ISSN 0956-9979 DECEMBER 2002

THE INTERNATIONAL PUBLICATION ON COMPUTER VIRUS PREVENTION, RECOGNITION AND REMOVAL

IN THIS ISSUE:

• **Dream catchers:** As the number of computer users frequenting Internet mailing lists rises steadily, Juha Saarinen looks at the security risks associated with mailing lists and concludes that they are a virus writer's dream. See p.8.

• **Dial my number:** Although intended originally as an efficient method of performing transactions online, porn dialers have become the tools of Internet fraudsters. As their prevalence increases, Andreas Marx asks if we should classify porn dialers as malware, and whether detection of dialers should be incorporated into security products. See p.12.

• **Your cheatin' heart:** When your anti-virus software reports it has blocked viruses, but your users claim their systems have become infected, whom should you believe? Is it possible that both parties are reporting the truth? Joe Wells unravels a very convoluted problem. See p.15.

CONTENTS

COMMENT

“ I must admit I did not envisage a multi-billion dollar AV business back then. ”

The Bigger Picture

I have a confession to make. In 1992 a friend of mine lent me all his back issues of *Virus Bulletin*. Very quickly I became hooked on the magazine. Before you attempt a guess I should let you know that my confession has nothing to do with worms of any kind …

Although my English was good enough to understand the footnote of each page of the magazine, I was not quick enough to read through all the issues in the short space of time available. Thus an illicit plan to copy all the three years of back issues entered my mind. Before I knew it I had copied all the magazines and returned them at the last minute to my friend (who obviously did not expect such reprehensible behaviour from me!). And now you know – I have confessed!

During the relatively short history of *VB* the content has changed a great deal. Do you remember the list of ‘known PC viruses’ with detection strings? How about the list of ‘known Mac viruses’? Oh dear, it’s been a while! I have a very clear memory of visiting the post office with a few hundred packaged diskettes to send out my anti-virus program’s quarterly (!) updates. The girl behind the glass looked back at me over the small mountain of diskettes and screamed while the line behind me stretched out onto the street. I must admit I did not envisage a multi-billion dollar AV business back then. My view of the AV industry has changed considerably over the years. It has been a great journey. I have seen it all, from freeware to shareware and from small to major corporate business. User requirements have changed a lot during these times, as has my understanding of the bigger picture. So what did I learn?

The same heuristics that appear to work with thousands of happy customers might not be an acceptable solution where there are millions of users. False positives (‘FP’s as we call them in the lab) might be generated when you start to scan millions of files. While your FPs with a thousand happy customers will never reach the visibility range, even a minor FP could cost you a fortune where there are millions of users who can become unhappy in the blink of an eye.

Some AV vendors might prefer to pack all the features into the scanner: we emulate DOS, *Windows* and the Internet. Where will be the end to all this? How much more code and data can be packed into our products? It is hard to say! An assembly-written scanner used to be a cool thing. ‘Speed is everything’, you say. Well hold on there, Tommy. Today’s diversified networks need integrated solutions from the desktop to the server on a variety of domains. IA32 is a good thing to support, but can the product run on *Itanium*? How about running on *Solaris* systems or AS400 to name just a few? *Virus Bulletin* tests have not covered such platforms but it would be interesting to measure these capabilities. And the assembly-written engine might not be able to resolve your problem. In this ‘everything, everywhere’ scenario you need portable solutions that can perform the same way both on demand and on access, and on a variety of platforms. It is interesting to see how many products already show differences in scanning performance on access and on demand.

The one thing you will always need is reliable detection. Reliable detection takes more than a good scanner that might work once in a while. You need a very stable solution. If a scanner does not handle the polymorphic and metamorphic threats very well, what can you expect in an emergency situation? Is it good enough to catch up with such detections six months or a year later? I do not think so. However, this argument often makes me want to tear my hair out as I work towards providing a standard response time. As Alan Solomon used to say ‘the virus lab always feels like being on a treadmill’, there is absolutely no time to waste.

In this business things happen so quickly! When we were all talking about macro viruses there was already something else knocking at the door: Win32 viruses. Now that we talk much more about Win32, there is already a set of new threats at the door: exploits built into computer viruses. These new threats need integrated security solutions. Yet more interesting times are ahead!

Péter Ször, Symantec Security Response

VIRUS BULLETIN **©2002 Virus Bulletin Ltd, The Pentagon, Abingdon, Oxfordshire, OX14 3YP, England. Tel +44 1235 555139. /2002/$0.00+2.50**

NEWS

Addendum: *Windows 2000 Advanced Server* Comparative Review

In the November 2002 Comparative Review *Trend's ServerProtect* was reported to have failed to achieve full detection of ItW virus samples and thus was not given a VB 100% award (see *VB* November 2002, p.23). The review stated, '*Trend*'s offering suffered from slightly dated virus definitions. A definition update was promised, but did not arrive.' Following further investigation, however, it has come to light that fate conspired against the developers at *Trend* who sent the update at the exact time that *VB* was suffering a mail server outage. Mail servers rectified, the updates were re-sent, installed, tested and we are happy to announce that *Trend ServerProtect 5.35 1047* earned a VB 100% award, having detected all samples in the ItW set – including W32/CTX.A – and generated no false positives. We apologise to *Trend* for the problems ∎

Who's There?

Last month a new security portal was unveiled by the publishers of *Information Security Bulletin* magazine. Alongside the generic security news stories, the odd back issue of the magazine and marketing blurb, is an intriguing section of the website known as '*Who's Who in Information Security*?' – effectively an online database of résumés.

A search of the database on the word 'virus' produced the biographies of nine eminent figures of the AV world – in alphabetical order, Niels Bjergstrom (Editor of *Information Security Bulletin* and brains behind the website), Vesselin Bontchev, Fred Cohen, Howard Fuhs, *VB*'s own Jakub Kaminski, Eugene Kaspersky, Andreas Lessing, Igor Muttik and Rob Rosenberger. Anyone upset by their omission from the database is reminded that the database is still under construction, and can take comfort in the thought that future developments to the site promise the facility to submit your own biography for inclusion in the *Who's Who* database, as well as the addition of mugshots of each entrant. Next stop: online anti-virus dating service? See http://www.isb-online.net/ ∎

Paying the Price

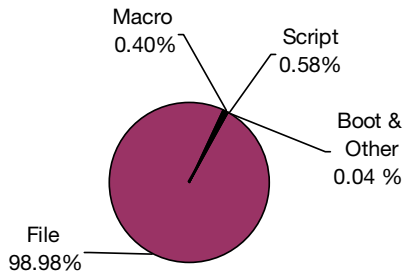
McAfee Security has become the latest security company to issue a press release estimating the potential costs to businesses of 'the next big virus attack'. Research carried out among members of the UK's Federation of Small Businesses led *McAfee* to conclude that small and medium businesses in the UK could lose up to £2.1 billion and 2.2 million 'office days' in downtime. Despite being aware of the risks, 12 per cent of survey respondents admitted they had no virus protection ∎

Prevalence Table – October 2002

Virus	Type	Incidents	Reports
Win32/Bugbear	File	12724	68.66%
Win32/Klez	File	4389	23.68%
Win32/Magistr	File	360	1.94%
Win32/Yaha	File	268	1.45%
Win32/SirCam	File	108	0.58%
Win32/Opaserv	File	99	0.53%
Win32/Nimda	File	87	0.47%
Win32/BadTrans	File	71	0.38%
Win32/Hybris	File	69	0.37%
Redlof	Script	60	0.32%
Laroux	Macro	36	0.19%
Win32/Higuy	File	23	0.12%
Win32/Onamu	File	17	0.09%
Win95/CIH	File	17	0.09%
Win32/Funlove	File	15	0.08%
Haptime	Script	13	0.07%
Kak	Script	13	0.07%
LoveLetter	Script	11	0.06%
Win32/Elkern	File	11	0.06%
Win32/Frethem	File	11	0.06%
Divi	Macro	7	0.04%
Marker	Macro	7	0.04%
Win32/Surnova	File	7	0.04%
Win95/Spaces	File	7	0.04%
Others [1]		103	0.56%
Total		18533	100%

[1] The Prevalence Table includes a total of 103 reports across 63 further viruses. Readers are reminded that a complete listing is posted at http://www.virusbtn.com/Prevalence/.

Distribution of virus types in reports



LETTERS

What the User Wants?

In 'Best Practice or Wishful Thinking?' (see *VB* October 2002, p.2), Phil Wood, *Sophos*, UK, writes: '…is it reasonable to expect any manufacturer to ensure that many millions of lines of code are free from the sort of error that leads to security vulnerabilities when the consumer appears more interested in features over security? Software manufacturers are only delivering what the user wants.'

Rubbish!

If Phil Wood talked to users, he would know that, in general, they do not want new features; they just want to go on using the system with which they are familiar. The new features are thrust upon them by the software manufacturer as an excuse for forcing them to upgrade, to ensure the continuity of the manufacturer's revenue stream.

The first step in this process is to ensure that each update produces documents which cannot be read by the previous version. Then the manufacturer withdraws support for the previous version, and insists that all new computers be supplied with the new version, so that the laggards soon find they cannot read documents sent to them by their colleagues.

In short, the manufacturers have us by the throat, and are robbing us of every penny they can extract. They know that users do not understand security, so they don't waste their time or money worrying about it. Nor do they care about fixing bugs; the bugs provide them with an excuse for bringing out yet another incompatible update. And they don't want the updates to be compatible with the previous version; if they were, there would be no pressure on users to update.

Roger Riordan
Cybec Pty Ltd, Australia

Friends Indeed

On 24 October 2002 the support desk at *Sophos* – and, no doubt, those of other anti-virus vendors too – received a barrage of phone calls and emails requesting more information about the 'Greeting Card' virus. At first it seemed as if there was a new virus doing the rounds, meaning that – as usual – we would inform our customer base that an update to our software was available and expect the reports to subside.

But this time it was different. None of the customers reporting the 'virus' seemed to be sending us a file attachment for analysis. Furthermore, there was no malicious script embedded in the email. It looked just like the sort of regular message you might receive if there was a greeting card waiting for you.

However, if you followed the link to the website mentioned in the email, you were invited to install an application onto your computer. Two lengthy end-user licence agreements (EULA) were displayed, the second of which stated that by installing the application you were giving permission to send a similar greeting card to all addresses found in your *Outlook* address book. Ouch! You probably didn't want to do that, but is it really viral behaviour?

FriendGreetings doesn't replicate itself, so it's neither a virus nor a worm. It's just a harmless email message until the user tries (and gives permission) to run code on the website.

So, is it a Trojan horse? Well, its Panamanian creators *PermissionedMedia* could argue that, since it doesn't do anything that users aren't expecting ('you did agree to the terms and conditions, right?'), it isn't a Trojan horse either. However, many users will not realise what it is about to do – regardless of whether they click on the 'Yes, I'm not paying attention' button, so it *does* do something that many people are not expecting.

Furthermore, FriendGreetings installs code on your computer which may prevent some applications from working properly.

The anti-virus community got its knickers in a twist with FriendGreetings – unsure as to whether their protection products should detect it or not, and if they did detect it, what to refer to it as. No one was keen on calling it something it wasn't, and potentially opening themselves up to peer ridicule or legal action from *PermissionedMedia*.

