FEATURE 1

The Virus Analyst Headache

Eugene Kaspersky
Kaspersky Lab

The careful analysis of new viruses appearing every day is the main indication of an anti-virus scanner's quality. Automatic or semi-automatic analysis, adding the detection and disinfection records to the anti-virus databases, may cause low detection rates and/or missed infections.

Then the scanner reports that all the files on the system are disinfected, but on next reboot the virus appears again and reinfects everything. Users are not happy with that, especially if the virus has its 'doomsday', erasing all their data so that the only way to get rid of it is by reformatting destroyed hard drives.

Sometimes, this does happen. Imagine a new variant of DOS virus which infects COM and EXE files, but which also infects SYS drivers. A careless analyst may miss this new feature, and SYS files will stay undetected. Then the scenario described above ensues. This is why it is necessary to pay attention to all virus branches and routines; each of them may have unexpected outbreaks. If an analyst misses this kind of thing, the virus may find its way into the wild.

Careful Analysis – Is it a Big Deal?

Ten years ago Jerusalem and Cascade were the 'scary monsters' which anti-virus experts spent days disassembling and trying to understand. These were the green years of the anti-virus industry – dreamland now – when new viruses appeared once a week. Several hundred new viruses and variants each month – that is the reality on today's virus conveyor belt.

That means that the average virus analyst has to process about ten or more viruses per day, if the lab employs around five virus experts. Do not forget that the same virus experts usually elaborate and support scanning and disinfection engines, and they also want weekends off and holidays.

It is pointless to say 'Hire more virus analysts' – it is very hard to find an experienced virus analyst (without a virus-writing past). Inaccurate processing by an inexperienced analyst will cause false positives and negatives. That will necessitate the hiring of more tech-support people, and even the occasional high-class expert to clear bugs in your company's anti-virus databases.

When a virus analyst's career starts and they analyse their first virus – that is a challenge. The first dozen is an interest. The first hundred is a hobby. The first thousand becomes a routine, a conveyor belt of viruses moving quickly from the Incoming to the Sorted area. Open a new one, disassemble and glance inside it, see that it is a variant of a known virus – this is nothing new. Until the virus conveyor belt stops. It stops not because the Incoming area is empty and there is nothing to do (I dream about that!), but because something complex and new has appeared.

Ancient History

Virus analysts who started out in 1990 may remember two DOS viruses which turned an ordinary day into a nightmare. They were Whale (a variant of Fish#6) and Pogue (based on the MtE – the first strong polymorphic engine).

The 9 KB Whale virus looked like a very complex addition to the DOS kernel – it hooked about 20 DOS functions, and corresponding virus subroutines ran the infection and stealth virus mechanisms. That was enough of a challenge in those days: locating all the hooks and infection routines, getting past anti-debugging tricks. Several days were lost just getting used to working with this kind of virus.

Pogue presented anti-virus scanners with a new generation of polymorphic virus, and as far as I remember, it stayed undetected by any of them for several months. Virus experts had to choose the way to analyse it: either by replicating several thousand samples and using statistical methods, or by analysing the very intricate polymorphic engine's subroutines.

A Whole Lotta Viruses

Imagine you receive a several megabyte archive full of new viruses – about fifteen thousand of them. Fridrik Skulason (*FRISK Software*) was the first to receive such a nightmare package, passing it onto other experts saying 'Let me ruin your day'. It was no big deal to write a generic detection and disinfection routine for all these samples, but it still needed to be tested – they all had to be replicated, and run against detection and disinfection tests. That meant that it was necessary to open and copy a sample, copy 'sacrificial goat' files, run the virus, infect the files, move them to a '\Replicated' directory. The computer had to be rebooted after that to be sure that there were no virus traces left in the system. Try repeating that fifteen thousand times.

That was a huge task. In my case, the virus-replication computer (a *Pentium-130*) was working with no long interrupts for about a week, and several times the hard drive ran out of disk space, overflowing with infected samples.

New Platforms and Formats

It is not easy constantly switching your focus to new types of virus. Boot and DOS parasitic viruses evolved into *Windows* viruses, then macro infectors, and then self-

replicating Java applications, VisualBasic scripts, HTML pages, and so on. Nowadays, viruses occupy all niches in the computer's 'biology'.

