ISSN 0956-9979 13666614 MAY 2002

# VIRUS BULLETIN

THE INTERNATIONAL PUBLICATION ON COMPUTER VIRUS PREVENTION, RECOGNITION AND REMOVAL



## *IN THIS ISSUE:*

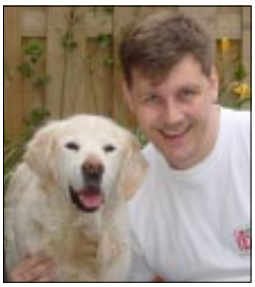
• **As complex as Euler's formula.** While some of virus writer The Mental Driller's previous creations have proved challenging to detect, W32/Simile shifts gears and moves another step up the scale of complexity. Frédéric Perriot, Peter Ferrie and Péter Ször start unravelling the code on p.4.

• **Reporting the facts:** David Ensign looks back on ten years of incident reporting at the US Department of Energy and argues the benefits of constant, real-time data collection combined with robust reporting and analysis. See p.13.

• **Shape-shifting software.** The disappearance of *Softwin*'s *AVX* coincided with the company's release of *BitDefender*. Matt Ham assesses the new incarnation on p.18.

## CONTENTS

# COMMENT



## Deciding Decisive Decisions

When thinking about what to write for this Comment piece, I realized that, whether it's making decisions about which AV product to choose for your corporate protection or decisions that influence your personal life, life is all about making decisions.

Last week Tommy, our 14.5-year-old golden retriever, had a seizure. My wife, Els, and I made an agreement a long time ago that we would not let Tommy suffer any pain, but suddenly we found ourselves in a position in which we might have to take a decision about what was best for Tommy, even though he wasn't in pain. The fact that our choice would be decisive for him frightened us. What if we made the wrong decision? Happily, Tommy survived and is almost fully recovered (a remarkable fact for a dog at such an age), which meant that, this time at least, we did not have to make that difficult decision.

*"Both the anti-virus industry itself, and its customers, must decide what it is worth to them to have and keep the WildList."*

Customers who have to select a product for their corporate anti-virus protection must face equally tough decisions – decisions which will influence at least the next year(s) of their corporate defence. But what are their selection criteria? Is it the GUI, additional features, management tools and possibilities, variety of supported platforms, etc.? One thing I know is an important criterion upon which these decisions are based is the product's detection rate of viruses that are on the WildList.

In March 2002, the WildList sent out to its reporters was accompanied by a letter explaining that, as a result of the financial situation of the WildList Organization, the decision had been made that the March 2002 WildList would be the last for the time being. Lack of funding had forced those who create the WildList to look for full-time paid jobs.

At the time of writing, the WildList Organization has at least found the financial means to create an April 2002 edition. If no further funding is forthcoming, the fate of the WildList will have to be decided. In the worst-case scenario, the WildList could be sold to a company within the anti-virus industry with low ethical standards. That would be devastating for the WildList and for the industry as a whole. But even if a company with very high ethical standards were to acquire the WildList, its continued objectiveness would still be questioned by that company's competitors.

A possible solution would be for all (major) AV companies to pay a part of the monthly costs involved in putting the WildList together. This might solve the issue of objectiveness from an industry point of view, but the outside world may see it differently. Besides, the economic situation at this time may prohibit companies contributing to this funding. Maybe it would be better if a non-AV company (with ties to the industry) acquired the WildList – for example one of the certification bodies. But again, its objectiveness might be questioned by other certification bodies.

Both the anti-virus industry itself, and its customers, must decide what it is worth to them to have and keep the WildList, while the WildList Organization must decide what would be the most ethical way to deal with the problem in case the necessary funding is not forthcoming. I feel certain that those behind the WildList will make the best decision they can, given the possibilities.

Another decision to be made, this time by *Virus Bulletin* (and by the many other reviewers and certification bodies) is, in the event that the WildList does not appear for several months, or not at all, what test criteria will replace the 100% detection of viruses In the Wild for the *Virus Bulletin* 100% Award? I sincerely hope this has never to be decided upon!

There are, of course, those decisions that are made 'out of habit'. Attending the *Virus Bulletin* Conference is something Els and I never have to make a decision on. It's a foregone conclusion: New Orleans, here we come! This time, however, without discussion, Els and I reached the same decision; Els will stay at home to take care of Tommy. If she does come to New Orleans it will mean we have had to make another difficult decision …

*Righard J. Zwienenberg, Norman, The Netherlands*

# NEWS

## Confusion Reigns

It is only ever a matter of a short time before any newcomer to the AV industry ponders 'Exactly what is the relationship between *McAfee.com* and *Network Associates Inc.*?' Attaining a clear and definitive answer rarely proves a straightforward task: *NAI* spun off *McAfee.com* three years ago and the two are, to all intents and purposes, separate companies. Now, however, *NAI* would like *McAfee.com* back – although they do already own 75 per cent of it.

Just when it seemed that the confusing situation was to be cleared up once and for all, with *NAI* making a share exchange offer for the 25 per cent stake of *McAfee.com* it does not already own, the company has had to drop its bid.

After an initial offer for the shares was rejected as being 'financially inadequate', *McAfee.com*'s board recommended, in early April, that stockholders accept an amended offer of 0.78 of a share of *NAI* common stock in exchange for each outstanding share of *McAfee.com* Class A common stock. However, the offer was rescinded on 25 April, resulting in share prices of both companies plummeting – *NAI*'s by 20 per cent and *McAfee.com*'s by more than 24 per cent. Recently discovered inaccuracies in *NAI*'s 1999 and 2000 financial statements, requiring restatement of results for those periods, have been cited as the reason for the withdrawal of the offer. (The company is already under investigation by the US Securities and Exchange Commission (SEC) for accounting issues in 2000, though the new findings are unrelated to the SEC's investigation.)

*NAI* chief executive George Samenuk is reported to have said, 'The *McAfee.com* offer is done right now. Done.' So it seems that the relationship between the two companies will continue to perplex AV novices for the time being at least ▮

## The WildList Saga

At a meeting of the Anti-Virus Product Developers' Consortium (AVPD) last month various members agreed to put forward funds to finance the production of May and June 2002 issues of the WildList. It is hoped that this will allow sufficient time for legal issues to be resolved within the organization and for a permanent solution to be found. *Virus Bulletin* is currently exploring the alternatives should there be no WildList for its forthcoming comparative reviews. As always, however, panic is not advised ▮

## VB Unveils …

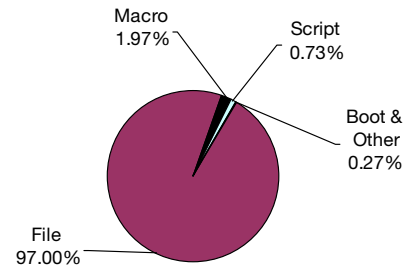
*VB* is pleased to announce the forthcoming release of its new-look website. A sleek new design accompanies new features and old favourites. Due to arrive late-May, keep your eyes peeled at http://www.virusbtn.com/ ▮

### Prevalence Table – March 2002

| Virus | Type | Incidents | Reports |
|---|---|---|---|
| Win32/SirCam | File | 2470 | 34.19% |
| Win32/Klez | File | 1321 | 18.28% |
| Win32/Magistr | File | 1038 | 14.37% |
| Win32/BadTrans | File | 796 | 11.02% |
| Win32/Fbound | File | 381 | 5.27% |
| Win32/Gibe | File | 376 | 5.20% |
| Win95/CIH | File | 160 | 2.21% |
| Win32/Hybris | File | 155 | 2.15% |
| Win32/Nimda | File | 75 | 1.04% |
| Win32/Mylife | File | 64 | 0.89% |
| Win32/MTX | File | 53 | 0.73% |
| Laroux | Macro | 45 | 0.62% |
| Haptime | Script | 33 | 0.46% |
| Win32/Aliz | File | 22 | 0.30% |
| Marker | Macro | 19 | 0.26% |
| Win32/Myparty | File | 18 | 0.25% |
| Win32/Gokar | File | 15 | 0.21% |
| VCX | Macro | 12 | 0.17% |
| Ethan | Macro | 11 | 0.15% |
| Win32/QAZ | File | 9 | 0.12% |
| LoveLetter | Script | 8 | 0.11% |
| Tristate | Macro | 8 | 0.11% |
| Form | Boot | 7 | 0.10% |
| Kak | Script | 7 | 0.10% |
| Win32/Goner | File | 7 | 0.10% |
| Win32/GOP | File | 7 | 0.10% |
| Win32/Navidad | File | 7 | 0.10% |
| Win32/Ska | File | 7 | 0.10% |
| Others [1] | | 94 | 1.3% |
| Total | | 7225 | 100% |

[1] The Prevalence Table includes a total of 94 reports across 50 further viruses. Readers are reminded that a complete listing is posted at http://www.virusbtn.com/Prevalence/.

Distribution of virus types in reports



# VIRUS ANALYSIS

## Striking Similarities

*Frédéric Perriot, Peter Ferrie and Péter Ször*
*Symantec Security Response, USA*

W32/Simile is the latest 'product' of the developments in metamorphic virus code. The virus was released in the most recent 29A #6 issue in early March 2002.

The virus was written by the virus writer who calls himself 'The Mental Driller'. Some of his previous viruses, such as W95/Drill (which used the Tuareg polymorphic engine), have proved very challenging to detect.

W32/Simile moves yet another step up the scale of complexity. The source code of the virus is approximately 14,000 lines of assembly code. About 90% of the virus code is taken up by the metamorphic engine itself, which is extremely powerful.

The virus was named 'MetaPHOR' by its author, which stands for 'Metamorphic Permutating High-Obfuscating Reassembler'.

The first generation virus code is about 32 KB and there are three known variants of the virus in circulation. Samples of the original variant which was released in the 29A issue have been received by certain anti-virus companies from some major corporations in Spain, indicating a minor outbreak.

W32/Simile is highly obfuscated and challenging to understand. The virus attacks disassembling, debugging and emulation techniques, as well as standard evaluation-based techniques for virus analysis. In common with many other complex viruses, Simile uses EPO techniques.

### Replication Routine

Simile contains a fairly basic direct action replication mechanism that attacks PE files on the local machine and the network. The emphasis is clearly on the metamorphic engine, which is unusually complex.

### EPO Mechanism

The virus searches and replaces all of the possible patterns of certain call instructions (those that reference ExitProcess() API calls) to point to the beginning of the virus code. Thus the main entry point of the file is not altered.

Sometimes the metamorphic virus body is placed together with a polymorphic decryptor at the same location within the file. In other cases the polymorphic decryptor is placed at the end of the code section, while the virus body is placed in another section. This is to conceal further the location of the virus body.

### Polymorphic Decryptor

During the execution of an infected program, when the instruction flow reaches one of the hooks that the virus has placed in the code section, control is transferred to a polymorphic decryptor which is responsible for decoding the virus body (or simply copying it directly since, intentionally, the virus body is not always encrypted.)

This decryptor, whose location in the file is variable, allocates a large chunk of memory (about 3.5 megabytes) then proceeds to decipher the encrypted body into it. It does this in a most unusual manner: rather than going through the encrypted data linearly, it processes it in a seemingly random order, thus managing to avoid triggering some decryption-loop recognition heuristics.