Doubtless there are those who have seen the confusion amongst the anti-virus vendors, and may be tempted to use similar tricks in future virus releases.

For their part, anti-virus vendors should ensure their products are sufficiently flexible that every unwanted piece of code running on a computer is not simply labelled a 'virus' or 'worm', but allow more granularity.

This would help anti-virus vendors provide a means of dealing not only with the likes of FriendGreetings, but also with some of the more controversial remote access tools – treated as useful applications on some networks, and unwanted security threats on others.

It would be interesting to hear what other readers of *Virus Bulletin* (both vendors and customers) think about the best way to combat issues such as these.

It's important that the products of anti-virus vendors mature to allow this kind of flexibility, but let's not forget that if you want to get angry with anyone about FriendGreetings, the people to direct your fury at are the guys in *PermissionedMedia* who used such an underhand sneaky trick to scoop up thousands of email addresses.

Graham Cluley
Sophos, UK

An Opportunity Not to be Missed?

I think the article 'Melissa Creator Vacationing at Club Fed' by James Wolfe (see *VB*, June 2002, p.2) is a bit too harsh, with violent language and an over-hypothetical view. Viruses are an opportunity for us all to learn how insecure our environment is, and how dumb our users are, not to point guns at the creators.

Wajid
Cerrado Support Services, UK

The Author Responds …

I have to admit that I am a little confused by these comments. Why would you want your network to be compromised so that you could be shown where the vulnerabilities are? A prudent administrator (at least one who actually qualifies to be a network administrator) actively reviews his network architecture to ensure that it is secure.

Following Wajid's line of reasoning, should a person crash their car to see if the seat belts and airbags work? Too harsh an example? OK, how about this one. Should you turn on all the ports on your firewall just to see which ones are going to be exploited?

It is unethical to allow something to infect a network purposely, just to test its security or to test the intelligence of its users.

I did admit that my opinion was somewhat militant in the article and that it was *my* opinion. Why shouldn't blame be pointed at the virus writers? Their creations, whatever the social engineering that causes users to execute them, are still the cause of the damage. They should be held responsible for their creations and should be given a dose of punishment that better fits the crime.

James M. Wolfe
Independent Researcher, USA

Do you have an opinion to air? Why not get your anti-virus woes off your chest and send your letters to comments@virusbtn.com.

CALL FOR PAPERS

VB2003 Call for Papers

Virus Bulletin is seeking submissions from those wishing to present at VB2003, the Thirteenth Virus Bulletin International Conference, which will take place 25–26 September 2003 at the Fairmont Royal York hotel in Toronto, Canada.

The conference will consist of 40-minute presentations in two concurrent streams, Corporate and Technical. Provided below is a list of suggested topics elicited from attendees at this year's *Virus Bulletin* conference. This list is by no means exhaustive and papers on these and any other AV-related subjects will be considered.

- Threats relating to *XP*, *.NET*
- *Linux* security issues
- Java and ActiveX controls
- MS Palladium
- 802.11 wireless LAN access, IRDA and Bluetooth as infection vectors
- Security issues relating to PDAs
- New methods of engine design
- New methods of testing AV products
- Heuristics
- Hacking/AV convergence
- Other security topics/technologies and their connection/integration with AV
- How to find vulnerabilities in your network
- Implementing an enterprise-wide computer emergency virus response program
- Anti-virus policy
- Educating users on security
- Balancing freedom and security
- Virus-naming schemes
- Hoaxes and their impact
- Instant messaging, data correlation and centralised reporting
- Encryption – how are organisations and vendors preparing for the challenges posed by encryption
- Virus/worm honey pots
- What motivates virus/malware writers

Virus Bulletin also invites you to send suggestions for any particular speakers you would like to hear from at VB2003. Please send speaker nominations, along with details of why you would like to hear the speaker (for example, are they an excellent presenter, is their field of research of particular interest, do they have very strong or controversial opinions?) to editor@virusbtn.com.

How to Submit a Paper

Abstracts of approximately 200 words must reach the Editor of *Virus Bulletin* **no later than Friday 21 February 2003**. Submissions received after this date will not be considered. Abstracts should be sent as RTF or plain text files to editor@virusbtn.com

Authors are advised in advance that the deadline for completed papers selected for the conference programme will be **Friday 6 June 2003**. This deadline cannot be extended. Full papers should not exceed 6,000 words.

VB Conference

Over its 12-year history, the *VB* conference has become a major highlight of the anti-virus calendar, with many of its regular attendees citing it as *the* anti-virus event of the year.

The VB conference provides a focus for the AV industry, representing an opportunity for experts in the anti-virus arena to share their research interests, discuss methods and technologies and set new standards, as well as meet with – and learn from – those who put their technologies into practice in the real world.

While the conference remains concentrated entirely on computer viruses and malware threats, the delegates range from dedicated anti-virus researchers to security experts from government and military organizations, legal, financial and educational institutions and large corporations worldwide.

For details of the sponsorship opportunities available at VB2003 please contact Bernadette Disborough at vb2003@virusbtn.com or call +44 1235 555139. Further information about the conference, including online registration, will be available from the *Virus Bulletin* website in due course, see http://www.virusbtn.com/conference/.

VIRUS ANALYSIS

Crack Addict

Frédéric Perriot
Symantec Security Response, USA

Over the last couple of months, there have been many reports In the Wild of a new share-crawler with a peculiar propagation method, W32/Opaserv.A. Its originality resides both in its use of an exploit to infect *Windows* shares, and in its cryptic payload.

W32/Opaserv.A is a worm that spreads to network shares of machines running *Windows 95*, *98*, *98SE* and *Me*. Instead of using the usual WNet* API functions to access network resources it communicates with machines through the underlying SMB protocol. SMB, which stands for Server Message Block, is the file and printer sharing protocol used by all versions of *Windows*. The reason for using SMB directly is that the worm can craft special SMB packets to exploit a flaw in the password checking mechanism of *Windows 9x*/*Me* and thus gain access even to password-protected shares.

Installation

When Opaserv is executed on a machine it copies itself to the Windows directory as 'ScrSvr.exe' and modifies the Registry so that it is run on every reboot, by creating a value named 'ScrSvr' pointing to ScrSvr.exe under the HKLM\…\Run key. Additionally, if it is not already being run from its Windows directory location, it creates a value 'ScrSvrOld' containing the path of its current instance, executes ScrSvr.exe and terminates itself. The newly created instance of the worm deletes the 'ScrSvrOld' value and the file that it points to, leaving ScrSvr.exe in the Windows directory as the only worm executable on the machine.

Then Opaserv attempts to create a mutex named 'ScrSvr31415' and terminates itself if the mutex exists already, in order to avoid multiple worm processes running at the same time. To go unnoticed on the system, the worm attempts to resolve the RegisterServiceProcess() API, calls it if it is available, and lowers its process priority.

In fact, not only does it resolve RegisterServiceProcess() dynamically but it also imports it from KERNEL32.DLL. This bug makes the worm incompatible with *Windows NT*/*2000*/*XP*, whose kernels do not export this function. Furthermore, the worm does not run on machines that do not have the Winsock 2 library installed because it imports functions from WS2_32.DLL.

After its installation process Opaserv starts looking for machines to infect over the network. It creates two threads for communicating over UDP, one sender and one listener.

Network Enumeration

The first thread is an infinite loop that scans class C-sized blocks of IP addresses on the NetBIOS name service port (UDP/137). For all IP addresses of the local host except the loopback address, the block of the host and the ones immediately above and below it are scanned. (For instance, if the host's IP is 192.168.200.300, then the networks 192.168.199.0/24, 192.168.200.0/24 and 192.168.201.0/24 are scanned.) Then 100 other random address blocks are scanned using a strategy that aims to eliminate unused segments of the IP address space in favour of live segments.

The scanning consists of sending a node status request on port UDP/137 of each machine. If a probed machine is running the NetBIOS name service, it replies with a node status response including information about its network resources, such as its NetBIOS name and group, and whether it shares files or a printer.

The worm listens specifically for node status responses announcing share servers. Whenever such a response comes in, it grabs the NetBIOS name of the server from the received datagram and spawns a new thread which creates an SMB session with the server.

O[pen Sesame]

Opaserv uses SMB over NetBIOS over TCP/IP, the SMB flavour compatible with *Windows 9x*/*Me*. This means it opens a connection to port TCP/139 of the server, then sets up a NetBIOS session, and all subsequent SMB traffic happens on top of NetBIOS.

SMB is a rather complex protocol and has several dialects that differ somewhat in the features they offer. Usually when two machines start an SMB session, they begin by exchanging a pair of setup messages to negotiate an SMB dialect. The agreed-upon dialect determines the set of SMB commands the peers will be able to use.

However, Opaserv does not bother with the SMB session setup. All the SMB commands it needs for the purpose of replication are simple primitives such as OPEN, CLOSE, READ and WRITE, so it simply uses the core protocol, skipping the dialect negotiation phase.

The first and most important SMB command that Opaserv sends to the server is a TREE_CONNECT_ANDX. This command is used to obtain access to a share. Its parameters are the name of the share and an access password. If the command succeeds, the server returns a 'tree connect identifier' (TID) to the client. Then the client uses the TID in its subsequent commands to access files on the share.

Windows 9x/*Me* offers a form of protection for shares called share-level security. This is a mechanism based on

password checking where anybody who knows the password to a share can access it, regardless of their identity (as opposed to the user-level security in *Windows NT/2000/XP* that allows fine-grained protection based on user or group identity). Unfortunately this mechanism is flawed in default install versions of *Windows 95*, *98*, *98SE* and *Me* and it is possible to access a share even without knowing the entire password.

This vulnerability was discovered two years ago by *NsFocus* and was announced in *Microsoft* S*ecurity Bulletin MS00-072*. *Microsoft* provided a patch for the vulnerability in October 2000. All *Windows 9x/Me* users should apply this patch to protect their systems.

The problem is located in the system file vserver.vxd which implements the share server under *Windows 9x/Me*. The routine that compares the password of a share with the password provided by the client checks only a number of characters specified by the client. If the client specifies a one-character password then only this character is compared with the first character of the share password, and in the case of a match, access to the share is granted to the client.

This is because the server expects the client to send a null-terminated password and to specify a length including the final null byte. Well-behaved implementations of SMB clients follow this rule, but it is easy to create a packet manually that does not follow the rule. The security update provided by *Microsoft* checks simply that the password provided by the client is indeed null-terminated before comparing it with the share password.

Opaserv exploits this vulnerability by sending a series of TREE_CONNECT_ANDX commands, trying all one-byte passwords between 0x21 ('!') and 0xff, until the connect operation succeeds. It specifies '\C' as the share name, hoping to gain access to the C: drive of the attacked system.

Propagation

Once it gains access to the remote share, Opaserv copies itself to the share under the name 'WINDOWS\scrsvr.exe' by issuing a CREATE SMB command, a series of WRITE commands and a CLOSE. These commands have the usual semantics their names imply. Each of them includes the TID obtained by TREE_CONNECT_ANDX.

