ISSN 0956-9979 NOVEMBER 2000

VIRUS BULLETIN

THE INTERNATIONAL PUBLICATION ON COMPUTER VIRUS PREVENTION, RECOGNITION AND REMOVAL

IN THIS ISSUE:

• **Beyond compare?** Our Comparative Review attracted no less than 19 products from all over the world in pursuit of the elusive VB 100% award. It starts on p.16.

• **Three reasons to stay alert:** this month's Virus Analyses paint a grim picture of the direction viruses are starting to take. Nick FitzGerald kicks off with MTX on p.6.

• **A scanner in the works:** anti-virus veteran Ray Glath believes that the traditional scanner is no longer sufficient protection from malware. His argument starts on p.12.

CONTENTS

COMMENT

“ First, you blame the pilot … ”

Airport Security vs Anti-Virus Security

Let’s suppose airport security works like AV security – and let’s suppose a terrorist wants to hijack an aircraft bound for the US. He shouts anti-American rhetoric as he approaches the X-ray machine. The airport security team doesn’t notice. They ask to see his passport, which identifies him as ‘Bin saden Olama’. A database search turns up blank. The name matches no known terrorist.

The terrorist places a package on the conveyor. It has ‘BOMB’ written all over it, but the security guards don’t look. They don’t even look at the X-ray image. A bomb-sniffing dog snoozes nearby. Our guy boards the aircraft and pays $4 for a mixed drink. The plane takes off.

The terrorist steals everybody’s credit cards, grabs his $4 back from the stewardess, and puts it all in an envelope. He breaks into the cockpit, shoots the co-pilot, and orders the captain to fly over international waters. He tosses the envelope out of a window and it lands in another terrorist’s boat. Then our guy asks ‘Does anyone have a light for my fuse?’ 47 passengers flick their Bics. Boom! Rescue ships follow a trail of seat cushion flotation devices. So, who gets blamed?

First, you blame the pilot. He – like Microsoft – controls the hardware for the passengers. ‘Pilots don’t care about security,’ the airport security experts scream, ‘those flyboys will do anything they please with no regard for who might get on board.’ Then, you blame the passengers. They – like computer users – want the hardware to do things for them. ‘Passengers are stupid,’ the airport security experts scream, ‘they’ll sit next to some weirdo with ‘bomb’ written all over his luggage. They’ll even light a fuse if someone asks politely!’

Reporters clamour to know the cost of this heinous crime. ‘Based on our most recent survey of airport security guards, we can extrapolate this act of terrorism cost US$6.71 million,’ the airport security experts say. ‘Worldwide, we believe this kind of terrorism alone costs US$20.3 billion each year. But remember! Few acts of terrorism are reported, so the damages may amount to trillions of American dollars.’ Reporters ask what new security measures would help. ‘First and foremost,’ say the airport security experts, ‘pilots must stop giving terrorists a comfortable aircraft seat. They’re in control of the hardware and they should make anti-terrorism a top priority. Uncaring pilots make it easy for weirdos to break into the cockpit and take control of the plane. Second,’ they say, ‘every passenger should look closely at every other passenger. Common sense tells you not to light fuses aboard an aircraft.’

One man stands up to ask why airport security guards let the terrorist get past them. Guards – like anti-virus software – get paid to protect us from terrorism. ‘Our folks did their job,’ the airport security experts protest. ‘They did it quite well, too. They stop terrorists all the time.’ But they didn’t stop *this* terrorist, the man notes. Can we can do anything to increase airport guard efficiency? ‘We *already* increased their efficiency,’ the airport security experts claim, ‘all airports worldwide now look for ‘Bin saden Olama’ passports. And we asked security guards to update their terrorist definition files on an hourly basis instead of daily.’ The man presses for answers. Can’t airport security guards look for packages clearly marked ‘bomb’? ‘They could do so in *theory*,’ the airport security experts concede, ‘but it would take extraordinary training. They’d need to look for ‘bomb,’ ‘grenade,’ ‘explosive,’ and so on. The complexity skyrockets if security guards look for multi-word phrases like ‘claymore mine,’ ‘semi-automatic handgun,’ etc. And that’s just for *marked* packages!’

What about the bomb-sniffing dog? ‘He’s part of our forthcoming *Doggy Immune System*’ one airport security expert explains. ‘We’ve been training him for six years to sniff for bombs after they get past our security guards. We trot him out for a photo-op after every act of terrorism.’ The man sighs. It just seems logical to blame security guards for a security failure. ‘Hey, if you can make a better airport security guard, then more power to you,’ the airport security experts say …

Rob Rosenberger, Vmyths.com, USA

NEWS

Tangled in the CNET

On his return from the *Virus Bulletin* conference in Florida, Joe Wells was horrified to spot a product review on *CNET*, the methodology of which was severely flawed. Not only had the reviewers used simulated rather than real viruses in their testing, they had actually 'modified' copies of the LoveLetter virus, essentially creating two new variants.

Drafting an open letter 'from the anti-virus industry', he gathered signatures and support from respected AV figures and dispatched it without delay. That was in early October – the enigmatic response received is that *CNET* is 'pondering' the issues. To date he has heard nothing further. We hope to follow up this story next month. The letter can be viewed at http://www.wildlist.org ▮

The Current Trend

Thank goodness the season of forest fires raging out of control is coming to an end. Pity we can't douse the media occasionally. Recently, mainstream PC magazines and journals have headlined stories about the Pokemon virus (spotted in late July) that border on hyperbole and have virtually no foundation.

The first sign of a botched job is when no-one knows how to classify this malware – it's been variously reported as a virus, a Trojan and a worm. It seems that one trigger happy journo got the wrong idea and the rest went down like a house of cards.

To *VB's* knowledge, most major anti-virus vendors are playing this virus down, saying that they have not had a single verified report from the field. Those that have are specifying very low numbers. So who fed *Secure Computing* the line for its October news story about Pokemon – 'a slow-spreading computer virus … zipped across the US, reportedly leaving downed PCs in its trail.' Likening it to the 'notorious' LoveBug, *SC* cannot really be blamed for reporting this reported report. And the source? *Trend Micro Japan*, shame on you.

Perhaps *Trend* needs to start being a little more discerning. After all, one highly amused AV researcher mentioned to *Virus Bulletin* that *Trend Micro* is currently reporting the *EICAR* test file as an 'in the wild virus'! ▮

VB2001

Following the success of VB2000 in Orlando, Florida, plans are underway to secure a venue for next year's conference somewhere in Europe. Prospective speakers should look out for the official call for papers – along with details of the proposed location – which will be published here in January 2001 ▮

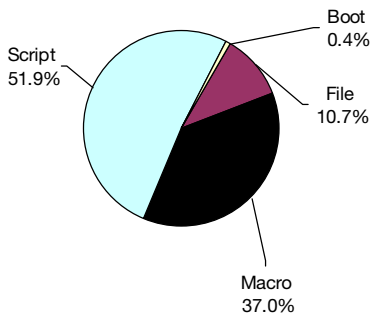
Prevalence Table – September 2000

Virus	Type	Incidents	Reports
LoveLetter	Script	424	30.2%
Kak	Script	211	15.0%
Divi	Macro	100	7.1%
Laroux	Macro	93	6.6%
Stages	Script	86	6.1%
Barisadas	Macro	50	3.6%
Marker	Macro	50	3.6%
Win32/Ska	File	43	3.1%
Win32/Pretty	File	37	2.6%
Win32/MTX	File	36	2.6%
Tristate	Macro	34	2.4%
Ethan	Macro	27	1.9%
Thus	Macro	21	1.5%
Class	Macro	20	1.4%
Metys	Macro	17	1.2%
Melissa	Macro	15	1.1%
Myna	Macro	13	0.9%
Win32/QAZ	File	13	0.9%
Jini	Macro	12	0.9%
Eight	Macro	11	0.8%
VCX	Macro	8	0.6%
Win32/Funlove	File	8	0.6%
Others [1]		73	5.2%
Total		1402	100%

[1] The Prevalence Table includes a total of 73 reports across 32 further viruses. Readers are reminded that a complete listing is posted at http://www.virusbtn.com/Prevalence/.

In order to avoid a distortion of the figures, data for the 'self-reporting' W97M/ColdApe virus (totalling 577 reports in September) have been omitted from the table this month.

Distribution of virus types in reports



LETTERS

Dear Virus Bulletin

Bigger and Better

Just wanted to drop a quick note to you in thanks for a great conference this year. It was only my second, the first having been Vancouver last year and both have proved to be informative. It's amazing to see that the anti-virus companies, whose marketing folks are openly critical of each other, actually have 'techies' who are prepared to gather around and praise each other's products in varying areas (at the same time as promoting their own, of course).

For a 'little' magazine, you sure have the greatest size in respect. I've moved on to Dallas now for a *Microsoft Exchange* conference, but the venue I just left is going to be hard to beat, and the information I learnt I am already using in meetings right here at my company's head office. So once again, thanks for a great time, both relaxing out of the conference and informative in it. Can't wait for next year.

Dave Doohan
Perot Systems Europe Ltd
UK

Lingering on Linux

I was fascinated by the letters from Eddie Bleasdale and Graham Cluley in September's issue. As I have just returned from the *VB* conference, where there were two talks on Linux and viruses, now seems like a suitable time to respond.

There is, contrary to Mr Bleasdale's belief, nothing inherent in the design of Unix (or *Linux*) such that 'only authorised software runs on a correctly configured and administered *Linux* computer'. Once I have a shell account on a Unix machine, I can run any software I can find or produce locally, or obtain remotely. One of the simplest script viruses of which it is possible to conceive is as follows:

```
for file in * do
cat $0 >> $file
done
```

This code has been used as a demonstration in security literature for many years, and can be found all over the Internet in such documents. Mr Bleasdale is welcome to try this code for himself (on an isolated machine, just to be on the safe side). There are obvious bugs with the above implementation, but placing it into a file called 'vir.sh' and executing with 'sh vir.sh' should be enough to prove the point (ahem).

Moving on from trivial script viruses, consider the cases of Lin/Bliss, Lin/Obsidian, Lin/Mandragore, Lin/Siilov and Lin/Vit. All of the above are *Linux*-specific ELF-infectors, and make use of knowledge of the ELF file format to infect standard *Linux* executable files.

I am confused to read on to the next section, in which Mr Bleasdale goes on to say 'The challenge for Sophos is to send an email with attachments. This will be read and the attachments opened.' This is of only tangential interest to the main point, but Mr Bleasdale seems to have moved on to the question of worms. I suggest that he investigates *StarOffice* (http://www.sun.com/staroffice/) – Jakub Kaminski's excellent VB2000 paper 'Linux Malware – Has The Next Battlefield Been Decided?' (which I have been using as source material as I write this letter) has a fine discussion of the many similarities between *StarOffice* and *Microsoft Office*, including the existence of a powerful scripting language and auto-macros. And we all know what *that* means …

Whilst I appreciate Mr Bleasdale's enthusiastic support of, and devotion to, his platform of choice (which, incidentally, is a platform with which I work every day), it is a shame that the reality is not quite as clear-cut as he implies. Whilst it is certainly true that *Linux* users currently are currently at a low risk of getting a virus on their platform of choice, this is due to *Linux's* still relatively low penetration into the corporate desktop market.

It would certainly be true to claim that there is currently no fast-spreading, automatically executing virus/worm for Linux-based computers (like the Kak worm for Win32, for example). This is a much more specific claim, however. In addition, readers should not loverlook the word 'currently' above – the existence of *StarOffice* is a taste of things to come.

Ian Whalley
USA

Get Well Soon!

Best wishes to Paul Ducklin (and Kim) for a speedy and uneventful recovery. We missed you both at VB2000.

Jimmy Kuo
NAI
USA

Who's Afraid of a Naming Convention?

