ISSN 0956-9979 APRIL 2001

VIRUS BULLETIN

THE INTERNATIONAL PUBLICATION ON COMPUTER VIRUS PREVENTION, RECOGNITION AND REMOVAL

IN THIS ISSUE:

• **Nemesis or not?** JS/Unicle has been tripping up even the most effective and reliable anti-virus products in our reviews recently. Costin Raiu gets to grips with this great pretender on p.6.

• **Nice set of features:** ranging from the Ramen worm's attack on *Linux* security to the implementation of a company-wide AV upgrade, our features start on p.10.

• **Shined or shattered:** the usual suspects line up for a crack at this month's *Windows 2000* Comparative Review. Check p.16 for a comprehensive set of results.

CONTENTS

COMMENT

“ the Internet community was still lucky … ”

No Lessons Learned from Love Bug?

Before I start, I think I need to point out that I work for a company that believes the best place for your front-line email defence is at the Internet level. So, caveat emptor; take my conclusions with a pinch of salt – perhaps you will think that the facts deserve a different interpretation.

So here we are, almost one year down the line from the Love Bug, and it seems that no lessons were learnt. Yet again, a simple script virus has employed social engineering to speed round the world in less than a day. Yet again, companies and individuals found their anti-virus strategies and solutions wanting. But is this a fair conclusion? Have we really got no further forward in a year?

My company logs each virus stopped by our virus scanning towers, and we can use this data to analyse trends and make predictions. Since we detected both Love Bug and Anna (also known as VBS/SST.A@mm) heuristically, we know that our data is complete. The figures show that Anna reached its peak twice as quickly as Love Bug. Anna took four hours to reach its maximum spread rate, whereas Love Bug took nine. However, total numbers caught per hour are perhaps misleading – we have taken on a lot more customers over the past year. Analysis of total email throughput shows that Anna only reached a third of the level that Love Bug reached – at peak rates around one in every 100 emails contained Anna, against one in every 30 containing Love Bug. Perhaps the AV industry has made some improvements after all. There may be many reasons for this. Firstly, many more AV packages were able to detect the virus heuristically this time around. This therefore limited the number of available hosts for the initial spread of the virus. Secondly, AV manufacturers were quicker at getting signatures out this time. Signatures started appearing at around 15.00 GMT, which is only 3.5 hours after the initial release, rather than the 10 hours for Love Bug.

The spread of both viruses slowed considerably once signatures were available. Perhaps I am being over-charitable on the speed of signature releases – this virus appeared in the middle of the day (for most AV researchers) rather than the middle of the night. On the other hand, many AV companies are deliberately spanning time zones with their research facilities to be able to cope with viruses appearing at any hour.

What about corporations? Many configure their mail systems not to accept certain attachments, such as VBS files, EXE files, *Office* documents containing macros, and so on. This also undoubtedly limited the initial spread of Anna. However, we saw this virus coming from many large companies, including software companies, law firms, manufacturing companies, IT companies, travel companies and so on. Indeed, the first copy we stopped came from a computer security firm. The human element? Probably more people wisely chose not to open Anna than Love Bug. I don’t have hard figures for this one, but on the various lists I subscribe to, it seemed more people were boasting they got the virus but were clever enough not to open it this time around. However, enough were fooled. Even a week after the event, our support desk was still being asked several times a day ‘to release the picture of Anna that we had quarantined as a false positive’. I take this as a sign that we will never manage to educate the general population to stop opening unexpected attachments.

In conclusion; generally there was improvement, but the Internet community was still lucky. The virus had no damaging payload, but achieved a wide enough spread that if it had, a lot of people would be very sad. Those companies that were infected need to re-appraise their AV strategies. Perhaps they should switch AV products to use ones with a proven heuristic track record – or at least put pressure on their supplier to improve their heuristics. The battlefield has changed over the last two years. Just as companies needed to create a desktop AV budget 15 years ago, they now need to create a gateway level (or dare I say Internet level) AV budget so they can put in appropriate defences against email threats. Training users will never work 100%, given our natural inquisitiveness, but it is still important and should not be neglected.

Alex Shipp, MessageLabs UK

NEWS

A Staple Diet of Worms

There has been little new of significance in the virus scene in the last month or so. VBS/Staple, another unremarkable script virus, mass-mailed itself in that oh-so-familiar way, and sent its anti-Israeli/pro-Palestinian message to 25 Israeli email addresses. It also tried to open some pro-Palestinian Web pages in the victim's Web browser.

CBS MarketWatch quoted the Global Education Director of a major AV developer as saying '[t]his is the first politically motivated virus we've seen in quite a long time'. Presumably he has forgotten that many *Word* macro viruses from Indonesia have political messages and/or politically oriented payload triggers? (Surely he was not unaware of this? Maybe the few weeks since the last such Indonesian virus variant should be considered a 'long time'?) Another spokespiece from the same developer said 'It's similar to the Melissa virus in that it has the potential to shut down multiple servers around the world'. Pity that worldwide there were less than a couple of dozen infection reports… (Oh yes, and of course it was called 'Staple' by some and 'Injustice' by others, just to keep the confusion ratio up!)

'Flash in the pan' [*No pun intended. Honest. Ed.*] would be a good description of Win32/Naked. It started with a hiss and a roar however, with the early victims reputedly in the US DoD and/or military – a traditionally handy place for a mass-mailer to get a boost. Distributing itself as an attachment named 'NakedWife.exe', it should never have got into any self-respecting corporate network. Most companies apparently saw its spread quickly stifled.

However, some other worms made more of a splash in the *Windows* gene pool, not just in the media. Win32/Magistr not only mass-mailed itself but had a CIH-like BIOS re-flashing payload and could 'leak' confidential information by randomly selecting several non-infected files from the victim's machine to send with an infected EXE. There'll be an analysis in next month's issue. Yet few vendors or media outlets seem concerned about VBS/San. Two variants were deliberately distributed by its writer planting messages in USENET newsgroups. Since it is a slow-mailer (like Ska and Kak) it seems set to become more common than all the above-mentioned, and according to some, already is ▮

Whose Side Are You On?

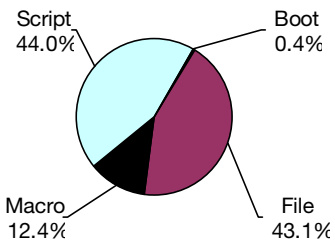
As we went to press, *Central Command* was publicly praising the technical skill of the author of the recently discovered proof-of-concept, cross platform virus capable of infecting *Windows* PE and *Linux* ELF files. Furthermore, *CC* did him the inappropriate courtesy of naming the virus as he would have wished. No wonder credibility is a fragile commodity in this industry. We take a closer look at this virus in next month's issue ▮

Prevalence Table – February 2001

Virus	Type	Incidents	Reports
VBSWG	Script	2076	36.4%
Win32/Hybris	File	1222	21.4%
Win32/MTX	File	647	11.3%
Kak	Script	288	5.0%
Win32/Navidad	File	280	4.9%
Onex	Macro	220	3.9%
LoveLetter	Script	110	1.9%
Ethan	Macro	89	1.6%
Laroux	Macro	75	1.3%
Marker	Macro	66	1.2%
Win32/Msinit	File	50	0.9%
Divi	Macro	49	0.9%
Win32/QAZ	File	39	0.7%
Win32/Funlove	File	38	0.7%
Win32/Plage	File	37	0.6%
Win95/CIH	File	33	0.6%
Win32/Ska	File	32	0.6%
Tristate	Macro	30	0.5%
Thus	Macro	29	0.5%
Win32/BleBla	File	25	0.4%
Class	Macro	24	0.4%
Myna	Macro	20	0.4%
Others		231	4.4%
Total		5710	100%

[1] The Prevalence Table includes a total of 231 reports across 45 further viruses. Readers are reminded that a complete listing is posted at http://www.virusbtn.com/Prevalence/.

Distribution of virus types in repo



LETTERS

Dear Virus Bulletin

University of Magdeburg

This is in answer to Peter Morley's letter in the March issue suggesting the removal of the Old Fashioned File Viruses (OFFVs) from virus test beds. I'm sure OFFVs don't matter much any more. However, in the case of ItW viruses, no test organisation will simply exclude them, because there are still some reported infections. To test these few samples only a small amount of time is needed.

However, in the case of zoo viruses and the large zoo virus collections many testers use, Peter is completely correct. To find nearly every non-interesting virus can be a huge waste of time – firstly, for the developers of the engine and the virus researchers who add them; secondly, for the user who has to wait longer until a scan job has finished. And last but not least it is a pain for the testers, since it can take a lot of time to scan 200,000 or 300,000 files in steps and to have a look at the report files to sort out which viruses are found and which are not.

In February 2000 we made the decision to test the DOS zoo file viruses only on special request and not by default any more. Moreover, we do not make improvements to the DOS zoo collection any more and our concentration is increasingly spent on Win32 and macro viruses. However, three tests (out of seventeen different ones we made) which are available at our Web site www.av-test.org still include DOS zoo viruses (2000-02, 2000-08 and 2000-10).

And if you compare the detection scores there is a surprise in store – on some programs like *Norton Anti-Virus* we saw a big decrease in the detection rate. It seems to be that a newer version of that program doesn't detect all previous viruses any more. What will happen if all testers exclude these types of virus? I hope there won't be other surprises, and it's my opinion that at least a small subset of the zoo viruses should still be included in future tests.

Andreas Marx
University of Magdeburg
Germany

Secure Computing

1. In our annual comparative reviews, we have not tested using our zoo collection for two years. This largely eliminates old DOS file viruses.
2. We are broadly sympathetic to the plight of developers for whom it must be a nightmare trying to program in protection against a soaring number of viruses, the vast majority of which never see the light of day.
3. Peter Morley's 'OFFV' is, however, an ill-defined term and this poses us with a problem.

At the end of the day, as a testing centre of long standing and considerable experience, we will continue to use our judgement on test methodologies. Currently this means that we do not use OFFVs (which are not in the wild) in our annual comparative tests or going forward in our smaller tests. Should the need arise in future to make use of such viruses, we will.

Paul Robinson
Secure Computing Magazine
UK

University of Tampere

In response to Peter Morley – thank you for your proposal to exclude DOS file viruses from anti-virus product evaluations. I certainly agree that DOS viruses are not a vital threat in most current computer systems. However, the argument that no-one is interested in DOS viruses cannot be true. There are still many old computers in use running MS-DOS, and DOS viruses are able to harm even new computer systems. As a counter example one could argue that Macintosh viruses are not a threat to most users since most users do not use Macintosh.

Nevertheless, my opinion is that each anti-virus product testing organisation should decide their own way to emphasise different virus categories and test methods. This on the one hand prevents any bias that would emphasise only certain aspects of anti-virus products and on the other hand allows concentration of resources for specific areas of anti-virus product evaluation.

From this point of view, a common exclusion of DOS viruses does not seem to be a profitable idea. What I understand as important is to categorise different types of viruses into different test categories so that a reader can decide which results are suitable for his environment.

However, more important than which virus categories have been used is to report exact information on how the tests were performed, which methods and which viruses were used, and most of all to ensure high quality of tests.

Marko Helenius
University of Tampere
Finland

Virus Bulletin

We broadly agree with the views of the University of Tampere on this issue but would like to clarify our position on the removal of 'OFFVs' from our test-sets. Currently, these types of viruses are *not* included in the criteria for the

VB 100% award test regime. In other words, a product which does not detect this kind of virus may still qualify for a VB 100% award. This, we feel, reflects both the requirements of the average PC user and the real-life, modern day status of AV protection. However, we do include 'OFFVs' in our other test-sets, specifically the Standard test-set, as a matter of interest to our readers and in response to requests for this kind of information.

