ISSN 0956-9979 JULY 2001

VIRUS BULLETIN

THE INTERNATIONAL PUBLICATION ON COMPUTER VIRUS PREVENTION, RECOGNITION AND REMOVAL

IN THIS ISSUE:

• **Elfin malware:** *Linux* binary executable viruses are on the increase and, these days, ELF is a standard file format. Marius van Oers looks at the issues that arise when detecting ELF binary malware. Starting on p.6.

• **Sitting ducks:** Large and small organizations alike must plan and prepare for situations in which they are one of the first targets of malicious code activity. Dan Dumond outlines the basics of corporate AV contingency planning, see p.14.

• **DOS comparative:** As an operating system DOS may well have seen better days, but as a tool platform it remains essential. We put eighteen DOS products through their paces, starting on p.16.

CONTENTS

COMMENT

“ Cyber-cops should work their own solutions out ”

NIPC Blues

On 19 May 2000, in the aftermath of LoveLetter, the US Attorney General and Michael Vatis of the *National Infrastructure Protection Center* (*NIPC*) gave one of the most surreal IT security interviews I have heard. The top *NIPC* cyber investigator’s broad and mildly accurate comments were based, as he put it, on a ‘preliminary analysis [of the virus] overnight’. Newlove is not excessively complex – in fact, I would have thought that anyone familiar with VBS would have been able to analyse it in an hour or so. So much for my optimism: either the *NIPC* lacked the ability to assess the threat properly, or they relied on what they were fed in a hurry by anti-virus vendors. Ouch! The virus wasn’t nearly as widespread nor as deadly as hyped and, in the end, the show seemed a bit pointless. Worse, since many administrators shut down their mail servers ‘just in case’, one could argue that the high profile warning caused more disruption to business operations than the virus itself.

Why did this happen? The *NIPC* obviously suffered from a lack of technical skills and put too much trust in vendors and their misleading estimates. True, law enforcement agencies often rely on the help of external experts for high tech investigations. But the IT security world differs from other fields: there are virtually no independent experts. Worse, while conventional experts often disagree in their interpretations of the facts, computer security experts often disagree about what the facts are in the first place. Unfortunately, so many biases come into play that assessing the real value of anti-virus experts’ advice can be harder than understanding a VBS polymorphic generator…

Since the IT security industry’s business model relies at least partly on FUD, anything that substantiates those fears (and what better than a misinformed governmental press conference to do so?) is, in the short term, a financial blessing for the industry. So, should we care? There are, after all, valid reasons to implement a security policy and customers who do so will ultimately benefit from their purchase, even if they acted with the wrong motive initially.

I think we should care, for a number of reasons. First, remember that if the pharmaceutical industry has been able to build trust and prosper in the long term, it is because independent governmental validations and regulations eradicated its charlatans. At the risk of alienating part of the audience, I claim that the computer security industry could also benefit from such a therapy. Secondly, we frequently see anti-virus companies commenting on what law enforcement agencies should or should not do about virus writers and intentional virus distribution. Yet we know that, in a healthy society, crime and punishment should not be defined by commercial organizations alone for, if that were the case, innovation would be stifled by established businesses who would try to criminalize anything perceived as a threat to their profitability. We need a knowledgeable independent organization to confirm and fine-tune the anti-virus industry’s positions. Anti-virus companies themselves are not legislative bodies nor moral authorities, even if at times they seem to believe they are. Thirdly: just as we haven’t seen a digital Pearl Harbour, we haven’t had a cyber Bay of Pigs yet. I wouldn’t want to be at the wrong end of a misinformed and incompetent governmental investigation. Would you?

Becoming an independent, competent and believable player in the IT security field is no mean feat. As the US General Accounting Office recently put it, even the *NIPC* is ‘significantly challenged’ in this endeavour. To succeed, cyber-cops should forgo their reliance on a biased industry and work their own solutions out. The devil, as usual, is in the details: properly funded and staffed law enforcement agencies should develop and constantly update a deep understanding, a true working knowledge of the IT security problems they tackle.

I wish them luck because I believe that a credible cyber-police may be an essential ingredient of cyber-democracy.

Pierre Vandevenne, DataRescue, Belgium

NEWS

There's a Worm in this Apple…

Another day, another file format. On Friday, 9 June, news began to filter through of an AppleScript worm doing the rounds. Named Mac.Simpsons@mm, the worm uses the scriptable nature of *Outlook Express* or *Entourage* to send itself out. The worm seems to have no malicious payload and does not appear to be widespread.

The worm follows the 'eighteen-month' rule for these particular mailers and, now the cat is out of the bag, we may see more. The fact that the worm needs certain applications installed as standard and Mac Users in general have eclectic installs, combined with the small install base should make this a flash in the pan. As this piece of Mac malware is a worm, even those AV companies who do not have a Mac product will have to sit up and take notice ▮

Patching the Macro Gaps

Microsoft has issued a security bulletin warning users of *Word* that they should update their software with a patch to fix a major hole in the application's security. It has come to light that, in current versions of *Word*, it is possible to modify a document to allow macros to bypass the user's security settings and run automatically – with no warning prompt – on opening the document.

Microsoft's security bulletin comes little more than a week after *Computer Weekly* reported that, speaking at the *Microsoft* Digital Britain Summit, *Microsoft* Chief Executive Steve Ballmer admitted his company should dedicate more time to security, saying, 'Our company has only performed medium in the security department.' ▮

Getting the Message Across

MessageLabs received some unexpected publicity when a contestant on UK reality TV show *Big Brother* appeared sporting a *MessageLabs* logo-emblazened t-shirt. Unfortunately for *ML*, the publicity was short-lived as the British public voted the contestant off the show by an overwhelming majority the following day.

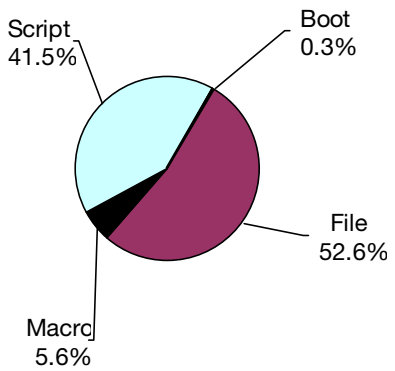
Meanwhile, *Panda Software*'s 'bevy of beauties' attracted attention at the recent NetWorld+Interop 2001. The three 'statuesque beauties' represented some of the viruses to have hit the news recently (Melissa, LoveLetter and Anna Kournikova). More than 500 corporate administrators visited the stand to have their photograph taken with the girls – Donna Wesley Rogers, VP Marketing at *Panda*, was 'gratified that the red-blooded American males had such a positive reaction to our lovely viruses', but pointed out that, in all seriousness, it was only the presence of *Panda AntiVirus* at the booth that enabled the crowd to admire the beauties in safety… which must have been a relief ▮

Prevalence Table – May 2001

Virus	Type	Incidents	Reports
VBSWG	Script	2158	36.1%
Win32/Magistr	File	1131	18.9%
Win32/BleBla	File	731	12.2%
Win32/Hybris	File	579	9.7%
Win32/MTX	File	330	5.5%
Win32/BadTrans	File	152	2.5%
Kak	Script	126	2.1%
Laroux	Macro	113	1.9%
Homepage	Script	71	1.2%
Divi	Macro	66	1.1%
Win32/Navidad	File	60	1.0%
Win32/QAZ	File	42	0.7%
LoveLetter	Script	37	0.6%
Tam	Script	35	0.6%
Marker	Macro	30	0.5%
Win32/Msinit	File	30	0.5%
Win32/Funlove	File	28	0.5%
VCX	Macro	25	0.4%
Haptime	Script	24	0.4%
Ethan	Macro	20	0.3%
Win32/Ska	File	16	0.3%
Tristate	Macro	12	0.2%
Melissa	Macro	10	0.2%
Others [1]		148	2.7%
Total		5974	100%

[1] The Prevalence Table includes a total of 103 reports across 42 further viruses. Readers are reminded that a complete listing is posted at http://www.virusbtn.com/Prevalence/.

Distribution of virus types in reports



LETTERS

Dear Virus Bulletin

The Debate Continues

I guess Richard Ford and I should be gratified to have drawn five letters (last month's VB) in response to the disinfection debate (VB, May 2001, p.14). Given my disputatious reputation, perhaps I should be more pleased that three of the respondents chose to disagree with me. Before addressing the flaws in my disputants' positions, I'd like to comment briefly on the other two responses.

How could I disagree with the view of VB's Technical Editor, Jakub Kaminski, when he effectively summarized my unstated pragmatic view. In the real world, disinfection will continue to be the extensively used, yet grossly flawed crutch of the lazy and ignorant. Of course, Jakub put it much more diplomatically than I would have!

Andreas Marx tantalizingly exposed the tip of what I suspect is a virtual iceberg of worrying test results regarding the true state of disinfection technologies. I'd particularly like to see *AV-Test* (or another competent testing body with the not insignificant resources it would require) extend such tests to include EXE, and especially PE, infectors and a tabulation of detection of monolithic replicators/spreaders once they have been disinfected of parasitic infectors by each scanner.

I'd like to address one specific criticism raised by Atli Gudmundsson of *SARC*. I agree that many of the detection failures I talk of (where Scanner X misses something it 'knows' after Scanner Y has disinfected the target file of some other virus) are due to poor detection processes in the first place. However, as Gudmundsson said, '[we] are in the business of providing solutions and fixing problems, not creating them'. Ignoring the fact that other AV developers do not take similarly 'advanced' approaches as your product does to detecting certain kinds of viruses is the height of arrogance and surely will contribute to the problem, rather than solve it.

Sadly, I can deal with the rest of the points raised by my detractors in one lump (and this covers the previous point as well). You see, my detractors were tricked. They have put pen to paper to explain the market realities of which they seem to believe I am ignorant. In so doing, they all failed to see that, although I never mentioned the word in stating my argument, my position is based on a strong ethical stand that has been held up as the leading light of the AV community. It is unethical for anti-virus developers to create new viruses (or other forms of malware) and release them. Therefore, it is unethical to provide imperfect disinfection, and all my detractors agree that disinfection will often be imperfect (and often in unpredictable ways and places).

I'll leave it as an exercise for the readership to decide the true significance of the fact that staff from three of the largest and most influential anti-virus developers have failed to notice the ethical dimension in all this, but we have unequivocal evidence of that failure. Personally, I think that is sad. What is sadder than this failure, though, is that the hoary old 'but this is what the market demands' excuse has been reeled out again and again, as if repeating it in some way improves it as a justification for breaking what has for so long been held up as one of the industry's leading ethical directives.

One of my letter-writing disputants came close to realizing my position had an ethical dimension, but he characterized it as an academic point of view. I often do tackle things head on from the theoretical ('academic') end of the spectrum. Such is one of the 'benefits' of not directly supporting a particular product and being more free to avoid the biases that so-doing are likely to impart on one's view of how things should be. I think it was George Bernard Shaw who said 'The reasonable man adapts himself to the world; the unreasonable one persists in trying to adapt the world to himself. Therefore, all progress depends on the unreasonable.' I make no apologies for unreasonably seeking progress…

Nick Fitzgerald
Computer Virus Consulting
New Zealand

Mind the Root

I have just read the article *Mind the Gaps* (see VB, June 2001, p.6) about the *Solaris*/sadmind worm.

Here are some more facts, taken from the *F-secure* Web site: 'The worm goes through random Class-B subnets looking for unpatched *Windows NT/2000* machines running *IIS web server*. If a vulnerable machine is found, the worm will copy the "\winnt\system32\cmd.exe" to "wwwroot\scripts\root.exe" directory and replace "index.htm", "index.asp", "default.htm" and "default.asp" files with its own.'

The most important thing this worm will do to a *Windows NT/2000* system is copy cmd.exe to scripts, renaming it as root.exe. This means that anyone in the world can access the root.exe through a simple URL, even if all security patches are applied after the attack.