Then it downloads the file 'WINDOWS\win.ini' from the share to the local file 'c:\tmp.ini' by issuing OPEN, READ and CLOSE SMB commands, adds the value 'c:\windows\scrsvr.exe' to the 'run=' key of the '[windows]' section in tmp.ini, and uploads the modified tmp.ini back to the location 'WINDOWS\win.ini' on the share. This will cause the worm to run when the machine is rebooted, thus closing the infection cycle.

Windows 9x/Me systems that share their C: drive in full-access mode are at risk of being infected, either if the share has no password, or if the share has a password but the patch for the password vulnerability has not been installed. Read-only shares cannot be infected whether the system is vulnerable or not.

The vulnerability affects only the password-checking mechanism, not the mode of access. If the worm attacks a read-only share it will successfully crack the share password and the TREE_CONNECT_ANDX command will succeed, but the next CREATE command will fail and the worm will be unable to copy itself to the share.

During tests in a lab environment *Windows NT/2000/XP* systems could not become infected across the network. Besides the obvious mismatch in the *Windows* installation path for *NT* and *2000*, the TREE_CONNECT_ANDX command sent by the worm always failed.

Self-update

Opaserv has the ability to update itself over the Internet. It continually checks the website 'www.opasoft.com' for the availability of a newer worm version. If one is available, it downloads it from the website and replaces the local copy of the worm with the new executable. Fortunately, this website has been shut down and the worm is no longer able to update itself.

DES Incarnate

From the 15 kilobytes of assembly language that make up the code of Opaserv, only about 40% are dedicated to replication. The other 60% implement an interesting payload. The code appears quite obscure at first glance, but further study reveals that it is a distributed DES cracking agent. DES is the venerable and ubiquitous Data Encryption Standard, an encryption algorithm developed by IBM in the 1970s and which was the official American encryption standard for decades.

The DES code in Opaserv was not written by the author of the worm himself. Instead, he ripped an optimized assembly DES implementation written by Svend Olaf Mikkelsen and Remi Guyomarch from the distributed.net client. Distributed.net is an organization that participates in challenges to crack cryptographic algorithms and specializes in leveraging the CPU resources of large numbers of hosts over the Internet. They offer clients for download so that any person can contribute a bit of computing power to their agents network.

By combining the power of hundreds of thousands of computers working in parallel, it becomes possible to break an encryption algorithm using brute force, by performing an exhaustive search of the key space to recover the key. (It is feasible only for algorithms that do not have too long a key. DES is a good candidate since it uses 56-bit keys which are considered small nowadays.)

The author of Opaserv attempted to build a similar agents network composed of worm instances. Each instance of the

worm connects to a central website (the same site that is used for the worm self-update feature) and gets a block of the key space from a script named 'scheduler.php'. It also gets a plain text and a corresponding cipher text from the scheduler. Then it tries all keys in its block of the key space until it either finds the right key that encrypts the plain text to the cipher text or exhausts all the keys in the block without finding the right one. The worm instance then reports status to the central site, sending back the right key if it was discovered, and gets new data to crack.

Since the website has been shut down, this feature is now disabled, but it does not necessarily mean that the author's attempt has been a failure. It depends largely on how long the site was up and how many instances of the worm were able to report results to it.

The last DES challenge took place in 1999 and successfully brute-forced DES in just 22 hours, using the CPU cycles of approximately 100,000 machines. (A special-purpose DES cracker also participated but it contributed less than half of the total computing power.) Since then, PCs have become much faster, so it's not too far-fetched to expect all instances of a very widespread worm to be able to crack DES in a few days.

Conclusion

Whether Opaserv succeeded or not is a mere detail. The point is that one must assume that a single, unscrupulous individual with close to no funding can leverage the resources of a huge numbers of vulnerable machines around the world.

Recent statistics suggest that W32/Opaserv.A is very widespread, probably second only to Klez.H and Bugbear. Does its use of a two-year-old password-cracking exploit account for its success, or do people simply have open shares with no password-protection at all? It features a distributed DES cracking payload a little more inventive than the usual backdoor/DoS. Who cares? Let's use triple-DES! [*Or AES, the successor to DES - Ed.*]

W32/Opaserv.A

Aliases:	W32.Opaserv.Worm, W32/Opaserv.worm, W32/Opaserv-A, Win32.Opaserv, WORM_OPASOFT.A, Worm.Win32.Opasoft.
Payload:	Distributed DES cracking.
Removal:	Delete the worm executable, restore the win.ini file and fix the Registry run key. Apply the relevant patch. Do not create shares without a password.
Patch:	Available from the *Microsoft* website. http://www.microsoft.com/technet/security/bulletin/ms00-072.asp.

OPINION 1

Internet Mailing Lists – A Virus Writer's Dream

Juha Saarinen
Independent industry commentator & technical writer, New Zealand

Perhaps due to the rising popularity of open-source software, Internet mailing lists are being frequented by an increasing number of computer users. Mailing lists can be an excellent way to learn about a number of different subjects, as well as receive quick and (sometimes) helpful peer support.

However, even on some of the hard-core UNIX mailing lists – for example, those dedicated to OpenBSD – you will encounter users who are running *Windows* operating systems. This means that you, and all the other subscribers to the mailing list, are exposed to email-transmitted viruses and other such malware, as well as the effects of the 'cures' for them.

In other words, running a mailing list can be a major headache for administrators, and subscribing to one can be a massive nuisance to subscribers.

An Anti-Virus Headache

In November 2002, the W32/Braid.A worm was sent out to all subscribers to a *Kaspersky Labs*' email newsletter. The anti-virus company claimed that its web server had been 'hacked' (though there has been speculation that the worm was sent to a list address, which forwarded the message to all subscribers).

Even if you're sceptical about *Kaspersky Labs*' explanation of the unfortunate mishap, there can be no doubt that the incident was embarrassing for the company and it provides a good illustration of how mailing lists can be abused to disseminate malware, whether accidentally or wilfully.

Kaspersky Labs can seek some solace in the fact that it is not the first organisation whose mailing list has been used as a malware vector.

A Small Virus in a Big Pond?

In January 2001, *Telstra Bigpond*, the largest Internet service provider in Australia, sent a message to some 300,000 customers on a mailing list. Unfortunately, one of the recipient systems was infected with W32/Hybris (see virus analysis *VB* January 2001, p.6).

The virus attached itself to the message, and sent itself back to the *Telstra* mail server, which dutifully forwarded the

infected message to all list subscribers. *Telstra* had to take the drastic action of closing down its entire email system to clean out the virus.

Excellent Vectors

If you think about it, mailing lists are to the virus/worm writer what 'smurf' and 'zombie' hosts are to the cracker executing a distributed denial of service attack: a simple way to amplify damage immensely.

Unfortunately, anti-virus vendors play into the hands of malware writers as we shall see later, thus (unintentionally) becoming part of the problem.

Given that mailing lists can act as 'excellent' vectors for malware dissemination, it would be a safe guess to assume that they will be targeted for future attacks. There are still plenty of lists that allow postings from non-subscribers, but even on those that don't, a single virus-infected subscriber can wreak plenty of havoc.

Many mailing lists are echoed to Usenet newsgroups and/or archived in various locations around the world. Some of the Usenet gateways and archivists make an effort to remove infected messages from the archives, but checking every single one is very difficult, thanks to the enormous volumes in question and encoded binaries being corrupted (e.g. through missing MIME headers) which could hamper detection.

The large number of 'virus found' notifications spoil the quality of Internet archives too. For evidence, do a simple *Google* Groups search on a common virus notification string, such as 'Antigen found virus' – there will be plenty of hits.

The Real Fun

The real fun starts when the malware-infected messages hit the defence mechanisms implemented by well-meaning administrators.

Yes, it's courteous to inform the sender that her/his system might be infected, but when the sender is a list Mail Transfer Agent (MTA) pumping out messages to scads of subscribers, your anti-virus solution has just become a weapon in the malware writer's arsenal.

The problem here is that on any given mailing list, subscribers' messages will go through several MTAs with anti-virus protection. Some of them are sensibly configured not to send out virus alerts or bounce malware attachments and potentially harmful active content to senders, but in my experience, they are a minority.

Even if the mailing list in question is set up to de-MIME messages so as to strip out attachments and HTML content, it's not always enough to stop the nuisance bounces. On some mailing lists, you cannot use the names of viruses, mention IFRAME tags, or even use the word 'virus' without receiving a shower of bounces from various anti-virus products in return.

This sort of hair-trigger response is reminiscent of Mail Marshal message content filtering, which made areas such as the North Lincolnshire (UK) town of Scunthorpe and the county of Middlesex unmentionable for email senders.

Now, imagine a miscreant finds a list with a number of easily-offended anti-virus products protecting subscribers, and engages in a spot of return address forgery: there's no need even to send a virus to the mailing list in order to create chaos, just insert, for example, 'IFRAME' into the message body.

Weeding

As with other mailing list scourges such as out-of-office auto-responders, sensible administrators will try to weed out automatically dispatched notifications from anti-virus protected MTAs before they hit subscribers.

However, this is easier said than done, since there is no standard format for the notifications, the message headers they use, or even whom the messages should go to – some anti-virus products send notifications to the list admin address, others to the list itself, the 'infected' subscriber, or both. You're looking at plenty of unnecessary extra work for the mailing list administrator.

Furthermore, with mailing lists that require subscriber verification before posting is allowed – a good idea to stop spam and list-bombing – admin role accounts can get filled up with mailed bounces and automated alerts from subscribers' anti-virus solutions.

Proposal

I would like to propose an industry-wide ban on sending out 'virus found' notifications from mail servers which filter for malware (or worse, which bounce infected messages).

Thanks to the large number of Internet-borne malware, the mailed virus notifications serve no practical purpose (apart from displaying vendor braggadocio). On the contrary, notifications often create collateral damage and increase 'Internet noise levels'.

As I'm coming to the end of this rant against indiscriminate anti-virus alerts my mailbox has started to fill up with notifications from ISPs and email providers, claiming that my computer is infected with W32/Nimda. It seems a spammer with a DSL connection is infected, and his ratware is forging my email address into the message headers. Sigh.

[*Are you fed up with an inbox full of automated alerts? Do you support Juha's campaign to ban virus notifications from mail servers filtering for malware? Virus Bulletin would like to know what you think! Send your opinions on the subject to comments@virusbtn.com.*]

OPINION 2

Comments on Viral Behaviour Blocking

Lixin Lu
Indefense, Canada

In his presentation at the Twelfth Virus Bulletin International Conference in New Orleans this September, Cary Nachenberg, chief architect for *Symantec*, gave a nice summary of the administrative benefits of Viral Behaviour Blocking (VBB) technology. Coming from a leading proponent of AV scanning technology, this is a clear sign to the AV industry that VBB technology should be reviewed and developed further as part of the AV solution.

At the same conference Andreas Marx, of *AV-Test.org*, reported the results of his retrospective testing of heuristic scanning. His report indicated that heuristic scanning produces an average rate of file virus detection of less than 15 per cent. AV signature scanning results typically show detection rate percentages in the nineties.

I believe that the best possible results will be achieved by integrating VBB into a virus protection system, along with signature scanning and heuristics.

Acceptance

Although Viral Behaviour Blocking technology has been in existence for more than 15 years, it has never before received such a degree of acceptance as that which it is experiencing now.

