ISSN 0956-9979 OCTOBER 2002

VIRUS BULLETIN

THE INTERNATIONAL PUBLICATION ON COMPUTER VIRUS PREVENTION, RECOGNITION AND REMOVAL

IN THIS ISSUE:

• **Laying down the law:** *VB*'s criteria for its comparative reviews have remained largely unchanged since the inception of the VB 100% award in 1998. We thought it time to iron out the creases and reaffirm the aims, procedures, rules and regulations. Check them out on p.3.

• **The laws of physics:** Although firmly devoted to the anti-virus industry these days, much of *Eset* VP Anton Zajac's early career was devoted to research in theoretical physics. Anton describes key events in his lifetime in the worlds of theoretical physics and anti-virus, starting on p.15.

• **Murphy's Law:** Peter Morley predicts the end of the production of new *Windows* viruses and Trojans … but that doesn't mean we can all go home and put our feet up. Assess the predictions on p.17.

CONTENTS

COMMENT

“ *Are anti-virus companies really doing justice to their customers by still providing only* part *of the anti-virus ‘solution’?* ”

Best Practice or Wishful Thinking?

We all know the rules. I spend a large part of my working day stressing the importance of safe computing when it comes to fighting viruses. Whether you call them ‘best practice’, ‘safe hex’ or by another name, the AV industry tells its customers that the use of these simple rules, along with appropriate anti-virus software, will keep them virus free.

Yet all around the world, customers of all AV companies continue to infect their networks with viruses. It’s not that the rules are flawed – after all, the AV community manages to stay largely virus free despite dealing with thousands of infected files every day. So we know that the rules work when applied properly. Or rather that they would work, if everyone followed them. At a fairly early stage of the initial Magistr outbreak, I dealt with a customer whose network had become infected after a member of staff double-clicked the attachment deliberately, ‘to see what would happen’. And there’s the problem.

In the real world, it is unrealistic to expect administrators to be able to control every node of a 50,000-machine network, including laptops. Every day, end users flaunt safe computing guidelines by clicking on unsolicited email attachments, downloading files from websites and running unsecured systems. Clearly, no matter how loudly administrators, anti-virus vendors and other security organisations shout, there are not enough end users listening.

As the guardians of their customers’ corporate castles, should AV vendors be looking for other ways to protect their customers – methods that take the onus away from the user entirely? The anti-virus industry is now about 15 years old. Okay, viruses and virus types have come and gone, but the general principles have remained pretty much the same since the days of Brain and Jerusalem. Are anti-virus companies really doing justice to their customers by still providing only *part* of the anti-virus ‘solution’? After all, if a firewall vendor offered a product that gave excellent security as long as users didn’t visit certain websites, customers would, quite rightly, give the software a wide berth.

There are those who would refer back to the age-old adage ‘you can lead a horse to water, but you can’t make it drink’, and might say that users who fail to protect themselves deserve what they get. To a certain extent that opinion might be justified, but can you really apply the same argument to a multi-national company that has been infected by a fast-spreading network-aware virus as a result of the actions of a single user who sees IT security as someone else’s problem?

Perhaps we should be calling on the manufacturers of operating systems to secure their products. Undoubtedly the current worldwide OS monoculture has contributed to the present situation where viruses can infect machines worldwide in a previously undreamed of space of time. But is it reasonable to expect any manufacturer to ensure that many millions of lines of code are free from the sort of error that leads to security vulnerabilities when the consumer appears more interested in features over security? Software manufacturers are only delivering what the consumer wants.

Perhaps open source is the answer, but one of the reasons why there are so few viruses for open-source systems is their relatively small and technically-minded user base. Once these systems become more widely used by less IT-literate users, security holes will go unpatched and the number of viruses that exploit them will increase. So we’re back to square one – anti-virus best practice needs to be employed to keep users of newer open-source operating systems virus free.

While the usefulness of anti-virus best practice, where it can be practically implemented, is not at question, perhaps the anti-virus community as a whole should consider that virus infections are an inevitable part of running a network that is not completely disconnected from the outside world. Until now, most AV vendors have focused on detection and prevention. Perhaps the time has come for anti-virus software developers to accept that virus infection cannot always be prevented, and attempt to address this within the products and services they deliver.

Phil Wood, Sophos, UK

NEWS

The Rules of the Game …

Although little has changed since the inception of the VB 100% awards in January 1998, *Virus Bulletin* would like to clarify its VB 100% award scheme. The VB 100% logo is awarded to anti-virus products that detect all In the Wild viruses during both on-demand and on-access scanning in *VB*'s comparative tests. Furthermore, in order to qualify for a VB 100% award, the product must produce no false positives.

On-access scanners are tested on 'close' as well as 'open', and all products are tested in default mode, meaning that the detection settings are in their 'out-of-the-box' state throughout the testing process. Each product may be tested up to three times on two different test machines. Should any product fail to work after three attempts the testing process will be aborted for that product. Where there is a problem with a product during the testing process, *VB* makes every effort to contact the developers to alert them to this fact. *VB* makes all reasonable attempts to answer queries and re-test products where a problem has been encountered during the test process. However, these cannot always be undertaken immediately, nor guaranteed.

Virus Bulletin's aim is to offer subscribers the best impartial advice about anti-virus security and the products on offer. For that reason, VB 100% awards continue to be platform-specific and clearly dated. Promotional material featuring VB 100% awards without dates should be reported to *Virus Bulletin*. The *VB* website provides reference tables listing the outcome of comparative tests by product and by platform, as well as a summary of the most recent comparative tests (http://www.virusbtn.com/vb100/). The full test results continue to be published in the print version of *VB*.

Please contact us with any problems and queries relating to VB 100% awards; email editor@virusbtn.com [*The next comparative review is scheduled for the November issue of* VB *and will be on* Windows 2000 Server.] ▮

Defenders of the Defenses

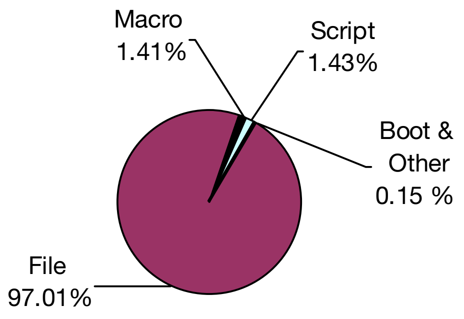
The US Government is urging consumers and companies to tighten up their computer security as part of its widely anticipated cybersecurity plan. Early drafts of the plan recommend that people keep anti-virus software up to date and call on Internet service providers to do more to protect their customers from viruses and other web attacks. There is also a call for security researchers and companies to do a better job of circulating information about vulnerabilities and ways to close them. Meanwhile, *Network Associates* and *Trend Micro* have been awarded five-year contracts to deploy anti-virus and security solutions to users in the US Department of Defense ▮

Prevalence Table – August 2002

Virus	Type	Incidents	Reports
Win32/Klez	File	4644	77.22%
Win32/Yaha	File	289	4.81%
Win32/Magistr	File	198	3.29%
Win32/BadTrans	File	147	2.44%
Win32/Frethem	File	135	2.24%
Win32/SirCam	File	104	1.73%
Win32/Nimda	File	66	1.10%
Win32/Hybris	File	61	1.01%
Laroux	Macro	49	0.81%
Redlof	Script	48	0.80%
Win95/CIH	File	32	0.53%
Win32/Elkern	File	30	0.50%
Win32/Higuy	File	22	0.37%
Win32/MTX	File	17	0.28%
Win95/Tecata	File	17	0.28%
Haptime	Script	16	0.27%
Divi	Macro	12	0.20%
Win32/Datom	File	9	0.15%
LoveLetter	Script	8	0.13%
Win32/Duni	File	7	0.12%
Win32/Ska	File	7	0.12%
Win32/Surnova	File	7	0.12%
Others [1]		89	1.48%
Total		6014	100%

[1] The Prevalence Table includes a total of 89 reports across 52 further viruses. Readers are reminded that a complete listing is posted at http://www.virusbtn.com/Prevalence/.

Distribution of virus types in reports



TECHNICAL FEATURE

Dealing with Metamorphism

Myles Jordan
Computer Associates, Australia

When the virus writer known as z0mbie released W95/Zmist.A in early 2001, much of the attention it drew from the anti-virus community was directed at its remarkable ability to intersperse its own code with that of its infection target.

However, W95/Zmist.A also embodied the continuation of z0mbie's work on viral evolution towards metamorphism – a form of camouflage being developed by virus writers that is so potent and radically different from common encryption that anti-virus scanners will soon need powerful new tools to confront this threat. This article will discuss one method that anti-virus scanners could employ to deal with metamorphism.

The Technology Arms Race

Computer virus technology has been evolving continuously since the development of the very first viruses. Consequently, anti-virus technology has been forced to undergo a parallel evolution in order to be able to detect new viruses equipped with the latest stealth and protection mechanisms.

Encryption and Polymorphism

When encryption was developed for viruses in order to hide their bodies, anti-virus scanners began to use algorithmic methods instead of simple template matching to detect the viruses. As encryption evolved into polymorphism, anti-virus scanners developed emulation in order to detect viruses that would be excessively difficult to detect using algorithmic methods.

It is evident that the majority of viral technology development has been aimed at disguising the viral code via fixed, oligomorphic or polymorphic encryption. Each of these techniques involves encrypting the virus body, and supplying a wrapper of code to decrypt when necessary.

Of these techniques, polymorphism is by far the most powerful with, theoretically, an almost unlimited number of different decryption code wrappers that could be created. However, it is the fundamental property of polymorphism as merely a wrapper that limits it as a form of camouflage; no matter how good the polymorphism, after sufficient emulation, the body of the viral code is laid bare and can be recognized easily, even by template matching. Despite its seemingly infinite complexity, polymorphism is, ultimately, a finite problem.

However, there has been a concurrent stream of viral development that does not necessarily involve encryption at all. Rather, this branch of camouflage techniques involves modifying the body of the virus itself instead of modifying some form of decryption wrapper. This technique is commonly known as metamorphism.

The Evolutionary Process

The first attempt at metamorphism in a Portable Executable (PE) virus was W32/Apparition. This virus carried around its source code and recompiled itself with some random junk code inserted whenever it discovered an appropriate compiler.

Following this, a more direct, though relatively simple, attempt at metamorphism was made by W95/Regswap, which swapped the registers used to perform particular tasks.

This form of camouflage was carried a step further by the W32/Evol virus, which swaps certain instructions for different instructions, but which performs the same function. In addition it is, and was, able to insert junk code between the essential instructions.

In early to mid-2000, two so-called 'permutating' viruses were released: W95/Ghost and W95/Smash. Both of these viruses have the ability to split their body into blocks, and then change the order in which these blocks appear in the body.

This technique of permutation was improved upon by W95/Zperm, which is able to reorganize its code completely and insert jump instructions everywhere necessary to maintain the correct flow of control.

As mentioned above, W95/Zmist was yet another developmental step, and contains the ability to reorganize its code, insert junk instructions, and perform instruction substitutions. This work has since been continued by the author of W32/Metaphor (aka W32/Etap), which demonstrates even more advanced metamorphism.

An Observation

W32/Metaphor's metamorphism works by disassembling its own code into a custom pseudo-code, which is a meta-language for describing the actions of the code of the virus without any reference to the actual code.

With this layer of abstraction, the virus dissociates function from implementation, allowing the virus to generate new copies of itself completely from scratch. This produces instances of the virus that appear very dissimilar, yet function identically – which is, of course, the goal of metamorphic camouflage.

