VIRUS ANALYSIS 3

Riotous Assembly

James Beckett

When people involved in the fight against viruses complain that the phenomena are predominantly rather dull, it is most certainly not an invitation to the authors to try harder. Now, however, another 'wish' has been fulfilled, in the emergence of a more advanced virus - Cyber Riot, the first which is capable of infecting the *Windows* Kernel.

Knowledge of *Windows'* internals is clearly becoming more widespread - Cyber Riot uses several *Windows* functions not documented in any of the Developers' Kits. Virus writers have once again found the knowledge they require, whether from published books such as Pietrek's *Windows Internals* and Shulman's *Undocumented Windows*, or through reverse-engineering *Windows* code. The fact that such information is not available from *Microsoft's* documentation makes the entire disassembly process doubly painful.

Previous *Windows* viruses have operated fairly simply - WinVir_14 was a non-resident one-shot virus, whilst Twitch searched systematically through the disk for targets. In DOS, these methods of infection have met with only limited success, so most DOS viruses intercept file or disk accesses to infect files 'on demand'. Cyber Riot is the first *Windows*-specific virus to remain resident and to intercept the execute function by infecting KRNL386.EXE - this is equivalent to infecting the DOS hidden system files (eg. IO.SYS etc.).

Windows System Files

The *Windows* system is based largely on special EXE files kept in the SYSTEM directory - KRNL386.EXE, USER.EXE, and GDI.EXE. Functions called by *Windows* applications are dynamically linked to these files at run time: for example, GDI.EXE contains functions for the Graphical Device Interface (basic line drawing, window operations, clipping, palettes and so on). USER has higher-level functions for Dialog boxes, Cursors, Icons, etc. The KERNEL module (from KRNL286.EXE or KRNL386.EXE) provides for fundamentals like task switching, memory allocation and event handling. All these functions combine to make the *Windows* 'set-up'.

One particular function of the Kernel module KRNL386.EXE, WinExec, is used for starting up new applications. This is typically called by Program Manager or File Manager when an icon or name is double-clicked, or when the Run command in the File menu is used. In DOS, viruses normally intercept Int 21h, subfunction 4B00h (the DOS Load_and_Execute command). The comparable *Windows* function is WinExec: this cannot be intercepted by an active application. Therefore, the authors have had to find an alternative method of subverting it.

Infection Procedure

When an application infected with Cyber Riot is run, the virus searches for the file from which the KERNEL module came, using the *Windows* function designed for that purpose. It opens the file, checks that it is a segmented executable and not already infected, by looking for a checksum value which corresponds to the text 'LROY' (the virus' infection marker). Before proceeding, it attempts to back the file up by changing the EXE extension to EXF.

Infecting *Windows* executables is a complex task - a subset of operations performed by the link stage in a program compilation. This is more difficult than the simple appending one can do to a DOS COM file and the minimal fixup required to a DOS EXE file. All the dynamic-linking which makes up the *Windows* API requires that a vast amount of information be held in the header of a program file, in order to control how it loads. When infecting a file, this must be analysed, copied and modified so that the resulting file still works as intended, albeit with the new virus code attached.

"On return from the MessageBox function the virus starts on a wave of destruction through the fixed disks"

Some 800 lines of code have been written to this end, enabling infection of both standard executables and the kernel file, which is structured in a different manner to other *Windows* files and needs a different strategy. The basic operation is to add an entry to the segment table (roughly speaking, the table of contents for the file) for its own single 3272-byte segment, then adjust the rest of the file to accommodate itself.

Functions which can be called from other programs must be declared so that *Windows* executables can link to them - this is done by entries in the header. The virus patches this information so that the WinExec function points to its own code. The address of the original entry point is kept, so the virus can call it. When this process is complete, the virus immediately jumps back to start the host application as if nothing had happened. This is new in comparison with older *Windows* viruses, which were not capable of passing control back to the host file - it was necessary to issue the execute command a second time. It is this property which represents a significant advance on the part of the virus authors.

Nothing more happens until *Windows* is exited and restarted - modifications have been made to the file on disk, but not loaded into memory. Once loaded, the new kernel module uses the virus code to subvert the WinExec function.