The previous lack of acceptance was largely due to the fact that VBB solutions have inherent technical limitations, such as a high rate of false positives, that can become annoyances to impatient or inexperienced users.

The new found interest in VBB solutions comes as a result of modern viruses and worms that are spreading at high bandwidth Internet speed; too fast for AV scanners to remain fully effective. Despite the technical difficulties of implementing VBB, it is now considered a useful weapon on the virus battlefield.

Use of VBB

The most effective use of VBB technology requires a well-planned design and implementation. For example, most modern viruses (as well as Trojans and worms) modify at least one of the *Windows* system Registry settings, allowing the program to run and propagate.

A VBB product must therefore monitor sensitive system Registry settings in order to detect any malicious modifications. Once a viral modification to a Registry setting is detected, the VBB program blocks any further actions from the offending process and notifies the user, preventing the infection from spreading.

However, if the virus spreads before changing the Registry setting, the VBB program must still be able to stop it from spreading. To accomplish this, the VBB product assigns a restriction level to each hard-coded rule.

Unrecoverable or non-reversible actions, such as sending out email, are assigned a high restriction level, while actions that are recoverable or reversible, such as modifying a system Registry setting, are assigned a lower restriction level.

The VBB software blocks the viral program immediately when the high-level restriction rule has been violated and restores the lower restriction level Registry setting modification.

System Performance

The effect of VBB software on system performance is another technical aspect of product design that requires careful planning. Poorly designed VBB products can have a huge impact on system performance.

Unlike signature scanning products, which can be implemented on the application layer (Ring3), VBB products are implemented primarily at the system level (Ring0), on a real-time basis. If properly designed, the impact of VBB software on system performance can virtually be eliminated.

Reducing the ratio of false positives is the most difficult and costly part of VBB design and implementation, and it can have a significant impact on user acceptance of VBB products. The key to reducing false positives is to be able to monitor a sequence of actions made by a process while it is active in memory.

It is not always possible to distinguish a single viral behaviour from the behaviour of a legitimate program. However, if a sequence of actions is analysed, it becomes easier to conclude that those actions together are the work of a malicious or viral program.

For example, if a process in memory attempts to send out an email, it could be viral or it could be legitimate. How-

ever, if the same process modifies a Registry setting, then creates a new executable on the system, then accesses the address book and attempts to send emails, it becomes more likely that the active process is a mass-mailing worm. (For an explanation of methods that can be used to reduce the VBB false positive ratio, see the VB2001 conference proceedings, pp.543–561.)

Beyond protecting systems from viral programs, VBB-based products can also provide protection to system resources and enhance system security. Combined with personal firewall implementation, VBB can be used to protect a system from Trojans, worms, and other backdoor types of attack.

Even for distributed denial of service attacks, VBB can be used to prevent the computer from becoming a zombie. A well-designed behaviour-blocking policy could be created that would protect the computer system from some security vulnerabilities and from backdoor hacks by preventing any unauthorized changes to monitored system parameters or files and stopping any kind of foreign unknown executables and port accessing.

The AV Solution

Clearly, VBB technology by itself is not the total AV solution. It is not a replacement for signature scanning but rather it should be implemented as another layer in a virus protection system.

Signature scanning technology has strengths that VBB cannot match due to the nature of the technology. However, in a layered virus protection system, VBB can provide real-time protection against those viruses not yet identified by AV scanning vendors (often referred to as 'unknown viruses'). In effect, VBB technology gives the user and the scanning vendor the time to update and install the signature or software patch that will identify the offending viral program.

At the same time, the AV signature-scanning layer can help to reduce the possibility of VBB false positives, and remains the only effective method of cleaning up a system that is infected before the software is installed. This combination of VBB with AV signature scanning forms a much more effective AV solution.

Conclusion

Behaviour blocking has its place in the AV and computer security industries. There is more and more evidence demonstrating the need for VBB technology in AV software products. Careful design and implementation of VBB products will ensure that, one day, OS manufacturers will realize the power of VBB technology.

Do you agree with Lixin's comments? Is there a need for Viral Behaviour Blocking technology in current and future AV software? Email your opinions to the Editor at comments@virusbtn.com.

RESEARCH

Virus Throttle

A new technique for slowing the spread of viruses has been developed by UK-based *Hewlett Packard* researcher Dr Matthew Williamson.

In his technical paper (see http://www.hpl.hp.com/techreports/2002/HPL-2002-172.pdf) Williamson presents a method for restricting the high-speed propagation of viruses automatically. His approach is based around the observation that, during virus propagation, an infected machine may attempt to connect to as many different machines as possible, as quickly as possible. An uninfected machine will make connections at a significantly lower rate and those connections will, in general, tend to be correlated – for example repeated connections to the same machine.

The thinking behind the approach is that if the rate at which a computer can connect to 'new' computers is limited automatically, the spreading ability of a virus will be severely compromised. A filter on the network stack uses a series of timeouts which restrict the rate of connection to new hosts. 'New' hosts are defined as any that do not tally with a list of recent connections. This way, most normal traffic (as appears on the recent history list) remains unaffected, while traffic attempting a higher rate of connection is delayed.

Should false positives occur a minor, but tolerable, delay in connection speed will be experienced. Where malicious traffic in concerned, however, the timeouts present a significant obstacle to propagation.

The paper details the results of applying the filter to web browsing data, suggests how the system could be implemented on a *Windows* system and discusses both the potential of the approach and its limitations.

The research has raised some interest in the anti-virus community, and *VB* hopes to bring readers a more thorough look at the technique in the near future. In the meantime, readers' thoughts on this research will be received with interest – email comments@virusbtn.com.

Hurry, While Stocks Last!

If you missed the *VB* conference, there are still copies of the VB2002 proceedings available, on CD or in printed format. This well-travelled 500-page tome comprising research papers from 36 of the world's top anti-virus experts is priced at £64.50 (postage and packing will be charged on the printed copy). To order your copy please contact Bernadette Disborough at bernadette@virusbtn.com or call on +44 1235 555139. Remember, these are available only while stocks last!

FEATURE 1

(Porn) Dialers – Another Class of Malware?

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

When making small payments (of a few pennies or cents only) it does not make sense to use a credit card, bank transfer or other expensive transaction method, because the bank fees would exceed the payment. An alternative solution is to make payments by telephone. For example, the customer can call a chargeable number and get a key code which can then be entered into a web front-end to finalise a transaction. Later, the telephone company will charge the amount to the customer's telephone bill.

But another method of payment makes it easier still for the customer (or, perhaps, tricks him more effectively). Web dialer applications are very common now, especially associated with 'pay per view' websites. Dialers are small programs (usually only 50–80 kb in size) which are able to disconnect the current telephone line and dial a cost-intensive number automatically, allowing the user to access the web pages he has requested.

Of course, to be legal (at least in Germany), the dialer must display the telephone number it wants to call, the cost per minute, an identification code for the provider and it must show the general terms and conditions of the provider on request before starting to dial. In addition, an ISDN card or modem must be connected to the PC – users of DSL-only connections cannot use this 'automatic' payment method.

It is very common for pornography websites to use this form of transaction. But other users of this payment method include online games companies and larger download archives – you pay for the product by telephone.

Technically, this method is easy to use and inexpensive for both the contractor and the customer. The customer pays only for the content he receives, at a special per-minute rate. However, since the beginning of 2002, instances of misuse of such dialers (also known as porn dialers) have risen dramatically, both in Germany and across Europe.

A number of websites have been set up in the past few months specifically to address this issue – for example http://www.dialerschutz.de/ – and other security-related websites, such as http://www.trojaner-info.de/ have been extended with a section on porn dialers.

The Issues

First, there are a huge number of dubious providers whose dialers do not display all the relevant information – they 'forget' to include the costs, display inaccurate costs, or conceal the costs somewhere in a greyed-out text box. Others fail to display such a window at all, but call the built-in telephone number automatically or, more craftily, simply change the configuration of the standard Internet dial-up connection to their own (see http://www.heise.de/newsticker/data/uma-30.05.02-001/). A few dialers have been seen that delete their traces completely after this expensive change, leaving no evidence that this program was ever executed on the PC. The user can expect an interesting phone bill at the end of the month …

One very old trick that still seems to work well is to tell the user that they need a special download utility in order to connect to a website with high speed, or that such a utility is needed due to the 'frequently changing IP address of the download server to avoid prosecution, because of the huge amount of illegal software that can be found'. Of course, much is made of the fact that this download tool (EXE) is 'free of charge' – in fact, it is free of charge, but connections using it are not.

Some websites (you can find these easily if you try searching for 'license codes' or 'cracks' in your favourite search engine) will tell you that 'the complete hard disk of your PC can be accessed by everyone on the Internet' and that 'the only tool worldwide to be able to prevent this' can be downloaded from their website – of course, you need to install the 'free-of-charge connection tool' first. And indeed, the content of your C: drive will be displayed in the middle of the web page, using a simple IFrame HTML trick. And some other 'secret' information will be displayed as well – such as the referrer string, the computer's IP address or the browser identification string, so it looks more dangerous to the home user.

Spam Always Works!

Spam emails with an EXE dialer attachment, or at least a link to a website with a 'downloadable dialer', are quite common too.

Often, people will delete such spam emails quickly – but they are less likely to do so if the subject is sufficiently intriguing (for example 'Complaint against you', 'Notice of Cancellation'), if the email appears to be a greeting card from 'a person who likes you very much' (see http://www.intern.de/news/3617.html), or if it appears to contain pictures of body regions that only a gynaecologist would usually want to see.

The following is typical of the content of these spam emails:

```
'Yes guys, we've cracked a dialer now. You
can get free access to the whole web page if
```

```
you use our cracked dialer. It has cost us
days and nights, but finally we were
successful. Have fun, but please do not use
the connection for more than 30–40 minutes at
once, because you could be detected.'
```

And would you lodge a complaint against a dialer that has caused you such a huge telephone bill if you were the one who tried to cheat first by using a 'cracked dialer'? It's a perfect win-win situation – for the provider only, of course.

Easy Money

These providers are often very hard to catch, because the cost-intensive telephone number may have been rented to a German company first, but they subsequently rented it to a Canadian company, and this company rented it once more to an Indian one, the next step is a Spanish one, then a Haitian one and so on. It's virtually impossible to track down the real 'bad guys' who finally get the money. Furthermore, companies that rent telephone numbers this way tend to be very short lived.

To make matters even more difficult, these companies usually have special 'webmaster program' offers. A so-called 'webmaster' (or, more accurately, spammer) can get up to 50% of the telephone fee if a user is online for long enough. And the more users and minutes, the more money. This way, it's easy for the providers to tell everyone that they have not sent out a single spam message that could be in conflict with existing laws (see http://www.heise.de/newsticker/data/jk-16.06.02-001/).

A recent study in the German *PC-WELT* magazine (issue 12/2002, p.14) reveals that only about one or two in a thousand users will install the dialer – whether by accident or intentionally – but such users have to pay about 100 Euros. If only half of the people pay this telephone bill (this can happen quite quickly if the telephone company warns the user that they are going to disconnect him if he does not pay straightaway!), and a spam message is sent out to about two million people at once, that's about 100,000 to 200,000 Euros in a few minutes!

What about AV and Firewall Protection?

