

TECHNICAL FEATURE

Moving To Windows 2000

Péter Ször
SARC, USA

At the 1999 *Virus Bulletin* conference Darren Kessner explained some of the new features of *Windows 2000* that could potentially be used for malicious purposes by virus writers. Some of the new features, such as the *Intellimirror* or *Microsoft Installer*, are still waiting to be discovered by virus writers in the very near future.

Although the new features make it easy to create new virus types and spreading mechanisms, some of the old 32-bit *Windows* viruses failed to work on the release version of *Windows 2000*. Many US corporations are planing to upgrade to *Windows 2000* during September. Moving to *Windows 2000* does make for a few clear advantages. This is, of course, a very costly procedure, and most corporations would do well to wait carefully until the first Service Packs are available.

Windows 95 viruses

Most *Windows 95* viruses will fail to work on *Windows 2000* altogether. About 50% of all 32-bit *Windows* viruses are classified 'Win95', meaning that they only work properly on *Windows 95/98*. Most of the *Win95* viruses that have the potential to spread were developed with the use of VxD functions. Since the VxD driver model is not supported under *Windows 2000* (or under *Windows NT*) a whopping 50% of all 32-bit *Windows* viruses will not effect someone's *Windows NT/2000* system. A workstation upgrade from *Win9x* to *Win2K* would mean the end of such viruses as CIH.

Win32 viruses

Half of all 32-bit *Windows* viruses are actually classified as 'Win32'. These viruses are able to replicate under at least two major Win32 systems. *Windows 2000* is an *NT*-based system with a number of major enhancements. Thus, someone might believe that all viruses that worked under *Windows NT* would still work under *Windows 2000*.

Another common misconception is that *Windows 2000* is so special that virus writers need to write completely new viruses to support it. There were announcements from several anti-virus vendors related to the W2K/Installer virus. The virus was quickly labelled as 'the only virus able to work under *Windows 2000*'.

This is why it was interesting to see if there were any Win32 viruses that did not work on the new *Windows 2000* release version. It turns out that a significant 25% of all Win32 viruses (half of all 32-bit *Windows* viruses) cannot

work on the release version of *Windows 2000*. This is due to minor incompatibility problems that appear in those viruses which were created in assembly.

KERNEL32.DLL Base Address

Several Win32 viruses search in the process address space at various locations for the loaded *KERNEL32.DLL*. Many Win32 viruses do not search the complete process address space but check the 'MZ', 'PE' sequence at the known *KERNEL32.DLL* base addresses.

The common location of the *KERNEL32.DLL* is at 0x77F0000 under *Windows NT*. *Windows 95* and *Windows 98* use a higher address since the DLL needs to be loaded to the shared memory address space. That is 0xBFF70000 for both versions. Some of the viruses check for the loaded *KERNEL32.DLL* at that address in order to identify the addresses of all APIs they need to call.

Early *Windows 2000* betas (*Windows NT 5.0* at the time) used very different *KERNEL32.DLL* base addresses for almost every release. For instance, the beta 1 release used the address 0x77EF0000; that was changed to 0x77ED0000 in beta 3. The RC2 version used the 0x77E80000 address. Not surprisingly, many viruses only work on the RC2 version of *Windows 2000* and fail to work on the release version. This is because the final version specified the address as 0x77E00000.

Viruses that check for the loaded DLL at one wrong location will fail to work. For example, W32/Cabanas does not pay attention to the moving DLL base address and fails to use that method. However, Cabanas uses more than one method to get the API addresses. If the actual host application has an import to the APIs *GetModuleHandle()* and *GetProcAddress()*, the virus can replicate to other files.

Viruses that use only one (wrong) method will fail. For instance, the W32/FunLove virus fails under the release version of *Win2K* although the virus writer paid attention to the new base address. The way W32/FunLove checks for the location of 'GetProcAddress()' is based on a string detection. Since the code changes, the virus is unable to locate the *GetProcecAddress()* API. FunLove also fails to use the W32/Bolzano trick of patching the *NTOSKRN.EXE* file because of the SFC (explained later). Almost 25% of all Win32 viruses fail to replicate because of similar problems. Obviously viruses that are created after the release version of *Win2K* are likely to work properly again.

System File Checker

Win2K introduced the System File Checker (SFC) feature. The SFC uses two directories under the *WINNT* folder, 'Driver Cache\I386' and 'SYSTEM32\DLLCACHE'. The

Driver Cache\I386 directory contains a CAB file called DRIVER.CAB. This file comprises an archive of all the *Microsoft* drivers for *Windows 2000* as well as other crucial system components.