Even though some AV products were not using an 'international standard industry naming scheme', I was convinced when I began to work in the anti-virus industry that a Virus Naming Convention existed. I stuck with this idea for many years – imagining that everyone in our industry recognized some of the existing *CARO* proposals (even if some did not apply them).

Working on a naming convention for the mass-mailers, and discussing things with some outstanding AV researchers, I discovered the truth:

- 'Our position (the *WildList Organization*) on naming is that there is *no* "correct" name for a virus.' [Sarah Gordon.]
- 'Several "standard" methods of virus naming exist. In other words, there is no standard.' [Joe Wells.]
- 'What was done was an attempt to standardize the naming of new viruses. To that purpose a generalized naming strategy has been agreed on and some attempts have been made to set up committees to decide virus names. According to us, the committees never worked because of individual egos and paranoia.' [Robert Sandilands.]

In 1991, a virus-naming scheme was proposed by *CARO*. Updated only once (in 1993), this document reappeared in 1998 when Gerald Scheidl wrote an article named 'A New Virus Naming Convention' and proposed improvements to the existing scheme. Unfortunately, and mostly due to lack of time, that *CARO* document has still not been updated. According to Gerald, the 1991 document 'was the first and only effort to set a naming standard'. However, I have come to realize this statement is not completely true. In 1996, *CARO* and, in particular, Vesselin Bontchev, updated the scheme to encompass macro viruses, and those proposals are what we use today.

The last publicly available *CARO* document lacks a lot of information, as it predates macro viruses, script viruses, and the various virus/worm families that have overtaken the world. Many new malware types have appeared at the 'virus' definition border, and many prefixes can be envisioned (DDoS or FDoS – Flood Denial Of Service; FLAT – FLood Attack Tool; RAT – Remote Access Trojan; *Palm OS*, etc). I hope my crusade for an additional notation on mass-mailing viruses (@m suffix if one message sent per action; @mm suffix for multiple recipients at once) opens the door for more advances.

Finally, this letter has two purposes. Firstly, I respectfully ask *CARO* members to continue their efforts and update the document which outlines the different parts of the naming scheme. Secondly, I encourage the AV companies to make use of this additional notation on mass-mailing viruses. I know that adding these suffixes for all existing 'mailers' and 'mass-mailers' will be difficult. So, we should do this work where it is important: when the viruses are in the field. I thank the *WildList Organization* for understanding that fact.

The use of email is a very important property of a virus. It is a primordial piece of information that has to be easily accessible to MIS people. It means that it can spread much faster and wider. With these suffixes, we incorporate a level of 'seriousness' and 'risk assessment' directly into the name. 'Medium Risk' and @mm means the virus could explode at any moment. With these suffixes, this information can be extracted from any report – much easier and faster and no need to search through an encyclopedia. People sharing this opinion are welcome to engage in this new common effort.

Francois Paget
Network Associates Inc
France

Action This Day

The late September *SANS* Alert 'Virus scanner inadequacies with NTFS' has fanned old flames, and people are asking about ADS again. The description of what an ADS is and the examples of creating them are accurate, if not twee. It is correctly stated that the information in an ADS is lost if copied onto a FAT partition/drive. This is a fundamental piece of information when looking at a virus threat. Just as DBR viruses cannot infect NTFS so a true NTFS infector utilising ADS cannot infect FAT machines. To facilitate its spread and be a good vector for its transmission, the virus must be aware of FilingSystems and have a FAT part.

Even if the AV product you use does not understand ADS, it will see the FAT part with no difficulty. If it sees this FAT, it can detect it and, once detected, it can ascertain what would be infected by the virus. The process may not be simple, but it can work. The alert mentions that an on-access scanner may detect the code when it is accessed. Well, that is why we call them on-access scanners I suppose. This, however, is a bad thing because the on-access scanner may be set to move these files – how is this different from the action of an on-demand scanner aware of ADS? The problem they describe is one of correct software configuration not of ADS or general virus scanner faults. The user of AV software is given the option to do many things – the product should default to 'report' or in the case of the on-access part 'report and deny access to'.

Will the knowledge of ADS provide new functionality to be exploited by virus writers? Yes, it has already and will continue to pop up on the odd occasion. The virus writer may do more research into ADS than the authors of this advisory. They made mistakes with regard to the execution of ADS that should probably not be pointed out in public.

Then the authors decide to lecture us (both the AV industry and user). The industry is told to scan ADS with their *NT* products. The user is told to begin to baseline the ADS files on their systems. If the user finds a suspicious ADS they are told in a very roundabout way how to remove it. I suggest that if you find a suspicious ADS you contact your anti-virus vendor to ask for instructions on how to send them a copy. Finally, we are told to request support for data streams from our anti-virus vendors. If this is done I suspect some people will go off half-cocked and implement bad support. Does this need to be 'actioned this day.'?

Paul Baccas
Sophos Plc
UK

VIRUS ANALYSIS 1

MTX-treme

Nick FitzGerald
Computer Virus Consulting, New Zealand

Another binary *Windows* virus seems to have caught a 'lucky break'. Win95/MTX@m is now being reported extensively in the wild, but it throws no light on the elusive nature of that lucky break. As with so many 'successful' viruses in the last couple of years, MTX is network-aware and depends on both naïveté and risk-taking in its victims.

In fact, MTX provides little by way of innovation, mainly following recent trends rather than setting new ones. Perhaps the combination of functions is its hallmark? It includes anti-anti-virus techniques, patching common OS components to go resident, 'tagging along' with legitimate email, and network-aware updating of the virus. Also in keeping with its forebears, MTX's success partly depends upon the over-liberal configuration of many of its potential target systems. MTX is the first virus to take advantage of a previously little-known trick with PIF files.

Happy 2001?

MTX could be an early applicant for 'Happy 2001', as its functionality has much in common with Win95/Ska. Like Ska, MTX patches the main Winsock component to send copies of itself where its victims send email messages. However, unlike Ska: it does not record those addresses; it installs a 'network updater' module, and; it has a parasitic infection function, infecting other files. Thus, aside from files it infects on its hosts, MTX code will be found in some files of its own making and in its patch to WSOCK32.DLL.

Most victims of MTX have become infected by running the virus-infected program that MTX distributes via email, so we will follow such an infection incident in describing MTX's workings. As with Ska, the messages carrying the virus attachment 'accompany' messages victims send. From the perspective of the next potential victim, these messages arrive contemporaneously (within the whims of the Internet's routing of SMTP messages) with a 'normal' email from an earlier victim of the virus.

The assumption that the potential victim has a degree of trust in email from people from whom they usually receive messages is a social engineering trick successfully employed several times since Melissa. The relative success of MTX to date suggests this trick still has a fair degree of potency, despite increased caution toward email attachments. However, the writers of MTX knew that sending .EXE file attachments would dramatically reduce the spread of their virus as many corporate email systems reject messages with such attachments, or remove the attachment and just deliver the message part of the email.

What MTX's writers knew, or presumably hoped for, was many of those content filtering measures are easily fooled due to common, poor configuration. It is good security practice to block everything not known to be necessary and only allow the truly necessary. Much painful experience shows that anything else results in bolting the door after the proverbial horse has bolted, and results in increased work, loss and damage. The issue here is that many content filters block .EXE file attachments, but not .PIF attachments. Why does this matter? Because many of those filters are based solely on the extension part of the attachment's filename, rather than on the file's content. Most of MTX's attachments are .EXE files renamed with .PIF extensions and they work just fine under *Windows*.

A Many Chambered Thing

As already mentioned, MTX has several components. There is the parasitic virus code itself, which runs from infected host programs. The virus code carries all the other parts, in compressed form, within itself. Those parts are the email worm installer, dropped to files named IE_PACK.EXE and WIN32.DLL, and the network updater, dropped to the file MTX_.EXE. Neither the worm installer nor the updater directly pass on the virus, but the worm sends WIN32.DLL as the attachment to its email messages.

To ensure the full virus is sent with the worm's email, the virus deliberately infects IE_PACK.EXE and WIN32.DLL when it drops them. To confuse the issue slightly further, although the contents of WIN32.DLL are sent as the attachment to the worm's email, the filename seen in those messages is something different. As the worm creates the whole message, rather than depending on any particular email client, the worm has direct control over the name in the attachment encoding headers of its messages. It uses different names depending on the day of the month.

Thus, the file that arrives in a potential victim's inbox is an MTX-infected copy of MTX's worm component installer. When that file is run, the virus eventually gains control, drops its worm installer and updater components, infects the former, runs the worm installer and network updater, then returns control to the host program.

Half my Kingdom for a Platform

MTX is, correctly, a *Windows 9x* virus. *NT* and *Windows 2000* will not load the infected form of IE_PACK.EXE (nor other MTX-infected PE files). Several vendors have dubbed it a Win32 virus, probably because it has code that checks the appropriate *NT* and *Windows 2000* kernel addresses (as well as *Windows 9x* ones) for the functions it needs to use. That code should work, but as the EXE loaders of *NT* and *Windows 2000* will not load the code, the point is moot.

MTX uses an entry point obscuring (EPO) technique. It does not modify the entry point of its hosts, nor code at or near their entry points. This is an anti-anti-virus method, although most scanners have long since dealt with EPO. MTX modifies a small part of the host's code, replacing a call to an imported function with a 'jump to virus' instruction. This means the virus does not necessarily run each time an infected program runs – whether it does depends on the program's logic flow branching to the overwritten location. Apart from inserting the call to its code, MTX copies itself to the end of the last section in the host and it makes the minimal changes to the PE header necessary to have the program load under *Windows 95*.

Once MTX gets control, a short routine decrypts the rest of the virus' code. This code tries to find the memory address of GetProcAddress from the current process's import table. If this fails, control is returned to the host by resetting the registers and flags to their state before the 'jump to virus' occurred, and jumping to the function that was originally called at the point the 'jump to virus' was patched. A bug here may make the virus' presence felt. If it fails to find GetProcAddress, the virus exits to the host before patching its code to prevent the decryptor running. Thus, if the host's execution flow should happen back to the 'jump to virus' location, the virus will 'decrypt' its already 'plaintext' code then jump into it. Following this, the virus' body is semi-random 'junk' and an IPF or GPF rapidly ensues.

Assuming MTX finds GetProcAddress, it gets the addresses of a list of functions it needs then patches the entry point to its own code to skip over the decryption routine (and thus avoid the problem described above). Next it unpacks the worm installer from its body, writes it to IE_PACK.EXE and copies that to WIN32.DLL, both in the *Windows* directory. These files are infected and the first is executed.

MTX_.EXE – the network updater component – is then decompressed from the virus' body, written to the *Windows* directory and executed. Should this progress without error, the virus then checks *all* files in the current, temporary files and *Windows* directories for suitability to infect. It checks each file to see if it begins with a 'MZ' EXE marker, is not an exact multiple of 101 bytes, and imports at least twenty functions. Files that meet these criteria are infected. Thus, even though MTX is a direct-acting virus, it can rapidly infect many files. On a fairly standard *Windows 95* test machine with *Internet Explorer 5* installed, I soon had files with the following extensions infected — AX, CPL, DLL, DRV, EXE, MPD, OCX, SCR, VWP and X32 (I did not know what they were all for!). This will raise detection and disinfection problems for users of scanners not set to 'scan all files'. When the virus has checked those directories for possible new hosts, it returns control to its own host.

Socket To Me, Baby …

To avoid problems with the file being locked while in use, MTX's worm installer first copies WSOCK32.DLL to WSOCK32.MTX. The virus then patches its code into the latter, hooking the 'send' function. At the next restart, the original WSOCK32.DLL is deleted and the patched DLL is renamed to replace it. This is achieved via commands the installer writes into WININIT.INI.

Once the patched WinSock DLL is loaded, the virus' code monitors SMTP connections and sends its own messages to the addressees to whom its victim sends email. The virus' messages are very minimalist, containing very few headers, no Subject: line and no message body. They are simply carrier mechanisms for the attachments – if you think this should be warning enough to be wary of the attachment, you are not cynical enough… Unlike Ska, MTX includes its attachment using MIME-encoding rather than UUencoding.