The big problem is that root.exe will not be detected by any anti-virus product, and only some will detect the message in HTML format – which is the only opportunity to find

root.exe. I hope that all anti-virus products will soon be able to detect the HTML message.

When installing a *Windows 2000* server *IIS Web Server* is automatically installed by default, so all test machines in a DMZ will be open to hacking. I saw this virus in our firewall logs eleven times between about 8 and 31 May this year. It only took two days from setting up a *Windows 2000* machine in DMZ to catching a live sample with root.exe.

There are many test-servers which have been hacked but, with no homepage, no one can tell that they've been accessed. And there are many hacked machines left open to access because, due to lack of information about it, the root.exe remains on the machines.

The interesting thought is that, if the virus has only made a copy of cmd.exe and renamed it to root.exe, very few of today's anti-virus products would be able to detect the copy of a legal file.

Klas Schöldström
Brainpool Consulting AB
Sweden

Another Scheme of Retrospective Testing?

After reading Peter Morley's Opinion in last month's VB issue I needed a while to think about his ideas to look at special virus collections from vendors for detection scores of the programs from the same and other vendors. Do we really need to test how many viruses can be found by *Symantec*'s *Norton Anti-Virus* in the two-month-old *Sophos* collection?

I fully agree that retrospective tests should be carried out to look at the program's ability to detect new viruses. At best, they should be carried out over a longer period of time to collect some data about the programs and see whether a program had only a 'good day' at one test or whether it really does have a high generic detection rate. Such periodic tests do not exist at the moment, but we are working with the University of Hamburg to try to achieve something in this area. Some papers and a lot of statements currently exist about this.

However, back to Peter's suggestion. I'm sure that we need some more comments from people both inside and outside the lab about the pros and cons of such tests. And, if we really need another test scheme, it would have to be a completely new way to test, and I have seen no papers on this subject yet – we need to build something scientific around it, before we can start. I'm sure nobody wants to do a test just because he can do it. And I do not want to look at my test results and see that *Kaspersky Lab* has the highest detection score in its own collection, even if it's three months old. ;-)

Andreas Marx
University of Magdeburg
Germany

VIRUS BULLETIN **©2001 Virus Bulletin Ltd, The Pentagon, Abingdon, Oxfordshire, OX14 3YP, England. Tel +44 1235 555139. /2001/$0.00+2.50**

TECHNICAL FEATURE 1

Evil Elves

Marius van Oers
McAfee AVERT, NL

Recently, Unix/*Linux* malware has begun to appear in significant quantities; a large number of worms have been encountered. Technically, viruses can infect office applications such as *StarOffice* using macros, while other opportunities for infection include shell scripts and Perl scripts.

However, infecting binary executable files is a little harder to achieve. Apart from root access issues, simply getting a binary executable to run successfully on different Unix/*Linux* flavours is not easy. Nevertheless, *Linux* binary executable viruses do exist and their number is on the increase. Nowadays, ELF (Executable and Linkable Format) is a standard file format. This article looks at the issues that arise when detecting ELF binary malware.

ELF Header

ELF files usually contain: [ELF Header], [Program Header Table], [Segments], [Sections], [Section Header Table] (although not all of these need to be present). Furthermore, a segment may contain multiple sections. (For more information see the VB 2000 paper *Linux Viruses – ELF format*.) Figure 1 shows the ELF Header from a sample file called 'arch', from *Linux/*RedHat v5.2. The File Header starts at offset 0x0 and ends at 0x34.

```
00000000  7F 45 4C 46 01 01 01 00  00 00 00 00 00 00 00 00  ▯ELF............
00000010  02 00 03 00 01 00 00 00  60 84 04 08 34 00 00 00  ........ y..4..
00000020  C0 07 00 00 00 00 00 00  34 00 20 00 05 00 28 00  À.......4. ...(
00000030  16 00 15 00 06 00 00 00  34 00 00 00 34 80 04 08  ........4...4y.
```

Figure 1: ELF Header of *Linux*/RedHat v5.2 file 'arch'.

ELF binary files start with the signature found at bytes 0x0-0x3. At byte 0x4 we have the class: value 1 for 32-bit, value 2 for 64-bit files.

On *Intel* systems the class is usually set to 1 for 32-bit file layout. This may change when *Intel/AMD* CPUs migrate to full 64-bit architecture. *Dec (Digital) Alpha* systems are 64-bit, but they are not in common use, and support for *Dec Alpha* is decreasing. Current *Sun Solaris* machines also support 64-bit and, although they are not very common at the present time, since the price has dropped significantly for some versions, it is likely that these systems may become more popular.

Byte 0x5 contains the data encoding; value 1 represents LSB (least significant byte) and value 2 represents MSB encoding (most significant byte).

Normally (i386), the data encoding is set to LSB. However, on *Sun Solaris* systems, it is possible to encounter files using MSB encoding. The ELF binary files included in the so-called *Solaris*/Sadmind worm used MSB encoding.

Byte 0x6 contains the ELF Header version number. The bytes from 0x7-0xf are reserved for future use. Some viruses use this area to mark an 'already present' infection. For example, *Linux*/Henky.482 inserts an 'H' (0x48) at offset 0x8 to mark its presence. *Linux*/Dido.478 even inserts a full 'Did0' marking.

```
00000000  7F 45 4C 46 01 01 01 00  44 69 44 30 00 00 00 00  ▯ELF....DiD0...
```

Figure 2: *Linux*/Dido infection marking.

Getting to the File Entry Point

The bytes at offset 0x10-0x11 hold the file type; [1: Relocatable], [2: Executable], [3: Shared object], [4: Core]. In the example in Figure 1 we see that the value is set to 2, so the file is executable. The bytes at offset 0x12-0x13 determine the machine type which, at value 3, means this is an *Intel 80386*-based system.

Most viruses target executable files. However, if the virus doesn't check the file type but just searches for the 'ELF' marker at the beginning of the file, it could end up with corrupted or mis-infected files. In the example shown, the data encoding is set to 1, LSB, so the entry point (EP) virtual address is set to the value 0x08048460 (see Figure 1, the DWORD at offset 0x18).

So how do we calculate the actual EP file offset in the file from this EP virtual address? That depends on the file type. An executable file needs to have a Program Header Table and corresponding Segments. For an executable file the Section Header Table is optional. A linkable file needs to have a Section Header Table and corresponding Sections. For a linkable file the Program Header Table is optional.

If an executable file has a Section Header Table, this may be used to calculate the EP file offset. However, a Section Header Table does not need to be present in order for an executable file to run.

In some cases, for example with the *Linux*/Obsidian.E variant, the virus sets the Section Header Table offset to a fixed value for all infected files, which renders the information meaningless because the true Section Header Table is not at that offset. If the complete Section Header Table is removed from an executable file, the file will still be executable and will run without problems.

EP Calculating Using the Program Header Table

Knowing the EP virtual address, it is not possible simply to deduce an 'Image Base' value to get to the file entry point as the 'Image Base' is not constant.

The Program Header Table contains items that describe the various segment entries. From Figure 1 we see that the segment Header Table starts at offset 0x34, and it has five (offset 0x2d) entries of 0x20 (offset 0x2b) bytes each. Each segment entry in the Program Header Table consists of the following: [TYPE] segment type, [OFFSET] segment file offset, [VADDR] segment virtual address, [PADDR] segment physical address, [FILESZ] segment size in a file, [MEMSZ] segment size in memory, [FLAGS] segment flags, [ALIGN] alignment, in file and memory.

For each segment entry it is possible to determine its segment virtual address and its memory size, thus giving a virtual address allocation range. So, read in the given EP virtual address (offset 0x18) and determine which segment entry it matches within its range. The difference between the virtual entry point (EP) and the segment memory offset can be used to calculate the file entry point by adding this difference to the segment file offset.

In this example we have five segments, numbered 0 to 4. Segment 2's virtual address start is 0x08048000, memsize 0x0585, so its virtual allocation range would be 0x08048000 – 0x08048585, and we see that the given EP virtual address 0x08048460 fits nicely within this range. So Segment 2 contains the entry point. The difference between 0x08048460 (virtual EP) and 0x08048000 (Segment 2 virtual start address) is 0x460 bytes. But, as the Segment 2 file offset start is set to 0x0, the EP is at file offset 0x460 from the beginning of the file.

Infection Spectra

Some viruses append or prepend the viral code. When it comes to native ELF binary infections, viruses may directly change the EP or leave the EP unaffected and change the actual bytes at the entry point.

Viruses may add or enlarge a segment and/or sections, and viruses that change only the EP and the Segment Table in the Program Header Table (but do not change the Section Header Table information) are often seen. For execution this has little impact, but for linking it would not be good. The dual character of ELF files means that care should be taken when cleaning files – beware of 'over-cleaning' files and be careful of areas where a virus may not have touched at all.

Getting ELF binaries to infect is difficult to achieve. In the *Windows* world, standard versions would be Win95, Win98, WinME, Win2k, and soon WinXP but, in the Unix/*Linux* world, there are many different flavours with different versions. Getting ELF binaries to infect can be hard to achieve. Sometimes viruses don't work at all, crash with a core dump, or they infect but corrupt the file. A remarkable item was seen in the form of Unix/Sizer which added script code to binary files. W32/Lindose came initially in PE format and, when run on a *Windows* system, it also searched for ELF binary files to infect. (These infections were then called *Linux*/Lindose.) Although this approach is remarkable, most systems don't have such a setup and its impact was small. It would have been astonishing if it had been a dual-initial (Pe+ELF) infector, but that is hard to do, if even possible.

Linux Worm Packages

Recently some *Linux* worm packages made it to the headlines (for example *Linux*/Adore). *Linux* systems can be set up to be strict in terms of access rights. But these systems do have vulnerabilities that worm packages exploit. Usually such packages consist of scripts and ELF binary files. The script routines may initialize the dropping of new files onto the system or replace files already present on the system with compromised files from the package. Port scanning is common with worms to search for possibly vulnerable systems to infect.

Sometimes files are dropped to hide active processes. For example, the file 'ps' might be replaced. The regular ps file gives a brief overview of current processes, so by replacing this file, the worm processes will no longer be directly visible.

Availability of open-source code means that it is easy to modify it and create a slightly modified file that has basically the same features as the official code except for the changed code.

The appearance of *Linux*/Cheese to apparently fix problems that *Linux*/Lion created was remarkable to see, but as the package was spreading itself, the cure became a virus itself. Other malware targets other operating systems such as *Sun Solaris* and *BeOs*. A recent example is the *Solaris*/Sadmind (alias BoxPoison) worm. *BeOs*/Kate.A is apparently a script virus but it did not replicate on our test systems.

Unix/*Linux* systems are not immune to viruses and other malware. Currently we have about 90 entries in total. Virus writers appear to have discovered the Unix/*Linux* world. When *Windows* virus writers use their programming knowledge to create *Linux* viruses, we should see an increase in the number and complexity of *Linux* viruses. For example, 'Henky' is an ELF infector, and W32/Lindose (*Linux*/Lindose) was, apparently, written by a virus writer who usually writes PC viruses.

Conclusion

It remains to be seen whether *Linux* versions will achieve a large share of the desktop OS market. However, Unix/*Linux* malware is appearing in increasingly noticeable quantities: many worms have been encountered recently and viruses, Trojans, denial of service attacks, flooders and rootkits have appeared. The number and complexity of ELF binary *Linux* viruses is increasing slowly. Software developers like *Borland* make easy-to-use compilers, and the availability of the open source version of *Kylix for Linux* (aka *Delphi for Windows*) from July this year will make it easy to create more worms.

TECHNICAL FEATURE 2

The 32-bit Augean Stables

Myles Jordan
Computer Associates Inc, Australia

I remember the first time I saw Win95.SK. It was so advanced for its time that two years later it remains one of the two most difficult Win32 viruses to detect (along with Win95.Zmist). At the time, it was the only one of its kind, a *Windows* virus with EPOT (Entry Point Obfuscation/Obscuring Technology).

