VIRUS ANALYSIS 2

Drill Seeker

Péter Ször
SARC, USA

At the end of last year we saw several variants of the W95/Drill virus. It uses the Win32 API and runs in user mode but only works under *Windows 9x*-based systems. However, the polymorphic engine of the virus, called TUAREG, sets it apart from the average 32-bit polymorphic virus. Drill is packed with functionality like per-process residency, a retro-mechanism and an activation routine in a huge 14–18 KB assembly-written virus body, depending on the variant. It implements anti-emulation as well as anti-heuristic features.

Initialisation

W95/Drill is executed via the main entry point of an infected Portable Executable application. First the virus decrypts itself. It is encrypted with two layers of polymorphic code. The first decryptor is very long (several KBs) and placed in the original code section of the application named '.text'. The second decryptor is in the last section at the start of the virus body. This one is short but also polymorphic. Basically, the TUAREG polymorphic engine supports two different polymorphic decryptor generators.

Eventually, the virus is decrypted, but in some cases several million instructions need to be executed. This makes the use of code emulation more difficult. Initially, W95/Drill gets the addresses of all the KERNEL32.DLL APIs it needs to use later on. The list is impressive, considering that there are 34 of them (such as GetProcAddress(), CreateFileA(), CreateProcessA(), FindFirstFileA(), FindNextFileA(), etc). Their names do not appear in the decrypted virus body because the virus uses only checksums of the APIs called. This routine is protected with Structured Exception Handling. If an exception should occur, the virus simply executes its host application. After this, the virus calls its direct action infection routine.

Direct Action Infection

W95/Drill checks if the SFC.DLL (System File Checker) library can be loaded. If it is available the virus gets the address of the SfcIsFileProteced() function. This is because *Windows 98's* second edition supports the SFC just like *Windows 2000*. The virus tries to avoid infecting files that are protected with SFC, a mechanism we see in viruses that try to spread on *Windows 2000*.

The virus loads the IMAGEHLP.DLL (if available) to get access to its CheckSumMappedFile() API in order to be able to recalculate the checksum of an infected file and place it into its header properly. Drill is a retro virus. Before any attempt to infect a file in a directory, it looks for and deletes the checksum files of various anti virus software such as AVP.CRC, ANTI-VIR.DAT, CHKLIST.MS and IVB.NTZ. That happens even if the files are read-only since the virus changes the attributes of the files.

Then it looks for files with .EXE, .SRC and .CPL extensions. However, it does not infect every file it could. W95/Drill uses a random infection algorithm. It will skip some of the files without any attempt to infect it. However, in other cases, the file infection routine is called. The same directory infection routine will be called for the current, *Windows* and *Windows* System directories respectively.

Infection of Portable Executable Files

The infection routine is rather complex. First, the virus checks the name of the file. If it is a known anti virus file the virus will not infect it. Anything that starts with 'tb', 'cs', 'f-', 'pa', 'dr', 'no' or contains the letter 'v' will not get infected. Next, the virus checks if the file is protected by the SFC and skips the infection completely if it is. Otherwise, it zeros the attributes of the file in order to be able to infect read-only files. After this, the virus checks if the file is indeed a PE application.

Drill then starts to traverse the section headers. It checks if the file has a '.text' (code section), a '.bss' (global data section) or '.reloc' (relocation section) and saves their offset for later use. If the file does not have a section named '.text' Drill will not infect. Thus, Drill will not infect a Borland-compiled application that has a code section with the name 'CODE'.

If the file does not have a section named '.reloc', the virus will check if the last section is a '.rsrc' (resource) section. If the last section is 'reloc', then the virus will turn off the relocations and rename the last section with a random name. The name is either five random letters starting with '.' or a section with the name '.?text' where ? could be any character of the alphabet. If the last section is not '.rsrc' and the file does not have a relocation, the virus will not try to infect. This way, Drill avoids possible double infections.

Otherwise, Drill will create a new section in the section table using the above algorithm. The characteristics of the section will include the flags MEM_EXECUTE and MEM_WRITE. The virtual size as well as the physical size of the last section is set to 0x8000 (32768) bytes – rather large, but the virus needs to save the original content of '.text' section that will be overwritten by the first polymorphic decryptor.

The first two versions of the virus only used 0x6000 as the physical size of the last section. Regardless of size the virus might not make the file bigger.

