

# VIRUS ANALYSIS 1

## Shelling Out

Peter Szor

Data Fellows Ltd, Finland

Shell.10634 was found in the wild in June 1996 in USA, UK, Australia, Norway and New Zealand. Like another, related, virus [see *'Touching the Tentacle'*, VB, September 1996, p.11], it was distributed via the Internet several times. One such incident occurred on 3 August 1996, when an infected screen saver called PCTRSHOW.ZIP was posted to alt.sex.pictures and alt.binaries.pictures. (Note that there are also clean copies of PCTRSHOW in circulation.)

There are many similarities between the two viruses: in fact, it is probable that the same person wrote both viruses. Shell has a different infection technique than Tentacle, making Shell an interesting specimen. A non-resident direct action infector, it infects only *Windows 3.x* EXE and SCR (screen saver) programs which are of New Executable (NE) format.

There are as yet very few *Windows* viruses, but each example thus far has changed the Entry Point field in the NE header to take control: Tentacle did this. With Shell, the Entry Point address does not point to the start of the virus code, but to the original Entry Point of the host program. The virus' most interesting feature is that, rather than changing this address, it patches the host program's Segment Relocation Records to pass control to the virus code.

### Infection

When an infected program is run, the virus takes control. It then searches for *Windows* EXE programs in the current directory. The virus infects any uninfected NE file, then goes to the directories C:\WIN, C:\WINDOWS, C:\WIN31, C:\WIN311, and C:\WIN95, attempting to infect one file in each directory except C:\WINDOWS. Here, the virus infects two *Windows* programs with EXE extensions. Next, it infects one file with a .SCR extension in the current directory. All these directory names are encrypted in the virus body, thus are not visible by viewing the infected file.

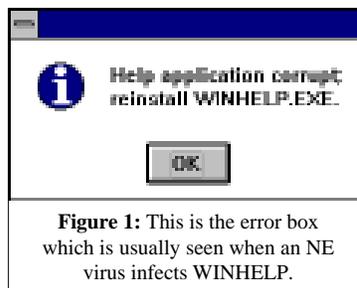
Before the virus infects a file, it checks for the read-only attribute, which it clears if set. Then it opens the victim and reads the EXE header: it does not infect the file if the MaxMem field (word at offset 0Ch in the EXE header) is not FFFFh.

Next, it searches for the host program's NE header. If found, the virus creates C:\TENTACLE.\*\*\*, a hidden temporary file – in Tentacle, this text was visible. Shell decrypts this string, then starts to copy the victim's code into the temporary file. While copying, the virus modifies the NE header fields, creates a new Segment Table which describes a new segment (this will hold the virus code), and writes itself to the end of the file. When the infected copy of the original

file is ready in C:\TENTACLE.\*\*\*, the virus copies this file over the original. Finally, it returns the host program's date- and time-stamp back to that of the original, and deletes the temporary file. Infected files will increase in size by 10634 bytes.

### Taking Control

Given that the virus does not change the original Entry Segment IP fields of the NE header, one might well wonder how it receives control on execution. Shell searches the Module Reference Table (MRT) for the strings KERNEL and VBRUN300. If neither is found, the virus terminates infection and deletes the file C:\TENTACLE.\*\*\*.



Otherwise, the virus picks up the Module Number of the found module name and reads the Segment Relocation Records of each Segment. If it found KERNEL in the MRT, it looks for Relocation Record 91

(INITTASK); if VBRUN300 was found in the MRT, it searches for 100 (THUNKMAIN). Both Relocation Records point to standard initialisation code which must called at the beginning of a *Windows* application.

For example, if KEYVIEW.EXE (a small *Windows* program) is not infected, it has a Relocation Entry for KERNEL.91 for its first segment. When this program is infected, the virus patches this record to point to the virus segment VIRUS\_SEGMENT. Thus, the infected file starts as it did before the infection, but when the application calls one of the above initialisation codes, control passes to the virus.

Within the VIRUS\_SEGMENT are three Relocation Records, one of which points to the original initialisation procedure KERNEL.91 or VBRUN300. Thus, the virus can start the host program after itself. This is new in *Windows* infectors, and means that Shell is an anti-heuristic *Windows* virus.

