ISSN 0956-9979 JANUARY 2002

VIRUS BULLETIN

THE INTERNATIONAL PUBLICATION ON COMPUTER VIRUS PREVENTION, RECOGNITION AND REMOVAL

IN THIS ISSUE:

• **Unstable Ground:** Alas, not even anti-virus software is immune to glitches and bugs. What happens when AV scanners encounter malformed files? Andreas Marx reveals all on p.14.

• **Worm Catching:** With four grown children, six children under the age of seven and three poodles, it's a wonder *TruSecure*'s Director of Malicious Code Research has any time or energy left for anti-virus research. We find out more about how Roger Thompson came to be catching worms, starting on p.16.

• **The Name Game:** Concerned by the abundance of malware naming methods, Frank Felzmann and Guenter Musstopf took matters into their own hands, wrote a paper, held a meeting and made some proposals. Now they are ready for others to join their campaign for a unified future of malware naming. Find out more on p.12.

CONTENTS

COMMENT

Malicious Threats of Peer-to-Peer Networking

“ Unfortunately, peer-to-peer networks are not invulnerable to malicious threats.”

Peer-to-peer networking is an alternative to the client-server model. Each computer is both a server and a client, commonly referred to as a servent. Recently, peer-to-peer networks have gained momentum with searchable peer-to-peer network file databases, increased network connectivity, and content popularity. Unfortunately, peer-to-peer networks are not invulnerable to malicious threats, privacy concerns, and security risks. Peer-to-peer protocols fall into three major categories. There are true peer-to-peer systems such as *Gnutella*, systems that require a centralized server such as *Napster*, and hybridized systems such as *KaZaA* that utilize super nodes.

In *Gnutella*, searches for content are passed to nearby servents, which pass the message along a chain until the maximum hops is reached or a servent replies with confirmation of matching content. The confirmation message is passed back through the chain and the originator contacts the servent with the matching content directly.

Napster’s peer-to-peer networking model involves a centralized directory server. A centralized server passes messages and keeps a global listing of all available content. Servents query this centralized server and only when transferring files do they make direct peer-to-peer connections.

Finally, hybridized systems such as *KaZaA* (based on *FastTrack*) take advantage of a super node. These nodes are determined by their available bandwidth and system resources. Such super node servents hold search listings for all nearby clients providing a reasonable subset of search listings on the peer-to-peer network.

Use of a peer-to-peer network introduces an additional vector of delivery. This could inadvertently transfer a peer-to-peer-unaware virus that has infected the open share. Of course, the virus still must be executed to become active. Also, viruses could take advantage of the regular use of a peer-to-peer network. The first *Gnutella* worm to be discovered, VBS.GWV.A, does this by copying itself to the *Gnutella* shared directory as a popular filename such as ‘Pamela Anderson movie listing.vbs’. The goal is to trick someone into downloading and executing the worm. Furthermore, viruses could harness the existing peer-to-peer network infrastructure to propagate. A worm could set up a servent on the victim’s computer, which could return exact matches for incoming search queries. Those downloading and executing the file will become infected. An example of such a worm is W32.Gnuman. Finally, software bugs such as buffer overflows could easily lead to CodeRed-type worms infecting peer-to-peer networks.

Peer-to-peer networks can be used for communication by malicious software. In many organizations, backdoor Trojans are not effective due to firewalls blocking incoming connection requests. However, generally, peer-to-peer software is not blocked by the firewall because they make outgoing connections. A backdoor Trojan could register with the *Napster* centralized server and pass a specific unique list of files. A hacker could perform a search on those files to identify infected computers. A request for a particular file would signal the infected machine to perform a task such as creating a screenshot. Information and control of the computer could then be exercised in this manner, bypassing the firewall.

Privacy is another concern. Users may configure peer-to-peer software incorrectly, allowing outside systems to obtain files from their computer – and nothing limits these to music files, they may be confidential data from an email inbox to proprietary design documents. A simple Trojan could perform such a configuration change. Even if the peer-to-peer network is configured properly, data is often transferred unencrypted and can be obtained easily by a network sniffing program.

Peer-to-peer networks pose a danger as an additional vector of delivery and a potential leak of confidential information. Administrators should begin analysing their networks for peer-to-peer network usage and configure firewalls and systems to limit or block their usage.

Eric Chien, Symantec Security Response, Netherlands

NEWS

And the Winner is …

December was a month for awards in the AV industry. *Sybari*'s *Antigen for Exchange* won a Commendation in the Information Security Product category at the Information Management Awards, and was also a finalist in the Product of the Year category. Meanwhile, *Sophos Plc* was named Company of the Year in the 2001 Real Business/CBI Growing Business Awards for its business performance, ambition, potential and management quality. In his acceptance speech, CEO Peter Lammer thanked *Microsoft* for making it all possible. See http://www.sybari.com/ and http://www.sophos.com/ for more details, respectively ▮

Empty Shelves

Softwin has parted company with US re-seller *Central Command*. *Softwin* will continue to deliver signature updates for the product formerly known as *AVX* (now known as *BitDefender*) to *Central Command* customers until November 2002. *Central Command* will be announcing the release of a new generation of AV software and security services shortly, although at the present time it appears impossible to purchase any anti-virus protection from the company. Those who pre-register for the brand new line of anti-virus solutions will be sent *CC*'s newsletter by email which, before giving details of current virus warnings, begins with the somewhat inappropriate slogan 'Without us, there's no defense.' Without an anti-virus product to sell, presumably, there's no business ▮

It All Becomes Clear

Following numerous customer queries, *DialogueScience* has issued a clarification of the differences between its two *Doctor Web for DOS* versions (16-bit and 32-bit). Although both versions use the same virus databases, it is only the 32-bit version (also known as *Doctor Web for DOS/386)* that possesses a full set of functions and capabilities. While *DialogueScience* strongly urges customers to use the 32-bit version, the 16-bit version will continue to be issued for use with 86/286 computers. Because of its functional limitations this version is available for download free of charge from http://www.dials.ru/ ▮

The Scores on the Doors

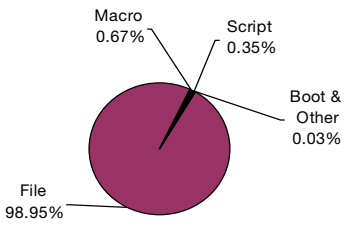
The Chinese astrological calendar had better make room for another sign as *MessageLabs* has declared 2001 to have been the 'Year of the Virus'. *MessageLabs* claims to have intercepted an average of one virus every 18 seconds over the year. In its end of year report, *MessageLabs* states that in 2001 it stopped a total of 1,628,750 viruses – 1,444,493 more than the previous year's tally ▮

Prevalence Table – November 2001

Virus	Type	Incidents	Reports
Win32/SirCam	File	17844	73.33%
Win32/BadTrans	File	2376	9.76%
Win32/Magistr	File	1518	6.24%
Win32/Aliz	File	1293	5.31%
Win32/Hybris	File	464	1.91%
Win32/Nimda	File	345	1.42%
Win32/MTX	File	79	0.32%
Laroux	Macro	67	0.28%
Win32/Klez	File	65	0.27%
Kak	Script	34	0.14%
Haptime	Script	31	0.13%
Nsi	Macro	20	0.08%
LoveLetter	Script	17	0.07%
VCX	Macro	17	0.07%
Marker	Macro	14	0.06%
Win32/Kriz	File	14	0.06%
Divi	Macro	12	0.05%
Win32/Funlove	File	12	0.05%
Win95/Spaces	File	10	0.04%
Win32/Hai	File	8	0.03%
Win32/Choke	File	7	0.03%
Win95/CIH	File	7	0.03%
Cap	Macro	6	0.02%
Form	Boot	6	0.02%
Tristate	Macro	6	0.02%
Win32/Navidad	File	6	0.02%
Win32/Pretty	File	6	0.02%
Win32/QAZ	File	6	0.02%
Win32/Ska	File	6	0.02%
Others [1]		39	0.16%
Total		24335	100%

[1] The Prevalence Table includes a total of 39 reports across 19 further viruses. Readers are reminded that a complete listing is posted at http://www.virusbtn.com/Prevalence/.

Distribution of virus types in reports



LETTERS

Out with the Old, in with the New …

[*As the party poppers are put away for another year, some major names in the AV industry reflect on the year 2001 and look ahead to the new year. Ed.*]

Norman, Norway

2001 was a memorable year for the AV industry. Not only have Win32 mass-mailers taken over the lead in the ItW reports (see the *VB* Prevalence Table), but also we saw the appearance of the 'file-less' CodeRed worms and Nimda. These worms were quite successful since they exploit (un)known security holes. Since both still appear on a frequent basis, it is obvious that there are corporates that have not applied all the relevant security patches yet. The 'file-less' viruses made the AV community aware that traditional scanning of files is no longer sufficient.

In 2001 both the number of hoaxes and their impact increased – SULFNBK.EXE is still confusing users. However, with such a major worldwide disaster as 11 September, an increase in hoaxes is to be expected. The conviction and sentencing of Kournikova author Jan de Wit, was disappointing, with a lack of corporates stepping forward and filing their damages.

In 2002 and beyond, the importance of *.NET* will rise. *.NET* has many possibilities, but it leaves plenty of room for security threats right now. It is something the industry will have to monitor closely this year. The number of security holes may increase, where the number is 'helped' by the newly-released *Windows XP*.

Righard Zwienenberg
Senior Virus Researcher

MessageLabs, UK

2001 was a challenging year in many respects, and there was plenty to keep me busy. 2001 was the year when we were first able to say that email-borne viruses are now an opt-in problem. I think that AV technology has matured to the point at which we can say to businesses that the cycle of infection, detection, and cleanup has finally been broken, and that there is now a credible alternative to scanning-only technology.

Looking forward, obviously there are other ways viruses spread apart from email, and I hope similar progress can be made here too.

Alex Shipp
Imagineer

Trend Micro, USA

It is that time of the year again when we look back at what's happened over the past 12 months. Living in California, I saw my power go off in the middle of the day, 'experienced' an increase in my electricity bill by hundreds of dollars, and eventually received notice that my electricity provider had declared bankruptcy. Certainly a summer to remember!

On the anti-virus front, I am glad to report that AV researchers from many companies are working closer together than ever. Credits go to numerous researchers (regular *VB* contributors) who have made this possible by providing the necessary forums as well as tools. Having seen this change in 2001, I look forward to the coming year. I wish everyone a happy holiday season and a successful start for 2002.

Joe Hartmann
Anti-Virus Research Engineer

F-Secure Corporation, Finland

In 2001 many computer worms like Ramen, Kournikova, Homepage, Magistr, Sircam, CodeRed, Nimda, Badtrans and Goner vied for pole position. This was the year of democracy, when nothing was confidential – Sircam went public (though not yet on NASDAQ), spreading users' documents all over the world. During 2001 the Internet became more 'worm-ed' than ever. Web sites were coloured in red, blue or green depending on the IIS (CodeRed) worm they were infected by. Users were Sircam-ed, Nimda-ed or in Badtrans, but this is probably not the end of the Internet. They are patient enough watching the endless battle between the AV community and the virus writers, even though it takes longer now to 'take the next hill' i.e. to analyse and kill Nimda.

One thing is for sure: there are more worms to come in 2002. Meanwhile, computer users, armed with the next AV updates, installing another patch that fixes a vulnerability, and feeling themselves victims of change, will wait for the next 'most widespread virus ever' to hit the Internet. That is the forecast for 2002: worms today, worms tomorrow.

