

VIRUS ANALYSIS 1

Magisterium Abraxas

Peter Ferrie
SARC, Australia

W32/Magistr.24876@mm is a polymorphically encrypted, entry point-obscuring, anti-heuristic, anti-debugging, memory resident, parasitic infector of Portable Executable .EXE and .SCR files. It can replicate across local area networks, and it has mass-mailing capabilities (using its own SMTP engine), some highly destructive payloads, an interesting visual effect and a number of bugs.

Initialisation

As an anti-heuristic device, files infected with W32/Magistr do not have their entry point altered. Instead, the virus will save the first 512 bytes of code, and replace them with polymorphic garbage which includes subroutines, jumps, and some Structured Exception Handling tricks to interfere with debuggers and code emulators. Eventually, an indirect call, the address of which is stored by the virus in the Import Table of the host application, will transfer control to the section that contains the virus body.

The virus body is decrypted by XORing it with a single shifting 32-bit key, however the decryptor is also polymorphic, of variable size, and contains another Structured Exception Handler trick. Fortunately for the AV people, there is a characteristic of the decryptor which allows the encrypted body to be located quickly and accurately.

Once the virus is decrypted, it will attempt to find the KERNEL32.DLL base address by taking the return address from the stack and searching the previous 1 MB of memory for the MZ header whose export table DLL name ends with the string 'EL32'. If the address cannot be found using that algorithm, then the virus will use one of two default values, based on the value of the high eight bits of the CS selector.

Using the KERNEL32.DLL base address, the virus will retrieve the addresses of 42 APIs it requires for system integration and replication on the local machine, the names of which are stored as checksums instead of strings. The checksum routine is a CRC algorithm using 16-bit registers that has been blindly copied into a number of recent 32-bit *Windows* viruses. It seems likely that not one virus author understands the algorithm well enough to produce a 32-bit version. A bug exists in the import parsing code which will cause a crash if an import cannot be found.

At this point is included a large chunk of code copied from the W32/Dengue virus. This process was introduced in *Windows NT*, and is always running. The residency code begins by converting the computer name to an encrypted string and creating a memory-mapped file using this name.

The memory-mapped file is part of the mechanism that the virus uses to remain memory resident. Then, one of two routines is executed, based on the *Windows* platform (*9x/ME* or *NT/2000*), to search for the EXPLORER.EXE process in memory. Perhaps the most embarrassing bug in the virus exists here, in such a simple function as string comparison: it will return a match even if the last character differs in the strings. Once *Explorer* has been found, a 110 bytes routine is injected into a writeable section, and the TranslateMessage() API from USER32.DLL is hooked to point to this routine. After this, the original host bytes are restored and the host is executed.

You've Got Mail

The injected routine gains control whenever *Explorer* calls TranslateMessage(). This function is part of the message loop in all GUI applications, so it is called frequently. When the routine is reached for the first time, a thread is created and the function is unhooked. The thread will wait for three minutes before performing any actions.

After the time has elapsed, the thread will retrieve the location of the *Windows* directory, the Program Files directory from the Registry, and the Program Files drive. Depending on the first character of the computer name, the virus will choose one of those locations in which to create its data file. This data file will contain the date of initial infection, and the full path and number of 'interesting' files, namely those files which contain email addresses: the *Windows* Address Books (*.WAB), *Outlook Message* stores (*.DBX, *.MBX), and the *Netscape Messenger* mail files.

The thread will also retrieve the user name and email address of the current user. These names are taken from the *Outlook Express*, *Internet Mail and News*, and *Netscape Messenger* Registry hives. The virus keeps within its body the email address of the ten most recently infected users. If the current user's email address is not already in this list, then it will be placed at the top of the list, and the other nine entries will be moved down. Then the search begins for the interesting files in the Program Files directory and the *Windows* directory.

After a one-minute wait the virus will check if an active Internet connection exists. If it does, the virus will search the Program Files drive for .DOC and .TXT files and choose from one of these files up to four words for the email subject and between 20 and 85 words for the email body. Additional code adds a period to the end of the email body and capitalises the first word, if required. Having formed the mail text, the virus will create the email headers, addressing the mail to up to 100 recipients, but explicitly avoiding the current user, and with 80% chance it will alter the second character of the return email address. This has

the effect of preventing people from replying to the email, in order to alert the user to the infection. The X-Mailer string is always 'Microsoft Outlook Express', but the version is chosen randomly from a table containing five version strings.

