ISSN 0956-9979 NOVEMBER 2001

VIRUS BULLETIN

THE INTERNATIONAL PUBLICATION ON COMPUTER VIRUS PREVENTION, RECOGNITION AND REMOVAL

IN THIS ISSUE:

• **Feathered friends:** WM97/Ostrich.A raises some interesting issues relating to disinfection – in fact, says Gabor Szappanos, it is impossible to disinfect perfectly. At the risk of re-opening the disinfection debate, Gabor analyses this polymorph, starting on p.6.

• **Dear Santa:** Max Morris writes an administrator's wish list for the perfect AV solution. AV vendor elves should take note from p.10.

• **Looking through the Windows:** No less than 21 *Windows NT* products lined up for this month's comparative review. See p.16 to find out how they all fared.

CONTENTS

COMMENT

What's in a Name?

“ The compulsive obsessive drive to name and classify everything seems to be encoded in our DNA. ”

One of the many abilities we humans have acquired during the development and evolution of our linguistic skills is the ability to name things. Assigning names to other living creatures, objects, feelings, events and ideas not only enables efficient communication, but also gives us a chance to sort and classify all of the complex and complicated things in the world around us into something easier to comprehend, accept and follow. The compulsive obsessive drive to name and classify everything seems to be encoded in our DNA.

Of course, the diversity of 'stuff' surrounding us triggers the variety of names, lists, catalogues and directories. No one is able to learn and understand all the existing names and naming systems. Specialization is unavoidable – complex schemes can only be studied, learned and expanded on by relatively small groups of people who dedicate their time and skills to very particular and narrow fields. The present structures of all implemented names and classification systems have been shaped by the history of their creation and their development. It's not accidental that all biological naming schemes use Latin terms, while in computer science we rarely find any non-English names.

As our world and our knowledge of it evolves, so do the names and naming systems we use. Scientists regularly discover formerly unknown specimens and assign them names. There are rules and conventions to be followed by a researcher baptizing a new species so that the name will be accepted by the rest of the community. No written rules will guarantee that a name will be considered perfect by all, however there are rules which ensure that it will at least be acceptable.

Not surprisingly, many naming conventions use 'negative' rules – rather than defining what must be done, they clearly spell out all naming no-no's. A researcher assigning a new name to a species that is already known or using an old name to describe something new will quickly become the subject of ridicule and the target of justified anger. Correcting names is harder and takes much more effort than naming the things correctly in the first place (ever heard about Indians living in America?).

Discovering completely new families is rather rare, and new orders even rarer. A researcher who fails to match a new specimen to its obvious family and genus will certainly have his/her competency questioned (and rightly so). Avoidance of unreasonable creation of new entities is one of the main restrictions that prevents any naming system from overflowing with separate and unrelated classes and families and from reflecting the egos of those who, in selecting fancy names, seek publicity and a moment of fame.

If you're wondering what all this has to do with computer viruses, let me assure you that these are exactly the same problems we face every day while discovering and naming new viruses and Trojans. Assigning new viruses to the proper families, avoiding names that are already taken, inventing names for new viruses and Trojans, avoiding obscene and offensive terms – these are our daily problems. Additionally, many virus researchers have agreed to avoid using the names suggested by the virus authors or naming new malware by the name of the carrier file or by the message located in a virus body. The reasons behind this seem obvious to anyone with some experience in dealing with computer malware.

Currently the anti-virus industry finds itself under significant pressure to organize and integrate all computer malware names and naming schemes. The voices of users, and especially large corporate users, are forcing virus researchers to cooperate much more closely as far as malware naming is concerned. At the same time, the media jumps on any sensational announcements, making fixes to any mistakes almost impossible. That's why those who don't play according to the rules or those who make too many mistakes will become the target of angry attacks from the rest of the industry and excluded from important forums and initiatives.

Jakub Kaminski
Virus Bulletin Technical Editor

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

NEWS

Primeval Marketing

A week before this *VB* issue went to print, there arose a media flutter over an 'Anthrax' computer virus. Upon investigation, this proved to be the most cynical and tasteless marketing *VB* has reported for quite some time.

Originally, the 'VBS/Antrax' virus story appeared on an Argentinian AV news Web site which had been sent a hand-crafted email with a semi-functional VBS attachment. The virus was reported under the name its writer desired and its functionality seems to have been described by reading the code rather than analysing it properly – it was described as a mass mailer, yet the mass-mailing code did not work.

The story was picked up by the virus alert centre of the Spanish Ministry of Science and Technology, and its errors repeated in an alert posted on the Ministry's Web site. Spanish AV developer *Panda* obtained samples and forwarded one to REVS around the time it issued a press release about the virus. Despite the simplest of analyses showing that the virus' mass-mailing code cannot attach a copy of the virus, and despite clearly having been 'written' with the VBSWG kit, *Panda* stuck with the 'VBS/Antrax' name in its press release.

Shortly thereafter, complaints about the name arose on several industry mailing lists. The name was too similar to 'Anthrax' and there was already an unrelated virus family with that name. It was grossly insensitive to name a computer virus anything close to 'Anthrax' at the time. The virus was clearly a member of the VBS/VBSWG family. And so on…

Worse was to follow. Some vendors briefly used the name 'VBS/Anthrax' and/or listed it on their Web sites as an alias (despite the mass mailing not working, the sample *Panda* supplied replicated via IRC scripting mechanisms, so VBS/VBSWG.AF really is a virus). Somewhere in the middle of all this a journalist caught the whiff of a scoop and this nearly non-working and otherwise entirely uninteresting virus became a contender for the number three news item of the day.

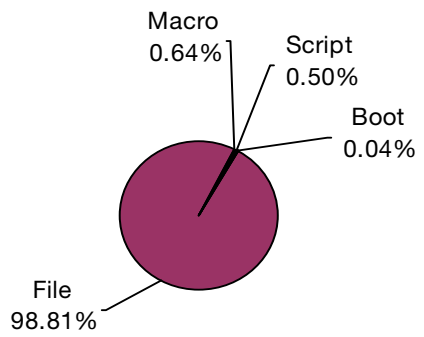
But the responsibility for the most shameful incident of the whole event lies with *SoftWin*, whose Web pages included screenshots of the email messages reputedly created by this virus. As the virus *cannot* create such email messages, those 'screenshots' must have been faked. One can only wonder why an AV developer would mock up false screenshots and include them in a description of the mass-mailing capabilities of a virus that clearly cannot send itself via email. The only guaranteed result of such forgery is to increase the FUD factor around the virus, which we presume might increase the likelihood of the gullible parting with a few more of their hard-earned shekels ▮

Prevalence Table – September 2001

Virus	Type	Incidents	Reports
Win32/SirCam	File	32990	85.3%
Win32/Nimda	File	2838	7.3%
Win32/Magistr	File	1013	2.6%
Win32/Hybris	File	636	1.6%
Win32/Apost	File	178	0.5%
Win32/MTX	File	153	0.4%
Win32/CodeRed II	File	98	0.3%
Laroux	Macro	89	0.2%
Win32/Funlove	File	74	0.2%
Haptime	Script	73	0.2%
Kak	Script	66	0.2%
Win32/BadTrans	File	59	0.2%
Win32/Cabanas	File	36	0.1%
Win32/QAZ	File	34	0.1%
VCX	Macro	33	0.1%
VBSWG	Script	32	0.1%
Divi	Macro	28	0.1%
Marker	Macro	28	0.1%
Solaris/Sadmind	File	19	0.0%
LoveLetter	Script	18	0.0%
Win32/Bymer	File	17	0.0%
Win32/Navidad	File	15	0.0%
Win32/Jerym	File	13	0.0%
Win32/Ska	File	13	0.0%
Melissa	Macro	12	0.0%
Win32/Pretty	File	10	0.0%
Others [1]		117	0.3%
Total		38692	100%

[1] The Prevalence Table includes a total of 117 reports across 47 further viruses. Readers are reminded that a complete listing is posted at http://www.virusbtn.com/Prevalence/.

Distribution of virus types in reports



LETTERS

Dear Virus Bulletin …

Tougher Sentencing

In September, while the Virus Bulletin 2001 Conference was underway in Prague, Jan de Wit, aka 'OnTheFly', was sentenced for his role in the writing and distribution of VBS/VBSWG.J@mm (popularly called AnnaKournikova) and the resulting damages of that action. Originally the prosecutor asked for 240 days of community service, but the suspect was sentenced to 150 hours, replaced by 75 days in jail in case of refusing the community service.

The sentencing is extremely light compared to the overall problems and damages caused by the virus. The reason for this light sentencing was the lack, or rather shortage, of evidence in the investigation. Only 55 instances, with a total damage value of USD 166,827 were registered.

It seems that those corporates that were hit by the virus refused to report their damage to the justice departments, probably fearing public exposure resulting in negative advertisement of their company. Combine this with the fact that the suspect in cases like this is usually an individual who will not, with his lifelong income, be able to pay for all the damage he created, and it can be seen how the threshold for not reporting damages is easily passed.

Although a vast number of countries have a computer crime act nowadays, I would like to encourage politicians to create laws that prohibit the public exposure of corporates (or individuals) reporting their infections and damages. Certain countries already use this model to protect those reporting possible criminal information to the Intelligence Services. It should be relatively easy to adjust or complement this system for the above-mentioned situations.

As long as we continue the current trend of not reporting the damages, kiddies like Jan de Wit will feel themselves as free as a bird and it will encourage them to try harder. And, even if they *are* caught, they will think, 'What's 150 hours of community service?' Let us all work together to present them with real sentences …

Righard Zwienenberg
Norman
The Netherlands

Charity Begins at Home …

Poor old Jan de Wit. The author of the Kournikova worm is appealing against his sentence for writing and distributing one of the most widespread computer viruses ever seen.

I feel sorry for him. It can't be easy finding the time when you're an unemployed Dutch virus writer to do a spot of gardening, wash a few cars, or shuttle old ladies back and forth from hospital.

I have a modest proposal. I would like to volunteer my services to the Dutch courts. I am prepared to do the community service for him as he's too busy.

That should give Jan the opportunity to follow up on the job offer that was made to him by Sieboldt Hartkamp, the mayor of Sneek, who described the virus as 'a joke' and expressed an interest in employing de Wit in the town's IT department.

Meanwhile, Melissa man David Smith is waiting for sentencing two years after pleading guilty. Is it just me or are the courts perhaps not taking virus-related crime seriously?

Graham Cluley
Sophos Anti-Virus
UK

Setting the Record Straight

I was not present at this year's *VB* conference and therefore not at the AVIEN presentation/panel session that made up part of the proceedings, but it seems to me that some issues need clarifying. I should stress that I do not speak for AVIEN in an official capacity, but Robert Vibert (who *is* mandated to act as official spokesperson) and a number of other active members have expressed their agreement with the following points and their willingness to be quoted as co-signatories. These include: Andi Lee, Paul Schmehl, Ken Bechtel, Joe Broyles, Jerry Isaacson, David Bass, Dave Phillips and Tom Bowers. However, I take full responsibility for any errors or misunderstandings contained herein.

A quote apparently attributed to *Sybari* in John Leyden's very competent piece in *The Register* suggested that it is 'customary' procedure for AV vendors to wait until they have analysed a threat and put a fix together before making an authoritative announcement. This encapsulated nicely exactly why virus management cannot be left to the discretion of the AV vendors. If we did so:

- Fast burners would have appreciably more time to spread unchecked.
- We'd still be waiting for some vendors to notice Code Red, or at least to accept that their customers expected them to address the problem.
- Many of the steps necessary to manage convergent, multiple vector threats like Code Red and Nimda would remain untaken because vendor advisories tend to fixate on the measures that conventional AV can handle comfortably, such as identifying and removing

backdoors. Unfortunately, handling outbreaks like these and the DDoS attacks of a year or two ago involve many issues that most AV products do not address at all.