Although they have been around for about eight years, there remain very few viruses with EPOT in existence. Despite their small numbers, EPO viruses have been the cause of a hugely disproportionate amount of work for AV researchers – mostly due to the extreme difficulty of detecting, let alone removing, certain EPO viruses. There are, however, a number of issues relating to the removal of EPO viruses that have been largely avoided by the AV community, in particular the removal of multiple viral infections when one or more of the infecting viruses is an EPO virus.

Properties of EPO Viruses

As anyone familiar with these viruses will be aware, EPO viruses disguise their presence in a unique manner. Where non-EPO viruses will change the entry point of a program to point to the viral code, or insert code at the entry point to redirect control to the viral code, an EPO virus will insert its redirection code at some quasi-random location inside the program's code.

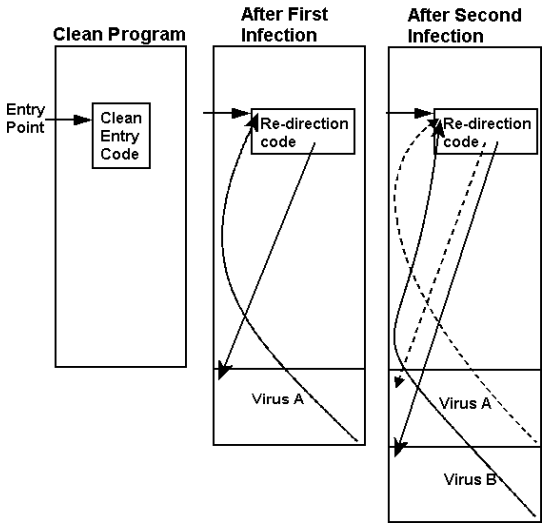
In many cases, the redirection code is not within a linear code flow from the beginning of the program, i.e. it is not possible to disassemble or even emulate from the entry point to the viral redirection code. Importantly, some of the particularly challenging EPO viruses are only realistically detectable by finding this redirection code (e.g. Win95.Zmist).

This means that AV software must not only be able to detect and probably decrypt the body of an EPO virus, it may also have to search the entirety of the program's code for this (usually) very small piece of redirection code and replace it with the original code.

Multiple non-EPO Viral Infections & their Removal

As previously noted, when a non-EPO virus infects a program, it will, in some manner, redirect the initial flow of control directly to the virus. If another non-EPO virus then infects the same program, it will do the same, i.e. redirect the flow of control (away from the first virus) to the new infection (see Figure 1). After it has finished executing, however, the latter infection will restore (what it believes to be) the clean entry code, and return control to that code. In fact, this has actually returned control to the first virus, which will then execute and, in its turn, restore what really is the clean entry code. Finally, the clean program will be executed.

Figure 1



Thus, when AV software examines the program (and follows the flow of control) the only virus that is immediately obvious is the latter because this virus is, in effect, 'hiding' the former from the view of the AV software.

This is important because, in order for the AV software to remove both viruses, it must remove the viruses in reverse order of infection. This is for two reasons: first, the AV software scanning the program will probably not be able to detect the former infection until the latter has been removed (because it is 'hidden') and, secondly, even if it could detect the former before removing the latter, the latter virus will probably contain clean code inside its own body which it saved from the program before overwriting it with its own code. If the latter virus is removed before this clean code can be extracted and restored, the program will never be cleaned successfully.

Normally, the 'Last First' criterion for removal of viruses poses no particular problem to AV software – after it has detected and removed the latter virus, the former becomes visible, and the software consequently detects and removes that virus also. However, if one or more of these multiple infections is an EPO virus, then satisfying this 'Last First' criterion becomes crucially important.

Multiple EPO Viral Infections

When AV software scans a program for viruses, amongst other things, it runs through its list of relevant known

viruses and attempts to detect each of them in this program. In the aforementioned example, it would not matter if the software attempted to detect the former infection first; that infection will remain hidden until the latter infection (which is not hidden) is removed, i.e. irrespective of the scanning order, the latter virus will be detected first; the 'Last First' criterion will be met.

However, if one of these viruses were an EPO virus, then (due to the fact that EPO viruses do not necessarily change the program's entry code) neither of the infections would be hidden from the AV software.

This might appear to be a good thing; but consider what implications this has for the 'Last First' criterion: with both infections visible to the AV software, only one factor remains to determine which of the infections will be detected and removed first – the order in which they are scanned for.

Without the inherent support non-EPO viruses have for the 'Last First' criterion, it is obvious how easy it would be for AV software to corrupt programs during the cleaning process: because the former infection is earlier in the list of viruses to be scanned for, it is detected first, and, because it is an appending virus, the program is truncated at the start of this virus' body, inadvertently removing the body of the latter virus also. The piece of clean code replaced by the latter virus with its redirection code can now never be restored.

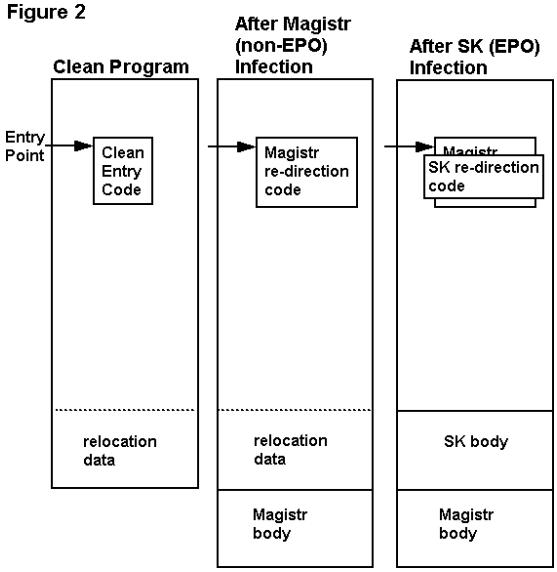
The harmful effects of this problem cannot be underrated. There was a period last year during which both Win32.Funlove (non-EPO) and Win95.MTX (EPO) were very common In the Wild. At that time, we received samples of files that had been infected repeatedly by both viruses, but cleaned as if Win32.Funlove were the only infection in the file, which meant that there were still many pieces of remnant MTX redirection code inside the code section of these files, causing them to crash consistently. In this case, Funlove would have been earlier in the list of relevant detections, but even had MTX been detected first, the file would simply have been corrupted in a different way.

A Possible Solution

If we consider that the crux of this problem is AV software inadvertently removing the infections in the incorrect order, then a possible solution presents itself. Instead of removing any infection as soon as it is encountered, the software could continue scanning for viruses until its list has been exhausted, storing the location of the main body of any viruses detected in the file.

The stored locations of the main virus bodies can then be used to determine the order of infection (later in the file means later infection) and the viruses can be removed in the reverse order. Using this method, the 'Last First' criterion can still be met in the majority of cases.

Figure 2



A Further Problem

Unfortunately, even using the method described, real cases exist in which AV software can go wrong and permanently corrupt files. This can occur if changes to the program made by one infection interfere with the detection of another. Take the situation shown in Figure 2, for example: Win32.Magistr (a non-EPO virus) has infected the file, appended itself to the end, and overwritten 512 bytes of the clean program's code with its redirection code. Later, when Win95.SK (an EPO virus) infects the file, it overwrites the relocation data with its own code but, importantly, it chooses to overwrite some of the area used by Magistr for its redirection code.

If AV software should scan this file, the chances are that it will not be able to detect the Magistr infection at all, so it will go ahead and remove the SK infection. This involves restoring the code overwritten by SK and truncating the program at the beginning of SK's body (even though SK is classed as an overwriting virus, the relocation data that it overwrites is almost always located at the end of the file, and almost always unnecessary). Unfortunately, this will also remove the body of the Magistr infection, leaving a piece of Magistr's redirection code in the file, never to be removed.

Conclusion

Although the situation described in the above example is somewhat contrived, the fact remains that the spiralling complexity of many 32-bit viruses, coupled with the burgeoning size of programs, leaves an enormous amount of room for unpredictable events and situations to occur. With the situation that AV software may be unable to perform virus removal in *Windows* programs with a 100% guarantee that it will not leave the program irreparably corrupted (especially when the corruption may not be noticed for some time), the question must be asked: 'Is it even worth attempting to clean *Windows* programs at all?'

FEATURE 1

Scriptography

Righard J. Zwienenberg
Norman Development, NL

Script viruses become more common every day. Several script languages support virus ability, but the most popular platform today is Visual Basic Scripts (VBS).

Back to Basics

Why did the Visual Basic scripting language become so popular? Since the scripts are written in a reasonably simple-to-understand language and are usually stored as plain ASCII files, they are easily viewable in, for example, *Notepad*. More importantly (and more worryingly), they are easily modified in *Notepad*. Any fourteen-year-old kid who wants the world to focus on his nickname for a few hours can change a script, or at least learn the basic elements to create a complete new virus.

Initial distribution is easy as well: a virus can be distributed worldwide almost instantly using USENET. Just post it to a few erotic-themed newsgroups disguised as an interesting pose of a famous beauty and someone is bound to be gullible enough (perhaps driven by hormones) to double-click the attachment. Closer to home, any scrupulous employee can use a script (virus) to bomb their employer's network, causing downtime which results in financial damages.

The effect of scripts or script viruses can easily be seen by the spread of different mass-mailing variants of VBS/LoveLetter and VBS/VBSWG in recent history. Several variants of these two viruses have been widespread around the world, and most people will have encountered them on their system. (If you are one of the lucky few who did not encounter a copy, just browse your anti-virus vendor's Web site where you are sure to find descriptions.)

Mass-mailers

Most Visual Basic Script viruses are mass-mailers. They send themselves around using the address book of email application *Microsoft Outlook*. With many users who will blindly double-click any attachment they receive, a virus spread in this way has a high chance of success. The simplest way to erase this problem would be to stop using *Outlook*. However, since *Outlook Express* comes pre-installed on all *Windows* systems, and since *Outlook* has been made the default mail program by many corporate businesses, this is a vision that is hardly feasible (nor, in most cases, desirable).

Therefore, other possibilities must be explored to make the system as safe as possible. Stripping all script attachments at the email gateway is certainly an option (why would you want your employees to receive scripts by email anyway?), but it will only stop the viruses deploying mass-mailing techniques. It will not stop Visual Basic Script files being executed within the organization, especially not if they were created by someone on the inside.

Removal of the *Windows Scripting Host*

All the functionality of Visual Basic Scripts is taken care of by the *Windows Scripting Host* (*WSH*). This is present on almost all systems operating under *Windows*, though most people neither have a need for it nor even know it is there. The most abrupt and secure solution, therefore, is to remove the entire *Windows Scripting Host*.

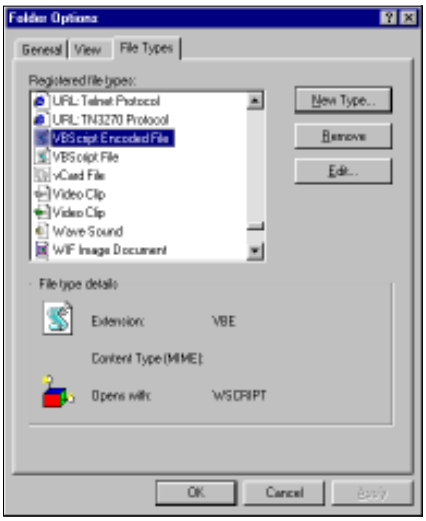
For *Windows 98* (only) the following procedure will remove the *Windows Scripting Host* from the system: from the Control Panel, select 'Add/Remove Programs' and switch to the tab labelled 'Windows Setup'. Select 'Accessories' and click on 'Details' at the bottom of the box. Scroll down the box until you see 'Windows Scripting Host' and deselect it. Select 'OK' and press 'Apply'. This will complete the un-installation of the *Windows Scripting Host* from the system.