The code below was generated by W32/Metaphor, and is used to find the address of the Kernel32.dll:

```
mov       dword_1, 0h
mov       edx, dword_1
mov       dword_2, edx
mov       ebp, dword_2
mov       edi, 32336C65h
lea       eax, [edi]
mov       esi, 0A624548h
or        esi, 4670214Bh
lea       edi, [eax]
mov       dword_4, edi
mov       edx, ebp
mov       dword_5, edx
mov       dword_3, esi
mov       edx, offset dword_3
push      edx
mov       dword_6, offset GetModuleHandleA
push      dword_6
pop       dword_7
mov       edx, dword_7
call      dword ptr ds:0[edx]
```

All of these lines of code are testament to the power of W32/Metaphor's metamorphic engine, as they could be replaced by the following five lines:

```
mov       dword_3, 6E72654Bh
mov       dword_4, 32336C65h
mov       dword_5, 0h
push      offset dword_3
call      ds:[GetModuleHandleA]
```

This example illustrates an important point: no matter what the form of the actual code, there are certain 'higher' actions that are always performed.

A higher action is a phrase used to describe the purpose of a related group of instructions, and can, for example, be anything from locating the Interrupt Descriptor Table (a single instruction, SIDT), to hooking an API (usually a small series of instructions), or even decryption (variable, but often a large number of instructions).

In the example shown above, two higher actions are performed:

1. Construct 'Kernel32' string
2. Call GetModuleHandle API

Depending on how much detail is required, it could be a single, even higher, action:

1. Locate the Kernel32 dll in memory

The dissociation of the function of the virus from its actual code is used commonly by anti-virus scanning software for heuristic analysis, but it becomes particularly useful when dealing with metamorphics.

In particular, an anti-virus scanner's heuristics need to be capable of analysing the effects of multiple individual instructions and coalescing these effects into higher actions, as demonstrated by the example above.

The scanner can then analyse these actions heuristically, completely disregarding their implementation. Effectively, this makes the literal instructions used irrelevant, and thus bypasses a significant portion of the power of metamorphic viruses: the junk insertion and instruction substitution techniques.

This method of heuristic analysis can be equally well applied to all known types of metamorphic virus, even those that recompile themselves, such as W32/Apparition. This is because the method of metamorphism becomes irrelevant once the core functionality of the virus (which never changes between infections) can be examined.

Another Observation

The other metamorphic technique used currently by viruses is code reorganisation, or permutation.

As mentioned previously, the first PE virus to use unrestrained code reorganisation was W95/Zperm. This has the ability to move variable pieces of its code to anywhere within its body, and then insert jumps, invert conditional branches, or simply change the relative offsets in existing jumps.

Combining these features with the ability to insert limited simple junk code, it is evident that this virus could never be detected reliably by template; nor would an emulator seem to be much use – there is nothing to decrypt. It appears as though a specialised algorithmic detection is required. Or is it?

As noted in the example of W32/Metaphor, it is possible to ignore the instructions themselves, and analyse only the higher actions of the code. This idea also extends to the apparent ordering of the code, which does not really matter either; the same higher actions will be performed in the same order no matter how much the code jumps around.

So, by employing an emulator with heuristics capable of discerning higher actions, it is possible to circumvent the metamorphic technique of code reorganisation.

A Powerful Tool

Originally, emulators were designed to allow anti-virus scanners to decrypt simple, complex, or polymorphic encryption generically. This allowed the scanner access to the decrypted code and thus relieved the burden of having to run many specialised algorithmic detections.

However, it has been demonstrated that all known metamorphic techniques can be thwarted by the use of an emulator, coupled with heuristics capable of coalescing the effects of multiple instructions into higher actions.

But what exactly should be done with the higher actions once they have been discerned? A common solution is simply to collect them and analyse them later, looking for particular sets of actions that would be indicative of some virus or virus family.

This type of analysis has been around for a long time, but it is notoriously prone to both inaccurate virus family recognition and outright false alarms.

The problem with inaccurate family recognition arises because many viruses share similar functionality (e.g. infecting files), and the problem with false alarms arises as many legitimate programs use similar functionality (e.g. searching for files, then writing to them).

Fortunately, the simplistic technique described above is not the only way to analyse higher actions. In fact, that technique discards important, implicit information regarding the higher actions – namely the chronological order in which they occurred.

This ordering of information can be crucial in discriminating between a sequence of actions which is viral, and a sequence of actions which is harmless. For example, consider the following sequence of higher actions:

1. memory map file
2. modify memory area file is mapped to
3. close memory map

Depending on what modifications are actually made to the file, this sequence could be considered viral. However, consider the same actions, in a different order:

1. memory map file
2. close memory map
3. modify memory area file *was* mapped to

This second sequence is definitely not indicative of viral activities, and thus a potential false alarm situation is avoided, solely due to the inclusion of the chronological ordering of data in the heuristic analysis.

Conclusion

This heuristic analysis of chronologically ordered higher actions has also proved useful in decreasing the susceptibility of heuristic analysis to false alarms, and it continues to demonstrate its effectiveness against all known forms of metamorphism in computer viruses.

It is interesting that an answer to the seemingly infinite complexity of metamorphism is to disregard the smoke and mirrors, and simply examine the meaning.

FEATURE 1

Cat Herding: Malware Management across Autonomous and Semi-autonomous Sites

David Harley, Independent virus management researcher & author

What are the problems associated with malware management where direct control from the centre is constrained by organizational perimeters, and where only limited assumptions can be made about product choice and environment?

Having fairly recently exchanged a user-base of 2,000 or so for one of over a million, this is a topic close to my heart (although I shan't be making direct reference to the organization that employs me).

Worlds Apart

Anti-virus vendors live in a slightly different world from that of their customers. The vendor cannot possibly envisage every permutation of hardware and software that a client may be using, nor the infinite variety of administrative architectures that their IT infrastructure may have to work within.

Reasonably enough therefore, vendors tend to assume that organizations implement centrally-managed anti-malware solutions – and offer administration tools accordingly. This approach may work very well, even in large organizations, where growth is planned and security is properly integrated into the infrastructure from day zero.

In the real world, however, malware management does not often work out so neatly. Growth is not planned – at least not in terms of the IT architecture.

Organizations move to larger premises and take on more staff to meet evolving business needs. They acquire or merge with other organizations, and it isn't always possible or desirable to attempt immediate (or even mid-term) rebuilding of large chunks of infrastructure. It may take months or years to address issues that are too fine-grained to arouse interest in pre-merger boardroom discussions, but which may be vital to the health and productivity of the enterprise.

Expectations and Assumptions

Just as end-users expect the IT professional on the helpdesk to offer immediate solutions to any problem relating to hardware – be it laptop, VAX, router, or fax machine – and

every application and operating environment known to mankind, there is a common assumption that all IT issues are the responsibility of a single IT unit.

However, even in relatively small organizations, it is common to find that various parts of the IT infrastructure are the responsibility of a number of different units, which may or may not intercommunicate and interoperate, and which may or may not come under the same administrative banner. Often, several units perform the same function without reference to each other's work in the same area.

Thus, to take an example that may be familiar to many corporate readers of this publication (especially those from the anarchic cultures of academia, I suspect), advice and alerts on virus and anti-virus issues may be distributed by the security team, the help desk, the intranet publishing team and the estates department.

Sometimes such material is integrated and complementary: often, it represents political divisions and inadequate information resources and verification. This can result in such anomalies as one unit uncritically forwarding or publishing a virus hoax, while another publishes a warning against the very same hoax.

Double Trouble

Mergers between disparate organizations engender similar, but probably more pronounced, problems. Doubling the support staff may be seen as an opportunity to reduce costs by rationalizing software and hardware usage and downsizing the support establishment, but even if the planning is right, the amount of time, effort and expense involved in re-licensing, re-training support staff and end-users, and the standardization of network infrastructures, is likely to be considerable.

Given that it is characteristic of many organizations to overestimate the efficacy and ease of maintenance of malware management measures, it is not surprising if there is a reluctance to interfere with licensing and maintenance measures that are already in place.

Also, even if an organization used an anti-virus vendor or third-party dealer, an internal project team or external consultancy to plan implementation and to integrate systems across the whole organization, there is no guarantee that the same care will be taken to review and re-implement malware-related security measures as part of the merger process.

Public sector umbrella organizations, such as healthcare and community service providers, may yoke together many largely autonomous client organizations. In some ways, this is the 'worst of all worlds'. Not only do the previous problems of interoperability apply, but solutions involving standardization and central administration may be significantly hampered by conflicting political and operational requirements.

Autonomous IT units supporting their own choice of products in highly disparate environments may mistrust and resist (perhaps with good reason) attempts to regulate from the centre, fearing the imposition of layers of bureaucracy and enforcement of implementations inferior to those presently in place.

Potholing on the Superhighway

So what are the implications of these problems for threat assessment and coping with potholes on the Misinformation Superhighway? Most well-secured organizations recognize the need for good communication, education, discipline, standards, and deployment of technology.

By 'communication', I mean not only liaison and co-operation between disparate units and between the IT department and every user, but also pre-emptive and reactive information sharing in the specific area of security.

This is a sword with two edges: what I have referred to previously as 'threat assessment', and incident tracking – though, in reality, the borders between the two are pretty blurred.

In the security field, if not in real life, discipline and education are closely related. Whether the prevailing organizational culture is draconian or laissez-faire, good security practice relies on clear and effective communication of protocols and practice.

Guidelines, policies (including documents such as policies of acceptable use of inter-networking and published standards), agreed security standards and audit methodologies, are expressions of the corporate will to maintain a standard of discipline.

However, the reactive application of sanctions against offenders is a poor substitute for ensuring that training and up-to-date informational and policy resources are widely available – and that the user community takes advantage of them when appropriate.

Such measures can have a beneficial impact on primarily non-technological issues such as management of hoaxes, 419 scams, and so on.

Nuisance Mailstorms

Recently, I've been very aware of numerous instances where a mailstorm of forwarded hoax alerts has been followed by a mailstorm of counter-alerts warning that the original 'alert' is a *hoax*, followed by messages of further debate. Each message is addressed to each of the many recipients of the original message, and quotes in full every message in the thread that preceded it, each of which quotes a full list of recipients!

This sort of nuisance is simple (in principle, if not in practice) to counter by restricting the number of people authorized to forward alerts, offering instead a central assessment and alerting service.

Traditionally, anti-virus technology proposes technological solutions for social problems. After all, it's infinitely easier to write a program that detects malicious code than to implement educational and penal programmes that will reduce the problem by dissuading malefactors from writing, or at least disseminating, malware.

Low Expectations

It is easier, in principle, to save the user community from itself by attempting to detect and block malware transparently, than it is to train end-users to take even minimal precautions and develop sufficient scepticism to resist psychological manipulation.

Security tends to work best when it expects the least of the end-user, and it is probably no longer the case that functionality and business needs will always win out over security. As a consequence, the use of generic filtering to block potentially dangerous file types is becoming increasingly common amongst corporate organizations.

The file types that are blocked include not only those that rarely have legitimate reason to be sent by email (.LNK, .BAT or .PIF files, for example), and types that are considered sufficiently dangerous to be worth the occasional inconvenience of blocking legitimate examples (.EXE or .DLL, for instance), but also types which cannot be blocked without risking significant negative impact on business processes (.DOC, .MDB and so on), despite having the potential to contain dangerous code.