Handling a new fast burner is not at all like reporting a vulnerability on BugTraq. Politely waiting until the vendor produces a fix when there are interim emergency measures that can be taken is not only unnecessary but irresponsible if it leaves an organization open to attack, and therefore at risk of becoming a source of infection in its own right.

I referred in a recent *VB* article (see *VB*, September 2001, p.15) to the issue of timely alerts as a trade-off between 'timely but not necessarily correct in every detail' and 'obsessively accurate'. AVIEN/EWS works because we can accommodate the first option so well by virtue of our combined experience and expertise.

Membership of EWS is not a substitute for the pool of experience and expertise offered by the best vendors and researchers, but a supplement. Information and advice shared there is not always the best possible advice, but sometimes it's all there is. It is reviewed as other information becomes available, and it is subject to the input of some very experienced individuals. EWS is an essential resource for people who can't wait for their vendor of choice to update their Web site.

There are a number of points worth making or reiterating in the wake of the panel session at VB 2001 and the subsequent media attention:

- It has been assumed in some quarters that AVIEN is in the business of beating up on vendors. It ain't. It's a self-help/mutual support group. It can also be a rather effective pressure group, but it doesn't exist to promote black propaganda of the type that flourishes on some of the lists where black, grey and white hats mingle and full disclosure roolz OK. We are anti-virus professionals who respect the work of other professionals and often work closely with them, but reserve the right to disagree with them.
- It is not AVIEN's job as an organization to snuggle up too cosily with the vendors. Members pay their vendor of choice for a service, and most of us, as professional AV administrators, have increasing influence on whose service we actually pay for, as well as the expertise to be rather specific about the service we require. There is no longer room for substituting what the vendors are comfortable with for what customers need. We don't necessarily expect them to furnish a complete security system – let's face it, we've learned not to expect even a complete anti-virus system. (Hands up any sysadmin who thinks part of their job description should include patching the holes their anti-virus solutions leave open? Yep, me too.) Competent security administration is rarely about buying an off-the-shelf solution, installing the default configuration and assuming the job is done. AV software is *certainly* no exception.
- The customer is not always right. But sometimes the customer knows better. Despite some of the recent media suggestions, AVIEN is not primarily a meeting place for the sort of high-level manager who talks in business-speak about HackingAndViruses as if it was all a single, simple issue. It does include some very able system administrators with considerable practical and strategic skills across a range of security areas. It also includes a number of genuine independent AV experts. Apart from our rather significant combined customer base, our membership has chalked up an impressive array of interviews, articles, conference papers, Internet resources, even books. Not to mention membership of industrial-strength professional AV organizations. Our collective CV would be pretty impressive, and we are not going to be told to go away and not worry our pretty little heads about it.
- I keep detecting this undercurrent of suggestion that AVIEN members exchange samples, which is exactly what we've gone to some pains to discourage. Members of AVIEN face the same issues of trust, responsibility, ethics and morality as anyone else when it comes to sharing samples between individuals, but they don't use AVIEN as a vehicle for exchange. In particular, anyone saying 'Could I have a sample of X? After all, I am a member of AVIEN' can expect very short shrift.
- Disinfection and disinfestation are not always the same as restoring the pre-infected environment, and they never have been. Some vendors are very good at supporting corporate victims in the throes of a cleanup, but automated disinfection is often a poor substitute for local knowledge, and sometimes does more harm than good. In such scenarios, access to a pool of vendor-independent expertise is not to be dismissed lightly.
- Vendors do have limited access to AVIEN, and we value their contributions in the appropriate lists. They are no more entitled access to all AVIEN lists than we are to CARO. There are plenty of alternative venues where vendors can interface with their potential and actual customers. We are entitled to exclude certain classes of vested interest from some areas, just as vendor secret squirrel lists are entitled to enforce their own selection criteria. Vendors *should* be concerned about getting to hear what we have to say. However, it would be more useful if they set up more effective feedback mechanisms themselves, rather than trying to gatecrash our party. They are not entitled to control the guestlist or the winelist, let alone flood us with suggestions for buying from their particular vineyard.

Virus management and research is not the exclusive property of the vendor community, and maybe they should be glad of it. We are not your enemy, but we're not going to shut up and go away. Live with it.

David Harley
NHS Information Authority,
UK

Unfair Comparisons

Dr. Igor Muttik's VB2001 paper 'Comparing the Comparatives' is a very good read, raising important issues in virus detection testing that should be of interest to a wide audience. The paper addresses some problematic aspects of detection testing, ranging from issues of test set size through testing the quality of generic and heuristic detection abilities, and suggests some possible solutions.

I have very few points of disagreement with the paper itself. Dr. Muttik's simulations of test results based on random subset selection of test sets turned up some very interesting – in fact, surprising – findings regarding 'unfair' test results. However, it was a comment during his presentation that spurred me to write this letter.

Accepting a test as fair if it rates all products in the correct order (relative to theoretical perfect tests), Dr. Muttik looked, amongst other things, at simulations of tests involving several products with small detection rate differences. These simulations showed that fair results should be expected about 98% of the time if 20 scanners are tested, so long as about 25% of all viruses are included in randomly selected test sets. Including fewer samples makes for many more unfair tests, but comparing fewer products with the same sampling rate improves things slightly.

Put another way, when comparing 20 scanners with modest detection differences, about 75% of test results were unfair when the test set comprised a random selection of about 6% of viruses. Bolstering the test sets to 10% of viruses but retaining all other factors, still just under 50% of tests were unfair. These conclusions are undeniable as mathematical results. However, during the presentation of his paper, Dr. Muttik's ad-lib comment that these just-repeated results reflect the *Virus Bulletin* standard test-set was not only unfair, but quite misleading.

VB's standard test set is, in fact, quite static. It is not a new subset randomly selected from all known viruses (of a certain type) for each test. Furthermore, in recent years its content has remained almost unchanged from test to test (e.g. while I was at *VB* [1997–99], very few viruses were added to this test set other than the non-macro file infectors that fell from the 'In the Wild' test-set). So, in reality, Dr. Muttik's analysis hardly applies to the *VB* test sets. The content of *VB's* test sets does not match the continual, random re-selection modelled in his simulations.

Further, in the real world, quality scanners are expected to maintain detection performance against 'old' viruses – that is, to 'remember' what they did yesterday and repeat it today and tomorrow. Good product development and internal testing practices by the AV developers should ensure scanners easily display such 'memory' and *VB's* tests are designed to (partially) test this.

Nick FitzGerald
Computer Virus Consulting Ltd
New Zealand

VIRUS ANALYSIS

Heads Stuck in the Sand

Gabor Szappanos
VirusBuster, Hungary

WM97/Ostrich is a potentially damaging parasitic polymorphic macro virus, of which there are at least two variants. One of these, Ostrich.B, has been a recent addition to the WildList. This analysis concentrates on Ostrich.A, since this variant highlights a number of interesting problems relating to disinfection. In fact, it is impossible to disinfect this virus perfectly, and even the restoration of a functionally equivalent original document is not possible in all cases. The problems arise when the original document (prior to infection) contains macro programs.

WM97/Ostrich.A

The virus body resides in the Document_Open macro, but Ostrich redirects the Document_Close macro to the virus code as well. The virus will activate whenever an infected document is opened or closed, and further documents will be infected upon opening or closing.

Once activated, the virus removes those menu items and command bar buttons that could provide access to the virus code. Since these items are referenced by their ID numbers, this payload will work in all language versions. The items removed are (listed by ID and name) 30017: Macro, 751: Templates and Add-ins, 797: Customize, 522: Options, 336: Protect document, 30045: Toolbars from the Word command bar and 930: Macros, 522: Options from the Visual Basic Editor Command bar.

In addition, the virus modifies the CodeBackColors and CodeForeColors settings so that all text in the VBE window is white on white – virtually invisible. (The settings are in HKEY_CURRENT_USER\Software\Microsoft\VBA\Office.)

Next, Ostrich reads back its version information (if present) from the registry key SpockVersionNumber in the section HKEY_LOCAL_MACHINE\SoftwareMicrosoft\Windows \CurrentVersion. This version number consists of three parts in the form 00/34/125, where the first part is unused (always 00), the second part is the generation counter, and the third part is the infection counter. The generation counter is incremented whenever a new computer is infected and the infection counter is incremented whenever a new document is infected – thus this number measures the length of the current infection

chain. Using these values, the virus will rename the Help|About menu item – in this case 'Mister Spock 0.34.125'.

Next, Ostrich tries to determine whether it is running from the global template or from an infected document. To do this it searches for an ID string in the global template. This ID string is the decoder table used by the virus, which is shuffled on each infection, therefore it is unlikely to be matched if the virus is not running from NORMAL.DOT.

Having determined the infection source and the target, the virus pre-processes the target document. This involves removing all lines starting with the OPTION EXPLICIT keyword. This keyword can only be placed in the general declaration area of any module, in front of any function or variable definitions; nevertheless, the virus will remove it from any position in the code module. The purpose of this operation is not clear since all the variables used by the virus have been explicitly declared (this plays an important role in the poly engine of the virus); leaving this option on would not interfere with the virus.

Infection Process

Next the virus attempts to determine whether the target has been infected already. It uses the aforementioned decoder table as an ID. This table is really a large string used for decoding string constants. Upon each infection it is shuffled. The virus uses the sum of the ASCII codes in this table for self-recognition, as this value will not change. If the target module contains a string constant declared by the Const keyword and having the checksum 18134, the target is considered already infected; otherwise the virus will infect it.

A bug in the virus means that it will only check the first such constant. Given the fact that the virus code is appended at the end of the Document_Open procedure, if the module to be infected contains code that has a Const string declaration, it will always be considered uninfected, and a new copy of the virus will be appended at the end of the procedure during each open and close operation – increasing the size of the module to practical infinity.

If the target document is found to be uninfected, the virus extracts its source, makes a polymorph transformation of it, and inserts the new code into the target. During the infection, the virus will remove all defined error handler traps in the current module.

Finally, the virus will register the aforementioned version number in the registry, and if it infected the global template it will quit *Word*, forcing the user to restart it, thus activating the virus macros copied to the global template.

Entry Point Hook

Usually the activation of macro viruses relies on the automatic macro (AutoOpen, Document_Open and the likes). Ostrich does not use these traditional methods. Instead, it will hook two procedures, Document_Open and Document_Close. In fact, the latter will not contain the virus code, only a call to the Document_Open procedure. If any procedure does not exist prior to infection, the virus will create an empty one, then append itself at the bottom of the Document_Open procedure.

After infection, the two procedures will look like this:

```
Private Sub Document_Close()
On Error GoTo NC2kAl1yp0gsG9dTaBy
Original_DocClose_code
NC2kAl1yp0gsG9dTaBy:
Document_Open
End Sub
```

and

```
Private Sub Document_Open()
On Error GoTo JEfW790yQ8Ki
Original_DocOpen_code
JEfW790yQ8Ki:
Virus_code
H2u4mk6bAG3HQN7qR8y:
End Sub
```

Ostrich has two possible entry paths: either the execution falls to the last instruction, or a runtime error will occur somewhere, which will hijack the execution to the virus code. However, there is no guarantee that the virus will activate at all. If the infected macro does not produce runtime errors, and its normal exit point is not the end of the procedure, then the execution flow will not reach the virus code.

Polymorph Engine

The polymorph engine of the virus is not remarkable; it is essentially a combination of old techniques such as variable name polymorphism and string constant encryption with varying encryption key.

During the infection process, the virus extracts its code from the source document. The code is recognized as any instruction between the starting label, JEfW790yQ8Ki, and the ending label, H2u4mk6bAG3HQN7qR8y (or whatever name these have in the current sample). Both label names are mutated during the infection.

