ISSN 0956-9979

MAY 2001

VIRUS BULLETIN

THE INTERNATIONAL PUBLICATION ON COMPUTER VIRUS PREVENTION, RECOGNITION AND REMOVAL

IN THIS ISSUE:

• **Analyse this:** as promised, we cover two prominent, recent discoveries – W32/Magistr and the cross-infector Lindose. This month's Virus Analyses start on p.6.

• **The candidates debate:** two of *Virus Bulletin's* ex-Editors occupy opposite corners over the issue of virus disinfection. Richard Ford locks horns with Nick FitzGerald on p.14.

• **Internet special:** in his Comment on p.2, Andy Nikishin advocates integrity checking for the WWW, while Richard Wang's Feature on p.10 suggests that the Internet is the virus writer's best ally.

CONTENTS

VIRUS BULLETIN **©2001 Virus Bulletin Ltd, The Pentagon, Abingdon, Oxfordshire, OX14 3YP, England.**
www.virusbtn.com /2001/$0.00+2.50 No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form without the prior written permission of the publishers.

COMMENT

“ Integrity checkers are not fussy about what they check ”

Hacked Off with the Internet?

We are living in the era of the Internet. It has penetrated so deep into our lives that many of us cannot even imagine a world without email and the World Wide Web (for example, I spend more than an hour every day just reading emails and catching up with news). Moreover, a special term has recently been coined – Web Lifestyle.

Millions of people all over the world subscribe to this way of life. It’s possible to divide clients of the greatest Net in the world into three unequal parts: firstly, ordinary users or consumers (the biggest category); secondly, system administrators of Web sites who support and take care of sites; and lastly, hackers (here, when I say ‘hacker’ I mean Web site hacker, i.e. intruder). The first group of people just surf the Net and are largely harmless by nature. The third group tries to do harm to the second on a regular basis.

Why they do such things? Sometimes it is due to natural malice, sometimes to show off their skills, sometimes because of their political views (for example, the recent sensational hack of the site http://www.hizbollah.org). This small group of Internet users brings daily headaches to Web site administrators because the latter are forced continually to think how to protect their sites from the intrusion of malefactors. The saddest thing is that most of these attacks are based on bugs in the server’s software. This means that a new bug has been discovered and this bug has not been fixed yet and thus the probability of an attack is increased.

Almost every week we get notification that malefactors have hacked some site or other and changed its content. Sometimes it takes a relatively long period of time to discover such a hack. What can we do to minimize losses and reaction time? The a so-called traditional method to solve this problem involves bug reports, monitoring of server software, installation of all the latest patches, and the use of software or hardware firewalls and Intrusion Detection Systems. This method minimizes the risk of hacking and helps avoid the hack itself, but it cannot guarantee recovery of hacked or changed data. Neither can it help to find exactly what was changed by the hacker, nor minimize the reaction time.

Let us imagine that an intruder who hacks a Web store does not change anything except the price database. This is an invisible hack – without analysing the data you cannot find exactly what was changed. Customers are going to be happy – the site says that a *Palm m105* costs US$100 less than the recommended retail price! But I do not think that the store owners will be happy when they discover thousands of dollars worth of losses at the end of the month. What is the potential result of this kind of hack? At the very least, big financial losses and compromised reputation (nobody knows if the hacker stole customers’ confidential data). The worst case scenario would be the bankrupting of the Web store.

Take this as a further example: imagine that there is an ostensibly popular economic news server which publishes reliably good financial forecasts for company ‘A’ (obviously, this is a fake forecast and author of this ‘news’ is a malefactor). As a result of such a forecast stock prices could well rise. This financial fraud would be uncovered eventually, but by that time the malefactor has already earned a few extra dollars.

You can ask me ‘What do viruses and hacker attacks have in common?’ I would have to answer, ‘Nothing, certainly.’ However, I do believe that it is possible to use anti-virus technology to discover invisible hacks like this. I am referring to technology from integrity checkers which have been used for over 10 years to protect workstations against computer viruses. Indeed, integrity checkers are not fussy about what they check – it can be a disk on a PC or a WWW root folder. Unfortunately, the idea of integrity checking is not popular nowadays but I would like to advocate that monitoring the critical content of a Web server is an ideal place for this really good technology.

Andy Nikishin, Kaspersky Lab, Russia

NEWS

CLSID Trouble

Famed Bulgarian bug hunter, Georgi Guninski, has blown the lid on an old, albeit little-known, issue with *Windows* Class IDs (CLSIDs). Some CLSIDs can be used instead of extensions to filenames such that the standard *Explorer* interface entirely hides the extension and displays the default icon matching the file-type the CLSID represents. Double-clicking such a file results in the 'correct' handler, as determined by the (invisible) CLSID, being called to do what it sees fit with the file.

Of course, the so-called 'double extension' trick can be used with this to get a file that displays what looks like a full filename and extension but that has quite different behaviour from that expected. The demonstration Guninski posted to his Web site involves a file that is apparently a .TXT file but with the CLSID of the HTA file type as its true extension. Running it runs the file as an HTA program rather than opening it in *Notepad*.

The only malware known to employ this trick is the mass-mailing virus VBS/Postcard. It was discovered in early March but has not spread. *VB* recommends that if you are not already scanning all files you should add '.{??', '.{*' or '.{*}', etc as appropriate to the product to the 'executable file types' list of your scanners. There is no known 'legitimate' reason for CLSID-extensioned files to be distributed via email, so block such files (inbound and out) at your email server content filter ▮

On a Whim and a Prayer?

On 12 April the French publication *L'Express* carried a story on a connection between *Panda Software* and the Church of Scientology. What makes the story even more bizarre is that the French Government is involved and the implications are causing quite a scandal.

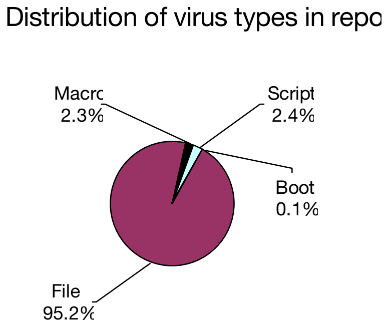
The French Ministry of the Interior signed a standard contract to receive *Panda's* anti-virus service in its central office and across local police stations. So far, so good. Then some bright spark noticed that *Panda's* founder, Mikel Urizarberrena was listed in *Impact*, the official Church of Scientology magazine, as a familiar benefactor and giver of several 'gifts', including a substantial financial donation. The plot thickens. *Panda* allegedly gives 'some of its benefits' to the *World Institute of Scientology Enterprises* (*WISE*), a US-based organization of 2,500 companies.

The righteously outraged journalist finally gets to the point – by association, and a fairly flimsy association at that, the French government is effectively paying the Church of Scientology to protect highly secret databases across France. Conspiracy theorists will have a field day with this one! ▮

Prevalence Table – March 2001

Virus	Type	Incidents	Reports
Win32/Naked	File	19010	84.7%
Win32/Hybris	File	917	4.1%
Win32/Navidad	File	731	3.3%
Win32/MTX	File	486	2.2%
VBSWG	Script	234	1.0%
Kak	Script	198	0.9%
Laroux	Macro	93	0.4%
LoveLetter	Script	86	0.4%
Win32/Magistr	File	78	0.3%
Onex	Macro	72	0.3%
Divi	Macro	66	0.3%
Ethan	Macro	62	0.3%
Marker	Macro	46	0.2%
Win32/Funlove	File	33	0.1%
Win32/QAZ	File	28	0.1%
Thus	Macro	24	0.1%
Win32/Ska	File	24	0.1%
Win32/BleBla	File	22	0.1%
Class	Macro	21	0.1%
Tristate	Macro	19	0.1%
Win95/CIH	File	16	0.1%
Story	Macro	12	0.1%
Bymer	Macro	10	0.0%
Cap	Macro	10	0.0%
Melissa	Macro	10	0.0%
Stages	Script	10	0.0%
Others [1]		132	0.6%
Total		**22450**	**100%**

[1] The Prevalence Table includes a total of 132 reports across 47 further viruses. Readers are reminded that a complete listing is posted at http://www.virusbtn.com/Prevalence/.



LETTERS

Dear Virus Bulletin

Do Not Tolerate Payment for Viruses

I already tried to initiate an ethical discussion with the company *GateKeeper* about its financial challenge to write a virus – what have you or your company done? People in the anti-virus industry get asked two questions on a daily basis:

1. Do you write viruses?
2. Do you pay anyone to write viruses?

My standard response so far has been a big 'NO!' to both questions, but when *GateKeeper* announced a challenge to write a virus with a reward of $10,000, this changed forever. If the AV industry can prove that they stopped the challenge, this incident could be changed into good press. Please make contact with *GateKeeper* or get your lawyer to do it.

This challenge can result in three scenarios:

1. A virus succeeds in hitting *GateKeeper* but not the rest of the world
2. A virus succeeds in hitting *GateKeeper* and the rest of the world
3. A virus does not succeed in hitting either *GateKeeper* or the rest of the world

The big concern is not if someone can make a virus which succeeds or not, but if the world gets hit and how badly in the attempts to hit *GateKeeper*. What will *GateKeeper* say if anyone succeeds in hitting it and by mistake the rest of the world? Will they pay? The thing is, for *GateKeeper* to prove that code be controlled there is no need for a virus at all – a Trojan is sufficient.

There could be some ethical justification for non-replicated code being proven to run on *GateKeeper's* system. This will do no harm outside the company, but with a virus there is no such validation. I hope someone in the AV industry will succeed in stopping this wilful challenge.

Klas Schöldström
Brainpool Consulting AB
Sweden

Misguided Marketing

How disappointing to read that *GateKeeper* is trying to promote how effective its product is by offering $10,000 to the first person to produce a virus capable of bypassing it. The anti-virus industry has been accused of encouraging people to write viruses in the past and there have even been conspiracy theories suggesting that we might produce the viruses ouselves. Anti-virus vendors have to be very careful about not hyping up the virus threat and activities like this certainly don't help.

Endeavours such as the *GateKeeper* 'competition' only perpetuate the misconception that anti-virus companies contribute to the virus problem, rather than working together to defeat it. No doubt *GateKeeper* thought it was a marvellous marketing idea at the time – but the plan does seem to have backfired somewhat, and hopefully they will have realised the error of their ways.

Maybe independent bodies like *Virus Bulletin*, the *ICSA* and the *WildList Organization* can do their part by co-operating with companies introducing new products to the marketplace, and detailing how they can have these products tested in a safe, competent way which does not appear to encourage the virus writers?

Natasha Staley
Sophos Anti-Virus
UK

Are We All Just A Little Guiltier?

Two stories have taken a lot of bandwidth on the discussion boards of the AV heroes during the last two weeks. One was about a company, or should I say a competitor, that dared to encourage people to write a virus and try to infect their product, promising to reward the successful attempt. The second was about AV companies gaining market shares in China for the price of virus source codes being handed over to Chinese officials. Though both differ in objective and impact, they do have one thing in common; they both are just one more example that ethics and morality are negatively reciprocal to the rise of prosperity and social wealth.

History books are full of examples of ethical and moral principles being sacrificed on the shrine of mammon. So what has happened here? Why the outcry – if there even was one? First of all, we have to admit that nobody committed an offence or a crime and that this might be just because of the lack of laws and regulation. However, nobody can and will be charged. But does that mean business as usual? I think we all – and I mean all of us who claim to belong to those who would not be excluded as a 'bona fide researcher' – we all are a little guiltier because we let it happen.

In the first case, a company just did what others have done before – 'hack my firewall and you get an award'. So far, so good. From the marketing point of view this is probably a laudable strategy and we now understand why other competitors did not like the idea. But there is a slight difference here and maybe the marketing strategy will result in an overwhelming result that nobody really wanted. This strategy encourages people to write viruses and those who are not capable of doing that may just go and find a construction kit on the Net and get started. One way or another, this will result in more viruses creeping up – was that the strategy?