This MTX component also introduces its most interesting feature and the only really novel idea in the virus. It acts as a form of 'personal firewall', although not in a manner many would find endearing or useful – it blocks http and email access to the major anti-virus developers. This is achieved by string-matching partial domain names against the requested URL or email address. With email, if the virus finds a match it not only prevents the message being sent but crashes the email program trying to send it.

Up to Date Viruses

MTX_.EXE is the network updater component. I spent very little time analysing this as the site it attempts to connect to was closed soon after MTX's discovery two months ago. It appears to be designed to check a URL for a list of files and to download and execute whatever it finds in the list. This would allow the virus' writer to provide updates or installers for unrelated malware.

Win95/MTX@m

Aliases: I-Worm.MTX, PE_MTX, Win32/Matrix, W32/Apology, and variations thereon.

Type: A multi-component *Windows 9x* PE infector with self-mailing and updating parts. While the virus is direct-acting, the updater and emailer are resident.

Self-recognition: File size is an exact multiple of 101.

Payload: None.

Removal instructions: Identify infected files and replace from clean backups or original sources (ensure your scanner checks *all* files). From DOS, remove the updater (MTX_.EXE), remove the copies of the infected worm installer (IE_PACK.EXE and WIN32.DLL — the files are hidden) and replace WSOCK32.DLL with a copy from a clean source.

VIRUS ANALYSIS 2

Looping the Kloop

Costin Raiu
Kaspersky Lab

'Unpack the source and look for a scan string', they said. When the first *Office 97* macro virus was reported a couple of years ago, anti-virus researchers all around the world rushed to their labs and pulled all those hacking tools out of the box again in a quest to uncover the mysteries of yet another virus-friendly *Office* platform. Those of them who had the chance to ask the almighty *Microsoft* for some hints regarding the new file format got the rather informative and powerful response with which I started this article.

Well, for at least two different reasons not everyone followed that advice. For example, I was part of the unlucky crowd which did not get any hint from *Microsoft*. That is why I first implemented something which nowadays might sound silly. More precisely, after looking at some different samples, I noticed that in all the samples the compressed source looked exactly the same. So, I picked a scan string from the compressed source and used it to detect the respective virus.

However, after a few days of success, the method proved to be very wrong. I got another sample of the same virus in which, unfortunately, the compressed source looked different. Now I know that was caused by different parameters in the ATTRIBUTE statements of the compressed macro source, but back then it only meant more trouble to me. So after a while, with a little bit of help, I implemented a decompressor, unpacked the source, removed the ATTRIBUTE lines, then computed a CRC on the macro.

This worked like a charm, until another problem appeared. In order to find the offset of the compressed macro source inside the module, I went to offset 0xD0 in the macro module, read the value stored in there, then did a 'seek' to that offset, and unpacked the data found there. As I was saying, another problem appeared when I got a sample in which the offset to the compressed macro source was not stored at offset 0xD0 in the macro module! Resuming the hacking work, I went to the stream named 'dir' inside the macro storage, uncompressed it with the routine I used earlier to decompress the macro itself (the compression algorithm was the same), parsed the contents of the 'dir' stream and managed to obtain a reliable way of getting the offset to the compressed source.

At this point, I thought I had a pretty good detection engine, one that was able to parse all the samples I had, and all the new things I was receiving – so I assumed this would be the end of the story. However, as I should have by now expected, when everything seemed fine, yet another problem appeared.

Incredible as it may sound, inside *Office 97* files the actual code that was executed by the *Office* Visual Basic interpreter was not the one from the compressed source, but something else, which looked very much like the opcodes of *Excel 95* macros – the so-called 'pcode'. So, one could very well wipe the compressed macro source, and this macro would still work. Not only that, but the compressed source would be regenerated dynamically from the pcode.

Once again, my 'so far so good' engine proved to be incomplete. More precisely, it was not detecting the virus, but a 'shadow' of it, which may very well have been missing from the module. Blessing the wisdom of the wizards from Redmond, I went back to the hacking tools, offered some silent thanks to a guy named SEN from Russia, and with some extra help from friends I wrote a parser for the *Office* macro modules. I managed to create a pcode parser, and eventually obtained some kind of detection for macro viruses using the thing which seemed to be the real form of the macro virus – the pcode.

All was fine, until a while ago, when an angry customer sent me a sample of a Laroux variant which had the pcode and the source wiped out. Nevertheless, the thing was able to drop the PLTD.XLS template in the *Excel* startup directory. I learned that in some cases, the virus is not executed from the pcode and it is not executed from the compressed source either.

Actually what is executed (if present) is something called the 'execodes' form of the virus, which is stored in a couple of streams with names like '__SRP_x'. Again, frustrated with my current macro engine code, I coded a small routine to detect that particular anomaly – which incidentally, was caused by an anti-virus product which I will not name here – and hoped that someday I would be able to implement a proper detection routine for macro viruses.

So, what exactly is executed after all? The answer, as a good friend of mine would say, is not so simple, and is quite tricky. For example, let us look at a sample of a *Office 97* macro virus which contains the execodes. When it is loaded in *Office 97* the execodes will be executed. If the respective sample is loaded in *Office 2000*, the compressed source will be the originator of the code which gets executed. However, if we have a sample without execodes, but with valid pcode and source, and the sample and the platform are directly compatible (I mean, they are both *Office 97* or *Office 2000*), the pcode will get executed. Confused? You bet!

The main problem which results from this paradox of having three totally different forms of a VBA macro is, of course, related to the detection of macro viruses. For example, some anti-virus products have implemented the detection of macro viruses using the compressed source.

This method has the advantage of compatibility, meaning that the same code can be used both for *Office 97* and *Office 2000* without any code changes. So far, so good.

However, what if the source code is trashed? Unless loaded in a different *Office* version from the one used to create it, the macro will work perfectly, and *Office* will not even report an error. So, other anti-virus programs have implemented detection using the pcode, which causes the slight compatibility problems of converting the pcodes from *Office 2000* to *Office 97*, and so on. Now, regarding execodes, I am currently aware of only a few products which are able to detect viruses this way. And all of them do this only for a very restricted set of viruses.

The W97M/Class.EZ Virus

One Monday morning I received a sample of a new macro virus. Why are Mondays always associated with problems? This virus sure looked like trouble. The problem revealed itself when, after replication, I ran my CRC extraction tool and it reported an error while parsing the pcode in the sample. Interestingly enough, the source-based CRC extraction part completed successfully.

At this point, I had no suspicions, and I cursed again all those secret and undocumented opcodes which were probably tricking my parser. However, there was more trouble ahead when I ran a specific tool called F-VBACRC which I use to report and classify new viruses as part of my role in an international macro virus discussion forum. Out of the ten different plug-in modules used by the tool, only three of them provided some output. With some dark thoughts, I took the virus source which I had briefly analysed, and proceeded to do some real research.

W97M/Class.EZ (also known as W97M/Kloop.A) is a class infector, quite similar to thousands of other macro viruses I have seen before. The tricky part, however, is an executable stored inside the virus, named KLOOP.EXE which is dropped during replication and subsequently run on the sample which is currently infected by the virus. The virus itself is not very complicated but this executable took some considerable effort to figure out.

Analysing the samples infected with this virus, the first thing I noticed was that the VBA project version was 0x89 in one of the samples, 0xa8 in another, and 0xe1 in the last one. Usually, for *Office 97* macros, this is 0x5e, but apparently something messed this version number during the replication of the virus. Rather odd. Also, I noticed that all the samples had an invalid pcode 'line table', which appeared to be wiped with zeroes.

Since the executable inside the virus seemed the only reasonable explanation, I blew the dust off my old IDA installation, and started analysing the file. The Win32 executable KLOOP.EXE first attempts to open the file provided as a command-line parameter, then it initializes the internal library random number generator. Next, it determines the size of the file provided as input, allocates a chunk of memory, and reads the file in there. Then comes the ugly part. The 'patching' component of the virus searches for two particular scan strings, both four bytes long, and sets a random value for the VBA project version of the document, then also wipes 24 bytes from the start of the pcode line table.

Usually this is enough to make the pcode invalid, and force *Office* to load the source instead of the 'pcode'. Quite ugly, I repeat.

On the other hand, the method used to patch the respective data structures is rather brutal – no parsing of the OLE2 file is performed, and the method is also likely to cause a lot of problems, even damaging documents during this operation. Eventually, the executable writes back the stream to the document, and exits.

From this point on, there is little to be said about the virus. It does not even delete C:\KLOOP.EXE, and does not care to hide its tracks by wiping the C:\KLOOP.DAT image of the macro which was used during replication. If it were not for the Win32 executable inside the file this would have been a rather uninteresting and ordinary virus.

The Solution

Hopefully, nowadays, many anti-virus products have the ability to scan the source to detect a macro virus. Also, since this virus does not wipe the entire line table, the remaining few entries can be used in order to extract some pcode which can later be used to detect the virus. So, from the detection point of view W97M/Class.EZ will not pose a big problem for most AV products.

However, I still wonder how the virus writer was able to figure out all the information required to write the virus, and if he really did it to cause problems for scanners which happen to use the pcode in order to detect macro viruses. I mean, this kind of information is extremely hard to obtain, and I cannot imagine a virus author figuring out all those tricky formats all by himself.

On the other hand, the patching method is very silly, and if the author knew so many things about OLE2 and VBA macros why did he not use OLE functions to patch the respective streams instead of brute-force scanning for signatures in the OLE2 file? Maybe he disassembled a particular macro engine, and tried to implement some changes in the OLE2 file to avoid detection? Lots of questions, very few answers …

W97M/Class.EZ	
Aliases:	W97M/Kloop.A.
Type:	Macro class infector.
Detection:	Use any good, updated AV product.

VIRUS ANALYSIS 3

Just a Phage?

Jarno Niemelä
F-Secure Corporation

At the end of September 2000, the first real virus for any PDA system was found. This virus is known as Phage.

It runs on the *Palm OS* platform, and while being very simple and unlikely to spread, it still fulfils all the characteristics of a typical virus.

Palm/Phage is a trivial, overwriting direct action virus, which seems to be written in C. The host file actually still contains the compiler-generated debug information. The virus body contains only the replication mechanism, and carries no payload.

Phage is best classified as a 'proof of concept' or experimental virus, which shows that it is possible to create a virus for the *Palm* platform. And while it can cause no danger in and of itself, Phage may spur other virus writers to create new viruses for *Palm.*

The virus runs on any version of *Palm OS* operating system and also on any device that uses *Palm OS*. Thus, the virus does not just affect *Palm* computing devices, but also *IBM Workpad*, *Handspring Visor*, *Sony CLIE*. Any other device which uses *Palm OS* is also susceptible to the virus.

Execution

When an infected application is executed the user will see a blank screen for a couple of seconds and the application returns to the main launcher screen (desktop).

During those seconds Phage infects all the files in the device, and if the user tries to run any other installed applications they will show a blank screen and exit.

Palm/Phage infects all applications which are located in the device RAM, but does not affect any ROM applications – so it seems to the user that while the installed applications are broken, all built-in applications work normally. If the user HotSyncs with a workstation, the infected applications will overwrite any potential automatic backups on the workstation.

Direct Action

When an infected application is launched, Phage infects all applications in the device. The infection mechanism is very simple; instead of inserting itself into the host application code segment, Phage replaces the entire code segment with its own, thus destroying the host. In a sense, Phage does not actually infect the host, it just carves the host into an empty shell, where it then inserts itself.

As Phage replaces the code segment in the host programs, the infection can easily be seen as the application size changes dramatically. Usually the application file size reduces by several kilobytes, with the exception of very small files in which the file size may increase. Also, all infected applications have identical code segments.