The extracted source is processed line-by-line in several consecutive runs. In the first run the random comment lines are removed from the source. Moreover, depending on the values of a random number, 0 to 4 spaces are inserted in front of each code line. Additionally, with a 1:8 chance, a random comment line is inserted at the current position. Each comment line consists of 1 to 9 numbers, each of these being a real value, ranging from 0 to 10000. With a 1:10 chance, a line break is inserted after each code line.

Following this, Ostrich extracts the decryption table from the virus code (which is recognized as anything that is in the first Const variable). As the version info is appended

after the decoder table, the virus can update that as well. If the global template is infected, it will increase the generation counter; otherwise, it will increase the infection counter. The content of the encryption table is shuffled by randomly replacing the character pairs in it.

Then the encrypted string constants are re-encrypted with the new decoder table. The constant encryption is simply a moving XOR algorithm:

```
decoded_string[i]=original_string[i] XOR
decoder_table[i]
```

Since all the string characters used by the virus are encrypted, the operation is simple: it finds all string constants that appear on the right-hand side of an equation mark, decrypts them with the old table and re-encrypts with the new table.

In the next process run, Ostrich replaces all variable names with randomly-generated ones. Since the virus explicitly declares all its variables, this step is simple too, it just has to search for the *Dim var as …* lines in the code, extract the variable name from there, generate a new one and replace all occurrences in the code.

The variable names are between 12 and 22 characters long (although, as it is possible that one variable name is contained within another, the actual length can be longer), and in each character position, with equal probability an uppercase letter, a lowercase letter, or a number appears – with the exception of the first character, which is always an uppercase letter.

In the third process run, the label names used by the virus are mutated. The labels are recognized as lines ending with a colon. The same rules apply to the generated label names as to the variable names.

Ostrich uses minimal stealth functions in the sense that it will remove the following macros which represent a danger to the virus: ViewVBCode, ToolsMacro, FileTemplates and ToolsOptions. Once these have been removed, Ostrich creates new procedures with these names and empty content. This way, as the menu and command bar items are already disabled, there will be no way to access these commands.

Problems

The mechanism is good for virus mating: if the virus infects an already infected Document_Open macro, Ostrich will append itself to the end. Upon infection, the first virus infects the new document, then Ostrich activates, and if the new macro has no other Const lines, it will find the ID string in the target and abort. If the first virus replicates simply by copying the entire content of its Document_Open procedure or the entire class module (and the majority of macro viruses falls into this category), then it will bring along with itself the non-mutating copy of Ostrich. This mating has been observed in a sample of WM97.Rendra.C.

Disinfection Problems

Recently a lively debate went on in *VB* about virus disinfection (see *VB* issues May–July 2001). Ostrich is a good example of how perfect disinfection, or even decent disinfection, is impossible in some cases.

Without wishing to re-open the debate, I would define perfect disinfection as:

1. A procedure that removes all of the virus code.
2. A procedure at the end of which the remaining document matches, byte-for-byte, its state prior to infection.

Given the nasty things *Word* does to documents, point 2 is impossible, therefore a modification is required:

2a. A procedure at the end of which the disinfected code modules match, byte-for-byte, their state prior to infection.

In the case of Ostrich this is clearly impossible, as the virus removes lines from the original code. So I would redefine acceptable disinfection as:

2b. A procedure at the end of which the remaining code modules match, in functionality, their state prior to infection.

I think it is obvious that if not even this criterion can be met, there is no sense in which the virus can be said to have been disinfected. Let us assume that we have a virus scanner that is sufficiently intelligent to parse the macrocode, remove the virus code properly, then remove all the calls to the virus code, finally removing the error traps defined by the virus.

The fact that this virus removes the OPTION EXPLICIT lines is not a problem, as this statement plays a role only in the macro development stage – once a macro is released, it will not be missed if removed.

The removal of the error traps does, however, cause major problems. The entire code flow could change if those traps are missing. Error traps are very important components of legitimate macro programs, not only because they handle unexpected errors, but also they handle the errors that would normally occur. A macro program can enumerate the available drive letters, falling into a trap whenever the drive is not present – a common practice for querying the available drives.

Not only would the code not work in the same way, it would even abort at the first runtime error. One could argue that it was the virus that did the damage, but this claim will not calm the angry customers, who will only notice that after disinfection their utility macros will not work (the fact that the macro did not work before the disinfection either will not hold them back).

The only acceptable option in the case of this virus is to wipe out the entire ThisDocument storage.

CONFERENCE REPORT

VB Goes Czech

Helen Martin

In the period immediately following the events of September 11th, *Virus Bulletin* received a number of inquiries as to whether VB2001 would be going ahead – particularly in light of the cancellation of numerous other conferences around that time. Despite some initial concerns *VB* remained confident that the conference should and would go ahead as planned. And we weren't to be disappointed – indeed, the turnout at VB 2001 truly demonstrated that the AV industry is not easily deterred by physical terrorists and we were delighted to welcome close to 300 delegates to the strikingly beautiful city of Prague.

Inevitably in the light of current events there were some faces missing from this year's conference, including a number of the scheduled speakers. We were grateful to David Phillips, Vincent Weafer and Eric Chien who gallantly stepped into the breach to cover for colleagues who had been unable to make the trip, enabling us to maintain the full and original programme of presentations.

Events kicked off with a Czech beer reception, at which we were treated to quite a spectrum of entertainment including a spectacular fire eating display which was enough to rattle the nerves of hotel staff and conference organisers alike.

At the conference opening a treat awaited us in the form of Eugene Kaspersky's Doc Brown and Andy Nikishin's Marty McFly in their own adaptation of *Back to the Future*. Their Delorian time machine gave the audience an overview of what has happened in the virus arena since the first computer as well as showing us an alarming future in which all the top anti-virus experts have chosen alternative career paths: Peter Ször a cover model for Men's Health, Vesselin Bontchev an Icelandic fisherman, Mikko Hyppönen having taken over the driving seat of Mikka Hakkinen's F1 car and Eugene Kaspersky having opened 'Eugene's' – his very own pub. Happily, Marty and the Doc were able to avert disaster and return us to the present day in time for coffee and the start of the first session.

Looking to the future was a strong theme at the conference. Papers by Richard Wang and Philip Hannay and by Eric Chien predicted the need for changes in AV protection with the onset of *Microsoft .NET*, while Scott Molenkamp predicted potential anti-virus problems and solutions for *Palm OS*. *Microsoft*'s Randy Abrams received a mixed reaction from AV vendors to a tentative proposal to distribute *Microsoft* security patches with their virus definition updates.

Vendors also had plenty to say when David Phillips lead a panel of members of the Anti Virus Information Exchange Network (AVIEN) in an open discussion of the aims and practices of the network. This represented a rare opportunity in AVIEN's one-year existence for AV vendors to be privy to some of the workings of this network. Some healthy debating ensued, during which the legitimacy of AVIEN's early warning alerting system (EWS) was questioned by a number of vendor representatives.

Vesselin Bontchev's fascinating look at the anatomy of a virus epidemic chronicled the spread of self-reporting W97M/Groov.A, which uploads a file to a *Frisk* ftp site. Vesselin took the opportunity to confess that his previous declaration that 95 percent of the population are idiots was misjudged. He has subsequently re-calculated that figure to stand at 97 percent of the population.

Jessica Johnston looked at the anti-virus industry from an unusual angle, though one that was of particular interest to a newcomer such as myself. She had researched the issues of trust and perceptions within the anti-virus industry, in particular relating to CARO. She exposed some opinions which provided interesting food for thought.

Graham Cluley posed the question: what qualifications does a person need to become a 'virus expert'? After revealing the opinons of well-known AV expert the Dalai Lama, Graham raised the pertinent question of the media's responsibility in virus reporting and finished with a test of his audience's attentiveness, inviting David Perry to wander around the room, representing an alluring email attachment.

Concluding the conference, this year's speakers' panel continued the theme of debate from the previous day's AVIEN panel discussion. The session drew to a close as Vesselin Bontchev and *Ford*'s Shawn Campbell had transformed the session from speakers' panel to floor show. Shawn's memorable remark, 'Vesselin you have got to get out of the business' raised a good humoured laugh from all sides of the argument.

The Big Easy – VB2002

Following the resounding success of VB2001, *VB* is pleased to announce the dates and location for the 12th International *Virus Bulletin* Conference. VB2002 will be held in one of the USA's most colourful and energetic cities, New Orleans, Louisiana. The conference will take place on Thursday 26 and Friday 27 September 2002 at the Hyatt Regency hotel. A call for papers will be issued early in the new year. For sponsorship opportunities please email editorial@virusbtn.com. Put the dates in your diaries now and let the good times roll!

FEATURE

Building the Perfect AV: An Administrator's Wish List

Max Morris
First Union/Wachovia Corporation, USA

In my capacity as Enterprise Anti-Virus Administrator for First Union/Wachovia Corporation over the past several years, I have had the opportunity to work with and provide feedback to many anti-virus companies around their solutions, support and communications. While the anti-virus industry as a whole has come a long way towards providing customers with better products and service, there continue to be improvements that could be made.

This article will concentrate on some of the key areas in which we could improve our defences in the united fight against new malware threats. Some vendors have implemented many of these suggestions already, but hopefully this is a good blueprint to strive for across the board.

The Industry

In working with multiple vendors it has become evident that, while there has been a movement towards better sharing of information, we still have a long way to go on co-operation and consistency of data being provided to those of us who try to protect our systems against new threats.

One of the most talked-about shortcomings in our industry continues to be that of a standard naming convention for malware threats. While it is not necessary for a new threat to be given the *exact* same name by all companies, there must begin to be some level of commonality.

One suggestion for a naming procedure is the use of a generic name in the early stages of a new threat being discovered. This could be something as simple as a combination of a date and threat type indicator, with the assignment of a more definitive name as soon as possible. Another option, which would require industry-wide concurrence and participation, would be to determine a single authoritative source that would establish a name for each new threat, based on some set of commonly accepted standards, then release that name to all anti-virus and security sources.

Another major issue encountered repeatedly is the inconsistency between threat levels assigned by different vendors. We need a common set of criteria to be defined so that all companies use the same criteria to determine just how much of a risk a new threat poses. It is very difficult for administrators to decide upon what action should be taken when one AV company describes a threat as low risk and another describes it as medium risk, simply because one

company has received submissions and the other has not.

Finally, we need improved sharing across the industry of information about new threats. One situation I have seen frequently is where one AV company has received submissions of new malware, yet hours and in some cases days after it has been received, no other company is even aware of it. Working together by releasing both information and submission code helps all of us. We are still seeing AV companies that are more interested in being the first vendor to have discovered a new threat than in helping to spread information to other companies (and therefore customers) to help contain the threat. Vendors need to worry less about bragging rights and more about dissemination of information.

The Information

Vendors have significantly improved upon the detail they provide in new threat write-ups. But, while we are seeing more comprehensive data earlier on, there is room for further improvement.

Many vendors don't seem to understand that, early on in a malware outbreak, one of the most critical pieces of information is that of preventative characteristics. Even in the best of circumstances, a new pattern file will not be available immediately, so you are faced with a period of time during which the only options you have are filtering on a threat or shutting your company's email and Internet connections down.

Many vendors seem to assume that all companies have the ability to carry out complete content filtering. Unfortunately this is not always the case, whether due to performance, budget or political reasons. So it is crucial, especially early on, to know any and all unique email characteristics including subject line, body text and attachment names.

In addition, I think we would benefit from two unique sets of threat indication. The first threat level would be limited to what the malware can do *potentially*, from a propagation and payload perspective. The second would define the actual current threat level based on the current wild characteristics and the rate of propagation being seen. A third

recommended threat indication could be projected by the vendor by combining the previous two threat indications. However, most administrators will determine the actual threat applicable to their own company, based on factors that the vendor does not have access to. In my mind, it is far more important for a vendor simply to provide the data upon which we can base a decision, rather than worry about the threat level they perceive.

