000'. It is sent by a person named 'Administrator'. The body of the message contains a message written in Spanish and English encouraging users to use the email attachment to check for Y2K compatibility. Unfortunately, several corporate users believed it.

#### Initialization

When executed, the worm installs itself on the local PC's *Windows* system directory with the name Fix2001.EXE. It modifies the Registry's …\Currentversion\Run field to execute itself during subsequent reboots. When executed for the first time, it will display the following message:

```
Y2K Ready!!
Your Internet Connection is already Y2K, you
don't need to upgrade it.
```

The worm checks if a window procedure with the name 'AMORE_TE_AMO' exists. An already active worm creates this window procedure in order to send itself to other locations in the background. This way, there will be only one active copy of the worm in memory. Instead of modifying system DLL files on the hard disk, the worm hooks APIs to itself in memory by patching the process address spaces. Thus it will gain execution each time any Internet activity happens on the local machine. The technique and its implementation are unique to Fix2001.

When RNAAPP.EXE (Dial-up Network Application) is not running the worm executes it with the '-l' parameter. This will load RNAAPP.EXE silently. RNAAPP.EXE has import functions from RASAPI32.DLL and this is in the interest of the worm. Fix2001 patches a hook routine to RASAPI32.DLL's DialEngineRequest() API later on when RNAAPP.EXE is loaded. It puts a jump that points to its hook routine at the entry point of the DialEngineRequest() API, and patches its short code right after the import address table of RASAPI32.DLL. A string should appear right next to the empty area. Then the worm checks if a long enough area filled with 0 bytes is available and only patches the process if this is the case.

Fix2001 also hooks the 'send' and 'connect' APIs of WSOCK32.DLL loaded by Internet applications such as *Internet Explorer* or *Outlook Express*. This is a very similar technique to the one used by Win32/SKA.A, with the important difference that this patch is done in memory and not in the file. This provides the worm with the same potential to spread as SKA – a proven technique.

Once RNAAPP.EXE is patched, the worm hides it from the task list by registering it as a service process. The worm itself is registered as a service process too and therefore it does not appear on the task list. Since many utilities that list processes do not display service processes (that can be accessed only by specifying an additional bit for the process query function) it is not particularly easy to notice that the Fix2001 worm is loaded in memory.

The hook routine on the 'send' API looks for the 'RCPT' field of the mail header during postings. The worm sends its message with the Fix2001.EXE attachment to the very same place right after the original message. This is much the same idea as that used by several known email worms. The received email headers will always contain a header reading: 'X-Mailer: PUPI-MAIL v.0.1'.

#### Posting Dial-up Passwords

Via its hook function, Fix2001 is capable of searching for user information in the address space of RASAPI32.DLL. The function searches for a 'T' or 'P' character at specific locations – the locations of the user information data. This routine sets a flag when successful and only sends the information once to one of the hacker's three email addresses. Used email addresses are encrypted in the code of the worm. The phone line text message might start with 'T' or 'P'. (The first line is the machine name, the next is the dial-up number, then the user name comes and the last line is the password.)

#### Payload

The payload is activated after the worm has already posted itself to another location and an active connection exists. Then, a routine will perform a checksum on the last detected email address. If a particular email address encounters a checksum match, the worm will delete the C:\COMMAND.COM file and create another 16-bit COM program, named COMMAND.COM, that is 137 bytes long.

The Trojan will be executed next time the computer is booted. When the trojanized COMMAND.COM is executed, it will destroy the hard disk data (it overwrites it using I/O port commands) whenever the hard disk is an IDE drive. This can be a targeted attack against specific people, but the checksum can all too easily match someone else's email address by accident.

### 3. Win2K/Infis.4608

[*Readers are advised to refer to p.8 of November 1999's issue when reading this analysis. Ed.*]

A week after *Windows 2000* shipped, the WinNT/Infis.4608 virus was updated to support *Windows 2000*. Win2K/Infis, a 'memory resident', parasitic *Windows 2000* Kernel-mode driver virus, only operates under *Windows 2000* and is already likely to fail under the first service pack. It does not have a payload.