At first, viruses infect new popular and modern (at the moment they appear) operating systems (*Windows, OS/2, Linux*). It is necessary to have the right tools to disassemble them, and to be informed about internal executable file formats. That is relatively easy.

Unfortunately, they often have additional features, and to locate virus code, in some cases, it is not sufficient to find out the address of the program's entry routine. The virus code can be linked with other parts of segmented new executables – to file Exports. Win/RedTeam (see *VB*, May 1998, p.6) affects exports in Win16 NE files, Win32/SKA (see p. 6 of this issue) exports in Win32 PE files. It is also unfortunate that *Windows* viruses with quite simple internal file structures run under quite complex environments.

Win32 kernel's internal formats and features are not described in any documentation, but we need them to add detection and disinfection for memory-resident *Windows* viruses, and these formats are different again from good old, well-known DOS MCBs (Memory Control Blocks). Needless to say, it is a good idea to have in mind formats of protect-mode Global, Local and Interrupt Tables – often they help to understand what the virus does.

The number of viruses (including those discovered in the wild) depends on the popularity of the operating system. Fortunately, now we have only one – Win32. Imagine that any other (say, *Linux*) will be also very popular, and incoming *Windows* stuff will be doubled by *Linux* viruses.

The Macro Problem

We have to put up a big flag here with *Microsoft Office* written on it. The internal binary formats of *Office* documents, sheets, presentations and other components are much more complex than *Windows* file formats and disk space allocation tables. To detect and disinfect macro viruses the anti-virus scanners have to support these formats, so anti-virus experts have to be familiar with all of them. These formats are undocumented, and anti-virus labs have to start their own investigations to build this knowledge-base. That is why 'true' detection and disinfection methods were only embedded into anti-virus scanners six months after discovering the first Concept macro virus.

This chapter is not finished. There is more room for macro viruses now and in the future (for instance, VBA is licensed for use in *CorelDraw*).

High-Level-Language Viruses

Up-to-date disassemblers are familiar with most DOS HLL (High Level Language) executable files written in C/C++ or Pascal, and work with such DOS viruses is no more onerous than with average viruses written in assembler. The disassembler detects the compiler which was used to compile the virus, loads necessary libraries database, locates main program's routine and comments all calls to runtime library.

It is not the same for all Win32 compilers. There are several that generate 'black boxes' for the analyst. Delphi and the latest VisualBasic compilers produce easily comprehensible executable files. Imagine an average Delphi program (just 500KB). There is no really good tool to disassemble it, disassemblers just output several megabytes of pure commented code. Sometimes it is quite difficult to separate viruses written in Delphi from non-viral programs. The virus analyst must run it on the test computer, watch its behaviour, and log results. It is not hard to see why this is not the best way.

Terrible Tricks

This last section is dedicated to the special tricks that virus writers add to their creations. The SSR, Zhengxi and Nutcracker viruses, to name a few, are fat, complex, often stealthy and extremely difficult-to-analyse programs. They use many anti-debugging and anti-disassembling tricks like on-the-fly en/decryption, hidden branches – everything virus writers can imagine to make virus analysts frustrated.

The Lexotan, TMC and some other viruses use self-mutating algorithms. That means that the virus is not encrypted, but its whole 'working' code is mixed with junk instructions. The virus changes the sequence of routines and branches, mutating data offsets in its assembler instructions, constants and so on.

The Latest Thing

A new *Windows* virus I received recently turns these tricks against the *Windows* platform. This polymorphic, Win32, memory-resident virus, named Harrier after the text in its body, appeared to have about 100 KB (yes, one hundred kilobytes!) of assembler code.

It stays in memory as part of an infected program, hooks about 30 (that is correct, thirty!) *Windows* functions, manipulates PE files sections and Import tables, and so on. Even after several layers (from 9 through 17) of polymorphic decryption loops have decrypted the virus code step by step, the virus routines do not appear in 'easy-to-analyse' form. All virus instructions (about four thousand lines of assembler code) are randomly mixed in the virus code and linked by JMP opcodes.

Needless to say, it is impossible to analyse the virus in this form, and it is necessary to assemble its disassembler to ordinary readable state. I spent about 10 hours 'compressing' the virus disassembler and used specially developed helpers, but anyway that was a crazy task. This kind of virus is not the thing virus experts dream about, but it happens, and when it does, no amount of pain killers will ease your headache!