Katrin Tocheva
Team Manager Anti-Virus Research

Computer Associates, Australia

2001 was an exciting year for security technology. It meant new products, new viruses, new OS vulnerabilities and new threats. Far from viruses disappearing, thanks to 'new improved' operating systems, we saw completely new types of virus emerging, combining the characteristics of worms, Trojans, Win32 viruses and security exploits. Nimda was a milestone from this point of view.

Of course, every single technical challenge was overshadowed by the events of September 11 and the resulting plunge of world economy. I was at our company headquarters in the state of New York on that very warm and sunny morning and I will never forget the feeling of shock and disbelief we all experienced then.

I think that 2002 will inherit this very unfortunate combination of growing technical sophistication of incoming threats, and increased vulnerability of global networks. On the other hand, every company in the world – as well as every government organization – is now well aware of the potential implications if they don't take their IT security seriously, which creates the right environment for a better collective defence against common threats. We just have to make sure that when people come to us we have the answers ready for them.

Dr. Eugene Dozortsev
Assistant Vice President Research & Development

Sophos Anti-Virus, UK

Phew! We made it to 2002 in one piece. 2001 will be remembered as the year email-aware viruses (Kournikova, Nimda, Sircam, Badtrans) really took off – spreading faster and more furiously than ever before. Sadly it won't be remembered as the year companies realized that blocking files with double extensions at the gateway, or applying the vulnerability patches *Microsoft* released weeks (if not months) before, might be a good idea. Maybe this year, eh?

What will 2002 bring? Well, 'more of the same' seems to be the overriding message. More Win32 executable viruses like Badtrans and Sircam, more worms exploiting vulnerabilities in Web servers like Nimda, Sadmind and the infamous CodeRed, more hyperbolic warnings from AV vendors and self-proclaimed security experts claiming the end of the Internet is nigh. The hacking and virus-writing communities seem to be becoming less distinct and we can expect to see more Remote Access Trojans (RATs) exploiting the increasing number of home users permanently connected to the Internet via cable modems and ADSL. Will this year see the much predicted explosion of viruses affecting mobile devices? It's unclear. The good news is that there hasn't been a new *Palm* virus for over a year – and there isn't one in the wild. Happy holidays and a very good year.

Graham Cluley
Senior Technology Consultant

Symantec, USA

Are things getting worse? Yes. But does this mean virus writers are getting smarter, or more malicious? No. While it may seem to some patently obvious that 'virus writers are getting smarter', they aren't. There are actually four reasons why some threats appear more complex than in the good old days, none of them related to any increase in the intellectual ability of the 'bad guys'. First, there is easier access to the technologies. Second, there is greater access to information about the technologies. Thus, it's easier to learn ways to exploit them. Third, the technologies are more sophisticated overall, so anything done with them will *appear* more complex to the uninitiated. Finally, the design of the technologies themselves facilitates the compromise. These last two reasons are the ones that lead me to say 'things will get worse' in 2002 – 'If you build it, they will come.'

Sarah Gordon
Senior Research Fellow

ESET, Slovak Republic

When I pulled out my copy of *VB* from December 2000 and read over my 'predictions' for 2001, I realized that most of the issues I suggested might happen have, in fact, happened.

The events of 2001, without a shadow of a doubt, will present and imply a breakthrough in our concept of both physical and 'electronic/virtual' security. Perhaps the time has come to stop awhile and contemplate the reasons behind the problems in the distribution of the 'service-packs' for ethics behaviour ☺. The bugs deeply embedded in the ethics are, most likely, the source of all security incidents. I am not, of course, so crazy as to believe that the ethics updates will be applicable to everybody. However, it may not be worth it to go into all the details. This reminds me of an old joke which made the rounds in the communist era: a man comes to a news-stand, buys a copy of a daily newspaper and leaves. He starts browsing the newspaper and is surprised to find all the pages are totally blank. He returns back to the news-stand and asks the seller angrily: 'Where are the letters?'. The seller answers nervously: 'Why do you need the letters if everything is CLEAR?' ☺

Miro Trnka
Technical Director

McAfee AVERT, USA

2001 presented us with many new challenges and other frivolity. This year gave us CodeRed and Nimda, but it started back a bit with that little-known tennis pro, and was followed by a home page that was to die for.

From there Mawanella appeared, which sounded more like a milk virus or something you get from bad beef – I could imagine the headlines: 'Daisy the Cow Fresh Off The Farm Infects Computers round The World.' Over the summer months things slowed as usual, but began to pick up late in the season. This was something I don't recall happening in the past four years and holidays seem to be in harm's way.

What seems to be apparent is that we've turned a corner once again and 2002 will undoubtedly have its challenges, CodeReds, Greens, and Purples.

Vincent Gullotto
Sr. Director, McAfee AVERT

TECHNICAL FEATURE

Generic Detection for Visual Basic Internet Worms

Andy Nikishin and Mike Pavlyushchik
Kaspersky Lab, Russia

Recently, we have seen a growing tendency for virus (worm) writers to write their creations using high-level languages such as C++, Pascal (Delphi), Visual Basic and so on. This trend has placed a strong demand on anti-virus experts to find methods of generic detection for such programs using heuristics.

It is no secret that Internet worms hold the 'number one spot' in all virus-related charts and lists, and Visual Basic is one of the most popular languages among today's worm writers. For these reasons, we decided to start looking into the possibility of generic detection of Internet worms written in Visual Basic.

Starting Point

To determine whether or not a program is an Internet worm, we have to analyse the program's behaviour, determine what undesirable things the program does and how it does those things.

According to statistics, we know that most of the Internet worms written in Visual Basic (VB) spread using *MS Outlook*. The reason for this is that *MS Outlook* represents a COM object and as such can be accessed by any external program. From another angle, Visual Basic makes working with COM objects very easy. It does most of the 'dirty' work of creating and deleting instances of objects, performs binding, takes care of method calling and transferring parameters and, finally, it controls method results and performs error handling. Visual Basic performs these actions 'transparently', meaning that even the program author may know nothing about how it really works.

To detect a Visual Basic program as an Internet worm we will determine whether the program uses *MS Outlook* and, if so, how it uses it. To do this we need to get inside the Visual Basic executable and decipher the structure of Visual Basic's executable file format.

VB File Format Overview

There are several versions of *MS* Visual Basic, but we shall examine only versions 5 and 6, since most of the recent Internet worms have been written with these versions. The executable file format for the two versions is very similar, so we will analyse them as one. The internal structure of files compiled with *MS* Visual Basic differs from those created by other compilers. The file contains not only program code, but also a lot of data that describes the code and which is used at run time.

Usually, anti-virus scanners check program code from the entry point, but in VB files this method is useless. The entry point of a VB file points to a short stub that simply calls a run-time function that never returns:

```
0040273C   push  0004028B4 ; sInitData
00402741   call  000402736 ; MSVBVM60.ThunRTMain
00402746   add   [eax], al
00402748   add   [eax], al
0040274A   add   [eax], al
```

Further program execution is under the control of a run-time library that simply calls the program's procedures from file. To prepare the program for running, the run-time library uses data stored in the sInitData structure. Its pointer is passed to the ThunRTMain() run-time function which initiates the program execution.

A great deal of useful information can be obtained by analysing the sInitData structure and its sub-structures. For example, we can find the name of the project and compiled file, all imported and declared functions, used OCX files, begin and end of the native code stream, structures that describe all modules and forms, and so on.

MS Visual Basic compiler can create two types of executable – 'Native Code', which contains procedures compiled to native Intel x86 code, and 'P-Code', which contains the byte code interpreted by the Visual Basic virtual machine at run time. Of course, each code format is reflected in the sInitData structure in a different way, and needs to be processed separately.

Native Code Analysis

From the sInitData structure we can see that the native code stream or 'segment' lies within the file as a persistent piece. It does not contain statically linked run-time code as, for example, Delphi code does. This means that the stream contains only author-defined code, without any run-time procedures, which only take up valuable time during analysis. So, the analysis range is limited quite strictly by the code stream.

Let's return to Internet worms that use *MS Outlook* to spread. *MS Outlook* represents a COM object with ProgID (OLE Automation programmatic identifier) 'Outlook.Application'. To work with this object the program has to create its instance in some way. For example, it can be done as follows:

```
Set objOutlook = CreateObject("Outlook.Application")
```

Next, the program uses the object instance by calling its methods and properties. Depending on the definition of the

variable that holds the object instance before calling any method or property, Visual Basic performs either early binding (during time of compilation) or late binding (at the run time) automatically.

Late Binding

Late binding is performed if the type of variable that holds the object instance is defined as Object or Variant: Dim objOutlook as Object or Dim objOutlook as Variant.

The VB run-time library has a set of functions for late binding calls. Their names are constructed using 'LateMem' with various prefixes and postfixes: __vba[Var]LateMem[Named][Call][St|Ld][Ad|Rf].

For example:

__vbaLateMemSt

__vbaLateMemCallLd

__vbaLateMemNamedStAd.

We will call these 'LateMem functions'. Each of them receives the name of the calling method (as a string), the number of the method's parameters, the parameters themselves, and (optionally) a pointer for the result value. For those who are familiar with COM technology basics, we can say that all LateMem functions use IDispatch interface. The LateMem function transforms the method name to memberId by calling IDispatch::GetIDsOfNames(), then invokes the method with parameters by calling IDispatch::Invoke(). For example:

```
Set objNamespace = objOutlook.GetNamespace("MAPI").
```

The compiled code is as follows:

```
00401764   sub   esp, 10h
00401767   mov   ecx, 8                ; VT_BSTR
0040176C   mov   edx, esp              ; Param1
0040176E   mov   eax, offset aMapi     ; "MAPI"
00401773   push  1                     ; params count
00401775   push  offset aGetnamespace  ; "GetNamespace"
0040177A   mov   [edx], ecx
           ; Param1.vt=VT_BSTR
0040177C   mov   ecx, [ebp+dummy+4]
00401782   mov   [edx+4], ecx
00401785   mov   ecx, [ebp+var_14]
00401788   push  ecx
00401789   mov   [edx+8], eax
           ; Param1.bstrVal="MAPI"
0040178C   mov   eax, [ebp+dummy+0Ch]
00401792   mov   [edx+0Ch], eax
00401795   lea   edx, [ebp+objOutlook]
00401798   push  edx
00401799   call  ds:__vbaLateMemCallLd
           ; objOutlook.GetNamespace
```

As can be seen, before the __vbaLateMemCallLd function is called, the pointer to the object instance (objOutlook), method name ("GetNamespace") and one parameter ("MAPI") were placed on the stack.

Thus, by going through the code and analysing LateMem function calls, we will find all the late binding calls to COM objects.

Early Binding

Visual Basic performs early binding if the type of variable that holds the object instance is defined as an application defined type:

```
Dim objOutlook as Outlook.Application
```

In this case, the compiled code looks completely different. For the same example, the compiled code will be:

```
00401794   mov   eax, [ebp+objOutlook]
00401797   lea   edx, [ebp+objNamespace]
0040179A   push  edx
0040179B   push  offset aMapi          ; "MAPI"
004017A0   mov   ecx, [eax]
004017A2   push  eax
004017A3   call  dword ptr [ecx+4Ch]   ; GetNamespace
```

Here, the pointer to the method name is not pushed on the stack as a parameter. Instead, the method function is called directly, using the virtual function table (vtable). By analysing this code we can determine which method has been called, based on the vtable offset (in our example 4Ch), but we need to know the interface type to bind the method number with the exact method name.