The virus also checks if the code section is long enough and compares that to 13,988 in the latest variant (1.2). This is because the first polymorphic decryptor will be rather long and placed into the code section of the application. Obviously, this is an anti-heuristic infection. The main entry point of the host will be changed to point to the start of the code section. Then, a couple of polymorphic engine functions are called and finally the infected file is written back to disk with the original file date stamp and file attributes. The checksum field of the PE header will be recalculated with the CheckSumMappedFile() API.

Polymorphic Engine and Anti-emulation Tricks

Drill's polymorphic engine is complicated. It has the support for two different polymorphic decryptors. Both layers support XOR and SUB encryption methods – the virus will select a combination of them. The first polymorphic decryptor is several kilobytes long. It has the support for 'do nothing' loops as well as random memory writes.

Interestingly, the virus pays attention to the '.bss' sections. If there is a global data section the virus will generate writes to that area. Some emulators might not be able to handle the situation properly since the physical size of the '.bss' section is typically set to zero. This section is at least page-size, however, and can be written, though some applications might not appreciate the changes.

Before the first decryptor is built Drill uses an interesting function. It checks a list of 32 APIs in the import address table of the host. More exactly, this table has 28 active API names. The virus uses a CRC calculation of API names here. One API CRC is set to -1 in the table, i.e. it is not active. Three other CRCs will not resolve to any known KERNEL32.DLL APIs of any *Windows 9x* release including *Windows ME*. The remaining set of APIs is:

> GetCommandLineA(), GetStartupInfoA(),
> GetEnvironmentStrings(), GetVersion(),
> GetModuleFileNameA(), MulDiv(), GetACP(), GetOEMCP(),
> GetCPInfo(), GetStdHandle(), GetLastError(), GetLocalTime(),
> GetSystemInfo(), GetCurrentProcess(), GetCurrentThread(),
> GetConsoleCP(), GetCurrentDirectoryA(),
> GetWindowsDirectoryA(), GetSystemDirectoryA(),
> GetDriveTypeA(), GetComputerNameA(), IsBadWritePtr(),
> GetTickCount(), IsBadReadPtr(), IsBadCodePtr(),
> LocalHandle(), LocalSize(), LocalFlags()

All of these are KERNEL32.DLL APIs. If there is an export to any of them in the host application's import address table, Drill will save a reference. The polymorphic engine will use these references later on. The virus will be able to make a call to any of these available imported functions. The polymorphic engine will support the proper number of parameters on the stack to make the call possible. After that, the virus will place code to check the proper or improper return values returned by the actual function called. This way it forces a 'proper' environment and thus this function is implemented against emulators. First-generation 32-bit emulators might not be able to emulate even a subset of the Win32 APIs. However, several emulators that I know have the ability to be extended with any APIs that a polymorphic virus uses to challenge emulators. This was predicted by several AV researchers. W95/Drill is the first virus to do this with part of the TUAREG engine v1.2. The list of APIs could be changed in the virus forcing the emulation of a different subset. This makes detection of the virus more difficult.

Activation Routine

After infecting directories, Drill checks the system date. If it is a Friday before the 8th of any month or between the 14th and 22nd of any month the virus will activate. Drill loads the ADVAPI32.DLL and gets the addresses of five registry APIs and changes the start page of *Internet Explorer* and *Netscape* to www.thehungersite.com. After checking the activation routine, the virus hooks the import address table of its host application to become per-process resident, a technique first used by W32/Cabanas .

Per-process Residency

Drill hooks GetProcAddress() in order to return the original API addresses to the host. It will also hook a set of APIs and call its infection function from them. This way, every time there is an access to a PE file by the host application the virus will try to infect. Finally, the virus is loaded and executes its host application. The large part that is overwritten in the '.text' section is written back from the end of virus body to its place. The virus does not handle cases where the import address table is placed in the '.text' section. Apparently, some *Microsoft* applications are compiled this way and as a result, infected files like that will crash since the System Loader will patch the actual polymorphic decryptor of the virus.

Conclusion

Several AV products did not detect Drill even at the end of December 2000. Some researchers were taken by surprise, others need to take the time to go through the dirty details of the virus. I hope the details here will help them implement appropriate detection. The detection of such viruses is rather difficult. Their repair is state of the art.

W95/Drill

Aliases:	Tuareg, Mental.
Type:	Win95 PE appender.
Self-recognition in files:	The virus checks if the last section is not named '.reloc' or '.rsrc'.
Payload:	Changes the start page of *Explorer* and *Netscape* to www.thehunger.com/.
Removal:	Replace infected files from clean backups.