

realtimepublishers.comtm

The Definitive Guidetm To

Email Management and Security

 **singlefin**
e-mail protection services

Kevin Beaver

Chapter 2: Fighting Malware	21
Malware History	21
How Malware Was Transmitted	21
How Malware Behaves Now	23
Where Malware Is Headed.....	27
How Malware Affects Email	29
Other Types of Email-Affecting Malware.....	31
A Word About P2P Networks.....	32
Preventing Malware Outbreaks.....	33
General Software Protection.....	34
Client Malware Protection	35
Server and Perimeter Malware Protection	37
Malware Policies.....	38
User Awareness	39
Dealing with Hoaxes.....	40
Summary	43

Copyright Statement

© 2003 Realtimedpublishers.com, Inc. All rights reserved. This site contains materials that have been created, developed, or commissioned by, and published with the permission of, Realtimedpublishers.com, Inc. (the "Materials") and this site and any such Materials are protected by international copyright and trademark laws.

THE MATERIALS ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. The Materials are subject to change without notice and do not represent a commitment on the part of Realtimedpublishers.com, Inc or its web site sponsors. In no event shall Realtimedpublishers.com, Inc. or its web site sponsors be held liable for technical or editorial errors or omissions contained in the Materials, including without limitation, for any direct, indirect, incidental, special, exemplary or consequential damages whatsoever resulting from the use of any information contained in the Materials.

The Materials (including but not limited to the text, images, audio, and/or video) may not be copied, reproduced, republished, uploaded, posted, transmitted, or distributed in any way, in whole or in part, except that one copy may be downloaded for your personal, non-commercial use on a single computer. In connection with such use, you may not modify or obscure any copyright or other proprietary notice.

The Materials may contain trademarks, services marks and logos that are the property of third parties. You are not permitted to use these trademarks, services marks or logos without prior written consent of such third parties.

Realtimedpublishers.com and the Realtimedpublishers logo are registered in the US Patent & Trademark Office. All other product or service names are the property of their respective owners.

If you have any questions about these terms, or if you would like information about licensing materials from Realtimedpublishers.com, please contact us via e-mail at info@realtimedpublishers.com.

Chapter 2: Fighting Malware

In this chapter, I'm going to focus on malware—malicious software—that affects email. First, we'll take a look at how the malware train got rolling and where it's headed. We'll explore how email-based malware works, then finish up with some practical tips about how you can secure your email systems. The only place to start is at the beginning...


Malware History

The roots of malware go back to 1949 when mathematician John von Neumann suggested in his paper "Theory and Organization of Complicated Automata" that computer programs could reproduce. This science-fictional idea ended up being a very accurate prediction. In fact, only one year after this paper was released, before we experienced the first true computer virus, some employees at Bell Labs had already put von Neumann's theory to work in a computer game called Core Wars. It took several decades, but in 1981, the Apple II Elk Cloner virus, the first known computer virus, was released.

In 1983, a student at the University of Southern California named Frederick Cohen, one of the original virus pioneers, and his doctoral thesis advisor Len Alderman, noticed a distinct similarity in the method of replication between computer viruses and biological viruses, thus giving computer viruses their name. Who would have thought that this newly used phrase would become part of everyday computer speak?

 It's interesting to note that the term *computer virus* was not coined until 1983—2 years after the release of the first computer virus.

Later, in 1986, the Brain virus, the first virus known to affect the Intel/Microsoft platform, was released into the wild. This virus was supposedly released by two owners of Brain Computer Services, Basit and Amjaad Farooq Alvi. Brain was not harmful. The developers' intent was to leave their contact information on the floppy disks of infected computers in an attempt to stop their software from being pirated in their home country of Pakistan. Based on this inconspicuous beginning, who would have thought that this new virus craze would take off like it has?


 What's in a name? So how is malware deemed virus, worm, Trojan horse, and so on? Malware doesn't have a specific reference collection of code to compare it with (at least initially) and there is no central naming organization, so malware protection solution vendors are forced to come up with names in a very short period of time that they believe best fit the piece of code so that users can be notified. The result is various names for the same piece of malware...ah, the confusion!

How Malware Was Transmitted

Once upon a time, computing was much simpler. Obviously, there were networks, and threats to those networks. In fact, the Morris Worm of 1988, which was written by Robert Morris Jr. to demonstrate software vulnerabilities, disabled approximately 6000 systems—10 percent of the Internet's hosts! This feat was so monumental that the federal government realized the seriousness of this potential problem and prompted the Defense Applied Research Projects Agency (DARPA) to form the Computer Emergency Response Team (CERT) at Carnegie Mellon University. A greater threat appeared a few years later in the form of boot sector and macro viruses.



As you might recall, back in the mid-to-late 1990s, boot sector viruses were the craze. These viruses found their home in the boot sector of floppy disks. As these disks were shared among friends and coworkers traveling from computer to computer, every computer that the diskette came in contact with became infected, especially if up-to-date antivirus software was not loaded. The program would load into memory and run after the first reboot. Similar file-infecting viruses became prevalent as well, attaching themselves to harmless executable files. Both boot sector and file-infecting viruses did everything from display a harmless message on the computer screen to completely erasing the computer's hard drive.