One set of information that I feel is sorely missing is the detail around the actual wild characteristics of the threat. Vendors do not seem to understand that a company's reaction to a new threat is based to some extent on exactly where the threat starts and how quickly it is spreading.

Detail around where, geographically, a threat was first identified and whether the first submissions are consumer- or business-based can give administrators a good sense of what steps they need to take. In addition, when a new threat is seen in a business environment, knowledge of the type of industry in which it has been encountered can be beneficial, since businesses in the same industry have a tendency to see email communications between themselves more regularly than from other companies. Finally, knowledge of specific and up-to-date wild numbers and a historical rate of propagation allows an administrator to understand exactly how a new threat is spreading.

Another shortcoming I have encountered is the predominant lack of virus write-up revision histories. Frequently, if not always, there are changes to the data around a new malware threat, particularly in the early stages of its appearance. While these changes are not always critical, some, such as those related to propagation methods/characteristics and payloads, can affect threat levels for companies and the actions being taken. I believe it is the responsibility of the vendors providing the information to detail any changes made to earlier communicated data and in a way that is easy to assess quickly. Time is critical, and no administrator has the time to scrutinize every write-up in an effort to ascertain what is different.

The Company

We have long since had to worry about whether a vendor's scanner detects a new threat and today the major AV companies' scanners provide more than adequate protection. But there are a couple of improvements that come to mind from which we could benefit in the product development area.

First, the future of anti-virus scanners needs to be a more robust solution. Today's threats are becoming more complicated and utilizing multiple methodologies for propagation, with the line between malware, intrusion and exploits becoming increasingly less well-defined. While today we have multiple options for anti-virus, firewall and content filtering protection, we need to move towards a one-solution-does-all approach and have all of these components combined in a single product.

Second, many companies' products are just now moving to the enterprise level of providing reporting and alerting. While most solutions have detailed information available, a significant number are not built with the corporate level in mind, stopping short of a complete overall picture.

Often, robust reporting requires the use of manual batch processes to collect data, combined with third-party reporting solutions to genuinely provide the detailed level of information that is required. In addition, it is crucial to know when a new threat is first encountered within a company so that immediate alerting can be provided to an administrator.

Moving away from products and into the support arena, one of the most important things for any person dealing with malware outbreaks, especially someone who is in charge of a large enterprise with significant numbers of devices and critical production business functions to protect, is timely communication of a threat from the vendor or security company. In my experience it has been only recently that these companies have begun building more robust notification systems for their customers.

Email is, of course, the most common form of communication. However, the reality is that multiple methods of delivery, the opportunity for customization based on times (to the day and hour level) and specific threat levels are needed to ensure that enterprise administrators are always alerted to new malware outbreaks that potentially could threaten their company.

Finally, we continue to see a predominance of a consumer mentality by our vendors. This is particularly the case from a deployment perspective. Vendors need to understand that, in an enterprise environment where there are tens, sometimes hundreds, of thousands of devices ranging from desktops to file/print, mail and application servers to email gateways, significant planning, testing, communications and lengthy rollouts are required to ensure minimal end user impact. While the answer for an individual consumer is simply to patch their scanner, in the business world there is no such thing as a simple upgrade.

Striving for Perfection Together

Just as we continue to strive for 100% detection rates for malware, we need to recognize that detection and eradication of new threats is a constant battle and one over which we cannot become complacent. It seems that whenever we appear to have achieved a certain level of adequate protection against the virus writers, we are faced with yet another variant, a new propagation method or system exploit that must be overcome.

The key is that we must remember that we are all in this together. Only through sustained co-operation and ongoing feedback between anti-virus vendors, businesses and the security industry can we achieve the ultimate goal of ensuring the protection of our data and systems.

FEATURE SERIES 1

Worming the Internet – Part 2

Katrin Tocheva
F-Secure Corporation, Finland

[*The first instalment of this series looked at Sendkeys; in part two Katrin discusses the spread of worms using the CreateObject function.*]

CreateObject is a powerful Visual Basic Script (VBS) function that allows the opening of one *MS* application from another. This function was added into VBS v2.0, which is included by default in *Internet Explorer* (*IE*) *4.0*. *IE 4.0* is a part of the default installation of *Windows 98*. Later, *IE* 5 was included in *Office 2000*, increasing the number of installations supporting VBS even further.

All this, combined with the fact that virus writers realized the power of the CreateObject function (which is also available in VBA), resulted in the development of new macro viruses that use this function to run one application from another and infect them – the so-called multi-application or cross-application macro viruses.

The first successful cross-infector, O97M/Shiver [I.Muttik], used Dynamic Data Exchange (DDE) to cross-infect *MS Word 97* and *Excel 97*. Later, the O97M/Tristate virus used the Component Object Model (COM) feature, also known as ActiveX, that allows one application to be accessed from another. It uses the CreateObject function to 'open' and the GetObject functions to 'switch' to another of the three applications that it infects: *MS Word 97, Excel 97* and *PowerPoint 97*. Another example is the first *MS Project* virus [K.Tocheva 2], which cross-infects *MS Word* and *MS Project* documents using CreateObject to open these applications and the GetObject function to 'jump' from one opened application to the other.

W97M/Coldape.A, discovered at the beginning of November 1998, is the first virus to make the connection between VBA macro viruses and VBScript viruses [K.Tocheva 1], but it is also the first virus to use the CreateObject function to send an email message. This virus creates a VBS file that sends an email message, using *MS Outlook*, to Nick FitzGerald, the former editor of *Virus Bulletin*. This was the first attempt to create a mailer, although it was not a mass-mailer. Later, many newly discovered VBScript viruses and droppers like Loud, Hopper and Break used the CreateObject function to infect *MS Word 97* from Visual Basic Scripts [K.Tocheva 3].

A few months later, a new method of virus spreading was developed – the method used by the mass-mailers. Nowadays, CreateObject is used by many viruses mostly to run an email application (usually the most popular email client, *Outlook*) 'silently' and to spread via the Internet. Viruses such as W97M/Melissa (the first mass-mailer that caused a global epidemic), VBS/Freelink (the first VBS mass-mailer), VBS/Loveletter (the most widely-spread virus to date) and many others use the CreateObject function to open *MS Outlook* and send themselves to enormous numbers of recipients.

MAPI&AddressLists

The first time the CreateObject function was used to open an email application and send malicious code via email was in W97M/Nail.A@, also known as Automated Chain Mail (ACM) worm. To spread via email this uses the MAPI&AddressLists method. This method is similar to CreateObject and Outlook.Application&AddressLists described below, but it uses Mail Application Programming Interface (MAPI) object (CreateObject('MAPI.Session')) instead of Outlook.Application object. This method is intended to spread a worm regardless of the email client installed, as long as the client has MAPI support – as is the case in most of the modern *Windows* email clients. (However, it turned out that Nail is unable to use any client other than *Outlook*.)

The interesting thing in Nail's replication mechanism is that its VBA code was located in a template on a remote Web site and writes a reference to that remote template to the affected user's files. This made Nail unlike previous *Word* macro viruses – it does not infect documents or templates by copying its code there. Instead, it inserts a link to the remote template. This remote template contains the actual email worm – a VBA code – and sends the active document via email to all recipients listed in the first address book.

Such remote template infection bypassed detection by anti-virus scanners simply because, at that time, scanners knew how to detect viruses inside documents and templates but were unable to recognize references to a remote template as suspicious. Also, the fact that the mass-mailing code in the remote template was located on a remote Web page, gave the virus writer the advantage of being able to modify the code. From a different point of view, however, the spread of a virus whose code is located on a remote Web page depends on the availability of that page. As soon as the Web page is closed, the virus stops.

The MAPI method, used by Nail, did not prove to be as widely used by viruses as expected. Another method that uses *Outlook* directly became more popular – the method used by the notorious Melissa and Loveletter viruses.

Outlook.Application&AddressLists

The main difference between this method and the MAPI method is that this uses only *Outlook* email client to spread. The Outlook.Application&AddressLists method consists of

the following steps: first it opens the *MS Outlook* application using CreateObject ('Outlook.Application'); then it searches for email recipients (AddressEntries) in the user's address book(s) (AddressLists). Next it creates a new message (CreateItem) and builds its subject and body texts; to that message it adds the collected email addresses (Recipients.Add) and the worm file (Attachments.Add). Finally, it sends the message it has assembled (Send).

Worms using this method usually collect recipients (all or some of them) listed in each address book. This method has been used by most of the known widely spread email worms, such as Melissa, Loveletter and Homepage, and continues to be the main method of spreading used by mass-mailers. This is helped not only by the prevalence of the above-mentioned worms, but also the development of worm creation kits such as VBSWG and others. This resulted in the creation of worms like VBSWG.J@mm (also know as AnnaKournikova) by people who do not even need to have a detailed knowledge of programming languages.

Outlook.Application&GetDefaultFolder

GetDefaultFolder is another method of mass-mailing that uses the CreateObject function. While it is not very common, one example of its use is in W97M/Mimir – an overwriting macro virus that contains a fast mass-mailing routine implemented with the GetDefaultFolder method.

Like the Outlook.Application&AddressLists method, the Outlook.Application&GetDefaultFolder method opens the *Outlook* application first, using the CreateObject function. Then a new message is created (CreateItem) and its subject and body texts built (the last two are optional). Next it collects email addresses, searching in the default folder (GetDefaultFolder). This varies depending on the worm – it might be the *Outlook* contact folder, as in W97M/Mimir.A, the SentMail folder, as in the W97M/Lucia.A virus, or one of the other *Outlook* folders. Next, this method adds the collected email addresses in the 'To', 'Cc' or 'Bcc' fields of the message, then it attaches (Attachments.Add) the worm code (the file in which it relays) to the same message. Finally, the prepared message is sent (Send) via the email.

Worms Using IE Weakness

Two methods of spreading have been developed this year using a remote Web site for hosting the mass-mailing code. There are two types of worm (examples are VBS worms Vierika and Loding) that use the CreateObject method but do not contain the mass-mailing code in an attached file – instead it is on a Web site. This is similar to the Nail worm but, while Nail was using a security hole in *Word 97* and was able to 'skip' the built-in macro virus protection, these methods use the weakness in *Internet Explorer* instead.

Lower the Security Setting of Internet Explorer

The method the Vierika worm uses requires two worm components: one to lower the security settings in *IE* and another to execute the mass-mailing code. Vierika arrives in a message with an attachment that is a small VBS file, but this does not contain the actual spreading routine. Once the user clicks on the VBS file, it lowers the security settings of *IE* and changes the start page to point to a Web site where the second part of the worm resides. The next time the user opens the browser, the second part of the worm will be executed from the Web page.

The second part of Vierika is another mass-mailer that uses the CreateObject and Outlook.Application&AddressLists method. This part creates a VBS file in the root of the C: drive ('Vierika.JPG.vbs') which contains the first part of the worm. It also contains the mass-mailer that spreads it to all recipients listed in all of the user's address books. By using the .JPG extension and the body text 'Vierika.jpg' the worm tries to disguise itself as a picture. By spreading a VBS file that does not contain mass-mailing code, this virus also tries to avoid generic and heuristic detection. But, like Nail, this worm depended on a remote Web page, and as soon as this was disabled the worm was unable to spread.