The blank screen seen by the user is a side-effect of the simplicity of the virus. As Phage does not display any forms (application screens in *Palm*) the user will see only a blank screen during operation. A blank screen is also shown by the two currently known *Palm* Trojan programs, Liberty and Vapor.

Will Phage Ever Go Wild?

Given the fact that the Phage virus appears to 'kill' its host, it is very unlikely that any user might transmit the virus to anyone else unintentionally. Fortunately, so far there have been no cases of anyone spreading the virus intentionally, so it is unlikely that Phage will ever get in the wild. Having said that, *Palm* users are very active in swapping applications between each other either by beaming directly or through email. And almost no one keeps the original distribution files, so when someone asks for a copy of some handy application a user will either beam it or email it from *Palm* desktop backup directory.

Conclusions

While being very simple in its operation and so obvious that it has no chance of spreading, Phage is still the first of its kind and it is very likely that more will soon follow. The *Palm* community still respects assembly coding, and there are plenty of assemblers and disassembly tools available. So, anyone wanting to create non-trivial viruses for *Palm* will find the necessary tools and information easily.

Like every first virus on a new platform Phage is a 'proof of concept' virus, and most probably the virus situation for the *Palm* platform will follow the same pattern as for other platforms. So, more sophisticated viruses will follow, and we should get ready for them.

OPINION 1

Chopping Off the Tail, Again

Peter Morley
NAI, UK

My previous article (see VB, September 2000 p.8) started discussion of how to maintain the efficiency of anti-virus software, by removing detection and repair of old malware and viruses in which no one is interested any more.

I pussyfooted around, and discussed several categories, drawing debatable conclusions, but I ducked the main issue: how do we set about removing OFFVs (old-fashioned file viruses) which have been dead for many years?

My conclusions, rethought and slightly reworded, were that we cannot remove *anything* which will lead to field queries from customers who have kept copies of the viruses which have infected them. Furthermore, we cannot remove *anything* which will cause reviewers to mark us down.

The first point is easily handled, because we know which viruses are which. The second cannot be handled at all, because we do not have an updated list of what each reviewer uses for testing. And we never will. So, I then came to this conclusion: we need to persuade reviewers to stop testing against old-fashioned file viruses!

The case is overwhelming. Look at the last few issues of *VB* and you will be pushed to find any mention of them. The death is inevitable, even if you say 'Not yet, Peter!'. Can I use the argument 'If it's inevitable, the sooner we do it, the better'? Whilst that argument is not entirely logical, I have used it before. The strongest argument of all is the fact that Technical Support units are not getting calls about them. This should be a clincher.

Popular magazine reviewers are not a problem. They do not have proper virus collections, so they do not review detection and repair capability. That leaves *Virus Bulletin*, *Secure Computing*, the *ICSA*, VTC Hamburg, and the University of Tampere. Reviews from any of them have a noticeable effect on business. I believe they will all agree, eventually, if not immediately.

But …

There is also the question of Far Eastern reviewers. These include the Chinese Government, and they may still have lots of old-fashioned file viruses in circulation. I dare not remove anything which will cause them to mark us down, because we are now taking world-wide marketing seriously. So, while I believe the Americans, the Europeans and the Scandinavians will agree readily (even if not immediately) we still cannot proceed until the Far East situation is understood and resolved.

Will we have to wait till our anti-virus products start to wither and die? Luckily, since my first article, there has been a development, which will give us time! It may give us so much time, in fact, that the problem can be put on the back burner.

The Misery Test

You may be aware that all our detection and repair capability is kept in a single data base, which, when it is compiled, produces three files. The files, FIND.DRV, NAMES.DRV, and CLEAN.DRV, are used with all our products. The Misery Test consists simply of doubling the size of the data base by including everything twice, and seeing if the engine still works.

Up to and including our latest engine in the field (4070), it has never worked, and I had almost given up. However our 4100 engine, due for release before Jan 2001, works fine! (Flushed with success I tripled everything, and tried again. The compiler worked fine, the virus count was correct, but the engine crashed. So be it.)

Now the Misery Test works, I can run it every three months, and scream my head off if it ever fails, rejecting any new engine, until it gets fixed, and demanding suitable maintenance if it is the current engine which fails. I may even get QA to run this test.

The tests were run on a 700MHz AMD *Athlon*, running *Windows 98* (Second Edition), from the primary DOS prompt. I used the C: partition, which was (I believe!) completely virus-free, but does hold a number of fairly complex tools and utilities, as well as all the baggage used by *Windows*.

I used *ScanPM*, and database 4097. The results were:

i) Now – 54502 viruses – 1 min 36 secs – 100%
ii) Double – 109004 viruses – 1 min 52 secs – 116%
iii) Triple – 163506 viruses – crashed

The virus count is currently rising by less than 500 per month. At that rate, we have nine years before disaster strikes. If we halve it, because we will be detecting more and more complex Trojans, we still have four years, which should enable us to deal with the problem. In this period the 2 Gigahertz processor may well have appeared, which could postpone it further.

The Final Solution?

It could even be that the development in Ray Glath's October article (see p. 10) will start to appear, but do not hold your breath! We do not need to cut off the tail yet, just trim a few of the long hairs!

OPINION 2

Playing the Odds

Raymond M Glath, Sr
PCsupport.com, USA

In the October issue I wrote that we needed new technology in the AV industry to *prevent* virus infections (see p.10). And last month's conference really showcased the current state of 'same old' AV technology … Scan, scan, scan.

The only 'new' technology presented was *Symantec's* 'Script Firewall' which really turns out to be the old 'Behaviour Blocking' technology we were using in 1988 and 1989. Give them an 'A' for effort, though. Just because technology had been used in the past with varying levels of market acceptance and effectiveness does not mean it should not get a second chance. And the approach being taken by *Symantec's* Mark Kennedy sounds like it has been well thought out and that it will be effective in its mission of dealing with threats from rogue scripts while still being 'friendly' enough for practical use in the business world. At least *Symantec* is building *something* in addition to more scanning algorithms. The only bad news I see with this effort is that the product is still in development stage, with no release date established as yet.

Nick FitzGerald's presentation strongly made the case for the AV community to take a new look at the possibilities for utilizing 'Integrity Management', given the overall advances in computing power and resources that have been developed over the years. Sadly though, it appeared to me that AV developers seemed to think that integrity management was old technology and not worth a new look.

This kind of 'head-in-the-sand' attitude must be overcome or we are destined to spend the rest of our lives going from one epidemic to the next while uttering the same old excuse: 'The users didn't have the latest update installed'.

If you look at the technology that has been patented by the AV community over the past decade, you will see that, by far, the bulk of the R&D efforts were put into methods to detect viruses through scanning. Think of the kind of technology we could have today if just 10% of the R&D revenues spent by AV companies to develop and refine (and refine, and refine, and refine) emulation technology to *attempt* to deal with polymorphic viruses, was spent to develop new *virus protection* techniques. (The reason I say *attempt* to deal with polymorphic viruses is because even though the emulation allows the product to strip the encryption from the virus, the product still needs to identify the virus specifically to satisfy its detection/removal needs.)

Fortunately, some companies, in addition to *Symantec's* effort with script behaviour blocking, have finally started to look outside the box. Even *Microsoft*! Their *Outlook Security Update*, while applying draconian limitations and restrictions, at least shows that they're doing *something* other than scanning to help deal with viruses. *Microsoft's* approach reminds me of that old saying about computer security only existing if the PC is turned off, but hey… if .EXE files cannot be emailed from a PC with the *Outlook Security Update* installed, then we can be sure that file-infecting viruses will not be sent to anyone from *that* PC.

OK, by now you are probably thinking that I am dead set against scanning technology. Well, that is just not the case. Scanning is an absolute requirement if you have an existing virus infection to deal with. It is also quite effective in gateway/firewall environments to help screen out trouble from one central location as opposed to having the problem appear multiple times on multiple PCs throughout the organization. Where I am dead set against the use of scanning technology is on the desktop as a *protection* method. It is just not good enough!

Some will immediately say that desktop protection is not a requirement with today's sophisticated network solutions, but in reality, the desktop is the most vulnerable component of the entire organization. This is where the user installs that unauthorized software; downloads a file from that nasty Web site; runs that script sent via email from their friend; brings in the infected documents or even – yes folks, it still happens – a diskette infected with a boot virus, from their home office; etc, etc, etc. And furthermore, with all the home computers in use these days, there is no protection available for them other than desktop scanners.

By now, most people are knowledgeable enough about viruses to know that there are different *types* of viruses that infect different *objects*, and therefore require different detection/prevention algorithms. Currently, we have products that attempt to deal with all types of viruses by utilizing a single basic premise. To whit: scan the data stream at each entry point for all the known viruses that may come through that entry point. This is like attempting to stop drug smuggling by having Customs officials compare the faces of all people entering the country to a database containing photos of all known drug smugglers, and allowing entry to all those whose photo is not in the

database. This would really eliminate the drug problem, right? (Just say no! ☺)

If we look at developing detection/protection methods tailored for each type of virus, we *should* be able to perfect that protection and, over time, eliminate entire categories of virus.

1. We need multiple, unique components deployed simultaneously to act as an aegis for vulnerable desktops.
2. There is nothing wrong with products that only perform a 'small' but effective function towards stopping a serious threat.
3. These product components do not need to be obtained from a single vendor. If the software is developed properly, you should be able to select your AV components like you would select dinner from a Chinese restaurant: 'one from column A and two from column B'.
4. No matter how small the component's threat-stopper technology may be, it could prove to be a blessing if it stops that next epidemic that is heading our way.

For years, technology has existed which provides for the instant detection and clean-up of a boot virus infection, no matter whether it was from a known or unknown virus. However, since the computing community first took the view that 'a virus is a virus, and I've already got an AV scanner', and then took the view that 'boot viruses are dead', they were reluctant to consider purchasing or even installing a *free* product that could have eliminated the entire category of boot viruses. So, we still have boot virus infections occurring. We should be encouraging the development of technology of this nature, not stifling it.

These are some other simple, singular approaches that could prove effective in stopping threats – they use minimal system resources and CPU cycles, yet still permit an open computing environment:

1. Examine (or filter) email attachments for files that have utilized the 'hidden extension' or 'double dot' trick, and strip away any file attachment that uses this trick. If the filename is nnnnn.txt.vbs or nnnnn.yyy.zzz, get rid of it. Surely there cannot be many legitimate files of this nature that need to be sent via email. This single step can keep out the majority of worms, both known and unknown, without restricting the usage of scripts.
2. Examine email attachments for documents and spreadsheets that contain macros, and strip the macros from the file. This single step can permit documents and spreadsheets to be reviewed in their intended style, and yet keep out the majority of macro viruses, both known and unknown.
3. Examine (or filter) email attachments for files with an extension of .EXE. If the file name is *not* INSTALL.EXE or SETUP.EXE, get rid of the file. By allowing only packages that appear to be 'installable' and thus legitimate, we can turn away many of the file infectors, both known and unknown, that are apt to be inadvertently run by an unsuspecting user.

Are these approaches simple in concept and in execution? Certainly. Will they stop all viruses? Of course not. Will they stop *a lot* of viruses? In my opinion, yes. Especially those that are seriously making the rounds. Do they impose severe restrictions on the user's freedom to do their job? Not at all. You have got to tell your AV vendors, though, to start thinking along lines similar to these, or we will never see any effective new technology. It just will not happen.

I can tell you right now that members of the anti-virus community will immediately tear apart the above suggestions by saying 'but it won't handle situation x' or 'it won't prevent virus y', etc, etc. And many are sure to say 'An infected file named INSTALL.EXE will get right past this protection'. And they will be right. But how many infected files will be named INSTALL? Not many.

What is important is that a large number of viruses, *including those not yet written*, will be stopped in their tracks without imposing drastic limitations on users.

If the above tactics cannot deliver 100%, that is OK. Remember that scanners *will not* detect new viruses. And many times, even updated scanners *will not* detect new viruses because they actually require an upgrade to detect a particular virus. We are playing the odds here, folks. There is no 100%. Remember that, and remember that well.