Matt Ham
Virus Bulletin
UK

Reporting Back

On Saturday 3 and Sunday 4 March *EICAR* members networked at the conference in Munich as much as possible while getting much work done in committee meetings. Naturally, fun was also high on the agenda. These committees are a very effective way to get involved with *EICAR*, meet other *EICAR* members and, most importantly, keep abreast of great new developments. It was at one of these meetings that I gave a tutorial concerning the 'Protection of IT resources against Viruses and Malicious Code'. It was unbelievable to see that even on a Sunday morning at 9 o'clock we got a room full of participants for such a workshop.

This year marks *EICAR's* 10th anniversary. This was felt to be as good a reason as any to be innovative and launch a few new activities. The most important one is definitely the *EICAR* Anti-Virus Enhancement Program (EAVEP) through which *EICAR*, anti-virus vendors and users are trying to learn more about how and what to improve with AV solutions. We presented the questionnaires for the first phase of the *EICAR* Anti Virus Enhancement Program (EAVEP) and the comments received were quite positive. The questionnaires will now be distributed and Aalborg University will do the evaluation.

Another initiative is *EICAR's* Task Force on Risk and Trust in E-Commerce, launching a few research programs including one about risks and how this might affect user behaviour and E-Commerce success. Finally, the increasing demand for free Web-based content as well as M-Commerce content to fill all the additional GPS and UMTS capacities coming on-line soon (while making investments required more likely to become commercially viable) raised privacy and security issues. *EICAR's* Task Force on Risk and Trust has, therefore, launched a project addressing these issues.

For more detailed information about the EAVEP Program, this conference and forthcoming conferences and initiatives, check the *EICAR* Web site: http://www.eicar.org/. I hope I can see you (again) at the next conference to be held from 8–11 June 2002 in Berlin.

Eddy Willems
EICAR Director of Press and Information
Belgium

VIRUS ANALYSIS 1

Unicley Different

Costin Raiu
Kaspersky Labs, Romania

For most people, the story of the JS/Unicle virus began in the early spring days of 2000. There is one curious detail related to this, which is that back then, for various reasons, the *WildList Organization* was unable to release its March 2000 issue. This is just one of the curious things related to this even more curious virus.

The problems were eventually solved, and there was an April 2000 WildList – and, importantly, that was the first List to include reports of the Unicle virus. Unique from this point of view, the April 2000 WildList is also the last issue to contain reports of the Unicle virus, because ever since there has not been a single further JS/Unicle report.

One can, of course, argue that Unicle was one of those fast-living viruses which explode for a day or two, then simply die. One important aspect supports this theory – Unicle depends on the availability of an Internet connection to replicate. Moreover, it depends on a set of specific components being available for download at a couple of specific Internet locations – an idea also implemented by many other viruses. So, starting from the moment the components were removed from the Internet, the virus was no longer able to spread around the world.

Another interesting detail regarding JS/Unicle is that it is a Chinese-specific virus. By default, this virus will only work on Traditional Chinese *Windows* with Traditional Chinese *Internet Explorer 5.0* or later installed. I'll explain later the reason for this 'compatibility' issue. Also, I believe it is worth mentioning that experience shows that language-specific viruses have few chances of surviving 'ItW'. They may cause local problems, yes, but they will never reach the distribution of, say, W97M/Melissa.A@mm, VBS/VBSWG.J or even W97M/Marker.C.

The Virus Bulletin Tests

Following JS/Unicle's inclusion in the WildList, *Virus Bulletin* (with which you may be familiar,) added a couple of JS/Unicle virus samples to its test-sets. From the start, this proved to be a major problem for several products, which no longer achieved the 100% perfect ItW detection for a VB 100% award. In short, products which were able to detect every single sample from the WildCore (a collection of viruses found in the WildList mainly used as a reference test-set for many AV comparative reviews), were missing the JS/Unicle samples from *VB*'s test-sets.

The mysterious reasons for the problems reported in the *VB* tests were unfortunately unavailable for quite a while, but they become obvious to me when I tried to replicate Unicle, as you can see below.

The Unicle Virus

The Unicle virus arrives at a computer via an infected HTML email, which mainly contains a short JavaScript routine which drops the virus body onto the system. Under normal conditions, the JavaScript routine from the infected email should not be able to do things such as writing to your disk, but in this case, using a known vulnerability in *Internet Explorer's* HTML processor libraries (also exploited by the JS/Kak virus family), the routine will create a 'Scriptlet.TypeLib' (.HTA) file in the system's Startup folder.

Unicle will perform a lot of tests in order to find the right name of the Startup folder. It will check if the current *Windows* installation resides in the following directories: WINDOWS, WINDOW, WIN, WIN98, WIN95, WINDOWS.000, WINDOWS.001 both on the C: and D: drives.

There is an interesting aspect regarding the routine which drops the virus in the Startup folder – the routine does not drop the file in the directory %WINDOWSDIR%\Start Menu\Programs\Startup. Instead, it will use a construction of the form %WINDOWSDIR%\Start Menu\Programs \<UNICODE Sequence> where <UNICODE Sequence> is the Traditional Chinese equivalent of the Startup folder: a 4-byte UNICODE string containing the following ASCII characters: 177, 210, 176 and 202.

For this reason, the virus will only work on Traditional Chinese *Windows* installations. Moreover, if one tries manually to create the respective directory and force *Windows* to use it as an alternative Startup folder, the dropping component still does not work. Apparently, *Internet Explorer* will refuse to create the .HTA file unless it is the Traditional Chinese *Internet Explorer*.

This detail is important because I believe it explains why Unicle so problematic to AV products in the *VB* tests. Basically, to create the (.HTA) instance of the virus, both Traditional Chinese versions of *Windows 95/98* and *Internet Explorer* are needed. Tweaking the path and similar tricks in English versions of *Win9.x*, will simply not work as the *IE* library components will not drop the Scriptlet.TypeLib file if the path contains UNICODE characters.

Now, I believe some AV producers do not pay attention to viruses which require special, complex setups to replicate. Others simply do not have access to the respective operating systems and tools. Coupled with other factors such as false .HTA Unicle files being distributed in various anti-virus collections, files which were actually the .HTM form of the virus, the more or less complicated setup required to

replicate, Unicle prevented some anti-virus developers from getting a .HTA form of the virus. So, products which needed special definitions to detect the virus in .HTA files were simply unable to detect it without an .HTA sample to extract a detection definition from.

From the Start(up)

Returning to our analysis, upon next system reboot, the file named MICROSOFT INTERNET EXPLORER.HTA dropped by Unicle in the Startup folder will be loaded and executed as any regular .HTML file containing JavaScript code. When executed, yet another component – called MSIE.HTA – is dropped, this time in the SYSTEM folder.

The virus will run the respective file after dropping it, and moreover, it will append a line to the system file WIN.INI to execute the dropped component each time the system is started. After that, it will cover its tracks by deleting the file initially dropped into the Startup folder.

MSIE.HTA

The primary purpose of this part of the Unicle virus is to connect to one of a list of ten FTP sites and download the main replication component, found in the form of a file named MSIE.EXE. The list of sites used by the virus is: dany777.homepage.com; iopi999.homepage.com; pop123.homepage.com; todo888.homepage.com; ftp.todo.com.tw; hammer.prohosting.com; hammer.prohosting.com catv169.homepage.com; catv170.homepage.com and todo 168.homepage.com. Please note that hammer.prohosting .com is deliberately listed twice.

Various usernames are employed, depending on the target site, randomly selected from the above mentioned list. For all the FTP accounts the password is the same: 995119. In order to connect to the respective sites, Unicle creates a simple FTP script file which is fed into the FTP.EXE utility using the '-s command-line switch.

At the time of writing, all the accounts on the FTP sites used by the virus are no longer working. Either the accounts were deleted or the passwords were changed, but in all cases, they are not accessible to the virus. Thus, the main replication component cannot be downloaded.

The conclusion is that now (March 2001) Unicle will be unable to replicate, in any place in the world, except under lab conditions. Even though the Win32 binary components of JS/Unicle are not available for download, back in the days when the virus was still able to replicate correctly and without any help, many AV researchers collected the respective files for further analysis.

MSIE.EXE – The Archive

The file MSIE.EXE, obtained from one of the ten custom sites, is a PKZIP SFX archive which, when executed, will unpack two files on the disk, namely EXPLORER.EXE and MSWINSCK.OCX. The former, a VB5 program, is directly run by Unicle, and will take care of the important task of replication. The virus will also modify WIN.INI to run this program upon each system startup.

Using MSWINSCK.OCX, a simple TCP/IP socket library module, EXPLORER.EXE attempts to send copies of the worm to as many email addresses as possible – obtained by scanning the disk for files with the extensions *.SNM, *.DBX, *.NCH, then scanning the content of such files for addresses. These files are various mail formats' indexes, for example *.SNM files contain email addresses for the messages received in *Netscape Messenger*, *.DBX for *Outlook*, and so on.

To mass mail itself, Unicle uses direct SMTP access – it will extract from the Registry the address of the default SMTP server, connect to it, and email copies of itself to every email address collected by the brute force scan operation. Probably in order to make things simpler, EXPLORER.EXE contains a copy of the initial HTML part of the virus, which is sent along with the emails.

Unicle also includes an 'I am alive' function, common in most of the Internet-aware viruses we see nowadays. Thus, notification emails will be sent to one of following: leebill_001@yahoo.com, or leebill_023@yahoo.com, but not to leebill_006, leebill_013 or leebill_16@yahoo.com. Moreover, EXPLORER.EXE contains code to 'backdoor' the system on which it is running, another pretty common facility for today's malware.

Conclusions

The JS/Unicle virus is a strange combination of loaders, droppers and Trojan modules. Designed to run on Traditional Chinese *Windows* alone, this virus is a little bit tricky to replicate, which I believe is the main cause for the AV product misses we've seen in previous *Virus Bulletin* Comparative Reviews.

Unfortunately, this proves once again that every virus AV labs receive should be replicated, analysed, and then proper detection should be implemented in the products. I have no doubt many AV companies did this with Unicle, but unfortunately I have no doubt some did not.

JS/Unicle@mm

Aliases: W32/RunFtp@mm, I-Worm/Unicle.

Type: Email propagated Worm, written in JavaScript and VB5

Detection & Disinfection: Use an anti-virus program able to detect and remove the worm. An AV solution implemented in the mail router/server level is recommended.

VIRUS ANALYSIS 2

Tricky Relocations

Péter Ször
SARC, USA

Last December I came across a bizarre Dynamic Linked Library (DLL) that our product had a minor issue with. It was part of a *Borland Quattro* product and linked with a *Borland Linker*. The Base Address value of the DLL was set to 0x2CC – weird. That value is clearly incorrect. Applications could never load to such an exact address since the file is always mapped from the start of a given page. Furthermore, the address is far too low to be correct. I realized such a file would be incorrect, not accepted by the system loader and never run. I was partially right. The LoadLibrary() function did not load the DLL under *Windows NT/2000*. However, *Windows 9x* systems loaded the file nicely. This was well worth testing.

Basically, the *Windows 9x* loader does not check for such a specific case. Therefore, if the application or DLL has relocations (in the .reloc section) the loader will try to relocate the image to an available correct address in the process address space. One more mystery solved.