This 'Pseudo-Random Index Decryption', as the virus writer calls it, relies on the use of a family of functions that have interesting arithmetic properties, modulo 2^n.

While the virus writer discovered this by a process of trial and error, it is possible to produce a mathematical proof that his algorithm works in all cases (provided the implementation is correct, of course). Such a proof is beyond the scope of this article but the proof, by Frédéric Perriot, is available at http://www.peterszor.com/.

The size and appearance of the decryptor varies greatly from one virus sample to the next. To achieve this high level of variability, the virus writer simply generates a code template and then puts his metamorphic engine to work to transform the template into a working decryptor!

In some cases, the decryptor may start with a header whose intent is not immediately obvious upon reading it. Further study reveals that its purpose is to generate anti-emulation code on the fly: the virus constructs a small oligomorphic code snippet containing the instruction RDTSC ('ReaD Time Stamp Counter'). This retrieves the current value of an internal processor ticks counter. Then, based on one random bit of this value, the decryptor either decodes and executes the virus body or bypasses the decryption logic altogether and simply exits.

Besides confusing emulators that do not support the somewhat peculiar RDTSC instruction (one of The Mental Driller's favourites, which he used previously in W95/Drill), this is also a very strong attack against all algorithms that rely on emulation either to decrypt the virus body or to determine viral behaviour heuristically. Effectively, it causes some virus samples to cease infecting completely upon a random time condition.

On initial execution, the virus body will retrieve the addresses of 20 APIs that it requires for replication and for displaying the payload.

Next the virus will check the system date in order to determine whether either of its payloads should activate. Both payloads require that the host imports functions from User32.dll. In this case, the virus checks whether it should call the payload routine or not (which is explained below).

### Metamorphism

After the payload check has completed, a new virus body is generated. This code generation is carried out in a number of steps:

The first step is to disassemble the viral code into an intermediate form, which is independent of the CPU upon which the native code will execute. This allows for future extensions, such as producing code for different operating systems or even different CPUs.

The second step is to shrink the intermediate form, by removing the redundant and unused instructions. These instructions were added by earlier replications to interfere with disassembly by virus researchers.

The third step is to permutate the intermediate form, for example reordering subroutines, or separating blocks of code and linking them with jump instructions.

The fourth step is to expand the code, by adding redundant and unused instructions.

The fifth step is to reassemble the intermediate form into a final native form that will be added to infected files.

Thus Simile can not only expand, as most first generation metamorphic viruses do, but it can also shrink (and shrink to different forms!).

### Replication

Next the replication phase begins. It starts by searching for *.exe in the current directory, then on all fixed and mapped network drives.

The infection will scan recursively into directories, but only to a depth of three subdirectories, and avoiding completely any directory that begins with the letter 'W'.

For each file that is found, there is a 50% chance that it will be skipped explicitly. Additionally, files will be skipped if they begin with 'F-', 'PA', 'SC', 'DR', 'NO', or contain the letter 'V' anywhere in the name.

Due to the nature of the comparison, other character combinations are skipped unintentionally, for example any directory that begins with the number 7, any file that begins with 'FM', or any file that contains the number 6 anywhere in its name.

The file infection routine contains many checks to filter files that cannot be infected safely. For example, the file must contain a checksum, it must be an executable for the Intel 386+ platform, and there must exist sections whose names are '.text' or 'CODE', and '.data' or 'DATA'. The virus also checks that the host imports some kernel functions, such as 'ExitProcess'.

For any file that is considered infectable, random factors and the file structure will determine where the virus places its decryptor and virus body.

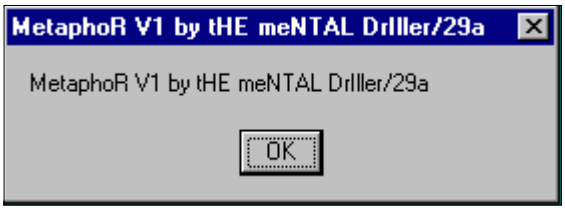
If the file contains no relocations, or with only a small chance, the virus body will be appended to the last section in the file. In this case, the decryptor will be placed either immediately before the virus body, or at the end of the code section.

Otherwise, if the name of the last section is '.reloc', the virus will insert itself at the beginning of the data section and move all of the following data and update all of the offsets in the file.

### Payload

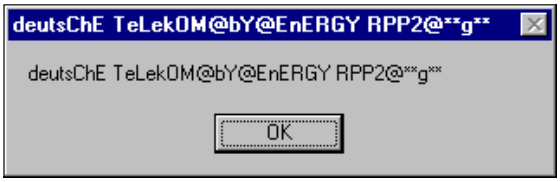
The first payload activates only during the months of March, June, September, and December. Variants A and B of W32/Simile display their message on the 17th day of these months. Variant C will display its message on the 18th day of these months.

Variant A will display the message 'Metaphor v1 by The Mental Driller/29A':




and variant B will display 'Metaphor 1b by The Mental Driller/29A':


Variant C attempts to display 'Deutsche Telekom by Energy 2002 **g**':



However the author of variant C had little understanding of the code, and the message rarely appears correctly. In all variants, the message appears in randomly mixed letter cases.

The second payload activates on 14 May in variants A and B, and on 14 July in variant C.

In the second payload, variants A and B will display the message 'Free Palestine!' on computers that use the Hebrew locale. Variant C attempts to display the text 'Heavy Good Code!' but, due to a bug in the virus code, this message is displayed only on systems on which the locale cannot be determined.

**Conclusion**

During the extensive and detailed tests carried out with W32/Simile replication on test systems we have noticed that the virus code generates garbage unintentionally or trashes some files accidentally as the direct result of its extreme complexity.

It seems that obfuscated code is not only challenging for virus researchers to analyse, but it is very challenging for the author of the code to debug.

As the saying goes: 'there are three kinds of lies: lies, damn lies, and statistics'. The complex infection mechanism coupled with the powerful metamorphic engine make it very difficult to reach 100% accuracy using only empirical evaluation methods, and indepth analysis of the virus code is essential.

Exact identification becomes a problem even for humans. How long does it take to be sure if something is really variant A or C or a new one? Is it modified or is it the same? It is becoming more difficult to know. The need to understand metamorphic code in a quicker fashion must be the subject of further research.

As this issue of *VB* goes to print, W32/Simile (aka W32/Etap) appears on the preliminary April 2002 supplemental WildList.

## W32/Simile

| | |
|---|---|
| Alias: | W32.Etap, Metaphor. |
| Type: | Direct action Win32, portable executable infector, complete metamorphic virus. |
| Removal: | Detect and delete infected files and replace them from clean backups. |
| Payload: | Displays messages on certain dates. |
| Unintended payload: | Trashes some portable executable files. |

# FEATURE 1

## Testing Behaviour-Based AV Products

*Lixin Lu*
*InDefense Inc., USA*

There are significant differences between behaviour-based and signature-based anti-virus products. The aim of this article is to answer some questions relating to anti-virus product testing, emphasizing the differences between the two types of product. Tests include the basic functionality test, the false positive test, and performance and compatibility tests.

Viral simulation test cases are important for testing behaviour-based products. These must include the simulation of viral behaviour that does not exist in current viruses but which may be a threat in the future.

The article will describe how to produce viral simulation test cases as well as the use of basic tools to simulate viral activity, thus presenting efficient methods for QA engineers and product evaluators to test behaviour-based products without the use of live viruses.

**Introduction**

Although the term 'behaviour blocking' is not new to the AV industry, few commercial behaviour-blocking products exist at the present time.

Viruses have evolved into fast-spreading Internet worms and hybrids, leaving insufficient time for AV scanning products to produce and distribute signature updates before significant damage can occur.

Customers are looking for an alternative solution to the signature update cycle and, as a result, behaviour-blocking technology has been receiving an increasing amount of attention.

It would be easy to draw up an extensive list of AV products whose testing indicates that they detect a huge number of actual viruses or '99.99 per cent of all viruses'. These are signature-based products. However, it is difficult to find any published information regarding products that are purely behaviour-based.

Part of the reason for this lack of information is that there is no standard method for testing behaviour-based products. How do you test how well behaviour-based products block new or unknown viruses? How do you evaluate behaviour-based products except by catching existing viruses? Which functionality is the most important to consider while conducting testing? How can the consumer select and verify behaviour-based products without using live viruses or industry-specific tools?

### Computer Viruses and Viral Behaviour Blocking

A computer virus is a piece of executable code designed to spread by replicating. If the malicious code doesn't replicate, it's not a computer virus. Many computer viruses carry some sort of payload, however, payloads cannot be used to define a program as viral. In order to replicate, a virus must execute. The following describes the main properties of virus replication.

File viruses replicate by appending or inserting code into a host program, then modifying the program's entry point to execute the viral code. Whenever the infected host program runs, the viral code runs, potentially infecting other host programs. Executable files usually contain many entry points, such as the main program entry point, main exit entry point, section entry point, sub functional entry points, and so on. Any of the file's entry points can be targeted by a file virus.

Macro viruses infect *Microsoft Office* storage files. Macro viruses can replicate by inserting viral code into the *Office* application's global template. When a new *Microsoft Office* file is created or an existing file is opened, the viral code runs and infects the newly created/opened file.

Boot sector viruses replicate by inserting viral code into the boot sector. When a computer is booted from an infected floppy, the virus runs and infects the hard drive. Boot viruses propagate slowly, through floppies.

Worms replicate through email or Internet channels. Worm programs can be macros, scripts, or standalone executables. To run, they may drop executables into the automatic start up folder, change the system registry to execute dropped files, or replace a system file that always executes. Worms propagate at Internet speed. They can infect thousands of computers in just a few hours.

The fundamental goal of behaviour-based anti-virus products is to prevent all types of malicious program from infecting or replicating. From an architectural perspective, the design of behaviour-based products should be based around preventing the core functionality of viruses, i.e. infection, replication, and spreading. If an anti-virus product can block all of these actions, it will indeed be a powerful tool.

### Test Strategy for Behaviour-Based Products

The testing or evaluation of behaviour-based products should focus primarily on three areas. The first of these is functionality, i.e. the tests must verify that the product can prevent viruses from infecting and replicating.

Viruses append or insert code into a host program. In order to avoid detection, many viruses maintain the infected program's normal functionality and attempt to spread before executing a payload. The virus has a greater chance of being able to spread itself if it can avoid detection, and executing a payload or altering an infected program's functionality are both actions that will raise suspicion and increase the chances of detection.

Some viral programs create a back door or carry out a DDoS attack and carry no payload. Therefore, preventing viral programs from modifying existing files and replicating must be the core functionality for behaviour-blocking products.

The second focus of testing should be on false positives. Although it is not difficult to build behaviour-blocking software, it is not easy to create a behaviour-based product that can achieve an acceptable level of false positives.

A product that blocks all activity will certainly block viruses. However, this is not what users need. Distinguishing viral behaviour from similar functions of legitimate programs is the most challenging part of creating a behaviour-blocking product. Testing should determine whether the product maintains an acceptable level of false positives.