The AV community *must* play Devil's Advocate to any new approach to AV that is suggested. However, the suggestions listed above are not aimed towards developing the 'be-all, end-all, keep-every-virus-out' product. This animal does not and will not exist.

These measures *will* go a long way towards keeping out the viruses that are using today's common techniques and trickery. Occasional updating of protective measures to deal with newly emerging tricks developed by virus authors is a lot simpler, faster, and more effective than constantly updating for each new virus that gets written.

The I Love You worm was *not* a technological wonder, and, most likely, the next major epidemic will not be caused by a technological wonder either. Let us all put on our collective thinking caps and explore new ways to keep this garbage off our PCs. There are enough aggravating situations in life to deal with. We do not need viruses in our lives. Let us make them history!

If you have any thoughts or comments on the suggestions I have proposed in this article or wish to discuss any of your pet ideas, I would very much like to hear from you. Please contact me either through Virus Bulletin or email Ray.Glath@Pcsupport.com.

CONFERENCE REPORT

No Mickey Mouse Outfit

Francesca Thorneloe

Young at heart but increasingly well-developed, and some would venture precocious – the *VB* conference celebrated its tenth birthday in Orlando, Florida on 28 September 2000. Like most youngsters, this one's continually learning to master the basics – relationships, discipline, language, interaction. Thus, VB2000 navigated the choppy channels of communication, sometimes surprisingly articulate and perceptive, often blatant, occasionally subversive, appreciating when to put up and when to shut up.

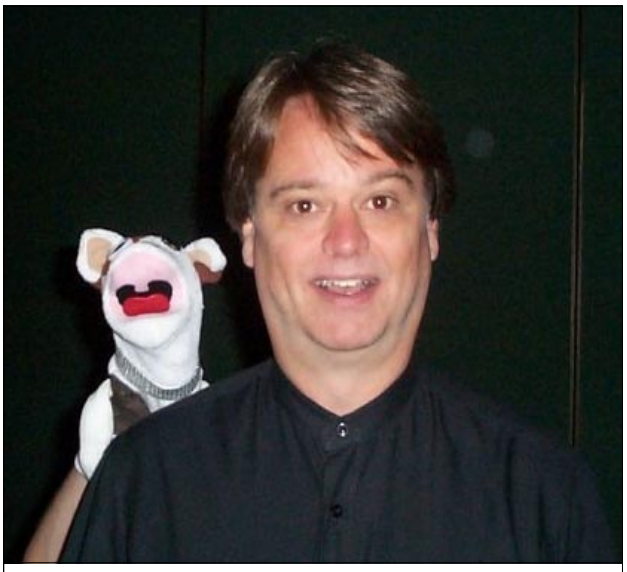
Does Size Matter?

Maybe it's my fault for going barefoot everywhere, but I felt more like a pygmy than ever among the giants of AV and corporate IT security. What do they feed them over there? Despite the fact that I had to climb onto a chair to hold several conversations, I never felt that my horizon was limited by my small stature. Neither is *Virus Bulletin's*. The journal may be relatively little but this was undoubtedly the largest conference yet. And it felt more like an extended family than ever with an unprecedented number of partners and children attending our coming of age. And this family's just like any other – all noise, character and attitude.

Those of you who know me will chuckle to recall how depilation became something of a feature of VB2000. And we're not just talking whiskers here. The Crew happily pulled each other's hair out in their sterling efforts to conjure additional conference merchandise out of the humid air when we realised we had woefully underestimated our popularity this year.

Our keynote speaker's hairloss was more deliberate. It takes a confident man to orchestrate 300 or so double-takes but *IBM's* Steve White managed to disorientate all the delegates, not to mention his wife, when he shaved off his trademark beard for his hugely popular futuristic retrospective. Pat Nolan from *NAI* pulled the same stunt – it took a while for everyone to recognize the Bear of Beaverton in his shirt, tie, short back and sides.

Like nearby Disney, VB2000 had something for everyone. And unification was our aim as we opened the conference on Thursday morning with the perils of misinterpretation versus the merits of communication [*I seem to recall that we did that bit a little* too *realistically! Ed.*]. Credit for the pleasing symmetry in the proceedings must go to our surprise guest, Steve White's little friend Bob, and his intrepid 'trainer' Ian Whalley. They kicked off both the keynote and Graham Cluley and Carole Theriault's entertaining closing paper, which reunited the corporate and technical streams.



The Dynamic Duo – Steve White and his assistant Bob

Ten years' worth of topics seem to have been covered in two days. And what a range! It was a time for looking back and reflecting, like Bruce Burrell and Allan Dyer. It was a time, too, for looking forward like Ian Whalley, Aleksander Czarnowski and Carey Nachenberg.

The breadth of technical knowledge could not have been more diverse. One participant wasn't sure what IRC stood for, while others were unfazed by a complex dissection of *Linux* ELF formats. Despite these differences, the delegates weren't afraid to face up to brave new threats like those to PDAs and WAP phones, as discussed by Eric Chien and Mikko Hyppönnen. In fact, some of the humour during the Q&A sessions revealed an increasingly well-informed and enlightened audience. John Bloodworth won't forget being told 'I don't trust you – nothing personal by the way' by one listener to his 'AV Industry – Smug or Smart?'

The usually controversial suspects were charming in their offensives. Vesselin Bontchev's tackling of the VBA upconversion problem and Nick FitzGerald's damning indictment of AV scanners were positively contemplative compared to recent years. Less lucky was Paul 'Duck' Ducklin, who introduced his boss and deputy speaker Jan Hruska from his hospital bed, following an unforgiving altercation between his bike and an articulated lorry! The subject of his paper – safe virus exchange, was to become one of the main talking points of the conference.

The recurring issues of trust, honesty and open relationships culminated in a public peace treaty between *CARO* and REVS. Later, a true confession revealed the gulf between AV marketroids and AV 'techies' – the latter faction not afraid to promote open and friendly co-operation and support between rival companies.

Food for Thought

Unlike the daunting Florida portions we were confronted with at meal-times, the sessions this year were organized into bite-sized chunks. Thus, between lunch and tea corporates were served back-to-back, real-life case studies by *Boeing's* Jeannette Jarvis and *Prudential's* Joe Donovan while techies digested two courses on the threats to *Linux* from Jakub Kaminski and Marius Van Oers. Other specially prepared sandwiches included VBA and Win32. One of my fondest memories of the conference is of Péter 'Mr Win32 to you' Ször's beautiful wife standing outside his presentation, eyes raised heavenwards and both fingers crossed! Talk about personal service from tech support!

CNN admires the VB2000 speaker's gift…oh and Sarah Gordon

We couldn't be accused of élitism either. The common PC user found spokesmen in Righard Zwienenberg and Joost de Raeymaeker. Familiar faces from anti-virus regulars put a new spin on everyday subjects like USENET, cryptography and AV engine infrastructure. 'VB virgins' Mark Kennedy, Szilard Stange, Vanja Svajcer and Mark Sunner tackled script-based mobile threats, virus collection management, VBA and ISP scanning. All the speakers were worth their weight in customised Times Atlases. Someone suggested a DVD version to give the airport carriers a break – where's the fun in that?

Thursday night's traditional Gala dinner offered the best in American cuisine, but this year I think we should have put small bottles of Pepto Bismol next to the Mickey Mouse ears at each place setting. The unique award ceremony to honour the individual who contributed the most to the anti-virus industry over the last decade played hell with my digestion – and I knew who the winner was! And while the rest of the audience did not have that foresight, by coffee time the magic, music and games of chance which later characterized the evening faded as a very clear picture came into focus.

In fourth place, the VB2000 delegates had voted for Duck – 'an accessible and passionate guru for the younger AV set'. Jimmy Kuo's popularity ensured that he needed no introduction when the nomination for third place came in; everyone knows his name. A tense joint second place twinned Jakub Kaminski – a man with a reputation for impeccable ethics – and Joe Wells – the founding father of the famous WildList. Way ahead of the crowd though was Vesselin Bontchev, one of the anti-virus industry's most respected and recognisable figures, who fully deserved his engraved plaque and a place in *Virus Bulletin's* history.

Man of the People – and this is before Vesselin Bontchev's opened the bottle!

The Speaker's Panel

Six of the best (plus one, for luck I guess!) summarised what had been gleaned, begged, borrowed and downloaded over the past couple of days. I'll remember this year's speakers' panel as one of the frankest I've heard and the closest to the walls coming down. A particularly brave but nervous presenter had one eye on his boss as he actually admitted 'Maybe I've said too much already'. The veils are certainly being lifted, users and subscribers alike are demanding honesty and straight talking. And the camera never lies – for the first time this year we went global as *CNN* filmed *VB* stalwart Sarah Gordon's participation in the conference – as reliably bang up-to-date, timely and informative as ever.

The communication pot bubbled away nicely as the *WLO* and *CARO* convened to chew what was left of the fat. Which left us to pick the 'men of the match'. This is never an easy task but the delegate assessment forms are indisputable this year. And I should know better, I have to change the epithet to a gender-neutral one! So congratulations to 'Flat Eric' Chien and Jeannette Jarvis for their popular and enthusiastic deliveries.

And that's about it for this year – apart from this. One of my favourite images of the conference has nothing to do with the official programme but it does illustrate what a serious business we're in, and how an international effort and perseverance against the odds can produce spectacular results. One extremely jaded but enlightened ex-reveller confirmed to me on Thursday morning the longheld suspicion that Fins are world-class drinkers [*I just know that Eugene and Dmitry are going to protest! Ed.*]. Apparently, the night before, grown men had paled at the sight of 30 club sandwiches delivered by room service as medicinal aid to a secret drinking location within the hotel. Happily, no casualties were sustained, even when, in mid-juggle, two wayward mayonnaise bottles sailed off the balcony and into the balmy night. Better start practising for next year!

COMPARATIVE REVIEW

Compare and CoNTrast

Matt Ham

Every month has its theme as far as Comparatives are concerned. In my carefree youth, I may have been able to construe that light-heartedly, but it now seems that a more 'grumpy old man' state of grouchiness has been entered. This might not be entirely due to age, however, as the products this month were in some cases worthy of insults not printable in a family journal.

Specific rants will come later but include the obligatory blue screens, a few buckets of application lethargy, a dash of unscannable files and a sprinkling of obtuse terminology. Those of you who have a spare moment or two might well wish to link the problem to the product before starting to read – and may well be surprised.

There were added to this a few upsets in the pursuit of VB 100% awards and a few near misses either through oversight or misadventure. Overall, despite being responsible for the destruction of several vendors' hopes this month, it was definitely an interesting review to write and, it is hoped, will make interesting reading too.

Test Procedures

The last *NT* Comparative was featured in September 1999's *VB*. Readers are advised to refer to the testing procedures and protocol detailed there. For this Comparative, test-sets were updated and the ItW File and Boot aligned to the September 2000 WildList.

As before, full details of the results are presented in the tables. The results featured under the product headings are all for on-demand scanning unless otherwise indicated.