From the code shown above, we cannot see the interface type, thus it looks as if we have come to a dead end. Fortunately, there is a way out of this situation. If we look at the code just after the method's call, we will see:

```
004017A6   cmp   eax, esi
004017A8   fnclex
004017AA   jge   short loc_4017BE
004017AC   mov   ecx, [ebp+objOutlook]
004017AF   push  4Ch
004017B1   push  offset GUID__Application
           ; {00063001-0000-0000-C000-000000000046}
004017B6   push  ecx
004017B7   push  eax
004017B8   call  ds:__vbaHresultCheckObj
004017BE   ...
```

The __vbaHresultCheckObj() function shows an error message if the method called returns an error value. Let us check the input parameters of this function. The third parameter is a reference to the GUID of the interface called (which, in this case, is _Application) and the fourth parameter is offset in the method table (vtable) – in fact, the number of methods multiplied by four (here we have 4Ch; this value corresponded to the GetNamespace method).

Tracing the __vbaHresultCheckObj() functions shows us all the program's calls of COM objects using early binding. As a result, we are able to find all the calls of COM objects in a program. Moreover it is unimportant what kind of binding was used – late or early. We filter all calls of interest to an *MS Outlook* object to understand the algorithm's interaction with *MS Outlook*. Finally, using the evidence we have collected, we can pass verdict on the program: guilty or not (i.e. worm or not)!

P-code Analysis

During the analysis of P-Code compiled files we find that there is no code executed by CPU (except the entry point).

All procedures are compiled into byte code, which is interpreted, controlled and run by Visual Basic's run-time library. Of course, such code needs different data to organize work with objects, local procedures data, constants, and so on. Therefore, the format of sInitData is slightly different.

In the process of investigating sInitData and its substructures, we look at the module description tables (in fact, these are descriptions of classes). Among other data there is a table of constants that is used by P-Code. Every module has its own table. These constants are references to strings, GUIDs, declared and run-time functions. Note that these tables are not present in files compiled in native code. Of course, this is understandable – all references to constants are already put into executable code. The following is an example of a constant table:

```
004017CC  dd  offset  rtcShell
004017D0  dd  offset  aCProgramFilesN
          ; "C:\\ProgramFiles\\NortonAntiVirus\\*.dat"
004017D4  dd  offset  aOutlook_applic
          ; "Outlook.Application"
004017D8  dd  offset  rtcCreateObject
004017DC  dd  offset  aMapi           ; "MAPI"
004017E0  dd  offset  aGetnamespace   ; "GetNameSpace"
004017E4  dd  offset  aOutlook        ; "Outlook"
004017E8  dd  offset  aGuest          ; "Guest"
004017EC  dd  offset  aPassword       ; "password"
004017F0  dd  offset  aLogon          ; "Logon"
004017F4  dd  offset  aAddresslists   ; "AddressLists"
004017F8  dd  offset  aCount          ; "Count"
004017FC  dd  offset  aCreateitem     ; "CreateItem"
00401800  dd  offset  aAddressentries
          ; "AddressEntries"
00401804  dd  offset  aRecipients     ; "Recipients"
00401808  dd  offset  aAdd            ; "Add"
0040180C  dd  offset  aSubject        ; "Subject"
00401810  dd  offset  aBody           ; "Body"
00401814  dd  offset  aAttachments    ; "Attachments"
00401818  dd  offset  aSend           ; "Send"
0040181C  dd  offset  aLogoff         ; "Logoff"
00401820  dd  offset  a_vxv           ; ".vxv"
00401824  dd  offset  kernel32_OpenProcess_
00401828  dd  offset  kernel32_GetExitCodeProcess_
0040182C  dd  offset  rtcDoEvents
00401830  dd  offset  rtcGetTimer
00401834  ...
```

Here we see that the names of all COM's object methods are present in this table. Even a simple analysis of these strings gives us the opportunity to detect an Internet worm in a program with a high probability. In addition, it is possible to analyse P-Code itself. Such analysis shows us all COM's methods calls as Native code analysis does. However, this variant is more difficult and more laborious and it needs in-depth knowledge of P-Code structure and its additional data, so we shall not examine this method here.

Conclusion

Usually, we finish our articles with a warning, saying that the situation on the virus front goes from bad to worse. This time, however, we can turn our backs on tradition. In spite of the apparent difficulty, it is not difficult to write generic detection procedures to reveal Visual Basic worms, regardless of the code's type. Thus, in this case, we are able to say that the situation has gone from bad to better.

CONFERENCE REPORT

AVAR 2001

Helen Martin

December 2001 saw the fourth annual conference of the Association of Anti-Virus Asia Researchers (AVAR). The inaugural event of the Association took place in Hong Kong in 1998, and it was to this spectacular city that the conference returned in 2001, this time, co-hosted by the Information Security Special Interest Group of the Hong Kong Computer Society.

The single-stream event boasted some 19 papers and two panel sessions over the two days, all papers were presented in English, with simultaneous translation into Putonghua available.

Following introductory addresses from Seiji Murakami, Chairman of AVAR, Sunny Lee, Vice President of the Hong Kong Computer Society, and from Alan Wong, Director of Information Technology Services for the Government of Hong Kong, the papers began with keynote and honorary speeches.

Frisk's Vesselin Bontchev gave a keynote address on the responsibilities of the anti-virus researcher, during which he declared educating users to be a waste of time and that a better idea would be to *force* users to 'behave properly', suggesting that one way to do this might be to get *Microsoft* involved. Conversely, Vesselin advocated education of another kind – that of anti-virus researchers, in an attempt to ease the growing burden on today's experts. He indicated that the number of competent top-level anti-virus researchers is extremely small and one thing that is vitally important for the industry is the education and preparation of new anti-virus researchers.

Jan Hruska, CEO of *Sophos Anti-Virus*, followed with a speech entitled 'Is Virus Writing Really That Bad?'. He concluded that yes, virus writing is always bad, but that the punishment should fit the crime. History was made early on in proceedings as, following a brief exchange after Jan's speech, Drs Hruska and Bontchev agreed to, well, agree.

Zhang Jian of the China Accredited Laboratory Anti-Virus Products Testing and Certification Center reported on the progress in the anti-virus field in China, and there were updates on the current status of cybercrime in Japan and Japan's Information Security Policy, from Masao Tatsuzaki of Japan's National Police Agency Community Safety Bureau and Takashi Kume of the Ministry of Economy, Trade and Industry of Japan, respectively.

Robert Vibert brought more word from the trenches, representing members of the Anti-Virus Information Exchange Network of which he is moderator. Although this

was a continuation of a theme begun just a couple of months earlier by David Phillips at VB2001, Robert's presentation generated a good deal of interest.

Tuesday evening's banquet was a themed event, the organizers having chosen 'Anti-Virus Begins with Education' as an appropriate topic. AVAR Vice President Karen Cheung began with an old saying: 'It is better to light a candle than to curse the darkness'. And to demonstrate her point, the banquet hall was plunged into darkness before she lit a single candle followed by a chain of candles, slowly lifting the room from darkness. Rather than an economy drive, her candles represented education building a stronger force of anti-virus and awareness of information security. Hijacking Karen's analogy, Vesselin Bontchev, heckled from the floor that a technological solution (turning the electric lights back on) would have been even more effective.

Following her address, Karen introduced three young students from a local college who presented some of their work: two games and two short cartoons intended for children with an educational message, teaching the basics of computer security.

The following morning saw a number of delegates bleary-eyed, not from over-consumption of alcohol at the previous evening's festivities, rather as a result of a new malware outbreak having kept their telephones ringing from the small hours. While Badtrans.B was beginning to plateau, W32/Goner.A had stolen the limelight, though this was much to the surprise of many attendees, so lacking are Goner's methods in sophistication.

In her keynote address, Eva Chen, co-founder of *Trend Micro* went out on a limb in presenting the myths of virus protection, confessing that she (and in fact the whole industry) had been lying to the user community for years since the honest truth is that anti-virus products cannot protect against new viruses. She posed the question: has vaccination *ever* worked?

Dennis Longley, Visiting Professor at the City University of Hong Kong, put a whole new angle on matters. He challenged Vesselin Bontchev's belief that 97% of users are stupid, or just don't care, his own view being that 97% of users merely believed that the IT industry would provide them with a system that works. The uncomfortable question left for delegates to ponder was is it 97% of the IT industry that are the idiots?

Kyuchul Han of *Ahnlab* speculated on the future of ASP deployment, using *Ahnlab*'s own anti-virus ASP service, *myV3*, and integrated security ASP service, *Security Clinic* as examples. *F-Secure*'s Katrin Tocheva took us through the different types of Internet worm and their spreading methods, while *Network Associates*' François Paget highlighted that virus authors are wising up to the fact that the anti-virus community is quick to react and thus increasingly looking at ways to spread their creations with maximum speed.

During the final panel session, panel members Dennis Longley (Queensland University of Technology), Igor Muttik (*Network Associates*), David Banes (*Symantec*) and Nick FitzGerald (independent) were asked for their predictions for the future of viruses over the coming year. They predicted an acceleration of what has been seen over the last couple of years, with malware becoming increasingly 'network security vulnerability aware'.

AVAR and EICAR in Mutual Support

As is traditional, the end of the conference was followed by the Annual General Meeting of the members of AVAR. Following discussions between the directors of AVAR and the chairman of EICAR Rainer Fahs, an agreement of mutual recognition and cooperation was signed at the AGM by the chairmen of both organizations as part of a new initiative for a Cyber Defence Alliance (CDA).

EICAR's CDA is a framework designed to encourage any endeavours to create a user-friendly information society, its objective being to combine efforts worldwide to support any initiatives or programmes that would help towards achieving a more secure information society.

This global initiative includes legal frameworks, research, technical measures and organizational cooperation, including worldwide cooperation with anti-virus and security organizations, support of the EC convention on Cyber Crime, and support of research which stipulates enhancements in defence mechanism. The CDA strives toward such goals as global warning and reporting systems, unified naming conventions, government certification and licensing schemes for AV vendors and better education and awareness. Rainer Fahs pointed out that the members of EICAR are aware that this is a tremendous task, and these are ambitious goals. More details will appear on the CDA in the forthcoming issue of *VB*.

Though no larger than previous years' conferences, AVAR 2001 attracted a very broad international spread of delegates and a wide-ranging programme covering technical, ethical and practical issues presented by some of the top names in the industry was well-received by all. AVAR 2002 will be held in Seoul, Korea, with further details to be announced on the AVAR Web site in due course (see http://www.aavar.org/.)

FEATURE

Peer-to-Peer Swedish Style

Jaak Akker
SIG Security, Sweden

Recent statistics show that the 120 largest corporations in Sweden employ a total of 900,000 people. Of these, 500,000 work outside Sweden. The wise AV professional will realize immediately that this equates to 500,000 PCs and as many mail accounts outside the country (since, today, almost every employee has both a PC and a mail account).

Given that the total work force in Sweden is about 4.5 million, the relative percentage of workers and their PCs located outside the country is rather large. Inevitably, this means that virus propagation within the country is enhanced by the large number of mailboxes connected to subsidiaries of Swedish companies all over the world – not to mention the fact that sending emails has become a popular pastime during the dark winter months when in some parts of Sweden the sun never rises!

SIPS SIG Security Malicious Code Committee

Since August 2000, a group of Swedish AV professionals has been meeting on an irregular basis. Given the statistics outlined above, it is not surprising that members of the group represent corporations that, together, have local branch offices or subsidiaries in almost every country worldwide.