Until now, the standard user has been completely unprotected. Anti-virus programs usually do not find dialers, and identify only viruses, worms, Trojan horses and other programs that are malicious (or that are almost certainly malicious, e.g. Win32/Friendgreeting).

A few attempts by German AV vendor *H+BEDV* to detect dialers at the beginning of this year were unsuccessful. They made the mistake of detecting dialer programs and giving the alert message 'Infection: some dialer virus' – of course the dialer developers did not like this classification very much.

Eventually, the signatures had to be removed from the anti-virus program for legal reasons and the matter has cost *H+BEDV* both a lot of time and a lot of money (for more about the case see http://www.heise.de/newsticker/data/ku-14.05.02-000/ and http://www.pcwelt.de/news/software/23834/).

On the positive side of things, a lot of web dialers try to install themselves automatically using well-known security holes in *Internet Explorer*, just like the Win32/Nimda virus, for example. And most AV programs are able to intercept this – there may be a confusing warning such as 'Infection: Mime-Exploit.gen virus detected', but the result is that the dialer is blocked!

Personal firewalls do not protect the user either. Such programs check only the data packets which were sent over an existing connection. Currently, personal firewalls do not check the dial-up of the connection itself, e.g. the telephone number dialled using a white- and/or black-listing approach like the one used for applications for a long time now.

However, a lot of hobby programmers have created dialer protection programs, or warners, that work in exactly this way – they check the telephone numbers called, check whether a called application looks like a dialer, and so on. The most well known of these are *YAW* (*Yet Another Warner*, http://www.yaw.at/) and *0190 Warner* (http://www.wt-rate.com/).

Cat and Mouse

But the dialer industry does not sleep and has reacted very quickly to the appearance of these warner programs – and the events which currently take place within a matter of weeks remind me of the whole 15-year history of the anti-virus industry (the usual 'cat and mouse' game).

One of the first actions the dialer developers took was to change the telephone number to avoid detection. As well as '0190' in Germany, a lot more numbers are billable, like '118xx' which is reserved for information desks, or '0191' and '0193' which are reserved for Internet service providers, and even '005xx' numbers for a connection to some far away islands in the Atlantic Ocean. It's not easy to blacklist all of these numbers, because ISPs like *T-Online* and *AOL* use these numbers legitimately, as do thousands of other providers. But these problems are solvable using black- and white-lists.

The next reaction of the dialer developers was to avoid using the controlled *Windows* built-in dial-up networking and communicate directly with the modem instead, using Hayes-compatible 'AT' commands or ISDN using CAPI (Common ISDN Application Programming Interface) or TAPI (Telephony Application Programming Interface). In addition, some dialers try to access the serial interface (e.g. COM3) directly, in order to bypass any dialer protection programs. The warner applications were updated accordingly.

A lot of dialers try to kill warner applications in memory automatically, without further notice. Some also delete

installation files, or their activation 'Run' key in Registry, so the warner program is unable to start. Another method we've seen is to add the dialer application or telephone number to the white-list simply by changing the warner's configuration file. (Should we call them 'retro dialers', just like 'retro viruses'?)

Newer versions of the warner programs have increasingly robust protection against such kill attempts – for example, *YAW* 'injects' itself into all running processes. This method works better than expected (and more reliably!), because it does not cause any problem to the user (besides higher memory requirements) and it really is unkillable. To enforce it even further, all of the program files are always open and cannot be deleted. The activation points in the Registry (the 'Run' key, for example) will also be checked every 1/10 seconds. Perhaps anti-virus and firewall programmers could learn from such dialer protection programs and prevent Win32/Bugbear-like 'anti-virus killers'.

A few warner programs have replaced the Windows DLL with their own – and if the main program is not loaded, no Internet or other connection is possible. (Some personal firewalls have similar features now.) But even this line of defence was penetrated easily – a lot of dialers include a list of renamed original DLLs or functions the dialers can use 'safely'.

As soon as some dialer protection programs started to include signatures and checksums to detect known dialers easily, the strike-back was that the developers changed their creations. At first they changed every day, but now some developers release new (slightly changed) dialers every hour.

The first dialers were runtime-compressed using UPX mainly to reduce the size, but as soon as some warning programs included a UPX unpack routine, this method changed. A lot of dialers now have a significantly changed UPX extraction routine, they are compressed using other programs and hardly even protected by a lot of anti-debugging tricks and additional encryption routines. (Isn't this the method to hide Trojan horses and backdoors in the virus world right now?)

One interesting point the dialer developers forgot at first was the (mostly uncompressed) resource section of every program where a lot of information is stored, for example the program name or version as well as icons. I hardly need to tell you what happened after the first dialer protection programs checked this section, too …

There is still one Joker left however: if the EXE program has a digital certificate, the name of the company is always included in plain text. I have seen companies that have more than one certificate, even if it's a little more expensive, but think about what you can get back in return.

You may ask why such dialers are digitally signed. The answer is that a lot of dialers try to install them as ActiveX controls first (usually making multiple attempts, if the user does not want this) – disguised as 'chat plug-in', 'security update', 'special graphic viewer (with extreme zoom)', and so on.

If this fails (for example when the browser being used is *Netscape* or *Opera*, neither of which support ActiveX), the user will be prompted to save an EXE file (multiple times, too). In order to avoid an ActiveX warning at the next installation, a number of dialers install their certificate as a trusted publisher in the *Windows* certificate list. Alternatively, they try to execute a small (only 5–7 kb) program automatically, using web browser security holes and later they can install any kind of ActiveX control (not only dialers) without further unwanted questions.

Last, but not least, many dialers – once executed – install themselves in the Registry (Run key), Autostart group, win.ini so that they are always started. In addition they put their logo on the desktop, in the Start Menu, Systray etc. and they change the standard web page ('Home') to their own.

Usually, it's very difficult to get rid of dialers even if they have not been able to dial and cause the user unexpectedly high costs.

Conclusion

Now you've read the full text of this article, wouldn't you agree that dialers as they exist now could be classed as malware? They have been created as a small, inexpensive micro payment method, but a lot of companies use such dialers now for Internet fraud. An ideal protection program should prevent possibly malicious dialers, but allow 'good' dialers the user has accepted.

But existing solutions – anti-virus scanning and firewalls – do not protect the user from this threat. At the moment, many more integrated security products are becoming available – why not include dialer protection in these packages as well? In this case, we would be in a much stronger position to protect the user. For example, if a dialer tries to delete or deactivate the security product, we can call it a Trojan horse and can add detection for it easily.

At the moment, dialers are extremely common in German-speaking countries, but the number of dialers is growing very rapidly across Europe and worldwide, because it's an extremely lucrative billion-dollar business. And it will probably take a very long time (a matter of years?) before the law in so many countries has been changed.

A few weeks ago I saw the first OEM AV package that was bundled with a dialer protection program. When can we expect more?

[*VB is interested in hearing readers' opinions on this issue. Should we classify porn dialers as malware, and should detection of dialers be incorporated into security products? Send your thoughts to comments@virusbtn.com*]

FEATURE

Protecting the Door – Not Just the Mail Slot

Joe Wells
Fortinet, Inc., USA

When your anti-virus software reports that it has blocked a large number of viruses, but your customers claim that their systems have become infected with those viruses, whom should you believe? Is it possible that both parties are reporting the truth?

For Donald O'Rily, it was a tech support nightmare come true. Donald is a friend of mine. He owns '*US in Touch*' (aka *Teknett*), the largest ISP in Nye County, Nevada. His office is right next to mine and often when we meet we discuss the latest virus trends. Earlier this year, he was faced with a very convoluted problem.

In fact, by the time Donald explained the conundrum to me, he had formulated his own theory (which was subsequently proved) to explain the cause and effect of the problem.

The Problem

Teknett has anti-virus protection in place with automatic, hourly updating. However, many of the ISP's customers were calling Donald in a panic, claiming that viruses were reaching their desktops and that their systems were suffering massive infections from the Internet.

How did the users know they were infected? Donald's system was telling them they were. When he checked his anti-virus logs, there were indeed an unusually high number of virus reports – many more within just a few days than would typically appear in a month.

The evidence seemed contradictory. The viruses were being reported, which meant that the anti-virus protection was working. At the same time, the users were becoming infected – viruses were reaching their systems. Yet the anti-virus logs reported that viruses were being stopped. Moreover, the customers were blaming Donald for the infections.

What was happening?

After questioning several of the users whose systems had become infected, Donald formulated a theory based on two things: the anti-virus log files, and his belief that the users were lying to him.

As it turned out, the anti-virus protection was working correctly, users were getting infected, *and* some users were lying to him.

Virus Implosion

Maybe it's human nature, or maybe it's a cultural phenomenon but, when asked whether they used webmail, many users who did use it, denied it.

Perhaps they felt that using webmail in addition to their regular email account represented some form of infidelity, and they didn't want Donald to know they were 'cheating' on him. Whatever the case, the users *did* use webmail and they were getting infected.

A user would check his or her webmail and click on an attachment. A virus or worm would become active. It would go through their address book and send itself to everyone and their aunt Hilda.

The infected mail would be sent out as normal email (rather than webmail), and the anti-virus protection in place at the ISP would stop each of the infected outgoing emails.

For each infected email the anti-virus stopped, it would send the user a message informing them that their system was infected. The next time a user checked his or her email, there would be dozens – or even hundreds – of virus alerts. Panic and angry phone calls ensued.

The anti-virus product at *US in Touch* is on the mail server. The malware bypassed this because it came in via the web. Some of the users had a desktop anti-virus program installed, which plugged into their mail reader. Again, the malware bypassed that protection because it came in via their Internet browser.

So the ISP had fully-functional virus protection that worked exactly as advertised. The users themselves were circumventing the protection and experiencing the results. But try explaining that to an angry mob.

Thou Shalt Not Commit Webmail

Could it be that webmail is a form of infidelity, wherein the 'cheaters' get unwanted infections and are deserving of punishment?

There are more than a few large corporations that believe this to be the case.

Many managers preach, puritanical-style, that one of the commandments should be 'thou shalt not use webmail' (note: God did not deliver The Ten *Suggestions*) and that

the use of webmail in a corporate environment is strictly forbidden (note again: in the Garden of Eden, the apple wasn't the *discouraged* fruit).

As an example, the application of Donald's theory to a large corporation situation demonstrates the real magnitude of the problem.

A company may have email scanning, but even in this case, the theory still applies. A user accesses their personal web mail, launches a virus, and two things happen:

1. The virus sends itself to everyone in his or her address book within the corporation. Since this doesn't go through the email protection between the company and the outside world, the virus is able to spread unhindered within the corporation.
2. If the company does not scan outgoing mail, the company becomes a major vector for the spread of the virus. It sends itself to everyone in the user's address book outside the company – including clients, partners, representatives and contacts.

Like Donald's users, corporate users may have desktop anti-virus protection in place, that plugs into their primary email program. The virus still goes undetected. It still launches. It still spreads.

As a simple solution, you might wonder why these corporations (and Donald) don't scan web content.

Slow Protection is No Protection

From a user's perspective, a 'delay' in receiving email is, with a few exceptions, more or less non-existent – it simply arrives when it arrives. This is a good thing. It allows for comprehensive scrutiny of the email content before delivery wherein time consumption is not a major factor.