Aladdin eSafe Desktop v2.2

ItW Overall	98.1%	Macro	95.1%
ItW Overall (o/a)	97.9%	Standard	93.9%
ItW File	98.0%	Polymorphic	80.9%

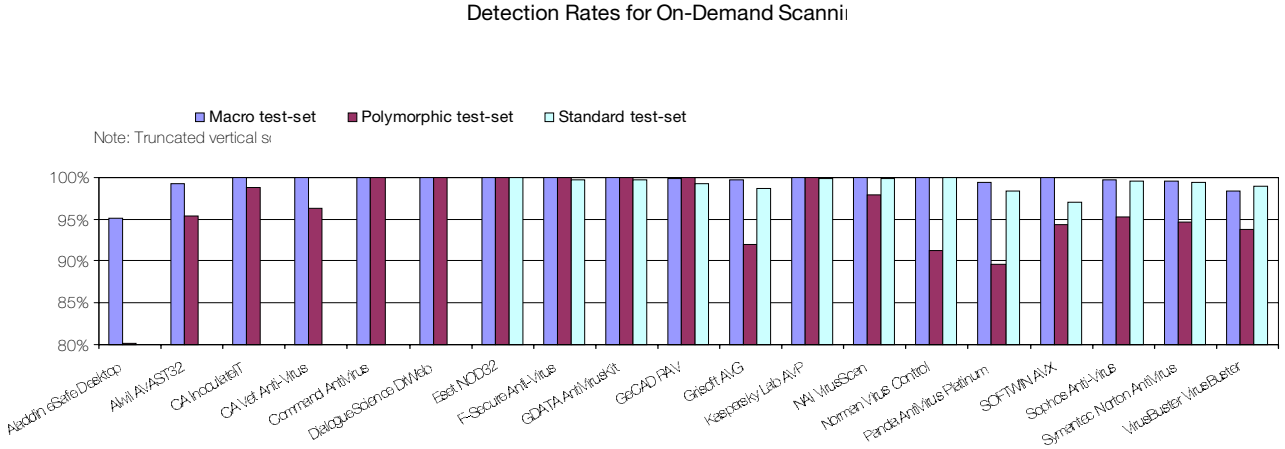
The *eSafe Desktop* is a whole range of programs forced into one application, with some odd interrelations as far as accessing the virus scanner part is concerned, and no note as to version number included within the applications. This complexity might be behind the mystery of the disappearing scan – whereby a scan was started, the operation was clearly occurring as far as disk accesses went, and yet no scan could be discovered through any of the methods available. This proved an isolated incident, however, and other scans progressed without further hitches.

The few problems incurred in producing the results were not particularly indicative of great detection. On-access there were considerable misses in the Polymorphic sets, and the Macro set threw up some weaknesses too. On many occasions in the latter set the product detected a virus in all but the template form.

Alwil AVAST32 v3.0.293.0

ItW Overall	100.0%	Macro	99.2%
ItW Overall (o/a)	n/t	Standard	98.9%
ItW File	100.0%	Polymorphic	95.4%

AVAST32 has a most remarkable on-access component, which seems to be triggered only by the method of not wanting it to trigger. Straightforward on-access testing for viruses proved, after exhaustive fiddling, to be an impossible task. However, since the *AVAST32* engine has heuristics and checks for such operations as copying files, the



On-demand tests	ItW Boot		ItW File		ItW Overall	Macro		Polymorphic		Standard	
	Number	%	Number	%	%	Number	%	Number	%	Number	%
Aladdin eSafe Desktop	0	100.00%	1	98.13%	98.18%	191	95.13%	1144	80.09%	117	93.92%
Alwil AVAST32	0	100.00%	0	100.00%	100.00%	31	99.21%	28	95.36%	13	98.93%
CA InoculateIT	0	100.00%	0	100.00%	100.00%	0	100.00%	9	98.87%	2	99.61%
CA Vet Anti-Virus	0	100.00%	0	100.00%	100.00%	0	100.00%	178	96.37%	0	100.00%
Command AntiVirus	0	100.00%	3	99.70%	99.71%	0	100.00%	1	99.98%	13	99.23%
DialogueScience DrWeb	0	100.00%	0	100.00%	100.00%	0	100.00%	0	100.00%	0	100.00%
Eset NOD32	0	100.00%	0	100.00%	100.00%	0	100.00%	0	100.00%	0	100.00%
F-Secure Anti-Virus	0	100.00%	0	100.00%	100.00%	0	100.00%	0	100.00%	21	99.71%
GDATA AntiVirusKit	0	100.00%	1	99.50%	99.51%	0	100.00%	0	100.00%	2	99.71%
GeCAD RAV	0	100.00%	1	99.75%	99.76%	8	99.79%	0	100.00%	8	99.25%
Grisoft AVG	0	100.00%	2	99.50%	99.51%	11	99.71%	124	92.01%	30	98.67%
Kaspersky Lab AVP	0	100.00%	1	99.50%	99.51%	0	100.00%	0	100.00%	1	99.81%
NAI VirusScan	0	100.00%	1	99.93%	99.93%	0	100.00%	17	97.87%	7	99.86%
Norman Virus Control	0	100.00%	0	100.00%	100.00%	0	100.00%	286	91.23%	0	100.00%
Panda AntiVirus Platinum	0	100.00%	0	100.00%	100.00%	26	99.35%	889	89.69%	50	98.34%
SOFTWIN AVX	0	100.00%	2	99.69%	99.70%	2	99.95%	55	94.36%	63	97.07%
Sophos Anti-Virus	0	100.00%	1	99.93%	99.93%	13	99.65%	191	95.24%	14	99.55%
Symantec Norton AntiVirus	0	100.00%	0	100.00%	100.00%	17	99.53%	264	94.74%	16	99.46%
VirusBuster VirusBuster	0	100.00%	29	96.16%	96.27%	66	98.34%	292	93.77%	10	99.01%

on-access scanner was all too easily triggered by the overhead testing regime which employs the notorious XCOPY command. Adding insult to the already considerable mental injuries imparted by these circumstances, the product failed, during floppy on-access tests, to detect Michelangelo.A and Stoned.June_4th.A.

Unfortunately, Clean set testing produced a single false positive, but *AVAST32's* scan times were very much in the 'respectable' range. All in all, *AVAST32's* performance ItW was impeccable, but the lack of a testable on-access scanner, and the false positive, denied it a VB 100% award.

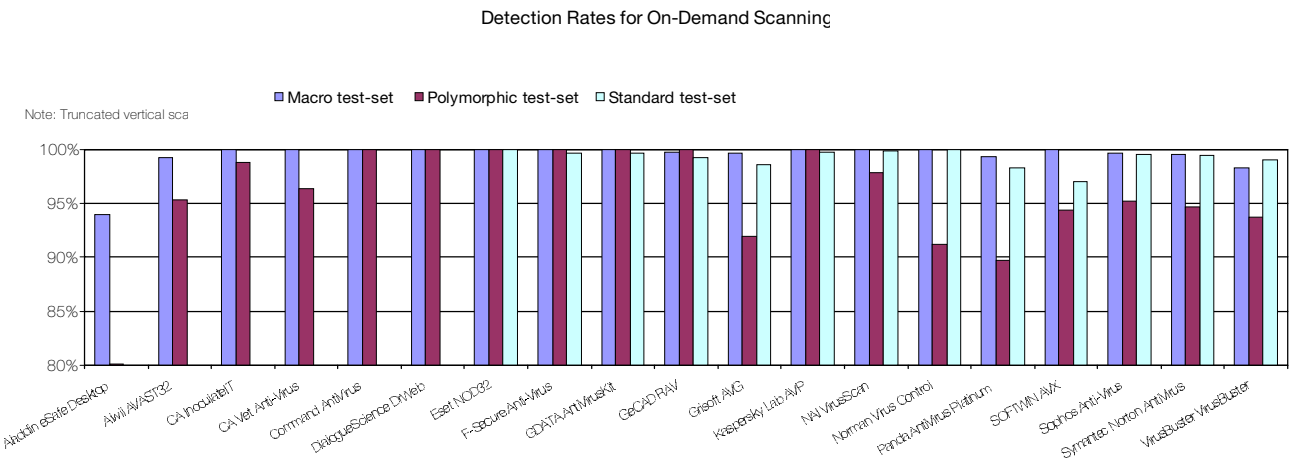
CA InoculateIT v4.53 16.24

ItW Overall	100.0%	Macro	100.0%
ItW Overall (o/a)	100.0%	Standard	99.6%
ItW File	100.0%	Polymorphic	98.9%

The main niggle with *InoculateIT* turned out to be at the installation stage. This process required several different patches, some self-extracting, others using *CA's* own custom decompression utility. Having worked through this and a subsequent install with numerous option selections, all was plain sailing.

Despite being the first product to claim a VB 100% award this month, it must be mentioned that the usually reliable *InoculateIT* did display signs of instability, eventually performing well after several false starts. Having said that, the results speak for themselves and the first of *Computer Associates'* products can rest assured that its reputation for a solid performance has been maintained.

There are currently rumours abounding about changes to *CA's* anti-virus product lines. It may be that by the next Comparative, *CA* no longer offers two distinct products. So, how did *Vet* compare this time round?



CA Vet Anti-Virus v10.2.2

ItW Overall	100.0%	Macro	100.0%
ItW Overall (o/a)	100.0%	Standard	100.0%
ItW File	100.0%	Polymorphic	96.4%

The traditionally stable *Vet* managed to get off to an impressively unusual start with a blue screen during browsing for a scan area. The product also performed oddly in that its default 'action' mode for files only reported viral infections – it did not deny access to them. Added to this was the continuing offer of a 'format' after the accessing of any infected floppy.

When combined with the developer warnings of 'bugginess' within the virus definitions, there were no great hopes held out. However, no further problems ensued and *Vet* turned in a solid performance. *CA's* second product is, once more, the proud possessor of a VB 100% award.

Command AntiVirus v4.59.4

ItW Overall	99.7%	Macro	100.0%
ItW Overall (o/a)	100.0%	Standard	99.2%
ItW File	99.7%	Polymorphic	99.9%

Command AntiVirus was something of a pleasant exception to the rule in this review, exhibiting no real problems, glitches or irritations in its operations.

The product was let down by its on-demand scanner, which detected slightly fewer viruses than its on-access counterpart. An average scan speed placed *Command* pretty much in the middle of the pack, and while no false positives were discovered, the only thing that really distinguished this product was ease of use and stability.

DialogueScience DrWeb v4.21

ItW Overall	100.0%	Macro	100.0%
ItW Overall (o/a)	97.1%	Standard	100.0%
ItW File	100.0%	Polymorphic	100.0%

The oddities evinced by *DrWeb* were thankfully of the non-destructive sort, especially in the case of reboots. Unlike another product's unannounced reboot feature, *DrWeb* states that a reboot will occur and is required, though this never comes to pass. This feature was particularly glaring due to the nature of the on-access component. Each alteration to this requires a reboot to be effective, irritating in a normal environment and enraging when testing a product under various configurations.

The singularity of the on-access scanner was not limited to these antics, however, since it operates a 'smart mode' for deciding which files should be scanned. No files were detected as being viral, however, since this 'smartness' was not pronounced enough to trigger a reaction.

Selecting 'open' as the trigger proved rather more effective, though it should be noted that the detection rates on-access are therefore not those produced under a default configuration. This alone would be sufficient to deny *DrWeb* a VB 100% award, though the point was moot given the lack of on-access boot sector scanning in this product.

Eset NOD32 v1.47

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

This month *NOD32* was denied a VB 100% award for the first time in living memory. This was not due to poor detection, however, as every file in the *VB* test-sets was detected as viral. The problem came in this case with false positives – the little-known HLLC.Fataler virus apparently showing up in some Clean set files.

A few new (to this reviewer at least) features cropped up as well, most of which appeared to be for the sole purpose of securing *NOD32* from those interfering busybodies also known as users. This took the form of password-protection for settings within the program. This product remains the fastest in terms of scanning speed for executables – its handling of OLE files is hardly sluggish either.