The virus will then attempt to locate a file to send. The choice is made by examining the first 20 Portable Executable .EXE or .SCR files that are smaller than 128 KB. If no such file is found, then an empty email will be sent. Otherwise, one of those files will be infected and attached to the mail. The mail Content-Type will be set randomly to 'image/gif' or 'application/octet-stream'. There is a 20% chance that the file from which the subject and body text were taken will also be attached to the email. The virus now sends the email and disconnects.

See Spot Run

There is a 25% chance the virus will search the Program Files drive for the first 20 Portable Executable .EXE or .SCR files, choose one, make a copy of that file, decrement the fifth last character of the filename, and infect that copy. If the *Windows* directory is not found, then up to 20 Portable Executable .EXE or .SCR files will be infected. The other 75% of the time, one Portable Executable .EXE or .SCR will be copied, the fifth last character of the filename will be decremented, the copy will be infected, and the Run key in the Registry will be altered to include a reference to the copy. The name of the Run value will be the filename without the suffix. This forces *Windows* to run the infected file whenever *Windows* is started.

After another one-minute wait, the virus will search each local hard drive for the first 20 Portable Executable .EXE and .SCR files and infect all of them. If the *Windows* directory is located on a drive that is not the current one, then the 'run=' logic will be executed for that drive. This code is also applied to every shared directory that is visible to this machine on the entire local area network.

Infection

Magistr infects Portable Executable files that are not DLLs, and are smaller than 1 GB. The infection marker is one of two values (0xCECD, or the first two characters of the computer name ORed with 0x9183), in one of three locations (NumberOfSymbols field in the PE header, or PointerToLineNumbers or NumberOfLineNumbers field in the first section header).

The first 512 bytes at the host entry point will be saved and replaced by polymorphic garbage, however this routine contains bugs that can produce either non-working code, or code longer than 512 bytes. A new polymorphic decryptor will be generated and the virus body will be encrypted. If a file contains a relocation section that is large enough to hold the decryptor and virus body, then the relocation section will be overwritten and the section name will be the first four characters of the computer name, preceded by a

period. Otherwise, the virus will append itself to the last section in the file.

Seek and Destroy

Having completed the replication phase, the payload triggers are tested. If the machine has been infected for at least one month, if at least 100 people have been sent emails, and at least three .DOC or .TXT files contain at least three phrases from the list of 55 phrases contained in the virus, then the first payload will activate.

This payload appears to have been adapted from W32/Kriz, though it is functionally equivalent to W95/CIH's. It begins by deleting the last file found by any of the virus search routines. Under *Windows 9x* and *Windows ME*, it will also erase the contents of the CMOS memory and flash BIOS, and overwrite a single sector on the first hard disk. This sector is always cylinder 0, head 32, sector 1. The location is never updated. Under all platforms, it will delete one in every 25 files on every local hard drive and shared network directory, and overwrite every other file with the text 'YOUARESHIT' as many times as will fit in the file.

After waiting less than a second, the entire first payload is repeated. This loop occurs infinitely. The second payload occurs after the machine has been infected for at least two months. On odd days, the desktop icons will be repositioned whenever the mouse pointer approaches. Given the nature of the rest of the code, it is likely that this routine is copied from another source. The third payload occurs after the machine has been infected for at least three months. Each time the injected routine is executed, this payload will delete the last file found by any of the virus search routines. Then, after every four minutes, the payload triggers are tested again.

W32/Magistr is certainly a complex virus, but presents nothing really new in virus writing. However, the virus is in the wild and could possibly become as widespread as W32/Funlove, and as damaging as W95/CIH.

W32/Magistr	
Aliases:	I-Worm/Magistr, PE_MAGISTR.A, W32/Magistr@mm.
Type:	Polymorphic, EPO, memory resident, parasitic mass-mailer.
Infects:	PE .EXE and .SCR files.
Self-recognition:	Magic value in PE header of files, memory-mapped file in memory.
Possible Payload:	File deletion, flash BIOS erased, message box, moving icons.
Removal:	Delete infected files and restore from backups.