MessageBox

When a new program is executed, the ersatz WinExec first calls the original *Windows* handler, so that the program starts immediately. The virus then considers the file for infection, just as it did the Kernel file. However, after infection, a trigger routine may activate. The virus has already made a dynamic link to a USER module function for displaying a message box, and now checks the date. On certain dates, it displays a message in a window using the MessageBox function. All bear the title:

```
Chicago 7: Cyber riot
```

A different message is printed in the box according to the date of the trigger - from April 29th to May 1st:

```
Happy anniversary, Los Angeles
Anarchists of the world, unite!
```

On any Friday before the 13th of a month:

```
When the levee breaks, I have no place to stay...
(Crying won't help you. Praying won't do you
no good)
```

And on any Saturday in March 1994:

```
Save the whale, harpoon a fat cat.
```

Harmless enough, perhaps, but pressing OK might not be a good idea: on return from the MessageBox function the virus starts on a wave of destruction through the fixed disks, writing part of the virus code over the first sector of each of tracks 1 to 255, heads 0 and 1. It omits the Master Boot Sector and DOS Boot Sector, but many files will be at least partly corrupted - one sector of corruption can be disastrous. One wonders why the authors conscientiously back up each infected program when such a routine is incorporated.

Chicago 7

The virus contains a number of text messages scattered throughout the code. They are not encrypted, nor has any attempt been made to hide them. Some are the text for the messages mentioned above, but several are never printed.

The first of these seems to make a rather contradictory claim of the date of writing, mentioning both January 1993 and summer of the same year. Another hints at more to come in the same line, with an askance poke at the soft drink industry's advertising: 'Coming soon: Diet Riot. Same great aftertaste, fewer bytes.' If this really is targeted at compression algorithms, anti-virus software companies will have to think carefully about their compressed-file scanning.

Yet another message offers the source code, for $15,000,000, but probably nobody will take advantage of the authors' kind offer. The file position of another suggests it might have been meant as part of the second MessageBox, though the text, 'Convict the pigs', is unrelated. There is also a complaint, which could be a genuine grouch, or a red herring: 'Why does *IBM* need to lay me off? Oh well, their loss.'

Finally, a cryptic comment, accusing anti-virus product vendors of making money out of hype and user confusion: 'McAfee's FUD equation: !!!!!!+?????? = $$$$$$'.

Whether copyright infringement could become an issue with regard to virus disassembly by researchers has yet to be seen, but many viruses contain messages claiming ownership. In this case, there is almost a biography: 'This program was written in the cities of Hamburg, Chicago, Seattle and Berkeley. Copyright (C) 1993 Klash/Skism/George J/ Phalcon/Henry Buscombe and 2 ex-Softies, collectively known as the Chicago 7'.

Skism and Phalcon are well-known names, creators of PS-MPC (Phalcon/Skism Mass-Produced Code generator), but Chicago 7 seems to be a new alliance, promising many amusing days to come.

Summary

As part of the kernel, the virus is not readily detectable in memory, but a checksummer should detect the changes made to the infected files. In order to be precise about detection specifications, the structure of the executable file and ways of tracking down entry points need discussion, but this would require much greater depth than time and space allow here. Simple patterns suffice for a basic scanner, but any sensible system would locate the area required in the file before accepting this string as significant.

Cyber Riot is limited to *Windows* systems and cannot, as such, propagate under DOS. Unfortunately, this does not mean that spread of the virus will be restricted, as many people now use only *Windows*, finding a DOS command-line interface problematic. The virus may yet get somewhere, if people do not notice the extra time the hourglass is on their screen. As with all viruses, it is essential to check out any anomalies in your operating system; only thus is it possible to limit their propagation.

Cyber Riot

Aliases:	None known.
Type:	Parasitic file infector.
Infection:	*Windows* Kernel programs.
Self-recognition in Files:	String 'LROY' in EXE checksum.
Self-recognition in Memory:	Not applicable.
Hex Pattern:	offset 013Ah from end of start segment.

```
B40D CD21 0E07 8B5E F8B9 8000
518A D1B9 FF00 518A E9B8 0302
```

Intercepts:	*Windows* Kernel WinExec function.
Trigger:	Displays message on various dates.
Removal:	Delete infected EXE files, and rename corresponding EXF to EXE. Reload KRNL.EXE files from *Windows* disk.