On-access tests	ItW Boot		ItW File		ItW Overall	Macro		Polymorphic		Standard	
	Number	%	Number	%	%	Number	%	Number	%	Number	%
Aladdin eSafe Desktop	0	100.00%	12	97.98%	98.04%	191	95.16%	1144	80.09%	122	93.58%
Alwil AVAST32	2	91.67%	n/t	n/t	n/t	n/t	n/t	n/t	n/t	n/t	n/t
CA InoculateIT	0	100.00%	0	100.00%	100.00%	0	100.00%	0	100.00%	2	99.61%
CA Vet Anti-Virus	0	100.00%	0	100.00%	100.00%	10	99.86%	768	91.10%	3	99.81%
Command AntiVirus	0	100.00%	0	100.00%	100.00%	0	100.00%	1	99.98%	9	99.22%
DialogueScience DrWeb	24	0.00%	3	99.88%	97.07%	19	99.79%	0	100.00%	0	100.00%
Eset NOD32	0	100.00%	0	100.00%	100.00%	0	100.00%	0	100.00%	0	100.00%
F-Secure Anti-Virus	0	100.00%	1	99.93%	99.93%	0	100.00%	0	100.00%	21	99.71%
GDATA AntiVirusKit	24	0.00%	649	22.26%	21.63%	1488	60.82%	623	83.30%	34	98.26%
GeCAD RAV	0	100.00%	1	99.75%	99.76%	8	99.79%	0	100.00%	8	99.25%
Grisoft AVG	24	0.00%	3	99.61%	96.81%	12	99.74%	292	89.47%	46	97.22%
Kaspersky Lab AVP	24	0.00%	1	99.50%	96.70%	0	100.00%	0	100.00%	1	99.81%
NAI VirusScan	0	100.00%	1	99.93%	99.93%	0	100.00%	99	95.71%	8	99.85%
Norman Virus Control	0	100.00%	7	99.50%	99.51%	26	99.46%	300	90.40%	2	99.77%
Panda AntiVirus Platinum	0	100.00%	0	100.00%	100.00%	26	99.35%	889	89.69%	52	98.21%
SOFTWIN AVX	24	0.00%	2	99.69%	96.89%	2	99.99%	56	94.36%	77	96.59%
Sophos Anti-Virus	0	100.00%	0	100.00%	100.00%	13	99.66%	191	95.24%	37	99.15%
Symantec Norton AntiVirus	0	100.00%	0	100.00%	100.00%	17	99.53%	264	94.74%	18	99.44%
VirusBuster VirusBuster	24	0.00%	29	96.16%	93.46%	66	98.34%	292	93.77%	292	93.77%

F-Secure Anti-Virus v5.2 Build 6382

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

FSAV's system of logging – entailing large amounts of data being held for analysis after scans – again seemed the cause of instability during testing. This manifested itself in an apparently innocent pause, which unfortunately turned out to be a hang sufficient to prevent reloading the scanner without a reboot. As with other products, the circumvention of stability problems involved detection by deletion.

On-access boot scanning, despite being 100% effective on the detection front, showed a peculiarity with alerting. Upon detection, two windows pop up. The topmost one is unusable and it is in the hidden window that choices, not easily apparent in this state, must be made. It would presumably make more sense in a network setting, though the software was installed in a dedicated standalone mode.

Despite being capable of detecting the .DLL part of W32/MTX on-demand, *FSAV* somehow missed it on-access ItW and thus avoided a VB 100% award. Other misses were more consistent over the on-access and on-demand scans, including the .BAT forms of 911.A and 911.B.

GDATA AntiVirusKit Generation 10

ItW Overall	99.5%	Macro	100.0%
ItW Overall (o/a)	21.6%	Standard	99.7%
ItW File	99.5%	Polymorphic	100.0%

The first sighting of this line in a *VB* Comparative would suggest a new product, though beneath its exterior beats a reliable heart – the *AVP* engine. Having spent many happy,

and a few not so happy, hours with *AVP* I noticed that the products definitely share a similarity in approach. One major difference lies in the matter of macro virus detection.

On-access, these files are, by default, simply not searched for. This might seem a glaring omission yet it is not quite as bizarre as it might seem. *AntiVirusKit* includes an *Office*-integrated virus scanner which would lead to effective redundancy were OLE files scanned on-access. Whether this is a good or bad idea overall is open to debate, but the on-access detection rates are very much altered by this fact. The objects and actions scanned are subject to some alterations in scope, though until the product has been through a full standalone review the options selected were deliberately limited to a simple 'on/off'.

The perils of a product not 100% home-built were apparent in its uncharacteristic (for *AVP*) instability. This was noted during on-demand floppy scanning, where alerts consisted of three different windows – the alert itself, an analysis and a report. With many samples to scan, speed is usually of the essence, though in this case there were altogether too many visits to Dr Watson.

As well as the misses produced by the option of not scanning for macros, *GDATA's* product also missed other files all of which (apart from VBS/Netlog.D) were detected successfully by *AVP*. The problem is mainly the choice of extension scanned, and some old favourites, namely W32/Marburg-infected screensavers and W95/Navrhar-infected VXDs, made an unwelcome return to the missed list. More disturbingly, there were some simply unaccountable misses, including several samples of the venerable Digital in the Polymorphic set.

GeCAD RAV Desktop v8.0.56.29

ItW Overall	99.8%	Macro	99.8%
ItW Overall (o/a)	99.8%	Standard	99.3%
ItW File	99.8%	Polymorphic	100.0%

RAV has undergone something of a facelift in its latest, pre-release incarnation – to the extent that it now sports skins in the same way as programs such as *WinAmp* do. Admittedly, one of those supplied would make all but the most ardent dog-lover cringe, though the other is agreeable in an 'oval' kind of way.

Such improvements will remain unseen by some users, however, as several of the configuration screens are of a fixed size and too large to use in lower resolutions. Even with the correct resolutions it was not possible to activate all features and in the absence of a functioning log file the scan was performed by deletion.

The scan itself was notably slow, though by no means the worst on offer, with Neuroquila proving particularly soporific for the *RAV* engine. Having said all this, detection rates showed a significant improvement over *RAV's* last outing in an *NT* Comparative.

Grisoft AVG v6.0.198

ItW Overall	99.5%	Macro	99.7%
ItW Overall (o/a)	96.8%	Standard	98.7%
ItW File	99.5%	Polymorphic	92.0%

The finest hour in *AVG's* attempt upon the reviewer's sanity came in, of all things, the update procedure. Having downloaded the correct version of the virus definition updates file and installed it, nothing happened. Consultation with the developer led to the interesting revelation that the English (UK) and English (US) versions are mutually incompatible. It also seems that there is no immediately obvious source for the former on the *AVG* Web sites. When an update was finally triggered the installation required the program to restart – which, in turn, triggered an unannounced reboot of the machine. With such a start it came as no great surprise that scans are quite fiddly to set up under the *AVG* Task Manager.

On-access misses included the now notorious JS/Unicle and the extensionless O97M/Tristate.C, together with the .OCX part of W32/Funlove. The remaining Tristate samples in the Macro test-set were also missed in the same extensionless form, though overall *AVG's* performance was respectable, with only the WM/Password and A97M/AccessiV samples missed otherwise. The Polymorphic set too showed only the 'usual suspect' misses of ACG.A and .B, plus the samples of Win95/SK8044 and Win95/SK7972.

Kaspersky Lab AVP v3.5.133.0

ItW Overall	99.5%	Macro	100.0%
ItW Overall (o/a)	96.7%	Standard	99.8%
ItW File	99.5%	Polymorphic	100.0%

AVP was denied a VB 100% award in the *NetWare* Comparative by dint of dubious default extensions and the missing of a single sample of VBS/Netlog.D. This glitch was a cause of some consternation since the chaps at *Kaspersky Lab* were adamant that they detected this virus. Exchanges of samples proved this to be a naming issue – their Netlog.D was most other folks' Netlog.B, though numerous other names popped up on competing scanners.

This might cause some readers to wonder how the *VB* test-set samples are chosen, if the AV developers cannot decide how viruses should be named. The answer is thankfully simple, our ItW samples are replicated from WildList samples which have been directly replicated from the wild. Thus, we can be sure that the *VB* Wildset reflects precisely those samples in the WildList.

The non-detection of VBS/Netlog.D in this month's Comparative was the only thing which stood between *AVP* and 100% detection of all file samples on-access. *AVP* was also, however, another of those scanners whose *NT* on-access boot scanning capability is notable by its absence, and thus missing the VB 100% award was not simply a naming problem after all.

NAI VirusScan v4.5.0.534

ItW Overall	99.9%	Macro	100.0%
ItW Overall (o/a)	99.9%	Standard	99.9%
ItW File	99.9%	Polymorphic	97.9%

The *NAI* scanning front end has mutated recently from an all bells and whistles affair to one which stresses purity and simplicity. If only this were matched in the field of virus detection. At first, problems seemed to be centred upon sluggish performance, but as the tests proceeded this became progressively worse. Left to its own devices the scan crashed repeatedly and was thus performed under a more watchful eye and by deletion. This soon proved to be far too painful, as upon scanning samples of W97M/Splash affairs became all but stationary.

W97M/Splash is a polymorphic macro virus, but it is polymorphic in the most basic way – by the insertion of random comments at each generation. Since these are never deleted the viral macros tend to become rather large and *VirusScan* accordingly had problems with the sizes. Earlier generations took minutes to scan, later ones were left to their own devices after the best part of a day had passed.

When the on-demand scan was eventually completed, I regarded the on-access scan with some trepidation but it proved eventful for other reasons. W97M/Splash samples were presumably subject to a time-out within the on-access scanner since there was no detection of these as viruses after a certain size.

The scan did, however, succeed in unloading the *McShield* component of the application after a certain point. Further investigations proved this to be the fault of the W32/Parvo virus, one sample of which could reproducibly unload the on-access scanner.

VirusScan was by no means alone in missing the .PIF versions of W32/MTX.B. The addition of Win95/SK8044 in the Polymorphic set and the .PIF portions of BAT/911.A and BAT/911.B rounded off its misses during both on-demand and on-access scans.

Norman Virus Control v4.86

ItW Overall	100.0%	Macro	100.0%
ItW Overall (o/a)	99.5%	Standard	100.0%
ItW File	100.0%	Polymorphic	91.2%

Usually a safe bet as far as stability is concerned, *NVC* was thankfully still on good form. There was a rather tedious delay incurred by the slowness of the zipped throughput test files but otherwise no problems were encountered.

NVC suffered the same fate as others with misses on the .PIF W32/MTX.B files, though a smattering of other misses on-access took the VB 100% award from *Norman's* grasp anyway. These misses were, unlike in most other cases this month, seemingly without rhyme or reason.

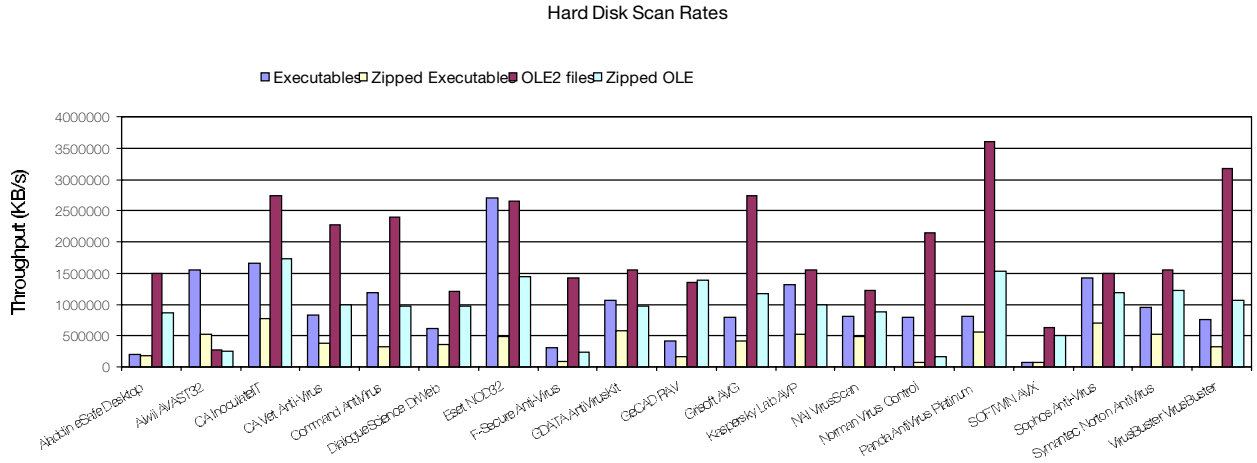
Panda Antivirus Platinum v6.20.00

ItW Overall	100.0%	Macro	99.4%
ItW Overall (o/a)	100.0%	Standard	98.3%
ItW File	100.0%	Polymorphic	89.7%

A good, solid performance by *Panda Antivirus Platinum* was nevertheless shanghai'd (as far as the VB 100% award goes) by the discovery of a single fasle positive. This product showed an admirable stability under most circumstances and was one of the more user-friendly on offer.