It crossed my mind that a 32-bit virus could probably use a trick which would force relocation to be used, and then add an item to the 'to do' list: 'Applying relocation items for PE files.' This trick can be applied at least two different ways. It can either use an incorrect Base Address as mentioned above (this only works under *Windows 9x*) or try to load the image to an address that is not available, since a system DLL is already loaded to the address (which works, with limitations, under *Windows NT/2000*).

What could be the use of such a trick in a virus? Relocations were used in the DOS days as an anti-emulation/heuristics method. For instance, the Tentacrille virus used a trick that was based on EXE relocations. It is not difficult to see how a 32-bit *Windows* virus could implement such an approach for a similar reason. 'It is always good to know something before the virus writers catch on, since we might have the support for it in our engine before they realize the possibility' I thought, and informed my fellow anti-virus researchers about the problem. Who would think of such a trick? Less than a week later I saw the first virus to use forced relocations. It is obviously a virus created by a 29A member. The actual virus is not really new since it is largely based on the Resur virus – we called this new one W95/Resurrel, indicating the relocation trick.

W95/Resurrel is the first encrypted binary virus that does not implement a decryptor. The virus code is encrypted and runs just fine when the application is executed. It uses the forced relocation trick and lets the system loader decrypt the virus via relocations.

How does Resurrel work?

W95/Resurrel is written entirely in C and 80% of the code is based on the W95/Resur virus. When an infected file is executed, the virus will execute the original host application as a thread. Resurrel can work silently in the background and infect local as well as network drives with their base address value set to 0x00400000. It does not infect DLLs.

The virus has four different sections. It will modify the entry point of the host to point into its own code section. The four different sections of the virus code are patched into the section table if there is enough space in the header. Resurrel is careful not to corrupt the host by overwriting the code right after the section table area. If there is a .reloc section, it will overwrite it. Resurrel uses a mutex set to '29A' in order to run only one active copy at a time. Other executed copies will only run their host program. The virus traverses the directories of each drive and infects files everywhere but the SYSTEM directory.

The virus sets the Base Address of infected files to a DWORD value of 0xBFxxxxxx where xxxxxx is a random value set via the GetTickCount() API. It adds a relocation entry for each DWORD value of its own code section. Each DWORD is encrypted in the following way – the virus adds the new Base Address random value to the actual DWORD of its code section, then subtracts 0x400000 from it. Before being placed into the virus' relocation section, each entry is set to IMAGE_REL_BASED_HIGHLOW type. Finally, the virus adjusts the necessary field of the PE header and the infection is complete.

When the actual infected application is executed the system loader will try to load the image. The value is either wrong (does not start at the beginning of a page) or simply indicates a load over the KERNEL32.DLL which is placed in the same area of the process address space under *Windows 9x*. Therefore, the system loader will check for the available relocations and then it relocates the code completely, thus decrypting the virus code section. Every infected sample is encrypted differently.

Conclusion

A simple string picked up from the code section of the virus will not be sufficient to detect this virus. The virus does not encrypt its other sections in its first release. However, it is better to apply the decryption logic that is simple enough for algorithmic detection. The data section of the virus carries the 'Win95/SVK by Tcp/29A' string and is visible in each infected file.

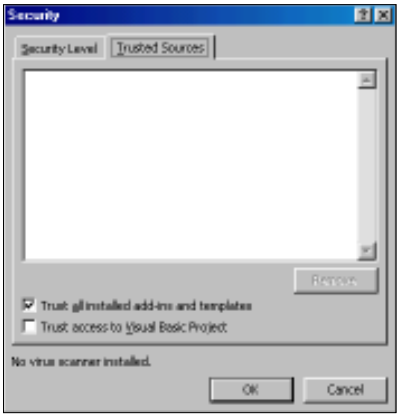
It is time to implement support for relocations for PE infections – other viruses could challenge 32-bit emulators with a similar trick in the very near future.

OPINION

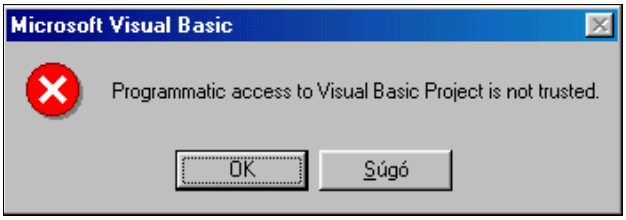
Great XPectations

Gabor Szappanos
VirusBuster, Hungary

As I was test-running *MS Office* Beta 2 (build 10.2202. 2202) I noticed an extra checkbox at the bottom of the familiar Security->Trusted Sources dialog saying 'Trust access to Visual Basic Project'.



As I was one of the people who suggested the complete isolation and hiding of the VBE object model, I felt warmth in my heart and immediately started testing this new feature. As I expected, most of the viruses I tried to replicate failed to infect without any obvious error messages. Running the macros in the Visual Basic Editor, the following error message was generated:



The viability of macro viruses depends on two object models. These are the application object model, which provides the necessary event triggers for virus activation (e.g. a procedure called Document_Open is triggered whenever a document is opened) and the VBE object model, which provides the procedures necessary for VBA code manipulation.

Some VBA applications, like *Microsoft Office*, provide an easy connection between the object models; other applications, like *WordPerfect*, do not. The latter approach practically eliminates the virus threat, as a macro code has access to the document object it is running from, but has no way to insert VBA code into other documents. This is well reflected in the number of known *WordPerfect* VBA macro viruses – zero at the last count.

By way of illustration: the dots in the following expression 'Application.ActiveDocument.VBProject.VBComponents' (commonly used in macro viruses) do not all mean the same thing. The first and the third are references to the property of the object on the left hand side of the dot. The second is different: it is really a gateway between the two separate object models. It would appear that *Microsoft* tried to seal the gateway between the two object models with this setting.

At this point, I felt unsure about my future. As most of the currently known macro viruses use CodeModule object methods to spread (InsertLines, AddFromFile, etc), only accessible through the VBProject object, I thought I would be without work within a year or two. I could not expect virus writers only to create viruses like XM97/Jini, or to warm up the idea of attached templates as used in WM/Dietzel, so it would be a start to hunt for a new job.

Then I realized that this option can be turned on and off. It is a shame that *Microsoft* could not make up its mind and simply remove the access to the VBE object model. There are some applications (code management tools like Code Librarian in *Microsoft Office 2000 Developer* or self-modifying VBA solutions) which, in my opinion, could have been given up, but *Microsoft* once again decided to prefer VBA solution developers to virus experts.

Anyway, as it is it is a selectable option, and the access to the VBA project can be granted. This is not a big problem, as I do not expect users to turn it on, since the isolation of the VBE object model will not be noticeable to 99.9% of them. The only problem is that the value of this setting is stored in a very obvious location and under a very obvious name in the Registry, which makes extremely easy for a virus to disable it in an instant and go on with the infection.

It seems the *Office* application reads this setting during startup and does not notice any change in it until the next startup, so infection procedures have to happen in two sessions: the first time access is granted, and then the second the actual intrusion takes place.

So how much protection will this security 'enhancement' provide? Even less than the improvements introduced in *Office 97 Service Release 1*. It could effectively stop the migration of the vast majority of currently known macro viruses, but the *Office XP*-aware macro viruses that will without doubt appear will easily bypass it. This does not mean, however, that upconverts of *Office 97* viruses will not appear. As this protection can be turned off, there will always be users who will do that, successfully upconverting non-*Office XP*-aware macro viruses. Consequently, the upconversion issues should be addressed anyway.

This poor design made this potentially great security enhancement practically useless. At least it is off by default. It could have been a great move making *MS Office* almost invulnerable to macro viruses. I am not worried about my future any more: I will have my hands full as long as *Microsoft* develops new *Office* versions.

FEATURE 1

Safe or Sorry?

Adrian Marinescu
GeCAD Software, Romania

Linux is a subject that is yet to become commonplace in the pages of *VB*. Last year at the conference in Florida we heard two very good presentations about *Linux* threats – it might be a good idea to see what has happened since then, and how the anti-virus industry should evolve in the next few years in order to be one step ahead of the virus writers who'll choose *Linux* as a target.

Many people consider *Linux* safer than the well-known *Microsoft Windows* systems – most of the *Linux* lovers I know are pretty convinced that nothing can hit them on their favourite Unix flavour. Even the appearance of *Linux* viruses wasn't enough to convince the most sceptical of them – but it seems that where the anti-virus researcher fails, Ramen succeeds.

In my opinion, the only real reason *Linux* viruses/worms are not as successful as the *Windows* ones is the fact that *Linux* used only to be used by advanced users. Once the distributions became more complete and easy to install/use, many users chose *Linux* instead of other operating systems.

When Distributed Denial of Service tools started to be used more and more often to attack various known sites, one problem became clear to the anti-virus industry. Namely, the recompilation of the source code on various machines (or with various compiler switches) will produce different binaries, making the detection of such tools harder to implement. Simple heuristic-based rules were created to detect recompilation of such programs – but Pandora's box was already open.

Since the source code for many software packages is freely available, one can easily recompile every standard package to fit its needs. This is widely used when it comes to gaining root privileges on a remote machine.

For example, once a hacker has entered one system they can easily download the well-known 'login' program and, in less than ten lines of code, trojanize this utility in a way that allows further authentication based on a secret username and password. While this can easily be detected using the 'md5' utility and a pertinent system administration, there are a lot of users/administrators who are simply not that 'paranoid'. In such cases, the system is compromised and special tools are required to detect the changes in order to ensure the system's security.

Anti-virus scanners evolved from simple signature-based detectors of the first known viruses to complex programs able to detect not only viruses, but also Trojans and worms. This flow was quite normal – the anti-virus scanners had all the capabilities to include detection for such threats. Maybe now it's time for another step – should anti-virus scanners detect such 'Trojanized' programs? I strongly recommend this – the technology inside today's programs allows it.

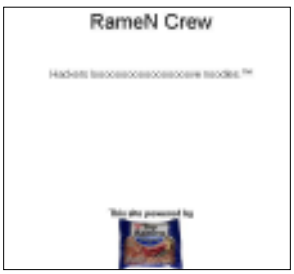
Many analysts suggest that when *Microsoft* created VBS language, they opened the way for mass-mailing worms, but think – under *Linux* it's enough to write one shell-script line to send an email. This is confirmed by the fact that, less than one day after the VBS/LoveLetter discovery, we saw a Unix shell port of this infamous worm.

The Ramen Case

Ramen is the first worm written for *Linux* to be reported in the wild. Several Web sites were affected by this worm back in January. It is a collection of script files and i386 ELF binary files designed to exploit three vulnerabilities in software packages that are installed by default in many *Linux* distributions. Even if Ramen was designed to work only on *RedHat 6.2* and *7.0*, the packages required in order to replicate are present in many other *Linuxes*.

The three security holes were known for a very long time; unfortunately, there are still unpatched systems that can be affected by these holes, more than six months after the problem was solved by the developers. All the vulnerabilities enable remote users to gain root privileges and run malicious code on the remote machines. Ramen uses the 'wu-ftp' and 'rpc.statd' vulnerabilities for *RedHat 6.2*, and the 'LPRng' vulnerability for *RedHat 7.0*.

When a host is affected, Ramen files are automatically copied to the new host and installed in /USR/SRC/.POOP. The starting point of the worm is a script called START.SH. When executed, it will immediately call the payload routine, searching for files named INDEX.HTML from the root directory. When such files are found, Ramen will overwrite them with its own copy, making every affected Web site look like this:



Next, Ramen will delete the file /ETC/HOSTS.DENY, so that no connection restrictions can be used in the future. A script named GETIP.SH is used to find out the host's genuine IP address, which is saved for further use into a file called MYIP.