It all began with a seminar organized by the *Swedish Information Processing Society's Special Interest Group Security* (*SIG Security*). *SIG Security* is Sweden's largest association of information security professionals. Currently its membership totals 2600.

Members of *SIG Security* cover a broad spectrum of IS professionals, ranging from information processing students to corporate security managers who represent companies with as many as 200,000 employees.

Of course, not all members of *SIG Security* are interested specifically in malware issues. With this in mind, *SIG Security* has instituted the concept of a number of smaller groups for sharing experience in specific areas.

As the association is based on voluntary work, there has to be a driving force strong enough to develop and maintain these groups through the inevitable obstacles that arise when getting two or more people together.

The core motivation and driving force behind these groups

is one of the strongest there is (no, not sex!) – self-interest. As many have experienced, sharing one's own concerns, griefs and laughter with peers is one of the most rewarding ways to interact with others.

The Woes of a Security Expert

Many security experts working in large corporations have become accustomed to hiring a consultant whenever in doubt or dire straits. Anyone who attempted to hire a virus protection consultant when, in the pre-Melissa days, anti-virus matters weren't high on the agenda, would have found it a shocking and deeply disturbing experience. Searching the consultant market for somebody skilled in protecting corporations from malware showed that there was no expertise to hire. And when I say none, I mean *none* – not even for money!

Of course, you could always ask your vendor for assistance, but, in general, vendors' experience is limited to how you should apply their own product.

Managing corporate AV protection is a far more complicated task than simply distributing AV software updates. It is a matter of organizing staff and developing service levels, as well as creating alerting mechanisms, early warning systems and reporting channels for the malware incident response teams.

In addition, managing corporate AV protection is about obtaining and maintaining funding by balancing on the razor edge between failing severely in your malware protection and overperforming in such a way that your budget will be cut for the next year. It is about getting a second opinion on all matters – the architecture, the products, the cost for the staff and so on.

Often, being a security manager is about being lonely with your concerns, fumbling when trying to achieve your goals. Those who specialize in AV protection are isolated still further, as both the target and the methods change rapidly.

The longest period of time during which the malware threat was relatively constant in its nature was between 26 March 1999 and 18 September 2001 (which, of course, is the period between Melissa and Nimda, when preventing mass-mailers was *the* trick of the AV trade) – barely 18 months! To say that it never gets boring is an understatement. Someone once said 'You only live twice: once when you are born and once when you are trying to thwart a major virus outbreak in your corporation.'

All of these problems have a solution: the peer group. Peers share your concerns, they are eager to share their experience, since they know that it means you will share yours with them. Finally, a peer group is an excellent debriefing forum, since there are few other people who understand the unique nature and consequences of malware outbreaks in the corporate environment.

Assembling Peers

So, how does one go about creating a peer group? Well, we were able to read about how the *Anti-Virus Exchange Network* (*AVIEN*) was started in a feature in *Virus Bulletin* a few months ago (see *VB* August 2001, p.14).

However, the *SIPS SIG Security Malicious Code Committee* was started following a lunch-to-lunch seminar on malware. The seminar was organized by *SIG Security* on the classical scheme of 'how to improve things':

- Where are we at now? (It is a mess.)
- Where do we want to be? (We want orderliness in our malware protection.)
- What steps do we have to take to get there?

One of the few vendor-independent anti-virus consultants presented an excellent and thought-provoking talk at the seminar on the nature of viruses.

Working in small discussion groups, participants began to recognize some common problems, which they subsequently shared with the rest of the seminar group.

Of course, not all problems are common. One participant from a military organization told how a malware outbreak in a regiment had resulted in an economic gain, since regular military training had been disrupted. Savings in fuel and staff field service *per diem* were considerable. His question was how do you motivate management to spend more money on AV software? Unfortunately, the participant in question did not return.

Finally, as part of the seminar programme we had invited representatives of the major AV companies in Sweden to participate in a debate on the future of the malware arena. On this occasion it was a little disturbing to realize that they did not know much about the future either!

Meeting IRL

As usual after *SIG Security* seminars, the most valuable aspect highlighted by participants in their evaluations of the event was meeting peers and holding informal discussions in the bar.

After the seminar, a group of participants formed an agreement to meet on a more regular basis. We agreed to meet in real life (IRL). To our taste, meeting IRL has many advantages over IRC and mailing lists. For example, there is the multimedia aspect: when meeting IRL discussion of architectural configuration can be aided by drawing on a white board. Arguments and different angles which could take months to evolve through mailing list discussions can be covered within half an hour.

Another very important aspect of meeting IRL is the degree of certainty that goes with any communication. When meeting IRL, linguistic interjections as well as facial expressions and body language, allow us to interpret what significance we should attribute to a statement. Communication by mail, on the other hand, lacks any of these useful indications.

Our meetings are held a couple of times a year, although additional exchange by mail takes place more frequently. When there is a major malware outbreak, we try to give each other early warnings, however this is not our main aim, nor do we have any explicit rule about this.
The winning formula of *SIG SEC* is the interchange of personal experiences.

In order to give our meetings some structure, the format is as follows: a lecture in the morning, followed by lunch, followed by another lecture – lectures are usually given by someone from the group. Examples of themes for these lectures and the discussions which continue thereafter include, 'The corporate AV architecture' and 'Parameterization of server AV protections software from a performance point of view'.

At our December 2001 meeting we welcomed a representative of *Microsoft* who elaborated on *Microsoft*'s new security initiative. However, the core objective of the group is to share our own experiences. On many occasions, by the end of the meeting our members will have obtained as much information during one day as a consultant would have charged them two weeks-worth of work for writing a report on.

Rules

We have never discussed specific products and vendors during plenar sessions. This is not a formal rule, but I guess that everybody has the notion that such discussions will very easily end up with a 'My vendor is bigger than yours' discussion, from which no one will benefit. We do, however, have a number of rules:

1. We don't write protocols.

First, writing protocols is time-consuming and nobody wants to do it. The other reason that we don't write protocols is that anything written may be lost and found by somebody else. Remember that many of our discussions revolve around weaknesses and vulnerabilities in our respective environments which, from a security viewpoint, would not be sensible to disclose.

We do not want the content of our discussion to reach the daily press either. What would be considered to be the normal crop of viruses on a 'quiet' day would make the

headlines of the tabloid papers if revealed – how difficult would it be explain to the public that viruses that are *caught* before they enter an organization are not a problem? As mentioned at the start of this article, a number of members of the group represent trans-national Swedish companies, and from two State agencies that the media just love to write about. This is also one of the reasons for our second rule.

2. We accept new members only by elective consensus.

We accept new members only by elective consensus of the whole committee ('consensus' is the Swedish way of doing things). There are a number reasons for this. For example, we do not want any AV vendors in the group. This is because a great deal of the information we reveal in our discussions would be highly advantageous to vendors in a licence-negotiating situation.

Furthermore, we want all members of the group to contribute to the discussion. (A couple of years ago I chaired another experience exchange group, several members of which were retirees who seemed to be there just to have a nice chat. Consequently, agendas were spoiled and those members who were genuinely busy stopped coming to the meetings because of the small talk.) Our third rule of membership is also for this reason.

3. Members must work in a corporate environment.

As mentioned previously we want all members to make valid contributions to the group. We feel that running malware protection in a 200-employee company is a completely different ball game from doing so in a large corporation.

A Common Goal

As you may see, our group has many things in common with *AVIEN*. We love the initiative of *AVIEN* and encourage our members to join (at least those who can afford it – these days some businesses won't even fork out enough money to buy a pencil!).

As our group has been run on a voluntary basis, we are currently in mid-discussion as to whether the group should be administrated by one of the member's companies, just as *Segura Solutions* acts as administrator for *AVIEN*.

Like *AVIEN*, we have issued a code of conduct (see *VB* June 2001, p.13). However, *AVIEN*'s code is geared towards the anti-malware professional, whereas our code of conduct is not technical and is intended to be geared towards 'aunt Lizzie in Tallahassee'. We feel that it makes sense for every computer user to take a stand on the issue of ethical behaviour and hope that our code will help to educate computer users in general.

Malware is a concern for the common Internet user. We know that the Internet brings trouble. And if *we* who have the detailed knowledge won't ring the bell in our respective societies, who do you think would?

SHOWCASE

A Confusion of Tongues at Babel?

Frank W. Felzmann, BSI, Bonn
Guenter Musstopf, perComp-Verlag, Hamburg

The naming of viruses has presented a problem since the dawn of the 'virus era'. In 1991 the Computer Anti-virus Research Organization (CARO) published a set of now well-known conventions for virus naming. Rules were defined for the selection of names as was the basic syntax and the terms 'family' and 'variant'.

Nevertheless, only a small number of manufacturers of anti-virus scanners use CARO names in their products and virus descriptions today. At present, most manufacturers use only the notation <name>[.<variant>][.<length>] as well as the prefixes for macro viruses such as W97M/. Due to the fact that the analysis of new viruses is a time-consuming process and that cooperation between anti-virus manufacturers has not always been good, many different aliases have been used.

It is not only the number of viruses and other malware that has grown over the years. New types of virus, such as macro and script viruses, as well as worms, Trojans, backdoors and other malicious software have appeared. The first 'mixed-type' virus was the so-called 'multipartite virus', this was able to infect executable files as well as boot sectors and MBRs.

Current Naming

The intention of the AV scanner developers was to use additional prefixes and postfixes in order to give users some basic information about the features of a virus reported by a scanner. Unfortunately, each manufacturer defined their own prefixes and postfixes (for example: 'Trojan/', 'Trojan.', 'Troj.', 'Troj/' and 'TR.' for a Trojan) as well as delimiters (for example: '/' '.' '_').

Furthermore, a prefix such as 'Multiplatform/' does not provide the user with any concrete information about the platforms under which the corresponding malware will be active. The same holds true for the prefix 'O97M/' because it does not state which *Office 97* products may be infected by such a macro virus.

An additional problem has arisen from the fact that an increasing amount of malware has been developed which falls into different 'types', like the early multipartite viruses. For example, Badtrans.B is a worm as well as a Trojan. Another problem arises with viruses that affect a number of platforms (e.g. CodeRed.C: *Windows NT, 2000, XP* and *IIS*). It is not practical to use all available prefixes

and postfixes in the scanner reports. Therefore, more and more manufacturers are misusing alias names, such as W95/Badtrans.B@mm alias 'I-Worm.BadtransII', 'Trojan/Badtrans.B' and 'Win32/Badtrans.B', in their full text malware descriptions. This is somewhat confusing for many users. For example, a user might search in the database of malware descriptions for 'W95', 'Win32', 'I-Worm' or 'Trojan' and not for 'Badtrans'.

A Meeting and a Proposal

During the *Virus Bulletin* Conference 2001 a brief, unofficial meeting was organized during which a small group of anti-virus professionals discussed the issue of naming malware. The discussion was based on the paper 'Identification of malware from the user's point of view' by Frank W. Felzmann, Klaus-Dieter Moeller and Guenter Musstopf. A shortened version of the paper was made available to all conference delegates.

The paper gives more information about the present pre- and postfixes of malware names along with some examples of present naming. Furthermore, it presents a proposal for the future of malware naming. The basic idea is that three levels of information should be made available:

- malware name (including variant, such as Badtrans.B)
- classification
- full text description.

The malware name is used in the scanner report. The pre- and postfixes are substituted by the classification, but this is not a replacement for full text descriptions, which are stored in a virus database developed and maintained by the AV manufacturers.