Use an Exploit in Internet Explorer

VBS/Loding is a worm that sends email messages without any attached file. Instead the message body contains a link to a remote Web page. The message text is intended to make the user click on the link, which points to the worm code. If the user's default browser is *IE 4* with security setting 'Medium' (the default), and he visits the Web page, the code of Loding (a combination of JavaScript and Visual Basic Script) will execute. To do this Loding uses a vulnerability known as 'Microsoft Virtual Machine ActiveX Component'. The VBS code embedded in the HTML page is the mass-mailer that uses the CreateObject and Outlook.Application&AddressLists method to send the messages to all recipients listed in each user's address book.

CreateObject – Why is it so Successful?

Of all the CreateObject methods described, the most common is Outlook.Application&AddressLists – most of the known viruses use this method to propagate. One important reason for this is that Melissa was the first widely-spread mass-mailer to use this method. Its success is due to the fact that it uses the most popular email client and was posted to several newsgroups. Also, after Melissa caused a global epidemic on 26 March 1999, its source code was made available on a Web site. This, together with the prevalence of Melissa, resulted in many copycats and later in more and more similar creations implemented in Visual Basic Script and Java Script languages.

On 4 May 2000, the biggest case in the history of computer viruses, LoveLetter, caused a global epidemic. Like Melissa it resulted in many copycats: 30 new variants were created in just three days. This shows that virus writers want to use a proven successful spreading mass-mailer, adding their 'fingerprints' thus creating many new virus variants to make their creations as widespread as Melissa and Loveletter.

FEATURE SERIES 2

Combating Viruses via Email Part 1

Carlos Ardanza
Panda Software, Spain

Without a doubt the most common virus entry point is email. Some studies suggest that as many as 90% of infections are brought about in this way. The issue has been widely discussed and the facility with which some mail clients allow viruses to propagate has come under much criticism. However, there is rarely any mention of the tools that manufacturers of mail clients and servers offer anti-virus manufacturers to fight against viruses transmitted via this entry point.

You don't have to be Einstein to understand how anti-virus protection works on file systems. Despite the technically complex nature of these solutions – they have to interact with the operating system at a very low level – there is the advantage that we are dealing with files on disk that simply need to be opened and the typical scan and disinfection tasks carried out, using the file access functions that we are familiar with.

However, email scanners are in a hostile environment, as they have to operate with files that are not stored in a directory on disk, but embedded in huge message databases. The simplest solution, which is used by the vast majority of anti-virus manufacturers, is to extract the file to disk, scan it and disinfect it, if necessary, with the usual functions and then return it to the message database. This requires four complete file read/write operations in addition to the bytes that are read in order to scan and disinfect it. This does not seem to be the most efficient solution.

Another problem for email scanners is that they have to scan not only attached files, but also the message body. In fact, there are usually two, and in some cases even three, message bodies (plain text, RTF, HTML, RTFHTML, and so on). It is important to remember that viruses that are transmitted in the message body (such as BubbleBoy, Kak and Forgotten) have caused millions of infections world-wide over the last year. These are not viruses that cause huge waves of infections, but they have caused and continue to cause a constant trickle of infections every day.

Microsoft and *Lotus* mail applications have been estimated to represent over 90% of the world market of mail and groupware applications. In this two-part article I shall describe the means offered by *Microsoft* and *Lotus* to the anti-virus industry to combat viruses, in both clients and servers. The first part will concentrate on *Microsoft*, and in the second part, next month, I shall be looking at the contributions of *Lotus*.

Microsoft Exchange/Outlook Client

The system offered by *Microsoft* for accessing their messaging systems is Messaging Application Program Interface (MAPI). Although *Microsoft* is moving away from this system in favour of other more modern systems such as Collaboration Data Objects (CDO), many anti-virus products use it in their on-demand scanners. It is an extremely powerful and flexible API, although quite complex to use.

Without a doubt the aspect that most helps when developing an efficient anti-virus with MAPI is that the ISTREAM interface is totally implemented. This allows an anti-virus to scan and disinfect a file, transferring only a few KB of each file from the server. Unfortunately, this feature is not used by the majority of anti-virus utilities, which continue to extract attached files completely to disk in order to scan them, resulting in loss of performance and the corresponding load on the network.

One negative aspect of MAPI is that it does not have a synchronous system for the interception of messages. This impedes the development of a real-time virus scanner that guarantees that a user cannot access a message until it has been scanned and disinfected. *Microsoft* gets around this point in mail clients through *Exchange*/*Outlook* client extensions. Anti-virus manufacturers can develop a client extension, so that the client will inform the extension every time a message write, read, send or receive event occurs. These events are synchronous so the user cannot access the message until it has been completely screened by the anti-virus. This prevents infected messages from 'leaking out'. The only negative aspect that I have noticed are some problems with the integration of the extensions in the user interface in *Outlook 97* and some versions of *98*.

MAPI provides everything necessary for developing an efficient and secure anti-virus product. Also, it allows the protection of personal folders (.PST) and direct connections via modem that cannot be protected from a server.

Exchange Server

As with the client, all anti-viruses use MAPI for on-demand scans. The problem in developing a good anti-virus for *Exchange Server* lies in the real-time system. For this reason, the different anti-virus manufacturers have used up to four different APIs.

MAPI

The main advantage of MAPI is that it is the standard system and allows efficient access to attached files through its ISTREAM interface, provided that its maximum capacity is used and the files are not extracted to disk. The

main drawback of MAPI is that it does not have a synchronous hook system. An anti-virus that uses the MAPI events system AdviseSink is another client, so that in theory, the scanner is informed of each event in the Information Store at the same time as the rest of the clients. In other words, it is more of an alert system than a hook system.

Fortunately, the scanner runs in the same server, so usually it has enough time to scan and disinfect before the event reaches the *Exchange*/*Outlook* clients in the workstations. Therefore, it is vital that a scanner using this technology is very efficient and is able to detect and disinfect messages before the notification reaches the workstations through the network. At *Panda*, our first approach to solving this limitation was to maximize the performance of the scanner and assign very low priority to the on-demand scans. In later versions, we implemented an autotuning system that allowed the scanner to adjust the CPU load by thousandths of a second, depending on the global load on the server.

ESE

ESE (Extensible Storage Engine) is the database management system that *Microsoft* uses to store *Exchange* messages in versions 5.5 and later, replacing JET. *Microsoft* has also used this engine to store the Active Directory data in *Windows 2000*. Although no API has been released for operating with ESE, two anti-virus manufacturers have developed their real-time solutions on this engine, probably using reverse engineering, as *Microsoft* has released neither the database format nor an API for accessing this engine.

Microsoft Product Support Service formerly obliged users to uninstall anti-virus products that used this technique before providing technical support for *Exchange 5.5* and *Exchange 2000 Server*. Recently, however, *Microsoft* reached a compromise with these anti-virus manufacturers (http://support.microsoft.com/support/kb/articles/Q250/5/00.asp). Yet, as *Microsoft* indicates, there is still a risk of data loss or database corruption due to an incorrect implementation of a non-standard interface.

This is a technique that, one way or another, covers the limitations of the capacity of MAPI to intercept messages and the multiple limitations of AVAPI (described below). As this is a non-standard solution, it is vulnerable to the slightest change to a function parameter made by *Microsoft* in subsequent Hotfixes or Service Packs, potentially causing irreparable damage to message databases.

AVAPI

In September 1999, *Microsoft* introduced, with *Exchange 5.5 SP3*, a new API to improve shortcomings in the architecture of *Exchange Server* for developing an anti-virus application, i.e. the fact that MAPI only has asynchronous events. We analysed this API when it was still in the Beta process and even then it seemed to have a sufficient number of limitations to warrant immediate rejection of the adaptation of our anti-virus products to this API.

AVAPI gives access only at attached file level; the message that contains the file being scanned cannot be accessed. This means that it is impossible to generate a report and adequate alerts. It is impossible to know the most basic information about the origin of the file, such as the mailbox, folder and message to which it belongs. Nor is it possible to know the name of the sender or the recipients of the message. Imagine an administrator who has a server with ten million messages and the anti-virus product informs him that the virus MTX has been detected in the file QI_TEST.EXE. He would need an entire army to open each of the ten million messages and find the infected file.

In addition, this API does not allow the message body to be scanned. As mentioned earlier, viruses that are transmitted in the message body have accounted for the majority of the infections carried out over the last year.

Not all of the ISTREAM interface functions used for reading and writing on attached files are implemented in AVAPI. One of the functions that this interface lacks is the capacity to change the size of files (which is vital in disinfection operations). In addition, it does not allow the object itself to be read or written on, therefore it is necessary to completely read and write a file in order to disinfect a virus, resulting in a load on the CPU and memory.

An attached file cannot be deleted. The only viable operation for eliminating a virus that cannot be disinfected (Trojans, dropper, companion, etc.) is to overwrite it completely.

The date and the time of the attached files cannot be obtained.

AVAPI does not intercept (and therefore does not allow the anti-virus to scan) messages sent via *Outlook Web Access* (*OWA*), SMTP and, in general, any medium used to access a mailbox through non-*Microsoft* mail clients.

The system does not launch the scan until some time after the inbound messages reach the mailbox. All sorts of problems can occur if a user tries to open, send, etc. the blocked message before it is scanned: error messages, transmission errors, time outs, and so on. (Many of the problems described in this point are due to the fact that AVAPI is not multi-threaded. However, this problem was fixed in *Exchange 5.5 Service Pack 4*.)

AVAPI has a cache scan system which is positive if implemented well but is, in reality, not valid. The system does not call the anti-virus again once it has confirmed that the file is clean. Even if the administrator modifies the settings, making the scan configuration more restrictive (e.g. activating the compressed file scan or adding to the list of extensions to be scanned), the file is not passed to the scanner again. In order to resolve this shortcoming to some extent, AVAPI allows the anti-virus to scan all files every time the signature file is updated. This causes a problem where there are frequent updates, as with *Panda*'s daily updates, since a server could contain millions of messages.

In addition to the shortcomings of this API for developing a good anti-virus product, AVAPI had many bugs. Some of the more serious bugs are referred to in the following Web pages: http://support.microsoft.com/support/kb/articles/Q264/7/31.ASP, Q262/4/91.ASP, Q263/9/47.ASP, Q276/0/56.ASP and Q263/7/10.ASP. (Note: Most of these bugs – and the other shortcomings discussed – were dealt with in *Exchange 5.5 Service Pack 4*. However, those considered most significant by *Panda* are still present.)

VSAPI

Microsoft has developed a new API that deals with the majority of AVAPI's limitations: VSAPI (Virus Scanning API).

The advantages of VSAPI include:

- High performance due to: low-level implementation; the fact that it implements the ISTREAM interface correctly; and that it uses 'single instance scanning', meaning that a message sent to 50 recipients is scanned only once. (Whereas with MAPI, for example, it would be scanned 51 times.)
- High reliability, since complete and exclusive access to the object to be scanned is guaranteed before sending (therefore messages are scanned in the Outbox), or before the user opens it. Also, unlike AVAPI, it covers all access points to the server: MAPI clients, OWA, SMTP/POP3 clients, IMC, X400, etc.
- VSAPI has a more refined and efficient 'version control' system than that of AVAPI, which sends messages to be scanned in the background whenever there is a new version of the virus signature file.
- It has a thread pool (which depends on the number of server processors) for the scan, which allows optimization of performance/effectiveness. One of the most notable characteristics of this system is that files which reach the server are placed in a queue in this thread pool but, should a user try to open one of the messages in the queue, *Exchange* passes the message directly to the front of the queue so that it is scanned immediately.
- Similarly, a proactive system allows priorities in the background scan queue to be ordered such that a message will be passed immediately to a higher priority when a user tries to open it.

I believe that *Microsoft* has done a great job with this API, combining effectiveness, flexibility, performance and stability. Perhaps the only aspect that could be improved is the complete implementation of the ISTREAM interface to allow reading and writing at the same time on the same object. However, this is a minor point, bearing in mind all the positive features offered. VSAPI is included in *Exchange 2000 Service Pack 1*, released by *Microsoft* in June 2001. This is, without doubt, a great opportunity for administrators to switch their systems from *Exchange 5.5* to *Exchange 2000 Server*.