The second case is different and leaves a lot of questions open. The first one coming to my mind is the question about the rationale and reasoning behind such deals. Could it be that China was a 'Virus-free' zone up to now and selling AV products would sound like selling refrigerators in Alaska? I have my doubts. Secondly, why does China have to ask for samples if they are, so they say, all available on the Net anyway? To extend this thought even more, if they are all available on the Net, why do AV companies have to deliver them – and even worse, publicly admit this?

There are a lot more questions and there may be a lot more answers or there may even be no answers at all. In the end it does not matter what drove the decisions. The fact is that we all have lost again – irretrievably, irreparably. The damage is not quantifiable and intangible but has left a bitter taste of losing, nothing more. My respect to those who have refrained from such deals and to those who have not commented but silently closed their fist in their pocket.

Rainer Fahs
EICAR Chairman
Brussels

NAI and the Chinese Government

US and European, ANZ and Japanese companies have several good certification sources to rely on. The Chinese government did not feel they could rely on these groups because they are distrustful of certification organizations they do not have under their control. The samples were provided to help them set up such a certification program which was completed and is now in operation in China. The sample set was very basic, smaller than those collected by several IT organizations outside the AV industry. It's important to remember the samples were given to a GOVERNMENT, explicitly for certification purposes.

Vincent Gullotto
Director AVERT-NAI Labs
USA

AV Algorithms Revisited

I am Alexander Otenko. Currently I am a full-time MSc overseas student at Salford University, here in the UK. The project I am involved in is concerned with Privilege Management Infrastructures, and is called PERMIS.

For some time I have been interested in computer infections and how they work. I cannot tell you how long I've been involved in it, because it happened spontaneously. My desire to explore was supported by the anti-viral contest held by the *Komi* anti-virus centre (www.virus.komi.ru) several years ago. Since then, I have been eagerly participating in such contests, and became a franchise author for an informal electronic AV edition called Zemskij Fershal. I believe its focus is virus detection, though some people could express other opinions.

My latest experience includes polymorphic virus detection. Certainly, I am not capable of producing efficient code for detecting any virus in that way, because I need to work on this problem more, I think.

My idea could sound not very new, since I might not be aware of theoretical research into this field, but I could explain the polymorphic virus detection using positions of syntax analysis of the underlying code. My practical proofs of its applicability are the 'Belka' polymorphic virus detection and 'PLY' full morph virus detection algorithms.

In particular, the detection routines express syntax of the output the virus engines could generate. Therefore, I find it profitable to work in this direction to express general rules of constructing a virus body, in order to be able to detect it by a syntax analyser of this kind.

The other article I am going to share with you concerns the well-known theorem – Cohen's theorem about a perfect virus-detecting algorithm. Basically, Cohen's problem is not that it is impossible to classify a certain paradoxical program P: if A(p) then exit, or spread.

It is for algorithm A(p) to give a correct answer for both the program and anti-virus; its ambiguity can be compared to the well-known logical paradox: 'This statement is false' in some respects.

Cohen, in particular, finds that it is impossible to build such a function A(p) which would project all programs into the set of true/false values. I must admit this is true. But I find that from the point of view of anti-virus software, the value 'false' is redundant. It wants to know only positive truths.

My approach, by extending Cohen's theorem to the algorithm A(p), merely returns 'true' for viruses and leaving the result undefined otherwise, expresses several solutions to such a problem. Though, at the first glance it might seem impossible to give anything but 'false', if it cannot give 'true', but I think there are ways.

Certainly, the latter point is of rather theoretical grounds, and does not suggest anything particular about virus detection (how precisely it will distinguish paradoxical programs, or how it will separate good and malicious programs), but expresses the need to extend the problem, or perhaps to define it in some other way. Notoriously, one of the solutions of the last problem might give some food for thought for classification tasks, where it is impossible to give a distinct answer.

Alexander Otenko
Salford University
UK

VIRUS ANALYSIS 1

Magisterium Abraxas

Peter Ferrie
SARC, Australia

W32/Magistr.24876@mm is a polymorphically encrypted, entry point-obscuring, anti-heuristic, anti-debugging, memory resident, parasitic infector of Portable Executable .EXE and .SCR files. It can replicate across local area networks, and it has mass-mailing capabilities (using its own SMTP engine), some highly destructive payloads, an interesting visual effect and a number of bugs.

Initialisation

As an anti-heuristic device, files infected with W32/Magistr do not have their entry point altered. Instead, the virus will save the first 512 bytes of code, and replace them with polymorphic garbage which includes subroutines, jumps, and some Structured Exception Handling tricks to interfere with debuggers and code emulators. Eventually, an indirect call, the address of which is stored by the virus in the Import Table of the host application, will transfer control to the section that contains the virus body.

The virus body is decrypted by XORing it with a single shifting 32-bit key, however the decryptor is also polymorphic, of variable size, and contains another Structured Exception Handler trick. Fortunately for the AV people, there is a characteristic of the decryptor which allows the encrypted body to be located quickly and accurately.

Once the virus is decrypted, it will attempt to find the KERNEL32.DLL base address by taking the return address from the stack and searching the previous 1 MB of memory for the MZ header whose export table DLL name ends with the string ‘EL32’. If the address cannot be found using that algorithm, then the virus will use one of two default values, based on the value of the high eight bits of the CS selector.

Using the KERNEL32.DLL base address, the virus will retrieve the addresses of 42 APIs it requires for system integration and replication on the local machine, the names of which are stored as checksums instead of strings. The checksum routine is a CRC algorithm using 16-bit registers that has been blindly copied into a number of recent 32-bit *Windows* viruses. It seems likely that not one virus author understands the algorithm well enough to produce a 32-bit version. A bug exists in the import parsing code which will cause a crash if an import cannot be found.

At this point is included a large chunk of code copied from the W32/Dengue virus. This process was introduced in *Windows NT*, and is always running. The residency code begins by converting the computer name to an encrypted string and creating a memory-mapped file using this name. The memory-mapped file is part of the mechanism that the virus uses to remain memory resident. Then, one of two routines is executed, based on the *Windows* platform (*9x*/*ME* or *NT*/*2000*), to search for the EXPLORER.EXE process in memory. Perhaps the most embarrassing bug in the virus exists here, in such a simple function as string comparison: it will return a match even if the last character differs in the strings. Once *Explorer* has been found, a 110 bytes routine is injected into a writeable section, and the TranslateMessage() API from USER32.DLL is hooked to point to this routine. After this, the original host bytes are restored and the host is executed.

You’ve Got Mail

The injected routine gains control whenever *Explorer* calls TranslateMessage(). This function is part of the message loop in all GUI applications, so it is called frequently. When the routine is reached for the first time, a thread is created and the function is unhooked. The thread will wait for three minutes before performing any actions.

After the time has elapsed, the thread will retrieve the location of the *Windows* directory, the Program Files directory from the Registry, and the Program Files drive. Depending on the first character of the computer name, the virus will choose one of those locations in which to create its data file. This data file will contain the date of initial infection, and the full path and number of ‘interesting’ files, namely those files which contain email addresses: the *Windows* Address Books (*.WAB), *Outlook Message* stores (*.DBX, *.MBX), and the *Netscape Messenger* mail files.

The thread will also retrieve the user name and email address of the current user. These names are taken from the *Outlook Express*, *Internet Mail and News*, and *Netscape Messenger* Registry hives. The virus keeps within its body the email address of the ten most recently infected users. If the current user’s email address is not already in this list, then it will be placed at the top of the list, and the other nine entries will be moved down. Then the search begins for the interesting files in the Program Files directory and the *Windows* directory.

After a one-minute wait the virus will check if an active Internet connection exists. If it does, the virus will search the Program Files drive for .DOC and .TXT files and choose from one of these files up to four words for the email subject and between 20 and 85 words for the email body. Additional code adds a period to the end of the email body and capitalises the first word, if required. Having formed the mail text, the virus will create the email headers, addressing the mail to up to 100 recipients, but explicitly avoiding the current user, and with 80% chance it will alter the second character of the return email address. This has the effect of preventing people from replying to the email, in order to alert the user to the infection. The X-Mailer string is always ‘Microsoft Outlook Express’, but the version is chosen randomly from a table containing five version strings.

The virus will then attempt to locate a file to send. The choice is made by examining the first 20 Portable Executable .EXE or .SCR files that are smaller than 128 KB. If no such file is found, then an empty email will be sent. Otherwise, one of those files will be infected and attached to the mail. The mail Content-Type will be set randomly to ‘image/gif’ or ‘application/octet-stream’. There is a 20% chance that the file from which the subject and body text were taken will also be attached to the email. The virus now sends the email and disconnects.

See Spot Run

There is a 25% chance the virus will search the Program Files drive for the first 20 Portable Executable .EXE or .SCR files, choose one, make a copy of that file, decrement the fifth last character of the filename, and infect that copy. If the *Windows* directory is not found, then up to 20 Portable Executable .EXE or .SCR files will be infected. The other 75% of the time, one Portable Executable .EXE or .SCR will be copied, the fifth last character of the filename will be decremented, the copy will be infected, and the Run key in the Registry will be altered to include a reference to the copy. The name of the Run value will be the filename without the suffix. This forces *Windows* to run the infected file whenever *Windows* is started.

After another one-minute wait, the virus will search each local hard drive for the first 20 Portable Executable .EXE and .SCR files and infect all of them. If the *Windows* directory is located on a drive that is not the current one, then the ‘run=’ logic will be executed for that drive. This code is also applied to every shared directory that is visible to this machine on the entire local area network.

Infection

Magistr infects Portable Executable files that are not DLLs, and are smaller than 1 GB. The infection marker is one of two values (0xCECD, or the first two characters of the computer name ORed with 0x9183), in one of three locations (NumberOfSymbols field in the PE header, or PointerToLineNumbers or NumberOfLineNumbers field in the first section header).

The first 512 bytes at the host entry point will be saved and replaced by polymorphic garbage, however this routine contains bugs that can produce either non-working code, or code longer than 512 bytes. A new polymorphic decryptor will be generated and the virus body will be encrypted. If a file contains a relocation section that is large enough to hold the decryptor and virus body, then the relocation section will be overwritten and the section name will be the first four characters of the computer name, preceded by a period. Otherwise, the virus will append itself to the last section in the file.

Seek and Destroy

Having completed the replication phase, the payload triggers are tested. If the machine has been infected for at least one month, if at least 100 people have been sent emails, and at least three .DOC or .TXT files contain at least three phrases from the list of 55 phrases contained in the virus, then the first payload will activate.

This payload appears to have been adapted from W32/Kriz, though it is functionally equivalent to W95/CIH’s. It begins by deleting the last file found by any of the virus search routines. Under *Windows 9x* and *Windows ME*, it will also erase the contents of the CMOS memory and flash BIOS, and overwrite a single sector on the first hard disk. This sector is always cylinder 0, head 32, sector 1. The location is never updated. Under all platforms, it will delete one in every 25 files on every local hard drive and shared network directory, and overwrite every other file with the text ‘YOUARESHIT’ as many times as will fit in the file.

After waiting less than a second, the entire first payload is repeated. This loop occurs infinitely. The second payload occurs after the machine has been infected for at least two months. On odd days, the desktop icons will be repositioned whenever the mouse pointer approaches. Given the nature of the rest of the code, it is likely that this routine is copied from another source. The third payload occurs after the machine has been infected for at least three months. Each time the injected routine is executed, this payload will delete the last file found by any of the virus search routines. Then, after every four minutes, the payload triggers are tested again.

W32/Magistr is certainly a complex virus, but presents nothing really new in virus writing. However, the virus is in the wild and could possibly become as widespread as W32/Funlove, and as damaging as W95/CIH.