Technological Deployment

By 'technological deployment', I refer not only to the safe configuration of security software and vulnerable applications, but also to patch management, anti-virus update management, and the coordination of incident management initiatives (which can, and should, include incident tracking, logging and reporting back to top management and the user population, incorporated into the general risk management process).

These problems are compounded by the amount of misinformation available from supposedly authoritative sources: computing journalists who derive their expertise in virus management from half-understood vendor press releases; consultants, systems integrators, inadequately briefed helpdesk and sales personnel; and security specialists who believe that knowledge of cryptography or firewall technology automatically makes them virus experts.

Thus we see software set to out-of-the-box defaults that are inadequate for dealing with common threats, or too draconian not to impact adversely on business processes – and sometimes both. Even worse, we see proposed virus management guidelines that reflect a similarly ill-informed balance of under- and over-engineering.

We should perhaps consider the relationships between malware management, network management, and other areas of security management. These areas have a strong logical kinship. File servers, mail servers, desktop machines, web servers etc. may each be managed by a very different group, but each of them requires anti-malware measures, network security and so on.

However, communication between groups with similar responsibilities can be hampered by empire building and by adherence to imaginary and unrealistic borders between domains, even in small and relatively coherent organizations, let alone corporate bodies with a WAN and multiple intranets. Even where the issues are understood by both sides, dissonant objectives may result in very different courses of action!

Conclusions

We might speculate as to whether the responsibilities of good netizenship that are applicable here might be applicable across the Internet community as a whole.

Today, there are a wide range of opinions on acceptable practice amongst individuals responsible for the protection of their co-workers from networking security threats – or so regular consultation of firewall lists, *BugTraq* and other mail resources concerned with the discussion of vulnerabilities and solutions would suggest.

Consider the well-worn debate between the advocates of full, partial or non-disclosure of security threats. While there are vendors and users who would be pleased to see disclosure of threats to their armouries banished, there are numerous individuals who seem to spend their waking hours exhaustively testing software for new weaknesses – not with the intention of exploiting them maliciously, but in the hopes of alerting vendors and the user community – and perhaps garnering some personal glory in the process.

Consider also the contrast between those who advocate countering IIS-related threats with software-patching counter-worms, and those with a pathological fear of overstepping another organization's boundaries. Many organizations simply discard email carrying viruses/worms at the gateway, or even all mail carrying virus-friendly attachment types.

Measures like this can do a good job of keeping threats out of the organization but don't necessarily alert the sources of such mail to their potential or actual problem. This is in sharp contrast to the hail of wrathful emails and complaining posts to discussion lists that can descend upon a source of fast-burners within a WAN linking many sites and organizations. It seems a pity that there is no immediate prospect of a centralized body to co-ordinate piecemeal Internet responses to such crises.

FEATURE 2

The New 'Internet Background Noise' – Windows Worm Probes

Juha Saarinen
Independent industry commentator & technical writer, New Zealand

In 1989, Robert Braden described the 'Robustness Principle' as follows:

'At every layer of the protocols, there is a general rule whose application can lead to enormous benefits in robustness and interoperability:

> "Be liberal in what you accept, and conservative in what you send."

'Software should be written to deal with every conceivable error, no matter how unlikely; sooner or later a packet will come in with that particular combination of errors and attributes, and unless the software is prepared, chaos may ensue.

'In general, it is best to assume that the network is filled with malevolent entities that will send in packets designed to have the worst possible effect. This assumption will lead to suitable protective design, although the most serious problems in the Internet have been caused by unenvisaged mechanisms triggered by low-probability events; mere human malice would never have taken such a devious course!' (See *Request for Comments* (*RFC*) memo 1122, 'Requirements for Internet Hosts'.)

Although I understand where Braden is coming from, as malware authors are becoming increasingly radical (and careless) in what their products send out, these days it is difficult to remain liberal in what you accept.

Certainly Braden was right about the network being filled with malevolent entities, although he underestimated the human capacity for malice and deviousness – just like the majority of software and network engineers, it seems.

Recently, one of the Java developers working on a web application at a company I have been contracting for complained about a number of entries in the web server log files (this is a more recent entry, not one of those he alerted me to initially):

```
210.201.88.125 - - [01/Aug/2002:08:15:03 +1200]
"GET /scripts/..%252f../winnt/system32/
cmd.exe?/c+dir HTTP/1.0" 404 - "-" "-
```

On seeing the entry I thought, 'Right, a *Windows* worm' – I guessed it was a Code Red or Nimda variant, or perhaps even someone or something pretending to be one. This was odd; *Windows* worm activity in July 2002 still? Since the servers in question run a Unix-clone operating system, I wasn't overly concerned about the security implications for the systems themselves. However, as I delved further into the web server log files, I was stunned to see how many 'worm hits' had been recorded.

One virtual server that had been used for testing purposes only, and which had run for less than a month, showed just under 3,000 entries in its log file. In fact, all the HTTP GET requests in the recent logs for that server were 'worm signatures'. More grepping through the log files for other virtual servers brought the total number of *Windows* worm hits to around 50,000 over a two-month period – many more than expected.

The *Virus Bulletin* prevalence table for June 2002 lists W32/Nimda in tenth place, with a minute 0.46% share of all reported instances. Code Red doesn't even appear in the table, so I was surprised to see such heavy worm activity recorded in the server logs.

A straw poll on the NZ Network Operators Group mailing list indicated that this company's servers weren't alone in being awash with *Windows* worm attacks. One administrator reported having seen over 5,000 hits in about eight hours.

Where Do They All Come From?

Most of the worm probes on our hosts emanate from systems in adjacent APNIC net blocks. Most of these are in China and Korea, as IP address space used in Australia and New Zealand is adjacent to those and other Asian countries. For instance, I now know that there are *Windows* users in Kathmandu who are infected with Code Red.

Unfortunately, sloppy systems administration seems at least as common in the APNIC area as it is elsewhere, with the added aggravation of a language barrier. While communicating in English works fine with ISPs located in Europe, Africa, the Americas and India, use of the English language doesn't usually work well for complaints about worm probes, spamming, and other malfeasance to Korean and Chinese network operators. Some of the ISPs operate an abuse desk, and return an auto-acknowledgement email with incomprehensible double-byte characters, but generally nothing happens.

However, my initial assumption that the frequent worm probes were an APNIC area phenomenon turned out to be wrong. A check on systems in the US showed that they too receive a large number of *Windows* worm pokes. One host recorded approximately 30,000 probes in three months, and is hosting only some small, personal websites.

What's more, the worm probes were from a wide variety of hosts, in geographical terms. As most of the IP addresses that are targeted by the worm are adjacent to that of the infected host, and only a small number are selected randomly, this indicates that there are still a large number of unpatched *Windows* systems plugged into the Internet without, or with deficient, anti-virus protection.

A quick check of the website of the volunteer Distributed Intrusion Detection System (*DShield*) organisation that records worm activity and system probes around the world appears to confirm this suspicion. The graphs for Code Red probes (see http://www.dshield.org/coderedhistory.html) and port 80 scans indicate that *Windows* worm activity has indeed become the new 'Internet background noise'.

A Nuisance, or a Real Danger?

Being located in New Zealand, where bandwidth is expensive and is usually charged by the megabyte (for instance, ADSL users pay 20 NZ¢, or approximately 9 US¢, per MB), it's important to reduce data traffic wastage. The worm probes weigh in at around 600 to 1,000 bytes each, taking into account TCP/IP overheads for setting up and tearing down the connection, as well as the GET request and server response. For a single IP address, an additional 50 MB over a two-month period is neither here nor there; however, the worm sweeps through entire 24-bit CIDR blocks, so the total traffic wastage is likely to be substantial.

The frequency with which the probes are rolling in is another concern: two to three per second per IP address, on each virtual server. The servers shrug off the frequent probes with ease, nevertheless, system resources are consumed that would be better utilised dishing out actual content instead. *Windows* worm probing is definitely not the load test that was envisaged.

Then there's all the additional administration work caused by the *Windows* worms: the chaps and chapettes in media sales who produce reports based on the server logs do not have worm filters on their tools, so all the extra hits and bandwidth consumption cause a certain amount of analysis distortion to start with. Fortunately, Unix comes with a good set of text processing tools with which the logs can be 'washed', before they go off for analysis.

Come to think of it, I hope everyone else cleans their logs like this. It would be terrible if worm probes were passed off as evidence of visits to websites.

How to be Less Liberal

Naturally, traditional anti-virus solutions aren't much good for dealing with incessant worm probes. It is not acceptable to deploy, for example, a Nimda cleaner to an infected host probing your network, tempting as it might be.

Blocking out the worst offenders based on IP address or host names wouldn't work either, because the worm-infected hosts are so widespread.

Some routers can be configured to drop typical worm HTTP GET requests for 'cmd.exe' and similar as a 'stop-gap' measure.

Content-filtering firewalls are certainly an option, but the majority of these seem to focus on stopping infected emails and web pages, rather than those pesky HTTP worm probes. Filtering proxies can also be employed to intercept harmful HTTP traffic.

However, these passive solutions add both cost and complexity, and only hide the symptoms rather than tackling the fundamental problem – you still end up receiving the probes.

LaBrea

One interesting idea that attacks the modus operandi of the worms is Tom Liston's *LaBrea* application (see http://www.hackbusters.net/).

Named after the famous tarpits in Los Angeles, *LaBrea* traps probes on unused IP addresses on your network, and completes the TCP handshake excruciatingly slowly. This slows down the worm (and other scans), rendering them effectively useless. Liston estimates that, in order to maintain a TCP connection in 'persist' state (i.e. in a constant wait state), only 1,215 bytes per hour of bandwidth is required.

To be really effective, Liston says a 'chunk' of real or publicly-routable IP addresses are needed. You can use *LaBrea* on networks with a single public IP address and NAT (network address translation), but the problem is that you won't be able to tarpit the ports that you are providing services through. You can, however, forward unused ports to a system running *LaBrea* on the internal network, and trap other probes there.

Network operator Joe Abley, of Canada, says *LaBrea* has an interesting beneficial side-effect in that it mops up ARP (Address Resolution Protocol) requests from routers and hosts that would otherwise go unanswered, yet take up system resources and possibly kill the router.

If you have an always-on connection (cable and/or ADSL, for example) and you're sick of incessant *Windows* worm probes, *LaBrea@Home* could be the ticket. It's a *Windows* application that listens on port 80 only, and works in conjunction with a firewall to trap probing hosts indefinitely in a wait state. Both *LaBrea* and *LaBrea@Home* are free for non-commercial use.

The Way to Go?

I would welcome some further discussion of whether active solutions like *LaBrea* are the best way to attempt to combat a situation that's already serious, and which is likely to worsen in the future. [*Please send your thoughts on this subject via the Editor of* Virus Bulletin *– email comments@virusbtn.com.*]

FEATURE 3

24-Hour Anti-Virus Service

Jaime Lyndon 'Jamz' A. Yaneza
TrendLabs, Trend Micro Inc., Philippines

The anti-virus industry has changed from being just a collection of product providers to becoming a fully-fledged service. Gone are the days of simply providing pattern solutions on a scheduled release.

In these times of increasing Internet connectivity, and with the growth in computing mobility, it is an obvious fact that virus outbreaks in localized regions can become global threats in the blink of an instant. Thus, support services that deal with various forms of information dissemination have been created to address the needs of customers reporting numerous infections. Not to mention that several avenues of providing stable pattern solutions from non-corruptible or non-assailable deployment sites have also been implemented. All of these changes have come about as a result of the rising growth in mass-mailing malware that, in recent years, has become the norm.