Try scanning a user's entire incoming web content, however, and they will notice, and revolt.

I, like them, and you, and almost all other 'normal' members of the Internet generation, find it impossible to wait patiently more than four seconds for a web page to load. A wait of more than eight seconds is totally intolerable. Of course, this is merely the current incarnation of an ancient anti-virus truism.

As I recall, way back in the shadowy beginnings of the anti-virus industry (when I was just a lowly programmer working on a product called *Novi*), I had the dubious honour of coining a phrase that went on to become a widely-used marketing sound bite: 'Slow protection is no protection.'

In DOS scanners this was a self-evident truth. If a scanner was slow, users simply would not tolerate it. Invariably, they would find a way to disable it. As a result, any slow scanner (regardless of all its other fine features) would equate to zero protection.

Even today, computer users habitually break out of startup and full-system scans because they find them too time-consuming. I'm not certain of the current tolerance-to-time ratio as compared to the above-mentioned eight-second benchmark, but I do know it's quite short.

During my time as an instructor in a college computer lab I don't think I ever saw a student allow a startup scan to complete. To be honest, I don't think *I* ever let one complete – and we were using a reasonably fast scanner at the time.

Now extrapolate these facts to the concept of scanning web content.

Volume Control

Scanning the content of an average web page is not a problem in itself. Algorithmically, any of today's scanners can scan a web page quite efficiently. However, the problem is one of sheer volume.

No matter how good a scan algorithm is for the average page – even if it is logarithmic (log N) – the runtime increases linearly (N) as content volume increases. Very quickly, software scanning of all incoming web content becomes intolerably time-consuming. (Since I currently develop hardware-based anti-virus products, I shall leave it at that.)

However you look at it, it's fair to say that more comprehensive content filtering (including web filtering) is looming on the horizon for our industry.

Mars Attacks

In the movie *Mars Attacks*, Martians landed here in Pahrump, Nevada. And that brings me back to my friend Donald's ISP company.

Current corporate handling of web and email content is probably tolerable. Corporations can scan email, lay down the law, and, if they deem it necessary, burn webmail-users at the stake. (Point in trivia: in mediaeval times CEO stood for Chief *Executional* Officer.)

But, if you think Martians landing in Pahrump would pose a problem, try running an ISP here. Whereas corporations may or may not choose to permit web access, ISPs *have* to. They do it for a living. Web throughput is their livelihood and delays equate to disaster.

How long would an ISP last if it tried to prohibit webmail? While corporate security can still squeak by guarding the mail slot instead of the entire door, ISPs cannot. (For that matter, I wonder how much longer corporations will be able to get away with it.)

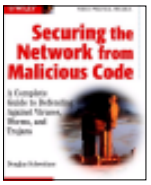
By the way, according to Donald, the burning of persistent webmail users at the stake has a direct, proportional impact on the monthly income of an average ISP …

BOOK REVIEW

Secure Networks

Peter Sergeant

Securing the Network from Malicious Code
Author: Douglas Schweitzer
ISBN: 0764549588
Publisher: Wiley
Price: £29.95



My opinion of this book changed several times over the course of reading it. Initially, it was both the cover design, and the fact that the author highlights his *BrainBench* certification that led me to the cardinal sin of judging a book by its cover.

The first chapter presents a general overview of the problems caused by viruses. Although leaning towards sensationalism in places (I found the term 'plundered by e-pirates' particularly jarring), it provides a good brief on why having malicious code on your network is a bad idea.

Chapter two was my favourite, and for one reason: the author's view of virus writers seems very balanced. Schweitzer makes the point that we are often quick to stereotype virus writers and hackers as maladjusted teenagers lacking in social skills – thus it's all too easy to victimise your friendly neighbourhood programmer while ignoring the trendy kickboxing teenage girl you employed in the post room. With the invective employed by certain anti-virus personas against virus writers, he argues, you could become lulled into a false sense of security, and that would be a very bad thing.

Chapter three looks at other threats to your organization: spyware, adware and er, malware – isn't this a book *about* malware? In this case, it would seem that the author's definition of malware is 'next-generation' Trojans. I was more than a little confused. It became obvious that, while clearly the author has done a lot of research, he seems to lack real-world experience.

Chapter four is unnoteworthy, other than the distinction between Unix and *Linux* viruses. The author seems to view Unix and *Linux* as fundamentally different entities, although he describes *Linux* as Unix-like, rather than seeing that the systems are vulnerable to pretty much exactly the same types of attack.

Chapter five is a meandering look at how computers work, with a mention of an *XP* remote vulnerability thrown in for good measure. It discusses the difference between ROM and RAM, looks briefly at different operating systems but says little about malicious code. Chapters six and seven are of a similar ilk.

Chapters eight and nine are probably the best of the book: they look at what you need to do to implement good security policies throughout your company – firewalls, patching policies; everything you'd expect the book to cover.

Having started to redeem itself a little, the book slides into chapter ten. Since I consider myself an expert on 'Server-side exploits', I had been looking forward to this chapter. Four pages in, the author refers to CGI as a scripting language (along with VBScript and Java), capable of being run on the client end. His coverage of SSI fails to identify or explain the problem. Cross Site Scripting is abbreviated to CSS instead of XSS, and then we move on to the section that convinced me the book had not been through technical review: 'Servers that produce static pages have complete control over how the client interprets the pages that the server provides. Conversely, servers that generate dynamic pages do not control how the output they generate is interpreted by the client's Web browser'. My life as *Virus Bulletin*'s web developer would be so much easier were this accurate.

Rob Rosenberger has written a great article (which can be found at http://www.vmyths.com/) about self-proclaimed anti-virus experts. He claims that those who get a virus and deal with it tend to become the people others come to to talk about viruses, regardless of their merit. I fear that this book could lull readers into becoming empty vessels that drown out the voices of experience.

Additionally, this book has trouble nailing down definitions: 'social engineering' is referred to as the method of educating your users, and later, it's termed as hacker-speak for gaining access to systems by tricking users. 'Hackers' is another good example – it may seem a useless crusade to insist on the use of 'hacker' and 'cracker' in the 'traditional' manner, but for an author to say that hackers look down on virus writers, and later say that copy-cat viruses are created by hackers who just modify 'source code' (another term he defines appallingly) is bound to confuse.

Chapter 11 contains a statement by the US Joint Economic Committee, and discussion of the Patriot Act, ethics and identity theft. Whether this belongs in a book about securing networks I'm not sure. The book ends by looking at PDA viruses, Wireless LANs and Internet banking.

In conclusion, I would not recommend this book – while it does contain some useful information, this would be better found from other sources that don't create bewilderment in readers. This book would be better were it a lot shorter, the title changed, and were it subjected to a tough technical review, where the small but glaring and confusing errors could be fixed, leaving, hopefully, the useful and interesting information that is hidden in its chapters.

PRODUCT REVIEW

Sybari Antigen 7.0 for Microsoft Exchange

Matt Ham

Sybari's *Antigen* is a scanning solution for a variety of mail gateways, incorporating the engines of many well-known anti-virus companies.

For the purposes of this review pre-existing machine images were selected for the test process, namely *Microsoft Exchange 2000 Server* running on *Windows 2000 Advanced Server* with *Outlook Express* running on the server machine. Clients used in the testing procedures were *Outlook* and *Outlook Express* running on *Windows XP*.

Other versions of *Antigen* available are those for *Lotus Domino* and *Microsoft SharePoint Server*. *Antigen* for *Domino Server* is available in a variety of versions for a wide range of platforms. The *Antigen* product for *Exchange* is, obviously, limited to *Windows* server machines.

The third-party scanning engines included in the version of *Antigen* tested were those of *Computer Associates* (both *Vet* and *InoculateIT*), *Kaspersky Lab*, *Network Associates*, *Norman* and *Sophos*. Since these are API versions of the software, the updates are collated and stored centrally on *Sybari*'s servers.

Installation and Update

Installation of *Antigen* begins simply enough with the standard forced acceptance of a licence agreement in order to continue further.

Local or remote installation must be selected next. Remote installation has one option: to install the scanning software on a server. Local installation offers both this and an administrative client as options. For the purposes of this review both scanner and client were installed locally.

Next is the selection of the scanning method to be used, the choice being either the Extensible Storage Engine API (ESE API) or the Virus Scanning API (VSAPI).

ESE API is a custom *Sybari* API, which is applicable to *Exchange* versions from 5.0 onwards. VSAPI is a *Microsoft* built-in API but is available only for *Exchange 2000 SP1* and higher versions. ESE API differs from VSAPI in that VSAPI allows scanning to occur while the scanned objects are within exchange storage, while ESE API scans data as it enters the information store. This means that VSAPI-based scanners are able to scan a message store after, for example, configuration changes or virus definition updates. However, a known weakness of VSAPI is that some methods of external email propagation will not be scanned, thus making it potentially a less-than-perfect implementation for external gateways. For the purposes of the review VSAPI was selected as the mode of operation.

Having selected the scan type, there is the option to update upon install. This is selected by default but, due to the closed-lab nature of the test network in use, was deselected manually. The usual choice of installation target follows, as does the choice of program folders to be added to the start menu, after which the files are transferred, completing the installation process.

During the file transfer and program activation phase two messages are produced stating that a reboot is required in order for particular parts of the installation to perform successfully. Appropriately, a reboot is offered as the final stage of installation, though this may be delayed as required.

As noted earlier, the update point for the scan engines is a central repository on the *Sybari* website. However, since it is not desirable to link machines containing large numbers of viruses to the outside world, the indirect update options were examined for the *VB* test network.

The web location given in the program itself as the source of updates resulted in a 'page not found' error. After this inauspicious start the documentation was inspected, and was found to refer to a marginally different address as well as an FTP address. Sure enough, the latter URL contained the necessary files for update. These were copied onto transferrable media and then onto a further server on the test network so that updates could be tested.

Again there were minor problems, since the UNC paths required for updates are almost, but not quite, identical to those paths on the FTP server – which caused some frustration and the need for some manual directory creation and file movement.

Once this process had been completed, however, the update routine worked very quickly and successfully. Using a network source, the updates were all but instantaneous and did not require reboots or interruptions in operation.

The default method of updates is via a scheduler, although other methods exist – for example, it is possible to set the update process to load at boot-up (though this is a setting which is adjusted elsewhere in the interface). In a display of stringent attention to detail it is explained in the manual which Registry settings are changed when this option is selected.

The method used for the purposes of testing was a manually triggered update, which required individual triggering for each engine. It was also necessary to select an update path

for each engine in turn – the presence of an 'all' button would have made this process easier. Also less than convenient is the fact that the default settings for updates are such that updates are performed only once. Again, resetting this to a more sensible level must be performed manually on an individual basis.

What is more, should the update process fail due to an incorrect path having been specified for the update files, no warning is given. Since there seems to be no central area in which update levels can be checked, this requires that the settings be checked and verified on an engine-by-engine basis, with no warnings which might aid this visual vetting process.

Not all the updates here are concerned with the third-party anti-virus engines included in *Antigen*. One category, the *Sybari Worm List*, is unique to the product. This consists of a set of definitions for various known worms, these being treated somewhat differently from other infected objects. This feature will be described in more detail later, though it was not tested since it is non-functional in VSAPI mode.