W32/Magistr

Aliases:	I-Worm/Magistr, PE_MAGISTR.A, W32/Magistr@mm.
Type:	Polymorphic, EPO, memory resident, parasitic mass-mailer.
Infects:	PE .EXE and .SCR files.
Self-recognition:	Magic value in PE header of files, memory-mapped file in memory.
Possible Payload:	File deletion, flash BIOS erased, message box, moving icons.
Removal:	Delete infected files and restore from backups.

VIRUS ANALYSIS 2

Tossing the Penguin through a Broken Window

Jakub Kaminski
Computer Associates Inc, Australia

When the very first virus infecting both PE and ELF binaries was announced at the end of last month (see *VB*, April 2001, p.2), cries of wolf were heard once again and the hype was swiftly reflected in the media reports.

PE – Portable Executable – is a native format of 32-bit *Windows* programs, while ELF – Executable and Linkable Format – is a binary standard adopted by the most popular non-*Microsoft* Operating System, *Linux*. A virus which is capable of infecting files from those two different platforms is certainly worth looking at and, from a researcher's point of view, is good fun indeed if one considers hundreds upon hundreds of boring minor variants of the same old viruses we have to deal with on a daily basis.

However, what happens to be interesting to virus analysts does not necessarily have to be of concern to all PC users. Creating a sensation and spawning hundreds of virus alerts because of a piece of malware which cannot possibly become a widely spread threat is of course irresponsible and in the long run, harmful.

So, were those alarm reports of that new virus (named 'Lindose' by the anti-virus industry) justified? Should a wider range of PC users seriously worry about the new discovery which, by the way, lacks any payload whatsoever? Let us have a closer look inside Lindose and draw our own conclusions.

Executing an Infected Program under Windows

When an infected PE file is executed, the virus, which is located at the entry point, takes control. At the beginning, Lindose locates and searches the kernel address space in order to find the addresses of all the necessary APIs.

The virus implements two methods of locating the kernel image. The first, very fast one, relies on the known values which are most commonly seen in the real world. Lindose checks for the presence of the MZ and PE headers at the five selected locations: 0x77E00000, 0x77E80000, 0x77ED0000, 0x77F00000 and 0xBFF70000. These numbers are the base addresses of the KERNEL32.DLL for the following systems: *Windows 2000* – final release, *Windows 2000* – RC2, *Windows 2000* – beta 3, *Windows NT*, *Windows 9x*, respectively (see *VB*, August 2000, p.8).

When the method described above fails, Lindose applies the scanning procedure. Starting from the address 0x77000000, with the step of 0x1000 bytes, the virus checks for the presence of the kernel image. This piece of code does not have any limiting counter and since the virus implements its own exception handler, it will either execute until it finds what it is looking for, or end up in an endless loop.

Once the kernel image is located, the virus proceeds to find the addresses of the selected 15 kernel functions, namely: FindFirstFileA, FindNextFileA, FindClose, CreateFileA, CreateFileMappingA, MapViewOfFile, UnmapViewOfFile, CloseHandle, VirtualAlloc, VirtualFree, WriteFile, SetFile Pointer, GetCurrentDirectoryA, SetCurrentDirectoryA and OutputDebugStringA. These names cannot be seen inside the code since the virus uses only the checksums of the names. Lindose calculates 32-bit CRCs on names of all exported kernel APIs and matches the values to the ones hardcoded in the virus body.

The method of calculating the CRCs and some of the stored values are identical to those from known viruses created by the same author. If any of the APIs the virus is looking for is not found, Lindose abandons any further attempts and simply executes the original host program.

Next, the virus tries to infect all the files in the current directory and then all the files in its parent directory. This whole process is repeated up to 20 times. When a file is found, Lindose checks its size and rejects all files longer than 4 GB and all files shorter than 16 KB. Interestingly, a similar check is not performed while running the virus under *Linux*, which is why one cannot make any assumptions as to the size of infected files.

The virus infects only those PE files which have a relocation section present, which is listed as the last section. Moreover, the relocation section has to be greater than 2,631 bytes in length. Lindose infects only *i386* binaries and it does not infect DLLs or system files. The original size of infected PE files does not increase, because the virus stores its code in the first 2,132 bytes of the relocation section (over-writing its original content).

The virus infects only ELF files if a section which contains the code at the entry point is at least 2,784 bytes long. The virus overwrites the beginning of that section, but it keeps the original code and stores it at the end of the infected file. Infected ELF programs grow by 2,784 bytes. The virus completely ignores file extensions and identifies potential victims by analysing their format.

Lindose avoids infecting already infected PE files by setting the virtual address of the relocation section (offset 0x00A0 in the PE header) to zero. When verifying ELF files, the virus compares the first four bytes of its code with the beginning of the section which includes the code at the entry point.

Executing an Infected Program under Linux

When an infected ELF binary is executed, the virus code, which is located at the entry point, takes control. First, it makes sure the original command-line is remembered, then it relocates the virus code to the stack and continues executing it. Then the virus searches the current directory and tries to infect every file found there.

With every file found, the virus tries to open it and get its real size. Then Lindose tries to identify if a given file is an ELF binary or Win32 executable. The former is identified by the 'magic' string: '[0x7F]ELF' (and, once confirmed, the virus proceeds to infect ELF binaries), the latter by the signatures 'MZ' and 'PE' at the start of a file and at the start of the extended header, respectively.

Additionally, a PE file suitable for infection must be an *Intelx86* executable, but it cannot be a DLL or a system file. It seems that Lindose also tries to avoid infecting system drivers, but instead of testing the Subsystem Flags (word at offset 0x5c in the PE header) it tests the byte storing the Subsystem major version number (byte at offset 0x48 in the PE header).

Lindose determines if the potential victim has more than one section, if the relocation section is the last section of a file, and if it is at least 2,632 bytes long (its 'virtual size'). It gives up if these conditions are not met. Otherwise, it sets two values: the virtual address of the relocation section (other-wise known as the Fixup Table) and sets its size to zero. This marks the file as one which does not use the relocation data and prevents the virus from reinfecting already infected files.

Almost immediately afterwards, Lindose checks the '.reloc' section again – this time against the size of 2,784 bytes. If the size is at least 2,784 bytes, it continues; if it is smaller, it extends the section virtual size to the next section alignment. Next, the virus modifies the file's entry point, pointing it to the beginning of the last section while storing the original values of the entry point and the Base Address inside the virus code.

Then it copies 2,132 bytes of the effective virus body to the start of the relocation section and makes sure that this section has its attribute set to 'Writable'. The infected file is closed and the virus moves to the next target, unless all suited files in the current directory are infected.

If a file is recognised as ELF binary, the virus makes sure that it is an *Intel386* binary (interestingly, it doesn't bother to check if the file is executable). Then it tries to locate the entry point. The way the virus locates it and the number of assumptions it makes seems to confirm that, as far as ELF binaries are concerned, this particular virus author still has plenty to learn.

First, the entry point is located by parsing the Section Header Table. ELF executable files do not have to have sections or Section Header Tables at all. Secondly, the first section which ends higher than the Virtual Entry Point specified in the header is assumed to be the one containing the code at the entry point.

This approach, although it usually works in practice, may cause the virus to target the wrong section. Lastly, the virus makes another, quite unsafe assumption – that the entry point always matches the start of a section. If it does not, an infected file will be unable to execute since the virus sets the new entry point simply by adding 0x545 to the old one rather than calculating it from the start of the section. Interestingly enough, such corrupted files can still be cleaned and restored to their original state.

Despite some mistakes, during lab tests the virus seemed to work without bigger problems. However, the spread of Lindose in the real world is limited by conditions necessary for successful cross-infection. Not only that, having both types of binary files – PE and ELF – stored on one machine is not common, but also having these two types of executables in the same directory (or directory tree) is still a very rare scenario these days.

Moreover, even if we forget about the virus supporting two binary formats and treat it as two separate viruses (one PE and one ELF), current statistics show that binary, non-resident, direct infectors cannot possibly be considered a great threat in the real world.

Conclusion

Summarising, one has to acknowledge the implementation of a 'new' idea with the Lindose virus. However, at the same time, one has openly to dismiss all hype emanating from the anti-virus industry as completely unjustified. In its current form, Lindose cannot constitute any serious threat to anyone.

Lindose

Alias: Winux.

Type: Direct infector.

Infection: PE executables and ELF *i386* binaries.

Self-recognition in files:
PE – the relocation section unused.
ELF – bytes 0x60 0xE8 0x09 0x00 at the start of the section containing the code at the entry point.

Non-displayed text:
'[Win32/Linux.Winux] multi-platform virus by Benny/29A' and 'This GNU program is covered by GPL.'

Trigger: None.

Removal: Identify and replace infected files.

FEATURE 1

The Internet – The Virus Writer's Friend

Richard Wang
Sophos Anti-Virus, UK

The Internet is part of daily life. People use it to exchange email, play, shop, manage their finances, read, watch and listen to the news and much more besides. Unfortunately, malware authors are not ignorant of the diversity of services and protocols available on the average desktop. Towards the end of 2000 and during 2001 network-aware viruses for the *Microsoft Windows* platform have enjoyed significant success in the wild. The most obvious (and most common) reason for this success is the use of Internet email to spread as email attachments.

There are so many different viruses that do this now that it is not my intention to discuss them here. There are other ways in which the authors of viruses and other malware choose to use the Internet and it is these that will occupy the rest of this article.

The operation of a virus is not limited merely to spreading. A spreading mechanism will earn it the classification of virus or worm, but many virus authors choose to implement much more in their programs. The public face of a virus is its payload, but behind the scenes it may be updating itself, reporting infections to the author or distributing information from the infected computer. Once a system has an active infection, most of these activities will be invisible to the normal user. Only an examination of network traffic to and from the machine would reveal the virus' activity.

What are they Doing?

To make use of the brave, new, virtual world a virus needs to communicate with other machines. Doing so involves the use of a networking interface. The methods for doing this range from the simple use of APIs – leaving the details of the operation to the Operating System – to the virus containing its own client software and needing only a socket connection to the machine it is communicating with.

APIs supplied by the Operating System are usually used for file transfer. Viruses download files as updates to their own code or in order to insert further malicious code into the infected system. Uploading files is often used as a means of reporting infections either to the author, a specific person, organization or the world in general. Although moving files to and from a computer can give a large measure of control over that machine, it is often easier to use a more complex form of communication. Exerting some form of control over the behaviour of the virus itself is usually implemented through a custom interface.

The most common use of complex network communication in malware is not in viruses at all, but in backdoor Trojans. These programs often have their own client/server protocols but a few do use standard protocols to obtain access to a machine. Some backdoors contain simple FTP or HTTP servers; once the backdoor is active, the affected machine acts as a file server open to anyone with standard client software such as a Web browser.

Other backdoors have no networking components of their own but use the scripting facilities of other programs such as Internet Relay Chat (IRC) clients to communicate between client and target machines. The majority of backdoors, however, use their own custom protocols and client software enabling a greater range of control over the target machine.

How has Network Use Evolved?

Viruses' use of networked systems has become more complex over time. As with other technical advancements, the rate at which virus complexity has increased is not necessarily uniform. Some virus writers embraced networking earlier than others and some VX groups use more advanced techniques.

One of the simplest network protocols a virus can use is *Microsoft's* network services. This enables viruses to search a network from a single infected machine. Simple API calls enable the virus to gather information about available network resources and infect any files to which the infected machine has access. Effectively, the virus can treat the network filesystem just as it does the local filesystem on the infected machine. While it is relatively easy to implement this technique, the virus does not actively spread itself beyond the network it has infected.

To spread to remote networks viruses can exploit another facet of *Microsoft* networking – namely, file sharing. There are many systems connected to the Internet which not only have file sharing enabled, but which have writable shares not protected by passwords. These systems are easy targets for viruses.