The provision of customer solutions during a virus outbreak is a time-critical situation. Therefore it is imperative that a controlled process should be in place to address each event properly, even as customers continue to report other infections. Solutions do not end with providing pattern updates – a total solution includes full system cleaning as well as information to patch vulnerabilities and to trace the source of infection and close it.

Product Support

A full service company provides pre-sales and after-sales support. When *Trend Micro*'s customers encounter a problem, their first line of contact is a product support specialist. In other areas they are called technical account managers and support engineers.

The basic technical training for a specialist includes network and operating systems, product knowledge, as well as a good smattering of anti-virus concepts. Customer-centred training, focused on telephone and email support techniques, rounds off the process.

Operating systems training includes troubleshooting for all the major platforms on which the range of enterprise products can be installed. This covers flavours of Unix, *Linux*, *Novell* and *Windows* including proprietary systems like *HP OpenView*, *IBM AIX*, and *Lotus Groupware*, to name but a few.

Naturally, product competence will range from desktop to enterprise-wide products that span gateways, file servers, email servers, and the like. It is not uncommon for third-party solutions such as firewalls to require parallel learning curves for various developed products.

Customer focus takes into consideration the fact that, as a global operation, clients may have regional language differences. For hiring and training purposes, support personnel assign added value to engineers who can speak more than one language. The major languages include English, Japanese, German, and Mandarin.

This combination of competence is what guarantees that customers will receive quality responses without too much follow-up. During an alert, specialists are in charge of remaining constantly in contact with their clients while providing immediate updates and developments. This relationship continues until the problem is solved.

Anti-virus Support

The heart and soul of an anti-virus vendor is represented by its anti-virus engineers, who are required to analyse and investigate malicious files, create detection and cleaning signatures, as well as provide the initial solutions to virus-related cases.

Engineers go through rigorous training that includes disassembly, low-level code analysis, basic network topology and architecture, as well as knowledge of the products currently supported for a given platform.

Anti-virus engineers are sticklers for time. Each case received through the dispatching system is assigned a solution cycle time, depending on the 'level' of the customer, which is based upon direct customer feedback and the results of market survey. 'Premium support' clients are guaranteed a two-hour solution delivery, while the aim is to resolve cases from business units, as well as regular cases passed through retail, within 12 to 24 hours.

Outbreak situations are classed as Local and Global. During such instances the response time worked to is 60 minutes. This timeframe includes analysis, scan and clean signature creation, quality assurance and product integration, as well as information and pattern upload to deployment servers worldwide.

Depending on the severity of the malware in question, fix tools and product-specific backup processes are also initiated. This interim step bridges the gap from the time a case arrives for analysis to the point an updated pattern is released and deployed.

Sometimes, personnel with an exceptional technical background are given the option to join the research team – a specialized group that handles escalated issues related to scan engine modifications or performance optimization.

Support engineers will often escalate inquiries found to be virus-specific, rather than product-related, to engineers for simple feedback or full case handling.

Dispatchers

It is annoying for customers to receive ‘canned responses’. Nothing beats human intervention. You can only go so far in providing generic template responses. For this reason a full service company should be aware of specific customer requirements in terms of solution formatting and wording. Dispatchers can man an implemented dispatch system for case distribution to anti-virus engineers. In doing so they represent the front-end to customers and, as such, must be in charge of checking how solutions are worded for clarity and thought sequence.

Quality Assurance

Each case requiring a signature modification, addition or removal goes through two major testing stages, namely false positives and false negatives.

False positive testing requires the comparison of detections against a base of normal files. These ‘normal files’ include common applications from different platforms, regions, and verified customer submissions. This group of files must be updated and added-to regularly, ensuring that past, present, or future software releases from major application vendors are not misidentified as malware.

False negative testing compares the results of scanning the current set of viruses available to each vendor. The aim is to make sure that no pattern detects the same sample of malware more than once. Duplicate identification results in wasted signature size, which is a rising problem today, as well as confused product reporting. Likewise, exact identification is another issue raised by customers as it would help them to prepare for whatever contingencies are needed to combat the threat.

Considering the bulk of file types to be scanned, a good method of false detection testing is to segment scanning into files of related type. As an example, macro viruses are scanned against files that can actually contain macro viruses, while 32-bit based malware is scanned against a set of files that are likewise 32-bit in type. This is, of course, only the general idea.

Deployment

With all the new malware that keeps being foisted accidentally or intentionally on us over the Internet every day, it is unimaginable that floppy disks or CD-ROM packages received through snail mail can still be considered an acceptable form of update. After all, the Internet is already an essential part of our daily life, so why not use it for immediate updates?

In the not-very-distant past, monthly updates were considered ample. Later, this frequency became bimonthly, weekly, and until recently daily. However, current customer demand dictates that updates be released as often as needed. Given an estimate of the malware sets currently available for processing, a safe time-frame would be to make ‘as needed’ updates available every four hours.

How should the updates be made available? I remember an instance some time ago where W32/MTX rendered site-based updates useless. This caused a stir as customers were understandably upset. FTP updates were available but this rendered some automatic features of their installed anti-virus software useless. Decentralized updates through *Akamai* servers are now in general use.

Discussions on the downside and redundancy of signature updates are another issue of debate among certain large and small corporate circles. Indeed, isn’t it possible to have signatures that will detect all malware generically and perhaps for all time? It is an interesting concept, but one that I would rather not discuss in this article.

However, let us acknowledge that the anti-virus industry is doing what it can to address this issue given the fact previously mentioned with regards to the increasing size of current signature files and the resulting slow-down of scanning and detecting malware as a side effect in some scanning engines.

Service Level Agreement

Can you guarantee that the service you provide works as advertised? This is a new reality that service companies have to face today. It will surely raise the bar and separate the men from the boys.

When you subscribe to a mobile phone service, you expect to get the advertised amount of free airtime minutes and that the system works.

Similarly, as an ISP subscriber you expect no downtime, especially when you host a high-traffic e-commerce site. As customers we expect to get what we pay for!

Relating this to the anti-virus industry, can this be achieved? Essentially this should be possible. When a piece of malware is submitted it goes through several standard stages of processing. When customers subscribe for premium support, they receive their solution at the time agreed upon. Is that so hard to do?

Conclusion

Full service does not end with system disinfection but should include system clean-up. It is no longer enough to promise an estimated time for case resolution. Neither is it acceptable for solutions to be of inferior quality, thus dragging out the issue for days. Signature updates need to be rethought in terms of size and deployment strategies.

The anti-virus industry needs to reinvent itself if it is to meet current customer demands.

FEATURE 4

Fishing for Hoaxes: Part 1

Pete Sergeant

For some time, *Virus Bulletin* has invited subscribers and visitors to its website to forward emails they suspect may be virus hoaxes to *VB*. Sorting through hoaxes and suspected hoaxes can become quite a time-consuming process, so one of the first ideas that struck me when I joined *VB* was to try to implement some form of automated hoax classification.

Early attempts were fairly simplistic: certain keywords or concepts were searched for in messages, and awarded a score. The higher an email's score, the reasoning stood, the more likely it was to be a hoax. This is very similar to the way in which *SpamAssassin*, an open-source spam identifier, works – in fact *SpamAssassin* was the inspiration for this approach.

Keyword-based Identification

So, which keywords and concepts appear in most hoaxes? I spent some time pawing over *VB*'s collection and came up with a fairly long list:

Mention of a cash reward

Some hoaxes are nothing to do with viruses, but cause many of the same problems as those caused by virus hoaxes. A particularly prevalent type seems to be those that mention some kind of cash reward for forwarding the email to your contact list, or to a specified email address. Therefore, emails that contain mention of a cash amount can have their score raised slightly. Example: the 'Honda' hoax – see http://www.virusbtn.com/resources/hoaxes/honda.xml.

Mention of a major news source

Many hoaxes attempt to validate their claims by indicating (falsely) that the virus about which they're warning has been mentioned by a major news source, for example *CNN*. In fact, *CNN* is very popular, so emails that mention *CNN* have a raised score. Example: 'A virtual card for you' – see http://www.virusbtn.com/resources/hoaxes/virtual_card.xml.

Instruction to forward the email to everyone in your address book

In order to survive, a hoax needs to be passed on. Emails containing phrases such as 'forward this to' and 'address book' score highly. Example: the 'budweiser frogs' hoax – see http://www.virusbtn.com/resources/hoaxes/frogs.xml.

Mention of McAfee

This is a strange one. Like the mention of a news source, hoax writers like to use the names of well-known anti-virus products. Either 'product X doesn't detect it' or 'vendor X has said that …' – the idea is quite simple. *McAfee* seems to enjoy particular popularity with hoax writers, and thus the occurrence of *McAfee* in an email is enough to raise the hoax score a little. Example: the 'jdbgmgr' hoax – see http://www.virusbtn.com/resources/hoaxes/jdb.xml.

Well known keywords

Some common hoaxes, mention some very specific keywords. These are often the hoaxes that ask you to remove system files. Any email mentioning 'sulfnbk.exe' or 'jdbgmgr.exe' will have a raised score. Example: the 'sulfbnk.exe' hoax – see http://www.virusbtn.com/resources/hoaxes/sulfnbk.xml.

Problems with this Approach

This approach was found to be far from infallible:

- Emails giving instructions on how to *restore* jdbgmgr.exe or sulfnbk.exe could conceivably be marked as hoaxes, most likely confusing the recipient somewhat (especially if they were sent from *McAfee*, and/or trip other keywords too).
- It would be quite easy to write a hoax that specifically avoids using these keywords if you have a rough idea on what it will be filtering on.
- You could spend a lot of time going through hoaxes trying to find effective keywords – a Pyrrhic victory if you like.

An 'almost good enough' solution is, unsurprisingly, not really good enough – especially when you consider the potential effects of misclassification. If a single hoax slips through the net, it's not the end of the world. The user may identify it as a hoax themselves, or they may forward it to all their friends – annoying, but imagine what happens if a real virus alert is identified as a hoax: users with faith in the system could suddenly feel quite happy about clicking on files sent to them 'in order to have [their] advice' …

Hoaxes – Nature's 'Bozo Bit'

The concept of a 'Bozo bit' was introduced in Jim McCarthy's *Dynamics of Software Development*. When someone says or does something you deem to be stupid, you flip the 'bozo bit' on them. From then on, contributions and comments made by that person are taken to be pretty worthless. Forwarding hoaxes to people in your company is likely to lead to your 'bozo bit' being flipped.

Hoaxes can and do cause problems other than just making those who forward them look stupid. First, some hoaxes ask users to perform potentially harmful actions on their computers, like the removal of system files (jdbgmgr.exe or sulfnbk.exe). More often, they clog up mail systems, annoy the recipients, and soften users up a bit – a user who has been chastised time and again for being paranoid and forwarding hoaxes could become blasé about clicking on attachments.

How Bayes Works

Bayesian categorization works out the probability of a document belonging to a given category (such as ‘hoax’ or ‘non-hoax’), based on documents you’ve trained it with. Each word has a score for the probability of its appearing in a given category – given a document for which you don’t know the category, you look at the words in it and use their probabilities for determining its category.

In this case we can already guess some of the words that’ll have a high probability of residing in a given category: *McAfee* will have a high hoax-rating while a product like *Sophos Anti-Virus*, by virtue of its not being a home-user product, is likely to appear less frequently in hoaxes, and thus have a lower hoax-rating.