Since it uses different exploits for *RedHat 6.2* and *7*, the Ramen worm needs to check which version is currently running on the machine. To do this, it checks the existence of /ETC/INETD.CONF. If the file is present, the version is

assumed to be *6.2* – but other *Linux* distributions do use this configuration file as well.

After copying all the required files, Ramen will register itself in the /ETC/RC.D/RC.SYSINIT script to start every time *Linux* boots. Next, it will register a service into the /ETC/INETD.CONF file, called ASP, that is responsible with the binary file delivery for the next targets. The service listens on port 27374 – when a client connects it will respond with a compressed file with the entire Ramen package (RAMEN.TGZ).

To avoid reinfecting the same machine, Ramen patches all the vulnerabilities, depending on the version of operating system. On *RedHat 6.2* it will stop the exploitable service (RPC.STATD), remove and disable anonymous access (the 'wu-ftpd' exploit is based on that). Otherwise, it will stop the LPD demon and replace the binary file with an empty one, and then disable anonymous access.

A tool named 'synscan', modified by the author to fit Ramen's needs, is used to scan for victims. Randomly generated class B IP addresses are checked systematically against the available exploits. If the remote system is vulnerable, the exploit code will create the directory /USR/SRC/.POOP, set the terminal type to 'vt100' (in order for 'lynx' to work) and use the 'lynx' utility to download the package from the attacker.

If the package is received in good order, it will copy the RAMEN.TGZ file into the '/tmp' directory (in order to replicate further), decompress it and call the START.SH script, which will continue the worm's infection process. When a system is successfully infected, the Ramen worm sends an email to each of the following email addresses (stored in encrypted form in the worm's binaries): gb31337@hotmail.com and gb31337@yahoo.com, with the subject the IP address of the infected machine and the message body 'Eat Your Ramen!'.

Viruses

We have already seen a dozen *Linux* viruses. At first they were quite unstable and worked only on some *Linux* versions/distributions. In time, more and more features borrowed from DOS/*Windows* viruses were included in these *Linux* creations. Per-process residency and body encryption are good examples of how virus writers ported some widely used *Windows* techniques on *Linux*. Till now, no polymorphic *Linux* viruses have been discovered, but it's only a matter of time.

Backdoors and Rootkits

Despite the false opinion that backdoor programs are something new, designed mainly for the Win32 operating systems, this kind of program has been present on Unix machines for some time now. Despite this, they do not have such advanced features as their Win32 counterparts – the main feature is the remote-access over the Internet on the affected machine. Using this access, any malicious action can be executed on the remote system.

Rootkits are programs which allow you – having illegally entered a system – to have further superuser privileges. In time, rootkits have evolved from simple tools to very complex collections of programs, designed to install and hide the presence of intruders.

'T0rn' is one of the many rootkits freely available on the Internet. Once executed, 'T0rn' will install itself into the hacked system and provide further superuser access. It will create a directory named /USR/SRC/.PUTA and copy itself in there. Then, it will replace the standard utilities used for file listing and process listing to hide its components from being detected.

Next, it will Trojanize the /BIN/LOGIN binary, to allow the logging of any user that uses a secret password. Finger, which is a common utility installed on *Linux* systems, is also changed to install a bindshell when executed – the bindshell allows untrusted users to connect to the affected machine. SSH, which is a common package that allows a higher level of security, is also Trojanized to allow further logging of the hacker.

Conclusion

It is very hard for anyone to say that one computer system is more secure than another. While researching for this article, I came across one interesting argument which says that *Linux* is safer than *Windows* because it is open-source. While this will certainly help to find and remove bugs inside the software packages, most of the time the exploits discovered are used in malicious attacks. Even though problems are solved very fast by the anti-virus developers, the Ramen case ably demonstrated that the usual problem of users simply skipping security updates applies to those who favour *Linux* too.

Think of recent DDoS attacks for a moment – most of them were initiated from *Linux* systems, and that may be because some of them were quite easy to break. In today's heterogeneous networking environment, you cannot just deny that other kinds of malware do not affect you – a complete security solution *should* cover them all.

While the large number of *Linux* distributions and the differences between them are certainly making the task of designing an all-round, solid protection package harder for the AV developers, the quick reaction of the anti-virus industry when Ramen was discovered gives me good reason to believe that this will not be a problem in the future.

The growing number of anti-virus products for *Linux* should result in a complete solution for the users. Merely detecting thousands of viruses will no longer be good enough for *Linux* users who will need all the security that they can get from an anti-virus program.

So, is *Linux* safer? Maybe… only time will tell.

FEATURE 2

Upgrading Made Easy

Max M Morris
First Union Corporation, USA

I recently headed a project to replace the anti-virus solution in *First Union* bank's corporate environment, and decided to share with you the specifics of the rollout and explain how a large corporate company prepares for and handles such an undertaking. Not long ago, we completed a massive upgrade of the anti-virus product we use for our corporate environment. The huge scope of this project involved upgrading over 325 *Novell NetWare 4.x* and *5.x* file servers and over 22,000 work-stations which logged into them in an eight-week timeframe. During the project, we made sure that there was no impact at the server level as we removed the old anti-virus program and installed and loaded the new solution.

Out of the total number of calls taken by our Help Desk during this rollout, we received 1,064 calls that were categorized as upgrade-related. Of these, the greatest majority (396 or 35%) were actually those which were not specific to the upgrade, but general virus information calls (infections, calls inquiring around new viruses, etc). Of the remaining 668 actual product upgrade calls received, the largest number – 321 or 29% – were install-related. These dealt with install problems caused by insufficient Admin rights on *NT/2000*, lack of disk space, third-party utilities running at the time of install, and so on. We also received 136 (or 12%) general category calls around the product itself (how to use it, why we were upgrading, etc) and the project's schedule.

The issue that got the most publicity during our rollout was performance-related. 166 calls (14%) were received concerning performance problems. The main reason for performance changes was that with the new solution we were forced by the new wave of viruses being created to change our method of scanning files – from scanning selected modified file extensions only to implementing 'all files' access. There was a trade-off in performance moving to this level of scanning, but the alternative was a potentially major virus outbreak that could destroy significant data and cripple our company's business.

The general performance loss was seen most significantly on older, slower workstations with very limited resources (memory and hard drive space). We also ran into one specific corporate-wide application used for mainframe access that in some cases experienced significantly longer load times. This was determined to be caused by how the application was written to load on startup (the actual EXE was called again and again in the source code) and was applicable only when the application was being run from a file server across slower network links. Following is the breakdown of the types of calls to give you an idea of what type of problems you might run into:

Virus infection or enquiry call	35%	396
Install problems (not Admin rights-related)	27%	300
General information calls	10%	114
MF app-specific performance problems	9%	98
Calls incorrectly entered as virus-related	6%	64
General performance issues	5%	58
Insufficient Admin rights for install	3%	32
New AV product problems	2%	30
Project schedule enquiries	2%	22
Printer problems (SPOOLER errors)	1%	11
Vertical application incompatibility problem	0%	3

While we did experience some problems, in all the ratio of total number of true product-specific calls received around the upgrade (668) compared to the total number of work-stations installed (22,030) during our migration was only 3%. We considered this more than acceptable for the introduction of a new product corporate-wide. The biggest positive impact made by the upgrade, in addition to the low impact, was that we now have over 5,000 more work-stations running AV protection than when we started the project. 92% of the total number of *Novell* users who log into these servers now have an anti-virus scanner in place.

So, how did we do such a good job at reducing the potential impact? The following steps that we used can easily be applied and modified to any large or small scale firm to help minimize the problems that you may encounter.

Establish a project team early on

It is very important to identify all your stakeholders and create a project team in preparation for the upgrade. By doing this, you ensure there is no duplication of effort and each group knows exactly what responsibilities they have. I would recommend direct representation from any major support areas, your Information Security department and the area(s) responsible for defining the product standards and configuration for workstations and servers. If you have an outside vendor who handles your problem calls and/or installations, it is very beneficial to have them involved as well.

It is also a great idea to have management representation on the team, at least at the start of the project. Regularly scheduled meetings are important, with their frequency dictated by the progress being made and the timeline of the project. Keep the meetings short and whenever possible have questions answered outside the meeting so that the meeting itself is just a summary of progess to date. Don't just meet to meet – time is precious and everyone is busy.

Test, test, test – and then test some more

You can *never* test too much. Complete testing not only uncovers any problems in the new code but also gets your support areas very familiar with the new product and allows feedback on how the product will be received by your customers. It is very important that the testing includes not only the actual anti-virus code verification but also any upgrade methodology you are using (i.e. silent and automated install process, login script pushes).

Be sure that you test for network-specific dependencies, such as serverless site installs across slow WAN links and dial-up work-stations. Also, in our environment, without the benefit of a locked-down desktop, it was critical for us to allow our departments access to the new code for vertical application testing to help ensure we did not run into problems later.

We ended up performing in-depth testing over several months on over 25 *Novell NetWare* test servers (which mirrored our production standard server configurations) by running real-time and prescheduled scans (stress tests with multiple scans). We also completed detailed workstation-level testing on over 250 individual computers, representing all standard desktop operating systems (*Win9x*/*NT*/*2000*) before the upgrade was started.

A large percentage of our HelpDesk staff was part of this testing phase, allowing them to become familiar with the interface of the new product before the calls started coming in. We also did verification on all our major department workstation images and completed departmental pilots with several major customers.

Customize, and develop a rollout schedule to minimize impact

Do not rely on the defaults of a product. Look closely at the options you have and tailor it to your level of protection. In our rollout, we changed the virus pop-up messages to tell customers to call our Help Desk, changed the prescheduled scans of our servers to avoid our nightly backups and changed the timeframe that clients polled for new pattern files to reduce network traffic.

When you are doing an upgrade which involves not just a large number of devices but multiple departments, it is very important to build a detailed schedule. Do not just grab the first 10 servers for the first night and so on. Build your schedule by starting with a low number of devices (servers and total number of workstations using those servers). As the upgrades are done and little or no impact is seen, increase the total number of workstations per day that are to be upgraded. Always try to increase this total number gradually and keep it even across all the days.

You can never communicate too much

You always want to send out mass communications, leaving 'no stone unturned' so to speak. We provided 'heads-up' messages to let people know the upgrade was coming through emails to our major technology distribution lists as well as corporate broadcasts on our in-house television network, our mainframe system and various Web sites.

We provided a detailed presentation of the upgrade to senior leadership and information officers and utilized a new enterprise email communication process which provided details of the upgrade to all corporate email users when they logged into the email system through a pop-up information box.

It is very important to have one single, easily accessible source for the project communications. We used a Web server because all our customers and support areas had access to our Intranet. All our other communication methods pointed to this one central location.

In all communications, you want to detail all aspects of the upgrade, as well as try to anticipate and answer questions, including why are you upgrading, the scope of the upgrade project, a full detailed schedule, possible impacts, what happens if the upgrade is not done, where to go for updates and whom to call with problems.

Stay up to date with post-migration communication

During the actual implementation of the rollout, we used the same communication methods that we had for the heads-ups. In addition, we provided weekly email reminders detailing the schedule for the upcoming week and a daily update of each night's upgrade with any impacts to our support areas.

Certainly, you will want constantly to monitor calls received by your Help Desk, looking for new problems and/or additional communications which may be needed around the types of calls that are being received and/or escalation methodologies.