COMPARATIVE REVIEW

Windows NT

Matt Ham

The line-up of products in this comparative included a number of newly packaged products, but no true newcomers. However, this gave me no cause to imagine that the path of testing would be a smooth one – past tests on *NT* have shown a host of oddities in behaviour which act as pitfalls and banana skins for the unwary scanner. Given 21 products to review, the time for prevaricating is over – so on with the details.

Test Sets

VB2001 was deemed momentous enough that the September WildList was delayed to allow reporters to wend their way back from Prague. As a consequence, the test sets in this review are based on the somewhat antiquated August 2001 WildList. This should give the products every chance of doing well on In the Wild detections, and developers should be warned that any misses in the ItW test set will be particularly noteworthy, with a month's preparation time available to all. Making their debut in the WildList are the usual selection of macro viruses in addition to the combined VBS/EXE worm W95/Linong.A.

Most noteworthy (in terms of press interest at least) is W32/Bady.C, better known as Code Red II. This leads to the question 'what about Code Red?' The original Code Red had no file-based portion and, while the later derivatives contained some code, this can more accurately be considered Trojan. The Trojan parts have not been included in the test set, since they are no more than dropped payload files of the worm and are not part of the infective process. Technically, the fileless nature of the worm portion of these specimens is rather problematic as far as testing detection is concerned.

Two possibilities were considered: testing on a real infected machine or using files which contain an image of the infected memory. The latter was dismissed quickly since experiments with floppies and file images of disks have shown there to be major differences in behaviour between these two forms of the same data – the same could be expected of file and memory representations of data, which would render meaningless any results gained in this way. The ideal solution would be the use of infected machines, but this also was forced into the reject bin by virtue of the additional manpower and hardware required. Active Code Red detection is thus not included in this test.

Additions to the other test sets included two of particular interest, W32/Zmist.D and W32/Nimda.A. W32/Zmist.D is of note simply because it is widely considered to be a

difficult virus to detect due to its use of advanced polymorphic techniques (see *VB* March 2001, p.6). Not a threat in the wild, Zmist can be considered indicative of the complexity of detection to be expected in new generations of the virus threat. W32/Nimda.A, on the other hand, needs no introduction and will be featuring in the ItW set in the next comparative review. Here, Nimda is notable for the additional extensions it uses: .TMP, .EML, .NWS and .ASP are all potentially testing additions for those products not scanning all files.

Test Procedures

Testing procedures remain unchanged from those performed recently. Tests were performed on a *Windows NT4* server with Service Pack 6 and *Internet Explorer 5* installed. Scans of the test set were performed on a local hard drive using the default settings for the scanner as far as files to be scanned and methods of scanning were concerned.

Results for on-demand scans were, by preference, logged using the log generation facilities of the program under test, with deletion of infected files being the method used if log files proved resistant to parsing for usable results. On access testing was, by default, performed by attempting to open files and testing for blocking of this process. If not blocked by default, copying the files was attempted, checking for denial of attempts and logging the results.

Aladdin eSafe Desktop 3.0.33

ItW Overall	100.00%	Macro	99.31%
ItW Overall (o/a)	99.92%	Standard	98.17%
ItW File	100.00%	Polymorphic	92.47%

The greatest mystery concerning this product was its version number – invisible to the naked eye and only apparent while the product was being installed. Happily, viruses were much more easily detected, with lack of null extension scanning causing the only misses in the ItW test set. This lack of scanning applied only on-access and was expected by the developer as a result of a design decision.

The files are detected as viral when run but *Aladdin* is of the opinion that adding no-extension to the list of files which should be scanned is an unnecessary overhead. Unfortunately for *Aladdin* running each and every missed file to check for such behaviour is not really feasible.

Elsewhere there were problems in the clean test sets where the scan process repeatedly hung on the clean executable files set. The OLE set was scanned in a very respectable time with both compressed and raw data, but the zipped clean executables were somewhat sluggish. The problems encountered on executables are probably due to a high percentage of dynamically compressed files in the test sets. The product scans such files more slowly than might be hoped and as a result of the same underlying issues there may possibly be instability.

Alwil AVAST32 3.0

ItW Overall	100.00%	Macro	99.45%
ItW Overall (o/a)	99.07%	Standard	98.87%
ItW File	100.00%	Polymorphic	93.10%

Like the previous product, *AVAST32* showed misses due to extension issues, here only on demand, these being the .MDB files of the never-threatening ItW A97M/Accessiv.A and B viruses. However, these files were picked up as infected by the on-access scanner. Misses ItW were relegated to the single sample of W32/Badtrans.A, which was missed on access. This was something of an anomaly, since most differences between on-access and on-demand scanning were in the more recent and complex additions to the polymorphic sets.

An additional similarity was that *AVAST32* suffered from a frozen scan on the clean set – though on this occasion on the clean OLE file set. This was a disappointment as other clean set scanning times were respectable. On several occasions this timing would have been even more impressive if the internal timer was to be believed – this had a habit of claiming an elapsed time of zero seconds. A few additional niggles included the selection process for these scans which still does not offer browsing for the selection of targets.

Computer Associates eTrust Antivirus 6.0.96

ItW Overall	100.00%	Macro	100.00%
ItW Overall (o/a)	100.00%	Standard	99.98%
ItW File	100.00%	Polymorphic	97.50%

Although sporting an all-new box, fashionable 'e-name' and lurid splash screen graphics, *eTrust* is not perceptibly different from the *InoculateIT* it replaces. Stability and ease of use have been preserved, together with the usual high rates of detection. Misses were confined to two viruses: W32/Zmist.D was missed in all 43 samples in the polymorphic set, while a .HTM sample of W32/Nimda.A was missed in the standard set.

eTrust performed well in the clean test sets, with no false positives and reasonable speed of scanning and is thus given a VB100% award. Testing was performed using the default product engine, derived from the *iRiS* product of yesteryear, but it can also use the *Vet* engine.

CA Vet Anti-Virus 10.3.8

ItW Overall	100.00%	Macro	100.00%
ItW Overall (o/a)	100.00%	Standard	100.00%
ItW File	100.00%	Polymorphic	99.35%

Vet, like *InoculateIT*, shows signs of a slight migration in designation, with the *eTrust* logo being visible on the box (though in a very much less obtrusive manner than its sister

product). As far as speed of scanning the clean test sets is concerned, there was little to choose between the two products, with *Vet* slightly faster on the non-archived sets while losing out on the archives.

Traditionally, these two products have been distinguished in the polymorphic test sets, and this test was no different. *Vet* detected 32 of the 43 W32/Zmist.D samples in the test set and a lone sample of ACG.A was its only miss in the remaining viral samples. A good result for the team at *Vet* who, once more, help *Computer Associates* gain a pair of VB100% awards in the same comparative.

Command Antivirus 4.62.4

ItW Overall	100.00%	Macro	100.00%
ItW Overall (o/a)	100.00%	Standard	99.95%
ItW File	100.00%	Polymorphic	97.50%

In terms of detection, *Command Antivirus* missed two of the eight W32/Nimda.A samples (the .ASP and .TMP samples), while all of the W32/Zmist.D samples evaded detection. From the remaining test sets there were no misses.

In terms of speed, *Command* was at the faster end of the pack when scanning of clean files was performed and, with no false positives to its name, a VB100% is awarded. It should be noted that scanning of archives is off by default, which is quickly becoming an anomaly in these tests.

The fact that this product gained a VB100% award is not to say that there were no niggling problems; the floppy scanning tests proved somewhat awkward. In fact, general awkwardness in the scan process, and the alert boxes being hidden beneath other windows, almost gave rise to misses being reported where there were none.

DialogueScience DrWeb 4.26

ItW Overall	100.00%	Macro	100.00%
ItW Overall (o/a)	100.00%	Standard	99.78%
ItW File	100.00%	Polymorphic	97.50%

DrWeb detected 15 suspicious files in the clean executable test set but was denied the title of 'most paranoid' for this review. It also was denied the past glories of its full detection of all files in the test set, W32/Zmist.D and W32/Nimda.A being primarily but not the sole cause of this. There were also misses in the newly-added W32/Vote.B and .C samples in the standard set – though only the executable portions were missed. Other than these there were full detections of all files in the test sets and thus another VB100% award is winging its way towards St. Petersburg.

The slight problems encountered in past reviews recurred in the changing of on-access scan parameters – even changing the location of the log file required a reboot. Also there was a crash during the on-demand boot scan test – though other than this momentary instability the boot scanning process was one of the more user-friendly encountered.

Eset NOD32 1.114

ItW Overall	100.00%	Macro	100.00%
ItW Overall (o/a)	100.00%	Standard	100.00%
ItW File	100.00%	Polymorphic	99.53%

Eset have begun mentioning *VB* not only in their splash screen but also in the CD wallet information – referring to their past record of no misses, ever, in the ItW test set (failures to gain VB100% awards have been due to false positive issues). Their claim record remains unbroken, with only eight of the W32/Zmist.D samples being missed in the on-access or on-demand testing procedures.

Additionally, *NOD32* remains one of the fastest products on review, a speed which it combines with a recent record of no false positives or suspicious files. It will come as no surprise, therefore, that *NOD32* is the recipient of the fifth VB100% of this comparative.

FRISK F-Prot Antivirus 3.11

ItW Overall	100.00%	Macro	100.00%
ItW Overall (o/a)	100.00%	Standard	99.89%
ItW File	100.00%	Polymorphic	97.50%

F-Prot managed to throw a single exception early in the scanning process which, thankfully, was not reproduced later in the tests. There was also a degree of poor change detection apparent in the on-access floppy scanning procedure, with many disks having to be scanned four times with intervening clean disks before detection could be triggered.

After these complaints there was full detection in the on-access scanning, together with ItW and macro test sets. Considering that there were numerous new samples added to the macro set, this is somewhat more impressive for all products gaining clean sweeps in that set than might otherwise be assumed. Misses were W32/Nimda.A and all the W32/Zmist.D samples, with the addition of partial detection of W32/Vote.C and W95/SK.8044. Once more a VB100% award is gained.