When the INF.SYS driver takes control the virus allocates a memory from the non-paged pool, reads its complete copy from the INF.SYS file for future use in its infection routine, and hooks INT 2Eh by patching the Interrupt Descriptor Table (IDT). This is all possible because drivers have the most powerful rights on a *Windows 2000* machine

INT 2Eh is the main *Windows 2000* service interrupt (just like in *NT*) and it is completely undocumented. A Win32 application normally calls an API from the Win32 subsystem. The subsystem translates the documented API calls to undocumented once exported from NTDLL.DLL. The NTDLL.DLL is the native *Windows 2000* API. It has hundreds of undocumented APIs. NTDLL.DLL is running in User mode, but it switches to Kernel-mode by using the INT 2Eh service interrupt with a function ID in the EAX register (on *Intel* platforms). Each function ID is created by a macro when *Microsoft* compiles *Windows 2000*. Therefore, the ID can be different between new releases of *W2K*.

Since Infis uses hard-coded IDs it will not be compatible with all *Windows 2000* releases. The most important modification in the virus is the new ID number usage. The parameters of the API calls are passed on stack. This way the appropriate *Windows 2000* kernel API will be called.

The INT 2Eh hook of the virus intercepts the file opening function only, checks the file name and extension, then opens the file, checks the format and runs the infection routine. (Infis only uses INT2Eh functions, even when an infected User mode application is executed and the virus User mode entry point is called. Thus, it completely bypasses *NT's* Win32 subsystem.)

Checking the loaded driver list can be tricky because Windows *2000* places the driver list under the Computer Management. First, you need to turn on the 'Display Administrative Tools' option for the taskbar. Then, click on the 'Computer Management' and select 'Device Manager'. The View has to be changed to 'Show hidden devices'. The 'inf' driver should appear on the list. With a right-click on the driver name you can disable the driver. The 'Properties/ Driver' tag also allows the driver to be stopped (this is because Win2K/Infis has a driver unload routine).

While Win2K/Infis still infects some files incorrectly, it is more stable than its predecessor. Unfortunately, such new driver viruses can use the CIH damage routine under *Windows 2000* since drivers can execute port commands.

# FEATURE

## A Nightmare on Researcher Street

*Andy Nikishin & Mike Pavluschick*
*Kaspersky Lab, Russia*

As is often written in *Virus Bulletin*, it is always a little daunting to predict the future – what if predictions come true? Some time ago we discussed polymorphism in macro viruses and in the last part of that article we talked about the future of polymorphic macro viruses (see *VB*, June 1999, p.14). Back then we said that it would be possible to create real, strong polymorphic viruses using VBA5. It looks like our predictions came true.

At the end of December 1999, a Russian virus-writing group released its magazine – *DVL*. The issue contains write-ups on different kinds of viruses and various other articles. One of them piqued our interest – it was a little essay called 'Polymorphism in Word 97'. To be honest, we have read a lot of this kind of thing and we must say that most of them are pretty dull, but this one really impressed us. The author of this particular piece approached polymorphism in a different way.

Most recommendations for polymorphism suggest adding either comments in random places or unusable variables in code to confuse heuristic analysers and complicate virus analysis. This method has one main disadvantage – in a few 'virus generations' the virus will grow, so the macro stops working. A good example of such a virus is W97M/Groov. The size of its original code is about 6 KB, but the third generation is about 10 KB and so on. In the *DVL* article a virus writer suggested using good old file virus technologies – encryption and a polymorphic decryptor containing a garbage instruction which looks like a useful one:

```
RKFe5 = 1 ' Decryptor's part
Do While RKFe5 <= Len(Y7) ' Decryptor's part
Do Until o0Bukn4 > 30
o0Bukn4 = o0Bukn4 + 2
Loop
LjPvXw8 = (UsRgNN5 + BgaB0) Mod 255 '
Decryptor's part
jEcmjs1AXhT5 = 78
DpOjLoB1QaZzu8 = 151
LNTL1oGAFc7 = 0
IsMb2io0 = 175
Do While LNTL1oGAFc7 < 52
LNTL1oGAFc7 = LNTL1oGAFc7 + 5
Loop
cxJJIVJ3 = Asc(Mid$(Y7, RKFe5, 1)) Xor
LjPvXw8 ' Decryptor's part
cZS7 = cZS7 + Chr$(cxJJIVJ3) ' Decryptor's
part
kqNCQI5XUE6 = 5 YOCk3FY6 = qekoP8 + qRdlho3
For evsGCm1iMOuB6 = 5 To 30 Step 3
nGyydTy0howSOOO7 = 2
```