The third area is performance and compatibility. In order to prevent viral programs from replicating, behaviour-based products sometimes interfere with the system kernel. Such interference can affect system performance. Therefore, it is important to test the performance of a behaviour-based product. Compatibility testing is another important issue, since behaviour-based products may affect the ability of other software programs to perform legitimate tasks.

Other aspects of the product, such as the user interface, configuration, installation, etc. should be assessed too, but these are not unique to behaviour-blocking products.

The following test cases illustrate how testing can be carried out in the three main areas without using any special tools or live virus samples.

### Basic Functionality Test

Basic functionality testing should focus on ensuring that the behaviour-blocking product detects the actions that viruses exhibit and identifies the process attempting those actions. Different tests must be created for each type of virus.

Most file infector viruses change a host program's entry point and insert or append viral code into the program file. The simplest way to test whether a product stops this type of virus is to use a file editor, such as Notepad (binary editor is preferable but not essential), to modify a test executable file.

For example, use Notepad to open a copy of CALC.EXE. The first two bytes must be 'MZ'. This is the EXE file signature. All EXE files on the *Windows* platform start with 'MZ'. The content after the initial MZ is usually unreadable. To modify the file, insert a couple of blank spaces after MZ. This will change the program's entry point. Save the file to a different name and exit the editor. Attempting to run the modified program will produce an error message,

such as 'Program too big to fit in memory'. This happens because the program's entry point has changed and now points somewhere it shouldn't.

A behaviour-based product should prevent the editor from making the change, protecting the file from modification. If the behaviour-based program fails to block the modification, it will fail to stop viruses that change a host program's entry point. If the behaviour-based product protects the file and identifies the text editor as the modifying program, it should be able to identify and block real viruses that spread this way.

Some behaviour-based products monitor only a certain class of executable, such as executable files downloaded from the Internet or attached to email. Those products provide only partial protection for the system, blocking the main channels that Trojans and worms use to spread. When testing this type of product, send the text editor attached to an email to an address on the system, and then use it to change another executable file.

For macro viruses, *MS Word* can be used to perform the test. Open *MS Word* and use the Visual Basic Editor tool to insert the following in ThisDocument under Microsoft Word Objects:

```
Sub AutoOpen()
    MsgBox 'Hello world!'
End Sub
```

This is a simple automatic macro that activates when the file is opened. If a behaviour-based product doesn't prevent you from inserting automatic macros into an *MS Word* file, it won't stop a certain class of macro viruses. Real macro viruses are more complicated than this, but the basic action of inserting a macro into the global template file or an opened *Office* file is the same [*Ed – it would be quite straightforward to simulate more complicated macro infections programmatically*].

Most modern Trojans and worms spread by email. In order to test malicious email programs, create a simple script program, macro or batch file that sends emails with executable attachments. The behaviour-based product should not allow the email attachments to be sent.

Using an email application to send or receive an email with an attached executable program or *Word* file containing a macro will also work. A behaviour-based product should block this type of activity.

Trojans and worms may drop or modify executables on the infected computer. A simple method of testing for this behaviour is to copy or move one or two executables from one folder to another, especially to the auto start up folder. Renaming executable files should also be tested. These tests can be carried out using the command line or Windows Explorer. If the behaviour-based product has process certification functionality, the tested process should not be certified.

Another behaviour of malicious programs is the modification of certain registry keys. Those most commonly modified are the Run, RunOnce, and RunOnceEx keys in the HKLM\ or HKCU\ hives. Behaviour blocking products should identify and prevent any changes to sensitive registry keys. This can be tested by using the Registry Editor to insert or modify a string value at one of these keys. A behaviour-based product that blocks these modifications will stop most malicious programs that install themselves by making changes in the registry. Other keys, such as RunService, or keys that point to a specific folder, such as the Startup folder, or to a certain file, such as the global template, are also common targets for viral programs. These keys are also valid for testing.

Detection of boot sector modifications can be tested by making changes to the CMOS settings, however, this type of testing is not recommended outside of QA labs.

For a QA engineer or product testing organization with the relevant testing tools available, more comprehensive testing may be carried out. The key point is to establish a standard that lists and tests all viral behaviour. However, most viral behaviour can be simulated, eliminating the need for live virus samples or special tools.

### False Positive Test

The usefulness of a behaviour-based AV product is reduced with its frequency of false positive alerting. Therefore, it is very important to determine the level of false positive alerting in such a product. Most false positives are generated in three areas; modifying executable files, changing system registry settings and accessing email or Internet ports.

A behaviour-based product that alerts on all modifications made to executable files will generate many false positives. File viruses make changes to host program files for two reasons; to insert/append viral code into the program file or to modify an entry point so that the inserted code executes when the infected program runs.

In Portable Executable (PE) files, the main entry point is the AddressOfEntryPoint in the PE header. File viruses may change this field so that it points to the viral code and consequently, whenever the infected program runs, the viral code will be run first. Another interesting field is SizeOfImage in the PE header. If a virus inserts or appends its viral code into the file, this field must be adjusted accordingly. Other fields that can be used by viruses include the relocation table, section head, alignment space, exit entry point, and so on. All of these are sensitive areas within PE program files and any unexpected change in any of these areas is considered suspicious behaviour and should be monitored.

If, on the other hand, a modification occurs in a non-sensitive area of the file – such as a data field in the binary code for keeping track of time or licensing reasons – this is

not considered suspicious behaviour. False positives result if the product alerts on modifications to non-sensitive fields.

In order to test for false positive alerting on executable file modifications, use a program that modifies non-sensitive areas of the file (for example older DOS games, which modify the game executable each time the game is run in order to store score information or time). Low-level utilities, such as Scandisk or Defrag, alter executables in a non-sensitive way. Also, compression utilities, such as Winzip or PKZip, can be used to create or modify self-extracting executables.

Modification of system registry keys can also produce false positives. Behaviour-based products should monitor only those registry keys which are considered 'sensitive'. This should be combined with some other measurement to reduce false positives. Software installation and upgrade programs commonly modify sensitive registry keys. Running an installation or upgrade program known to modify sensitive registry keys is a way to test for registry modification false positives.

As mentioned previously, many recent malicious programs have used email to spread, and the most effective method to block malicious programs distributed in this way is to inspect emails and prevent executable code within or attached to emails from passing through to the user.

False positives may result from misidentification of executable files, Internet port access, or from other applications using the Internet communication resources. If the product does not contain additional filters for allowing normal Internet access, it will generate false positive alerts. Running on-line programs, Internet browsers or email applications can test this.

### Performance and Compatibility Test

Any behaviour-blocking software will cause some level of degradation of system performance. The key is to develop the software to provide complete virus protection while maintaining an acceptable level of system performance and compatibility with other applications.

Behaviour-based programs use resource-intensive real-time components that monitor system activity. The effects of the product on system performance can be tested by using a file search utility, such as one that searches for a particular string inside all files on the system. This will access all files and quickly generate lots of file-open, file-read, and file-close activity. Scanning programs and backup utilities will also generate lots of file activity. Running searches or scans with the behaviour-based program enabled and then doing the same with the program disabled will allow a comparison of the performance degradation caused by the behaviour-based product.

Other system applications, such as backup utilities, network applications, disk management applications, and file management software are good testing programs. These test product performance and compatibility.

Compatibility problems usually occur with low-level utility programs, such as hard disk defragmenting programs, backup programs, installation programs, etc. These can be good tools for testing product compatibility.

### Summary

Behaviour-based products are designed to detect viruses by monitoring system activity and detecting viral behaviour. If no viral program runs, there is no viral behaviour to detect or block. This is a basic difference between AV scanners and behaviour-based products – the behaviour-based product cannot detect a dormant virus residing within a file on the hard drive.

However, testing a behaviour-blocking product does not necessarily require live virus samples to be executed. Most viral behaviours, such as changing the registry, modifying executable programs, and so on, can be reproduced or simulated.

In fact, running tests using live viruses only will be insufficient since this limits the tests to current viruses only. Behaviour-blockers must protect against unknown viruses, which may exhibit behaviour not exhibited by current virus samples.

For example, many current viruses change the registry Run key to run a dropped file automatically. A future virus may change another key to achieve the same result. The designer of the behaviour-based product must anticipate the possible vulnerabilities of the future in order to prevent new viruses from capitalizing on them.

Behaviour-based product testing must include testing for false positives, performance and compatibility. From an engineering perspective, it costs far more to reduce the level of false positives, minimize the impact on system performance, and maintain high compatibility than it does to build the behaviour-blocking engine itself. Omitting the testing of these factors will lead to incomplete and misleading results.

Finally, behaviour-blocking products cannot identify viruses by name. Behaviour-based products make no attempt to determine or reveal the name or type of virus or malicious program detected. It is only viral *behaviour* that is detected and the offending process is identified. All viral programs displaying similar behaviour, including those that are unknown to AV scanners, are detected.

Other aspects of behaviour-based products, such as the GUI design, usability, online help, etc. are important and should be evaluated along with functionality. However, the tests that have been covered here are those that are most important and unique to behaviour-based products. As more and more attention is drawn towards behaviour blocking, it becomes more important to establish a standard for testing and evaluating this type of product.

# FEATURE 2

## Rescue Me: Updating Anti-Virus Rescue Systems

*Andreas Marx, AV-Test.org*
*University of Magdeburg, Germany*

The problem is an old one: if the PC of a home user becomes infected by a virus, the user is advised to boot the machine from a 'known good' virus-free disk to scan and disinfect the local hard drives.

In times of DOS, *Windows 95*, *98* and *ME* this presented only a minor problem – most virus scanners included bootable disks in their retail package, otherwise the user was able to create them during or after installation. Another solution was a bootable installation CD-ROM, which has the benefit of being write-protected and therefore safe from viruses. The FAT16 and FAT32 platforms were well known and caused no great problems, if file or even boot viruses had to be disinfected.

### The Problem: Windows XP

The situation has changed with the arrival of *Windows XP*: NTFS drives have become common as primary file systems for home users. And that's the problem – the majority of virus scanners are unable to scan NTFS drives, if started from their rescue disks or CD-ROMs (despite the fact that the manufacturers of these scanners claimed that their products were completely ready for and compatible with *Windows XP*).

Currently I know of only a few working solutions that can be started directly from the bootable CD. The first is *AntiVirusKit* by *G Data*, which uses a *Linux* version of *Kaspersky Anti-Virus* and is included in the retail product. *Kaspersky Labs* has its own solution, however this is no longer included in the retail product; it is only sold separately. The third is *AntiVir* by *H+BEDV Datentechnik*, which uses *Linux*. The rescue system is included in the retail product as well as on all demonstration CDs of *AntiVir Personal Edition*. The same product is also distributed under the name *Vexira Antivirus* by *Central Command* in the US. According to Igor Muttik, *Network Associates Inc.* has developed a similar rescue system on floppy disks which is based on DOS. However, it is not available due to the high licensing cost of the third-party drivers and tools used.