F-Secure Anti-Virus 5.30

ItW Overall	99.83%	Macro	100.00%
ItW Overall (o/a)	99.73%	Standard	99.69%
ItW File	99.82%	Polymorphic	97.50%

Derived directly from the previous product, *FSAV* might be expected to have a similar detection rate – until, that is, it is

On-demand tests	ItW Boot		ItW File		ItW Overall	Macro		Polymorphic		Standard	
	Number missed	%	Number missed	%	%	Number missed	%	Number missed	%	Number missed	%
Aladdin eSafe Desktop	0	100.00%	0	100.00%	100.00%	31	99.31%	74	92.47%	35	98.17%
Alwil AVAST32	0	100.00%	0	100.00%	100.00%	22	99.45%	71	93.10%	23	98.87%
CA eTrust	0	100.00%	0	100.00%	100.00%	0	100.00%	43	97.50%	1	99.98%
CA Vet Anti-Virus	0	100.00%	0	100.00%	100.00%	0	100.00%	12	99.35%	0	100.00%
Command Antivirus	0	100.00%	0	100.00%	100.00%	0	100.00%	43	97.50%	2	99.95%
DialogueScience DrWeb	0	100.00%	0	100.00%	100.00%	0	100.00%	43	97.50%	6	99.78%
Eset NOD32	0	100.00%	0	100.00%	100.00%	0	100.00%	8	99.53%	0	100.00%
FRISK F-Prot	0	100.00%	0	100.00%	100.00%	0	100.00%	43	97.50%	3	99.89%
F-Secure Anti-Virus	0	100.00%	3	99.82%	99.83%	0	100.00%	43	97.50%	22	99.69%
GDATA AntiVirusKit	0	100.00%	0	100.00%	100.00%	0	100.00%	43	97.50%	2	99.95%
GeCAD RAV	0	100.00%	0	100.00%	100.00%	0	100.00%	51	97.57%	13	99.67%
Grisoft AVG	0	100.00%	1	99.97%	99.97%	20	99.50%	167	89.91%	66	96.92%
HAURI ViRobot	0	100.00%	75	91.34%	91.82%	363	90.42%	10836	35.38%	656	65.18%
IKARUS virus utilities	0	100.00%	14	98.83%	98.90%	143	96.67%	426	90.73%	89	95.14%
Kaspersky Lab KAV	0	100.00%	0	100.00%	100.00%	0	100.00%	43	97.50%	2	99.95%
NAI NetShield	0	100.00%	7	99.57%	99.60%	3	99.97%	2	99.88%	19	99.00%
Norman Virus Control	0	100.00%	0	100.00%	100.00%	0	100.00%	61	95.47%	0	100.00%
Sophos Anti-Virus	0	100.00%	0	100.00%	100.00%	13	99.66%	234	92.98%	20	99.36%
Symantec Norton AntiVirus	0	100.00%	0	100.00%	100.00%	0	100.00%	0	100.00%	0	100.00%
Trend ServerProtect	0	100.00%	0	100.00%	100.00%	3	99.94%	255	92.87%	9	99.78%
VirusBuster VirusBuster	1	91.67%	0	100.00%	99.53%	4	99.90%	71	92.97%	10	99.72%

noted that the extension list for *F-Secure*'s offering has been kept deliberately restricted. Detection of the .BAT and .LNK samples of W32/SirCam.A and the .DLL sample of W32/MTX.B ItW is thus effectively off by default.

In the standard set the BAT/911.A and B samples were missed for the same reason, along with the .TMP file associated with W32/Nimda.A. Other than purely extension-based misses, only samples of W32/Zmist.D went undetected.The reasoning behind the decision to restrict the number of extensions scanned is the customary one of reducing scanning times – which, admittedly, are already rather slower than might be considered ideal. Quite whether this is the best method of dealing with such a speed issue is, however, open to debate.

GDATA AntiVirusKit Generation 10

ItW Overall	100.00%	Macro	100.00%
ItW Overall (o/a)	94.42%	Standard	99.95%
ItW File	100.00%	Polymorphic	97.50%

A product derived from *Kaspersky Anti-Virus*, the similarity in speed for the clean test sets tends to suggest that no huge inefficiencies have been introduced. A major difference does exist, however, that on-access boot sector scanning is absent from the *GDATA* product – or at least not triggerable by any deducible means. From the point of view of detection in files the news was better, with the predictable pair of W32/Nimda.A and all the W32/Zmist.D samples causing the only misses throughout the entire test set.

On-access tests	ItW Boot		ItW File		ItW Overall	Macro		Polymorphic		Standard	
	Number missed	%	Number missed	%	%	Number missed	%	Number missed	%	Number missed	%
Aladdin eSafe Desktop	0	100.00%	2	99.92%	99.92%	34	99.29%	74	92.47%	38	98.07%
Alwil AVAST32	1	91.67%	1	99.51%	99.07%	0	100.00%	43	97.50%	11	99.62%
CA eTrust	0	100.00%	0	100.00%	100.00%	0	100.00%	43	97.50%	1	99.98%
CA Vet Anti-Virus	0	100.00%	0	100.00%	100.00%	0	100.00%	12	99.35%	0	100.00%
Command Antivirus	0	100.00%	0	100.00%	100.00%	0	100.00%	43	97.50%	2	99.95%
DialogueScience DrWeb	0	100.00%	0	100.00%	100.00%	0	100.00%	43	97.50%	5	99.80%
Eset NOD32	0	100.00%	0	100.00%	100.00%	0	100.00%	8	99.53%	0	100.00%
FRISK F-Prot	0	100.00%	0	100.00%	100.00%	0	100.00%	43	97.50%	3	99.89%
F-Secure Anti-Virus	0	100.00%	4	99.72%	99.73%	0	100.00%	43	97.50%	23	99.66%
GDATA AntiVirusKit	12	0.00%	0	100.00%	94.42%	0	100.00%	43	97.50%	2	99.95%
GeCAD RAV	1	91.67%	0	100.00%	99.53%	0	100.00%	51	97.57%	13	99.67%
Grisoft AVG	12	0.00%	0	100.00%	94.42%	0	100.00%	43	97.50%	7	99.67%
HAURI ViRobot	12	0.00%	77	91.25%	86.16%	368	90.37%	10836	35.38%	659	65.11%
IKARUS virus utilities	1	91.67%	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Kaspersky Lab KAV	0	100.00%	0	100.00%	100.00%	0	100.00%	43	97.50%	2	99.95%
NAI NetShield	0	100.00%	0	100.00%	100.00%	0	100.00%	2	99.88%	11	99.02%
Norman Virus Control	0	100.00%	0	100.00%	100.00%	0	100.00%	61	95.47%	10	99.65%
Sophos Anti-Virus	0	100.00%	0	100.00%	100.00%	13	99.66%	234	92.98%	20	99.36%
Symantec Norton AntiVirus	0	100.00%	0	100.00%	100.00%	0	100.00%	0	100.00%	0	100.00%
Trend ServerProtect	0	100.00%	0	100.00%	100.00%	3	99.94%	255	92.87%	9	99.78%
VirusBuster VirusBuster	1	91.67%	0	100.00%	99.53%	4	99.90%	71	92.97%	11	99.70%

GeCAD RAV 8.2.1.12

ItW Overall	100.00%	Macro	100.00%
ItW Overall (o/a)	99.53%	Standard	99.67%
ItW File	100.00%	Polymorphic	97.57%

The testing of *RAV* did not start well since installation did not complete due to errors with Visual C runtime libraries which are required to be particular versions. Some manual fiddling got the process back on track, but the lack of these files in the installation package is a weakness. The process of updating was also somewhat more convoluted than might be expected – doing so from a file was explained poorly in the help files. Matters improved when detection was considered, with 65 missed files out of the whole test set – once more exclusively from the standard and polymorphic sets and including four of the W32/Nimda.A and all but two of the W32/Zmist.D samples. Unfortunately for *GeCAD*, Michelangelo was missed in the on-access boot tests and a grand total of 21 false positives and one suspicious file were present in the clean test set. Although not the most paranoid of this review, this was a sufficient harvest to deny *RAV* a VB100% award.

Grisoft AVG 6.0 285

ItW Overall	99.97%	Macro	99.50%
ItW Overall (o/a)	94.42%	Standard	96.92%
ItW File	99.97%	Polymorphic	89.91%

AVG certainly wins prizes on the on-access boot mystery front – although claiming to have such a feature, this proved to be untriggered in numerous attempts. On demand this did not prove to be a problem, so the capability is in the product somewhere. It managed to produce a smattering of false positives in the clean test set which, akin to the previous product, scuppered *AVG*'s attempt at gaining a VB100% award. *AVG* was also notable in this test for missing files in all of the test sets rather than the more limited selection which characterised detection rates over all products. Particularly surprising was the repeated missing of the .HTA sample of JS/Kak.A which has been in the wild for a number of years.

HAURI ViRobot Professional 3.0

ItW Overall	91.82%	Macro	90.42%
ItW Overall (o/a)	86.16%	Standard	65.18%
ItW File	91.34%	Polymorphic	35.38%

ViRobot distinguished itself by performing very quickly on the clean executable test sets, though some might suggest that this is because it is not really looking for much. Overall detection rates were roughly 50 percent of files, with more misses ItW than can be considered by any means comfortable. On floppy scanning the detection rate was exactly half of all samples since, despite there being full detection on demand, there was no detection on access.

The interface was pleasant enough, but the much-needed improvements have not been made since the last time *ViRobot* was reviewed. The reasoning that there are differing anti-virus needs in Korea from the rest of the world may be applicable here, but will be no great comfort to a western user of this product.

IKARUS virus utilities 5.03

ItW Overall	98.90%	Macro	96.67%
ItW Overall (o/a)	N/A	Standard	95.14%
ItW File	98.83%	Polymorphic	90.73%

This rates as the most over-paranoid of the products on test, with a grand total of 29 suspicious files and five false positives in the combined clean test sets. Its powers of looking for what was not there were not only very efficient but also somewhat time-consuming, making the scan times decidedly slow. Heuristics did prove to be of use in the on-demand boot sector tests, this being the reason for AntiExe's detection, but this did not carry over to the detection of the same virus on access.

Indeed, on-access scanning was something of a nightmare, with no automatic treatment available and those which were available not seeming to perform consistently in the manner they suggested would work. Log files contained large amounts of useless information and were size-limited which, after ten hours of testing, led me to abandon on-access file scan testing for this product. The fragments of data retrieved from logs suggest slightly worse detection on access than on demand, on demand showing large numbers of misses in both standard and polymorphic test sets.

Kaspersky Labs Kaspersky Anti-Virus (AVP) 3.5

ItW Overall	100.00%	Macro	100.00%
ItW Overall (o/a)	100.00%	Standard	99.95%
ItW File	100.00%	Polymorphic	97.50%

Clearly, product recognition is something that the *Kaspersky* folks are concerned about, hence the inclusion of the parenthesised *AVP* in the splash screens of this product. However, naming matters proved the most complex of the issues on hand here, with all tests going smoothly and as expected.

It was mentioned earlier that *GDATA*'s *AVK* and *KAV* share the same engine. Indeed, with only one exception, the detection rates were identical. However, this exception was rather major in that *KAV* showed perfect detection for on-access boot sector viruses. This is the difference that wins a VB100% award.

NAI NetShield 4.5

ItW Overall	99.60%	Macro	99.97%
ItW Overall (o/a)	100.00%	Standard	99.00%
ItW File	99.57%	Polymorphic	99.88%

The *VB* comparative test is often a frenzy of patching of products when testing is about to begin – this time, both a Service Pack and a SuperDat file had to be added before *NetShield* was ready for operation. However, the line was drawn at the inclusion of a suggested scan-all-files patch, since this was hidden away on a section of the *NAI* Web site reserved for patches which should not be applied under normal circumstances.

The result was fairly predictable, in that *NAI* missed out on a VB100% award due to extension-related misses ItW which would have been solved by the patch. The good news is that on-access, where contents rather than extensions are considered, these files were scanned and detected correctly, and all W32/Zmist.D samples were detected. There were also a number of new misses in the standard set of ancient viruses – possibly removed from the datafiles for reasons of space saving.