The only slight problem they present to the virus author is how to find them. A simple scan of subnets using common default share names will reveal their presence to the virus. The subnet chosen can be hard-coded into the body of the virus, chosen according to the network configuration of the infected machine or simply generated at random. As long as the virus remains undetected the infected machine can continue to search for further targets.

Virus writers who release their viruses into the wild may be interested in how widespread their virus becomes. How do they find out this information? Various anti-virus and security Web sites list common viruses, some give them a threat rating, others indicate where the virus has been seen. Another way to obtain the information would be to give the virus the ability to report infections. A simple report that the machine is infected is merely of interest to the recipient but some malware takes the reporting a step further.

Network address details and password information about the infected machine can be included in a virus report. This information could reveal a security weakness and open the door to further attacks on the machine. It is not uncommon for backdoor Trojans to send a report once they become active, alerting an attacker to the vulnerability of the target machine.

The usefulness of this kind of reporting can be limited. Reports are of no use if they cannot be received. Many are delivered by email or using an instant messaging system and the accounts receiving these messages can easily be traced and shut down, even if their owners cannot.

A less traceable method of reporting is to send a message to a public forum such as a USENET discussion group. If the message identifies the infected machine it is then possible to advise its owner of the infection and appropriate countermeasures. Curiously, some virus authors choose to have their viruses report to an AV company.

One of the more interesting developments from an anti-virus researcher’s perspective is that of viruses capable of downloading updates or plug-ins for their code. Traditional virus scanners rely on frequent updates to detect new viruses, updates which are usually available from the vendors’ or distributors’ Web sites. Protection against new viruses is usually available shortly after they are discovered. Virus authors have realised that updates can also be useful to their creations.

Early attempts, such as W95/Babylonia, involved the virus contacting a web or FTP site to download another program which it then runs. Alternatively, a virus may cause an installed Web browser to download the file and obtain the plug-in code from the browser’s local cache. In order to avoid arousing the suspicion of the user, the plug-in can be attached to another file such as an image which is normally displayed by the browser.

As it is relatively easy to trace and cancel an email account, most Internet service providers will close a site for hosting viruses. To avoid this problem, a virus author may allow the update to change the site from which future updates are obtained.

This particular approach presents its own obstructions to the success of the virus. If infected machines are to have access to the updates then AV labs can also obtain and analyse them. The sites hosting the updates will still be closed and the updates will only succeed if an infected machine obtains the latest update, including the location of the next one, before it is removed. The virus author must continually supply fresh updates.

A more successful solution is that used by W32/Hybris. Its plug-ins are posted to a USENET newsgroup via anonymous gateways. The virus can then use a free news server to obtain updates. Anonymous gateways serve many useful functions and USENET is a distributed system making the plug-ins hard to cancel.

It has been suggested that the virus could be successfully neutralised by supplying it with a plug-in which causes it to remove itself from the infected machine. Apart from the ethical implications of running code on machines of unsuspecting users, the virus uses digitally signed plug-ins, meaning that only the virus author can produce plug-ins which the virus will run.

Another technique used by W32/Hybris is to allow infections to communicate between themselves. Each infected machine can post the updates it has to USENET to be obtained by other infected machines. The virus itself acts merely as an engine for distributing and executing the plug-ins. In the case of W32/Hybris, the plug-ins have included changes to the distribution mechanism, new infection techniques and a payload.

The Last Word?

There is no doubt that computer viruses have become significantly more sophisticated in their use of networked systems. Fortunately, most of the useful countermeasures available to users remain the same. Regularly updated anti-virus software is important, but safe computing practices must still be followed. If the virus does not reach your network, it cannot exploit it, no matter how complex it may happen to be.

FEATURE 2

Full of Sound and Fury?

Berni Dwan
Freelance technology writer, Ireland

If, according to Jonathan Swift, ‘War is the child of Pride, and Pride the daughter of Riches’, then the ongoing war of attrition between the virus writers and the anti-virus vendors deserves some scrutiny. Could it mean, perhaps, that the war waged upon society by virus writers springs from a pride that is misplaced, while the pride of the anti-virus vendors in their battle against the virus writers is most definitely a source of riches?

I had originally thought of this as a battle, but changed my mind and relegated it to a war of attrition for several reasons. Firstly, a battle is dramatic and exciting, like any of the great battles we know from history, while a war of attrition is incessant, debilitating, repetitive and predictable. A battle has a definite end, whereas in this war the virus writer does not seek outright victory, recognising this as a futile ambition. Instead, the object is the wearing down of the victims by constant disruption and downtime resulting in loss of productivity and consequently loss of revenue.

While cynicism is unproductive, I am beginning to become cynical about the ageing virus phenomenon, and sometimes I wonder if some virus writers and anti-virus vendors are clenched in some kind of demonic pact. Rob Rosenberger sees it more as a symbiotic relationship, ‘Take Superman for example. Wherever he goes, he finds a supervillain to combat. Every once in a while, Superman will hang up his cape and leotard and say “no more”. What happens? The supervillains force him back into action. Crime won’t go away if you give up protecting everyone. So, yes, anti-virus vendors have a symbiotic relationship with virus writers. Give the good guys credit: they perform an absolutely essential service.’.

There is no doubt that hyperbole in the media gives those who are more lax about security a sudden wake-up call, and the anti-virus vendors’ phones are hopping. In the case of viruses like Melissa or Love Bug the wake-up call was warranted, but increasingly the hyperbole amounts to nothing more than a punctured wet football. Disappointing or what? When I shared this cynicism with Fred Cohen, he thought that perhaps I was going a bit far. ‘It’s just that people want entertainment rather than function from computers and the programmers we have created are not very good at what they do. Add in vendors anxious to do foolish things for the cool of it, corporations making gobs of money from hyperbole, and you have a recipe for disasters.’

Speaking about ‘gobs of money’, my cynicism was further fuelled by a recent article in the *Wall Street Journal*, reporting that Security officials in Beijing have been requiring that leading anti-virus software companies must provide samples of destructive computer programs and rogue wiretap software. This in exchange for being allowed to sell their products in China!

In 1999/2000, *Network Associates Inc*, *Symantec* and *Trend Micro Inc* gave the Chinese security ministry roughly 300 different samples of the most common, malicious software found on the Internet, in exchange for permission to market their products in China. *F-Secure Inc* of Finland said it negotiated last summer to let Chinese researchers conduct virus studies at its new laboratory in Beijing, but declined to surrender the samples directly. Interestingly and unusually, *McAfee* President Gene Hodges said that within 90 days of complying with the Chinese request, his company notified the U.S. government that it had provided the samples, adding that the government officials with whom the company spoke expressed no specific concern.

Naturally enough, there *is* some concern about the potential military usefulness of the common viruses turned over to China. It is the request to trade virus samples and other software programs for market access that is astonishing, not to mention the companies’ uptake of the offer. Apparently, software companies had negotiated to hand over to China only samples of relatively common viruses and not their more substantial collections of tens of thousands of dangerous programs.

Notwithstanding, the *Wall Street Journal* article notes that China’s military is developing a ‘Net Force’ of young computer experts trained in information warfare. Furthermore, it reports that in late 1999, the Chinese army's official newspaper discussed the need for 'software and technology for Net offensives so as to be able to launch attacks and countermeasures on the Net.'

The use and abuse of hyperbole will only serve, in my opinion, to create a jaded audience who will merely lapse into indifference or intolerance, depending on their disposition, their energy levels and their priorities. The discerning public might just start to place viruses further and further down their list of concerns.

Rob Rosenberger sees users being taken for a ride both by the anti-virus industry and the media – 'The media has a fetish for juicy virus stories, and the industry prostitutes itself for free ink.' So, if the purveyors of unnecessary hype are also the purveyors of necessary anti-virus solutions, are they not then gradually cutting off the hand that feeds them, and ultimately giving the laurel wreath to the virus writers?

Happy in the knowledge that I am not a voice in the wilderness, I was relieved to read an article by Carole Fennelly expressing similar concerns. The article was prompted by a radio news item giving dire warnings about the latest 'hacker danger lurking on your PC'. The result of a press release from a previously unknown security company (*NetSec*) about an apparently new Trojan called Serbian Badman Trojan, it transpires that the Trojan was already in the wild and known as SubSeven. If the company discovered a potentially dangerous situation with regard to a known Trojan, asks Fennelly, wouldn't it have been more appropriate to alert the anti-virus vendors or at least check the signatures with them?

While many industry experts considered it to be nothing more than an attempt at cheap publicity by a relatively unknown computer security company, Fennelly acknowledges that news about security incidents helps sell security services. Even though she herself is a partner in a security company, she is enlightened enough to see the downside of such revenue-generating panics. 'I plan to be in this industry for the long term. Eventually, people will get immune to hearing that "the sky is falling" and ignore all security warnings.'

Fennelly believes that there are people who obviously benefit by exploiting FUD (fear, uncertainty and doubt). While acknowledging that a bit of sensationalism is sometimes necessary to get the appropriate resources to address a problem like in the case of the over hyped Y2K, which did albeit require attention, she concludes, 'Sometimes a little hype is a good thing. Too much, though, will eventually backfire.'

Similar sentiments were apparent in a 1999 article by Dave Gussow, whose introduction reads, 'The panic generated by computer viruses such as Melissa has exceeded the damage. While the potential for harm is real, computer experts say the paranoia is overblown.' Gussow reported that not everyone was persuaded that Melissa posed the threat to computer security that officials painted. He cites Rob Rosenberger, who saw it as another in a string of over-hyped virus alerts that turned out to be much less significant than had been predicted, but that helped software companies sell anti-virus protection. Quoted in the article, Rosenberger said, 'I do believe the world needs anti-virus software.'. However, he objected to software companies' marketing, saying they amount to 'immoral' scare tactics to get people to buy their products.

Rosenberger's Virus Myths Web site's motto 'Mundus vult decipi' (the world wants to be deceived), helps us to understand where Rosenberger's loyalties lie, and that is not to anti-virus companies or the media. AV software companies, he said, foster 'fear, uncertainty, doubt' about the dangers of viruses, helped by a public still learning about computing and media hype surrounding such events.

Continuing along my road of cynicism, I toyed with the idea of leaving my fate to chance, with suites of security products becoming so complex – or perhaps this is irresponsible? To put it bluntly, effective information protection is complicated, so I just have to face it head on. 'People want everything simple', says Fred Cohen, 'but everything is not simple. So they pay people to make it seem simple, and they get burned, and its part of the price they pay for oversimplifying.' Rob Rosenberger was more forthright in his response to my seeming irresponsibility. 'If you do leave your fate to chance, let me be the first to say au revoir!'

I speculated that perhaps this whole virus phenomenon would turn a corner and give the anti-virus vendors a run for their money, or was I now moving into the realm of fiction? Fred Cohen's response is one word, 'Fiction.'.

Rob Rosenberger sees change in the next two to three years. 'The anti-virus industry spends a lot of money to support users who update their software. Those users pay the same amount whether they update once or a million times, but vendors must pay for all the needed bandwidth and hardware to support those updates. Add to this the fact that vendors need more bandwidth and hardware with each passing scare.' Due to the resulting heavy financial impact, Rosenberger sees proactive anti-virus technologies augmenting the traditional reactive technologies, to decrease bandwidth costs.

In conclusion, and to allude to the novelist Swift again, the war of attrition between the virus writers and the anti-virus vendors is as futile as the battle Lemuel Gulliver encountered on his famous travels. That particular battle was fought between two parties, those who decreed that eggs should be broken at the small end versus those who objected vehemently and believed they should continue to be broken at the big end! It occurs to me that the anti-virus 'big-endians' and the virus writer 'small-endians' will continue to be slogging it out in that cyber amphitheatre for some time to come.