Lastly, constantly update the central communications point for your customers with any changes to the schedule, problems seen, etc. Once the upgrade is completed, send out a final communication summarizing the upgrade, including the numbers of devices upgraded, any problems encountered, etc.

Major corporate upgrades to new products are always a challenge. The key to success is to ensure that you plan, test and provide communications from start to finish, always being prepared for the unexpected. Good luck!

A DAY IN THE LIFE

Life Support

Peter Cooper
Sophos Anti-Virus, UK

[*Peter Cooper lives in Oxfordshire with his fiancée, three computers and a cactus called Bonehead – advice on virus removal, network security tips, breaking bad news and meteorological forecasts a speciality. We asked him to describe an average day. Ed.*]

It appears there is a common breakdown in most software houses, more specifically anti-virus ones. There are developers who make software, sales people who tout the software, marketing folks who publicise the software and the support engineers who deal with every conceivable query about anything you care to imagine.

I'm part of a busy support team at *Sophos Anti-Virus* and we're lucky enough to be able to deal with anyone who chooses to use our software. Many employees of anti-virus companies will have a fairly standard day; they arrive in the morning, then sell/market/develop software, have lunch, do some more selling/marketing/developing, then go home. True technical support people are hardcore techies, not your average call centre droid, and can be found working at silly hours of the day. While most corporate users are fast asleep, technical support people are diligently talking to insomniacs with viruses on their sleep-starved minds.

Some may argue that support people just consume valuable resources and company profit. Not so. There are tens of thousands of known viruses. Sure, it's fair to say that the average Joe User isn't going to contract many on their computer-based travels – the numbers are all academic; all it really takes is a momentary lapse of concentration and the damage has been done. I probably don't need to explain how two seconds of double-clicking an attachment of unknown origin can result in the virus writers creation twiddling and diddling the contents of the computer, and a subsequent phone call to the support desk for help 'before the boss finds out'.

The typical anti-virus company support desk will receive many emails and telephone calls throughout the day. Most will be standard requests for assistance, or the occasional 'Oh no! What have I done?' plea after running a suspect file. We occasionally come across phone calls that demand a closer look, ones that give us an insight into the psyche of Joe User, a lone surfer in a crazy, crazy world. One such example follows.

I had a call this morning at 6:20am from a gentleman in the United States of America. His call to our support desk was simple; his anti-virus software had found W32/Prolin in one of his *Windows* .EXE files, and W32/Ska (Happy99) in

another two. Imagine the scene: I'm all set for a long phone call (it's a standard, yet sometimes lengthy, drill), removing these two commonly reported viruses. I anticipate this call lasting about 30 minutes, though it invariably goes on longer. If the caller is computer savvy, and keen, then the consultation may be over quicker. This guy was going to be different, I had that nagging feeling in my bones. The following exchange took place:

Me: Hello, what can I do for you?

Caller: Hi, I've found two viruses on my PC, my [other vendor's] software is not letting me access the files.

This, again, is fairly common. Home users find viral material on their PC, they get some software to remove it, they give the vendor a call, and continue doing what they were doing. Simple. At this early stage in the proceedings, most support engineers would no doubt use their own company's software to confirm the infection, and disinfect/remove material as necessary. This engineer did just that.

Me: Sir, could you please open the *Sophos* software for me. This is done by…

Caller: The what software?

Me: Our anti-virus software.

At this stage I was thinking of the person who wrote W32/Prolin, and how, at 6:23am s/he would be tucked up in bed, no doubt snoring, blissfully unaware of the PC users around the world with this virus on their machines, their MP3 files now sporting an unfeasibly long file extension urging them to change their operating system.

Caller: I don't have your software. I have [other vendor] on my PC. It said I should call you.

Hmm. Interesting support tactic; AV software finds a virus, flags the file as viral, then asks the user to call a competitor's support line. Very interesting indeed. Is the user

confused? Is he being dishonest? Is it a new virus that intercepts AV software and sends the caller into a telephone tailspin? Possible reasons mill around in my head, fighting for lebensraum. It's 6:25am, what's a guy to do?

Me: OK, well, I'm happy helping you out getting rid of the viruses, but you may also want to talk to [other vendor]'s tech support to…

Caller: But it said to call you.

The gentleman was pleasant and concise, fairly jovial, insisted on interrupting me. Knowing full well that we could remove the W32/Ska infection from his PC, regardless of what software he was using, I began to explain what to do. The W32/Prolin would be a little more tricky, and may involve using DOS, not an easy thing to do for many *Windows* users.

Me: The first file you need to find is…

Caller: Can I use [other vendor] to get rid of it?

Me: You can use the [other vendor] software if you wish, sir, but I'm afraid I can't really help you. We only support our own software.

Caller: So why did the software tell me to call you? Why not [other vendor]?

Me: The only thing…

Caller: So you're not [other vendor]? OK! Thanks anyway! [click]

I pause for a moment, collect my thoughts. I have one of those 'It's going to be that sort of day' thoughts. Fully expecting the gentleman to call back in 20 minutes with the same query, I delve a little deeper.

W32/Prolin arrives in an email as CREATIVE.EXE, flags itself as a Shockwave Flash Movie, modifies files, changes this, tweaks that. It distresses me to say it, but this is all quite normal with the current slew of *Windows* viruses. Gone are the days of relatively painless dialog-box style macro virus annoyances; users are starting to wise-up and treat .DOC files with the suspicion they deserve.

Microsoft Excel files are being seen as .CSV more and more often – is this the end of the macro virus plague? I doubt it. There will always be people who don't tread the line between security and productivity; these people will open anything recieved by email. Financial reports, job applications, pictures – I mean, it's an email from your friend and associate, it's got an attachment that says it's a picture of a Russian tennis star, it can't possibly be a virus… can it?

We're finding an ever-increasing raft of Win32 email-borne viruses, these are getting more difficult to remove. It is genuinely hard explaining to a person that the program they ran from their email, SNAPPLE.EXE, purporting to be a Snapple screensaver, was actually a file-infecting, re-mailing, potentially BIOS-trashing nasty, and not the harmless utility they thought it was. Oh, and because they also have virus X, Y and Z, the sensible thing to do is to reinstall the operating system. The documents they thought they had? Sorry, they have gone too. They were deleted by a macro virus that you contracted yesterday, 30 minutes before you downloaded the latest AV updates.

There's only a certain number of times a person can hear anti-virus people evangelising and lecturing on the importance of data security; they either listen, or they don't. Data security is vitally important, it also needs to be interesting to get the message across.

The gentleman caller never did get back in touch – I still wonder what he's doing with his PC. Maybe he called the other vendor. Maybe he gave up and reinstalled his OS. Maybe he switched off his anti-virus software and carried on regardless. It's only a virus after all, what harm could it do? What if his friends and associates get it? They can call technical support and get rid of it, shouldn't take them long.

Many of you will be familiar with the allegation that anti-virus companies write viruses to keep themselves in business. Not so. There are plenty of people writing viruses; we have no reason to. To some extent, support engineers are fire-fighters; virus writers light the touchpaper, the spectators watch the fireworks, the anti-virus guys come along to put out the fires.

The support and developers are looking for areas to fireproof, minimising a repeat performance. There are different types of fireworks, some will explode with a colossal bang, showering debris and sparkles worldwide (VBS/LoveLetter, for example), causing a large number of subsequent smaller fires, down to those that release a small, unimpressive payload that goes largely unnoticed (the VBS Kak worm). Which also makes me think about this morning's caller: why is W32/Ska still spreading in 2001? Will people never learn?

Then, of course, there was Y2K and the disaster that never materialised. I received a call from a tired-sounding network engineer based in Manchester, England. He was one of many techies world-wide conscripted to work in dark server rooms over the New Year. Having sat for four hours watching his organisation's servers not crash, he called us to ask about any new threats we'd found that evening. Everything was quiet, conversational topics dried up. His closing question:

Techie: What's the weather like with you?

Me: Well, it's dark and cold, it might even be raining.

Techie: [Long pause.] Cool. Thanks. See you, then.

Turns out he'd been confined to his vault-like server room until sunrise, with only a bottle of Chardonnay and a pack of Christmas crackers for company. The life of a techie knows no boundaries.

Remember this if and when you call technical support: we didn't write the virus, we're trying to help you get rid of it, and help you out by advising you how to ensure you never get it again.

COMPARATIVE REVIEW

Smash and Grab?

Matt Ham

Another month, another platform – and after the relatively meagre installation-base of *Windows ME* we are off to much more business-relevant climes in this *Windows 2000* Comparative Review. As was suggested in the previous test, new operating systems tend to play havoc with previously stable parts of anti-virus software, historically especially when floppy access has been considered.

Windows ME showed this to a very minor extent, no more than could be expected in any Comparative Review in fact – whereas affairs were not so pleasant on this occasion. The exact nature of these problems will be unveiled; so on with the preamble. There were also a number of errors and features firmly placeable in the 'bizarre' category, which will also be exposed in due course.

The Test-sets

The Comparative tests were performed on the standard *Virus Bulletin* test-sets, with the ItW samples aligned with the February 2001 WildList. There was a good deal of inquisitiveness, both in personal mail and in last month's Letters pages, concerning additional samples which might or might not be added to the test-sets. These were centred on the question of whether files of a more-or-less Trojan nature and which are dropped by ItW viruses, should be included in *VB's* ItW test-set.

An example of this type of file is the .EXE file which could at one time be downloaded by JS/Unicle or the modified AUTOEXEC.BAT files produced by the O97M/Cybernet.A virus, a newcomer to the ItW set on this occasion. To clarify the matter, our test-sets will include such files only as part of the Standard test-set, and even then may not be included in the final test results. The JS/Unicle-associated .EXE file was in this state for some months and the .INI file produced by W32/MTX has also been tested against but never included in results. Files included in the ItW test-set will only include those files which are a part of the infectious capability of the viruses in question, rather than those which are associated non-viral malware or ephemeral non-viral helper files.

Aladdin eSafe Desktop v3.0

ItW Overall	100.0%	Macro	98.8%
ItW Overall (o/a)	99.5%	Standard	98.8%
ItW File	100.0%	Polymorphic	94.5%

The Israeli product *eSafe Desktop* was the first to fall victim to the woes of floppy disk scanning on-access, with Michelangelo specifically the culprit. As this appears to be a non-formatted floppy to the watchful eye of *Windows 2000*, the operating system appears to pre-empt the on-access scanner with its declaration that the disk is not formatted and should be scanned.

This was seen when *Windows NT* entered the picture and *Windows* became more convoluted in the way that disk changes and contents were determined. Other than this, the detection rates were an improvement yet again, with, for the Wild set, only the extensionless version of O97M/Tristate.C being missed on-access.

Only one major barrier remained for the gaining of *eSafe's* VB 100% award if these two misses are dealt with and that was the 30 false positives during the Clean set scanning test. Perhaps as a result of the heuristic processes giving rise to the false positives, the scanning rate of the clean files was somewhat slower than the rest of the products tested.

Alwil AVAST32 v3.0.321.0

ItW Overall	99.4%	Macro	99.2%
ItW Overall (o/a)	90.5%	Standard	98.2%
ItW File	99.4%	Polymorphic	95.7%

As ever, the complications concerning *AVAST32* were primarily centred around on-access scanning in particular and the configuration of scanning in general. The *AVAST32* scanning configuration is, at the very best, labyrinthine in its control methods, making it a simple matter to set one small feature in the wrong manner and thus make results non-existent, unusable or in some other way awkward.