Thinking Outside the Box

Some other interesting trends should be revealed by the data, and working with it.

Brand name awareness

The likelihood of a vendor’s name appearing in hoaxes is related to awareness of its brand name amongst home users.

Searching through a large corpus of hoaxes, and comparing it with, say, archives from an anti-virus mailing list should indicate which vendors the hoax-writers have heard of, and which they think others will have heard of. How useful is the opinion of a hoax-writer? Not very, but some hoaxes are notably more successful than others – a hoax that inspires ‘confidence’ in its recipients is a hoax that speaks the language the readers want to hear. If a prevalent hoax mentions an AV vendor, we can perhaps assume it’s a name that those who’ve forwarded the message have heard of.

Foreseeable Problems

Bayesian filtering of spam has been shown to work well, but with a few caveats.

One size doesn’t fit all

Filtering of spam works well on an individual’s mailbox, but probably not so well for a large number of mailboxes. In order to be effective, a hoax filter will, most likely, need to be deployed across a company’s mail system. Different users may receive very different types of email: some users are likely to receive emails written in a very similar style to that in which hoaxes are written – sensationalist and with poor English. Misclassification could rear its ugly head.

Not all email is alike

By far the biggest problem I foresee is virus-related emails being unfairly penalised if the system has been trained with ‘everyday’ email. Words like ‘virus’ will be well represented in virus hoaxes, but probably would be fairly scarce in day-to-day email. One way to get around this is to train the system using, for example, the archives of a major anti-virus mailing list, or slurping from a newsgroup like alt.comp.virus. But, again, if most users aren’t getting virus-related email, then some everyday words that are unlikely to filter in genuine virus-related email will be unfairly penalized. To what degree this will present a problem remains to be seen.

Getting the goods

For someone looking to implement a system like this, obtaining large amounts of hoax and non-hoax but virus-related email could present a problem. Luckily for me, my pleas to AVIEN were answered, and I was sent large archives of both types, sanitised to preserve anonymity. Another good source is the vendors themselves – training using archives of their outgoing mailing lists has the added advantage of making it less likely that vendor mail shots will be caught.

What to do …

Assuming the system works (to be looked at next month), how is it best to implement this? Bouncing hoaxes at the mail server level seems like a bad idea – it worsens the problem of false positives. The best solution seems to be to add extra headers to the email itself, and let the client deal with it. The email RFC allows for ‘extended’ header types. For example:

```
From: Wordsmith <wsmith@wordsmith.org>
To: linguaphile@wordsmith.org
Subject: A.Word.A.Day—feisty
X-IMAPbase: 1032345651 2 NotJunk Junk
X-Keywords: NotJunk
X-UID: 1
```

This section of headers is a good example. My mail client (Mac OS X 10.2 mail.app) contains spam-filtering technology, and the ‘X-Keywords’ header is added to my messages, with a value which depends on whether the client suspects it is spam. This allows me to apply rules to my mail – if a mail has a junk header, put it in a certain folder.

The addition of an appropriate hoax header would allow the same thing – the system administrator could have a special ‘hoax’ folder on clients’ computers into which the mail client put the hoaxes. This has the effect of warning users that the email is *probably* a hoax, but allows them to look at the email to decide for themselves.

And for My Next Trick …

Next month we’ll look at examples of how well this system works – will there be words we hadn’t suspected that feature very heavily in hoaxes, or indeed words that appear commonly in non-hoax emails that very rarely appear in hoaxes? Which brand is the hoax-writers’ favourite? Is the system fast enough for real-world use? Watch this space.

INSIGHT

A Brief History of (My) Time

Anton Zajac
Eset, USA

My universe began in 1957. This was a remarkable year both for me and for three physicists: J. Bardeen, L. Cooper and R. Schrieffer (collectively known as 'BCS'). While I had just started learning the basic vital human functions, BCS had revealed the first microscopic theory of one of the most mysterious phenomena of nature – superconductivity.

As for the vital functions, I did well (this article serves as sufficient experimental evidence). The Nobel prize awarded to BCS in 1972 proves, beyond reasonable doubt, that Bardeen, Cooper and Schrieffer did not waste their time either – this in spite of the fact that their theory was later proved to be flawed. In 1957, I had no idea that the BCS theory would become the main focus of my research some 32 years later.

Dreamers of the Sixties

From the point of view of computer viruses, the sixties represented a pastoral peace. Computer viruses might have occupied the (wildest) dreams of a science fiction writer but, frankly, I doubt it. However, dreams were quite active at that time in the minds of Peter Pasko and Rudolf Hruby who (together with Miro Trnka, born in 1961) would become the founders of *Eset*.

The Happenings of the Eighties

After a decade of 'still waters', things really started to happen in the eighties. One of the first highlights of that decade was in 1986, when Ralf Burger gave a lecture on his 'Virdem' virus at the conference of the Chaos Computer Club.

Some time later in Vienna, a city located about 25 miles from Bratislava (Peter Pasko's hometown), a new experimental virus (Vienna) hit the streets. Historically, Bratislava was the coronation city of the Austro-Hungarian monarchy and the 'intellectual' distance between the two cities is even shorter than the geographical. Peter Pasko knew that, but it took him three days to figure out that the strange thing that was happening in his computer could be attributed to the programming efforts of a high school student in the nearby city.

Peter would not have had the opportunity to analyse the Vienna virus, nor to write the first virus scanner and cleaner (AV2), had he not decided to invest his (very) limited financial resources in his first PC rather than carpeting his apartment (Peter still owes an explanation to his wife). Later, the FallingLetter virus presented the second (and, due to its encryption features, even bigger) challenge to the author of the predecessor to the *NOD* scanner.

Meanwhile, 1986 marked a fundamental milestone in physics: G. Bednortz and A. Miller, *IBM* scientists in Switzerland, discovered what seemed to be impossible according to the BCS theory – the high temperature superconductivity which shifted the critical temperature to the realm of relatively cheap liquid nitrogen.

In 1987, Fred Cohen proved his legendary theorem regarding the impossibility of the creation of an anti-virus system that could, with 100% certainty, detect all imaginable viruses.

This, not exactly encouraging, prospect did not prevent Miro Trnka from outlining the development strategy for a comprehensive general-purpose anti-virus system and creating the environment which, when combined with Peter Pasko's scanner, would become the first version of *NOD*.

Miro also created a utility for checking the integrity of the MBS. Generously, he granted a free licence to a friend who worked at the nuclear power plant in the city of Trnava. A few months later, he received a worrying phone call from this friend, informing him that the utility was reporting constant changes in the MBS after reboot of the computer. That is how the DiskKiller virus was discovered locally. The nuclear power plant was not controlled by the infected machine, but the mere thought of what *might* have happened had the infected machine been in control of the plant is the stuff of nightmares …

By the time (1989) Peter Norton proclaimed computer viruses to be 'urban legends, like the crocodiles in the New York sewers', Peter (Pasko) and Miro had already answered Hamlet's famous question: 'To be or '*NOD*' to be?', since the first (coded in Turbo Pascal) version of *NOD* had been sold. It was not an easy sale, since the legal system of what, at that time, was the former Czechoslovakia, did not allow the opening of a private enterprise!

NOD had to be sold outside of Czechoslovakia, and Vienna was the natural choice. Any well-informed user of *NOD* in Austria could have traced the program's origins to the neighbouring country since the product owes its name to an extremely successful TV series named *Nemocnica na Okraji Mesta* which, in English, reads: 'Hospital on the Edge of a City'. The word 'Mesta' (City) was replaced by the more appropriate 'Disk', so the resulting acronym reads 'NOD' – 'Hospital on the Edge of a Disk'.

1989 was a very significant year for all former Eastern European countries. The iron curtain came down, opening up new entrepreneurial and scientific opportunities. Anti-virus vendors-to-be, such as *Alwil*, *DialogueScience*, *Grisoft*, *Eset*, *Kaspersky Labs* and *VirusBuster*, were slowly pawing their way into the international anti-virus arena. This was timely, since virus writers, especially in Bulgaria and Russia, also took advantage of the emerging 'markets'. At that time, I left Czechoslovakia to spend five productive years lecturing in San Diego, USA.

January 1989 witnessed the famous Jerusalem virus, followed by the Holland Datacrime virus. The latter was referred to in the US as the Columbus Day virus (due to confusion over its supposed activation day), and it was suggested that the virus had been written by a Norwegian expressing anger over the fact that the discovery of America had been credited to Columbus rather than his co-patriot.

The first edition of *Virus Bulletin* was published in this year. The tag line of the first issue claimed *VB* to be an 'authoritative' international publication. The years that have elapsed since that time have justified that claim and the fact no longer needs to be verbalized in the magazine.

1989 also saw an increased interest, amongst the general public, in virus threats and a raised awareness of their potential to cause damage. This may, in part, have been due to the media circus that surrounded some of the virus epidemics, but significant credit goes to the fact that major companies such as *IBM* suffered infections that affected their clients and/or operations.

Tequila with (NOD-)iCE?

The situation became more serious the following year (1990). Mark Washburn created the first polymorphic virus and the Bulgarian Dark Avenger debuted some of his novel virus-writing skills, e.g. the fast infector technology. As an additional impetus for virus writers, several new virus exchange BBSs were established, offering full access in exchange for a new virus. As a countermeasure to these activities, EICAR (European Institute for Computer Anti-Virus Research) was established to serve as a catalyst for anti-virus research in Europe. Peter Norton gave viruses a second thought and established *Symantec Corporation*.

The recent steep increase in Tequila prices certainly did not apply to the 1991 Tequila virus that was not intended by its Swiss writer to be released into the wild. But the virus broke loose. After a brief threat from the Dark Avenger, the Self Mutating Engine (MtE) was released in 1992, presenting an attractive topic for Miso Weis' student research paper and program (1993). This talented university student, who would later (together with Palo Luka) become one of the chief architects of *NOD-iCE version 7*, developed a 100%-efficient MtE detector and cleaner. This was a remarkable achievement since, with the exception of Thunderbyte, the average hit rate of most common systems was between 60% and 80%.

1992 was also the year of the Michelangelo virus, and saw the advent of virus-authoring packages and for-profit sales of virus databases. But, from my point of view, the most important event of this year was the creation of *Eset*. Naming the company after the Egyptian goddess (Isis) who was able to revive her dead and mutilated husband, the founders' ambition was to provide tools to revive virus-infected 'dead' computers and, of course, prevent such grim scenarios in the first place.

As the new virus clones became more complex and more frequent, new anti-virus technologies were imminent. AV companies began experimenting with and applying advanced scientific methods, such as neural networks, expert systems and heuristics. In the summer of 1993 a strategic decision was made in *Eset*, leading to the development of a new code emulator, which was released in version 7. The GUI remnants of version 7 can still be found in the DOS version of *NOD*.

Due to the emerging availability of new software platforms and deficiencies of the 16-bit architecture, *NOD* had to be completely upgraded into 32-bit architecture in 1997, a major task carried out under the supervision of Richard Marko and Maros Grund. *NOD32* (the third generation of *NOD* system) had its premier international performance in May 1998, when it earned its first VB100% award.

Dedicated to AV

I joined *Eset* in the autumn of 1998, where I carried out the English (and an additional six languages) localization process, and in 1999 I co-founded *Eset (US)* with the original *Eset* founders. I created, and now I am managing, *Eset*'s worldwide distribution network.