DEBATE 1

All Virus Disinfection is Evil: Discuss

Dr. Richard Ford
Cenetec LLC, USA

The next couple of pages have been brewing for a long time; the idea of debating an issue, no holds barred, is fun, and, I hope, informative. What better issue than virus disinfection, and what better sparring partner than Nick FitzGerald? Better yet, I happen to have occupied the 'against' position; Nick, with all due respect, this is a debate you cannot win!

You cannot win for any number of reasons. I will lay them out here one by one. I'm sure you can predict many of them, and it is going to be interesting to read your responses. When viruses first came out, they infected things like boot sectors and files – *executable* objects. While one could argue that it is a given that an infected MBR must be disinfected, one could also claim that this is not really disinfection, but more of a restore, as most 'cleanings' simply involve copying the unmodified boot sector back into place. For the sake of argument, I will agree with this.

But what of infection of an executable? Is disinfection universally wrong? I think not.

Clearly it is *better* (i.e. safer) to restore an uninfected copy of the file, but what if there is no copy? What action should one take if there is no back-up? Should one junk the computer? Buy a new copy of the Operating System/application? Reinstall, losing all customization? I think not. Disinfection is the answer.

As if this were not compelling enough, a huge change took place when the Concept virus was discovered: there was now a common virus that infected a file that was usually considered only consist to of *data*. Data is transitory and changing; there are many times that such a file will not be backed up – at least, not in its current state.

Thus, one is faced with a situation where even in the presence of a reliable back-up policy, it may not be possible to 'recover' the work. Arguing that a solid back-up is the only solution, while true in many circumstances, is not convincing in the presence of *data-infecting* viruses.

Moreover, few, if any, businesses have a completely robust back-up system. While recovering from a back-up is a better solution to infection, the reality is that this is frequently either impossible or expensive. Thus, once again, disinfection becomes a viable, or even preferred, solution.

The most powerful debunking of the position, though, comes from the simple premise that the cure itself should not be worse than the disease. If the answer is having the *Word* document (or even *this* document), on which I have been working *all* (well, okay, a large proportion) of the morning *deleted* rather than chance the oh-so-risky 'delete all macros' disinfection, I will take the risk, thank you so very much.

While 'total cost of ownership' and 'return on investment' may seem like clichés in the current economic environment, they have attained that by being underlying truths in terms of business decision-making. Every dollar spent must in some way advance the business, such that there is a positive return on that dollar – be it in risk avoidance, tangible return, or intangible growth.

Here, then, we have another strong argument for the role of disinfection: a huge decrease in the total cost of ownership. Disinfection helps minimize the impact of virus infection, allowing the business to carry on as normal.

While there is some small risk posed by unreliable and/or incomplete disinfection, the cost of avoiding this risk is large in terms of lost productivity, downtime, IT costs etc. If the incident requires manual intervention (manual kickoff of the restore process, manual request for a 'clean' copy of the infected document etc.) the costs increase further.

Compare this to a resident scanner, which automatically detects and repairs, while at the same time reporting the incident to a central company database. Nick will doubtless point out (correctly) that this is not 100% safe. I agree, it is not… but then again, neither is simply *booting* your computer, yet one takes this for granted because the rewards justify the risk.

Let me present a simple, and hopefully illustrative, analogy. Last time I visited my physician I was given an antibiotic. A little research with the manufacturer of the drug and a short search in the *Physician's Desk Reference* told me that there was a non-vanishing chance I would suffer a *severe* allergic reaction to it; further, that in more than 2% of people, side-effects resulted in treatment being discontinued. I took the pills anyway, *because the rewards outweighed the risks*.

That is really the ultimate argument, Mr. F, and no matter how eloquently I am sure you will phrase things, it is a truth you cannot avoid.

DEBATE 2

Some Disinfection is Evil

Nick FitzGerald
Computer Virus Consulting, NZ

It is impossible to debate the ‘for’ side of Richard’s proposition sensibly. So, at the superficial level, he wins but I’ll give some arguments about why, where and what disinfection is ‘evil’. Richard conceded some clearly unwinnable points to me. Then in his discussion of executable files he asks ‘Is disinfection universally wrong?’. I must answer even that with a considered ‘No’. However, I must also qualify this by adding that *much* disinfection of executable files is *very wrong*.

He then asks whether it is not better to disinfect than either to restore from backups or reinstall from originals. Ignoring for a moment the issue of which viruses it may be acceptable to disinfect, I suggest that the users in his example are equally unprepared for all manner of *other* likely systems disasters (hard-drive crash, theft, fire, earthquake, accidental file deletion, etc). Being properly prepared for all sorts of catastrophes likely to occur to your systems and data means you will necessarily be well-prepared to deal with a virus disaster *independent* of your AV software’s ability to disinfect the virus(es).

No-one expects the AV industry to assume *any* responsibility for assisting in the restoration of machines damaged by fire, flood, etc. Competent computer administrators will be prepared to deal with a virus disaster through their preparations for handling those other kinds of disasters.

Why is it often unsafe to disinfect executables? The common view of parasitic infection and disinfection of executables is that a virus adds its code to an executable, and when discovered that code is removed, supposedly leaving the host file as it was prior to infection. This is a naïve and almost entirely misleading view of what commonly happens. Even quite simple parasitic viruses do more than just add their code to executables. Viruses usually have to alter values in an EXE’s header. Host files are usually increased in length as a result of infection, and with PE infectors it is common for sections to be added.

This still sounds fairly simple, and thus presumably easily reversed. However, the reality is that often the changes a virus makes are not precisely reversible *unless the original state of the host file is known*. Simple EXE infectors commonly pad the host’s length to a paragraph boundary then append their code. When disinfecting such viruses, unless the host’s original size is recorded in the virus, disinfection ‘restores’ the host plus up to 15 bytes of padding. Apart from a few programs that make self-integrity checks, such changes are ‘harmless’ to the programs themselves. However, with more complex infection schemes (particularly with PE infectors), the problems multiply. New variants of mass-mailing executables have been created when copies of a known mass-mailer have been imperfectly disinfected of a parasitic infector they picked up in transit. This has happened many times despite the variant-creating scanner being able to detect the pre-infection form (see p.14 of my *VB 2000* paper or *VB* Feb 2001, p.16).

A big change did not occur when the Concept virus was discovered. First, ‘delete then restore’ need only apply to the macro code components of a file, and not to all the file’s content. In fact, most macro disinfection routines do just that, deleting all macros (or all virus-containing modules) from infected files, and leave it up to the user to restore or recreate any legitimate macros that may also have been removed. Corporate users have been dealing with this for several years now, so why not with executables too?

The inadequacy of backup systems is a poor argument here for the same reasons it was earlier. If ‘truly important’ work is being done in your word processors and spreadsheets, competent system design would employ journalling systems allowing roll-back and roll-forward, much as it is in crucial database systems such as those that track bank account balances. Expecting AV software to compensate for the inadequacies of popular applications is like expecting *AV* software to assist after a PC has been damaged by fire. Better that the software choosers specify and implement the system they need, rather than settle for the limited combinations of inadequacies Redmond deigns to ship!

Much of the rest of Richard’s case is based on a flawed cost/benefit argument. At its core is the unsubstantiated – despite its mass-repetition throughout the sector – assumption that it is cheaper to configure general purpose IT systems so users can trash them, then only ‘pay’ to fix those that become ‘unusable’. The real cost of the loss of a system’s data integrity is intangible, but immense. Properly configuring systems to allow good integrity maintenance, thus dramatically reducing the chance of such losses, is not substantial. As I argued in my *VB 2000* paper, the cost should be no greater than the incremental cost of adding an on-access scanner to well-planned and maintained system rollouts – a cost currently widely accepted despite its considerably lower benefit in terms of threat-reduction.

Finally, the antibiotics analogy is utterly bogus. Treating an individual ‘cell’ within the Internet ‘body’ works as an analogy only if the ‘malaise’ he was treating had the explosive distribution potential of a new mass-mailing worm created by imperfect disinfection. If that were the case, he would not be walking the streets, going to work, or self-medicating – his condition would have him in a bio-hazard containment ward and his doctors in moon-suits!

A DAY IN THE LIFE

You don’t know me but …

Peter Agocs
VirusBuster Ltd, Hungary

My name is Peter Agocs and I was born behind the Iron Curtain, in Hungary. Despite the possibilities open to me in my youth, computers made a deep and lasting impression on me very early, and dependence evolved very quickly. Partly because of this, I graduated as a programmer mathematician. I managed to bump into viruses rather early but if you think that this changed my life, you are wrong; I only met a virus infection as the sufferer of it.

I was buried deep in the swamp of game programming then and was a bit of a stray lamb. Having taken part in several projects with varying degrees of success, I had a sudden moment of clarity – ‘this is not the path I am meant to be taking!’ Unfortunately, the guiding voice from the heavens stopped talking at this point and I had no idea where to go.

Destiny stepped in around April 1996 and I suddenly found myself among the developers at *VirusBuster Ltd*. After a year of hard work, I was honoured with the task of leading the Development team, which I have been doing ever since. I have three children; one of them is my own ‘development’ and I ‘received’ the other two along with my wife. Yes, I like a challenge!

Just an Average Day

At 7am, one of the hardest periods of my day begins as my ‘boot sequence’ starts. In a mere hour and a half I succeed in getting myself ready and I prepare myself for the depressing traffic. The road provides a more or less pleasurable morning as it leads along the wharf of Budapest, but the behaviour of my fellow drivers on the road is disappointing. The truth is that it is hard for me to tolerate clumsiness. In an average half hour, I am able to cope with this unpredictable 13 kilometres and my day begins.

Daily Compulsory Exercises

As I arrive, my assistant warns me about the most important tasks of the day and the early morning calls. Meanwhile, I connect myself to the vital central tea supply with my favourite oversized mug interface. This operation is repeated several times during the day if the interface buffer is empty. Equipped with my caffeine quota, the usual morning rituals begin.

The first is the backlog of electronic mail. I run through the messages which have arrived since the previous evening and I tackle the most important ones. I automatically remove the server’s warning that I have overrun my mailbox size limit and swear that I will review the old messages and delete them (of course this will not happen). The next step is a quick review of on-line news and Web statistics and a visit to some important pages – generally nothing unusual, which is reassuring as it doesn’t upset my whole day. So, calendar, what shouldn’t I forget today? Phone calls, deadlines, meetings, questions that have not been answered… I won’t be bored today.

The review of the developers’ daily reports is my next task, combined with the tackling of new problems and how to solve them. Oh yes, this would be my next task, but the phone rings; it is the system administrator of an important client – a TV company (one of the best ones here in Hungary). After a short discussion, it turns out that there is a problem as the most up-to-date version of our program is not working properly on one of their machines – the machine of one of the most important people in the company. Discussion with Support follows, and I decide to visit the client myself – just a little contact maintenance.

Let me tell you one of my favourite stories: a few years ago there was a support call from an accountant’s office. They had problems with a One_Half infection which they were able to remove with the help of an AV product, but their computer hadn’t been able to start up again and the last back-up was two years old. My colleague told him to bring the machine in and we would see what we could do. The user arrived with the poor victim within the hour. Our man embarked on the task and he left happily two hours later. This would not be interesting in itself, but a year later he contacted us again and said that he had a problem: One_Half. The situation was the same as the year before apart from the fact that the last back-up was three years old! The data was restored again but this time we made a back-up without asking and told him again how he could avoid such problems. He hasn’t contacted us since.

I finish summarizing unfinished tasks, problems, plans and ideas and, armed with my notes, I visit the developers. On their door there is a sign: ‘Disturbance is forbidden and dangerous!’ – a timely reminder. Before a new version comes out, they work several nights in a row and, of course, they get frustrated with the continuous ‘pings’ due to deadlines. The situation never gets catastrophic, but the cleaners sometimes think so as they cope with piles of pizza boxes and energy drink bottles.