The on-access scans in the end resulted in a slightly less than stellar detection rate, especially on the .VBS files which were not registered when scanned on-access. There was also the recurrence of a hang while processing the on-access false positives testing on the OLE2 file set.

Further investigations failed to show any problems with the files which are apparently being scanned when the hang occurs, or indeed those directly before and after in the test-set, so this can be considered an on-going mystery, hopefully to be solved in time for the next review. The on-access problems ignored, however, the detection rate was good for all areas and on-demand scanning against the ItW test-set showed a 100% detection rate.

CA InoculateIT v4.53

ItW Overall	100.0%	Macro	100.0%
ItW Overall (o/a)	100.0%	Standard	100.0%
ItW File	100.0%	Polymorphic	98.8%

On-demand tests	ItW Boot		ItW File		ItW Overall	Macro		Polymorphic		Standard	
	Number	%	Number	%	%	Number	%	Number	%	Number	%
Aladdin eSafe Desktop	0	100.00%	0	100.00%	100.00%	49	98.82%	52	94.59%	21	98.89%
Alwil AVAST32	0	100.00%	2	99.45%	99.49%	30	99.23%	27	95.74%	23	98.21%
CA InoculateIT	0	100.00%	0	100.00%	100.00%	0	100.00%	9	98.87%	0	100.00%
CA Vet Anti-Virus	0	100.00%	0	100.00%	100.00%	0	100.00%	268	93.73%	0	100.00%
Command AntiVirus	0	100.00%	0	100.00%	100.00%	0	100.00%	1	99.98%	6	99.71%
DialogueScience DrWeb	0	100.00%	0	100.00%	100.00%	0	100.00%	2	99.99%	0	100.00%
Eset NOD32	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%	3	99.97%	1	99.81%
F-Secure Anti-Virus	0	100.00%	0	100.00%	100.00%	0	100.00%	0	100.00%	21	99.71%
GDATA AntiVirusKIt	0	100.00%	0	100.00%	100.00%	0	100.00%	0	100.00%	0	100.00%
GeCAD RAV	0	100.00%	1	99.77%	99.79%	4	99.90%	0	100.00%	1	99.90%
Grisoft AVG	0	100.00%	2	99.54%	99.58%	16	99.58%	124	92.01%	37	98.28%
Kaspersky Lab KAV	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%	19	97.86%	0	100.00%
Norman Virus Control	0	100.00%	0	100.00%	100.00%	1	99.97%	528	94.70%	32	98.74%
Panda AntiVirus Platinum	0	100.00%	0	100.00%	100.00%	6	99.83%	512	92.78%	19	99.51%
Sophos Anti-Virus	0	100.00%	0	100.00%	100.00%	28	99.38%	191	95.24%	37	99.15%
Symantec NAV	0	100.00%	0	100.00%	100.00%	13	99.62%	0	100.00%	14	99.85%
VirusBuster VirusBuster	0	100.00%	5	99.54%	99.58%	27	99.30%	6	99.24%	5	99.81%

The perils of the *InoculateIT* patching process were by far the most complex and irritating part of the whole of its testing process. A complex procedure at best, this was enhanced in the last Comparative by the wrong required patch list being supplied by *CA* for the *Windows ME* Comparative.

This time the test was performed with a new and extended set of patches in place, dispensing with February's Michelangelo detection problems and allowing *InoculateIT* to be the recipient of yet another VB 100% award, the first of many in this first ever *Windows 2000* Review. The changing of patches also removed the designation of all .VBS files as 'viral' – surely a good thing.

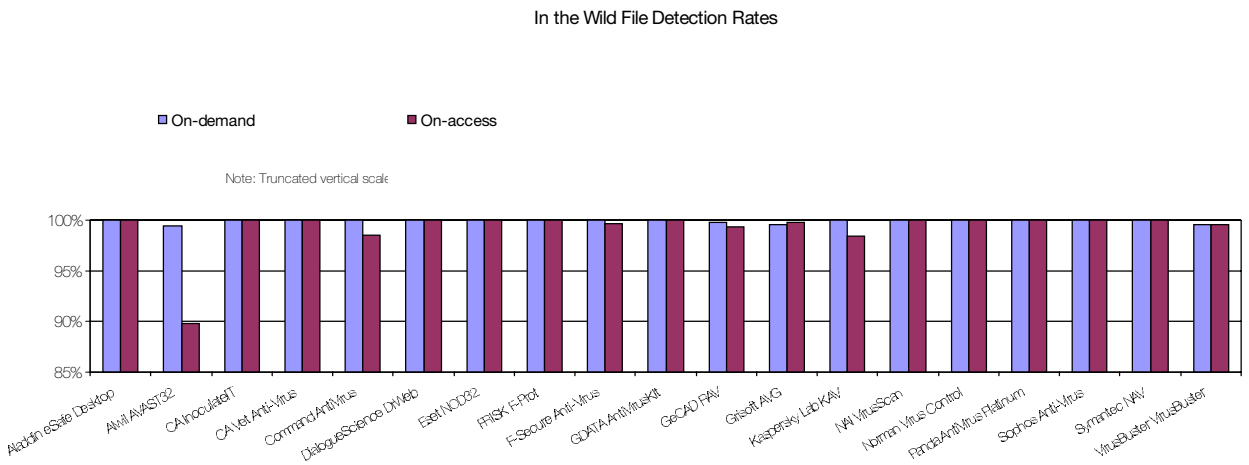
The only misses were in the Polymorphic set where the culprit was W95/Sk.8044. This has proved to be a stumbling block for many, with 11 out of the 19 products in this review having partial or no detection of this virus.

CA Vet Anti-Virus v10.2.10.0

ItW Overall	100.0%	Macro	100.0%
ItW Overall (o/a)	100.0%	Standard	100.0%
ItW File	100.0%	Polymorphic	93.7%

With another VB 100% to Vet's name, this has been another good month for *Computer Associates*. The main difference in performance here remains the Polymorphic set, where *InoculateIT* has the upper hand. *Vet* had misses in the cases of ACG.B, W95/Sk.8044 and W95/Sk.9972, all of them in the category of common misses across the board.

It is good to note, however, that their fellow polymorphic virus ACG.A is now becoming more universally detected rather than being in the same set of commonly missed viruses. Other than these, all files were detected both on-access and on-demand.



CA Vet and *InoculateIT* also showed a peculiarity in the scanning of the compressed OLE files for the Clean set – also shared by a number of other products. This is that the data throughput is faster for files which are compressed rather than uncompressed. Since the sizes used for the calculation of data throughput are uncompressed, this is doubly odd. It is possibly explained by the massively upgraded *VB* test machines used in the last two tests. With added memory and processing power, the manipulation of data may no longer be the limiting factor on these files, but rather the raw size which influences the rate at which data can be extracted from the hard disk. This would mean that uncompressed size is less important than compressed, thus giving the seemingly impossible results seen in the tests.

Command AntiVirus v4.61.0

ItW Overall	100.0%	Macro	100.0%
ItW Overall (o/a)	98.5%	Standard	100.0%
ItW File	100.0%	Polymorphic	99.9%

One of three products in this review using the *FRISK* engine, this offering was notable for the differences between the detection performance on-access and on-demand. This is not an infrequent occurrence admittedly, though in this case the differences were seen almost exclusively in .COM files due to an engine error.

This oddness aside, detection rates were good, with only a single miss of ACG.A in the polymorphics and a smattering of Bat/911 and several viruses in the Standard set against *CAV's* good name. This was the first product in this review which, although demonstrating full detection of floppies on-access, was definitely affected unhappily by *Windows 2000*. Change detection was poor and in some cases could only be triggered by alternating disks between the standard floppy and LS120 on the test machines.

A final comment must be made concerning the lethargic initialistaion of *Command AntiVirus's* on-demand scanner which certainly led me to think that it had crashed the first time a scan was attempted.

DialogueScience DrWeb v4.23

ItW Overall	100.0%	Macro	100.0%
ItW Overall (o/a)	99.5%	Standard	100.0%
ItW File	100.0%	Polymorphic	99.9%

The disk problems continued with *DrWeb*, and they were sufficient to deny the product a VB 100% due to the missing of Michelangelo on-access. A minor difference between this and previous results showed in two samples of the ageing polymorphic virus PeaceKeeper.B also being missed. This glitch in detection is possibly a result of a drive to reduce false positives – now standing at the relatively low number of fifteen warnings.

The developers at *DialogueScience* will no doubt be disappointed by the very narrow margin by which a VB 100% award was lost, although to be fair, the problems with *Windows 2000* are not likely to manifest themselves on any other current platform.

Eset NOD32 v1.70 NT

ItW Overall	100.0%	Macro	100.0%
ItW Overall (o/a)	100.0%	Standard	100.0%
ItW File	100.0%	Polymorphic	100.0%

This Slovakian product has gone from strength to strength, and after a momentary absence from the VB 100% holders list *NOD32* is once more a worthy recipient.

Again, all files in all sets were detected, new families which were added to the Macro test set proved no problem here. With speed tests as well as detection results being favourable, there is little to add but congratulations.

FRISK F-Prot for Windows v3.09

ItW Overall	100.0%	Macro	100.0%
ItW Overall (o/a)	100.0%	Standard	99.8%
ItW File	100.0%	Polymorphic	99.9%

On-access tests	ItW Boot		ItW File		ItW Overall	Macro		Polymorphic		Standard	
	Number	%	Number	%	%	Number	%	Number	%	Number	%
Aladdin eSafe Desktop	1	94.44%	1	99.96%	99.54%	47	98.93%	52	94.59%	22	98.85%
Alwil AVAST32	0	100.00%	39	89.79%	90.56%	37	99.12%	28	95.36%	60	94.18%
CA InoculateIT	0	100.00%	0	100.00%	100.00%	0	100.00%	9	98.87%	0	100.00%
CA Vet Anti-Virus	0	100.00%	0	100.00%	100.00%	0	100.00%	268	93.73%	0	100.00%
Command AntiVirus	0	100.00%	8	98.46%	98.57%	0	100.00%	161	96.55%	172	88.56%
DialogueScience DrWeb	1	94.44%	0	100.00%	99.58%	0	100.00%	2	99.99%	0	100.00%
Eset NOD32	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%	20	97.84%	1	99.81%
F-Secure Anti-Virus	0	100.00%	4	99.63%	99.66%	0	100.00%	0	100.00%	22	99.68%
GDATA AntiVirusKIt	18	0.00%	0	100.00%	92.37%	0	100.00%	0	100.00%	0	100.00%
GeCAD RAV	1	94.44%	5	99.31%	98.94%	4	99.90%	0	100.00%	1	99.90%
Grisoft AVG	0	100.00%	2	99.73%	99.75%	22	99.48%	292	89.47%	53	96.82%
Kaspersky Lab KAV	0	100.00%	5	98.39%	98.52%	0	100.00%	0	100.00%	3	99.71%
NAI VirusScan	0	100.00%	0	100.00%	100.00%	0	100.00%	19	97.86%	1	99.96%
Norman Virus Control	0	100.00%	0	100.00%	100.00%	1	99.97%	528	94.70%	32	98.74%
Panda AntiVirus Platinum	0	100.00%	0	100.00%	100.00%	6	99.83%	1012	90.14%	19	99.51%
Sophos Anti-Virus	0	100.00%	0	100.00%	100.00%	28	99.38%	191	95.24%	37	99.15%
Symantec NAV	0	100.00%	0	100.00%	100.00%	13	99.62%	0	100.00%	14	99.85%
VirusBuster VirusBuster	1	94.44%	5	99.54%	99.15%	27	99.30%	6	99.24%	5	99.81%

Traditionally strong against the Macro test-sets, *F-Prot* lived up to its reputation with a 100% detection here – add to this full detection in the wild, and it becomes the recipient of a VB 100% award, another in the growing list this month.