Although removing the *Windows Scripting Host* is the safest way to prevent Visual Basic Scripts, there may be several legitimate reasons why it should not be removed. Some companies use Visual Basic Scripts for in-house applications or distribution. For such companies this raises the question of how they can make their corporate environment safer without having to dismiss their Visual Basic Scripts with long-established functionality.

Minimizing the Risk

Although not the safest option, there is a way to minimize the risk and continue to use the in-house build Visual Basic Scripts. However, this method does rely on the renaming of all in-house Visual Basic Script files to filenames with the extension '.yourextension' (using almost any extension you choose – be creative!).

To start, we will need to remove the association of the .VBE (Visual Basic Encoded File) and .VBS with the *Windows Scripting Host*. In *Windows 98* this can be done as follows: double-click the 'My Computer' icon on the desktop, select 'View' and then 'Folder Options'. Next select the 'File Type' tab and scroll down until you find the item 'VBScript Encoded File' (VBE). Select the item, press the 'Remove' button and confirm the action. Repeat for the 'VBScript File' (VBS) item. The association between the extensions .VBE and .VBS and the *Windows Scripting Host* has now been removed.



As you will have to rename all your in-house scripts to the new extension, the *Windows* Operating System will have to be instructed in associating the new extension to your scripts. Before *Windows* is able to associate it, the newly created extension must be recorded in the registry. The easiest way to do this without becoming a REGEDIT expert is to create a small text file 'REGISTER.REG' containing the following few lines (where '.yourextension' is substituted with the extension you have selected). The text file can be made using *Notepad*:

```
REGEDIT4
[HKEY_CLASSES_ROOT\.yourextension]
@="VBSFile"
```

Start REGEDIT by clicking on the 'Start' button, select 'Run', type 'regedit' and press enter. After the program has started, select 'Registry' and then 'Import Registry File'. Browse to the file you have just created and open it. After the confirmation of REGEDIT, all files with the extension '.yourextension' will be treated as Visual Basic Script files. If you rename all your in-house script files to files with the extension '.yourextension' they will operate as before. (Note that for *Windows* Operating Systems other than *Windows 98*, the procedure may differ somewhat.)

Even if the above procedure is carried out, you cannot be certain that a mass-mailer or an otherwise malicious Virus Basic Script will never hit you. Scripts can still be built from within an organization by someone with inside knowledge of the new file extension. An employee with enough knowledge could actually reverse the above action to enable the '.VBS' extensions again.

Outlook on Security

Since almost all mass-mailing script viruses use the *Outlook* application, how can *Outlook* be made more secure? Luckily, starting with *Outlook 98*, *Microsoft* has added several security restrictions that will prevent most viruses from entering as well as leaving the system (see http://office.microsoft.com/Downloads/9798/Out98sec.aspx and http://office.microsoft.com/downloads/2000/Out2ksec.aspx). Some security features have been made for *Outlook 97* but not enough to cope with all the current possibilities.

Making sure that all the required patches are applied will guarantee that click-happy fingers will not be able to double-click any script received (or any other restricted file-type for that matter).

Level 1 Security Files are files that may contain executable code or contain links to files that may contain executable code. Visual Basic Script files are found among these file types. Whenever a message is received with an attachment that is a Level 1 Security File type, *Outlook* will deny access to the attachment. A complete list with extensions within the Level 1 Security can be found in the frequently asked questions (FAQ) list for *Outlook 98*, or later at http://office.microsoft.com/assistance/2000/Out2ksecFAQ.aspx.

When sending an email with an attachment that is classed as a Level 1 Security File, *Outlook* will warn the user that the recipient may be unable to access the attachment if the recipient is also using a version of *Outlook* with the security updates.

Another security feature is that *Outlook 98* and later will no longer blindly allow itself to be used for mass-mailing by external code using the address book or contact list without alerting the user and requesting permission to do so. The user is prompted by a pop-up box asking whether the action is allowed and, if so, for what length of time (selectable). If the action is disallowed, access to the address book will be denied.

With these additional security features in *Outlook*, everything will be much safer, but as long as there are people using older versions of *Outlook* (*Express*), or people who allow the mass-mailing, we will not have seen the last of it. And, of course, security updates to email applications will not stop those Visual Basic Script viruses that spread without using *Outlook* or that simply perform a direct action.

Scripting the Future

Script viruses are not limited to Visual Basic Script: Jscript, mIRC, CorelScript and PHP-Script are just a few other virus-infectable platforms. And the list of script virus-infectable platforms will most likely increase over the next few months.

A script virus for a new platform has been discovered recently – a mass-mailing virus for the Macintosh with the temporary name Mac/Simpsons@mm. At the time of writing this article, further information about this new script virus was unavailable, but its discovery clearly indicates that the virus writers are looking for new and different ways of deploying their creations en masse.

The anti-virus industry will need to follow the virus writers' path to protect their users and try to keep one step ahead of the virus writers. Who will win? Which platform is next? What application is next? Who will be the next victim? Time will tell…

FEATURE 2

Script Kiddies

Berni Dwan
Freelance technology writer, Ireland

'Beauty is truth, truth beauty – that is all
Ye know on earth, and all ye need to know.'

I am not sure if the beauty of programming code in the true hacker sense can ever compare with the beauty of the subject of Keats' Grecian Urn, nor if renegade copies of the urn ever stood up to the original. But the issue of people taking the beautiful (or clever) parts of an expert's work to save themselves the bother of creating something original is causing a significant problem in the world of computer security.

This is exactly how script kiddies (ankle biters, packet monkeys – call them what you will) operate – easily obtaining and trading malicious code for malicious intent. Script kiddies are dangerous not because of their technical prowess, but because of their lack of it. Many will thoughtlessly run a script on a system without having any understanding of the consequences of their actions – let any toddler hold the garden hose on full blast and you will achieve a similar mess.

The Rise, Fall and Rise of Virus Writers

Matt Richtel and John Markoff have charted the rise, fall and rise of virus writers. Over the years, they say, virus writing has been perceived as having lower status in the hacker community than cracking into government and corporate computers. Recently, however, virus writing, with its attendant publicity, appears to have become more attractive to hackers.

Richtel and Markoff make the point, though, that while virus writing traditionally attracted a more technically-oriented set, hacking accommodates a wider range of skill levels. But, with the proliferation of *Windows* technology and the World Wide Web, the less talented punters with dubious motives have jumped at the chance of snatching the easy pickings. Writing in the *New York Times News Service* (April 1999), Richtel and Markoff say: 'In recent years, virus writing has experienced a resurgence, generally attracting a less technically adept group. Increasingly, simple templates are available for use in virus writing and breaking into computers, making the endeavour open to copycats and less adept programmers. In the underground, these copycats are known as script kiddies. In the world of virus writing, they are termed "scripters".'

In 1985, hackers coined the term 'cracker' in response to the journalistic misuse of the word hacker. More than mere arrogance, it was a question of skill, enthusiasm,

talent, expertise and curiosity on the part of the hackers and not the crackers. The hackers did not want this gift list inadvertently bestowed on crackers just because journalists couldn't be bothered to learn the difference.

Once relegated further down the ladder, crackers, not being victims of pride, plied their trade quite nicely for many years until another generation of intruders became apparent during the 1990s.

The Birth of Script Kiddies

The 1990s saw the first of the wonderfully named script kiddies. The name misleadingly suggests gifted, youthful innocence, or child prodigy computer programmers. Of course, nothing could be further from the truth. While crackers reigned on the hacker parade but never made the grade, script kiddies seem to have the time and the vindictiveness to do what many even modest computer users could do, but just wouldn't bother.

Script kiddies have had everything handed to them on a silver platter, from which they can just mix and match like bored children in a room full of Lego pieces. Has technology created them, or the other way around? I'm not sure.

I would be inclined to think that script kiddies have more street cred than brains, in that they can work the system at a very superficial level, but do not possess a deeper knowledge of the cogs and wheels as it were. For them, a passing familiarity has bred contempt and, while their elders may not possess this familiarity this will not always be the case. Even script kiddies will grow old and make room for another generation, and with computer years being more akin to mouse years, the change may happen sooner than they expect.

In many ways script kiddies make me think of musicians who record cover versions of the well-known compositions of more famous artists – some also try to imitate the original recording artists. This musical equivalent of

copying, cutting and pasting assaults our hearing as brashly as script kiddies assault our computer systems.

The *Jargon File*, aka *New Hacker's Dictionary* (http://www.tuxedo.org/~esr/jargon/html/index.html) gives the perfect definition:

> '**script kiddies** pl.n. 1. [very common] The lowest form of cracker; script kiddies do mischief with scripts and programs written by others, often without understanding the exploit they are using. Used of people with limited technical expertise using easy-to-operate, pre-configured, and/or automated tools to conduct disruptive activities against networked systems. Since most of these tools are fairly well known by the security community, the adverse impact of such actions is usually minimal. 2. People who cannot program, but who create tacky HTML pages by copying JavaScript routines from other tacky HTML pages. More generally, a script kiddie writes (or more likely cuts and pastes) code without either having or desiring to have a mental model of what the code does; someone who thinks of code as magical incantations and asks only "what do I need to type to make this happen?"'

Easy as Child's Play

If you read the literature that accompanies most Internet security products, you will invariably find lists of all the DoS, DDoS, and the latest cutely named macro viruses and Trojans they will protect you against. The prevalence of these is not due to any groundbreaking programming discovery, rather it is due to their ease of creation and absurdly trivial ease of deployment.

Elizabeth Weise writes*:* '99% of attacks are launched by what security experts call "script kiddies". With no technical knowledge, these would-be "crackers" don't write their own code. They just drop in at one of the many illicit Web sites offering cracking programs, or scripts.' (*USA Today*)

There are countless off-the-shelf security products to protect our computers from viruses, Trojans, worms or break-ins using clever and innovative detection and prevention techniques that improve with every new release.

The Cost of Carelessness

Notwithstanding, businesses routinely get caught out, either because they do not have the correct type of product installed, or because they have failed to configure it correctly. How many companies still allow file attachments with .VBS extensions to pass through undetected to the recipient's desk? It is this type of carelessness that has played into the hands of the script kiddies, allowing them to make a significant negative impact on the business community, and undeservedly attaching the word 'phenomenon' to their folklore.

Carelessness, though, has probably only made the script kiddie a passing threat, while the traditional virus writers and hackers will always remain the real threat. Unlike script kiddies, they do not court publicity, as they wisely know that those who truly have something to hide do not engage in bravado.

As Jon Katz, writing in *Time Europe* (May 2000) says, 'Because most viruses require little in the way of programming skills, real hackers deride the "script kiddies" and "packet monkeys" who carry around tattered copies of *The Giant Black Book of Computer Viruses*, instant-message each other and hang out on Internet Relay Chat, trading bits of renegade code and bragging about their exploits.'

Notwithstanding the script kiddies' lack of a deeper technical prowess, the fact is that the fruits of their somewhat overrated labour constitute a very real and current threat to corporate, government and educational Web sites. Consider the ease with which DoS and DDoS attacks can be launched, causing loss of services to the most auspicious of organizations. Few things have been more aptly named than the infamous Ping of Death, although this can now be circumvented by most access control products.

The harder-to-prevent DDoS attacks, though, are a gift to the script kiddie because DDoS toolkits are widely available on the Internet (Tribe Force Network, Trin00 and Stacheldraht). Preventing DDoS attacks requires a combination of system administrator vigilance and the use of port scanning tools.

Point and Click Kiddies

The abysmal truth is that the work that goes into protecting computers against the script kiddie payload is far more onerous than the effort employed in launching their attacks. Point and click is the order of the day as most of the tools the script kiddies use to launch an attack are automated and require little interaction. When you think about it, many three-year-olds have already mastered the point and click part, so what payload can we expect when they have learned to read and acquired an 'attitude'?