All tasks are assigned and discussed in Development. Support and system integration come next. There’s a quick briefing about development discussions then an enquiry about the status of current projects. Everything seems to be running smoothly (I am always suspicious at these times, wondering if we have done anything wrong), everything is prepared for the project discussion which takes place in the afternoon. The last stop of my ritual pilgrimage is a visit to the virus lab. I quickly gather information on actual events and if there is an interesting virus about, I ask for detailed information so as to be able to pretend that I am fully informed about the problem if it occurs somewhere.

Then I return to my everyday tasks. I have to cope with messages which haven't been answered or have arrived in the interim before turning my attention to projects, development plans, specifications and other less inspiring things. I always have problems with this – as soon as I am able to get round to them and type in the first few letters, usually someone senses it and comes to bother me with an urgent matter. For example, today several people received the latest hoax doing the rounds and they wanted to understand the exact situation. Of course, it always turns out that they know what the exact situation is, but they want to be absolutely sure about it.

That reminds me of a case in the recent past, when a hoax caused havoc as its content appeared on a news provider. The press asked for information. We managed to explain to them that this was not another LoveLetter-type problem, but one of them published an article in which my name was the only thing reported correctly… of course, I was very ashamed of this and it was rather embarrassing.

Pushed for Time

Another phone call, but this time it is one of my colleagues informing me that we should set off to a meeting. As always, I am late again. The topic of this meeting is the current status of an on-going project, which seemed very interesting when it was started, as we integrated several other manufacturers' products into the system (security products, of course). The procurer is an important bank and we participate in the project through a big system integrator company. After major delays, the time finally came recently for us to join in the process and, of course, all of a sudden everything became very urgent.

We deal with the items on agenda, the second stage of licensing and the presentation of products. There are teething problems. Last time, everything was in order, apart from the fact that there was another product on the data carrier although the licence was correct. Now, the procurer's name is not right on the licence of the other product as it has been translated into English and the procurer doesn't want to accept it this way. After this we discuss further steps. Conclusion: software presentation prolonged. Exit stage right…

On my way to the TV company I encounter a common scene at a traffic light: a homeless man selling a weekly magazine. Its front cover is always very brightly coloured so that everyone can tell that it is a new issue. With the help of a little gymnastics, I am able to take some money out of my pocket, wind the window down, pay, express gratitude, wind the window up, and put the paper on the seat. After some more traffic lights, the situation is the same, so I produce the magazine as evidence. There is the usual afternoon traffic jam in the city and the client calls to ask if we are scheduled to arrive there today, as he is only free until 7pm. I calm him down, telling him that we are on the way, at least the man in the car behind me thinks so!

We arrive at last to be greeted by a rather pretty receptionist, who says 'Give me a name!'. The thought crosses my mind 'Poor little thing, what a miserable existence not to have a name!'. At last, I manage to squeeze out some words. The head system administrator approaches and bombards us with his experiences. I tell him that the error which had caused his problem has since been corrected.

We proceed to the client, who has problems. I am a bit moved, as this person is very famous in Hungary and I saw several of his TV programs as a child (I had time for that then). Meanwhile, the installation has ended, the problem really has been solved, everyone is satisfied and we can go home. On the way down, it turns out there is another problem. There is an *Excel* file which the program doesn't like. A few clicks sorts it out. After a short situation review we agree that he will send the file for further analysis. We talk for a while before leaving, but meanwhile I remember the funny situation I encountered, so I tell him about it and we have a good laugh.

Today has passed very quickly again. After some thinking I decide to strike! I won't go back to the bunker. On the way back I am very happy when I realize how early I will arrive home. I can already see the big cigar in my mouth and hear the soothing music in my ears… then suddenly I remember Ceskie's deadline. I'm fighting with myself until I arrive home repeating one word: 'tomorrow'. As soon as I arrive home, it turns out that I have lost the battle so I sit down to write this article (at least I can have that cigar!).

PRODUCT REVIEW 1

Norman Virus Control v5.0

Matt Ham

The release of *Norman Virus Control v5.0* earlier this year marked a significant change in the product, in a market where version numbers are all too often changed simply due to engine modifications behind the scenes. In this case, the scanning engine did indeed alter, but even more remarkable were the changes to the entire design philosophy of user interaction with the scanning process.

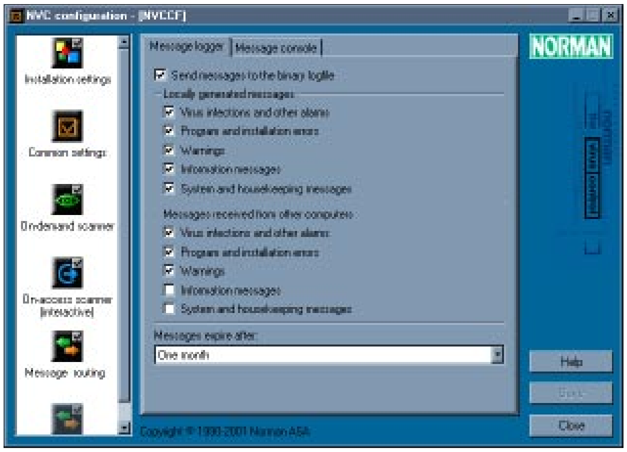
Norman as a company has been a long-standing participant in the *VB* Comparatives, though a little background information is not amiss at this point. Mainly concentrated in Scandinavia, *Norman's* services are largely security-related – encompassing access control, cryptography, security analysis, secure data erasure and data recovery. These are divided amongst what amount to sister companies of the anti-virus arm, which may explain the relatively small amount of demonstration software and information on these other activities provided on the CD I was sent.

Installation

Installation of *Norman Virus Control* (*NVC*) was performed from this CD, which constitutes the bulk of the package supplied. The CD is packaged in a small cardboard wallet with a 12-page mini installation guide – a far cry from the weighty tomes of some other products. Roughly half this mini guide is taken up with descriptions of the *Norman* product range – more information, oddly enough, than on the CD itself, leaving only space for a bare bones description of installation and scanning. Installation is simple enough, though scanning might be a little more confusing for those weaned on more standard scanning interfaces.

The products supplied on this CD are *NVC for Windows 95*, *98*, *ME*, *NT4.0* and *2000*, OS/2, *NetWare*, Groupware, *Exchange* and *MAILsweeper*. In addition, trial versions of *Norman Personal Firewall* and *Norman Privacy* are included for evaluation and *Adobe Acrobat Reader* is available for viewing the main documentation. As is indicated by the small size of the supplied package, the documentation is supplied electronically. What is more, this PDF documentation is formatted for printing rather than for viewing on-screen.

The front-end for the CD is activated by autorun and produced as a Standalone Macromedia Flash application. It sports a rather lurid set of graphics shared with the other packaging which brought alternate cries of ‘wow!’ or ‘yeeuugh!’ depending on the sanity of the observer. The first choice available here is one of language, with English, Norwegian, Danish, German and Swedish being available. Most of the options selectable are informational, though the obvious exception is the install routine for *NVC*. When activated, this brings up another language prompt, with the same choices as before.



☑ Send messages to the binary logfile

Locally generated messages

☑ Virus infections and other alarms
☑ Program and installation errors
☑ Warnings
☑ Information messages
☑ System and housekeeping messages

Messages received from other computers

☑ Virus infections and other alarms
☑ Program and installation errors
☑ Warnings
☐ Information messages
☐ System and housekeeping messages

Messages expire after: One month

Copyright © 1990-2001 Norman ASA

Like so many others of its kind, *NVC* is installed by means of InstallShield, though without the three mysterious fluctuating bars associated with that program in the past. After passing through the licence agreement, the first real stage is the Customer Information section, the information required here being user and company name, plus the serial number for registration. The registration number does suffer from being simply a long string, rather than split up into separate boxes as is the current norm with registration, and thus could be more prone to error.

The main setup options follow on, selecting whether *NVC*, Network Distribution Directories and Administration tools are installed. In a spirit of perversity no options were selected, producing an appropriate error, and thereafter for the main tests just *NVC* was selected. Administration tools is a category only really of relevance in a networked distribution of *NVC* and is mentioned later.

Following this choice comes a selection for where the installation should be performed and the confirmation that all selections are correct and that installation should proceed. While installation is in progress, notification boxes are displayed as to what precise part of the installation is occurring, which, in the standard installation state, results in a gradual and regular widening of this display area as the messages seem to have been sorted into ascending size order. Whether this is pure luck or aesthetic sensibilities on the part of the GUI designers is not known.

The final stage of installation is the choice of whether or not the README.TXT should be viewed and whether or not the Configuration editor should be launched. Viewing the README.TXT cannot really be recommended via this method, since it is viewed in a fixed size box which requires scrolling both horizontally and vertically in order to read the entire file. The contents include such things as system requirements, which would seem to be mentioned far too late in the installation. By the time a user has installed a product, it is rather irritating for them to be told that it was a waste of time because their machine is too antiquated. In the first installed copy, other information was a list of known issues and their fixes or workarounds.

Documentation and Help

The documentation is supplied only in PDF format, and comes in three versions – Reference Guide, Administrator Guide and User Guides. The 77-page Reference Guide is the most complete, and the manual which was subjected to the most scrutiny. This is in some ways less helpful than it could be, since despite there being ample information supplied about the operation of the software, the impression given is one of pure factual detail rather than ‘this might be a good idea because …’.

Within the program itself, help is also available. This is brief but generally helpful and bears much resemblance to the Reference Guide. Some perplexing advice is given – most notably in the case of file exclusion. While describing this option it is noted that ‘Files on the Exclude list are not scanned. We do not recommend that you select this option, because the files on the exclude list should be scanned regularly.’ On a similar note, there is an ‘option’ for on-access scanning which is later described as ‘mandatory’ – not quite an ‘option’ as most users understand it.

Despite these foibles, the help function does give useful information in most cases, at least, in an easily comprehensible fashion. After a short period of using the software, the manual often became redundant from an operational point of view, since all the required information was to hand through the use of this help function. This is not entirely the case, however, since some parts of *NVC* lack a help function of any sort.

Features

When installed, *NVC* adds seven actions to its portion of the Start menu, namely Configuration Editor, Internet Update, Release Notes, Scan diskette, Scan hard disks, Task editor and Utilities. The great difference between this installation method and most others is that none of these items could be considered an on-demand scanner as is generally understood. None of them allow a direct interactive choice of area and subsequent scan in one fell swoop, which is a massive departure from tradition. How then, are scans performed which are not covered by these basic functions?

The primary answer to this question is the right-click method, where scanning is an option given. This is a fairly basic scanning interface, options being limited to an individual on/off setting for scanning each of sub-directories, archive files, memory and boot sectors. The alternative method of performing scans on selected areas is started within the Task Editor and is more useful where scans are likely to be repeated on more than one occasion.

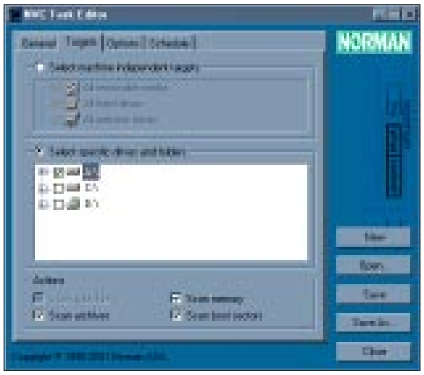
The Task Editor is used to construct and tweak on-demand tasks in a way that will be familiar to users of most anti-virus programs. This is controlled by the use of four tabs – General, used only for a description of the task, Targets, Options and Schedule. Targets, as would be expected, selects the areas to be scanned, offering options for independent machines such as ‘All fixed drives’ or for specific drives or folders.