### Targeting WINHELP

When the virus infects WINHELP.EXE, it patches one byte in WINHELP's second segment. A conditional jump (74h) instruction is replaced by a jump short (EBh) instruction. This technique is typical in cracking; I had to investigate for some time and partially reverse-engineer WINHELP.EXE to understand the idea. Finally I deduced that WINHELP has a procedure to calculate a checksum for itself. This is only a partial checksum, as it does not include every byte of the program. This procedure returns with zero when it does not detect changes in the code, and one if it does.

Shell replaces the conditional jump; thus the message box (Figure 1) will not be displayed, and the user will not notice the infection. As described in the article 'Touching the Tentacle' [see previous reference]: 'The most noticeable victim was WINHELP, one of the most-often-used *Windows* applications, which stopped working completely.' The fact is that the virus infected this file correctly, but WINHELP detected the change, displayed a message box for the user, and terminated, just like most programs which have a self-check routine.

It is possible that the author did not previously recognise this problem because only some WINHELP versions have this feature. *Microsoft* added the self-check function only in *Windows 3.1*.

Both Tentacle and Shell are direct action, non-memory-resident viruses, thus the virus writer could be happy to see that his virus spreads whenever a user hits F1. For this reason he did not simply avoid infecting WINHELP, but patched it. Not only WINHELP has the self-check feature – other programs, such as *Microsoft Mail* and *Microsoft Schedule+*, have it as well; thus the user will notice an infection. The problem is that these programs are not as widely used as WINHELP.

### Trigger Routine

When Shell completes infection, it checks the time, and triggers if it is between 01:00 and 01:05. The virus creates the file C:\TENTACLE.GIF with the hidden, system, and read-only attributes set, and starts to manipulate the *Windows* Registry. (TENTACLE.GIF is 7875 bytes long, which accounts for the large size of the virus.) To do this, the virus modifies the Imported Name Table by adding a SHELL entry (SHELL.DLL) during infection.

The VIRUS\_SEGMENT's Segment Relocation records refer to two standard functions, REGQUERYVALUE and REGSETVALUE. First, the trigger calls REGQUERYVALUE to check the \SHELL\OPEN\COMMAND line under the .GIF extension section. That refers to the command executed to view a .GIF file.

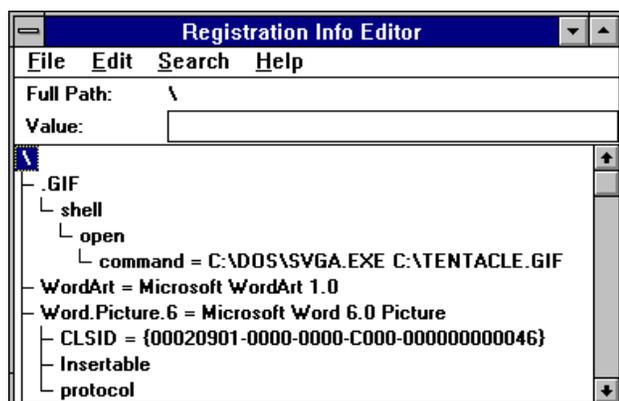


Figure 2: Modification to the registry means that TENTACLE.GIF will be used if a user double-clicks on any GIF file.



Figure 3: This picture will be shown after infection when any GIF file is viewed.

Then the virus replaces the %1 parameter at the end of this line with the line C:\TENTACLE.GIF by using the function REGSETVALUE. For example, the string 'C:\DOS\SVGA %1' will be replaced by 'C:\DOS\SVGA C:\TENTACLE.GIF' (Figure 2).

As a result, the system will show C:\TENTACLE.GIF when any GIF file is clicked (Figure 3). To the user, it looks as though the virus has overwritten all GIF files with its own picture. The virus' second trigger returns the registry to its original configuration from 01:15 till 02:00.

### Conclusion

Like Tentacle, Shell has several bugs, which makes it easier to find. However, this virus is one of the most successful *Windows* viruses, and without carrying out a complete disassembly of this virus, it is simply impossible to disinfect it correctly.

## Shell.10634

Aliases:	Tentacle_II, Tentacle.10634
Type:	Non-resident, direct action infector.
Infection:	<i>Windows 3.x</i> applications in New Executable format.
Self-recognition in Files:	FFFEh in MaxMem field in MZ EXE header.
Hex Pattern:	7BCD 210F 823E 01B4 40B9 6429 BA00 001E 0E1F CD21
Trigger:	Time between 01:00 and 01:05.
Payload:	System displays C:\TENTACLE.GIF image while viewing any GIF file.
Removal:	Delete infected file; replace with known clean copy.