The Honeynet Project (http://www.project.honeynet.org/papers/enemy/) observes the tools and methodologies of the script kiddie and points out that most tools employ the same strategy: develop a database of IPs that can be scanned, then scan those IPs for a specific vulnerability. Preying on the knowledge that many people do not monitor their systems, it is not difficult to exploit a system and subsequently use it as a launching pad.

Furthermore, Honeynet points out that scan results are often archived and shared for use at a later date, if new system vulnerabilities are discovered. More bizarrely still, script kiddies can buy databases of vulnerable systems, saving themselves the bother of even scanning a system before they exploit it. Once they know that port 139 is open, you might as well be a client on their Local Area Network, with the script kiddie as your system manager!

FEATURE 3

AV Contingency Planning

Dan Dumond, Security Consultant
Sensible Security Solutions Inc, Canada

When a business is one of the first victims of a virus outbreak, its survival can hinge on the speed of its reaction and recovery. Companies need to put procedures in place to mitigate damages caused by malicious code activity. These should include establishing a corporate anti-virus policy, determining an anti-virus strategy, and designing a recovery plan to counter a virus outbreak.

Establishing a Corporate Anti-virus Policy

Establishing and publishing an anti-virus policy serves as the foundation for dealing with malicious code threats in a computing environment. This is usually included in the company's IT Security Policy and describes in simple terms the anti-virus strategy, anti-virus products and configuration requirements that must be met at each tier in the computing environment.

Insisting that employees sign an Acceptable Use Agreement prior to granting them access to computer systems and network services presents an ideal opportunity to include a discussion of the anti-virus policy. The policy describes when and how employees should report suspicious activity, and where to get appropriate assistance. Such a policy helps prevent end-users from taking matters into their own hands, and allows anti-virus administrators to track incidents.

To enforce the policy, many companies use some form of penalty for non-conformance based on the type of incident and the critical value of the affected data or system. The policy should apply to each individual in the company, from CEO downwards, including contractors, and especially the system administrators. But the early establishment of an anti-virus policy is only a first step. Malicious code threats and information systems are constantly changing. To monitor these changes and revise the anti-virus policy accordingly, an anti-virus strategy must be implemented.

Implementing an Anti-virus Strategy

Anti-virus administrators must keep abreast of the latest security threats, and be prepared to respond to virus emergencies. Even when things appear to be running smoothly on the surface, some pro-active measures can help identify security risks and malicious code incidents that would otherwise go unnoticed.

External sources of IT security information should be monitored, including virus alert mailing lists, anti-virus vendor Web pages and IT security news sites.

Manufacturers' security bulletin pages should be checked to ensure that the latest operating system and application security patches are tested and deployed as soon as possible.

Internal resources must be monitored as well. The most obvious sources of virus incident information are the logs generated by the anti-virus software. Compiling statistics from these logs allows anti-virus administrators to discover vulnerable areas in their environment, and examine internal virus incident trends.

Security logs and reports generated by firewalls, network operating systems and intrusion detection systems are also important sources of virus activity information. Some viruses and Trojans have the ability to receive and transmit information over the Internet using specific ports. Documenting suspicious traffic at the firewall, and pinpointing its source, can help determine whether a Trojan, worm or virus has compromised an internal system.

Viruses such as W32/Funlove attempt to infect files over network shares using the current user's security context. If object access failures were audited on *Windows NT/2000* operating systems, administrators reviewing the logs would notice something suspicious taking place. This information can help determine the source machine and user account of the system compromised by W32/Funlove.

Monitoring email queues for suspicious activity, such as large mail volumes, or an unusual series of identical messages, can also help identify virus or worm activity. Helpdesk support call reports can also be monitored to determine whether problematic machines have experienced symptoms of malicious code activity.

Several tools can be used individually or in combination to determine the source of the activity, examining running processes, network communications, file system and registry access. Typically, these are used locally on the affected machine. Suspicious symptoms can be researched on the Internet, through support forums and anti-virus vendor virus encyclopedias. In addition, it may be possible to contact the anti-virus vendor's technical support service for assistance in identifying the threat. If the suspect file appears to be a new or unknown malicious code threat a sample should be sent to the anti-virus vendor's research

centre for further analysis. This is crucial for the development and release of signature updates.

Earlier this year, a virus named VBS/VBSWG.J (aka AnnaKournikova) spread rapidly around the world. The worm was not terribly new, nor original, yet it forced a number of companies to shut down critical email servers to mitigate the inevitable denial of service. Use of basic content filtering software at the Internet email gateway would have allowed administrators to block the messages by subject, attachment name, or attachment type.

Unused applications and services can also be disabled to prevent them from being exploited by malicious code, such as the *Windows Scripting Host*, chat programs, unused email and instant messaging clients.

Sometimes companies who have deployed state-of-the-art anti-virus solutions are hit by an internal outbreak because the software was not enabled and/or not up to date. Synergistic control mechanisms can be used to deter users from tampering with the anti-virus software. Many products allow administrators to lock down software configuration interfaces. Power users can circumvent many of these locks by editing the registry, or disabling the anti-virus software manually by deleting core files. Additional tools such as login scripts, anti-virus management software, or *Microsoft System Management Server* can be used in combination to help ensure that anti-virus configurations are audited and continuously enforced, making it more difficult for users to contravene policy.

The Road to Recovery

During an outbreak situation, the first step to recovery is to identify the threat and document the affected systems within the company. The identification phase should also include determining the distribution mechanisms of the virus, and its symptoms where possible.

In the event that a virus is spreading rapidly with a destructive payload, it may be necessary to take steps to contain the infection to the affected systems. This may involve physically disconnecting affected systems from the network, or powering them down to prevent further infection and potential file corruption.

It may also be necessary to shut down network services like Internet mail, corporate email, and file services during an outbreak. If these services are business-critical, it may be possible to contain the threat through a strategic service outage instead. For example, a company could stop mail connectors between infected and non-infected areas, or prevent only a handful of infected users from sending outgoing mail by imposing mailbox size limits.

In some cases it is necessary to restrict file services as well. For example, anti-virus products for *Windows NT* servers usually scan files on-open and on-close. However, newly created or remotely modified files can be fully scanned only after they have been successfully written to disk. This means that shared files may be destroyed by viruses even if servers are using the latest anti-virus product with the latest virus definitions. Making critical files read-only through share level permissions, and implementing some file level security can help eliminate these incidents. Despite these efforts, a virus running in the context of a network administrator may be able to circumvent some or all of these measures – a reminder that administrative rights should be assigned sparingly.

Disinfection and Immunization

Once the threat has been identified and isolated to a specific machine or location, administrators can begin to disinfect the affected nodes. Ideally this would be done using existing anti-virus software with updated virus definitions.

If the threat is not currently detectable by the software, the administrators must disinfect machines in person or remotely, using a manual or scripted process. To disinfect a handful of machines, remote control tools may be used. For infections that are more widespread, tools such as login scripts, *Microsoft SMS*, and *Novel Application Launcher* can be used to deliver a scripted fix or an emergency application update.

Scripted fixes can be used to disable malicious processes, remove malicious files, repair modified system files, and clean up remnant registry entries. A standardized desktop environment can make scripted fixes much easier to develop, test and deploy, since fewer borderline conditions need to be planned for. For example, login scripts can be used to associate VBS files with NOTEPAD.EXE rather than the *Windows Scripting Host*. This prevents users from launching VBS files by opening an attachment, or double-clicking on a VBS file.

After disinfection, it may be possible to immunize a machine against reinfection. When the worm or Trojan's filename and target location are constant, a machine can often be immunized by pre-emptively creating a read-only file or folder with the same name in the target location. Once the anti-virus vendor releases a virus definition update, this can be distributed to protect the machines further from the malicious code and its variants.

Conclusion

Anti-virus software is just one of the many tools that can be used to protect information systems from malicious programs. Large and small organizations alike must plan and prepare for situations where they are one of the first targets of a virus or worm. Developing an anti-virus policy is critical in fostering an environment where users and administrators work together to keep the corporate network environment virus-free. Developing and practising a virus outbreak handling process is instrumental in allowing companies to recover quickly and efficiently, even when dealing with a brand new threat.

COMPARATIVE REVIEW

To DOS or not to DOS?

Matt Ham

Following the last DOS comparative (see VB February 2000, p.16) there was a good deal of discussion as to whether the days of that particular test were past. After much pondering, the conclusion was reached that there was no clear answer as to whether the testing of DOS products should cease.

Today's anti-virus market is concentrated on those platforms which are commonly in use in business environments. There are some exceptions – often the result of contractual obligations or loss-leader products for corporations who demand an anti-virus solution for their *VIC 20*-based payroll system in exchange for a more lucrative contract protecting their modern desktop machines. DOS is now falling into the contractual or loss-leader niche and so is, perhaps, no longer a suitable product for review in VB.

So goes one persuasive argument, but the reverse is also true. The *Windows 9x* product range, however hard it may attempt to disguise the fact, is built firmly upon a DOS foundation. Most general users rarely, if ever, see this face of their trusty desktop, but for those dealing with viruses on a daily basis, the situation is completely different. When faced with a virus or worm which locks files or which would infect the scanner or operating system if a scanner were to be run within *Windows*, there is a standard piece of advice offered: reboot to a known clean DOS copy and scan from that. In such a situation a command line on-demand scanner is the tool of choice, on-access scanning being by and large unused.

The state of DOS is thus, as an operating system relatively redundant, but as a tool platform very important, which leads to the format of this test. The tested platform was *Windows 98* booted into DOS, an MS-DOS version of 7 though unreported as such by the VER command.

The tested function was the use of the scanner to perform on-demand scans of the machine rather than as a day-to-day protective application. On-access scanning was not tested in this comparative. On-access scanners were also notable by their absence in the environment tested, perhaps because the role of these products is seen by developers as being primarily in the reboot-to-DOS-and-scan scenario.

For this reason it was decided that VB 100% awards would not be given for this test, in order to retain a consistent basis for their award which includes on-access scanning. It has also led to some less vitriolic comments than might otherwise be expected when products have missed samples in these scans. Script and *Office* files cannot be activated within DOS and are unlikely to be locked or acting as fast infectors. For this reason the scanning of *Office* and VBS files is not of paramount importance as it is for a *Windows* scanner. PE files are of importance, however, since these are very likely to be the reason that a reboot-and-scan would be initiated.

The Test Sets

The test sets as used were aligned with the May 2001 WildList. This contained as an addition to the main list, JS/Seeker.A, which was not included in the test sets. The reasoning behind this is that JS/Seeker is not a virus *per se*, nor even a worm, but much more definable as a Trojan. Additions from the last test sets included the expected large number of VBS worms and macro viruses plus the potentially more interesting Win32 viruses and worms. W32/Magistr.A, the presumed ultimate cause of the SULFNBK.EXE scare, is certainly one to watch carefully. As for additions to the other sets, each had a sprinkling. Full listings of the test sets used are available at the URL given at the end of the test results.

Alwil Lguard 7.70–53

ItW Overall	88.81%	Macro	96.80%
ItW Boot	100.00%	Standard	94.38%
ItW File	88.63%	Polymorphic	92.92%

Alwil's offering opens the batting this month with a performance marred by the perennial googly of extension problems. (For the benefit of those unfamiliar with cricket that will be the last such metaphor.) The ItW display was definitely lacklustre as a result, all .VBS, .CHM, .POT, .PPT, .HTA and .HTM extensioned files being missed. Admittedly, these are not a major concern under DOS, though the omission of .OCX and .HLP files from scanning is more concerning if scans are being performed for unfussy Win32 infectors targeting those file types.

The lack of .PIF scanning is also rather a glaring omission, given that W32/MTX is often scanned for in DOS and uses .PIF extensions liberally in its standard spreading method.