A major limitation here is that individual files cannot be selected and that the scanning of subfolders cannot be deactivated. The latter can be worked around manually, but it is irritating nonetheless. Other options here are identical to those offered from a right-click scan, namely individual on/off setting for scanning each of archive files, memory and boot sectors.

The Options tab selects resource usage as Low or Normal and how the scanning window will be displayed when the task is activated. The latter allows for hidden or minimised until cleaning fails, minimised until an infection is detected or a normal window requiring a manual start. The Schedule tab is fairly standard in its choices. It is deactivated by default and offers a choice of once only, daily, weekly or monthly scans, with the first scan being indicated by a full date, including day of the week as well as the time for scanning. As an extra feature for more international companies, this date can be set to be considered as Universal Time Coordinates (UTC) which is a rough equivalence to Greenwich Mean Time (GMT.)

Sharp-eyed readers will be noting that so far there has been no way yet to activate these tests once set up – and this is the case from the Task Editor. Buttons are, however, provided in order that the task may be saved or loaded for editing as well as created. With a saved task what is done to activate it? It is here that *NVC* diverts from the mainstream. Saved tasks are simply clicked upon to activate them from wherever they have been saved – the installed Scan diskette and Scan hard disks actions on the Start menu are simply pre-configured instances of this.

This leaves the *NVC* configuration and Utilities options from the start menu as yet unexplained. The Utilities section is somewhat strangely named, being primarily concerned with information. Here, one may view a list of component version numbers, task files created, quarantined files and messages produced by the program. Configuration, by comparison, includes a whole



host of options, which can be fiddled with to the heart's content. Buttons allow for configuration of Installation settings, Common settings, On-demand scanner, On-access scanner, Message Routing and Message handling.

Starting with Installation settings – tabs here divide the choices between Install, Start, Internet, LAN/WAN and Authentication. Install is an interactive indication of those modules installed, and a selected language, which can be edited. Editing, however, often had no immediate discernable effect. Though a reboot is required for some changes to take place, this is not mentioned when the changes are saved within the Configuration Editor, nor is it mentioned within the help file. Start is a similar listing area, though only with the ability to choose whether the on-access scanner and/or scheduler are activated on startup.

Update mode and the remaining Internet, LAN/WAN and Authentication tabs are intrinsically linked. The updates can be selected from CD, Internet or LAN/WAN and the other areas then give configuration information for the selected process. Included are the ability to set a proxy server for Internet downloads, location on the network where updates are to be found and the account to use for collecting these. Where LAN/WAN updates are concerned not only the software can be set for updates but also the configuration files and available tasks may be collected and updated.

The next button, Common settings, sets Quarantines, Exclusions and what to scan objects for – New or Unknown viruses, Aggressive commercials and Security risks. The help file describes (without giving names) quite why each of these might be a problem and, usefully, reasons why such programs are likely to be present. There is a certain degree of redundancy between this button and the On-demand scanner button, since the latter allows for only a subset of selections available as Common settings. Since these are concerned with exclusions, it is clear that the production of help files is not the only confusing issue.

In a similar vein, the same selections can also be made independently in the area selected by the On-access scanner. The mandatory selection of scan files before they are used is complemented by a selection to scan new or changed files, which would seem redundant, but could possibly be of use under complicated imaginary circumstances. Selection of action on detection – deny access, remove or ask user – can also be activated here. The remaining two buttons are concerned with messaging and the binary log file. Here, the coverage of possible activities and the appropriate message action to take is alarmingly comprehensive, to the extent that the help file suggests a quick look at the manual, as it is too complex to be handled in a mere help file.

Updates

Updates are available either by download or media versions, or by the more convenient and speedy method of the inbuilt updater. This automatic method is proxy-aware and can also operate in conjunction with a dial-up account. Tests of the Internet update were performed on both dial-up and direct net connections and proved successful, taking less than a minute to perform in the latter case.

Somewhat more confusing was the situation with the 'update from files' option. A visit to *Norman's* Web site quickly revealed the virus definitions area, though the definitions provided were universally labelled as unsuitable for *NVC 5*. In fact, there was a categorical statement that *NVC 5* must be updated by use of the Internet updates, which was roundly contradicted by the developers.

Scanning

As ever in a standalone test, the scanning side of things is more a minor aside than the crux of the matter. The scanning engine has recent VB 100% awards on both *Windows ME* and more recently *Windows 2000* and thus full analysis was not performed here. Only one major matter can be tweaked within the engine as far as virus detection is concerned – the use or otherwise of heuristics.

Scanning on a machine used for browsing was the only area where oddites occurred – a virus description page stored as a temporary Internet file was detected as infected. This was not, however, a heuristic trigger and the option to disinfect was given. Choosing this disinfection option caused the machine to become unstable and at one point declare, falsely, that *Windows* must be reinstalled.

Conclusions

NVC has proved a bold leap for *Norman* – combining a major overhaul of the scanning engine with a new style of interface for scanning. The engine can be considered a success – the detection capability of *NVC* has increased considerably since its introduction, while the new methods of scanning are something of a mixed blessing. Although a little confusing initially, the controls are simple enough to master. There are certain aspects where simplicity has been taken too far – inability to 'Task scan' single files being a definite problem. It is early days, and some improvements have been seen in the interface. More are promised by the developers, so the product remains one to watch.

Technical Details

Product: *Norman Virus Control v5.0*.

Developer: *Norman ASA*, P.O.Box 43, N-1324 Lysaker, Norway. Tel; +47 6710 9700, fax; +47 6758 9940, email; norman@norman.no, WWW; http://www.norman.no/.

Price: NVC for workstations – 1 year, 1 user £40.00; NVC for Servers – 1 year, 1 user £312. Contact Norman for more details.

Test Environment: Three 750 MHz AMD Duron workstations with 128 MB RAM, 8 and 4 GB dual hard disks, CD-ROM, LS120 and 3.5-inch floppy, all running *Windows ME* or *Windows 98*. The workstations were rebuilt from image backups and the test-sets restored from CD after each test.

Virus Test-sets: Complete listings of the test-sets used are at http://www.virusbtn.com/Comparatives/NT/2001/02test_sets.html.

PRODUCT REVIEW 2

Symantec NAV Corporate Edition v7.51

Matt Ham

Norton AntiVirus, or *NAV* to its friends, has recently been enjoying a healthy string of VB 100% awards for its scanning prowess. Since its last review in these pages there have been numerous changes to the way in which *NAV* operates and a revisit is in order. What is perhaps surprising is that these enhancements seem to have been made primarily on the surface, as far as *Symantec* is concerned.

The aged will remember that *Symantec* enveloped *Intel's LANDesk* software a few years ago but the vanishing of that product is not yet complete. Much of the Registry information for *NAV* is stored under *Norton keys*, but a large portion is still under the name of *Intel* or *LANDesk*. When *NAV* and *LANDesk* became as one, the product certainly became better at management, network and messaging functions – the strong areas of *Intel's* offering. The surprise here is that the name is still used after several years hidden below the surface. This, however, is now pretty close to approaching ancient history – what remains of this article will concentrate on the present.

The Package

The product arrived in a sturdy yellow box in the trademark 'Symantec Yellow' with the odd feature that there appeared to be no place designed for it to be opened. Having bypassed this first level of security, the contents were bulky, of good quality and smelled nice. Unfortunately for such spring-inspired thoughts, the contents were mostly manuals which then required reading instead of being outside in the sun, together with a pair of CDs in a double jewel case.

The other contents were, in ascending order of size, 'What's in the box' and 'Read This First' cards and System Center and AntiVirus Implementation Guides. The 'What's in the box' card stated that with GroupWare or Gateway products a third CD would have been present, though the manuals would suffice for my level of installation. The most useful part of this card, however, is the listing of installation file locations and documentation sources (both printed and on CD) for the various applications which make up the *NAV* suite and its supporting software.

The 'Read This First' card is also a useful specimen of its type, containing details of the various components of *NAV* and how these interact. It continues with the various management options, giving a good basis for deciding which of the many methods of installation and administration are to be used. Finally, installation and post-installation tasks are considered. The most useful part of this card is that it gives details not only of the 'how' information but also the important 'why' information that can save a great deal of time in any organization other than the miniscule.

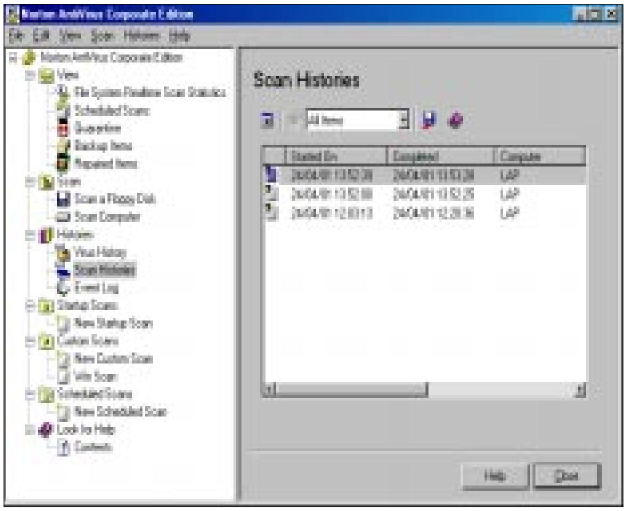
The smaller of the two manuals covers Symantec System Center – a generalised central administration application which has snap-ins for various products. Although it is not advertised as such directly, the manual provided is definitely *NAV*-specific – which avoids having to wade through extraneous information. Administration tools will likely be studied separately in future issues of *Virus Bulletin*, so the exact contents of the manual is a study for the future rather than present.

The last, and by far the largest chunk of the literature in the box is the Virus Protection for Desktops and File Servers manual. This weighs in at a hefty 360 pages and is at that size clearly quite exhaustive in its coverage of features. The only real complaint that can be made about this is that small details can be tricky to find within the manual's step-by-step description of how to perform some tasks, of which more later. It was also apparent that the index was not as complete as it could be – of the two macro virus descriptions provided in the manual only one was indexed.

Installation and Updates

Installation of the *Windows 98* version tested was directly from the supplied CD. Autorun triggers a menu from which the appropriate product is chosen and from this point onwards there is very little in the way of user input. Problems were encountered when attempting to install version 7.51 over version 7.5 however, when the process hung during installation. After this point, the Registry entries were sufficiently confused that neither installation nor uninstallation could proceed until the Registry had been cleared of *NAV* program references.

Updating of *NAV* is primarily performed from the Internet by means of scheduled updates, LiveUpdate, Virus Definition Transport, or the Intelligent Updater. The Virus Definition Transport method is of use in managed network installations and was not investigated further. LiveUpdate is an on-demand method which aims to keep definition file packages small by calculating what is and what is not required to be updated. Both this and the scheduled updates, defaulting to once a week, can be set to load from either Internet or LAN. There are also some interesting features for preventing network congestion when updates are triggered – including a randomisation method for which day the updates are triggered. Intelligent updaters are packaged as files for download and execution on a local machine and come either as monolithic packages or handy floppy-sized chunks more useful for updates of standalone or portable machines.



Started On	Completed	Computer
24/04/01 13:52:38	24/04/01 13:53:28	LAP
24/04/01 13:52:08	24/04/01 13:52:25	LAP
24/04/01 12:03:13	24/04/01 12:28:36	LAP

Program Configuration

Configuration within *NAV* is performed exclusively within one *Explorer*-style window – the left half containing a tree of configuration and action options. When one of these options has been selected, the right half fills with corresponding contents. A menu bar above this area by and large duplicates the available options in the windows, though adding options in the File and Edit menus.

The root of the left-hand window is the *NAV Corporate Edition* area, from which can be accessed the View, Scan, Histories, Startup Scans, Custom Scans, Scheduled Scans and Help areas. The Edit menu allows additions to be made to the contents of the three Scan areas or indeed to manipulate and edit those Scans already existing. The File menu, on the other hand, sets how long History files are to be stored, allows updates to be scheduled and gives an option for the on-demand LiveUpdate to be triggered.