A slightly lower detection on-access was made up for, at least in the eyes of a reviewer, by the ease of use of the scanner – particularly for the scanning of floppies both on-access and on-demand. The extra misses came from the ever problematical W95/SK.8044, various polymorphics and VBS/Verlor.F.

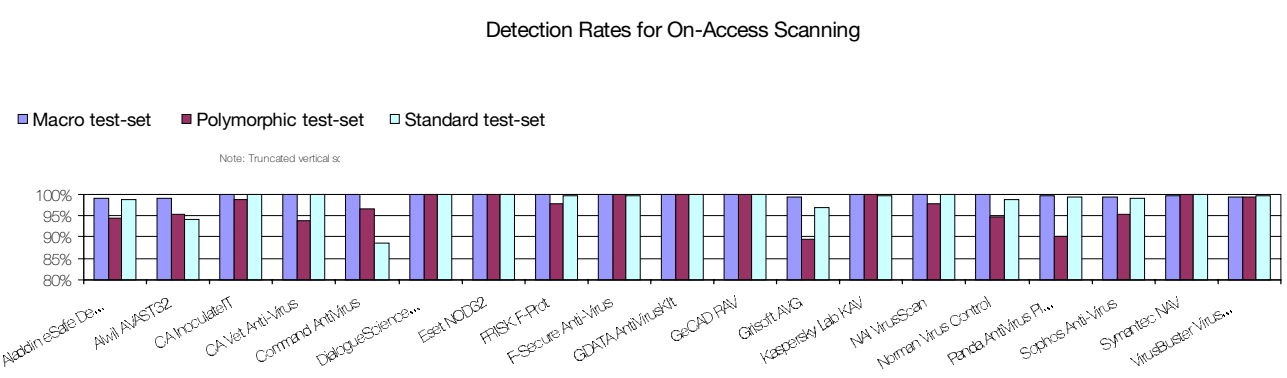
The *F-Prot* engine is also used by both *Command* and *F-Secure* in their products, and the speed ratings are fairly close between *F-Prot* and *Command AntiVirus*, with *F-Secure's* offering being notably slower. Detection-wise, however, *FRISK's F-Prot* is better than the other pair, as might be expected the engine's original development team.

F-Secure Anti-Virus v5.22 build 7072

ItW Overall	100.0%	Macro	100.0%
ItW Overall (o/a)	99.6%	Standard	99.7%
ItW File	100.0%	Polymorphic	100.0%

With talk of *F-Prot* in mind we move on to *F-Secure Anti-Virus* (*FSAV*). Although not really relevant here, the lower speeds seen in this product are possibly a result of the more network-integrated nature of *FSAV*, which results in many more options being built into the engine and the provision of HTML reports for cross-platform viewing. These are at least convertible to text format for analysis. They cannot, however, be held responsible for the rather long time taken to initialise the program.

FSAV also showed some problems on the on-access boot tests, though not enough to deny it full detection. The big disappointment will be the missing on-access of the



W32/MTX .DLL sample in the Standard and ItW sets, which removed a VB 100% award from *F-Secure's* grasp.

GDATA AntiVirusKit v10.0.1.0

ItW Overall	100.0%	Macro	100.0%
ItW Overall (o/a)	92.3%	Standard	100.0%
ItW File	100.0%	Polymorphic	100.0%

AntiVirusKit (*AVK*) is a relative newcomer to the *VB* test ranks and, after initial hiccups, has shown itself a worthy product. A sticky start on the on-demand floppy tests did not bode well for *AVK* on this occasion, though after many attempts the full set was detected, and no detection was possible on-access by design. This was in marked comparison with the tests on file viruses, since all other tests showed a full detection rate.

With such a good detection rate elsewhere, the floppy detection is something of a disappointment and denies *AVK* its first VB 100% award.

GeCAD RAV v8.2.1.4

ItW Overall	99.7%	Macro	99.9%
ItW Overall (o/a)	98.9%	Standard	99.9%
ItW File	99.7%	Polymorphic	100.0%

RAV was home, in this test, to perhaps the most bizarre of the idiosyncrasies seen in *VB* testing for a long while. Files on-access were at first impossible to scan at all, despite all being well on the installation front and the ability to detect the *EICAR AV* test file without problems. The test-sets were shuffled, moved and retested several times to no avail. In a moment of inspiration it was realised that the only difference between the *EICAR* files and the test files was that the test files were read-only. Sure enough, removing the read-only status of the files allowed testing to progress normally.

After such a mysterious start to the process the subsequent results were more prosaic. *RAV* missed Michelangelo on-access and suffered poor on-access floppy change detection. It also threw up four false positives and thirteen suspicious files in the Clean test-set. Admittedly, detection rates were actually towards the top end of the scale, though were let down by the numerous small problems seen.

Grisoft AVG v6.0.236

ItW Overall	99.5%	Macro	99.5%
ItW Overall (o/a)	99.7%	Standard	98.2%
ItW File	99.5%	Polymorphic	92.0%

AVG missed out on full detection ItW by dint of missing both JS/Unicle and O97M/Tristate.C, though the results were different on-access and on-demand to quite some degree. This difference was apparent across the test-sets, with the on-access scanner failing to detect a fair few more viruses than its on-demand counterpart. Most of these were polymorphs of the families already mentioned several times, to which were added misses in the Standard set which were unique to *AVG*.

The misses in the ItW set are small and should be relatively easily rectified, though the less important but more pronounced problems in the Polymorphic test-sets could be more complex to sort out. One area where *AVG* does shine, however, is in the aspect of speed, with OLE files being particularly fast. There are still false positives in the Clean set scan which is less speedy, perhaps due to the heuristics which cause the false positives.

Kaspersky Lab KAV v3.5.133.0

ItW Overall	100.0%	Macro	100.0%
ItW Overall (o/a)	98.5%	Standard	100.0%
ItW File	100.0%	Polymorphic	100.0%

Kaspersky AntiVirus (*KAV*) has suffered recently in the *VB* tests due to the spawning of new virus types with associated new file extensions. This month saw no new additions to the test-set as far as extensions were concerned, and sure enough the infected files were detected in their entirety during on-demand scanning.

The *KAV* engine has traditionally behaved identically on-access and on-demand, thus on-access results would be expected to be the same as those for on-demand.

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)
Aladdin eSafe Desktop	1847	296119	30	21	3777798		1126	141578	37	2016419
Alwil AVAST32	114	4797651		N/A	N/A		91	1751831	24	3108646
CA InoculateIT	92	5944915		15	5288918		51	3125815	12	6217291
CA Vet Anti-Virus	247	2214300		20	3966688		86	1853681	15	4973833
Command AntiVirus	154	3551508		18	4407432		59	2701976	20	3730375
DialogueScience DrWeb	275	1988844	[15]	26	3051299		121	1317492	21	3552738
Eset NOD32	104	5258963		14	5666698		86	1853681	28	2664553
FRISK F-Prot	189	2893821		17	4666692		102	1562908	46	1621902
F-Secure Anti-Virus	494	1107150		26	3051299		364	437958	65	1147808
GDATA AntiVirusKIt	216	2532093		35	2266679		108	1476080	37	2016419
GeCAD RAV	664	823693	4 [13]	15	5288918		396	402567	11	6782500
Grisoft AVG	217	2520425	4 [2]	14	5666698		90	1771295	14	5329107
Kaspersky Lab KAV	145	3771946		21	3777798		95	1678069	25	2984300
NAI VirusScan	295	1854007		29	2735647		81	1968106	21	3552738
Norman Virus Control	287	1905687		18	4407432		159	1002620	20	3730375
Panda AntiVirus Platinum	211	2592096		14	5666698		76	2097587	10	7460750
Sophos Anti-Virus	125	4375457		19	4175461		56	2846725	13	5739038
Symantec NAV	250	2187729		29	2735647		112	1423362	30	2486917
VirusBuster VirusBuster	223	2452611		17	2452611	[1]	139	1146882	22	3391250

Unfortunately, however, this was not to be. The on-access scanner for *KAV* now contains an option to activate the scanning of packed files, not activated by default. This is required for the detection of some files ItW and was sufficient to remove both full on-access detection and a VB 100% award from the *Kaspersky Labs* trophy cabinet.

NAI VirusScan v4.5.0.534

ItW Overall	100.0%	Macro	100.0%
ItW Overall (o/a)	100.0%	Standard	100.0%
ItW File	100.0%	Polymorphic	97.8%

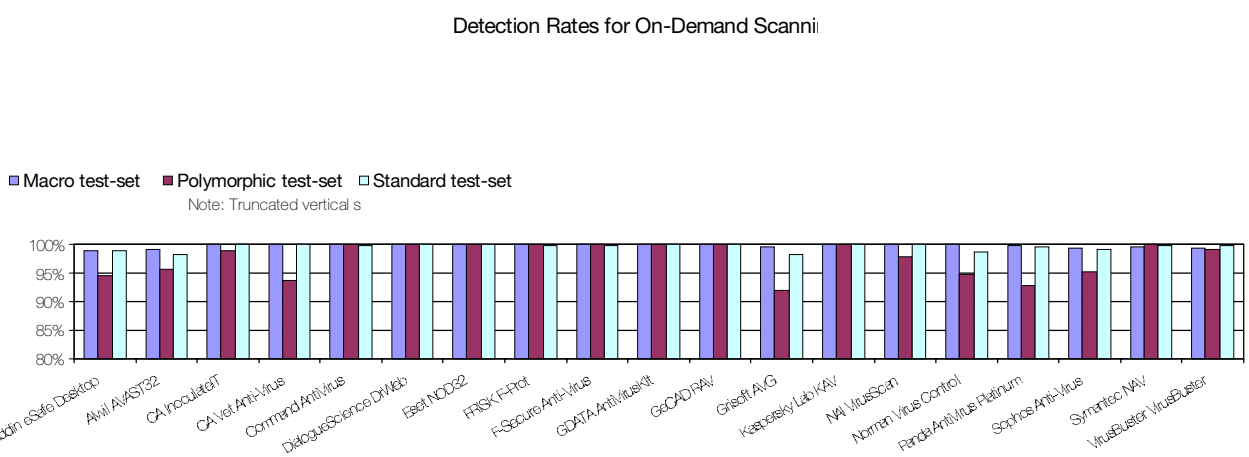
After vituperative words for the team at *NAI* for the last two Comparative Reviews, the tone is this time somewhat mellowed. Starting with old woes, *VirusScan* can still be convinced to deactivate its on-access scanner by one of the files in the test-set which continues to invoke ire in the *VB* test labs. The speed problems and instability are, however, a thing of the past, though possibly partially due to the increased power of the test machines. Floppy detection was an easy and pleasant affair and altogether the gaining of a VB 100% award by *VirusScan* is a well-deserved prize for recent improvements.

It should be noted that testing was performed with Service Packs and patches applied – one patch of the set being that which permanently activates the scanning of all files.

Norman Virus Control v5.0

ItW Overall	100.0%	Macro	99.9%
ItW Overall (o/a)	100.0%	Standard	98.7%
ItW File	100.0%	Polymorphic	94.7%

One of the more unusual programs to run in the Comparative, *Norman Virus Control's* (*NVC*) detection rate has been much improved since the inception of its newest scanning engine. The



interface problems encountered in the last review were markedly less apparent on this second encounter – some helpful prods from the developers and added familiarity making the whole affair much more pleasant.