As far as I am aware, all other anti-virus programs (regardless of whether they are started from the CD or from included or creatable boot disks) will completely ignore and skip NTFS drives. Some programs indicate that NTFS drives have been found, but not scanned, while other programs simply report after a few seconds that they have scanned all available drives and they are confirmed clean – although, in fact, nothing has been scanned.

Under *Windows XP*, there is a feature called Recovery Console, which (if installed) can be started at boot-up by pressing F8. After a log-in using the administrator account and password, the user can access all data on NTFS drives. Command-line operations like 'copy' or 'ren' will work, but no external programs can be started. Therefore, it can be a great help, but only if the problems are known.

### Computer Magazine Solutions

In response to a large number of requests from their readers, two German computer magazines have published articles on work-arounds for this problem.

An article in the technical *c't* magazine (*c't* 25/2001, p.250) included a manual instruction guide on how a rescue CD-ROM can be created, not only for use in the case of virus infections, but also for the recovery of accidentally deleted data etc.

Their solution was largely *Linux*-based, because *Linux* has built-in NTFS drivers (besides drivers for various other systems, like FAT16/32, HPFS, its own ext2/3, ReiserFS and so on), which are reliable for read operations. Write operations, however, are dangerous according to the author of the NTFS drivers and according to our own tests. The main reason for this is that very few of the facts about how NTFS works are public and the authors had to reverse-engineer a lot of these complex internal structures.

However, *Linux* is not needed at all – a DOS NTFS driver is available from *Sysinternals* (http://www.sysinternals.com/), which works quite well, even if it consumes a lot of memory. Write operations are not permitted in the freeware version; the registered version costs US$49.

The more end-user-focused *PC-WELT* has published a completely ready solution on their bootable cover CD-ROM, which is based on a *Linux* beta version of *F-Prot* (*PC-WELT* 4/2002, p.154). According to the authors, it took only a few hours to prepare the solution, because it is based mainly on the *Linux* rescue system Rip-45 (see http://www.ibiblio.org/pub/Linux/system/recovery/) and the Live System Knopper (http://www.knopper.net/). The bulk of the work went into creating a menu system, from which the user can select what he or she wants to do.

If two magazines have already published a working solution, it should not be a great deal of trouble for AV companies to do the same in order to provide their customers with better protection. A few other points should be taken into consideration, such as updates of the signature files (CDs are usually old) or the ability to create and save log files.

The following are the most important issues that must be dealt with in order to achieve a working solution. The facts are not limited to NTFS drives or *Windows XP*, but generic and useable for every platform. Furthermore, these features are useful not only for home user systems, but also for system administrators as well as computer retail stores, for example, if they are given a computer for further analysis.

### Linux, BSD or DOS?

One of the first questions that has to be answered is which platform should be used.

*Linux* is available free of charge, but solutions that are based on *Linux* must be published as source code, according to the GNU General Public License (see http://www.fsf.org/licenses/gpl.txt). Of course, the source code of the virus scanner does not need to be published, but all kernel and script modifications that are based on GPL-protected program code must be.

An advantage of using *Linux* is its easy implementation, because the system can be configured as needed and the 32-bit program code of the scanner and helper programs can be run without any memory management problems.

*FreeBSD*, *OpenBSD* and *NetBSD* include essentially the same (optional) NTFS driver as *Linux*, but the *Linux* version is updated more frequently than the BSD port. However, these operating systems do not require any source code to be published, even after changes have been made (see http://www.freebsd.org/copyright/). [In the rest of this article, I shall not differentiate between *Linux* and *BSD*, but use *Linux* as a synonym for all of the open source operating systems mentioned above.]

There are two main possibilities for using a DOS platform. MS-DOS is shipped with *Windows 98*, where licence fees must be paid, but is the most compatible platform for other programs. FreeDOS could be used as an alternative to MS-DOS (http://www.freedos.org). This can be used free of charge under GPL terms, just like *Linux*. However, FreeDOS is only 99 per cent compatible with other DOS applications.

DOS will require a few additional drivers for the CD-ROM (if needed) and SCSI drives, as well as for the memory management. The scanner itself is likely to require a DOS extender – as well, of course, as a tool such as NTFSDOS, to make NTFS drives accessible. It might be a good idea to include a few tools such as 'Fdisk', 'Sys' and 'Format'. Free file managers, like the Midnight Commander for *Linux* or the Volkov Commander for DOS can also be useful for later manual rescue operations.

### Start-up Process

The boot process is fairly similar for these platforms. For *Linux* systems, the compiled monolithic compressed kernel must be started, which includes all the necessary drivers and, for DOS, the usual configuration and start-up files config.sys and autoexec.bat will run. This can be done from a simple 1.44MB boot disk or a bootable CD-ROM, which uses almost the same method: the BIOS will simply load and start a disk image which is stored in a special area at the very beginning of the CD-ROM.

Of course, the user has to change the boot order first, so that the A: drive or the CD-ROM is used before the hard disk in the boot sequence. The rescue system should notify the user that these changes have to be undone after a successful scan or repair session to prevent boot virus infections.

After loading all the required drivers, the scan process should not start automatically. Instead the user should be prompted with a simple menu, with options such as 'scan all hard drives', 'scan selected drives', 'scan floppy disk', 'exit to operating system', 'run a special command or program', and so on.

The ability to test the complete hard disk for read errors (simply try to read everything, sector by sector) would be a useful feature. In many cases the root of the problem is not a virus, but hardware – and in particular hard disk – errors. Of course, a help window or help function with short instructions should be included as well.

First, however, the user should be able to update the virus scanner, because the CD-ROM or the rescue disks that have been created will usually be quite old. For this, it should be possible to update the signature files from an external drive, such as a floppy disk – alternatively the signature databases stored on the local hard disk should be used, if they are valid.

The process is more complex if engine or scanner updates are needed, because such files are more likely to be infected by a virus and are highly platform-dependent (a *Windows* DLL won't run easily under DOS or *Linux*), therefore only trustworthy sources should be used. However, this should not be a problem if a scanner uses an 'integrated' solution, where all data (both engine and signatures) are stored in one or more encrypted, digitally signed file(s).

There is one main limitation: only 1.44 MB of data can be stored on a disk. All scanner files should be able to fit on a disk, which means that none of the files should be larger than 1.44 MB. Due to the size of most scanner databases, it's likely that more than one disk will be needed.

It may be possible to download the required updates from the local network or even the Internet or other kinds of dial-up or DSL connections, after loading all the necessary network and TCP/IP drivers etc., but that is likely to be a prohibitively complex task.

### Scan Selected Directories

The scan process itself can raise a few problems, starting with the drive selection, especially under *Linux*. Here, the common drive letters such as 'A:' or 'C:' are not available,

instead names like '/dev/fd0' or '/dev/hda1' are used. For user convenience, it should be possible to display the *Windows* drives convention as well. This should be quite an easy task provided these mappings have not been changed manually using the *Windows* Drive Management tools. It may also be useful to display a few further details of the drive, such as the label, file system type and its size. The rest of the scan process is a simple run of the DOS or *Linux* virus scanner with a few parameters, like the selected drives or folders.

Currently, there is no known ability to scan NTFS online encrypted files and folders in this situation, even if the password of the user account is known. These files can only be accessed in the recovery console or if the user is logged in under *Windows*. If a virus scanner attempts to access such a file or folder, it should display a brief warning message rather than skipping the file silently. Therefore, it's important not to store programs in such encrypted areas of the hard disk, which will start automatically, for example from the Autostart folder or 'Run' registry key.

### Disinfection Trouble

Boot viruses are one of the oldest forms of these digital parasites, but even ten-year-old boot viruses are still found in the wild. They do not cause very much trouble for DOS-based *Windows* versions, such as *95*, *98* and *ME* – often, these can still be started after infection.

However, *NT*-based *Windows* versions, such as *2000* and *XP*, will not start after a boot virus infection, making a rescue system very important. While DOS-based scanners can identify and repair boot viruses quickly, in *Linux*-based programs the detection routines to scan the MBR and all bootsectors for this kind of malware are often not yet implemented, making boot viruses invisible to the scanner.

Macro viruses should not cause any greater trouble, but file viruses, Trojan horses, backdoors and worms can cause a few problems, if they change INI files or Registry values. For example, if they start automatically at boot-up time ('Run', 'RunService' keys etc.) or if they change the properties of a file type, as Pretty_Park and SubSeven will do for an EXE file. In this case, it would be a good idea to replace the malicious files by a helper program, which can undo the changes at the next clean *Windows* start. Direct writes to the complex, not completely documented Registry file structure should not be made.

Other kinds of change, such as the deletion of additional program files (for example the Badtrans.B keylogger DLL), or the renaming of files, as would be necessary in order to clean the QAZ worm (delete the worm file notepad.exe and rename the backup copy note.com to notepad.exe), should be easy, too.

However, all these write operations will be problematic in the case of the complex internal structure of NTFS drives. Both *Linux* and the registered version of NTFSDOS have some problems while creating, copying and changing a large number of files on an NTFS drive. Usually, the file structure will be a little corrupted, causing *Windows* to display an error message or record something in its Event Log at the next startup, but the good news is that they are often recoverable.

Furthermore, there are usually only minor changes needed in the file for the cleaning process, meaning that the size of the file is unchanged, or only decreased. Therefore, it should be easy to heal the file without greater, possibly corruption-causing changes in the NTFS file structure. However, such an NTFS clean functionality should be tested carefully and disabled by default.

Another issue is related to the backup of the infected files before the clean process is carried out. This should not cause any problems on FAT drives (unless there is not sufficient space available on disk), but on NTFS such write operations can corrupt the drive. If (infected) backup copies of files need to be kept, it's a better idea to save them on a non-NTFS drive, such as a floppy or ZIP disk. Under *Linux*, it would also be possible to burn a CD using a simple command-line tool, but that is probably too complex a task.

### Report Files

Every scan and disinfection process should produce a readable log file, in which all the relevant information is stored. This should include the date and time of the scan as well as the last update of the signature files and the main virus scan engine. One important task that is often forgotten is to record all the activities of the program during the cleaning process. Usually, the report includes information such as the name of the infected file and whether this file has been cleaned or deleted, but not what kind of Registry and other file changes have been made, which would make the clean process much more transparent.

The use of helper files should also be documented – for example the fact that a second restart of the computer will be necessary after the helper program is executed and has undone the malware changes. Finally, it should be possible to save the report file as ASCII text to disk or to print it out. Don't forget that *Linux* will use a simple line break only, and not two characters, like DOS or *Windows*.

### Conclusion

Today's anti-virus rescue systems are too limited to be useful against today's file system and complex malware. It seems that these routines have been written once and not updated for a very long time.

However, with a little research, it should be a relatively easy task to help infected customers with a powerful, menu-driven rescue system which will scan and clean the local hard disks. This could be extended to include a few more emergency rescue programs, such as an undelete utility or a disk editor.

# FEATURE 3

## Keeping Track – the Value of Incident Reporting

*David Ensign*
*DOE Headquarters ASSIST, USA*

Ten years ago, the US Department of Energy created the Headquarters Automated Systems Security Incident Support Team (ASSIST), whose primary focus was on enterprise virus protection. At that time, there were few viruses in the wild, and for the first three years the annual number of encounters was less than 100 – insignificant by today's standards.