In 1992, shortly before *Eset* was founded, I published four papers in the peer-reviewed *International Journal of Quantum Chemistry*. The papers present a unified theory of low and high temperature theory of superconductivity. In 1995 I founded a high-tech company (*S-Tech*) to develop *Solid*, a computer program serving as a tool for the design of new superconducting materials based on the aforementioned theory.

Solid was completed in the year 2000 and was applied successfully to obtain information on properties of existing and fictitious (never synthesized) materials. The *S-Tech* team received the 'Werner von Siemens Excellence Award' for *Solid*'s scientific excellence in 2000.

When the principal high-tech mission of my venture had been accomplished and my involvement in *Eset* required an increasing amount of time and energy, I made the tough decision to sell *S-Tech*.

Today, I belong to *Eset*, and to the anti-virus industry, entirely. Although, as my friends will testify, that is not *quite* true: some part of me belongs to the ocean.

OPINION

What’s Coming? Part 2

Peter Morley
Network Associates Inc., UK

This article, four months after the first one (see *VB*, May 2002, p.16), comes sooner than I expected but there have been some unexpected developments.

The 2002 ‘dead period’ I predicted has started, but it’s not as slow as I expected, and now I expect the number of new items of malware to remain at around 200 per month for a while longer, instead of reducing to 150 per month. I still believe that 2003 will be a very quiet year in terms of malware.

I received several comments about my last article:

1. ‘I was hoping you would say something about the *Microsoft Palladium* project.’
2. ‘You were a bit superficial about 64-bit processing.’
3. ‘I disagree with the view that *Linux* Trojans will grow seriously, because the organization of *Linux* makes this difficult.’
4. ‘I was hoping you’d mention the death of the hard disk, and its replacement by fast solid state devices.’
5. ‘What about the *Microsoft* Longhorn project? (or is it concept?)’

In contrast, there was apparent agreement that *Linux* viruses were not a serious hazard, and that *Windows* viruses and Trojans will be with us for some five years.

New Hardware and Palladium

Palladium is the next *Microsoft* project and is intended specifically to address security problems (as in Bill Gates’ injunction). It will offer much greater control over who accesses what, both locally and remotely. It will offer recognition of people using fingerprinting, and maybe recognition of eye patterns. I’m convinced that it will make it more difficult for *Windows* viruses and Trojans to spread.

It *must* offer full backward compatibility for previous *Windows* users. If it doesn’t, the users will not move, or will drift off to *Linux*. I believe 64-bit processing is a prerequisite to the new *Palladium* features.

Palladium requires several new processor instructions. It also requires a new motherboard chip in addition to the processor. These items have been discussed with Intel, AMD, and IBM, and will soon be revealed to motherboard makers, if they have not been already. Several magazine ‘Road Maps’ have appeared, summarizing Intel, and AMD plans.

This is my picture of the next three years:

In about the fourth quarter 2003, a new *AMD* chip and motherboard will appear (*Clawhammer X-64*), followed closely by *Intel*’s *Yamhill*. Both will have to be fully backward compatible, and I expect the transition to be as painless as the 286 to 386 transition. This assumes *X-64* is late since it will include the new instructions.

Yamhill is in the Road Maps for mid-2004, but I expect it sooner. *Itanium* (*Intel*’s initial 64-bit processor which lacked backward compatibility) has been bumping around the runway for a while, and few think it will take off! This has put pressure on *Intel*, and I believe they will respond.

I expect the new processors and motherboards to settle down and become standard by the fourth quarter, 2004.

The stage is then set for the arrival of *Palladium*. Although *Microsoft* says it is expected mid-2004, history indicates that such projects can slip. I think the last quarter of 2004 seems more realistic, particularly as it fits the hardware scenario.

Linux

Linux now:

a) Has become more popular over the last six months

b) Four *Linux* suppliers, (*Caldera*, *SuSE*, *TurboLinux* and *Collectiva*) have been collaborating. This is a good start to the process of chaos reduction. However, *RedHat* and *Mandrake* are not in this group, and this will limit its effect.

Linux between now and the new chip:

Nothing much will happen. *Linux* already supports 64-bit processing. There may be some behind-the-scenes work on adding support for the new processor instructions.

Linux between the new chip, and the arrival of Palladium:

Pandemonium! How can we implement as much of the new *Palladium* capability as we need to?

Linux after the arrival of Palladium:

This will depend on how effective the pandemonium has been. If it has been ineffective, the growth of *Linux* will slow. If it has been effective, it will be business as normal.

There is another factor which may affect all of this. *IBM* has poured cash and resources into *Linux* and may begin to get its act together at the bottom end of the hardware spectrum. (Currently, it’s chaos – I have a full-page *IBM* newspaper advert, from July 2002, which says ‘IBM PCs use Genuine Microsoft Windows’. Rather than being a case of the left hand not knowing what the right hand is doing, it's a case of the bottom end not knowing what the middle and top are doing!)

None of the above will make the spread of viruses and Trojans any easier and it may well become more difficult. Therefore, I shall accept the comment from Curtis H (see *VB*, August 2002, p.4), and withdraw my estimate that *Linux* malware will start to take off at the end of 2003. My position is now '*Que sera sera*.'

In case you're interested in what I'm doing about all this, I have set up a machine with two disks, one fairly small and the other large. I can install any operating system on the first disk, and do what I like with it, taking an image copy to the second disk whenever I want to. By re-imaging to the first disk, I can return to any situation I have kept.

Of course, it is possible to add a third and a fourth disk and to disconnect all the image disks before performing hazardous operations on the first one.

Hard Disks and SSDs (Solid State Devices)

I don't believe anything in this section will have the slightest effect on the incidence of viruses or Trojans.

However, it may affect backup and recovery procedures, and therefore it may affect the ease of recovery from them. It may give another gentle hit to the AV industry. So let's review it anyway.

Within the last two months, the billionth PC was shipped. Whilst many have no hard disk, some have more than one. It matters not – there are a lot of HDs out there!

SSDs have started to appear. However, they are small by today's standards, and they seem to need Firewire or USB2, and so far have been external.

IBM has sold off its hard disk operation to *Hitachi*, and is pulling out. You can take this as meaning 'Try as we might, we cannot compete with the Far East, and we think that Hard Disks are about to die, anyway. We reserve our position on SSDs to replace them.' As long as SSDs remain external, they will be irrelevant. In order to become possible components of a new PC, they have to have an internal interface.

If they are to require a *new* interface, they will need another new generation of motherboard. And if they are to be able to replace or supplement existing EIDE, or the various SCSI disks, they will have to provide the means to use the existing cable connection. (Or, you say, a new controller card. You're right, but I think it will need a 64-bit PCI slot, and that probably needs a new motherboard.)

The volumes are enormous, so it is not possible for suppliers to duck these questions. Luckily, we can all lie back, and see what happens, just as we have with previous Hard Disk developments.

Longhorn

Bill Gates now has the title 'Chief Software Architect' and has passed over a lot of his Chief Executive functions to Steve Ballmer. Bill is well into reviewing the subject of *Longhorn*. *Longhorn* is the new operating system to follow *Palladium*. It is new from the ground upwards, with emphasis on ease and convenience of use, irrespective of type of device. It is scheduled, optimistically, for mid-2005.

I believe that *Microsoft* will deliver (and that it will be more like the first half of 2006). Between now and then, Bill Gates will be concentrating initially on *Palladium*, and as *Palladium* is written and tested, the development teams will be moved on to *Longhorn*. This will ensure that *Longhorn* is consistent with the *Palladium* concepts, and there should be no lack of logic between the two. It will also ensure that the transition from *Palladium* to *Longhorn* is relatively easy for users.

I hope this next bit won't shock you. I predict that in the first half of 2008, the production of new *Windows* viruses and Trojans will have ceased. There will be several new hazards by then. Wait and see.

The Big Unanswered Question

What will be the effect of *Longhorn* on *Linux*? Clearly we cannot answer this until we know a lot more about *Longhorn*, and until we see how *Linux* development is moving during the period following the 'Pandemonium' period mentioned above.

However, at the top of the hardware spectrum (mainframes), *IBM*, who have poured resources into developing the use of *Linux*, will defend vigorously. Since *Microsoft* may lack the weapons to attack, I believe the defence will prevail, and that *Longhorn* will not dent *Linux* in this area.

Lower down the hardware spectrum (big servers), *IBM* will find it more difficult to defend. They can defend the *IBM* big servers, yes, but what about the awkward customers, who have both *IBM* big servers and non-*IBM* big servers in the same operation? Finally, down at the bottom, the answer depends on how good *Linux* is then, and how easy it is to replace it with *Longhorn*.

Final Conclusions

We are in for a quiet 18 months, followed by a period of hardware and software chaos, the likes of which we have not seen for a while. During such periods, customers tend to make changes more slowly, and everything quietens down.

Although the Anti-Malware Industry is safe for five years, because *Windows* viruses and Trojans will keep coming, from 2008 onwards, it will be playing a new game, and will have to learn some new tricks!

[*VB would like to hear your views on Peter's predictions – email comments@virusbtn.com*]

PRODUCT REVIEW

Quick Heal X Gen

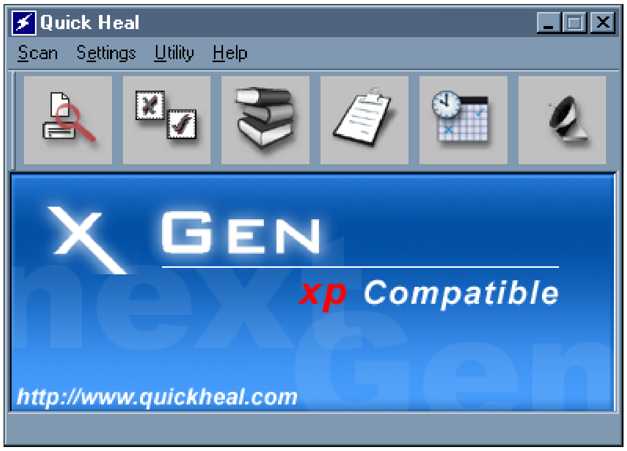
Matt Ham

Cat Computer Services' Quick Heal is a relatively recent addition to the selection of products included in *VB*'s comparative testing. However, *Quick Heal* is not a new product by any means, having been brought to market in 1993.

How, you may ask, has *Quick Heal* not received a wider press over the last nine years? The answer to this lies in the location of its market, this being primarily India. Indian software – admittedly much of it rebadged after having been commissioned elsewhere – is one of the great success stories of the last few years, so it is not surprising that pre-existing products are now being marketed to a worldwide audience. Anti-virus software developed for local markets can be prone to difficulties when faced by the US- and European-dominated WildList. However, this has not been apparent in *Quick Heal*'s performances so far – the product came close to obtaining a VB 100% award on its first attempt.

Quick Heal X Gen is the latest version of the software, which was released in September 2002. It is described as being compatible with *Windows XP*. This version can also be used on *Windows 95*, *98*, *ME*, *2000* and *NT*. Due to its recent release date, the full boxed contents were not available for review, and an electronic copy was tested instead. This consisted of the main program (a single file of 10 MB), DOC file documentation and the emergency disk files (an archive of 520 KB).

Also available from *Cat* is a DOS product, and a *Microsoft Exchange* version of the software is currently under development. Hidden away on the company's website are references to other *Cat* services – these covering web development, data recovery, system projects and GUI development. Judging by the amount of space devoted to these on the site, I would guess that they are not being pushed as a major source of employment for the *Cat* team.