Classification

The classification should contain (a minimum of) the following information:

- Type of malware: e.g. file, boot, macro virus, worm, Trojan, backdoor, dropper etc.
- Platform: operating systems or application software (such as *MS Office*) under which the malware is able to work.
- Distribution techniques: e.g. slow-infector, fast-infector, mass-mailing, LAN, WAN etc.
- Payload: e.g. no payload, temporary, permanent, overwriting infected object, physically overwriting, manipulating data files, stealing information, manipulating registry etc.
- Disinfection: information about disinfection (e.g. possible, impossible etc.)

Additional information may be given, such as extension of infected objects, size of code and programming language (e.g. VBS or JavaScript). For example:

Name:	Magistr.B
Type:	worm, virus
Platform:	*Win32*
Programming language:	Assembler
Distribution:	mass-mailing
Payload:	deletes files, overwrites (CMOS, flash BIOS) deactivates ZoneAlarm, manipulates WIN.INI and SYSTEM.INI
Disinfection:	infected files which are locked must be deleted manually.

At present some names are doublets. For example, 'Marijuana' is the alias of the boot virus Stoned as well as Win32/Marijuana@mm. The two names differ only in their pre- or postfixes. This would not cause a problem for the proposed notation. In this case the database will output two (or more) classifications.

The advantage of the classifications database is obvious. If new malware is found, entries can be added easily. If new types of malware appear, the database can be extended without changing the existing database entries.

The Future

There is a lot of work to be done. First, the structure and features of the classification need to be defined. For this purpose an active group of experts – representing both AV manufacturers and users (administrators) – will be established which will attempt to solve this task.

Also, a database for classifications must be developed. The idea is that this database is freeware which can be accessed by all users and companies regardless of which AV scanner they are using. The only investment for the manufacturers is to develop and maintain a utility which maps their names to the standard names required by the classifications database.

We believe that there is a good chance for such a project to succeed. Cooperation between anti-malware manufacturers is currently very good, as indicated by the fact that the number of different aliases for new malware has decreased over the last few years. Finally, we would like to mention that CARO is working on an extended version of the CARO conventions. The target group for the new notation will not be users and administrators but mainly malware research centres and developers of malware scanners.

Currently we are seeking active experts who would be willing to invest some time in getting involved with this project. We plan to organize the first meeting of this group during the *Virus Bulletin* 2002 Conference (New Orleans, 26–27 September 2002). A copy of the above-mentioned paper can be obtained by emailing gm@percomp.de and please send any questions or proposals to the same address; we welcome your comments.

OPINION

Trouble Makers

Andreas Marx, AV-Test.org
University of Magdeburg

It is a fact that no software – besides a single 'Hello World' program – is completely free of bugs or unintended side-effects. This applies not only to operating systems and *Office* applications, but also to anti-virus software.

Today's anti-virus scanners are very complex pieces of software. A small mistake in the program code or virus definitions can result in a small problem, such as a miss of a non-ItW virus, or a much bigger problem such as a crash – which is particularly serious if the program protects a server or Groupware system. Stability is one of the most important issues here: scan all incoming and outgoing traffic, but please don't crash!

History Lessons

In history we have had a few good examples of such crashes, especially on damaged or corrupted files. I can remember a problem in *Dr. Solomon's* anti-virus solution when scanning a 32-byte-long EXE file: only the normal MZ header was available, but the rest of the file was missing. This caused an exception fault in the command-line scanner.

OLE2 files also caused a lot of trouble in the past. The first macro virus scanners used the *Microsoft* OLE implementation, which worked quite well for standard files, but usually caused problems if the files were slightly damaged. Very fast, vendor-own implementations were developed, which included proper error checks to avoid crashes or infinite loops in the internal OLE file structure.

A few years later, Costin Raiu published an article in *Virus Bulletin* called 'The Little Fixed Variable Constant' (see *VB* October 1999, p.8), which discussed OLE documents having a 4 KB block size instead of the standard 512 bytes. A lot of programs simply ignored the files or did not find any virus, which could be fixed in later releases, because it was only an academic issue. However, at least two programs crashed – and that should not happen.

Usually, such malformed documents do not exist in a (relatively) trusted environment, such as within an organization. However, an attacker can send nearly anything to a company using email. These emails could contain viruses, may be malformed and so on, and their content can never be trusted. Usually, these will be scanned at the email gateway or – if this does not exist – in the Groupware environment, such as *Exchange* or *Notes*. And now the files are in the middle of an important, trusted environment, but still they can contain nearly any surprise.

In the Archives

On the Bugtraq mailing list a few months ago there was a posting about archive files like ZIP or ARJ, which contain files with names such as 'NUL.EXE' or '../NAME.EXE' (see http://www.securityfocus.com/archive/1/196965). The author looked at standard unpackers and found many problems: file names with reserved DOS names such as NUL, CLOCK$, AUX or PRN can cause *Windows 9x*-based systems to crash or simply to print out a file during extraction. *Windows NT*-based systems were not immune, but the trouble was limited.

We investigated how virus scanners would react if they found such a file: only one program crashed out of about 30 tested, but only two thirds were able to detect the viruses inside these files. In particular, this happens if they try to extract the file to disk under the name which is stored in the archive, which is not possible. A random name should be used instead, or the file should be scanned in memory – consider memory-mapped files in *Win32* environments or a RAM disk, for example.

However, this is only a small issue. A more interesting method is to embed files in an archive which contains files with names like '../NAME.EXE'. Such archives cannot be created using standard *Win32* tools, but can, for example, under Unix-based systems.

However, I was too lazy to start *VMware* so put a file in an archive with a name like 'XX_NAME.EXE' instead. Later, I changed the 'XX_' to '../' using a hex editor. It should be noted that this has to be done at two positions in ZIP files (simply use search and replace), but at only one location in ARJ archives. And, of course, more than just one '../' can be used for this – I used it up to six times for a test.

Put to the Test

At first, I tested archive programs and observed that nearly all of them were vulnerable and dropped the archived files in nearly every available subdirectory on disk. Using a virus scanner, the situation was much better: no command-line or GUI version seemed to be affected, all ran fine, found the virus and did not drop the files over the hard disk. Even if the program does not scan the files in memory, it has been extracted to a random file name in a temporary subdirectory, ignoring paths.

Next, I looked at *Exchange 2000* and mail gateway solutions – not only anti-virus, but also content filtering programs. Some of the products I encountered used the standard unzip utilities which are not secure. Sure enough, I was able to send an email with an archive and the files within it were dropped to a special location on the hard disk. Also, I was able to overwrite programs like

explorer.exe on *Windows* systems, because the extract process acts with administrator rights. I shan't continue here, but just point out that it was very trivial to hijack such an insecure system within minutes. Of course, I notified the vendors of all programs about this issue (this was in August 2001).

As a quick fix, I suggested putting the temporary (unzip) directory into another partition or onto a RAM drive. Following this, the systems were no longer affected by the problem.

Following notification of the issue, a few of the vendors responded that this problem is rather academic, but after I sent them our test files inside unencrypted ZIP archives (without viruses, but causing small problems on their own mail Gateway), they realized that it is a real problem. :-)

Heavy Nesting

A few large companies suggested that we investigate what happens if an anti-virus program has to scan a heavily-nested ZIP archive.

I obtained a sample file called '42.ZIP'. This was a ZIP in a ZIP in a ZIP etc. – it had six recursion layers, with 16 new files inside every layer (size was about 2 GB in every case), until I was able to see a file called '0.DLL', that contained only a few random (mostly zero) data. These data can be compressed very easily, which meant that the ZIP file was only 42 KB long.

The companies that had suggested this investigation had encountered great problems at their mail gateways – script kiddies had sent such files by email over and over again. Most virus scanners require about two to three days(!) on an Athlon 1,33 GHz system to scan such a file with 100% CPU load.

Therefore, I suggest strongly that an option should be added to all content security programs to set a time-out value (e.g. 180 seconds) for a file such that, if this time-out is reached, the file will be skipped, treated as a virus and quarantined. Also this would help if the scan process crashes unexpectedly.

An alternative would be to limit the maximum size of an attachment to be delivered, but for our '42.ZIP' example this would be nearly useless, since it is already very small. Finally, the number of recursion layers could be limited to a 'harmless' three or four. The user could select the maximum number of layers in the main program according to their requirements.

Testing Issues

We tested these issues in our last *Exchange 2000 SP1* test (the results of which can be seen on the Web site http://www.av-test.org/). Many programs had a problem with 100% CPU load and in at least one there was an option to limit the scan time, but this feature did not work.

Also, one email gateway scanner was unable to detect Win32/Sircam due to the malformed (non-RFC) headers it uses. The attachment was not found and the email was delivered to the clients.

Following the Win32/Nimda outbreak, many vendors added better detection for EML files in their scan engines, but there was also a problem with 'trusted data'.

One solution I tested internally reserved for memory operations only the number of bytes the mail header showed, which caused a buffer overflow if the attachment was longer than expected. Another program crashed if the attachment was truncated or if the MIME structure was corrupted – this can easily happen automatically due to transportation problems.

I can continue with *Win32* runtime compressors – simply attempt to change a few values in the main compressed files and the decompression routine of a scanner shows unexpected behaviour. With a little information about the compressed file structure and about which special tokens have been used, it should not be a problem to create such a file.

Of course, no scanner can find compressed worms or viruses in such files any more and I do not expect this. The only thing that should not happen is a crash of the scanner, neither as a result of a GPF nor a 100% CPU load problem.

Engine Developers Take Note

My suggestion for engine developers is that they should check all input from files carefully, especially all variables that the program uses internally for pointers, to reserve memory etc. I suggest that program developers include features to limit the damage of problematic files, for example setting a maximum scan layer for archives or a simple time-out.

The internal QA should check the behaviour of the scanner specifically on malformed files. This should include white box testing, where the tester knows the internal structure of the program and tries to find problematic routines, as well as black box testing, where the tester modifies a file with the intention of causing problems to the scan engine. This can be done automatically where random parts of a file which is likely to be problematic will be changed or overwritten in a loop, until a problem occurs.

Testers Take Note

For magazine testers, my suggestion is to include malformed files in a test set. It does not help a user if a scanner finds nearly all infected files, has no false positives, and is fast – but it crashes directly or requires 100% CPU load if it receives a malformed file to scan from a non-trusted source, such as the Internet. Currently, I suggest including only easy malformed files, such as the archive files described above. Later, we can add more problematic issues.

INSIGHT

A Mushroom, a Leprechaun and a Pot of Worms

Roger Thompson
TruSecure Corporation, USA

Born in 1952 in Brisbane, Australia, I was a high-school dropout – I'm self-taught. I'm married to Kate, and we have four grown kids, ages 20 to 29, plus six adopted children, whose ages range from 16 months to six years. In addition we have three toy poodles in the household – they're my burglar alarm system!

I've always played guitar or bass in bands, and these days I play mostly in my church band. Kate is a professional singer – in fact, that was how we met. My band was one of the few around that could read music, so we used to get lots of gigs backing singers.

Kate and I have our own recording studio now, and all our grown kids are very musical. We plan to move to Nashville, Tennessee (America's 'Music City') in the future since I'm lucky enough to be able to ply my trade anywhere, as long as I have a decent Internet connection. We're putting together a family act to play some serious music again for a while.