Nevertheless, from the beginning, the ASSIST instituted stringent policies so that every virus encounter was reported. As time passed, this became a cornerstone of the program and provided a foundation for a successful virus protection effort. But maintaining our ability to capture pertinent data has not been easy, as the growth in virus sophistication and diversification of vectors has increased the demand for new data capture mechanisms.

### Why Report?

At the start, part of the reason behind the institution of the reporting procedures was the structure of the DOE. The Headquarters environment is a conglomeration of semi-autonomous organizations – many with their own computer support staffs – which are scattered in buildings across the Washington, DC, region. Therefore, the ASSIST relies on organizational support staff to respond to virus incidents.

As a best-practices guideline, we require those staff to fill out an Incident Investigation Form for every internal infection. This allows us to review the actions taken to ensure that proper eradication has been carried out, containment (especially within the Headquarters area) is complete, and any protection lapses have been corrected.

Regardless of your organization's size or structure, detailed documentation of every internal infection should be mandatory, with secondary review essential for quality control. An infection implies that a protection failure has occurred, and it is critical that the failure is identified and corrected, that systemic remedies are taken if warranted, and that you ensure that other users have not been similarly affected. Especially today, any vulnerability cannot be ignored or it will be exploited.

We realized quickly that the information we had collected provided us with a valuable picture of virus activity. For one, because of the ASSIST's overall Headquarters purview, we were able to identify situations that span organizations, especially when we see that one organization is the source for numerous incidents spread across several other organizations, indicating a systemic problem that might go unnoticed with a narrower focus (the herd immunity syndrome at work).

We could see which vectors were prominent (diskettes at that time) or new (email with the advent of macro viruses) and escalate appropriate protection implementations. We could identify common sources, either specifically (a particular company or DOE field site) or generically (homes and schools) and work to rectify them.

Of course there is the age-old reason to collect data: it justifies our existence and our resources. While an exponential encounter curve is not a pretty sight, it does tend to get people's attention.

### Logistics

Having established the benefits of reporting, the issue over the years has been the logistics of data collection and organization.

At first, it was easy to keep track of information manually, but within a year we found that frequently we were developing new report requirements for a variety of time frames, and it became obvious that a database was necessary – after all, we are in the computer business.

Even when the number of encounters grew to hundreds per year, the data entry requirements were manageable, and our database, while antiquated, continued to do the job into 1998. However, a number of events started to impact our existing processes.

One of our early tenets for incident response was that the users should not be allowed to clear their own systems, because they may not do it correctly and often don't consider who else – either upstream or downstream – may be affected.

Our original anti-virus software allowed us to disable auto-eradication, forcing the users to involve computer support, but we lost that capability with new software obtained in 1995. It soon became obvious that, despite policies and for convenience, infected users were clearing their own systems and continuing with their work, so there were many encounters that were going unreported. We began to lose our ability to see the big picture.

Also there was a degradation on reporting from the various organizations. Because of the wide variety of anti-virus software in use and the different policies within individual organizations, very little automated reporting was occurring, so we were reliant upon the cooperation of others for our information.

Unfortunately, because virus response was considered a nuisance in the face of increased workloads and reduced resources, protocol was often sacrificed for expedience, even though there may be a cost to other DOE elements outside the realm of the immediate support staff. (This is no longer a problem; the constant barrage of new viruses is a sufficient reminder of the need for everyone to keep up their guard.)

### Vector Shifts

The advent of macro viruses changed the primary vector from diskette to email.

Every time there is a vector shift, we have a problem. We are reliant on various reporting mechanisms that are built into existing protections and policies. However, we usually have neither protections themselves nor robust reporting on the protections in place for a new vector, because most enterprises – normally for monetary or resource reasons – don't address new infrastructure requirements until they need to.

It seems that almost everything related to viruses is reactive. As a result, we may not recognize a new vector immediately because it is being omitted from the reporting pool – sometimes resulting in a degradation in reported encounter numbers.

While we recognized that email was a growing conduit, we continued to rely on desktop detection and reporting for our data. While, contrary to industry trends, we saw decreasing numbers of virus encounters, our data did provide sufficient evidence to indicate that we needed to address the email channel.

### Email Gateway

Fortunately before the appearance of the LoveLetter virus, DOE implemented an email gateway with automatic notification to the ASSIST. This had an immediate impact upon our data.

Numbers went through the roof. Where, previously, we had been identifying fewer than 20 incidents per month, we started capturing ten times that many.

Since it was unlikely that email infections made a quantum leap concurrently with the installation of our email gateway, these figures revealed the extent to which virus incident reporting was being ignored – previously each of these messages had been reaching the desktop. If you ever needed evidence of the ineffectiveness of human policies, this was it.

As a result of the escalation in encounters, the data entry requirements exceeded our resources. Not only did we have a new reporting mechanism, but our data was being provided in a new format: *Outlook* email. Fortunately, because notification was electronic, we could write programs which would parse the mail automatically. In the long run, this was very efficient, but it put short-term pressure on our resources to develop the programs.

The fact that the gateway was a network infrastructure device maintained by the central Headquarters support staff (including the ASSIST) gave us new control over data capture. This was the first truly enterprise-wide automated reporting mechanism, and the reliability of our numbers improved dramatically.

In addition, the gateway provided automated notifications to sources – a process that had previously been carried out manually.

### Definitions, Metrics and Statistics

We encountered a number of problems with definitions and metrics. In the early days, we had a statistic which we called the 'media-to-incident ratio'. This told us how many infections occurred as a result of each single incursion.

A media-to-incident ratio of 1.0 was excellent: it meant we stopped the virus at its point of entry, so only the source media (usually a diskette) was infected. We considered an enterprise monthly average of less than 2.0 to be our target metric. However, email viruses, especially mass mailers, changed this.

Now we stop nearly everything at the gateway, producing a high number of individual 1.0 ratios. With the sheer number of gateway captures, the average for the enterprise (constantly under 1.01) became meaningless, rendering our key metric antiquated.

Another conundrum involves identifying the important statistics. Is the most important statistic the number of vectors, with each email counting as an encounter? Each one of these is a contagion, and it only takes one successful penetration from the onslaught to start an outbreak, so vectors represent threats. But what if 90% of those emails come from a single SirCam source?

In determining threats, is the number of infected systems (which indicates protection failures) more important than the number of potential infection carriers? In the end, each has its place. Internally, we focus on infections; externally, we are concerned with vectors – but it means we have to build in logic to separate the two (and what do you do about viruses that spoof addresses?).

### Current Threats

Lately, CodeRed and Nimda have caused another vector shift. While we have firewalls in place, they are not configured to capture all encounters in a form (database) that can be processed and analysed in a meaningful way. This makes it difficult for us to judge the overall problem and the specific threat to DOE resources.

If we can't track these encounters, what about future IP-borne viruses? We are correcting this deficiency, but again it

will require us to write interface programs to take the captured data and collect it into our central encounter database. The greater the number of tools, the greater the resources needed to support them, and the greater the difficulty in consolidating results.

With the advent of the LoveLetter mass mailer and the current and seemingly endless barrage from SirCam, BadTrans, and Magistr, we now have monthly encounter numbers in the thousands.

Under these conditions, it may seem that data collection has less significance, but it is more important than ever. First, it allows us to spot new viruses early, capture them, and analyse how effective our many protections are against a new threat.

Second, we get a clear picture of what is happening in the real world, telling us where to focus our efforts. Knowing what viruses are impacting home users, for instance, allows us to provide guidance to our community that extends beyond the office.

And the original reason is more relevant than ever: data collection justifies our existence and the need to maintain or increase resources. Once again, a thousand hits a day gets people's attention.

### Conclusion

Some may believe that data collection and virus reporting don't provide sufficient benefits for the amount of effort invested, but we consider these activities to be essential.

The Headquarters community comprises 6000 systems and, while we haven't been immune to periodic outbreaks of the occasional fast-moving mass mailer, we still haven't lost a byte of data to a computer virus where it was reasonably preventable.

This is an exceptional record, and ultimately we believe that it is our constant, real-time data collection combined with robust reporting and analysis that has allowed us to achieve that.

If nothing else, the compulsion to gather data has led us to implement protective tools that allow us to do so, usually ahead of the curve, and to the benefit of the community.

If you take advantage of capabilities in existing protections, it doesn't have to be a major effort; if you're doing things right – with lots of infrastructure protections – the framework should be in place. For many, the first glimpse at the data is a shocking revelation of just how much an organization is at risk.

*A complete history of virus-related activity at DOE Headquarters from 1992 to 1998 is available at the ASSIST's Web site, http://www.microtech.doe.gov/assist/ (click on 'Reports'), giving an idea of how the statistics have been used over the years.*

# OPINION

## What's Coming?

*Peter Morley*
*Network Associates Inc., UK*

Since we receive virus swaps each month, *McAfee*'s Virus Lab at Aylesbury provides an interesting perspective, not only of what AV vendors are doing, but also of where the computer industry is going. This article is an attempt to forecast the future, based on recent history. My previous attempts at doing this have, in general, been factually good, although estimating the timing of events has proved more difficult. You may be in for some surprises, even if you're an industry guru!

### Recent AV History and Projection

2000 was what I would class as a 'normal' year in the anti-virus community. Despite various outrageous predictions (not mine!), it was a busy year, with the usual annual 'quiet patch' in late summer to early autumn – the period leading up to the *Virus Bulletin* conference.

2001 was a quieter year, and the annual quiet patch was very quiet. Over the course of the year, the lab processed an average of just over 200 viruses and Trojans per month. Even the number of Trojans fell off slightly, despite the fact that most of the AV vendors are taking them seriously now, and despite the fact that there were a number of major outbreaks, such as CodeRed, and Nimda.

I am writing this mid-March 2002, and the March inputs which in a normal year would be up to speed, are rather low. Since previous years have shown a late-summer slowdown, I anticipate that the same will happen this year. So, I predict that 2002 will be a very quiet year, with a nearly dead patch late summer to early autumn. I believe that we will be down to an average of about 150 viruses and Trojans per month by the end of the year.

I can see no reason why 2003 should be any more active than 2002. In fact, it may be quieter still.

There's no need to panic yet! While, for many industries, dramatic reduction of input material may be a disaster, leading to industry trauma, business failure, and consolidating takeovers, the same does not apply to the AV industry, because customers still need protection even if there are fewer new hazards.

### The Linux Phenomenon

You may have noticed some recent discussion about the rise of *Linux*, which is an operating system alternative to *Windows*. *Microsoft* has certainly noticed it and has introduced *Windows XP*, bringing together the various previous *Windows* versions, as well as moving, hell for leather, into the computer games market segment.

*Linux* is based on the Open Source philosophy, so if you feel the need for a modification, you can make it, or get someone to make it for you. I am unclear as to how this will affect its acceptance in the home computing market.

The spread of *Linux* is seriously hampered by two things:

- There are many sources, and if you get involved with several of them, chaos can result.
- *Linux* marketing does not appear to be as effective as it could be.