Other than misses due to extension, the results for Lguard are good. Most of the remaining misses came from *Word* macro viruses, whether in the Macro or Polymorphic tests sets, with the notable addition of W32/Tuareg.B in the Standard test set. Also a plus point is the scanning speed, very high up in the rankings on the executable clean set and respectable on clean OLE files.

One lack in capabilities was noted in that no archive file handling was supported. The option which seemed to suggest that it would have this ability did not work as described. Another disappointment came in the scanning of

floppies, where multiple disk scanning was supported, though taking many key strokes for each disk.

The DOS version of the *Avast!* product is one of the more complete suites of those tested. In addition to the Lguard on-demand program there are on-access, integrity and behaviour blocker applications installed by default. Installation is performed through a DOS GUI and gives a number of options as to where and what is to be installed.

The default method of installation activates the on-access scanner, which proved to be something of a liability under *Windows*. After rebooting into *Windows* with the on-access component activated there were sufficient blue-screens that *Windows* was rendered unusable – hardly an ideal situation.

CA Inocucmd 42.02

ItW Overall	100.00%	Macro	100.00%
ItW Boot	100.00%	Standard	100.00%
ItW File	100.00%	Polymorphic	98.90%

In the recent standalone review of *InoculateIT* it was noted that there are two scanning engines within the one product, and this is the case with the DOS scanners too. The *CA Vet* scanner is reviewed below, so the *Inocucmd* command line scanner was used here. This is simply installed by running the *InoculateIT* update program which also serves the *Windows* GUI versions. Installation on a machine with no such *Windows* product offers the option to install the update directly to a directory – a fully functional version of the DOS scanner is contained within the update.

InoculateIT has done well for itself recently, so did *Inocucmd* live up to this history? In a word, yes, but such short answers do not fill pages and incur the wrath of the Editor. Misses were absent from all but the Polymorphic test set, with the culprit being W95/SK8044. This has proved a stumbling block in the past for many products, and remains so for several in this test. In terms of speed, *Inocucmd* lies firmly in the middle ground. Those unsatisfied with such a performance as this have the option of using the *Vet* portion of the *InoculateIT* bundle – which conveniently comes next in the line-up.

CA Vet Rescue 10.3.2.1

ItW Overall	100.00%	Macro	100.00%
ItW Boot	100.00%	Standard	100.00%
ItW File	100.00%	Polymorphic	97.42%

The *Vet Rescue* scanner came packaged as a devoted .ZIP archive but is also available by the same update-file method as its sister product above. The .ZIP version is around 1 MB in size and fits easily onto one floppy. As far as detection goes, the Polymorphic test set was where the only misses lay; in this case the culprit was ACG.B with one errant sample of ACG.A. As a combination team, the two *Computer Associates* products on test detected the complete VB test set.

The small size of the *Vet Rescue* package has led to some slimming on functionality, however, as there is no documented archive scanning supported in this product. Speed of scanning for the non-archived clean sets was average rather than distinguished or dreadful.

Command AntiVirus 4.61.4

ItW Overall	100.00%	Macro	100.00%
ItW Boot	100.00%	Standard	99.61%
ItW File	100.00%	Polymorphic	99.99%

The first of three products based upon the *F-Prot* engine, *Command* was the largest of the three packages, being just slightly too large for a single floppy in its self-extracting form. The *F-Prot* engine is traditionally strong against macros and here this tradition extended to the In the Wild test sets and boot sector infectors. The misses were a total of three samples, the first being a single copy of ACG.A in the Polymorphic set. In the Standard set the misses were the newly added VBS/VBSWG.L and VBS/VBSWG.M.

The scanner here has archive scanning capability, which is off by default, activated only during the scanning of clean archive files for the speed tests. This is, and has been, standard test procedure during the scanning speed tests since they were introduced. (It should be noted that this might cause some relative slowdown in the scanning of the clean test sets for those scanners with archive scanning permanently on.) The possible overhead of determining whether files are archives should, however, be negligible. Even if not negligible, the overhead would also be seen in real-world scanning operations and is thus a valid part of the overhead scanning test.

DialogueScience DrWeb 4.24

ItW Overall	100.00%	Macro	100.00%
ItW Boot	100.00%	Standard	100.00%
ItW File	100.00%	Polymorphic	100.00%

Past reviews of *DrWeb* have noted that some portions still have a slightly DOS-feel about them, in terms of alerts, which might lead to expectations that the DOS product will be well developed. These expectations proved to be valid, with *DRWEB386* being the first product in this comparative to detect all the viruses in the VB test sets. It also proved ideal for scanning the floppy test sets, having an automatic option to continue scanning floppies after one has been tested.

There was one fly in the ointment, however, and that came in scanning speeds. These were towards the top end of sluggish in the products tested, to which was added a lengthy initialisation time when a test was instigated.

On-demand tests	ItW Boot		ItW File		ItW Overall	Macro		Polymorphic		Standard	
	Number Missed	%	Number Missed	%	%	Number Missed	%	Number Missed	%	Number Missed	%
Alwil Lguard	0	100.00%	48	88.63%	88.81%	129	96.80%	78	92.92%	61	94.38%
CA Inocucmd	0	100.00%	0	100.00%	100.00%	0	100.00%	9	98.90%	0	100.00%
CA Vet Rescue	0	100.00%	0	100.00%	100.00%	0	100.00%	91	97.42%	0	100.00%
Command AntiVirus	0	100.00%	0	100.00%	100.00%	0	100.00%	1	99.99%	2	99.61%
DialogueScience DrWeb	0	100.00%	0	100.00%	100.00%	0	100.00%	0	100.00%	0	100.00%
Eset Nod32DOS	0	100.00%	0	100.00%	100.00%	0	100.00%	0	100.00%	0	100.00%
FRISK F-Prot	0	100.00%	0	100.00%	100.00%	0	100.00%	1	99.99%	2	99.61%
F-Secure Anti-Virus	0	100.00%	0	100.00%	100.00%	0	100.00%	1	99.99%	2	99.42%
GeCAD RAV	13	0.00%	4	99.54%	97.95%	0	100.00%	0	100.00%	8	99.81%
Grisoft AVG	0	100.00%	7	98.77%	98.79%	22	99.49%	124	92.22%	53	97.37%
HAURI ViRobot	5	61.54%	64	87.23%	86.82%	543	85.79%	11035	35.54%	692	60.80%
Kaspersky Lab AvpDOS32	0	100.00%	0	100.00%	100.00%	0	100.00%	0	100.00%	0	100.00%
NAI VirusScan	0	100.00%	0	100.00%	100.00%	0	100.00%	0	100.00%	0	100.00%
Norman Virus Control	0	100.00%	0	100.00%	100.00%	0	100.00%	17	97.92%	19	99.52%
Sophos SWEEP	0	100.00%	4	99.21%	99.23%	10	99.73%	191	95.36%	37	99.15%
Symantec Norton AntiVirus	0	100.00%	0	100.00%	100.00%	2	99.93%	0	100.00%	15	99.81%
Trend Micro PCScan	0	100.00%	2	99.92%	99.92%	18	99.69%	42	97.56%	7	99.83%
VirusBuster VirusBuster	0	100.00%	0	100.00%	100.00%	42	98.90%	27	95.72%	37	99.18%

The matter of speed is an area where there can be two schools of thought. *DrWeb* scans all areas for possible problems before starting on a scan and therefore seems slow – but is it preferable to initialise quickly and run the risk of unnoticed problems albeit in relatively unlikely places? There is no real answer to this, though in this case, the paranoia may have been a little excessive in the scanning department, with 16 files declared suspicious in the Clean set.

Eset Nod32DOS

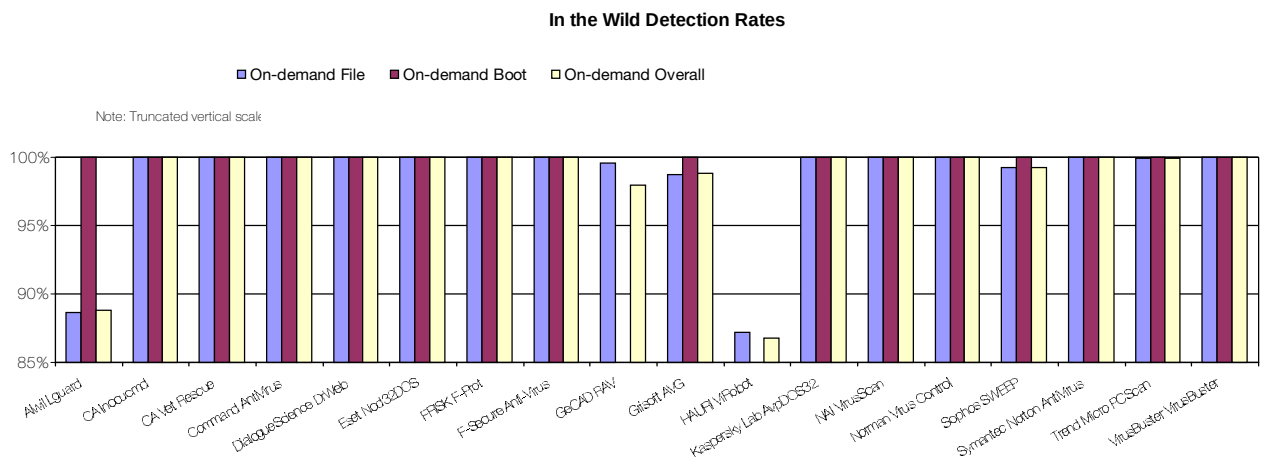
ItW Overall	100.00%	Macro	100.00%
ItW Boot	100.00%	Standard	100.00%
ItW File	100.00%	Polymorphic	100.00%

Another product where good detection rates have come to be expected, *NOD32DOS* has had a notable change in its packaging, doing away with the beautiful purple of recent years for a less stylised design. Despite using such a method of enraging my reviewing sensibilities there are few harsh words that can be brought to bear.

The full detection of all samples in the test set is almost expected now, but the speed of scanning was even more remarkable. When first scanning the Clean set and full VB test set the times seemed far too short to be correct and more time was spent checking for user error than was taken to perform the scans.

As far as interface is concerned, *NOD32DOS* tends to veer towards the DOS GUI. Tests were performed by default on all products from the command line, in order to give more exact control over the selected options. For *Eset*'s product this proved to be far preferable, since the user interface for scan area selection under the GUI could be improved greatly in ease of use and intuitiveness.

Using the command line invoked the same GUI, but with the correct options as instructed in the command line. In scanning floppy disks the GUI did turn out to be a better



mode of use, as it made repeat scanning much simpler than using DOSKEY to perform repeated scans.

FRISK F-Prot 3.09c

ItW Overall	100.00%	Macro	100.00%
ItW Boot	100.00%	Standard	99.61%
ItW File	100.00%	Polymorphic	99.99%

The original source of the three *F-Prot* products tested here, *FRISK*'s product behaved identically to the *Command* rebadging mentioned already. One exception was the smaller size of the archived package – small enough in this case to fit on a single floppy.

The speed tests for the three *F-Prot* products tested are reasonably typical of the variations seen in repeated running of the VB Clean set scanning tests. Variations are typically within ten percent after multiple runs, though for smaller values of time the percentage variation increases to a maximum of around 25 percent. For those readers who spend their free hours perusing the figures, this should give some idea of just how much faith can be put in comparisons of two products by use of the speed test data.

F-Secure Anti-Virus 3.09c

ItW Overall	100.00%	Macro	100.00%
ItW Boot	100.00%	Standard	99.42%
ItW File	100.00%	Polymorphic	99.99%

F-Secure Anti-Virus is the third of the *F-Prot*-based products on test, and much the same applies to this package as the other two (the archive size being slightly smaller than the others). Between the *F-Secure* and *FRISK* versions of the *F-Prot* scanner there are small differences in the documentation supplied which account for the size differential there. The *Command* version of the software has very much a custom installation routine, which accounts for its larger size.