Aside from my music, I have a second-degree black belt in karate, and I'm considering taking up jujitsu alongside my six-year-old adopted twins. They have some minor neurological problems from *in-utero* alcohol consumption, so the structure, discipline and skill training in jujistu will be good for them.

Marketing Lessons

My first virus experience was when I was running a team of *Oracle* contractors in a government department in Brisbane. I had just learned a valuable marketing lesson.

Back in those heady, pre-*Windows* days, someone had brought a copy of a keyboard-enhancing program into the office. The program allowed you to dynamically assign sets of keystrokes to function keys, which meant basically that you could type like blazes. In a matter of days everyone in the department was using the program, gloriously illegally (apart from me, naturally).

Before long a rumour surfaced that *Microsoft* was conducting spot audits of government departments, on the lookout for illegal software. Immediately the department placed an order for about 50 copies of this software, so that they would indeed be legal.

The marketing lesson I learned was that the software's author had managed, probably accidentally, to sell 50 copies of his software in a single order, with zero cost marketing, and zero effort. I remember thinking to myself, 'I need a bit of software that I can sell cheaply and that everyone needs.'

Magic Mushrooms and Serious Thinking

Around the same time, I had heard rumours of some things called 'computer viruses', but I had paid little attention to the rumours. One day, someone put a program called 'mushroom.com' into my autoexec. This was a nice little program which played the advertising jingle of an air-freshener advertisement of the time.

This was in 1986 or 1987, and was well before the time of multimedia and speakers. The program used just the little three-inch PC tweeter, and sounded remarkably good. A few days later, however, someone asked me, 'Have you heard about Mushroom? It's a virus!'

This encouraged me to, well, think about it a bit. Immediately it became obvious to me that the available utilities of the day (*Norton Utilities* and *PC Tools* for example) were good at finding 'lost' data, but not so good at finding data that wanted to 'hide'. Also, I realized that if only I had a checksum of every program on the disk, I would know whether mushroom had changed any of them.

I knew that an engineer in the department, Jack Kenyon, had written a program that made a pass of a disk and created a tree-image of the directory structure. I realized that this program found all the files already, and that 'all' that was needed was to open each program and checksum it.

Virus Busting

I explained my checksum idea to Jack, and said, 'You write it, I'll sell it, and we'll split whatever we make.' Thus was born our company, *Leprechaun Software*, and our product *Virus Buster*.

Mushroom, by the way, proved to be harmless and was simply an early example of a false positive, but it certainly got me thinking.

The *Reader's Digest* version of the rest of the story is that Jack and I did quite well for a while. I moved to the States to try out the big boys on their home turf, and Jack and I agreed to split the company; I would take the US and he would keep Australia.

Once I got to the States, I found out that there's this thing called 'timing' in marketing and it's just as important as timing in music. In Australia, I had been either the first, or very close to the first, but by 1991, the big boys were cranking their conventional marketing handles pretty hard.

PR-driven marketing had worked wonderfully for me in Australia, but simply did not in the States. I understand why now, but that's another story. Seminars were very effective for me, and allowed me to get some very big and prestigious clients, but did not scale well.

Changing Times

By the end of 1995, I figured I'd gone about as far as I was going to be able to go, so I sold the company. That was an interesting lesson too.

I got what I considered a fair price for the company, although it was nothing like the amount some other folks got for selling their companies. The lesson, though, was that the company that bought mine bought another at the same time, for more than ten times the amount they paid for my company.

I had great staff, a really quite solid product, great clients who loved us, and recurring annual revenue. The other company had few real clients, no staff and a not-quite-finished product. However, the other product was in an emerging industry, whereas by that time the anti-virus industry was ten years old, and was already considered mature.

Refocusing

After about a year with the new company, Larry Bridwell at the *ICSA* asked whether I would be interested in becoming the resident virus guru for them. One of the things that I had found difficult about my move to the States was that I had to focus on marketing and selling, and could not put as much time into pure research as I wanted and needed. The *ICSA* move, although it had a couple of downsides, would allow me to re-focus in technical areas.

I never imagined that I would stay in that job for more than a few years, but *ICSA* has proven a fun place to work, with lots of bright people. The security world is a much bigger place than just anti-virus.

ICSA is now known as *TruSecure*, and currently my title is Director of Malicious Code Research. My role is mostly to poke around any new technology, and know what threats are emerging. Of course, much of my spare time goes into *WormCatcher* (see *VB* December 2001, p.4). It's lots of fun.

Looking to the Future

I expect that the future of the anti-virus world remains solid, if somewhat repetitive. To put it another way, as long as there is even one new virus, customers will need something, be it an upgrade, support or whatever.

At the same time, a lot of effort each month goes into some pretty boring analysis of pretty minor variations. I expect that *Win32* programming is sufficiently well understood now that we will continue to see periodic breakthroughs, and periodic outbreaks. The CodeRed stuff was quite interesting.

In terms of virus writers, I think that most of what they do is boring, and I think that surely they must find it so too. Naturally, I wish they'd stop, but I expect that some new technology will become both widespread and homogeneous and we'll be off again in a slightly new direction. Perhaps *Microsoft .NET*?

I think most virus writers are just kids who eventually grow tired of it, and lose interest. Of course, history shows that there are always some new ones to pick it up.

I think that the legal and ethical issues we've debated in the past are largely irrelevant now. They simply pall into insignificance. The events of September 11 are having an impact in our industry as well. I expect that, in the fullness of time, the Patriot Act will probably make it quite uncomfortable for trivial virus writers, but I think it's quite likely that bright young minds somewhere in the world are plotting some form of genuine cyber-terrorism.

These days everyone's using known virus scanners. They're becoming more generic, and better at detecting minor variants, which is a *good thing*.

Every now and then, people come up with something different enough, à la CodeRed, Nimda or Badtrans.B, that it gets away for a while. This will continue until people start learning to harden their system security instead of relying on the AV scanner.

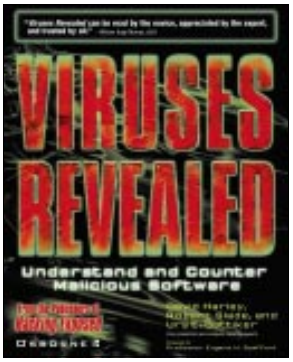
I anticipate that I shall stay with viruses until I think of something new that needs to be on everyone's PC. ☺

BOOK REVIEW

Viral Revelations

Paul Baccas
Sophos Anti-Virus, UK

Viruses Revealed
Authors: David Harley, Robert Slade and Urs E. Gattiker
ISBN: 0072130903
Publisher: Osborne McGraw-Hill
Price: $39.99



This is a large book, and somewhat unwieldy as a result. It is not the sort of book you can pick up quickly to read on the train. In fact, taking it with you will require that you reorganize your luggage rather than just slip it into your bag. However, its size is this book's only intimidating feature, with its title in nice large and friendly letters.

The book is divided into four sections (Parts I–IV) plus appendices. Each of the parts covers a different aspect of the skill set necessary for anyone calling themselves an 'Anti-Virus Consultant'.

Stylistically, the book is rather scholarly, written in a conversational style that is reminiscent of some postgraduate course books before they reached the hard mathematics. Fortunately, this book is free of mathematics and irrelevant source code. Individual Chapters can be read as separate papers, given some knowledge of the subject, as little reference is made chapter to chapter. However they make more sense when read within the respective parts of the book. The format is nice and straightforward, as seems to be the style with *Osborne*'s books, a paragraph or two at the beginning of each chapter indicating what is to follow, and a summary at the end of each.

Part I, 'The Problem', answers the big questions and this alone is what most other virus books try to be. It begins with the fundamental definitions of various malware, or at least a union of the definitions. Then a little history of the viruses is given, demonstrating the cyclical nature of virus technology.

Part II, 'System Solutions', deals with a subject that has not been covered well by other publications, except as short papers. However, this section of the book covers the whole gamut of the job description for an Information Security Anti-Virus Specialist. Explanations range from types of anti-virus software and how and where to deploy them, to cleaning up if the solutions have failed, criteria for testing your solutions, reviewing 'independent tests' and the most difficult of problems, dealing with the intractabilities in every network (problems between the keyboard and chair).

Part III, 'Case Studies: What Went Wrong, What Went Right, What Can We Learn?', is brave in its scope and attempts to teach by example. This section is further split into what can be described as 'Problems Pre-Macros', 'Problems with Macros' and 'Mass-Mailers'. In a few broad strokes each major virus incident is covered, giving the reader a deeper understanding of the problem.

Part IV, 'Social Aspects', describes issues surrounding viruses that are less quantifiable. There is no technological panacea for the social problems of virus writing. Malware exists on secure operating systems and the only way to stop it is to prevent people writing it. Only by taking a holistic view of both social and technological issues will the threats be diminished.

Finally, we have the appendices, glossary and index, the appendices being a combination of three distinct FAQS available here in printed form.

As mentioned, the book is rather long, and the editors have split it into five separate and distinct sections. It would be reasonable to assume that Part III of the book will be that most in need of updating – indeed, that section ends with a description of Badtrans found in April 2001. Since then the world has seen more case-study-worthy pieces of malware: Badtrans.B, Magistr.B, Sircam, Goner and, of course, Nimda are all different in their own inimitable way. Should the whole book be updated in 12 months just to add these case studies? The update plans for the book are unknown at this time, however I can see a perceived need to update some Parts (such as Part III) more frequently than others.

As malware writing techniques change so does the response to the malware – it is not a static field, yet the format in which this book was published is static. I would have preferred each section to have been produced as a separate book (sold together). Each section is a reasonable size and, as separate books, would have been easy to read anywhere. Updating sections would then have been a simpler task too.

This aside, I would suggest you do one thing: read the introduction. If you feel that the introduction is talking to you, then you will certainly not be disappointed in the rest of the book.

In some other place and time, no doubt someone will commission other books on this subject. However, when those froods at *Megadodo House* update their friendly publication, the entry listed under 'Computer Viruses and Assorted Malware' will read 'See *Viruses Revealed* by David Harley *et al*'.

PRODUCT REVIEW

Trend Micro ServerProtect 5

Matt Ham

[*As a result of technical difficulties, the second part of last month's review of RAV AntiVirus for Sendmail and RAV AntiVirus for Desktop Linux has been postponed. Instead, Matt Ham takes an in-depth look at Trend's ServerProtect. Part 2 of the RAV review will appear in the February 2002 issue of VB - Ed.*]

Trend Micro has the peculiar honour in Oxfordshire, the home of *Virus Bulletin*, of being the only anti-virus company to feature in advertisements on the back end of local buses – buses which serve the UK headquarters of both *Network Associates* and *Sophos Anti-Virus*. *Trend*, however, is a company whose origins lie in Taiwan, with a pair of head offices in the US and Japan to cover the world's largest pair of anti-virus markets.

The traditional area of specialization for *Trend* has been server-based and gateway products and thus a server-based product (along with the associated administration tools) was chosen for this review. Other *Trend* products include virus scanners for both servers and workstations controllable through the *Trend Virus Control System* management tool and a variety of more general malware- and content-scanning products. Platforms and applications covered by *Trend*'s products include various *Windows* incarnations, *HP-UX*, *Linux*, *Solaris*, *Sendmail*, *Lotus Notes*, *Exchange* and *Celerra*.

The Package

Full marks go to *Trend* for the speedy delivery of the product, which arrived by courier less than two hours after agreeing on which software would be reviewed. Not surprisingly after such a brief time in transit, the sturdy box was intact on arrival. Its contents proved to be a sum total of two objects: a manual and a wallet of CDs.

