VIRUS ANALYSIS 1

KAOS4: A Sexually Transmitted Virus?

The KAOS4 virus gained notoriety through its posting to the *Internet* newsgroup alt.binaries.pictures.erotica. Although KAOS4 has, as a result of this method of distribution, become widespread, it appears to be a relatively simple, non-resident COM and EXE file infector, designed to avoid detection by heuristic scanners.

A Simple Plague

KAOS4 is a rather primitive virus, which makes no attempt to hide its presence, either during or after execution of a file. As the virus does not become memory-resident, no stealth routines are included, and, excepting encryption of some text strings stored in the virus code, disassembly proved to be trivial. It will be stopped by any behaviour blocker, and any of the popular checksumming programs should be able to detect its presence.

Infection and Operation

The virus infects COM files by appending its code to the host file. When such a file is run, the virus receives control after execution of the starting JMP instruction, and some effort is made to restore the program's original registers before processing continues. No attempt is made to armour the code against disassembly, and the entire virus was pulled apart in a matter of hours.

The virus then sets up its own Disk Transfer Area and decrypts three text strings using a NOT instruction (the decrypted strings are *.COM *.EXE and PATH=). The purpose of this is to avoid detection by scanners which utilise heuristic detection techniques.

A pointer is set up to the string *.COM, and the infection routine is called. Once this routine has completed, the pointer is reset to point to *.EXE, and the process repeated. No checks are made on returning from the infection routine.

The virus then restores the image of the host file in memory, and returns control to it.

Up the Garden Path

The infection routine contained in KAOS4 is poorly written. It begins by searching the Environment Segment (held in the Program Segment Prefix) for the PATH variable. This is done in such a way that if the environment segment does not contain the character sequence 'PATH=', the code will enter an infinite loop. On all the versions of DOS tested, this string is present even if no path has been set. Although this information will not change while the virus is executing, the search for the path variable is carried out twice every time an infected file is run.

If a match is found, the address of the PATH is stored for later use. The virus then searches for a matching file using the DOS FIND_FIRST function. If no match is found, a routine which attempts to allow the virus to search along the path for an infected file is called. A check is made to ensure that the PATH name has been found, or that all parts of the PATH have been searched. If this is not the case, another search is made for a suitable file by searching along the path set up on the machine.

Self-infection Checks

One of the continual problems encountered when disassembling a virus is to ascertain what the author thought his code would do when he wrote it. This was the case when examining the KAOS4 virus, where the checks made before infecting a file are somewhat bizarre.

Firstly, the seconds field of the time stamp of the file is checked against a mask of xxx111x1. If a match is found, the file is deemed unsuitable for infection. This will cause the virus to reject files which have a seconds stamp matching this pattern (e.g. 58, 62). This appears to be a self-infection check, as infected files have the value 58 in the seconds field of the time stamp.

"in this respect, the virus functions very well indeed: infected files do not raise a single heuristic warning flag with ThunderBYTE"

If the test made on the time stamp is passed, KAOS4 checks whether the internal structure of the file is EXE or COM. This is carried out by checking the first two bytes of the file for the ASCII letters MZ or ZM. The virus is written in such a way that a heuristic scanner will not identify the true functionality of the code. In this respect, the virus functions very well: infected files do not raise a single heuristic warning flag with *ThunderBYTE*; a creditable achievement. For obvious reasons, the precise way in which the virus achieves this is not stated. Suffice it to say that it works, although the effort involved seems to be wasted, given the rather obvious way in which the virus operates.

In the case of COM files, the target file is only infected if it does not begin with the words E9??h ??20h, and if the length of the infected file would be less than 64K. For EXE

files, offsets 18h, 1Ah, and 12h of the EXE header are examined. These are the areas which contain the Relocation Table Offset, the Overlay number, and the Checksum. Given that the virus has already carried out a self-infection check, these further tests seem to be unnecessary. These tests completed, a flag is set to indicate whether the target file is an EXE or a COM file.

The infection routine is standard, and only of note because it does not work correctly. Under certain circumstances, the virus body can become corrupted, allowing subsequent infections to attack only the first COM and EXE file found in each directory on the path. These partial infections do not operate correctly, and can cause the system to hang after they have infected other files.

Conclusions and Thoughts

Due to the simple-minded way in which the virus is written, KAOS4 poses little long-term threat to the user community. Apart from its novel distribution method, the virus seems to be merely an ego trip for its author. According to a text string stored within the virus, this is none other than 'Köhntark', a virus writer who wrote a rambling description of how to avoid heuristically-based scanners. Users should be grateful that he is appallingly bad at his chosen pastime.

Although KAOS4 can usually replicate successfully, working its way down infected directory trees, on machines with a large PATH variable set up, the large amount of disk activity caused by the virus will soon become noticeable. Notwithstanding its rather obvious behaviour, anyone in the UK who has been affected by the virus is urged to call *New Scotland Yard's Computer Crime Unit*, on 0171 230 1177 so that, should the culprit be found, he can be held responsible for the actions of his creation.

KAOS4

Aliases: None known.

Type: Non-resident parasitic file infector.

Infection: COM and EXE files.

Self-recognition in Memory:

None necessary.

Self-recognition in Files:

Checks checksum value in EXE file, or the first four bytes of COM files.

Hex Pattern:

```
8C96 D102 2E89 A6D3 028C C88E
D0BC FFEF 2E8A 86B4 022E 8C86
D502 5006 1E0E 0E07 1FFF B6B0
```

Trigger: None.

Removal: Under clean system conditions, identify and replace infected files.