Web Presence and Support

The *Sybari* website is http://www.sybari.com/. The home page presents a series of links to company press releases. Unfortunately, the most recent story on this page is rather old, being the press release issued when *Antigen 7.0* first became available to the public, dated 8 October 2002.

With limited in-house research on viruses, the company's website is understandably less strong on virus descriptions than a typical engine developer's website would be. There is a Virus Alerts section which is linked to from the home page, but this is very much restricted to viruses which have received extensive publicity, rather than being an exhaustive resource.

However, the site has other content to tempt the potential customer. Of note is the area which provides information on the file filters which may be used in blocking commonly encountered worms which are limited in the variety of files they are capable of sending.

A unique (in my experience) feature of the *Sybari* website is the selection of online product demonstrations which may be booked here. These entail a visual demonstration which is viewed on the Internet, while a running commentary is provided over the phone.

Documentation and Help

The main source of documentation for *Sybari* products and briefings is in PDF format. However, the *Acrobat Reader* software is not included on the CD with the documentation – which was a minor irritation. Furthermore, the documentation is provided in a more modern version of *Reader* than is to be found on many machines, resulting in partially unreadable documentation if the *Reader* versions do not tally.

The *Antigen for Exchange* manual is a 128-page production. With such a bulk of page count, it was not surprising that the manual turned out to be a fine reference work where such matters as the aforementioned manual definition updates were concerned.

The online help is all but identical to the manual. Since this was the case, leafing through the printed manual seemed more convenient than referring to the electronic help.

Features

Installation of *Antigen* results in the installation of the Antigen Central Manager on the Exchange Server, as well as the Antigen Client, which may be located on a remote machine.

The Central Manager offers a more user-friendly and intuitive interface than the Antigen Client, as well as a different overall purpose. Also provided is Antutil.exe, a command line tool for status checking and basic status changes.

Central Manager follows the standard control layout of a left-hand side-bar paging between areas in the left-hand pane, which takes up most of the GUI area. To this are added a pair of drop-down menus, File and Help, and a server selection drop-down.

The File menu offers the opportunity to select which server is being controlled, the program exit, and control over templates. Templates are configuration settings for the server and may be loaded, saved, renamed and moved from this menu. The choice is set here as to whether to display all available templates or only those active at the time.

The Help menu offers a link to the help file for *Antigen*, a list of contact details and the general version number information for the program in use.

The areas on the left-hand side-bar are divided between three major categories: Setup, Operate and Report. Each category has a small number of sub-categories, each of

which has fewer settings than might be expected in a standard desktop or server scanner. The overall appearance is one of a streamlined control rather than a great clutter of details.

Setup sub-categories are the most numerous of those on offer, falling more into what might be termed 'configuration issues' rather than strictly setup. The sub-categories are Anti-Virus Job, File Filtering, Scanner Updates, Templates, Content Filtering and General Options.

Anti-Virus Job is a central control for determining where scanning should occur and what actions should be taken. At the top of the screen the current status of SMTP scanning, real-time and manual scanning of storage groups and scanning of the Message Transfer Agent (MTA) are listed.

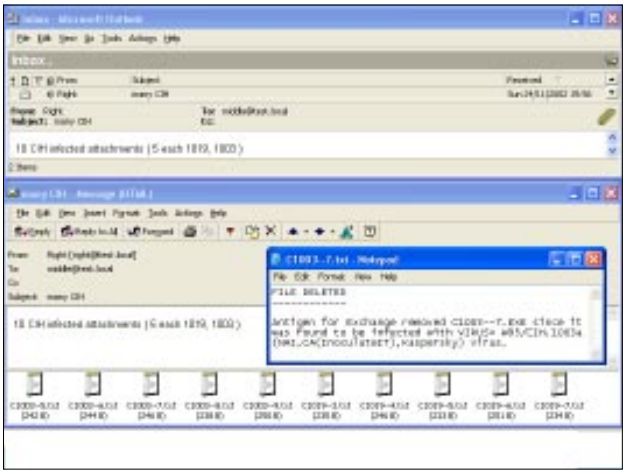
These details consist of whether virus scanning, file filtering and content filtering are each activated within the scan and whether the scan itself is active.

Antigen classifies file filtering as checking for certain file content in attachments, while content scanning is exclusively a term reserved for checking content within message headers. Selecting any of these scan types at the top of the screen results in the display of a more detailed configuration menu at the bottom of the right-hand pane.

For SMTP jobs the scanning may be performed on any combination of inbound and outbound mail, with the option to add a standard disclaimer to outbound mail. As these disclaimers often contain unintentionally amusing quasi-legal declarations, this is a feature of which I fully approve – from an entertainment point of view.

From the right-hand part of the pane it is possible to select which engines are to be used for scanning (all are selected by default), and whether notification of infections should be sent.

Options for action are limited to clean or repair of attachments, detection only, or to delete infections. In addition, files may be quarantined or text inserted in the case of deleted content.



Of interest is the Bias dropdown in this section of controls. This allows actions to be taken at five different levels of certainty, the ratings ranging between Maximum Certainty, through the default of Neutral and on to Maximum Performance at the other end of the scale.

The degree of certainty is derived using the number of scans applied to each file. In the case of Maximum Performance only one engine is used in each scan, while for Maximum Certainty all engines are used on each scan. Neutral Bias uses at least two engines.

In addition to this relatively simple method of determining which engine is used, two further factors are taken into account. First, each engine is weighted according to how recent it is and to a mysterious performance metric. Engines are more likely to be used in a scan if they score highly for newness and performance. The second metric used is the length of time since the engine was used – engines with similar ratings are rotated in use.

MTA scan jobs offer a set of controls that are almost identical to those of the SMTP scan jobs. The absence of a disclaimer text option is the only difference.

Realtime Scan Jobs have an identical right-hand set of engine-based controls, though the left-hand set gives a choice of which mailboxes and public folders should be scanned. The default is to scan all such areas, though there are options not to scan at all, or to select specific areas.

For the Manual Scan jobs the set of options is identical, though Manual Scan jobs are set to detect viruses by default, while Realtime Scans remove viruses by default. Due to the nature of real-time scanning the latter choice makes perfect sense, though perhaps it would be useful for the manual scanning settings to default to more scanning engines being used on each file.

File filtering is the next of the left-hand sub-categories, and again it offers SMTP, MTA and Storage Group scans as its fields of interest. For each of these, specific files may be selected for filtering, the default being that no files are searched for.

The example of 'readme.exe' was chosen as a sample file filter. The addition of the file name to the list of files to be filtered resulted in the appearance of a set of further instructions.

By default, filtering is activated for all files with the specified name, whatever their type. Since this might be overzealous in the case of a file which is always an executable, there is the option in the right-hand area to select that this is of EXEFile type. This may not seem particularly relevant where a full filename is supplied (as here), but might be useful if searching, for example, for 'readme.???' where only readme.scr and readme.exe are to be removed.

In addition to this selection of files to be filtered, the action to be taken on each file is determined here. As in the case of

viruses, this can be deletion or skipping, though clearly disinfection is not relevant in this case.

It was noted that changes in configuration were not saved until a new sub-category was opened, even if multiple changes had been made and selected using the Add button. Although it appeared to be linked to the more specific options, use of the Save button was required to save any of the configuration changes.

The next sub-category is Scanner Updates, which has been covered previously. The Templates sub-category follows which has also been mentioned, albeit in scant detail. Templates are slightly more complex than they might appear to be, in that they are broken down to cover various aspects of server behaviour, rather than having one template to control all aspects of server configuration.

Templates may be created for Internet, real-time, manual, MTA and filter set configurations. Templates also control the content of many notification messages. These templates can be used not only to change local configuration quickly, but also to deploy configurations to other servers.

It was notable that if the option is selected whereby inactive templates are displayed, these templates may be altered without them actually being loaded at the time. This could be of great use when preparing settings for servers with dedicated and distinct needs, prior to their installation.

Next is the Content Filtering control area, which again is divided into controls for SMTP, MTA and storage group scans. For each of these the selection of filters to implement is identical, though they are configured independently. Filters may be implemented on subject line, sender or domain of sender.

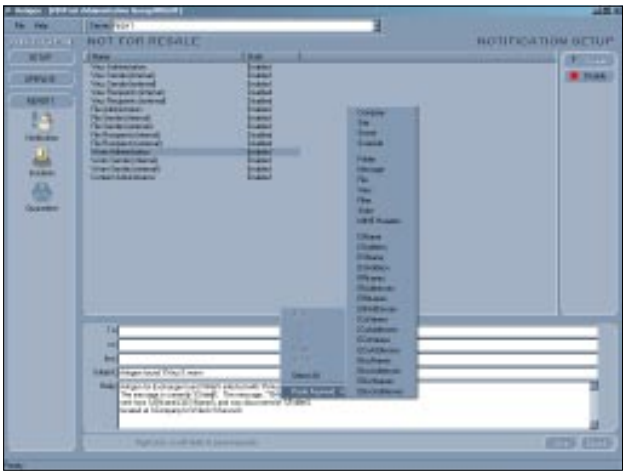
As with previous settings, a notable absence is the ability to copy individual settings from, for example, SMTP scan jobs to storage group scan jobs. If I were a very paranoid administrator who decided to block the sender hahaha@sexyfun.net under all scans, I would need to perform this input task individually for each scan category.

As far as responses to content which has been filtered is concerned, the message may either be purged or simply logged as a detection. As ever, quarantining or notification of the administrator are also options.

Last of the Setup sub-categories is General Options, which contains a range of features which do not fit in neatly elsewhere.

The first of these concerns diagnostics, and which scan types should have additional diagnostics associated with them. In the case of critical problems it is also possible to set a notification list.

Related to this is the logging control, which is next in this sub-category. By default this sends output to the *NT* Event Log, *NT* Performance Monitor and *Antigen* Program Log.



These may all be deactivated independently if so desired and a virus log enabled if required. By default, the program log file is unlimited though limits may be set, starting at 512 KB.

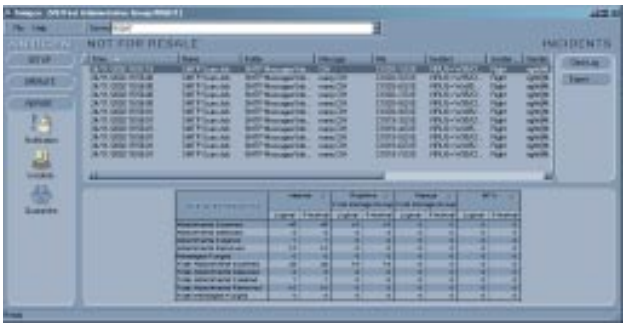
Of more interest are the scanning options which come next. Of these, only deletion of corrupt UUEncoded files is selected by default. Deletion may also be applied to either or both corrupted and encrypted compressed files.

With these three options much of the gain will be in bandwidth, since corrupted files will not often cause infections, though the rejection of files which are encrypted and compressed will be welcome from an anti-virus point of view, if not by some users.