Installation

The executable installer launches into a splash screen which warns that the program should not be installed if any anti-virus hardware or software is installed on the machine already. The option is given to abort the installation process at this point. Whether the mention of 'hardware' is a reference to such products as *Trend*'s *Chipaway* or more rigorous hardware methods is not obvious – the latter seems more likely. Choosing to continue the process brings up a display demonstrating the status of memory, MBR and system files scanning. When this is completed there is another warning, stating that no programs should be running in the background while installation is carried out.

Once this series of events is complete the obligatory assent to a licensing agreement is presented and personal data and a registration serial number are requested.

Next is the option to select where the application files are installed. Once the location has been selected, the files are installed speedily. The choices as to which portions of the program are enabled are determined within the programs themselves.

First to be launched is the email protection settings dialog. This provides a scanning functionality on *Outlook Express* or *Eudora*, if these are installed. *Outlook Express* was installed on the test machines. As well as a simple 'off or on' choice, there are settings for further fine-tuning of the actions of the mail scanner. Quite why these have a separate dialog is a little mystifying, since the only setting available is whether the user should be prompted if a virus is found. I imagine that these features will be expanded in the future.

Next to be launched are dialogs determining whether *Word* macro virus protection and shell extensions are to be installed – both are simple 'yes or no' options, with the macro protection option being greyed out when running on machines where *Office* is absent.

Further options present the choice of whether the on-access On-Line Protection and/or scheduler will be activated at each boot. On *Windows 95*, *98* and *ME*, it is also possible to produce rescue disks at this stage.

More options include the ability to configure the scheduler. This is close to the end of a process – a DOS box flashes up briefly before there is a reminder to register the product. After a reboot the program is ready to operate fully.

It is notable that *Quick Heal* displays both a DOS-style screen and a splash graphic on reboot as a matter of course.

Documentation and Web Resources

As mentioned previously, the documentation supplied was in DOC format – which could prove a potential embarrassment were an infected copy to be transferred via a third party. However, at this stage it is not clear in what format the documentation will appear on distribution CDs, if any.

The documentation consisted of a 70-page manual, covering the usual ground of features and marketing-related descriptions of the product. The manual was not used a great deal, thanks to the extensive help facilities within the program and the fact that most options were of an immediately obvious nature. The information within the manual is well laid out and on those occasions when information was required it was easy to find.

One area in which the manual scored more highly than the help feature was with respect to the mysterious black DOS screen which popped up and vanished speedily during boot. (This turned out to be the Quick Heal Sensor, which monitors programs flagged for execution at boot.) In other areas the help was not context-sensitive but was available from almost every screen encountered. It was also very different from the manual in both format and content – allowing for use of the two resources in parallel if required.

On first impressions the website appeared very similar to those of most anti-virus companies. The front page offers the usual links to virus descriptions, sales resources, product downloads and product descriptions. Amongst this usual fare there are some gems of oddity and hyperbole. The description of *Quick Heal* as ‘User Centric, Future Centric and Oh Yes! With a lot of Common Sense’ was high in the league of the former.

Somewhat hidden away on the website are support resources and an area in which product registration can be initiated.

The Interface

The installation procedure leaves ten programs installed in the *Quick Heal* program folder, in addition to the *Quick Heal Sensor*, which does not have an interface as such.

Quick Heal

When started, *Quick Heal*’s main program launches a splash screen, which looks rather outmoded in comparison with the general design scheme, and moves quickly onwards to a rather minimalist interface. In stark contrast with most products I have reviewed, this limits its initial interface to four drop-down menus and five large buttons. No information is displayed initially.

The first button, Scan Drive, offers just that, with drives selectable either in combination or as blanket categories of all floppy and/or hard drives. The second button is the somewhat more involved Options selector. This brings up a tabbed interface – the tabs are Scanner, Startup, On-Line Protection and Exclusions.

The scanner options are numerous. Under the general options sub-heading the default settings are sound disabled, with prompt on detection and create activity log enabled. Two alternatives are available for the course of action concerning unrepairable files: deletion or renaming, with renaming selected by default.

This is also the interface in which the objects to be scanned are chosen; the default is executable files. It is also possible to set the product to scan all files, or to opt for a user-specified file-type list. Initially this list looked reasonably comprehensive, though the status of blank extensions remained a mystery. In any case there was no facility to add blank extensioned files to this list.

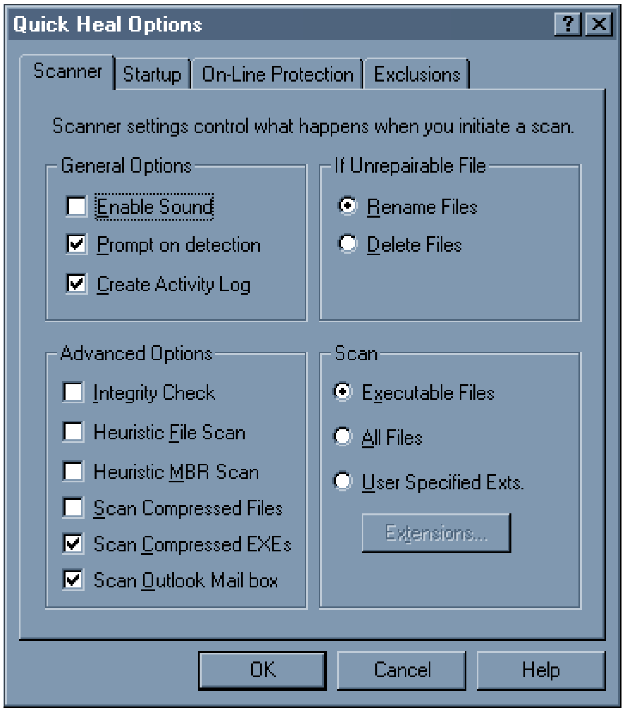
Last under the Scanner options tab are advanced scan options. By default, compressed exe files and the Outlook mail box are selected for scanning. Unselected by default were the integrity check, heuristic file scan, heuristic MBR scan and compressed file scanning.

The Startup tab is very much less involved in its contents. The automatic loading on boot, or otherwise, of On-Line Protection, Scheduler and Quick Heal Sensor are selected here. The initial settings of these features depend upon selections made during the installation process.

The third tab relates to On-Line Protection, the on-access portion of the product. As installed this is configured so it will scan files both on run/access and on download/create, though scanning can be deactivated for either set of actions. The initial settings here are set to executable files with the option of user-specified files; all files is not an option.

Floppies are considered separately. By default, on-access boot sector checking and scanning during shutdown are activated but can be deactivated. The selection of what action should be taken when a virus is detected presents the choice of whether a message should be displayed and the options of denial of access, repair or delete if impossible to repair, or automatic deletion.

The Exclusions tab deals with the exclusion of files. The use of the *.* wildcard allows exclusion of whole



Scanner settings control what happens when you initiate a scan.

General Options

☐ Enable Sound

☑ Prompt on detection

☑ Create Activity Log

If Unrepairable File

● Rename Files

○ Delete Files

Advanced Options

☐ Integrity Check

☐ Heuristic File Scan

☐ Heuristic MBR Scan

☐ Scan Compressed Files

☑ Scan Compressed EXEs

☑ Scan Outlook Mail box

Scan

● Executable Files

○ All Files

○ User Specified Exts.

directories, and this exclusion can be selected to be recursive or non-recursive.

The third of the buttons on the initial interface brings up the Virus DataBase. Information about any one of a lengthy list of viruses can be obtained here. First, the status of the virus' targets is given – MBR, file, script or macro. Secondly, the database states whether polymorphism, stealth, memory residency or worm/Trojan activity are a feature of the virus in question. Many uncommon viruses are included on the list, but more evident are those viruses which are in general circulation.

The fourth button is the Scheduler, which holds little in the way of surprises. Scheduled jobs may be created, edited and deleted. The unusual feature is that the results of scheduled jobs are logged here so that an administrator may know when, for example, boot-triggered scans were performed.

The final button on the interface provides a link to the Live Update application. This will be discussed later.

Drop-down menus on the interface offer Scan, Settings, Utility and Help. Scan offers a similar set of choices to that offered in the button-activated scan option – however, when using the drop-down menu, it is possible in addition to scanning drives to refine searches further, covering selected directories or files. Oddly, this menu includes the command to exit execution of *Quick Heal*.

From the Settings menu the main options dialog can be reached, as can the mail client protection options section and the scheduler. Although this is neither a very clear nor intuitive method of linking together the various options available in the program, it works well enough once *Quick Heal* has been used a few times.

The Utility drop-down menu covers a wider selection of options. Top of the list is Rescue Information. Under *Windows XP* this is non-operational, and when selected gives an error dialog stating that the Repair Disk Utility provided by the operating system should be used on *Windows NT*, *XP* and *2000*.

Second on the menu is Virus Database, which is a direct link to the virus database described above. The Activity Log offers a central clearing house for logs produced by all the *Quick Heal* applications. Thus logs from the shell extension, on-access scanner, mail scanner, startup scanner, Office scanner, scheduler and updates are all available here. One notable absence is a mention of on-demand scans.

The Quarantine interface makes its first appearance in the Utility drop-down menu. This allows the fate of quarantined items to be decided – either individually or en masse. Files in quarantine may be removed either by returning them to their original position, or by deletion. It is possible to add files to the quarantine manually in addition to automatic addition during scans.

Live Update is next on the Utility menu, and is a feature covered later in the review. This leaves the ability to delete integrity files. Files can be removed only from entire drives rather than being manipulated on a file or directory level. Since integrity checking is not mentioned within the rest of the dialogs available directly from the main *Quick Heal* program, it is rather an odd place to encounter this dialog.

The last drop-down menu offers a link to the main help program. However, that is not its sole function – there is also a printable listing of support addresses for *Quick Heal*, covering a range of countries, as well as a link to the website http://www.quickheal.com/.

On-Line Protection

On-Line Protection is activated automatically on boot if the default settings for installation are chosen, and its presence here simply allows it to be disabled or its options to be altered. However, these options can be altered through the main program described above. It is also possible to deactivate On-Line Protection and change its settings by right-clicking on its tray icon.

Live Update

At odds with other similarly named applications on the market, *Quick Heal*'s Live Update feature is confined to the manually triggered update process (in order to schedule this, the Schedule Updates feature must be used). First, this offers a small informational dialog and the choice of either Next or a settings button. It was not immediately apparent to what the settings button referred, so it was selected, producing further options of Internet Settings and a Live Update Scheduler.

Internet Settings is supplied in case the system uses a proxy to connect to the outside world. HTTP, SOCKS V4 and SOCKS V5 proxies are all supported. Ports can be selected and user information can be supplied if required for connection.

The Live Update Scheduler seems oddly placed here, since it is available before the live update options have been set up, and it might seem more logical to include it as part of that process or as an option through the scheduler itself. The default setting for the Live Update scheduler is a weekly update using the *Quick Heal Internet Center*. Selecting either the default or a user-designed schedule returns the interface to the Live Update entry point.

Choosing Next from this point initiates the process of setting download location and associated options. The *Quick Heal Internet Center* is the default source of update files. Alternative locations can be specified, though only if specific target files are being used can this location be browsed for, and persuading the Live Update program to use a specified directory seemed a futile task. It is also possible to flag all downloaded definition files to be backed up into a specified folder. After these choices have been made the program attempts to connect and the process is effectively complete.

MS Office Protection

MS Office protection requires *Office 2000* or *XP* to be installed. For the purposes of testing the more standard methods of scanning, this was not installed as a default.