One oddity here seemed to be a lack of any way to restore the on-access scanner after it had been unloaded, short of restarting *Windows*. This did, however, give plenty of time to admire the ghostly panda's head which appears in the pre log-on screen of *NT* when *Panda Antivirus* is active. On-demand too there were strange forces at work, the speed tests culminating in an access violation which caused the scanner to cease operation.

While this product was far and away the speediest of the pack when scanning OLE files, traditional weaknesses remain within the Polymorphic set, where it missed an assortment of both old and new viruses.



SOFTWIN AntiVirus eXpert 2000 Desktop v5.8.0.12

ItW Overall	99.7%	Macro	99.9%
ItW Overall (o/a)	96.9%	Standard	97.1%
ItW File	99.7%	Polymorphic	94.4%

A product which recently passed through the *VB* standalone review process, this product gave no great surprises. It was mentioned in the last review that on-access scanning was not tested and this turned out to be due to the absence of protection within *NT* DOS boxes. Using a native *Windows* test application allowed on-access results to be obtained on this occasion, though real-time overhead tests were still not available since the standard *Virus Bulletin* test is itself run in a DOS box.

On-access, the ItW misses were few – one of the JS/Unicle samples and a .EXE version of Babylon – while in the Macro set just a couple of Win95/Navrhar-infected documents slipped past. More misses were apparent in the Polymorphic set, though *AVX* managed to detect ACG.A in the majority of samples proffered, whereas usually this virus is an 'all or nothing' affair.

Sophos Anti-Virus v3.38

ItW Overall	99.9%	Macro	99.7%
ItW Overall (o/a)	100.0%	Standard	99.6%
ItW File	99.9%	Polymorphic	95.2%

The problems encountered by *SAV* on this outing were relatively minor, being relegated to a poor selection of files to scan. This was particularly galling given that the resulting failed detections only occurred on-demand. The offending files were the .PIF versions of W32\MTX.B which, although not scanned by default, triggered the file type detection algorithms within *SAV's* on-access scanner.

Other than this, the misses and hits achieved by *SAV* followed an almost predictable pattern – stability was traditionally excellent and the overall performance solid.

Symantec Norton AntiVirus 2000 v6.00.03

ItW Overall	100.0%	Macro	99.5%
ItW Overall (o/a)	100.0%	Standard	99.5%
ItW File	100.0%	Polymorphic	94.7%

Norton AntiVirus cut straight to the chase this month, blue screening almost as soon as it was installed. This proved a precursor to yet more blue screens on the on-access testing which was finally performed by deletion. The deletion method did show forethought in the choice of files to be deleted – Byway and DirII.A were not deleted despite being detected as viral. These two viruses act by inserting themselves in the directory structure and an infection fixed by simple deletion is surely a cure worse than the disease as it leaves data in a non-accessible form.

NAV's slight instability on-access was presumably accentuated by the continuous stream of alerts generated, even when these were turned off at every mention in configuration. The on-access process also seemed to hang at several points, only to be reactivated by keyboard activity, which remains a most mystifying 'feature'.

Having said all this, *NAV* turned in a characteristically good performance and certainly deserves its VB 100% award this month. It is also a distinctly user-friendly product. In terms of scan speed, *NAV's* time test results place it within the respectably 'average' category.

VirusBuster VirusBuster v3.002

ItW Overall	96.3%	Macro	98.3%
ItW Overall (o/a)	93.5%	Standard	99.0%
ItW File	96.2%	Polymorphic	93.8%

Having tested the *NT* version of *VirusBuster* recently there were few problems anticipated when its turn came. Logging seemed to have become substantially harder to perform than in that review, and once more deletion was used as method of choice when testing scans.

Hard Disk Scan Rate	Executables			OLE Files			Zipped Executables		Zipped OLE Files	
	Time (s)	Throughput (kB/s)	FPs [susp]	Time(s)	Throughput (kB/s)	FPs [susp]	Time (s)	Throughput (kB/s)	Time(s)	Throughput (kB/s)
Aladdin eSafe Desktop	2752	198739		53	1496863		927	171970	87	857557
Alwil AVAST32	352	1553784	1	300	264445		307	519272	298	250360
CA InoculateIT	329	1662407		29	2735647		205	777641	43	1735058
CA Vet Anti-Virus	658	831203		35	2266679		418	381379	75	994766
Command AntiVirus	457	1196788		33	2404053		499	319472	77	968928
DialogueScience DrWeb	889	615221	[25]	66	1202026	[1]	439	363135	77	968928
Eset NOD32	203	2694247	3	30	2644458		328	486026	52	1434759
F-Secure Anti-Virus	1802	303513.		56	1416674		1684	94665	330	226083
GDATA AntiVirusKit	515	1062004		51	1555564		280	569344	77	968928
GeCAD RAV	1337	409074		59	1344640		1003	158939	54	1381620
Grisoft AVG	683	800779	7	29	2735647		382	417320	64	1165742
Kaspersky Lab AVP	413	1324290		51	1555564		307	519272	75	994766
NAI VirusScan	677	807876		65	1220519		330	483080	84	888184
Norman Virus Control	689	793805		37	2144155		2483	64203	454	164333
Panda AntiVirus Platinum	672	813887	1	22	3606080		290	549712	49	1522601
SOFTWIN AVX	7756	70517		125	634670		2329	68448	146	511010
Sophos Anti-Virus	385	1420603		53	1496863		225	708518	63	1184245
Symantec Norton AntiVirus	569	961216		51	1555564		304	524396	61	1223073
VirusBuster VirusBuster	724	755431	18 [4]	25	3173350	[1]	500	318833	70	1065821

The product remains slightly behind the pack in terms of detection – changes are happening but they are fairly slow to be felt at present. Average scanning speeds are made up for by a reliable stability.

Conclusion

The products seem in many cases to have achieved the complexity of Windows NT with the stabilty of early versions of Windows 3.0. There is a place for products to achieve both stability and functionality, and those products which managed this took very little coaxing to produce good results. The products without stability are mostly associated with a constant push for more and better features, though is this really needed?

For some products the answer must be yes. The two great forces for constant change are *Symantec* and *NAI*, as a result of their pushing towards domestic sales – the domestic user is often swayed to an inordinate extent by a feature list. This acts as a further push to all other developers, and the features are included; whether they are the results of ego or marketing needs is irrelevant.

If this sounds all too familiar then it might well be because *NT* itself is subject to the same forces, responsible for such wonders as 'VBS and VBA for all'. Those nasty users and their demands – they're to blame for everything!

Technical Details

Test Environment: Workstations: Three 166 MHz Pentium-MMX workstations with 64 MB RAM, 4 GB hard disks, CD-ROM and 3.5-inch floppy, all running *Windows NT* with *Service Pack 5* applied. The workstations could be rebuilt from image back-ups. All timed tests were performed on a single machine that was not connected to the network for the duration of the timed tests, but was otherwise configured identically to that described above.

Virus Test-sets: Complete listings of the test-sets used are at http://www.virusbtn.com/Comparatives/NT/2000/11test_sets.html.

A complete description of the results calculation protocol is at http://www.virusbtn.com/Comparatives/Win95/199801/protocol.html.

VIRUS BULLETIN **©2000 Virus Bulletin Ltd, The Pentagon, Abingdon, Oxfordshire, OX14 3YP, England. Tel +44 1235 555139. /2000/$0.00+2.50**

SUBSCRIPTION RATES

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

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

Editorial enquiries, subscription enquiries, orders and payments:

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

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

US subscriptions only:

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

Tel (781) 2139066, Fax (781) 2139067

END NOTES AND NEWS

AVAR2000, the 3rd annual conference of the Association of Anti-Virus Asia Researchers will take place from 28–29 November 2000 at the Shinagawa Prince Hotel in Tokyo, Japan. For more details email haru@jcsa.or.jp or visit http://www.aavar.org/.

***MessageLabs* has chosen *F-Secure* anti-virus software as a key component in its Virus Control Centres (VCCs).** Email sent to the VCC is scanned for malware, a service available to all *UUNET*'s customers and through ISPs *Star Internet*, *KPN* and *INS* in the UK, Benelux and Germany. For more information contact Jos White at *MessageLabs*; Tel +44 1285 884444.

The Internet Business Exhibition & Executive Conference takes place at the Brighton Metropole, UK from 5–6 December 2000. For more details contact Richard Cole; Tel +44 1273 773224 or visit the Web site http://www.ibshow.com/.

***Elron Software Inc* announces the release of *Internet Manager Anti-Virus*.** *IM Anti-Virus*, powered by *Sophos Anti-Virus*, is integrated with *Message Inspector*, an additional filtering service for email, newsgroups and FTP messages. Visit the Web site for more information; http://www.elronsoftware.com/.

The 16th Annual Computer Security Applications Conference (ACSAC) will take place from 11–15 December 2000 at the Sheraton Hotel in New Orleans, Louisiana, USA. The keynote speaker is Dr Eugene Spafford and the extensive tutorial program includes a session entitled 'Investigating Computer Viruses'. For a brochure or more details email publicity_chair@acsac.org or visit the conference Web site http://www.acsac.org/.

The UK Security Show 2001, incorporating The IT Security Showcase, is to take place in Hall 2 of the Wembley Arena in London, UK from 14–15 February 2001. The line-up includes interactive product demonstrations and practical installer workshops alongside study-based seminars and debates and more traditional conference-style presentations. For more details about the event visit the Web site http://www.securityshow.com/.

Chen Ing-Hau, the author of the CIH virus, faces up to three years in a Taiwanese jail having escaped justice thus far. Watch this space for more details.

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

***Norman Data Defense Systems* has released *Norman Virus Control* (*NVC*) for *Content Technologies' MAILsweeper*.** The company also announces the release of its corporate defence tool, the *Norman Security Server* (*NSS*). For more details on features, pricing and availability contact Dawn Cooke in the UK; Tel +44 1908 520900 or visit http://www.norman.com/.

Following an unprecedented reception this year, **LinuxWorld Conference and Expo 2001** is scheduled to take place at the Frankfurt Trade Fair Grounds in Germany from 8–10 November. For more details, see http://www.linuxworldexpo.de/.

Kaspersky Lab* has released a new-look, redesigned version of *AVP which includes new features such as the boot system Rescue Kit. Demonstration versions of *AVP v3.5* can be downloaded from the Web site; http://kasperskylabs.com or email denis@avp.ru for more details.

Dr Solomon's*, an *NAI* business, has released *VirusScan Thin Client which can be downloaded over the Internet in a package as small as 2 MB. For more information contact Caroline Kuipers in the UK; Tel +44 1753 217500 or visit http://www.nai.com/.

The release of ***Sophos Anti-Virus for Exchange* (*SAVEX*)** has been halted after the discovery of 'potential security flaws' during beta testing. *Sophos* recommends solutions from *Baltimore* (formerly *Content Technologies*), *Sybari* and *Group Technologies* (which integrates with the *Sophos* engine. See http://www.sophos.com/.

And finally, all of us at *Virus Bulletin* would like to say a big 'cheers' to our ex-Editors and members of the Advisory Board who participated in the unique **10th anniversary issue** of the magazine last month – it was one of our most popular editions to date. A local charity, *The Macmillan Cancer Research Fund* was the very surprised and grateful recipient of wage cheques from more than one of you featured – a classy gesture indeed.