The manual is truly a tome to be contended with. Oddly enough, it has no page numbers, but a rough estimate would put it at in excess of 300 pages despite the fact that it is applicable only to the *ServerProtect* product line and those management tools associated directly with it.

The CD wallet contained no fewer than three CDs, emblazoned with the label *Trend Micro Enterprise Solution.* The need for so many CDs is explained by their content being pretty much the entire *Trend* product range in English, Chinese and Japanese.

The first of the CDs is the only one equipped with an autorun feature. After initial language selection, this leads onto a menu. Featured here are 'About the CD' (which seems to be a general overview of *Trend*'s products), the install area for software on the CDs, 'Product Information', 'Information About Trend Micro and its Worldwide Offices', 'Trend's Virtual Lab', a 'Security Information Center' and the 'Trend SolutionBank'. Information about *Trend Micro* is located locally on the CD and is informative if not gripping in content. Information contained under the other headings is provided by links to Web sites.

Installation

Installation of *ServerProtect* was performed with relative ease, with just a few caveats. Upon selection of installation, it was required to install *Microsoft ActiveX Control Pad* first – an application which inspired more than a little trepidation since it relates to the use of the potential malware-related horrors of HTML, VBScript and JavaScript. Once this had been installed, the installation of *ServerProtect* was initiated once more, and proceeded smoothly.

The default option is to install the *ServerProtect* Management Console, together with the necessary files not only for scanning but for network administration of *ServerProtect*.

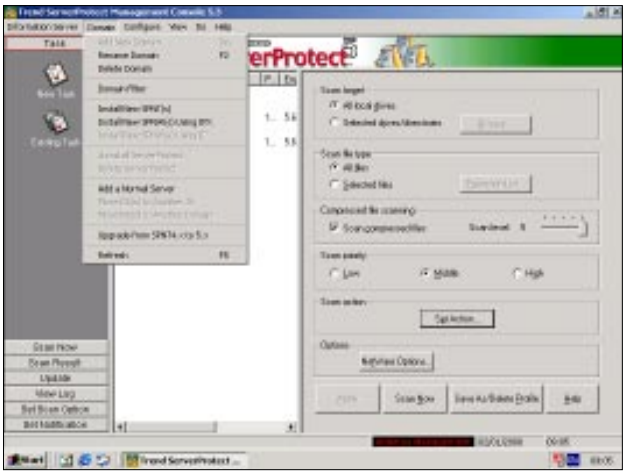
The network bias of the software is such that, for this default installation, a network path to the local machine must be present – experimentation showed that a stand-alone, non-network-aware machine could not be installed with *ServerProtect*.

Features

Being a specifically networked application, it is apparent that management of *ServerProtect* will be of great importance. This is a matter taken seriously by *Trend*, to the extent that, for a default first installation, the scanner cannot really be accessed outside the *ServerProtect* Management Console. This is considerably more network-oriented than the majority of default anti-virus installations, and a little daunting at first.

The general layout is the increasingly common left-hand bar of general commands influencing the layout of a larger right-hand pane. Drop-down menus above this more GUI-like part of the program add further control.

Before entering into the details of what is offered on the various parts of the interface it is necessary to give a quick overview of how the hierarchy of *ServerProtect* is structured. This is neither hugely complex nor surprising: there are two types of server involved – Information Servers, which act as central control centres for groups of Normal Servers termed 'Domains'. Since these are all servers there is a further, relatively invisible, layer of client machines which will, in most cases, be connected to these Domain machines.



The use of the word ‘domain’ here is a little misleading, since an Information Server is not restricted to controlling a *Windows* domain, rather it can control any collection of Normal Servers.

In the product under review both *NetWare* and *Windows NT/2000* servers are supported at the Normal and Information Server level. The Management Console offers control of *NT/2000/NetWare* and Information/Normal servers from centralized points, though it is limited to being installed on *Windows* machines only.

It is the fact that the Management Console can be used to control a selection of Information Servers and, through them, Normal Servers, that makes this more complex than most interfaces commonly encountered on AV software.

Moving onto specifics, the drop-down menus are the first port of call. These are ‘Information Server’, ‘Domain’, ‘Configure’, ‘View’, ‘Do’ and ‘Help’. Of these, ‘Help’ is the easiest to skim through, containing the standard help function, the unusual help function *for* the help function and the standard ‘About’ option for product information. ‘View’ offers views of the deployment log, general log, scan logs and virus encyclopaedia, the last of these through a Web link.

Drop-down Menus

The other drop-down menus are more complex. The ‘Do’ menu offers several options which will be covered later, since they are duplicated on the left-hand control bar. In addition, it is here that product serial numbers (effectively the registration codes) and passwords may be altered. Of great use in large networks is the ability to search for a specific Domain or Server by name and in addition there is a feature here to allow connection to machines which do not appear to be responding normally to requests for information.

A further collection of commands is linked to direct communication with *Trend*. These cover submitting suspicious files, creating Debug Info in case of problems, sending feature requests and submitting an on-line registration form. Knowing the nature of some anti-virus administrators (and testers for that matter) I can but pity the recipients of the mails produced by some of these features.

The ‘Configure’ menu is, again, largely a repetition of commands that are present on the side-bar area, with the addition of a control for the frequency with which the tree structure for Servers is refreshed.

The ‘Information Server’ menu is of more interest, allowing the selection of an Information Server and the viewing of the log file for that particular machine. Also, it is possible to move an Information Server (IS) totally, allowing for hardware replacement without the loss of Server functionality or data.

Since the IS has important data associated with it, such as defined tasks or Domain members, it is possible to back up and restore IS Data. This process can be automated so that it occurs whenever either of these parameters change or as a scheduled job, which might be preferable if large numbers of tweaks are performed by way of experimentation.

The last of the drop-down menus is associated with the administration of Domains and the Servers within them. As might be expected, Domains may be added, renamed or deleted here and viewed through a filter in the main GUI for added ease of use.

Once the Domains have been configured, machines may be added to them, with the allocation of Normal Servers to Domains and supporting ISs possible at will. For those machines selected *ServerProtect* may be installed on *NT* machines and on *NetWare* machines – with the latter having the option to perform this update via either IPX or IP. In reverse, it is also possible to uninstall or delete *ServerProtect*. For backwards compatibility, this is the area where *ServerProtect* can be upgraded from an older version.

It should be noted that not all of these functions will be available immediately for *NetWare* machines. In order to install upon a *NetWare* machine there must be at least one other *NetWare* server installed. At first this seems like a Catch 22 situation, but the installation of an initial *NetWare* machine may be performed through the original installation procedure. Having performed this once manually, other *NetWare* machines can be added through the Management Console. In a rather stunningly counterintuitive fashion, however, this first install is performed by selecting the *Windows* installation package and selecting a *NetWare* server as a target from within this program.

Side-bar Menus

As mentioned, several of the menu functions are duplicated on the side bar, the next area to be inspected. Again, this is divided into smaller portions for easy digestion, namely ‘Task’, ‘Scan Now’, ‘Scan Result’, ‘Update’, ‘View Log’, ‘Set Scan Option’ and ‘Set Notification’.

It is fairly obvious what 'Task' relates to, as is the case with most of these titles, but the details can bear further examination. On a default installation of *ServerProtect* there are three predefined tasks – 'Deploy', 'Scan' and 'Statistic'. In combination, these update and upgrade *ServerProtect* daily, scan every Friday and produce a .CSV data file once a month, giving details of virus detection statistics. This is a good minimum setting, but a great deal of customization may be performed by editing these tasks and adding new types. In addition to the three task types mentioned, there are tasks defining real-time scan settings and the purging, exporting and printing of logs.

One note should be taken in that the real-time scanning task is not the only way of controlling this activity, but it does allow for the fine control of on-access scanning, allowing for different settings to be imposed at different times in a scheduled manner.

'Scan Now' is an area which will be familiar to all readers since it is the immediate on-demand scan area which is, in *Trend*-speak, the 'Scan Now' task. By default, all files on the selected server are scanned, including compressed files with up to five levels of nesting and at high priority. The default settings for action are to disinfect with a back-up being made or to quarantine files which are detected as uncleanable. In addition the usual set of alternative actions – delete, rename and ignore – are offered, and quarantine directories may be defined.

If viruses are found the information is displayed as part of the next area, 'Scan Result'. Here, statistics are displayed for Real-time, Scan Now or Task-related scans. The files listed as infected in these results may be purged or acted upon in ways other than the defined Task defaults if required.

The first real complaint in this review is with the 'Update' section, where both updates and rollbacks may be performed. Here, updates include both the regular virus definition data and the more irregular *ServerProtect* software upgrades. However, upgrades are only available easily over an Internet connection. In a corporate environment, of course, this is unlikely to present a problem. From a *Virus Bulletin* point of view however, things are not so easy, since the virus-filled test set directories cannot be allowed anywhere close to such access to the outside world.

For such a sealed system – not unknown in the production departments of software developers – the requirement if a software upgrade is required is to use the whole new setup program, rather than a smaller patch file, and the former is not a small piece of code. This should, however, only be needed in the case of major engine revisions rather than the interim upgrades – i.e. the interval between such upgrades should be measurable in months or years rather than weeks.

Since this operation is performing an update on the Information Server portion of the *ServerProtect* hierarchy, the download of such updated information is only half the story. Deployment to Normal servers can also be manually initiated from this location – useful in cases where a new threat must be countered immediately.

As discussed, the 'View log' portion of the side bar is a repetition of part of the drop-down menu system. Items in the log file for each server are of the expected variety – infections, update attempts, temporary service halts, task information and more are covered and can be viewed through several filters, printed, exported or purged or the statistics displayed.

The last two sections of the side bar are where the greatest degree of control over basic on-access scanning and notification occurs and are thus descriptively labelled 'Set Scan Option' and 'Set Notification'. Since it is covered in the 'Scan Now' section, there is no control of default on-demand parameters at this point. Default actions are the same on access as on demand, with the same alternative actions of delete, quarantine or ignore.

For reasons of overheads the depth of compression which is scanned is set to only a single level. By default, all incoming files are checked, though outgoing files may be selected as an alternative or both may be scanned.

Various options are available for boot sector scanning which, by default, is triggered on shutdown and on access. The option to disable 'MacroTrap' is present here too – although it is not made very clear what, exactly, MacroTrap is, it looks very much like it is a form of macro virus heuristics.

Exclusion lists are configurable by file or directory, which omits drives. However, these areas can also be configured within the Task editor, which might be more convenient when drives or entire machines are considered. On a more security-related note, it is possible to deny writing into specified directories either as a blanket order or by file type.

Alerts are the last on this menu and fall into two categories, 'Standard' and 'Outbreak'. Again, Normal alerts are not exclusive to definite viral files or out-of-date definition files but allow alerts for attempts to change write-protected files as defined in the write-denial features, changes in configuration and changes in NLM status.

Outbreak alerts are defined as those triggered by reaching a defined threshold of virus detections within a defined time scale. Otherwise, these are little different from the other alerts, all of which may be configured to be delivered via the standard methods of delivery – message boxes, printing, pagers, email, SNMP and *NT* event logs. With *Motorola* – the largest producer of pagers in the US market – having ceased, recently, to produce pagers, this might soon become a legacy technology, perhaps to be replaced by SMS alerts.