Quick Heal Messenger

The Messenger acts as a central alerts and information vector for *Quick Heal*. By default it is launched at boot and remains active in the task bar. It can be set to check for new messages either once per day or at an interval of a number of hours. It is also possible to select sounds to be associated with the arrival of various message types and to adjust the settings by which the Messenger connects to the Internet.

When installed, messages are accepted from the *Quick Heal* servers and supply information on virus alerts, hoaxes and information of a more general nature such as successful installation and the availability of new update files.

User Manual and What's New

The User Manual option is a direct link to the PDF version of the manual. However, this is not a default download for the electronic version of the software and therefore I was faced with a broken link. The What's New document, on the other hand, is a text file and is present by default. This provides information on the latest changes to the software and associated news items.

Tests Performed

Since there were several features of the software which were not altogether clear, despite referring to the manual and program documentation, the first tests were designed to clarify behaviour in these cases.

The first test was designed to find out whether blank extensions are included for scanning, both with the default settings of the program and when using a user-defined extension list.

Since it has been a constant problem file in successive tests over the last few years, O97M/Tristate.C was chosen for these tests. This was detected easily enough both when the scanner was set to include all files and when the executable files only setting was selected. However, when the user-defined extensions option was chosen, the extensionless versions of this virus were missed. Since there appears to be no obvious way of inserting the blank extension into the list of files to be scanned, this seems to be a setting to be avoided.

It was not very clear as to which exact compressed file formats are supported for scanning. To check this a collection of EICAR test files, each compressed or archived in a different format, was used. With the default program settings only the standard EICAR.COM file was detected. Applying the 'scan inside compressed files' option resulted in detection of EICAR.COM within CAB, ARJ and ZIP files. With either setting, files in GZ, LZH, MME, RAR, TAR, UUE and XXE format were missed. AN SHS version of EICAR was also missed.

The on-demand scanner was investigated under three different settings: all files, user-selected extensions and the default of executables. The test sets used were those from the recent *NetWare* comparative review (see *VB*, August 2002, p.17). The software version used was that supplied for testing (dated 7 September 2002). Speed tests were performed not on the virus test sets but on the standard *VB* clean executable test set.

A scan using the default settings resulted in 14,530 detections out of 21,394 files scanned, with the misses being predominantly among the polymorphic test sets. Changing the settings to 'all files scanned' there were, once again, 14,530 detections but 21,407 files were scanned. Repeating the test with the unaltered base list of user-specified file extensions gave 14,516 detections out of 21,383 files scanned.

All of these figures were produced using the default settings for heuristics – that is, heuristics disabled. Returning the settings to scan executables another scan was performed, this time with file heuristics enabled. This scan showed the same 14,530 detections as before, with an additional 1,190 cases of files which were logged as likely to be infected. These figures are, however, irrelevant in some cases unless the time taken to perform a scan of clean sets is also considered.

With the full default settings, a scan of the clean test set took 48s, with five false positives encountered. The user-defined and all files settings were all but identical – no real surprise in a test set dedicated to obviously executable files. When heuristics were activated the scan completed in 62s with five false positives and 54 clean files throwing up heuristic warnings. Timings were also recorded of the time taken to scan the *Windows* directory. For the default settings this took 134s with no false positives. With the all files setting the scan took 175s, again with no false positives, and the user-defined extension setting took 130s. When the default settings were used with heuristics activated the time taken was 135s with no false positives.

Much as expected, the speed tests show that the ‘all files’ setting scans more files in a random set and takes somewhat longer to perform. However, in the tests performed this did not equate to a higher rate of detection. The testing of user-defined extensions reiterated its weakness as a selection. Heuristics proved to be slower on a set comprised entirely of executables, but much the same speed when applied to the *Windows* directory. This can probably be explained by the fact that *Windows* directory code is somewhat more innocent-looking in internal construction than the clean test set. However, the use of heuristics did trigger a very large number of false positives in the clean test sets.

Given this set of information the default settings seem reasonable enough. Under normal circumstances the user-defined extension lists supplied offer too little protection and the heuristics seem prone to cause false alarms. The ‘all files’ setting could offer an extra level of protection, though it is possible that this will add no files to scanning which will be detectable as executable, resulting in no additional detections.

Outlook scanning was the next to be tested. An email containing W95/CIH.1003 was sent easily from the machine with *Quick Heal* installed to another mail account served by *Microsoft Exchange Server*. However, the same message was detected as being infected when it was received by the protected machine. This was independent of the status of the on-access scanner. Since the file was also detected by the on-access scanner it could be argued that in a normal installation of the software the infected mail could not have been sent – but this rather relies upon having a standard installation as the only installation considered. The attachments were also flagged as being uncleanable and thus left in an infectious state – the alert produced being one advising the user to scan either the mailbox or the individual files involved. For many networked situations, scanning one’s own *Exchange* mailbox will be problematic and many users will simply be confused by instructions to rescan a file which has already been scanned.

An anomaly in scanning was noticed when on-access detection was triggered on a CD-ROM. The alert box produced stated that the file had been deleted – this was most certainly not the case. This ‘false deletion’ might cause problems in real-world situations.

Returning to *Outlook*, a few extra situations were considered. First to be considered were multiple attachments with different viruses present. This caused the scanner no problems in detection. This was followed with a test for identical attachments being in the same mail. This has recently been a problem situation in the real world, but did not cause any problems in this case. From the limited tests performed, the *Outlook* portion of *Quick Heal* offered reliable alerting of incoming viral mail – though to categorise it as protection would be overstating the case when this feature is considered alone.

Conclusion

Quick Heal offers a large number of features in a single package, though these features are not combined into one monolithic program. On the positive side, this potentially avoids confusion in that a user can concentrate directly upon those features they wish to use at that particular moment. This potential is rather under used however, since each portion of the program has features which link to other portions which are in some cases only tenuously related. The negative side of having such a fragmented program is that it is more likely to be confusing, which is certainly the case here. The interface offered by *Quick Heal* gives multiple methods of reaching the same dialog, while some can be reached only through non-intuitive routes.

The interface problem is exacerbated by *Quick Heal*’s fully featured nature. For a standalone product it offers a wide range of features, some of which – such as the central log file viewer – are pleasant in their implementation.

The detection rate is not at all bad, and is improving. However, the pros of detection and its single user features are battling currently with the cons of little capability for central administration and the confusing interface. Since an *Exchange* product is being designed at the moment, it is likely that the administration side of the product will improve. Whether the interface becomes more or less complex will be of definite interest as far as the product’s usability is concerned.

Technical Details:

Test environment: For in-lab tests, machines used in the test were identical 1.6 GHz Intel Pentium machines with 512 MB RAM, 20 GB dual hard disks, DVD/CD-ROM and 3.5-inch floppy drive.

Server software used: *Windows 2000 Server Service Pack 2* with *Exchange 2000 Server Service pack 2* and *Outlook 98*.

Client software used: *Windows XP Professional* with *Outlook Express*.

Test environment: For tests performed outside the secure labs the machine used was a 1600+ Athlon XP workstation with 512 MB RAM 20 GB dual hard disks, DVD/CD ROM and USB ADSL internet connection, running *Windows 98 SE.*

Prices: Prices start at US$35 for a single user licence including one year of support and upgrades.

Developer: *Cat Computer Services*, 58 Mangalwar Peth Patole Chambers, Maharashtra, Pune 411 011, India; tel +91 20 6119737; fax +91 20 6122173; email info@quickheal.com; website http://www.quickheal.com/.

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.

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, 6 Kimball Lane, Suite 400, Lynnfield, MA 01940, USA

Tel (781) 9731266, Fax (781) 9731267

END NOTES AND NEWS

The Fifth International Symposium on Recent Advances in Intrusion Detection takes place 16–18 October 2002 in Zurich, Switzerland. The RAID International Symposium series is intended to advance the field of intrusion detection by promoting the exchange of ideas on a broad range of topics. For programme details and registration information see http://www.raid-symposium.org/.

SANS Network Security takes place 18–25 October 2002 in Washington DC, USA. For details see http://www.sans.org/.

COMPSEC 2002 takes place 30 October – 1 November 2002 in Westminster, London, UK. More than 50 presentations and interactive workshops will be held in four streams, covering management concerns, infrastructure, law and ethics, technical issues and case studies. For more details, including the chance to read some of the abstracts and to register online, see http://www.compsec2002.com/.

A seminar entitled **E-business and security: New directions and successful strategy will be held 6 November 2002 at the Dali Universe, London, UK**. Graham Titterington, senior analyst at *Ovum*, will lead a half-day e-business security seminar organised in association with *Sophos* and *CipherTrust*. For full details and to register online see http://www.sophos.com/.

The CSI 29th Annual Computer Security Conference and Exhibition will be held 11–13 November 2002 in Chicago, IL, USA. The conference is aimed at anyone with responsibility for or interest in information and network security. For more information email csi@cmp.com or see http://www.gocsi.com/.

The 5th Anti-Virus Asia Researchers (AVAR) Conference takes place 21–22 November 2002 at the Ritz-Carlton, Seoul, Korea. Topics covered will include information on how the AV community works together globally, the latest virus and AV technologies, and reports on virus prevalence in various countries in Asia. The conference will be hosted by *Ahnlab, Inc*. For more information see http://www.aavar.org/.

Infosecurity 2002 conference and exhibition will be held 10–12 December 2002 at the Jacob K. Javits Center, New York, USA. For further details, including information on exhibiting and conference registration, see http://www.infosecurityevent.com/.

Papers and presentations are being accepted for the Black Hat Windows Security 2003 Briefings. Papers and requests to speak will be received and reviewed until 15 December 2002. The Briefings take place 26–27 February 2003 in Seattle, WA, USA. For details of how to submit a proposal see http://www.blackhat.com/.

***AV-Test.org* has published results of its recent Unix test of 21 anti-virus products** under *Linux* (*Suse* and *Red Hat*), *FreeBSD*, *OpenBSD* and *Solaris*/*Sparc*. The complete review can be read online at http://www.av-test.org/.

***Trend Micro Inc.* has joined the Nikkei Stock Average**. The 225-share Nikkei Stock Average is Japan's most widely followed stock market index, composed of leading companies listed on the first section of the Tokyo Stock Exchange. For more information see http://www.trendmicro.com/.

Network Associates Inc.* has completed its re-purchase of *McAfee.com Corp. Last month *NAI*'s final exchange offer for *McAfee.com* shares was accepted – six months after its initial offer – and the company bought 96% of *McAfee.com*'s outstanding shares, allowing it to execute a short-form merger under Delaware company law. 'The recombination of *Network Associates* and *McAfee.com* is an important evolution for customers and shareholders of both companies,' said Srivats Sampath, president and CEO of *McAfee.com*. '*McAfee.com*'s unique position as the pioneer and leader in online anti-virus and security services, combined with *Network Associates*' global presence, positions the integrated company to aggressively grow market share and build on the success we've had in the consumer and small and home office space.' *NAI* will pay $8 cash and 0.675 of an *NAI* share for every *McAfee.com* share. For more details on the merger and its implications see http://www.nai.com/.

Norman ASA* has launched *Norman Virus Control version 5.4, which includes an advanced version of its *SandBox* technology. The technology was introduced at the *Virus Bulletin* conference last year, and the *VB2002* programme includes a presentation on the next generation of *Norman*'s *SandBox* technology, which will be expanded to emulate networks inside the virus scan engine. See http://www.norman.no/, and for details of the presentation, including an abstract, see http://www.virusbtn.com/conference/.