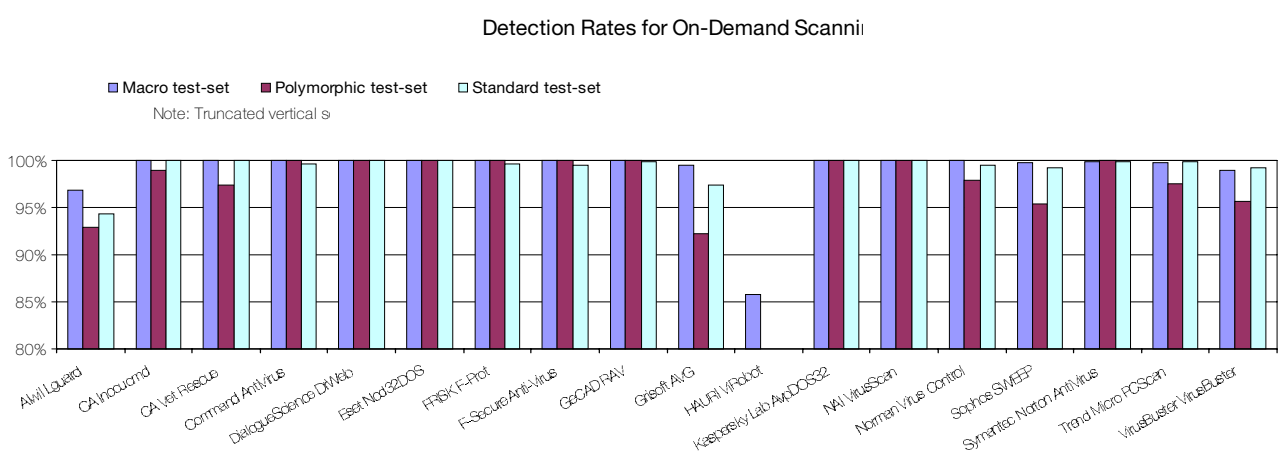
Those familiar with *F-Secure*'s product range will be aware that *F-Secure* was the pioneer of twin-engine scanning methods, using both the *F-Prot* and *Kaspersky Lab* scanners as the standard protective combination. The same is true of the DOS scanners available from the company, although rather than integrating the two engines into a bulky DOS package, the two are in distinct packages. *F-Secure* advised that the *F-Prot* engine be used in this test, but readers should be aware that the *Kaspersky Lab* results are also relevant when considering potential performance for this particular product range.

GeCAD RAV 8.1.001

ItW Overall	97.95%	Macro	100.00%
ItW Boot	0.00%	Standard	99.81%
ItW File	99.54%	Polymorphic	100.00%

RAVLITE, the *GeCAD* DOS product submitted for testing, was initially quite confusing, mainly due to slightly erroneous information in the command line help for the product. For each of the products tested help was available using one or more of the standard /?, /H, -? or –H switches. The amount of information produced by these switches varied from about four lines to four pages and, by and large, contained all the information required to operate the software. In *RAVLITE*'s case, however, matters were somewhat confused by the help claiming that RAVAV –H was the command for help when this was incorrect for both switch and program name.

As far as detection was concerned, *RAVLITE* performed well, with the exception of boot sector viruses where scanning with default setting resulted in no scanned objects and no detections. Misses occurred in both Standard and ItW sets. In the former set, all samples of X97M/Jini.A1 were missed, which is perplexing since the same samples were missed in the last Comparative (*Windows 2000*, see VB April 2001, p.16) and one would imagine they would have been made a priority in the meantime. Less surprising



were the misses of the newly added W32/Tuareg.B in the Standard set.

When speed of scanning is considered *RAVLITE* falls in the respectable middle of the range, though the number of false positives and suspicious files detected in the Clean set were the most of any product tested this month. For archive files matters were more impressive, the OLE scan of archived files being equal fastest, though requiring archive scanning to be activated by a specific command line switch.

Grisoft AVG 6.0.259

ItW Overall	98.79%	Macro	99.49%
ItW Boot	100.00%	Standard	97.37%
ItW File	98.77%	Polymorphic	92.22%

Grisoft's *AVG* shared with one other product the distinction of being supplied integrated within the main *Windows 98* scanner product with an update/upgrade binary being applied to that product in order to update the DOS command line scanner. Although the feature was not required, custom format results files are produced which are viewable within *AVG for Windows 9x*.

Although activating an additional text format report file was not difficult, the results were, in many ways, the most confusing as far as detection was concerned. The scanner was instigated either as a DOS GUI or command line version and the results for the former were far worse than the results for the latter. As with the other products, the command line version was tested, though the behaviour of the DOS GUI version might be of concern.

Also of concern will be the false positives total, the equal highest of any product, though without any erroneously suspicious files and at a speed approaching the slow end of average. Misses were also more than the average, though a number of these, possibly all of those samples missed in the In the Wild set, can be linked to the non-scanning of extensionless and .HTA files. This was not, however, a reason for the misses in the polymorphic set, which were caused by the usual culprits in the form of W95/SK.8044, W95/SK.7972, ACG.A and ACG.B.

HAURI ViRobot

ItW Overall	86.82%	Macro	85.79%
ItW Boot	61.54%	Standard	60.80%
ItW File	87.23%	Polymorphic	35.54%

ViRobot was unique in the test in being supplied as two programs, one scanning for macro viruses and the other dedicated to file viruses. This is not a bad idea if the reboot-and-scan scenario is considered as, by and large, only file viruses and worms are of immediate interest. For the purposes of this test the two products were run consecutively and the results combined.

Unfortunately the detection rate for *ViRobot* leaves a great deal to be desired, even in the ItW set. Misses there included all the .VBS samples and an assortment of other possibly extension-related non detections. More worrying still were misses on such viruses as W95/Marburg, W32/Pretty and W32/Hybris.D, all of which are at the more common end of ItW viruses and worms. Misses in the Macro set were more concentrated on the older *Word* viruses, while in the Standard set older viruses also dominated the misses (though newer samples were also represented).

Scanning speed was relatively slow, with no option to scan inside archives and a single false positive in the Clean set scan. However, there is good news amongst all this. Results are massively improved upon since the last comparative in which *HAURI* was a contender (*Windows ME*, see VB

February 2001, p.17), especially in the Macro set. With another similar improvement in the next few months *ViRobot* will be well on the way to a contender's place.

Kaspersky Lab KAVDOS 3.0.135

ItW Overall	100.00%	Macro	100.00%
ItW Boot	100.00%	Standard	100.00%
ItW File	100.00%	Polymorphic	100.00%

The *KAVDOS32* product was another that detected all the samples in the VB test sets and thus leaves me scratching in the dust for appropriate comments.

Floppy scanning was made easy by default in the product, with a series of disks being assumed as the target. Command line switches required for testing purposes were limited to enabling logs, as was true for most of the products tested. In general, the default settings for these command line scanners were sufficient to gain good results though, in some cases, subdirectory scanning enabling and disinfection disabling were also selected.

Back to the product in hand, a slow scanning speed for *Kaspersky Lab*'s product on the Clean set is the only comment approaching a fault, though as ever, slowness due to thoroughness is not necessarily a bad thing.

NAI VirusScan 4.7.0

ItW Overall	100.00%	Macro	100.00%
ItW Boot	100.00%	Standard	100.00%
ItW File	100.00%	Polymorphic	100.00%

The command line versions for the *VirusScan* product are more than one in number, though *SCANPM* is the most suited to the environment under scrutiny here and was thus tested. Documentation supplied with the various scanners suggests that a root program will select the most appropriate command line scanner for the situation, but this was not tested.

Gratifyingly for *NAI*, the scans resulted in full detection of all files in the test sets and Clean set timings were about the average mark. The previously problematic files in the Standard set, which caused instability in earlier NAI products, now scanned perfectly if noticeably more slowly than other infected files.

One disappointment came, however, with the scanning of floppies. A command line switch /MANY exists for the scanning of floppies, which activates the scanner for the testing of large numbers of floppies in a short time. The function worked perfectly for clean floppies but, on detection of a boot sector virus, the process comes to a halt and single disk scanning is the only option. Users of other *NAI* products should also be aware that superdats do not work to update the *SCANPM* product, the simple dat files being required.

Norman Virus Control 5.10.03

ItW Overall	100.00%	Macro	100.00%
ItW Boot	100.00%	Standard	99.52%
ItW File	100.00%	Polymorphic	97.92%

Norman Virus Control's command line scanner *NVCX* proved similar to its *Norman* sister products in detection rates, scoring full detection in the Macro and ItW sets. Misses in the Polymorphic set were confined to the ever problematic W95/SK.8044. In the Standard set single misses were noted against BAT/911.B and W95/Padania, all samples of W95/Tuareg.B and Raadioga.1000.

Speed tests over the Clean sets were the most mysterious part of the test here, an area where *Norman* has previously had no problems. OLE file scanning was very fast for unarchived files, though slow when the files were archived. More of note was the performance on clean executables, whether archived or not, where some files seemed to induce temporary paralysis in the scanning process.

Sophos SWEEP 3.46

ItW Overall	99.23%	Macro	99.73%
ItW Boot	100.00%	Standard	99.15%
ItW File	99.21%	Polymorphic	95.36%

Sophos SWEEP, the on-demand portion of the *Sophos Anti-Virus* product range, is another product provided with a DOS GUI for installation purposes. This both simplifies and complicates installation, by giving control where none is offered by most of the products on test. A promising start, but let down by the old bugbear of extensions scanned. Although providing no challenge to the *Windows* versions of *SAV*, the OCX sample of W32/Funlove.4099 and .VXD samples of the three W32/Pretty variants in the ItW set were all missed by the DOS scanner. This is almost certainly as a result of missing extensions on the DOS product extension list.

Other than this hitch in the ItW set, and a probably similar problem with BAT/911.A and BAT.911.B, results as far as misses were concerned were fairly predictable. Access viruses A97M/Accessiv.A and A97M/Accessiv.B, the polymorphics ACG.A and W95.SK.8044, the mid-infecting Positron and the .VXD samples of Navrhar were all missed but are usually misses for *Sophos Anti-Virus*. In terms of speed, *SWEEP* was among the faster of the mid-range of clean scanning rates and scored no false positives.

Symantec Norton AntiVirus 7.51.847

ItW Overall	100.00%	Macro	99.93%
ItW Boot	100.00%	Standard	99.81%
ItW File	100.00%	Polymorphic	100.00%

Another product to detect all the samples in the ItW test set, the *NAVDX* scanner missed samples in only the Macro and

Hard Disk Scan Rate	Executables			OLE Files			Zipped Executables		Zipped OLE Files	
	Time (s)	Throughput (kB/s)	FPs [susp]	Time(s)	Throughput (kB/s)	FPs [susp]	Time (s)	Throughput (kB/s)	Time(s)	Throughput (kB/s)
Alwil Lguard	108	5064187		84	944450		N/A	N/A	N/A	N/A
CA Inocucmd	644	849274		75	1057784		1108	143878	180	414486
CA Vet Rescue	703	777997		131	605601		N/A	N/A	N/A	N/A
Command AntiVirus	209	2616900		33	2404054		281	567319	28	2664553
DialogueScience DrWeb	1674	326722	[16]	243	326476		1241	128458	159	469230
Eset Nod32DOS	52	10517926		21	3777798		494	322706	66	1130417
FRISK F-Prot	219	2497407		31	2559154		252	632606	28	2664553
F-Secure Anti-Virus	219	2497407		38	2087731		263	606147	29	2572672
GeCAD RAV	667	819988	4[13]	43	1844971		411	387875	28	2664553
Grisoft AVG	1039	526402	4	169	469431		428	372469	63	1184246
HAURI ViRobot	1589	344199	1	152	521933		N/A	N/A	N/A	N/A
Kaspersky Lab AvpDOS32	1920	284861		158	502112		1392	114523	232	321584
NAI VirusScan	960	569721		187	424245		339	470255	47	1587394
Norman Virus Control	2769	197520		17	4666692		2149	74182	282	264566
Sophos SWEEP	1140	479765		193	411056		643	247926	97	769149
Symantec Norton AntiVirus	1260	434073		177	448213		2325	68566	236	316133
Trend Micro PCScan	806	678576	1	256	309898		986	161680	167	446751
VirusBuster VirusBuster	590	927004		79	1004225	[1]	1073	148571	164	454924

Standard sets. The Macro misses were two samples of PP97M/Vic.A while, in the Standard set, BAT/911.A and W95/Tuareg.B were only partially detected.