Deployment and Updates

The matter of deploying updates was mentioned earlier, but a brief return to the subject is in order. In more detail there

are three download types of note. The virus definition files are available in two flavours – the lpt1xx and lpt9xx families – which begs the question why there should be two different pattern files available for the same product. The answer is that this is a legacy feature: the lpt1xx definitions became standard in October 2001 and the second variety are provided to support older products. On the arbitrary day chosen for download, the files were 2.23 MB for both lpt183.zip and lpt983.zip.

As an aside, there are three different platform segregations as far as platform is concerned; the Wireless format has three different pattern files associated with it, Unix systems have a pattern file of their own, and the rest use the one just described.

Engine updates have also been noted before – in this case the engine itself could be upgraded by a file totalling 455 KB. Had the upgrade from the 4 series engine to 5 series engine been required, this would have been a somewhat less pleasant download of 21.9 MB.

For the purposes of the test a simple pattern and engine update was performed from the *Windows 2000 Professional* machine which was in use as an Information Server. This machine was not connected to the Internet, and thus local updates were attempted by specifying a UNC location for the update files. Unfortunately this was somewhat more easily said than done, with the UNC format used not being accepted by the updater.

Further investigation showed that, in fact, this was not a failure at all – what appeared to be an error message was actually an informational message on UNC format. Quite why this appears after the correct information has been provided remains a mystery.

Ignoring this oddity, the download was rather sluggish considering that the files were being 'downloaded' on a local machine. After being downloaded the files are stored on the Information Server and can then be deployed to Normal Servers.

At this point the process became simple once again: selecting 'Deployment' brought up a tree view of those servers which might need to be updated. The default is to update all, though individual machines or domains may be removed from the process if required. Deployment performed smoothly and quickly on both *Windows2k* and *NetWare 5.1* machines, as did the rollback process when this was selected.

On inspection of the files used for virus signature updates within the Information Server storage, they appeared to be the downloaded files (renamed). Therefore, it might be possible to find a way to bypass the built-in download, though this was not investigated.

Engine updates were stored similarly, though there were some extra files in the downloaded version which did not appear in the stored IS package.

Documentation

The sizeable manual included in the package was used in the installation and configuration of the product rather more than expected, due mainly to the initial need to understand *Trend*'s terminology of server types. To its credit, the manual and help files explained this clearly, with very little need to search about – an impressive amount of thought seems to have gone into compiling the index. Similarly, the confusing nature of installation to an initial *NetWare* server was explained in the manual in such a way that it was a simple procedure to follow, without an unnecessary degree of detail.

Online context-sensitive help was used frequently within the program to acquire more exact knowledge concerning the command being examined. Again, this was good overall, though if nit-picking, could have been slightly more feature-specific since the context in this case was often entire pages listing many commands.

Web Support

Cunningly, *Trend Micro* registered the Web domain http://www.antivirus.com/ some years ago, a helpful URL for the forgetful user and this is, in effect, their US site. In addition there are various country-specific Web sites, mostly of the URL-type 'www.trendmicro.xx', all of which can be accessed from the home page of the US site. By and large, the content on the local sites is a mirror of the main site, with differences only, for example, in employment opportunities.

Usually the Web site's home page is dominated by an advertising banner and on this occasion it was devoted to the declared market share of *Trend* as reported by a recent survey. Below the banner are a selection of press releases and to the right of it some predominantly business-related resources continue the domination of business matters on the page. Of more interest is the left-hand portion of the

page – where links to the downloads and virus information sections of the site can be found.

Since downloads have been discussed already, the focus lies upon the online data resources available which include general virus and anti-virus information, online scanning, real-time geographically-divided data on virus reports and hoax information. Real-time data is available for inclusion on third-party websites.

The area most likely to be used by administrators of a server-based system are those concerning hoaxes and recent additions to the virus horde. Hoax information is a little on the skimpy side and does not include a 'most recent hoaxes' section which would be useful in pre-empting hoaxes or for speedy reference.

The virus information centre is much more impressive, offering direct links to information on the most recent or real-time prevalent malware of the moment. This set of links includes Trojans but the real-time links have not for a long time had the quirk of including the EICAR test file. Each description on this page has two tabbed sub-pages, one of which contains a general description while the other contains more technical information on the nature of the beast. Usually this contains detailed information concerning registry changes made by the malware in question, in case complete reversal of such activities is required. There is a direct link to this page from the product CD.

Other useful information available as a Web service includes those items linked to from the installation CD. On a background reading front, this has links to white papers, general news information and press releases as well as PDF-format information on individual products. The 'virtual lab' looked like an interesting concept, offering a trial of various *Trend* configurations over the Web – unfortunately this suffered from a broken link.

A more useful link is that to the *Trend* SolutionBank which is a searchable database of technical support-related FAQS and their answers. There are currently (if the identification numbers of these FAQs are consecutive from zero) over ten thousand such support-related reports in this database.

Performance Tests

Since a *Trend* product is an intended entrant for *Virus Bulletin*'s next comparative review (to be published in the February 2002 issue of *VB*), tests were not run for detection capabilities on this occasion. The area investigated was kept to performance issues and for this purpose the standard *Virus Bulletin* clean test set was used for both on-access and on-demand testing.

When scanning on demand the scans were performed locally upon the clean sets with the same configuration as is used in comparative testing. Scan times here are thus compared with average times recorded in the last *Win2k* comparative (April 2001), though the time lapse since that review should be considered when viewing these figures.

Initial impressions were good – the speed of scans was high in the context of those comparatives. The clean set was scanned in 87 s (comparative average 320 s), the zipped clean set in 70 s (comparative average 180 s) and the OLE set in 6 s (comparative average 20 s). An exception was with the zipped OLE files in the test set which, at 29 s (comparative average 25 s), approached the averages for this test set.

The default setting for local scanning is high priority, and CPU usage did approach 90 to 95 percent during these scans. Setting a medium priority reduced this to 60 to 70 percent, taking 84 s, while low priority seemed much the same CPU usage as the medium setting and also took 84 s.

The test set is composed entirely of executable files – and since the default setting within *ServerProtect* is to scan all files on-access, this testing might be expected to show slightly higher overheads than in an average real-world scenario. With this in mind, the value of around a 75 percent increase in time to copy the executable test set is still rather on the high side. For OLE files the increase was in the region of a more respectable 30 percent, with the removal of MacroTrap having no noticeable impact on the data transfer rate.

Conclusion

The first time that I came into contact with a *Trend* server product was some five years ago, at which time its combination of network management tools and scanner was revolutionary. In the intervening period several products have arrived which offer integration of the two functions, though this is still the most tightly integrated of those which have been reviewed in *Virus Bulletin*.

As far as ease of use is concerned, the most problematic part of this product very much centres around the stumbling block of preconceptions of how a scanner should operate. The network-centric nature of the product leads to certain unexpected methods of scanner control but this learning curve is soon conquered. Performance issues were an apparent weakness, with overheads on executable scanning being on the high side. Other than this single complaint, overall *ServerProtect* acquitted itself well in this test.

Technical Details

Product: *Trend Micro ServerProtect Management Console 5.3*, *ServerProtect for Windows 5.630* and *ServerProtect for NetWare 5.600*.

Developer: *Trend Micro*, 10101 North De Anza Blvd., 2nd Floor, Cupertino, CA 95014, USA; tel +1 408 257 1500 or +44 1628 400 500 (UK); email sales@trendmicro.co.uk; Web http://www.trendmicro.co.uk/.

Price: For 25 users £401; for 100 users £1335.

Test Environment: Two 750 MHz AMD Duron workstations with 128 MB RAM, 8 GB and 4 GB dual hard disks, CD-ROM, LS120 and 3.5-inch floppy, running *Microsoft Windows 2000 Professional*. 500 MHz AMD Athlon server with 64 MB RAM, 6 GB hard disk, CD-ROM and 3.5-inch floppy running *Novell NetWare 5.11* with *Service Pack 3*.

SUBSCRIPTION RATES

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

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

Editorial enquiries, subscription enquiries, orders and payments:

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

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

US subscriptions only:

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

Tel (781) 2139066, Fax (781) 2139067

END NOTES AND NEWS

The Black Hat Windows 2000 Security Conference takes place 7–8 February, 2002 in New Orleans, LA, USA, focusing on the security issues created in the *Windows* environment. Training sessions take place 5–6 February. For more information, including details of future Black Hat events, see http://www.blackhat.com/.

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

Cost-Effective Risk Management for Information Security takes place at the Café Royal, London, 19–20 March 2002. Blue chip corporate case studies and expert organizations will examine the strategic issues surrounding cost-effective risk management for information security. For more details tel: +44 207 368 9300 or visit http://www.iqpc.co.uk/GB-1759/ediary/.

The 2nd Security Audit & Control of Information Systems Conference and Expo (SACIS) will be held 19–20 March 2002 in Istanbul, Turkey. Topics will include Internet/Intranet security, computer crime, denial of service attacks, forensic investigation, intrusion detection and email security. For more details email svs@svs.com.tr or visit the Web site http://www.smartvalley.net/sacis/.

Information Security in the Age of Terrorism takes place 25–26 March 2002 in Washington, D.C. Hear from a stellar faculty about the latest threats to information security and how to combat those threats. For more information visit http://www.frallc.com/, or email sldowt@aol.com.

Information Security World Asia 2002 will be held 16–18 April, 2002 in Singapore. The show will include an exhibition, and a number of interactive workshops. For further information visit the Web site http://www.isec-worldwide.com/isec_asia2002/.

Infosecurity Europe 2002 will run from 23–25 April 2002 at London's Grand Hall, Olympia. Over 40 free seminar sessions will run over the three days, explaining some of the key security issues facing organizations today. For more details visit the Web site at http://www.infosec.co.uk/.

The Southwest CyberTerrorism Summit, to be held 4 May, 2002 in Dallas, TX, USA, will feature presentations from both hackers and industry security experts. Topics include wireless hacking, cyber-attacks, information warfare, privacy, computer viruses, industrial espionage and identity theft. For more information visit the Web site http://www.DallasCon.com/.

Infosecurity.de 2002 and 2003 have been cancelled. This year's show was to have taken place 14–16 May 2002, in Düsseldorf. The organizer, *Reed Exhibitions*, cites a lack of interest due to the unfavourable economic situation as the reason for the cancellations. For more details see http://www.infosecurity.de/.

Information Security World Australasia 2002 will be held 19–21 August 2002 in Sydney, Australia. For full conference and exhibition details see http://www.informationsecurityworld.com/.

***Virus Bulletin* is seeking submissions from those wishing to present papers at VB 2002** in New Orleans, USA, on 26 and 27 September 2002. Abstracts of approximately 200 words must reach the editor of *Virus Bulletin* by Friday 22 February 2002. Please send abstracts (in ASCII or RTF format only) to editorial@virusbtn.com. For details of sponsorship opportunities at the conference, please email vb2002@virusbtn.com.

Ostis Software* has announced the shipment of *AVStripper, a stand-alone hardware product that uses *Trend Micro*'s scanning engine to prevent viruses from penetrating the corporate network. The unit is installed between the Internet and the network and will automatically request virus definition updates from *Trend*'s server. For more information see http://www.ostis.com/avstripper/.

F-Secure* has introduced a number of new features in its *Anti-Virus for Internet Mail 6.0. The latest version of the software allows administrators to define what types of email attachment are allowed to pass through firewalls and/or email servers. Attachments may be stripped based on file type or file name. In addition, *F-Secure Anti-Virus for Internet Mail 6.0* allows the use of a local user interface for monitoring the status and statistics of the product, helping evaluation and installation of the product and making configuration on small networks easier. See http://www.F-Secure.com/.