Norman Virus Control 5.20

ItW Overall	100.00%	Macro	100.00%
ItW Overall (o/a)	100.00%	Standard	100.00%
ItW File	100.00%	Polymorphic	95.47%

Norman Virus Control is one of those products looking for a bizarre niche role – in this case to have no method of

Hard Disk Scan Rate	Executables			OLE Files			Zipped Executables		Zipped OLE Files	
	Time (s)	Throughput (MB/s)	FPs [susp]	Time(s)	Throughput (MB/s)	FPs [susp]	Time (s)	Throughput (MB/s)	Time(s)	Throughput (MB/s)
Aladdin eSafe Desktop	N/A	N/A		26.0	3051.3		484.0	329.4	38.0	1963.4
Alwil AVAST32	290.0	1886.0		N/A	N/A		140.0	1138.7	6.0	12434.6
CA eTrust	293.0	1866.7		21.0	3777.8		101.0	1578.4	22.0	3391.2
CA Vet Anti-Virus	227.0	2409.4		16.0	4958.4		113.0	1410.8	26.0	2869.5
Command Antivirus	204.0	2681.0		24.0	3305.6		97.0	1643.5	14.0	5329.1
DialogueScience DrWeb	310.0	1764.3	[15]	28.0	2833.3		133.0	1198.6	23.0	3243.8
Eset NOD32	95.0	5757.2		15.0	5288.9		21.0	7591.3	4.0	18651.9
FRISK F-Prot	239.0	2288.4		24.0	3305.6		102.0	1562.9	16.0	4663.0
F-Secure Anti-Virus	594.0	920.8		32.0	2479.2		308.0	517.6	102.0	731.4
GDATA AntiVirusKit	270.0	2025.7		39.0	2034.2		136.0	1172.2	42.0	1776.4
GeCAD RAV	612.0	893.7	21 [1]	42.0	1888.9		124.0	1285.6	52.0	1434.8
Grisoft AVG	327.0	1672.6	4 [2]	21.0	3777.8		113.0	1410.8	21.0	3552.7
HAURI ViRobot	100.0	5469.3	[1]	40.0	1983.3		82.0	1944.1	44.0	1695.6
IKARUS virus utilities	2667.0	205.1	5 [17]	51.0	1555.6	[12]	2142.0	74.4	42.0	1776.4
Kaspersky Lab KAV	281.0	1946.4		33.0	2404.1		148.0	1077.1	43.0	1735.1
NAI NetShield	201.0	2721.1		22.0	3606.1		88.0	1811.6	23.0	3243.8
Norman Virus Control	2498.0	218.9		14.0	5666.7		304.0	524.4	25.0	2984.3
Sophos Anti-Virus	132.0	4143.4		20.0	3966.7		78.0	2043.8	21.0	3552.7
Symantec Norton AntiVirus	310.0	1764.3		37.0	2144.2		157.0	1015.4	43.0	1735.1
Trend ServerProtect	211.0	2592.1		19.0	4175.5		102.0	1562.9	30.0	2486.9
VirusBuster VirusBuster	272.0	2010.8		33.0	2404.1		143.0	1114.8	32.0	2331.5

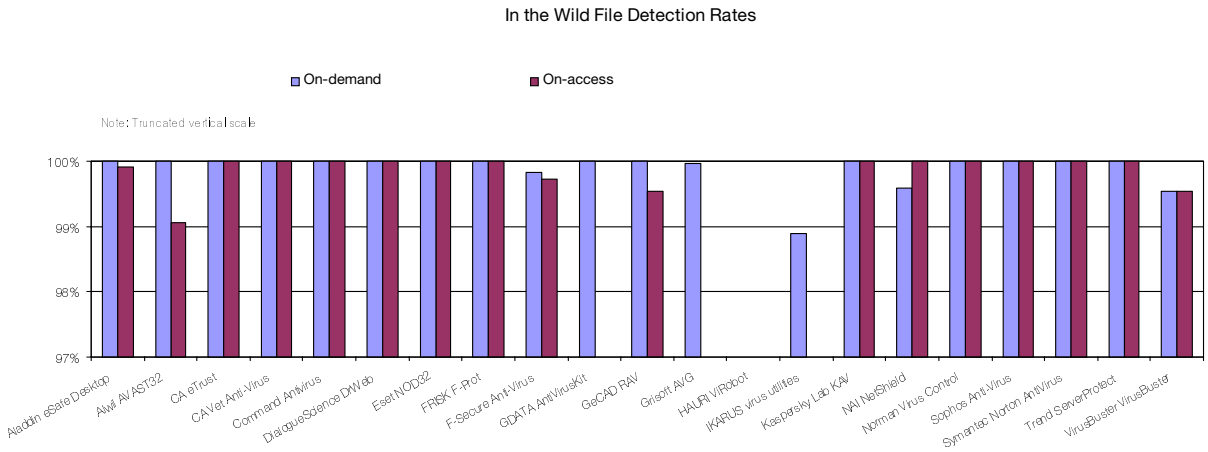
reporting without resorting to undocumented switches in the program. Once the initial disbelief at this 'feature' was over, the testing process was considerably more pleasant. Misses were W32/Nimda.A and W32/Zmist.D with a small selection of extra standard files for good measure. This, coupled with a lack of false positives on the clean test sets, sends *NVC* away with another VB100% award.

There were some problems and, as in the September 2001 *NetWare* review, these were in the length of time taken for the clean executable test set. For the *NetWare* test this has been tracked down to a design decision – gaps in scanning were introduced since server scanning could otherwise be too much of a constant load on a machine which can be expected to have many other duties. The same reason may apply here.

Sophos Anti-Virus 3.50

ItW Overall	100.00%	Macro	99.66%
ItW Overall (o/a)	100.00%	Standard	99.36%
ItW File	100.00%	Polymorphic	92.98%

Putting behind them the matter of extension-related problems, *Sophos* came forward with full detection of all files ItW and receives a VB100% award. Detection rates remained slightly lowered by the choice of extensions that are not scanned by default, and a new addition to the scanning engine is still forthcoming, leaving rather more misses in the polymorphic set than might be the case in a few months' time. The exclusion of extensions from scanning, and the fact that archive scanning is off by default, are for speed reasons, and speed of scanning was indeed good. Reports



proved to be a quirky part of the product, causing problems in parsing until it was realised that long filenames within them are always compressed to 8+3 format. This is at odds with the designated operating system and presumably is retained for backwards-compatibility with older and other-platform products.

Symantec Norton AntiVirus 7.51.847

ItW Overall	100.00%	Macro	100.00%
ItW Overall (o/a)	100.00%	Standard	100.00%
ItW File	100.00%	Polymorphic	100.00%

Since *Symantec*'s Péter Ször is notorious for bringing with him tidings of W32/Zmist.D and its effects upon the future of scanners, it was interesting to see how his company's product bears up when faced with the virus itself. *NAV* detected all the samples of W32/Zmist.D thrown at it. In fact, all samples in all test sets were detected, which left activity in the clean test sets as the deciding factor as to whether a VB100% was awarded. Although on the slow side, the clean tests proved completely lacking in false positives, so *Symantec* add a VB100% to their collection.

Trend ServerProtect 5.21

ItW Overall	100.00%	Macro	99.94%
ItW Overall (o/a)	100.00%	Standard	99.78%
ItW File	100.00%	Polymorphic	92.87%

The installation of *ServerProtect* proved to be slightly odd since there were such lengthy delays that crashes were suspected. Once installed, the logging was slightly problematic too – of a massive log file of some 60 MB, only 1000 lines were actively viewable. These problems were overcome and the results proved no great surprise. The usual combination of standard and polymorphic misses was noted, although with more misses in the polymorphic set than many products. In addition were misses of the polymorphic macro XM/Soldier.A and X97M/Soldier.A, but no ItW misses. On-access testing showed poor change detection for boot sector viruses and it was often difficult to tell when an infection was present. Despite this, the combination of complete ItW detection and no false positives gained *Trend* a VB100% award.

VirusBuster VirusBuster 3.06

ItW Overall	99.53%	Macro	99.90%
ItW Overall (o/a)	99.53%	Standard	99.72%
ItW File	100.00%	Polymorphic	92.97%

The testing of *VirusBuster* threw up a few problems, which were almost exclusively related to how logs could be produced. The results were good however, with standard and polymorphic test sets being the source of all but one of the misses, and a solitary macro miss in addition. There were no misses in the ItW test set, and fast clean set results with no false positives left this contender in a good position to claim a VB100% award. This was not to be, however, since both on demand and on access there were misses of the ancient Stoned.NoInt.A. A disappointing result for the developers, but one which should be easy to remedy.

Conclusion

As expected, a high harvest of VB100% awards resulted from the use of a dated WildList in the testing process. The future looks set to be interesting, however, since extension issues associated with W32/Nimda.A, in the current WildList, tripped up a few here – and there are some companies with a history of problems in the extension field.

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 NT4 Server SP6*. 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/2001/08test_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, reprints, orders and payments:

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

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

US subscriptions only:

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

Tel (781) 2139066, Fax (781) 2139067

END NOTES AND NEWS

The Black Hat Briefings and Training Europe take place in Amsterdam from 19–22 November. For more information, as well as details of other Black Hat events, visit http://www.blackhat.com/.

The First East-West Security Conference takes place 28–29 November 2001 in London. The conference aims to provide a platform for discussion between all those involved in the security industry, and to deepen business cooperation between the East and West. Forums will be held on subjects ranging from information security management and how to fight hacking, to security for financial institutions. For further information visit the organizer's Web site http://www.oecexhibitions.com/security-1.htm.

The 4th Anti-Virus Asia Researchers (AVAR) Conference takes place on 4 and 5 December 2001 at the New World Renaissance Hotel, Hong Kong. For full conference details visit the AVAR Web site, http://www.aavar.org/.

***Sophos Anti-Virus*' two-day training course on investigating computer crime and misuse runs 5–6 December 2001**. For full course details and booking see http://www.sophos.com/.

Information Security World Asia 2002 will be held 16–18 April, 2002 in Singapore. The show will include a wide-ranging exhibition, discussions of the latest security issues and a number of interactive workshops. For further information about the show visit the Web site http://www.isec-worldwide.com/isec_asia2002/ or contact Stella Tan: tel +65 322 2756; email stella.tan@terrapinn.com.

The VI Ibero American Seminar on Security Information and Communications Technologies takes place in Havana, 18–24 February 2002. Topics covered will include anti-virus software, network security, Web security and network remote diagnostics. For more information contact José Bidot: email jbidot@seg.inf.cu.

Infosecurity Europe 2002 will run from 23–25 April 2002 at London's Grand Hall, Olympia. Infosecurity Europe aims to heighten awareness of the commercial importance of secure and reliable access to corporate information. Over 40 free seminar sessions will run over the three days, explaining some of the key issues facing organizations today and the technologies available to address them. For further details see http://www.infosec.co.uk/.

Infosecurity.de 2002, the international specialist exhibition for IT security takes place 14–16 May 2002, in Düsseldorf. For the first time an accompanying Specialist Conference will run throughout the exhibition period. For more details about the exhibition and conference see http://www.infosecurity.de/.

Central Command* states that the Presidency of the French Polynesian Government is using its *AntiVirus eXpert product for AV protection of governmental computers. For more information see http://www.centralcommand.com/.

Kaspersky Labs* has signed retail and distribution agreements with Italian software republishing company *Questar, which will enable the company to sell *Kaspersky*'s software in the Italian market. See http://www.kaspersky.com/.

***Trend Micro*'s *OfficeScan* will be used to protect more than 400 workstations at four EU summits this autumn**. It is anticipated that some 5000 journalists and Ministers' assistants will make use of the workstations over the four summit meetings, in Brugge, Genval, Gent and Brussels. For further details see http://www.trendmicro.com/.

***EMC Corporation* has partnered with no less than four anti-virus companies** to provide anti-virus solutions on its *Celerra* (network attached storage) file server. The *Celerra Anti-Virus Solution* utilizes external anti-virus engines from *Computer Associates*, *McAfee*, *Symantec* and *Trend Micro* to provide 'on-access' anti-virus scanning by checking data and content files for viruses as they are updated. For more details see http://www.emc.com/.

Following discussions at VB2001 in Prague, *GeCAD Software* has announced two new distributors of *RAV*. *NetceNter AG* in Bremen, Germany and *R.A.E. Internet* in New York, USA join the team of *RAV* distributors worldwide. For more information about *RAV* visit the Web site at http://www.rav.ro/.

***Virus Bulletin* has a limited number of VB 2001 conference proceedings CDs for sale**. The CD costs £150 and, while stocks last, will be sent with a free rucksack-style conference bag. So, if you couldn't make it to the conference, there's no need to miss out! Contact Bernadette Disborough at *Virus Bulletin*, tel +44 1235 544034, fax +44 1235 531889, or email bernadette@virusbtn.com.