Despite these setbacks, *Linux* is coming. I have nine pieces of evidence pointing this way, some of which I shall outline here.

The original statement by *IBM* of their intention to invest a large sum of money in pushing *Linux* pointed to the fact that *Linux* is suitable for large mainframes as well as for PCs and servers, and that it closes the gap between them, working downwards. Three conclusions can be drawn from this:

1. *Linux* is coming from the top (mainframes) downwards, as well as from the bottom upwards.
2. When, in the 1980s, the computer industry split into PCs and traditional, we thought the split was permanent. It isn't. *Linux* can provide the means for the industry to integrate once more, but the process of integration requires takeovers and reorganization.
3. For big customers this is good, because within their own organizations, they can centralize control to a single group, and run the business as an entity.

More recently an *IBM* statement declared that, by mid-2004, there will be more *Linux* workstations than *Windows* workstations. This says 'We're winning!'. I am aware of the amount of internal work and approval required before such a comment can be made – it is considerable.

Recently, I discussed some of these topics with Alan Solomon, and discovered that all but two of his 28 machines are running *Linux*.

The recent discussions about *AOL* considering going *Linux* indicate that Internet use is no reason not to change operating systems, and that there may be some advantages. One of those advantages is that there are not nearly as many virus and Trojan hazards if you use *Linux*.

Articles have started to appear in publications (including *Virus Bulletin*) about the weak points of *Linux*, which could provide opportunities for the 'baddies'.

### Sixty-four Bit Processing

PC processor chips using 64-bit processing are coming, in the initial guise of Intel's Itanium, and AMD's Hammer. Both are some two years late, and I predict both will be available by the second quarter of 2003, if not sooner.

The effect will be (gradually) to provide much more powerful servers, and to close the gap further between servers and mainframes, working upwards, and making things easier for the use of *Linux*. The take-up will be gradual.

### Linux Viruses?

*Linux* viruses are the key question. Let's take another look back at history:

- Back in 1987, when I retired from *IBM*, the number of DOS viruses was less than 10.
- In mid-1990, when I joined Alan Solomon, the number of DOS viruses was about 220.
- At the end of 1993, when I moved into the Virus lab, it was 3,500, and rising fast.
- Now, in March 2002, there are 60,000–70,000 viruses/Trojans, of which 30,000 (legacy DOS file viruses) are of no interest to anyone except reviewers, and people who keep collections.
- Now, in March 2002, there are 130 *Linux* viruses/Trojans, and the figure is rising slowly.

It may be tempting to conclude that the 130 will rise over the next four years to 3000 or so. However, I cannot draw this conclusion.

Between mid-1990 and the end of 1993, several virus construction kits were developed in the USA (MPC, VCL and IVP). These kits made it easy for authors with DOS machines, to write many hundreds of simple viruses. These viruses helped the AV industry become established.

Even if several *Linux* virus construction kits become available now, there are not enough potential *Linux* virus authors waiting to take advantage of them. And they won't exist until the *Linux* explosion starts in the home as well as in business.

Even if the *Linux* explosion does happen in the home market, potential virus authors will be discouraged by the fact that we now know how to write generics, to detect and repair new viruses before they're even written.

So, I have to conclude that *Linux* viruses are not really a hazard.

### Linux Trojans/Malware?

Sadly, I think that this is the real danger area. I believe that *Linux* Trojans will come at us in increasing volumes, just as *Windows* Trojans did. You can expect Backdoors, Password Stealers, Mass Mailers, QZaps, sly Deletions, Illegitimate Accesses, and all the other hazards we know and love.

But, I'm afraid I can neither say when, nor how fast. My best estimate is starting at the end of 2003, and growing to 30–60 per week. The one consolation is that if they are produced using packages, they will quickly be brought under control.

### Summary and Consequences

The AV industry is safe, as long as new viruses and Trojans keep coming, because users still need to update their protection.

There are millions of machines which will continue to use *Windows* over the next five years, and they too, will need to update as long as *Windows* viruses and Trojans keep coming. If experience is relevant, they *will* keep coming for five years, and I shan't attempt to forecast further than that.

The AV industry is in pretty good shape to handle whatever *Linux* malware comes along.

The slowdown in the appearance of new viruses has a major consequence for me in that I can ease back on removing detection of legacy viruses from AV software. I could even stop for a year or so, without doing much harm. However, I still believe that reviewers and customers should separate legacy DOS file viruses from their virus collections, and stop testing against them! Perhaps I should take out a couple of old rubbish viruses each week, just to keep them awake!

### Unanswered Questions

If the computer industry does start to integrate again, it follows that Intel or AMD could become takeover targets. If so, when? My best guess is AMD, at the end of 2004.

Will the use of *Linux* on mainframes lead to a growth of the anti-virus market? I don't think it will, but it may lead to the introduction of specific product categories for use on mainframes. Time will tell.

What is the future of *Microsoft*? Is *XP* the last, (or next to last) *Windows* manifestation? I think it probably is. If I'm right, the world will breathe a sigh of relief.

But there is a rider to this one. Recently Bill Gates initiated a campaign to improve the security of all *Microsoft* products. This could lead to a new *XP* version. It will certainly lead to multiple patches. And it may even affect the AV industry. As for the future of *Microsoft*, watch this space.

Will the AV companies become takeover targets for the big mainframe companies? Experience suggests they won't, because previous attempts to handle viruses in such organizations have been dubious. But it will depend on volumes, and on whether completely new problem categories appear. Wait and see.

# PRODUCT REVIEW

## BitDefender Professional

*Matt Ham*

Those readers of *Virus Bulletin* who have an eye for seemingly vanished anti-virus products may have been wondering whatever became of *Softwin*'s *AVX*, a product which was reviewed a few times in these pages before disappearing into obscurity.

The vanishing act was, in fact, at the behest of *Softwin*, who subjected the product to a process of metamorphosis before renaming it *BitDefender*. As with all such alterations in external form there is the question of whether they signify a deeper underlying change in functionality.

However, the answer to any such question where anti-virus software is concerned is rarely definite. Products evolve constantly and there is a tendency for the underlying engines to become more modular. What were once monolithic blocks of code which underwent very gradual changes have become collections of modules which can be replaced on a case-by-case basis.

*BitDefender* is of this modular build, as judged by the components included in program updates. Therefore, the question as to whether *BitDefender* is a totally different beast from its predecessor *AVX* is unlikely to have a meaningful answer.

### Product

Versions of *BitDefender* exist for a variety of platforms, including some of *Microsoft*'s more specialized server platforms. These include *Exchange* and *Exchange 2000 Server* in addition to the less commonly encountered *ISA* and *SharePoint Servers*. Of most interest in the context of this review will be those products labelled 'Corporate'.

On a more generic front, support is offered for mail servers – more accurately SMTP servers running on a *Windows* platform. The desktop-based version of *BitDefender* is *BitDefender Professional*, which is available for *Windows* versions from *95* though to *XP*. A package named Enterprise Manager addresses the administration of this set of products.

Away from the primarily *Windows* desktop in a business environment, a number of other products are available. These include versions for *Linux*, *Windows CE*, *Palm OS* and standalone versions for *MSN Messenger*, *Yahoo! Messenger*, *ICQ* and *Microsoft Net Meeting*. These four messaging solutions are included in the *Windows* packages that were inspected.

Of topical note, it was announced recently that the *BitDefender* engine would be included in *GFI MailSecurity for Exchange/SMTP* – a new product, which I hope to review in *Virus Bulletin* in the near future.

### Documentation

The review copy of *BitDefender* was supplied electronically, as is becoming the custom, and thus neither box nor contents were available for inspection. However, documentation was available in the form of the ubiquitous PDF as well as the help function within the program itself.

The help function proved to be the common mix of useful information with not quite as much detail on some features as might be desired. For example, the section on the two heuristics features, one of which announces warnings and the other suspicious files, does not explain in any detail what differentiates the two types of alert.

Also, a problem was noted with the *Windows 98* version of the program – where the compiled HTML format of the help data was not accessible on a default installation of that platform. Despite these problems, the help function is useful in most cases.

The PDF documentation has much in common with the online help. Most salient features are illustrated by screenshots of the appropriate part of the GUI, together with arrows so as to make certain that the subject being discussed is clear.

The information presented is good and useful, though I have two very minor niggles. The first of these is that, at some points, the level of instruction is perhaps *too* detailed – explaining, for example, that a button labelled 'Next' will move on to the next part of the installation process. And there are some parts of the documentation in which the turn of phrase is decidedly odd – clearly the result of an imperfect translation – but always eminently understandable.

### Installation

For the purposes of review the primary platform used was *Windows NT 4 Server*, with *BitDefender Professional 6.4.1* installed.

It came as no great surprise that *BitDefender* is packaged with InstallShield, making its initial appearance indistinguishable from many of its competitors.

First in the installation procedure comes the licence agreement which, like many, is epic and sprawling. Also in common with many other products, there is a declaration that the software cannot be guaranteed to do anything. The licence agreement includes the standard warning that the product should not be used in a variety of named environ-

ments, which include such specifics as aircraft navigation, as well as the rather more catch-all category of 'any application where failure could cause property damage'.

From here onwards the path is through familiar InstallShield territory, passing via the selection of a path for installation, presenting the choice between Typical, Compact or Custom installs. It is in the interpretation of these three categories that products tend to deviate from the predictable.

Taking the Custom install as a starting point, the choices of components to be installed fall into six categories, one of which has several sub-categories.

As the default option, Murphy Shield, BitDefender Scheduler, BitDefender Quarantine, BitDefender Live! and BitDefender Shell extension are selected for installation. The somewhat bizarrely named Murphy Shield is the on-access component and BitDefender Live! is the update mechanism.

One component not installed by default is BitDefender P2P protection, in which there are several sub-categories for common peer-to-peer applications. MSN Messenger, Net Meeting, ICQ and Yahoo! Messenger are each selectable individually.

One part of the *BitDefender* functionality which appears, at this stage, to be installed by default, though does not receive mention in these installation choices, is the on-demand component. For the purposes of testing, all components were selected for installation, resulting in the maximum installation size of just less than 16MB.

A few questions concerning start menu naming follow, after which the *BitDefender* configuration options are reached. Quite a novel feature here is that this is a point of no return in the installation process – before any files have been copied, but beyond which the Back button does not operate, with the Cancel button being the only option available if a selection error has been made.

The Setup Type dialogue offers the choice of whether the on-access protection should be 'AntiVirus Only', or 'Act as Personal Firewall'. The definition of firewall here is somewhat broad, containing a variety of Internet protection options, which are covered later. These were selected en masse for the purposes of testing.

After this, the files are transferred and the interface changes to the *BitDefender* style. In its new guise the selection process becomes significantly more intriguing.

First, the user is faced with the decision of whether they wish to activate Real Time Virus Reporting. This feature sends virus detection reports directly to *Softwin*, where they are compiled with data from other *BitDefender* users in order to provide global statistics. On choosing to activate the process, the user is required to select their country from a dropdown list.

The registration process which follows is somewhat dull, but worthy. This completes the installation procedure and the user is presented with the option of rebooting immediately.