The *NAV Corporate Edition* area also allows triggering of the LiveUpdate feature, along with showing program versions and selecting whether Norton AntiVirus Services are to be loaded. It was notable that what this meant is not apparent from the page in question, and the on-line help and manuals proved full of other information which made searching for details of this a tedious task. The subject of Services was found in the index, but references there did not seem to be pointing to the correct information.

On to the areas deeper within the tree system – View is the first of these. This is further divided into File System Realtime Scan Statistics, Scheduled Scans, Quarantine, Backup Items and Repaired Items. These can be covered in quite a speedy fashion since they are all purely informational. It was good to see, however, that at this level, each area has a dedicated help button which gives much more speedy and useful information on the page in question.

Following this are the two preset scans under the Scan area – covering floppies and the computer in general, both allowing interactive selection of the exact areas to be scanned. Oddly enough, performing a floppy scan on a laptop with no floppy attached did not give any errors and the scan completed happily.

Histories, like View, is a purely informational area, providing details on past scans in a 'filterable' manner. Available information covers Virus History, with there also being the option to perform the usual set of actions upon selected infected objects.

Also present are buttons for View Item Properties and Take Actions. Scan Histories simply lists the scans which have been performed, while the Event Log lists, as might be expected, all events, both user and automatically initiated. Information within these History areas may be exported to external applications, though it is somewhat disappointing that only .CSV and .MDB export is supported here.

Next in the line-up are Startup, Custom and Scheduled Scans, consisting of divisions for each of the scans currently defined. These are configured, added and edited from either the Edit menu or the New Custom Scan, area so this seems an appropriate place to describe the process further. Initiating the creation of a new scan starts by inputting a name and description for the scan, with the length of space available being enough to include a good sized essay.

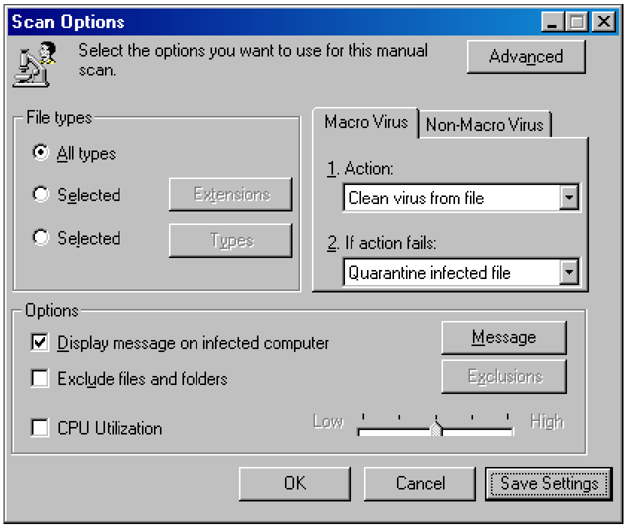
A tree representation of the machine is provided for selecting the areas to be scanned. Selecting an area more than once toggles between no scan for the folder selected, recursive and non recursive scanning. If the default settings for the scanner are acceptable, then all that is needed to complete the process is to press the Save button.

If the default options are not perfect, or inspection is required of what these might be, the Scan Options button allows viewing and alteration. Default settings are to scan all files, disinfecting all viruses found or quarantining if this is not successful. These actions can be changed to delete file or log only if required.

Rather than 'all files' the objects to be scanned can also be selected by extension or by the more limited 'Types' which divides all files into Document, Program or other files – the last is not selectable for scanning. Exclusions can also be set, on a file or folder level, the level of CPU utilisation and messages either toggled on, off or customised.

Advanced options are also available through a further button on this dialog. Here the scanning of files inside archives can be deactivated. It is also possible to set the level of recursion which will be scanned within, the default being three levels deep and the maximum ten. Also alterable here is whether any infected files are backed up before disinfection is attempted. Notable by its absence is any way to disable heuristics.

Only the Scheduled scan is different here – allowing for the setting of times for scanning as might be expected. These offer daily, weekly or monthly options and, usefully for machines which are not always on, an option to try the scan again if it is not performed on the first try.



Select the options you want to use for this manual scan.

File types

● All types

○ Selected

○ Selected

1. Action: Clean virus from file

2. If action fails: Quarantine infected file

Options

☑ Display message on infected computer

☐ Exclude files and folders

☐ CPU Utilization

Scanning

The most noticeable feature of the scanning process, at first glance, is that when a scan is triggered the main interface vanishes to be replaced by a very sparse window. This has controls to stop, pause and restart the scan, trigger the help and revert to the main GUI. In addition, the windowed area below the controls shows a listing of details of infected objects. If objects are detected as infected, two more controls become available – to View Item Properties and Take Actions. This will be recognised as being the same as was present in the Virus History area.

With no real scanning options that could be expected to change performance, readers are referred to the previous two Comparative Reviews for *ME* and *Windows 2000* for detection results (see February and April 2001 *VB*). In those tests *NAV* performed well enough to gain two VB 100% awards, taking it up to an unbroken run of eight such performances. In the April *Windows 2000* testing, the misses were confined to the BAT/911 and a sprinkling of the newer zoo macro viruses in the test-sets.

It was also noted in those tests, as in many before them, that viruses potentially causing file-system damage if deleted – Byway and Dir II – are logged as infected but not deleted even if this option is selected. Such attention to detail is admirable and would be much appreciated by a user infected by either of these viruses. The user is not told why this insubordination is occurring: this the only fly in the ointment. Scanning problems seen in these two Comparatives with large infected collections were not seen in scans with a majority of uninfected files.

The lack of scanning control is, at first glance, a surprising feature – in many products the degree to which scanning methods may be tweaked is huge – with levels of heuristics, completeness of scanning within the files, and CPU usage all being individually alterable. The tweak factor is limited to CPU usage on the on-demand scanner. The on-access scanner has not been mentioned as yet – and here there is even less control – the status being either ‘on’ or off.

This shows a distinct trend towards the corporate market – where homogeneity (and ease of knowing that all is homogenous) is of great importance. The average user cannot be expected to understand what the various levels of heuristics or behaviour blocking mean or do, so all is well and good if thes features are absent.

For the inveterate tinkerers – who seem to make up the majority of our readers – this will come as something of a disappointment, but to a standard user the world of viruses may seem a somewhat less complicated place.

It could also be seen as a reversal of the ‘feature bloat’ which has become associated with *Symantec* in the last year or so. If complexity is being cut down upon, and only useful features remain, added stability might be an expected and welcome effect of the process.

Conclusions

Symantec’s Norton AntiVirus is, as just discussed, a corporate-oriented product and a review of this type can only really give a feel for its behaviour in a massive organization, but it is clear that variety of users is planned for. The updates features, even while working on this small test scale, can be seen to have been designed with much larger ones in mind and the variety of tools available only serve to back this up.

This also, to a certain extent, accounts for the bulk and comprehensiveness of the manual documentation for administration. From the point of view of virus detection, as previously mentioned, this particular product has recently been the worthy recipient of a string of VB 100% awards.

The only real problems I encountered during testing of the Corporate Edition were the lack of a brief and useful help for some of the less well explained features, and the glitches thrown up when logging thousands of virus detections. Since the former are less important when dealt with by a full time administrator and the latter rather a ‘feature’ of detection testing rather than a real world occurrence, *Symantec* can be happy (but not complacent) in the near future.

Technical Details

Product: *Symantec Norton AntiVirus Corporate Edition 7.51*

Developer: *Symantec Corporation World Headquarters,* 20330 Stevens Creek Blvd. Cupertino, CA 95014, USA;
Tel +1 408 5178000; fax +1 408 253 3968;
WWW http://www.symantec.com/.

Price: Contact *Symantec* for details about prices and packages to suit.

Test Environment: Three 750 MHz AMD Duron workstations with 128 MB RAM, 8 and 4 GB dual hard disks, CD-ROM, LS120 and 3.5-inch floppy, all running *Windows 98*. Pentium laptop with 48 MB RAM, 1.4 GG hard disk, CD-ROM and 3.5-inch floppy running *Windows 98*.

Virus Test-sets: Complete listings of the test-sets used are at http://www.virusbtn.com/Comparatives/NT/2001/02test_sets.html.

SUBSCRIPTION RATES

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

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

Editorial enquiries, subscription enquiries, orders and payments:

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

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

US subscriptions only:

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

Tel (781) 2139066, Fax (781) 2139067

No responsibility is assumed by the Publisher for any injury and/or damage to persons or property as a matter of products liability, negligence or otherwise, or from any use or operation of any methods, products, instructions or ideas contained in the material herein.

END NOTES AND NEWS

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

If your job is based and you work in an IT security-related position, you might be eligible for **a free subscription to the general IT security publication Information Security.** For more information, visit http://www.infosecuritymag.com/.

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

On Wednesday 18 April at a secret location in London, the UK's first dedicated computer crime unit opened for business. The government funded National Hi-Tech Crime Unit (NHTCU) has been established to handle all aspects of computer crime including hacking and virus distribution.

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

***Sophos* is to host a two-day Anti-Virus Workshop on 22 and 23 May 2001** at its training suite in Abingdon, Oxfordshire, UK. For details about the different courses and training days available, or to reserve your place, contact Daniel Trotman; Tel +44 1235 559933, or email courses@sophos.com.

The *Internet World Event Network* has published autumn 2001 dates for exhibitions in Glasgow (Scotland), Dublin (Ireland) and Manchester (UK). For more information about exhibition opportunities and the full event line-up, see http://www.internetworld.co.uk/.

The full VB2001 conference programme is now available on the *VB* Web site, along with details of how to book your place in Prague in September 2001. See http://www.virusbtn.com for details.

A recent survey by UK-based email scanning company *MessageLabs* (of a sample of 50 million emails sent between 1 January and 28 February 2001) compared the rates of a predicted virus increase to the increase in the use of email in the workplace. The IT industry saw a predicted virus increase of 143% and an increase in the use of email of 252%, while the Government sector's figures saw a 222% rise in the prevalence of viruses as opposed to a mere 62% increase in the use of email.

GroupWare security specialists *Sybari Software* have announced a partnership with *SatelliteSafe*, developers of what is being marketed as the first satellite delivery system for anti-virus protection. Users of *Sybari's Antigen* product are set to receive instantaneous and automatically distributed anti-virus updates via the satellite system. For more details, see http://www.sybari.com/.

***NAI Labs*, a division of *PGP Security*, announces a $1.2 million, 2-year contract with the *National Security Agency* (*NSA*)** and its partners to develop the *NSA's Security-Enhanced Linux* (*SELinux*) prototype. For more information, contact Caroline Kuipers in the UK; Tel +44 1753 217500 or see http://www.nai.com/.

***F-Secure Online Solutions* (*F-SOS*) has launched its Managed Personal Anti-Virus Solution in North America.** The company has entered an agreement with Texas-based ISP *Internet Unlimited* and the Managed Security Services Provider *KnowledgeSentry* to provide the SOHO market with an automated, frequent update mechanism for their anti-virus protection. Email kimmo.alkio@f-sos.net or contact; Tel +358 925166363 for more details.

The UK *National Criminal Intelligence Service* has released details of Europe's first technology focused law enforcement event – **the International Law Enforcement Expo 2001 is to be held from 5–7 November 2001 at ExCel in London's Docklands**. Anti-virus companies will be among the exhibitors. For more details, see the conference Web site http://ile-expo.com/.

***Computer Weekly* reports a story about a disgruntled computer store manager in Devon, UK who sent a virus to a rival company.** Employees at *Complete Computers* became suspicious and did not click on the email attachment they had been sent. The perpetrator was sentenced to 175 hours of community service.