Installed with and upgraded through the *NAV Windows* product, the *NAVDX* scanner has the dubious distinction of possessing the largest supporting cast in terms of MB of data required to operate it. This bulk does not make for sleek racing lines and is at a fast average in its non-archive Clean scan rate and positively ponderous while scanning archived executables.

Trend Micro PCScan 7.37

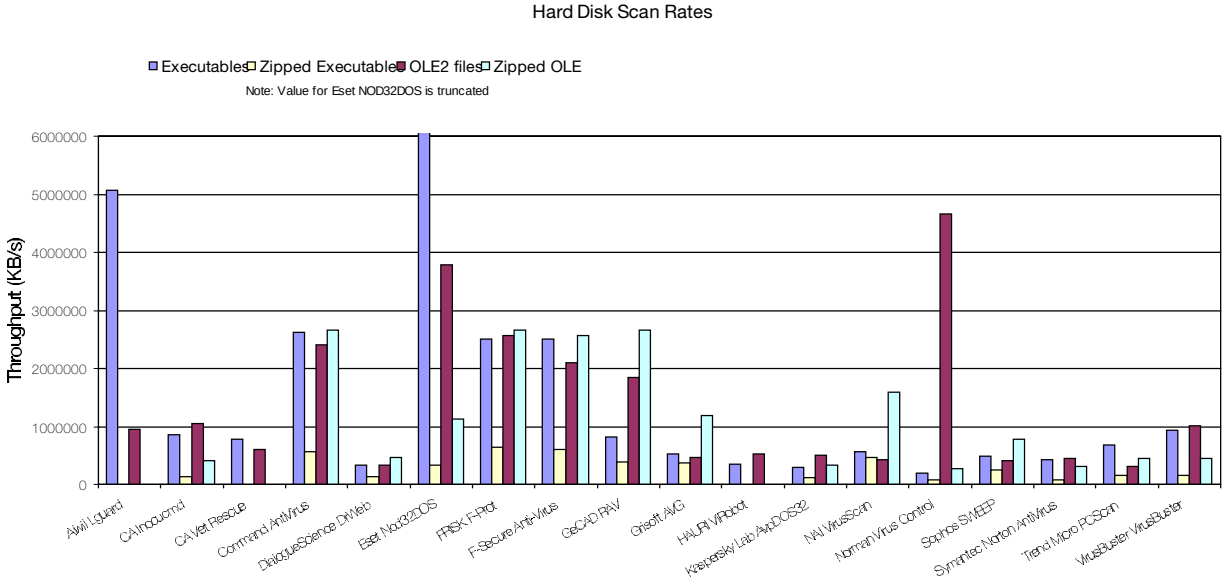
ItW Overall	99.92%	Macro	99.69%
ItW Boot	100.00%	Standard	99.83%
ItW File	99.92%	Polymorphic	97.56%

Trend Micro's products have been a long time absent from the VB testing regime, and it is good to be able to welcome them back. Close to full detection In the Wild, only the extensionless samples of O97M/Tristate.C and O97M/Tristate.D were missed, a problem easily remedied by alterations in the list of scanned extensions.

A selection of macro viruses were also missed, the bulk of these being samples of X97M/Soldier.A and XM/Soldier.A. All other misses were from polymorphic executable viruses, mostly the more modern sort (W95/SK.8044 and W95/Tuareg.B), but with a small number of Gripe.1985 samples also remaining undetected.

The help function available from *PCScan* was more helpful and user-friendly than some of those encountered in other products in this test, although scanning of infected floppies threw up an oddity – the program first declared that the disks were infected, but in the summaries given after this alert there were no infections noted. This was a known feature for several past products, as a result of boot sectors not being counted among those objects listed in scan summaries.

As far as scan speeds were concerned, *PCScan* was neither very bad nor startlingly good, but a single false positive is still one too many.



VirusBuster VirusBuster v11.00.000

ItW Overall	100.00%	Macro	98.90%
ItW Boot	100.00%	Standard	99.18%
ItW File	100.00%	Polymorphic	95.72%

Finishing on a high point for this review, *VirusBuster*'s product is the twelfth of the eighteen products tested to detect all ItW samples on-demand. A slightly less impressive feature was that *VBuster* came closest to recording a false positive in the OLE set; this was only in the shape of a suspicious file. However, this was combined with a slightly faster than average scanning speed over the Clean sets.

Other than ItW misses, *VBuster* managed to combine a few 'popular misses' from such viruses as W95/Tuareg.B and W95/SK.8044 with a scattered selection of more unusual misses in the Macro and Standard sets.

Conclusion

Another Comparative draws to a close and in truth was not as great a cause of hair loss as my first such test, back in February 1998. At that time stability was an issue for several products, command lines were inscrutable and the testing process seemed interminable. At that time DOS was just passing its heyday, DOS viruses were still common and Win32 viruses anything but a real world threat. It could arguably be claimed that, three years ago, DOS was in the same situation as *Windows 98* is in now and the problems encountered in *Windows 98* scanning now are similar in irritation to those three-year-old DOS problems.

One problem unique to this test was the that of determining the name of each product – product name often varied depending upon whether the program names, internal program comments or product documentation were examined.

With stability and ease of use not really an issue in this Comparative, what were the pitfalls? The answer must be extension lists. The issue of extensions to be scanned is likely to cause frustration to those developers who could have had full detection of ItW sets given a more extensive default list of files scanned and they will be itching to prove their worth. With a NetWare Comparative (traditionally home to similar extension issues) due in the September 2001 issue of VB, they haven't very long to wait.

For the user the answer to this problem can be as simple or complicated as the individual decides is required. At the brute force end of the scale lies the blanket scanning of all file types, which is currently relatively common among *Windows* products. If finesse is required the extension lists for most products can be tweaked to suit user preferences, though this involves both second-guessing virus writers and being more attentive to extensions than the developers of the software. Whether either of these is a simple or complex matter I will leave for you to decide.

Technical Details

Test Environment: Three 750 MHz AMD Duron workstations with 128 MB RAM, 8 and 4 GB dual hard disks, CD-ROM, LS120 and 3.5-inch floppy, all running DOS 7. The workstations were rebuilt from image back-ups and the test sets restored from CD after each test.

Virus Test Sets:
Complete listings of the test sets used are at http://www.virusbtn.com/Comparatives/DOS/2001/05test_sets.html. A full description of the results calculation protocol is at http://www.virusbtn.com/Comparatives/Win95/199801/protocol.html.

SUBSCRIPTION RATES

Subscription price for 1 year (12 issues) including first-class/airmail delivery:

UK £195, Europe £225, International £245 (US$395)

Editorial enquiries, subscription enquiries, orders and payments:

Virus Bulletin Ltd, The Pentagon, Abingdon Science Park, Abingdon, Oxfordshire, OX14 3YP, England

Tel 01235 555139, International Tel +44 1235 555139
Fax 01235 531889, International Fax +44 1235 531889
Email: editorial@virusbtn.com
World Wide Web: http://www.virusbtn.com/

US subscriptions only:

VB, 50 Sth Audubon Road, Wakefield, MA 01880, USA

Tel (781) 2139066, Fax (781) 2139067

END NOTES AND NEWS

The Black Hat Briefings will be held on 11 and 12 July 2001 at the Caesar's Palace Hotel, Las Vegas, USA. For further details see the Web site http://www.blackhat.com/.

iSEC Australasia will take place at the Sydney Convention & Exhibition Centre, Australia, from 6–8 August 2001. For information on how to sponsor, exhibit or register as a delegate, visit the Web site http://www.isecworldwide.com/isec_au2001/ or contact Chris Rodrigues; tel +61 2 9210 5756.

CeBIT Asia takes place from 8–11 August 2001 in Shanghai, China. For further information about the conference and details of how to register, visit the Web site http://www.cebit.de/.

i-Security 2001 takes place on 6 and 7 September 2001 at the Putra World Trade Centre, Kuala Lumpur, Malaysia. For further information tel +60 3 21696228 or send an email to sfrc@tm.net.my.

The 11th International Virus Bulletin Conference & Exhibition (VB2001) takes place on 27 and 28 September 2001 at the Hilton Prague. Take advantage of the special VB subscriber rates and reserve your place now! Contact Bernadette Disborough; tel +44 1235 544034 or visit the Virus Bulletin Web site http://www.virusbtn.com/ for a booking form and more details.

COMPSEC 2001 will take place from 17–19 October 2001 at the Queen Elizabeth Conference Centre, London, UK. For more details about the 18th world conference on computer security, audit and control, visit the Web site http://www.compsec2001.com/ or contact Tracy Collier; tel +44 1865 843297; email t.collier@elsevier.co.uk.

Internet Security, which runs from 23–25 October 2001 at ExCel, London, UK, is a new event addressing the security challenges associated with doing business in the Internet arena. A major exhibition will run alongside a comprehensive business solutions-led conference. For more details contact Andy Kiwanuka; tel +44 20 8232 1600 ext. 246, email andy.kiwanuka@pentoneurope.com or visit the Web site http://www.internetsecurity2001.com/.

Central Command has announced that its weekly email AV newsletter now has over 700,000 active subscribers. To join the 700,000, see http://www.centralcommand.com/.

Jan de Wit, the suspected author of the Anna Kournikova worm (VBS/SST-A), is due to appear before a court in the Netherlands on 12 September, charged with spreading information via a computer network with the intention of causing damage. De Wit, who is alleged to have identified himself as the worm's creator, "OnTheFly", faces a maximum sentence of six months imprisonment or a fine equivalent to over £27,000.

***Kaspersky Lab* and Russian Internet company *Port.ru* have joined forces in a project** to provide users of the email service MAIL.RU with free anti-virus correspondence scanning. *Port.ru* estimate that one in three Russian Internet users have a MAIL.RU email account; the new email anti-virus filter is based on the *Kaspersky* Anti-Virus version for the *FreeBSD* operating system. For more see http://www.kaspersky.com/.

DERA (Defence Evaluation and Research Agency) has announced an Internet download facility for its SyBard/Mail software and is offering a version of the software as a free download for personal computer use. SyBard/Mail can be used with most *Windows*-based email applications and will block all unauthorized email release. For further details or a free download visit http://www.dera.go.uk/.

***The Encyclopaedia of Computer Security* (*TECS*) has launched a new service for UK security managers**, in which they will have access to advice from top security experts. Over 100 security specialists have been enlisted, including representatives from *Entegrity*, *Reflex*, *Sophos* and *Checkpoint*. For more details visit the *TECS* Web site http://www.itsecurity.com/.

McAfee* has released *ePolicy Orchestrator* (*ePO*) *2.0, an anti-virus policy management tool that allows monitoring of both McAfee VirusScan and Symantec desktop anti-virus product. Also announced recently is McAfee's development of anti-virus support for *Adobe Acrobat 5.0* software. For more information see http://www.nai.com/.

ICSA Labs, a division of Trusecure Corporation will offer the industry's first continuously deployed testing and certification program for Network Intrusion Detection Systems, to test the functionality and compliance of intrusion detection products. For more see http://www.trusecure.com/.