Installation of the *BitDefender* Enterprise Manager requires *Internet Information Server* (*IIS*). Since this was not standard on the test machine image, the process stalled on first attempt. *IIS* was added, followed by the obligatory reinstallation of service packs, and installation of Enterprise Manager continued. Unfortunately, due to hardware failures, a full review of the Enterprise Manager component proved impossible.

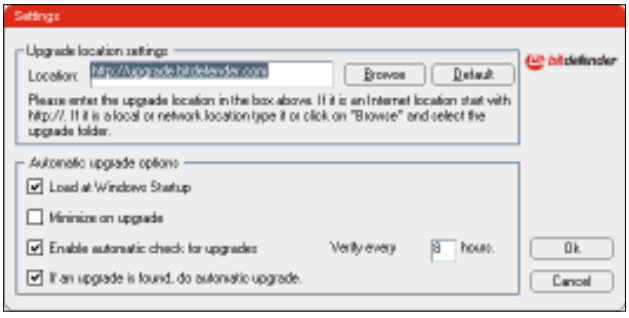
### Updates and Upgrades

As mentioned previously, the BitDefender Live! component is responsible for downloading updates and upgrades from the *BitDefender* servers or elsewhere. By default, the site http://upgrade.bitdefender.com is checked every eight hours for new files, with an extra check at boot-up and automatic installation of new files when they are detected.

A silent version of the upgrade is selectable, which minimizes the Live! interface while it is operating – though, since some updates require a reboot, this unobtrusiveness is shattered when a dialog box appears, requesting a reboot. The location for updates may be altered to a more local repository if required, so as to enable updating of LAN-based machines without the requirement for all of these to be permanently able to access the Internet.

For this purpose, of course, the files for updates and upgrades must be transferred through either an Internet or LAN connection, so the file sizes may be relevant. On the date of testing there were three file types available to download for use with *BitDefender Professional* . Whether or not the sizes of these files are typical is unknown. The three files downloaded were a weekly update .EXE file (2.1 MB), a zipped version of the weekly update (2.0 MB) and a daily zipped version (256 KB). However, the contents of the zipped and non-zipped versions of the weekly update were not identical. The two zipped versions of the update file were both simply collections of new files to be inserted in the *BitDefender* directory, while the .EXE file is a dedicated installer, though it performs the same function.

Somewhat obscurely, there is no indication as to which, if any, of these is the file which will be used by Live! if



present in a target directory. Downloading all three files and setting their location as the update location gave no joy, and no obvious means of triggering a local update could be found, either through experimentation or by referring to the documentation.

### Web Resources

The *BitDefender* website is located at the rather obvious http://www.bitdefender.com/, though any links or references to the parent company *Softwin* are less easy to detect. Readers who are interested in the background of *Softwin* can find more information at http://www.softwin.ro/, though some of the anti-virus material here still relates to the old *AVX* product line.

The look and feel of the *BitDefender* website is a good deal more colourful in its advertising portions than on the informational links. Of the main page approximately one third of the space is taken up by a central block of advertising, a further third by press releases and testimonials, while the remaining space has more informational content. Despite having such a high profile on the home page, advertising is significantly less obtrusive throughout the rest of the site.

Among the general commerce and press release-related portions of the site, there are some areas which are of more interest from a technical and informational point of view.

On the home page itself there are links to the 'Virus Top' (sic) and 'Latest Threats'. These are, respectively, lists of the five most prevalent and recent, potentially widespread viruses. The grading of the former is based upon a ranking system which is not readily apparent, since it does not appear to be linked to the Real Time Virus Reports (which can also be accessed from the home page).

The Real Time Virus Reports are derived from a combination of data from the *BitDefender* scanners and online scanning performed on the site. For privacy reasons, reporting can be turned off when using either of these services, which is the cause of more general problems as far as such statistics are concerned.

Where there is a conscious decision as to whether to supply information, the resulting statistics will be skewed in some fashion. Unfortunately, this is true for almost all virus statistics used in the industry today. Even in those countries, such as Japan, where there is a system of required reporting of viral incidents, there will be some degree of free will exercised by the public in their decision to report. Given this, the vast majority of virus statistics should be considered to be representative only of those who are willing to admit to their infected state.

The OnLine Scan comes complete with a lengthy legal preamble, which is effectively the same as that provided for the CD-packaged scanner. Provided this legalese is accepted the next step is to install and run the online scanner component – making it less purely online than other examples of its ilk, though about equal in functionality.

When the program has completed its initialization the user is prompted to enable Real Time Virus Reporting and regaled with terrible tales of *BitDefender*'s compromised response time if this feature is not activated. After this portion of the scan another round of file downloads commences before scanning can be initiated.

As might be expected, the speed of this scanning is not lightning fast – it took 49 minutes to scan one drive of an Internet-connected machine, containing 4 GB of information in 48,651 files. By contrast, a locally-installed version of *BitDefender Professional* took 27 minutes to scan the same objects.

Two main varieties of download are available from the site. As is to be expected there are downloads of products for evaluation, as well as a selection of removal tools. The tools are linked to from both the individual virus descriptions and a central removal tool repository.

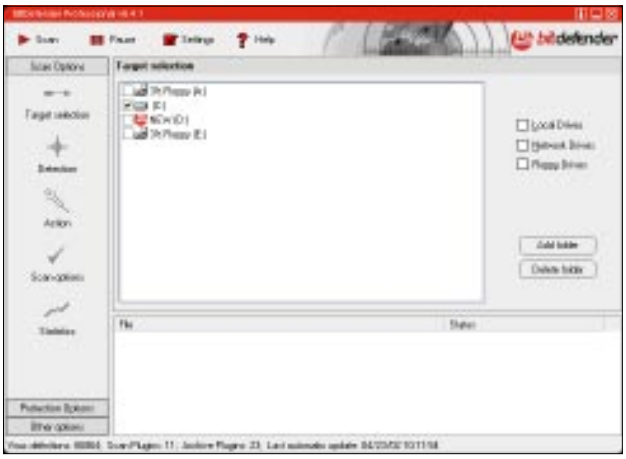
One fact that the latter brings to light is that some of the viruses which have a removal tool do not have a description associated with them. This seems to be very much a function of the age of the virus in question. The virus description list on the website is fairly comprehensive for major threats during the time period it covers, though this covers only the last few months. Prior to this only such major threats as VBS/LoveLetter.A are listed. It can be assumed that newly discovered threats will continue to be added.

On the more quirky front, the *BitDefender* website must surely win prizes for its insistence upon providing legal guidelines for the use of almost every file provided. This includes a detailed list of where and how the *BitDefender* logo may be used, containing among other restrictions the distinction that the logo cannot be used in a place where it cannot be seen.

There are further restrictions on the use of the logo on sites containing inaccurate information. Some might consider this restriction could apply to the *BitDefender* site itself since, in one section, hoaxes are described as generally being used for advertising purposes (the providers of colonic irrigation for cats have no doubt seen large increases in their sales as a result of that particular hoax, not to mention the manufacturers of undeliverable mail).

### Features

After initial installation and the obligatory reboot, a splash screen appears, which stresses that control of Murphy, the on-access component, is available from the system tray. Since the networked test machines were isolated from the Internet and all other outside interaction, it came as no surprise that the next alert stated that the update server could not be reached. Also immediately noticeable were the



appearance of two new tray icons, one being that for Murphy and the other for BitDefender Live!, as well as a large new section on the Start menu.

This section links directly to the main *BitDefender* scanner, from which the other components may be controlled. This does, at least, explain the requirement for the installation of this module. There is also a smaller link from the Start menu, which provides access to components of the software, though these are all available individually through the main GUI.

The colour scheme of the interface combines red with shades of grey, and the design features the rounded GUI shape which has become common in recent AV releases. Also akin to several recent releases, the interface features a left-hand panel of options with the right-hand area changing as appropriate to the option currently selected.

Options are divided into three main groups in the installation tested, namely Scan Options, Protection Options and Other options. Further control is offered by Scan, Pause, Settings and Help buttons at the top of the interface, though there is a total lack of the drop down menus which are commonly found in similar interfaces.

Since the buttons are few in number, they seem a good place to start the discussion, though they hold few surprises. Only the Settings button is less than obvious in its functionality, this being used to save and load program settings or to set the current values as the default.

The Scan Options side bar consists of the further sub-options of Target Selection, Detection, Action, Statistics and a further Scan options section. This nesting of options with only slight differences in capitalization to distinguish their names is more than a little confusing. Also confusing is the fact that there is no way of telling which of these options is selected, since there is no highlighting on the left-hand panel. Admittedly, the contents of the right-hand panel are a giveaway as to the current selection, though the lack of a clear indication is still an irritation.

Target selection allows for scans to be performed on whole drives, on all drives classified as network, local or floppy, or on folders. The targeting is able to browse to folders, but is not as precise as to be able to target individual files, which might be a desirable feature under some circumstances.

Although the same directory may not be selected more than once, it is possible to select a directory for scanning and then select all subdirectories individually if so desired. Since scanning options are applied globally, however, there is likely to be no great need for setting multiple scans of the same target.

Detection in its standard settings indicates that boot sectors and files should be scanned with packed files opened. Alternatively it is possible to set the scan to be directed to Programs or user-defined extensions. For those who may, for example, have bulky custom format files which should not be scanned, an option to exclude extensions is included. In addition to the targets already mentioned, it is possible to set scans to check memory, open archives, open email, verify Internet ports and check system security. The last two are described in the help files as scanning the local Internet and scanning for unsafe security or abnormal situations, respectively.

The Action portion of the side-bar is self-explanatory, with some less than obvious behaviour. Options available are report, prompt, disinfect, delete or copy for infected files. These come with the caveat that the delete option does not work on write-protected files, while the copy option does just that – copying, rather than moving the files declared to be viral.

The only option available for suspicious files is to copy them to a folder. In both cases these folders are user-defined, though two default folders do exist for the purpose. The situation as far as files which trigger a warning is concerned is not stated explicitly – though experimentation proved that these were treated as infected.

The other two modes for the left-hand panel relate, as mentioned, to external programs which are invoked by the main GUI. As such the right-hand panel is not used in these cases and a new interface pops up for the adjustment of settings and viewing of results. The Protection Options category covers Shield, Internet, Mail, Scheduler and Live Upgrade.

This is a case where the naming conventions used within the main GUI do not quite tally with the names used elsewhere to describe the services provided – Live Upgrade, for example, being an invoker for BitDefender Live!. All components but Live! are optional parts of the installation. However, the icons remain whichever installation type is selected – leading to some ugly error messages, which inform the user that their installation may be corrupt in addition to the possibility that this component was simply not installed.

Shield is by far the most involved component, referring to Murphy Shield. This application is launched separately, and gives numerous options, some related to the purely

anti-viral portion of the software, while others are in areas more traditionally regarded as security-related.

The security aspect starts with the first option in Murphy, since the changing of preferences within the program may be password protected. Three default levels are supplied, these being, at the lowest, the virus scanning Content Scanner component only, a medium level which does not offer Privacy or Active Content protection and the highest level which uses all of the components available. Each component can be activated independently of the others and fine tuning is available within components.