The DLLCACHE directory contains other DLLs and executables such as NOTEPAD.EXE or CALC.EXE. The directory might not necessarily mirror all the applications. A limit is specified according to the drive's free disk space during basic installation of *Windows 2000*. For a large disk, however, the DLLCACHE mirrors almost every standard application and DLL.

The WINLOGON process is always loaded on *Windows NT/2000*. WINLOGON was selected to contain the SFC extension. When WINLOGON starts, it registers a directory 'change notification request' callback function of its own to the system, so that whenever the contents of certain directories change, WINLOGON receives notification. Then WINLOGON looks for the change.

It seems the SFC of the *Win2K* release version uses a catalogue of cryptographic signatures of all system files. In the case of a hash difference, the SFC will use either the DLLCACHE or the DRIVER.CAB file to replace the modified file silently.

WINLOGON will only generate message boxes if an original copy of the file is not available under the SFC directories. If so, a message box pops up asking for a CD that contains the installed *Windows 2000* version. The changed files are copied from the CD and overwrite the replaced files automatically. The SFC compares the file contents in other ways. The version information seems to be a key element.

Closely related system component versions might be replaceable by presenting a new copy of the driver or executable in the SFC directory first. A newer version can overwrite the existing copy. Clearly, the SFC's primary purpose is not virus protection. However, it can be used even from the command line and it generates 'information logs' that can be viewed with the Event Viewer.

Some viruses will fail to work completely because of *Windows 2000*'s SFC feature. For instance, the W32/Kriz virus tries to create an infected copy of KERNEL32.DLL first in the SYSTEM32 directory. During boot time that newly created file would replace the old one. However, the SFC will not let the modification remain. When the machine boots, the W32/Kriz-infected DLL file is gone and KERNEL32.DLL is not replaced. This is because the SFC pays special attention to kernel components and checks their integrity prior to anything else.

Actually, the SFC *does* let the modification happen. It does not try to prevent the modification in any way. It tries to replace the modified file with the old copy. This is a special way of protecting against installation software that replaces certain DLLs or EXE or SYS files with a different, incompatible version. This problem is known as 'DLL hell'. The

SFC was not developed against viruses, and virus writers might find ways to fool it via different methods. However, the SFC has a certain benefit against the majority of Win32 viruses that modify files in the WINNT folder whenever they have the necessary rights to do so. If the SFC-related components are protected with standard system security such as file protection against writes, only Kernel mode *Windows 2000* viruses could challenge the SFC.

It is difficult to notice this automatic backup system. When a Win32 virus such as W32/MIX is executed under *Windows 2000* a very noticeable disk activity will follow the execution of the virus. This is because the virus infects new executables and the SFC starts to get the modification notices and tries to replace the files with original copies. As long as all the copies of the original EXEs are available, *Windows 2000* will be able to work silently without displaying a single warning about the fact that the SFC was used (zero user administration).

If the task list is checked during this time, WINLOGON shows very high processor usage. This is due to the fact that WINLOGON copies the files from the mirror directories one after another. This means copying megabytes of data on the disk. It is noticeable, although it would be better to have an optional warning message, say after the first 10 modified executables were changed!

It is strongly recommended that System Administrators to look into the Event logs very frequently. Personally, I believe that it would be better choice to put this feature under the 'Security Log' instead. The standard 'Information' message might not appear to be important but it could be a sign of virus infection. Since non-standard applications are not involved with the SFC, those executables that do not have cryptographic signatures are not protected in any way.

Some of the newer viruses around such as W2K/Installer, W32/Dengue and W32/CTX do not infect the executables if they are found to be SFC-protected. This is because virus writers can use the SfcIsFileProtected() API exported by SFC.DLL to check if a particular file is protected and thus avoid infecting it.

The SFC does not stop the spreading of certain viruses and it is not a virus security feature. However, it makes the spreading of regular PE file viruses less obvious. Virus writers will certainly try to switch off this feature of the system one way or another. In any case, it is strongly recommended to use the SFC regularly.

Conclusion

Obviously, real HLL-written Win32 applications will work perfectly well on *Windows 2000* without any major problems. Thus, *Win2K* will not prevent the W32/ExploreZip worm kind of episode. Moreover, *Win2K* supports VBS by default and therefore some of the newer viruses might 'appreciate' the *Win2K* environment more than *Windows 95* or *Windows NT*.