Detection results were identical on-access and on-demand, with full detection on both swelling this month's bumper crop of VB 100% awards. *NVC's* main weakness is in the Polymorphic sets where Digi.3547, W95/Sk8044 and a sprinkling of Sepultura:MtE-Small were the culprits.

The product's polymorphic detection percentages have, however, improved markedly. *Norman* will, perhaps, be looking for further improvements as the year progresses.

Panda AntiVirus Platinum v6.23.00

ItW Overall	100.0%	Macro	99.8%
ItW Overall (o/a)	100.0%	Standard	99.5%
ItW File	100.0%	Polymorphic	92.7%

Panda AntiVirus (*PAV*) *Platinum* is another product which suffers far greater weakness against the Polymorphic sets than in other areas, though this is more apparent on-access, the number of misses roughly doubling when the detection method is changed. These misses were scattered throughout a number of samples in the Polymorphic sets, with partial detection being more common than no detection at all.

The polymorphics were the only problems, *PAV's* otherwise solid performance being sufficient to reap it a well-deserved VB 100% award. A special mention should also be made of the speed with which *Panda AntiVirus* was able to scan OLE files which was the fastest in the pack whether the files were zipped or not.

Sophos Anti-Virus v3.43

ItW Overall	100.0%	Macro	99.3%
ItW Overall (o/a)	100.0%	Standard	99.1%
ItW File	100.0%	Polymorphic	95.2%

This was a far happier outing for the *Sophos* product than the last two *Windows* Comparatives, where the new viral extensions caused some misery. Detection has improved against the Polymorphic set and more gratifying will be the arrival of a VB 100% for the complete detection both on-access and on-demand.

As has been customary in past tests the detection was identical on-access and on-demand. There was a momentary scare when the on-access scanner was added and tests showed extra misses, but this was tracked down to the purging of temporary virus identities when the product is upgraded. The reasoning behind this is clear – these are not required when the product is properly upgraded, but perhaps more warning would be appropriate.

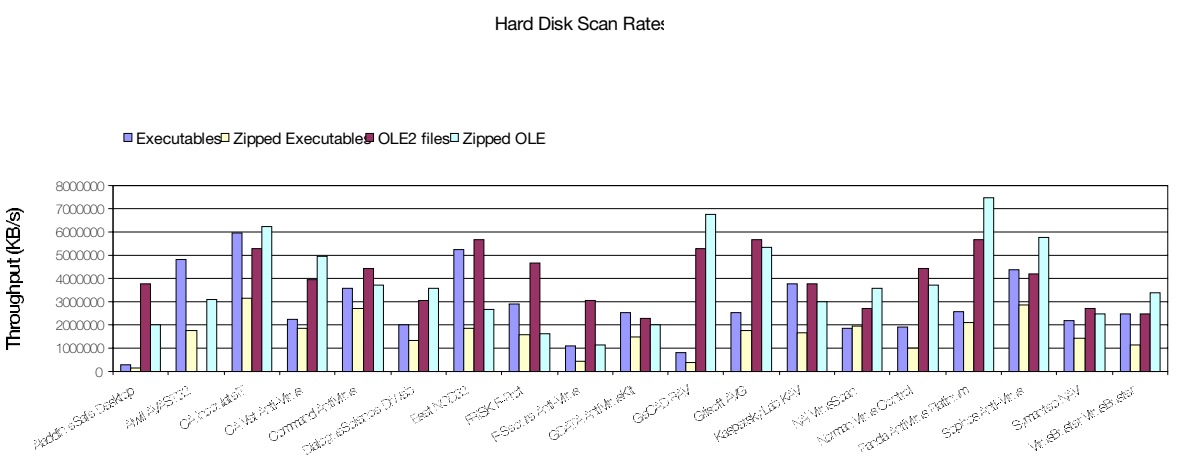
Symantec NAV Corporate Edition v7.50.846

ItW Overall	100.0%	Macro	99.6%
ItW Overall (o/a)	100.0%	Standard	99.8%
ItW File	100.0%	Polymorphic	100.0%

Approaching the final entries in this Comparative I can once more enter 'rant' mode. On-demand scanning again caused *NAV's* scanner to lock up – difficult to tell since the initiation process for any activity seemed interminable.

The post-scan reports were the sticking point, being the moment at which crashes occurred, but more oddly they could be exported only in comma-separated or .MDB format, rather than the plain text equivalent .TXT file which might be expected. All these combined to force detection by deletion.

Floppy scanning was also nightmarish, the process of beginning a scan taking up to 20 seconds to reach the scanning process, and with poor change detection this was the cause of many an unpleasant curse. Despite all these problems detection was at NAV's usual high rate, earning a VB 100% award with only a scattering of misses in the Macro and Standard test-sets.



VirusBuster VirusBuster v3.03

ItW Overall	99.5%	Macro	99.3%
ItW Overall (o/a)	99.1%	Standard	99.8%
ItW File	99.5%	Polymorphic	99.2%

Last but not least this month is *VirusBuster*, which suffered as others from the curse of Michelangelo, resulting in a miss for on-access floppy scanning. The February 2001 WildList was also responsible for misses, the samples of Win95/Caw.1262 being undetected in the ItW File test-set. This will be very much a frustration for the *VirusBuster* development team, which has missed a VB 100% by similar slim margins for several months now. With false positives down to a scant single warning and the speed still good, the future should hold better news.

Conclusion

The new platform made several differences to the results of this testing, the new test-sets made less impact – both bear some further examination. What is at first glance contrary to common sense and in opposition to the results of the *Windows ME* test can, in fact, be seen to agree with both these methods of reasoning.

The changes from *Windows NT* to *Windows 2000* are certainly more all-pervading than the changes within the *Windows 95/98/ME* product line, and where file access is affected, anti-virus products will always be impacted. Failure to detect Michelangelo when the operating system is claiming that nothing exists which should be scanned is something which could perhaps be expected more often than was seen in the tests here.

It is doubly compounded by this being only testable on real floppies, when much QA is done on disk images where detection is much simpler. Nor is it necessarily a sign of technical laxness – one anonymous developer stated that his product was only saved from not detecting it because it performed detection in 'not a very clever way.'

As for the matter of the test-sets – while *VB* does intend to continue the use of the Standard test-set, where the so-called Old Fashioned File Viruses mostly reside – most additions are made to the Macro and ItW test-sets.

Of these, the macro test-set is a category in its own right and one which is in general a game of catch-up with the virus writers, with regard to the volume of viruses written rather than complexity. Some major opportunities for mass failures in detection do exist but these are most often concerned with changes in *Office* imposed by *Microsoft*. This is not notably the case at the moment and so the impact on the detection rates is small.

This leaves the ItW set, by its nature a catch-all, where scanning results are likely to fluctuate as test-sets are changed. This is again only under certain circumstances, one of which at least is the introduction of new Operating System features.

The most likely circumstance, however, is currently the addition of new file extensions for scanning, which in this edition of the test-sets turned out not to be the case. So the factors this month seemingly acted to lessen detection failures due to changes in test-set and heighten those due to Operating System.

So much for the discussion, but what does the future hold? The likelihood of new virus types being the major impact upon detection is intrinsically tied in with the new Operating System features which make these viruses possible. So, in the case of *Windows*, the developers can only watch and wait as *Microsoft* expands the capacity for disaster within its various incarnations.

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 *Windows 2000*. 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/NT/2001/02test_sets.html. A complete 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

InfoSec 2001, Europe's largest IT security event, is to take place from 24–26 April 2001 in the National Hall, Olympia, London, UK. See the Web site http://www.infosec.co.uk, or find out more about the event by emailing infosecurity@reedexpo.co.uk.

Palm OS is the subject of many of the larger anti-virus companies' attention this month, with *Symantec*, *McAfee* and *F-Secure Corporation* all issuing press releases about their latest AV products specifically for handheld devices. For more information, see the individual Web sites at http://www.symantec.com/nai.com/ F-secure.com/.

iSEC Asia 2001 is to be held at the Singapore International Convention and Exhibition Centre from 25–27 April 2001. The conference and exhibition covers IT security topics from anti-virus through encryption to biometrics and digital signatures. For more information and a booking form contact Stella Tan; Tel +65 322 2756 or email stella@aic-asia.com.

Symantec* has released its *Personal Firewall 2001 v3.0 which, the company claims, provides users with out-of-the-box protection enhanced with new intrusion prevention capabilities. For details about pricing and availability (currently, the estimated price is £29.99), visit the Web site http:www.symantec.com or buy from the on-line store at http://wwwsymantecstore.com/.

InfoSec Paris 2001, the 15th information systems and communications security exhibition and conference, will take place at CNIT, Paris-La Défense, France from 29–31 May 2001. Companies wishing to participate in the exhibition are encouraged to contact the organisers; Tel +33 0144 537220, or email salons@mci-salons.fr.

***Sybari Software* has announced its upcoming support for *Microsoft's Exchange 2000* Virus Scanning Application Program Interface (VS API 2.0).** As *Microsoft's* partner, *Sybari* plans to implement support for the new VS API 2.0 into its AV product – *Antigen for Exchange*. For more details about the planned combination of technologies, visit http://www.sybari.com/.

For an at-a-glance look at which anti-virus products have been winning VB 100% awards for the past few months, visit the Virus Bulletin Web site http://www.virusbtn.com/100/.

Linux Expo 2001 Exhibition & Conference is to take place at Olympia, London in the UK from 4–7 July 2001. To find out about exhibition opportunities or to register for the show, email the organisers jonathan.neastie@itevents.co.uk or visit the conference Web site http://www.itevents.co.uk/.

iSEC Australia will take place in Halls 5 & 6 of the Sydney Convention & Exhibition Centre from 6–8 August 2001. For information on how you can be a sponsor, exhibitor or delegate, visit the Web site http://www.isecworldwide.com/isec_aus2001/. Alternatively contact Chris Rodrigues; Tel +61 2 9210 5756.

***Kaspersky Labs* has announced the release of the beta version of what it calls the world's first virus protection software for Postfix email gateways.** Postfix, an alternative to the popular Sendmail program, has recently become popular amongst Unix users. *KAV for Postfix* acts as an email filtering system and it is scheduled for commercial release at the end of March 2001. Plans are to distribute it as part of *KAV for Linux Server* and *KAV for FreeBSD/BSDi*. For further information, contact Denis Zenkin; Tel +7 095 7978700, email denis@kaspersky.com or visit http://www.kaspersky.com/.

***Sophos Anti-Virus* has published its new Safe Computing Guidelines.** There are two versions available; one for network administrators and another for everyday PC users. Both can be downloaded free of charge from http:www.sophos.com/virusinfo/articles/safehex.html/. For more details, contact Natasha Staley; Tel +44 1235 544160 or email natasha.staley@sophos.com.

***Mirapoint* has launched its new flagship gateway product *Message Director*.** *Message Director* offers fully integrated anti-virus, anti-spam and content filtering capabilities and is allegedly impossible to hack. For more information about the product or the company, contact Susannah Butt; Tel +44 207 357 7799.

F-Secure Corporation* has announced the release of *F-Secure Anti-Virus for Firewalls on Linux which offers detection and disinfection of Internet-borne viruses and malicious code passing through OPSEC CVP firewalls. The product is available now for *Red Hat Linux v6.1* or higher. For pricing details, contact Sari Lindroos in Finland; Tel +358 9 2525 5623, email Sari.Lindroos@F-Secure.com or visit the Web site http://www.F-Secure.com/.