By default, the Content Scanner portion of Murphy scans incoming mail, floppies on-access and files on-access. It may also be set to scan floppies when the machine is shut down. Also by default, only program files are scanned (this in contrast to the all-files setting used on demand), and while packed files are scanned, compressed files are not.

All file scanning is supported, as is the use of a user-defined extension list. Detection of an infection will result in denied access to the infected object, though disinfection, deletion and copy to quarantine are also selectable.

Internet App Control is the function which controls those programs that can initiate Internet connections. The initial state of this functionality does not assume that any programs should be allowed to access the Internet, responding with a dialogue whenever an application attempts to do so. As an alternative option all Internet connections may be blocked.

The Internet Filter is more controlled than the App Control, offering choices as to specific URLs, IP addresses and ports which should be blocked. However, this portion of the program is not quite as simple as responding to a dialog box when, for example, a URL is used – the entries here must be added manually. This would be a good place to offer an import function for such information, but alas this does not exist.

On the other hand, the combination of Internet Filter and App Control should allow a good control over the fundamentals of Internet security for those dedicated enough and knowledgeable enough to know what should be blocked.


The next component of Murphy is the Behaviour Blocker portion – a useful presence for malware, including viruses, undetected by other means. This looks for 'suspicious' activity in File System, Registry and Internet usage. Each of these can be activated independently – and, if behaviour blocking is triggered, the application exhibiting the behaviour can be added to a list of banned applications. Privacy control, by contrast, does not add any real anti-viral functionality, simply allowing the blocking of inbound or outbound cookies.

Murphy returns to more relevant (to *VB* at least) territory with Active Content settings. This allows the blocking of ActiveX, Java or scripts. If Active Content monitoring is selected, without in addition choosing to block each of the three types, then a dialog appears each time one of these potential perils is activated. This allows the user to decide on a case-by-case basis – and adds the names of the objects to a list displayed here. Alternatively, each of the objects may be blocked – in which case no dialog will be produced.

The remaining two features of Murphy Shield are more informational in nature, giving statistical information on objects scanned, in graphical and text form, and product information.

Returning to the main *BitDefender* GUI, the Internet and Mail options from Protection options also link to Murphy Scan, while Live Upgrade links to BitDefender Live!. This leaves the scheduler as the remaining link from here – which, again, links to an external program, a scheduler for on-demand tasks which holds no great surprises.

This leaves Other Options as the remaining left-hand section to be inspected. This contains four sub-options. Of these, Language selection and Help are fairly obvious in their effects. Language availability is variable among products, though for the installed version English appeared as the only option (this despite there being several languages available upon installation).

The Web Encyclopedia links to the virus descriptions area of the *BitDefender* website, while View Log is a simple text viewer for the logs produced on scanning.

### Detection

Several scan types are possible by utilizing the various GUI settings. Those chosen for experimentation were the option choice between all files and program files, and the toggling of warnings and unknown virus detection.

Of the files in the *VB* test sets, using the default *BitDefender* settings, 466 were missed out of 20,710 total files. Of these, 26 were detected as warnings and 176 suspicious. Setting the scanner to scan only program files added a further 78 misses. As would be expected, removing scanning for warnings or suspicious files removed these

from detection. Of those files detected with all file scanning and missed on program file scanning, none were classified as either a warning or suspicious file.

Misses in default mode were almost exclusively in the polymorphic test set – W32/Fosforo, W32/Zmist.D, W32/SK.8044 and W32/SK.7972 were all missed, along with a selection of other polymorphics in various test sets.

More worrying, since it has been classified as In the Wild, were the misses of W32/CTX, again a polymorphic virus. This weakness in the polymorphic test set was in contrast to the generally good detection elsewhere.

Suspicious files were notable in that all but five of the 176 files designated as such were macro viruses. The files where warnings were triggered were those containing Neuroquilla.4544.A, Morphine.3500.A and Plagiarist.2051. It seems rather odd that definite virus names were declared for these samples, despite the fact that they were only worthy of the warning tag.

Activating the scanning to verify Internet ports and check system security had mixed effects. The default test server, when isolated from its LAN, gave no comment on the port tests, save a mention that the scans were indeed occurring.

The selection of ports which were inspected included a relatively wide range, though this was by no means complete. The selection included such easily recognised old favourites as port 31337 – used famously by some versions of Back Orifice. The system security scan picked up three noteworthy problems on the machine: the enabling of ActiveX, locally stored cookies and active scripting.

For the purposes of checking in a more realistic environment an essentially unsecured machine was attached to the Internet as a sacrificial victim and the same test was performed. The on-demand tests discovered no notable problems with the system, despite it having sufficient open ports to be a hacker's dream.

On-access, matters were rather more reassuring, since attempts to use standard ports were detected as such and triggered Murphy into asking whether these actions should be allowed. The Murphy dialog box was even apparent over programs working in non-windowed mode which were selected on the basis of their usual fragility when faced with unexpected network blocking.

The Internet App Control is thus operational on outbound connections via a number of ports, both the common http and ftp and a variety of ports assigned by less standard programs.

However, the problem with the detection of connections is quite what the average user will make of these alerts. It is all very well to know that ftp.exe is attempting to use the Internet, but an average user will not have much awareness of whether this is a good, bad or indifferent program to allow such access on-demand.

There is also the age-old problem of *BitDefender*'s on-access scan capability being turned off if a user considers the alert messages to be too intrusive. In addition it seems somewhat odd that such programs as BitDefender Live! are not pre-registered as acceptable for Internet access.

Other functionality, such as script blocking and URL blocking, behaved exactly as advertised – though, given the wide variety of behaviour which could be considered testable, the results cannot be said to be indicative of perfect real-world behaviour.

### Speed and Overheads

When faced with the *VB* collection in the tests described above there was very little noticeable difference in scan times between the test runs. This is not always representative of the situation with clean files and thus these too were scanned in each of the configurations previously applied. In this case the speed only varied from the baseline by the institution of scanning program files only. Even in this case the scan was only faster by approximately ten per cent.

To all intents and purposes, therefore, the gains in scan speed derived from altering scan configurations from the default, are not worthwhile when compared with the loss of detection.

### Conclusion

The host of features available in *BitDefender* should certainly enhance its ability to detect new viruses and worms by their activities in a more efficient manner than a system designed entirely around scanner-based heuristics. In addition, the large number of macro viruses detected by heuristics would suggest that this, too, is an area where the product is at least competent.

The weaknesses seen in the product were, however, threefold. From the speed tests it became apparent that the scan rates achieved by *BitDefender* were at the lower end of those generally encountered, and definite weaknesses were displayed in the detection of polymorphic viruses.

Overall, *BitDefender* is a product with very well defined strengths and weaknesses but one which is much improved since its previous incarnation as *AVX*.

**Technical Details**

**Product:** *BitDefender Professional.*

**Developer:** Softwin SRL, Str. Fabrica de Glucoza, Nr.5, Bucaresti, Sector 2, CP 52-93, Romania.
Tel +40 1 233 780; fax +40 1 2330763;
email sales@bitdefender.com;
website http://www.bitdefender.com/.

**Server:** 750 MHz AMD Duron with 128 MB Ram, 8 and 4 GB dual hard disks, CD-ROM, LS120 and 3.5-inch floppy, running *Windows NT 4 Server SP6*.

**Workstation:** AMD Athlon XP 1600+ with 512 MB Ram, 10 and 20 GB dual hard disks, ADSL Internet connection and DVD drive, running *Windows 98 SE*.

**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

**Infosec 2002 takes place 28–30 May 2002 at CNIT, Paris La Défense, France**. This three-day event will run concurrently with SIMBIOM, the First International Biometry Exhibition. For more information, including an exhibitor list and details of the conference and tutorials, visit http://mci-salons.fr/infosec/.

**The RSA Conference 2002 Japan runs 29–30 May 2002 at the Alaska Prince Hotel, Tokyo, Japan**. Nine tracks have been named for the conference, including: network threats and security, PKI, government policies and rules, advanced technology, risk management and corporate security and new products and technology. More information can be found at http://www.rsaconference.com/.

**The 11th Annual EICAR Conference and 3rd European Anti-Malware Forum** takes place 8–11 June 2002 at the Forum Hotel, Berlin, Germany. Discounted rates for early registration apply until 24 May 2002. See http://www.eicar.org/ for full programme details or to register online.

**Internet Security takes place as part of Internet World UK 2002**, at Earls Court, London, UK, 11–13 June 2002. A series of seminars and case studies accompanies the exhibition. For information see http://www.internetworld.co.uk/london/.

**The Black Hat Briefings and Training 2002 take place from 29 July to 1 August 2002** at the Caesar's Palace Hotel in Las Vegas, USA. The briefings will consist of eight separate tracks over two days (31 July to 1 August), with ten different classes on offer for training (29–30 July). For further details or to make an early reservation see http://www.blackhat.com/.

**Information Security World Australasia 2002 will be held 19–21 August 2002 in Sydney, Australia**. The conference and exhibition represent the region's largest dedicated IT security show. For full details see http://www.informationsecurityworld.com/.

**The 9th International Computer Security Symposium, COSAC 2002, takes place 8–12 September 2002** at Killashee Hotel, County Kildare, Ireland. Cost of registration includes your choice of 40 symposium sessions, five full-day master classes, and the COSAC International Peer Group meeting, in addition to full-board accommodation and meals. Register at http://www.cosac.net/.

**The 12th International Virus Bulletin Conference will take place at the Hyatt Regency, New Orleans, LA, USA from 26–27 September 2002**. Watch out for the full programme details at http://www.virusbtn.com/.

**Black Hat Asia 2002 takes place in Singapore, 1–4 October 2002**. For further information see http://www.blackhat.com/.

**Information Security Systems Europe 2002 will be held in Disneyland, Paris, from 2–4 October 2002**. For more information visit http://www.isse.org/.

**The Third Annual RSA Conference 2002, Europe is to take place 7–10 October 2002 at Le Palais des Congrès de Paris, France**. As well as keynote presentations there will be more than 85 individual breakout sessions on topics ranging from enterprise security to hacking and intrusion forensics. See http://www.rsaconference.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 involvement or interest in information and network security, whether new to security or seasoned professionals. For more information email csi@cmp.com or see http://www.gocsi.com/.

***Sybari Software, Inc*., has announced that the company has forged a partnership with *Kaspersky Labs***, resulting in the addition of *Kaspersky Anti-Virus* scan engine technology to *Sybari*'s *Antigen*. The *Kaspersky* scan engine becomes the sixth to be included in *Antigen*, joining scan engine technologies from: *Network Associates*, *Norman Data Defense*, *Sophos*, and *Computer Associates*. For more details see http://www.sybari.com/.

***McAfee.com* has developed a new security technology, *Grid Security Services*,** which brings together Internet-based grid technologies with XML-based web security services to provide real-time alerts and updates to users on the Internet. Users of *McAfee.com*'s *SecurityCenter* act as relay nodes, sending data (anonymously) to a central hub. The hub monitors this data and in so doing is able to alert users about breaking security problems and send the relevant immunization and update data to all of the systems on the grid. More information can be found at http://www.mcafee.com/.