Body scanning is not selected by default, though it can be put in place for manual and real-time scanning independently. This is an area in which advantages are strong from either side of the argument. Those viruses which exist in the body of a message, such as VBS/Kak, can be detected only by body scanning, thus making it a distinct advantage in such cases. On the other hand, body scanning is a slow process and prone to false positives – ironically, especially where virus alerts are concerned. Furthermore, body scanning should be rendered unnecessary by properly patched *Outlook* clients (though this is rather more true in theory than in practice). These are definitely settings where each administrator will need to juggle priorities based on their local situation.

Related to these settings is the decision as to whether DOC files should be scanned as containers – a setting which is tuneable independently between the Internet, manual and realtime scans.

Settings for various maxima in the program settings can be found under the Scanning options heading. These are adjustable around a default. Defaults built into the program are a maximum container size of around 26 MB, with a maximum number of infections of five per container. Whether this is a reporting or scanning limitation was not readily apparent from the documentation. More obvious in meaning were the limits of scanning for nested



attachments (30 being a maximum), nested compressed files (a maximum of five), and maximum scan time (this being set at ten minutes).

With all sub-headings of the Setup category having been covered, the Operate heading is next on the agenda. This is very much more akin to the scanning options within a server or desktop scanner.

Run Job is the basic scan activation area and, once more, these scans are divided amongst SMTP, MTA and both Realtime and Manual Storage Group scans. All but the Manual Storage Group scan are effectively on-access scans and activated continuously by default. For each of these categories, virus scanning, file filtering and content filtering may be enabled or disabled independently of whether the scan is currently active.

The Manual Scan Jobs for Storage Groups are slightly different from the other scans by virtue of their on-demand nature. In addition to a start and stop option a pause option is provided, as well as the option to supply a scan summary notification. Logging is provided for all these scan types on-screen.

Schedule Job, the next of the sub-categories, is limited in its scope to manual scans. Though it is possible to think of wildly theoretical situations where scheduling of other scan types might be needed, this seems a sensible enough limitation if only to eliminate the problems of multiple concurrent on-access type scans. This operates with the same scan settings as the aforementioned manual scan job for the appropriate storage group.

Finally, we reach the Quick scan sub-category of Operate. This is, in effect, the on-demand scan portion of the product, with the same refinement in targeting as those on-demand scans created and triggered elsewhere. It does not, however, alter settings for scheduled and standard scan jobs and is thus useful for scanning without interfering with other settings.

The last of the left-hand categories is Report, which contains sub-categories for Notification, Incidents and Quarantine.

Notification is a central point for settings such as to whom notification is sent when an alert occurs and what form this notification should take. Initially these messages are very similar – though messages to administrators are very much more detailed in the information they contain. The messages are supplied with numerous keywords which are converted to, for example, message identifiers. In order to make insertion of these less error-prone and more intuitive, right-clicking while constructing or editing a notification produces a menu from which a guaranteed keyword may be selected.

Incidents continues the sub-categories in the Report section and offers a comprehensive breakdown of statistical information concerned with the scanning process.

One part of this which is not apparent at first is the distinction between a logical and physical scan of any particular attachment. This is best explained by the example of an email with one attachment which is sent to ten recipients. This attachment needs to be scanned only once, despite the number of recipients, thus registering a total of one physical scan with ten logical scans.

Finally in the Report category is the Quarantine sub-category. This offers both reporting and administering capabilities for quarantined messages. Thus, not only may quarantined messages be inspected, they may be delivered or transferred elsewhere.

Since there may be a large number of files in the Quarantine the store can be set to purge automatically for messages of a certain age or, more helpfully for administration purposes, filters may be imposed upon the data to be displayed.

The Antigen Client is a separate program which allows remote administration of servers. The interface is simple and stark, but effective nonetheless. From here installation or uninstallation can be performed remotely.

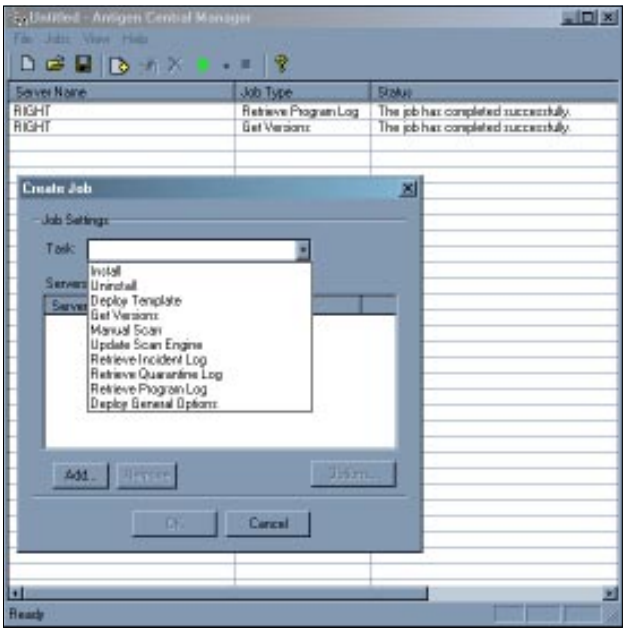
The remote installation feature deems it necessary to be able to browse to machines which are not yet servers, and in a minor irritation this freedom of browsing is also available for other tasks where an Antigen Server is required. Other features which exist here are manual scanning, updating of scan engines, deploying of options and templates and the retrieval of various logs and program versions.

Operation

With Andreas Marx currently working on his tests of gateway scanners (see *VB*, November 2002, p.12), the range of tests which can be performed here without duplicating results is limited.

For this reason, the majority of the tests performed were concerned with detection rates. The possibility of server overload situations through the creation of multi-attachment emails was also examined, though this seemed an unlikely problem area given the program's limitations on number of attachments.

A range of tests were performed with varying numbers of attachments in each email and, in most circumstances,



detection, disinfection and deletion occurred as would be expected from the configuration used. Some problems were encountered, however, when large parts of the WildList were sent as bulk attachments.

In some cases this resulted in the disappearance of the mail in question – something which warranted further investigation. Even taking these vanishing mails into account, full detection was achieved on the WildList-infected files.

The reasons for the disappearance of emails were not readily tracked down by the inspection of options. However, what seemed to be happening was that, rather than the attachment being replaced by the disclaimer text, certain files had triggered the purging of the mail in question.

The triggering of this purge was found to be unrelated to the scanning engine which detected the file. The anomalous behaviour occurred more often with worms than viruses and, given this fact, it was assumed that the behaviour was caused by the WormPurge feature of *Antigen*. The lack of any messages related to such a purge was in accordance with the description given of this feature – though there appeared to be no means to activate or deactivate this feature, so a definite diagnosis was impossible.

Conclusion

As with any gateway product, the scanning engine is only as important as the features that are integrated into the product, so how does *Antigen* fare on these fronts? From the point of view of engines, *Sybari* cannot be faulted, since *Antigen* offers a good range of well-respected detection engines.

The use of so many engines is a good method for the removal of what amounts to bad luck in the timing of engine and definition updates.

For most virus updates the major AV companies are limited in producing a new definition not by technical ability, but by a matter of blind luck as to how long it takes for a sample to arrive and who is awake when that sample arrives. Given this luck factor, the best method of mitigation is to use engines from various, preferably geographically (time-zone) separated, companies – which is something *Antigen* achieves very well.

As far as features are concerned, personal preferences and organisational needs are more of a priority than any arbitrary list of features which 'must' be present. As a guideline, however, flexibility is a good feature so long as it is tempered with a degree of ease in setting and overseeing. In this case, *Virus Bulletin*'s opinion is significantly less relevant than an end user's ideas of what they absolutely need or what they would find over-complex.

Taking these caveats into account, *Antigen*'s feature set seems full in general. However, there are notable omissions, in particular in the transferral of settings within the program, where scan settings for a particular engine or scan-type need to be duplicated manually since there is no provision for automation.

In a summary, *Antigen* offers a good package, though whether it is a great product or just average will depend upon the judgement of a particular user.

As an aside, the combination of several scanning engines within one product raises some issues. Although in many cases, the anti-virus companies themselves produce gateway scanners, they are equally likely to supply their scanning engines to third parties or indeed to occupy both positions.

By including more than one scanner, a third-party product can often gain significant advantages over a single-product scanner, thus reducing the market for the products created directly by an engine developer.

Once one engine developer has offered their engine for use by third parties, the others have little choice other than to follow suit – even though it seems entirely possible that the engine developers are, to a certain extent, reducing their own profits in the process.

Technical Details

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

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

Client software used: *Windows XP Professional* with *Outlook Express*, *Windows XP Professional* with *Outlook 2000*.

Developer: *Sybari Software Inc.*, 353 Larkfield Rd, East Northport, NY 11731, USA.

Tel +1 631 630 8500; Fax +1 631 630 8555; email sales@sybari.com; web http://www.sybari.com/.

SUBSCRIPTION RATES

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

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

Editorial enquiries, subscription enquiries, orders and payments:

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

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

US subscriptions only:

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

Tel (781) 9731266, Fax (781) 9731267

END NOTES AND NEWS

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

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

The 12th Annual SysAdmin, Audit, Networking and Security Conference (SANS) takes place 7–12 March 2003 in San Diego, USA. The conference will feature 12 tracks, night activities, a vendor exhibition, and additional special events. See http://www.sans.org/.

Infosecurity Italy will be held in Milan, Italy, 12–14 March 2003, for details see http://www.infosecurity.it/.

CeBIT, one of the world's largest information technology trade fairs, runs for one week in Hannover, Germany from 12–19 March 2003. All aspects of IT are catered for, with well over 7000 exhibitors. For full details see http://www.cebit.de/.

RSA Conference 2003 takes place 13–17 April 2003 at the Moscone Center, San Francisco, CA, USA. General sessions feature special keynote addresses, expert panels and discussions of general interest. This year's Expo will feature more than 138,000 square feet of exhibit space with more than 200 vendors. Optional tutorials and immersion training sessions will provide the basics of e-security technology, enterprise security and security development techniques, and 13 class tracks will feature a wide variety of workshops, seminars and talks. See http://www.rsaconference.net/.

Information Security World Asia takes place 23–25 April 2003, Suntec Singapore. See http://www.informationsecurityworld.com/2003/iswa_SG/.

Infosecurity Europe 2002 takes place 29 April to 1 May 2003, Olympia, London. A free keynote and seminar programme alongside almost 200 exhibitors is expected to attract more than 7,000 dedicated security visitors. See http://www.infosec.co.uk/.

EICAR 2003 will take place 10–13 May 2003 in Copenhagen, Denmark. Check http://conference.eicar.org/ for the latest details.

Black Hat Europe 2003 takes place 12–15 May 2003 at the Grand Krasnapolsky, Amsterdam, the Netherlands. For more details see http://www.blackhat.com/.

***Kaspersky Labs* is expanding its regional office network** with the opening of its French branch in Sophia Antipolis. The company already has regional offices in the UK, USA and Poland. For more details see http://www.kaspersky.com/.

***Panda Software Italy* is running the 'First National Virus Prevention Campaign'**, which aims to rid Italy's PCs of viruses. Throughout the campaign, which started in November and runs until 31 January 2003, *Panda* will be offering home users a special version of *Panda Antivirus Titanium* which includes 30 days of free updates. See http://www.pandasoftware.com/.

The *Virus Bulletin* team would like to wish all *VB* subscribers a very happy Christmas and a prosperous New Year!

Bernadette, Pete, Helen and Matt.