 Remember the first time you accidentally booted your PC with an infected diskette and ended up infecting your hard drive? Checking your drives before booting is a lesson that, unfortunately, many of us learned the hard way. Moral of the story: disable booting from the floppy drive on all the computers that you manage.

Both types of viruses were typically slow to propagate because they required the actual sharing of diskettes and manual transfer from person to person. The fact that local area networks (LANs) were not as prevalent as they are today helped prevent the virus from spreading. These viruses were usually easy to catch because most antivirus software (if it was loaded) scanned floppy drives on access. However, these viruses were spreading at a time when updating your virus signatures daily would have been considered insane. In fact, it wasn't even possible to get updates that quickly because the antivirus vendors typically released new signatures once a month!

In 1995, we were introduced to the first widespread macro virus called Concept. Macro viruses take advantage of various macro development language commands—mostly in the Microsoft Office macro language. Malicious commands are embedded into files for programs such as Word and Excel and usually transmitted via email. Whenever the recipient loads the files into Word, Excel, and so on, the macro executes and inflicts harm. The Concept virus was benign—especially compared with the macros viruses we would see later (such as the Melissa virus).

The Morris Worm and boot sector viruses exploited known computer vulnerabilities just as modern malware does—evidence that not a lot has changed regarding malware. What has changed is the Internet—the means of propagation—and the complexity and insecurity of modern software applications. Just a few years ago, client software did not provide specific services that could be accessed from anywhere in the world as they do today. There were no ports to probe, no inbound requests to respond to, and, perhaps most important, there were no dangerous email attachments. Modern computer networks, email software, and vulnerable programming methodologies have increased complexity and insecurity.

 Before the Internet went mainstream, standalone computer systems were the most vulnerable to malware attacks. Now, the opposite is true.

In the early 1990s, I learned how PC video controllers could be inadvertently programmed to burn the screen and render the controller and/or monitor useless; how hard drive controllers could be programmed to damage the heads of the drive, disable the drive, or simply cause the drive to thrash into oblivion; and how simple it is to corrupt the BIOS of a computer—effectively disabling the entire computer. These types of computer vulnerabilities were all very serious at the time, and they still are. However, these standalone attacks pale in comparison with what can happen when entire organizations and communications systems can be taken offline with newer attacks.

How Malware Behaves Now

Boot sector and other executable-infecting viruses were simply too easy to detect, which created a new trend of related attacks. Malware is now sneaking past firewalls on the coattails of email and posing dangers to organizations both large and small. According to MessageLabs, as of this writing, one in every 276 emails is infected with a virus. Based on the data I've gathered, there are more than 60,000 known types of malware in the wild. Luckily, only a small percentage of these are actually in circulation. Still, given these numbers, and considering that email is the most widely used Internet application, the chances of an organization being affected by a malware attack are astounding.


📄 Check out MessageLabs' Global Outbreak maps for a neat graphical representation of recent malware outbreaks (<http://www.messagelabs.com/viruseye/threats/outbreaks>).

With our newer, more interconnected computer systems and communication methods, everyone has email access. Hundreds of thousands, if not millions, of hosts can be infected within a very short period of time. In 1999, the Melissa virus took roughly 4 days to wreak millions of dollars in damages. The LoveBug virus only took 5 hours. Although the SQL Slammer/Sapphire worm was not email related, it's also a sign of things to come. It created most of its damage in its first 10 minutes of propagation!


Time is one of the factors that make email such a great malware propagation platform—through email, malware propagation speeds are in near real time. Enter the email malware dilemma we're in now. As Table 2.1 shows, in the mid-to-late 1990s, almost all malware was transmitted via diskette. Email has since taken over as the number one medium with diskette infections virtually non-existent.

Virus Source	1996	1997	1998	1999	2000	2001	2002
Email attachment	9%	26%	32%	56%	87%	83%	86%
Internet download	10%	16%	9%	11%	1%	13%	11%
Web browsing	0%	5%	2%	3%	0%	7%	4%
Don't know	15%	7%	5%	9%	2%	1%	1%
Other vector	0%	5%	1%	1%	1%	2%	3%
Software distribution	0%	3%	3%	0%	1%	2%	0%
Diskette: other	71%	84%	64%	27%	7%	1%	0%

Table 2.1: How malware is transmitted (Source: ICSA Labs 2002 Virus Prevalence Survey).

 You can review and even participate in ICSA Labs' annual Virus Prevalence Survey at <http://www.icsalabs.com>.

Malware is now doing more than just displaying funny messages on the screen and deleting files—it's taking down entire networks. Developers of these malicious programs know that many organizations don't buy into security or don't have the resources to put toward security, and the developers are capitalizing on this knowledge. They know, perhaps more often than the typical business manager or executive does, that malware poses significant threats, including lost productivity, business downtime, lost intellectual property, and some liability issues. Possibly the most obvious reason they delight in attacking email systems is because they know that most businesses are completely dependent upon it.

 According to the Meta Group, 80 percent of business people think that email is more valuable for business communications than the telephone. In addition, 74 percent of them believe that being without email would present more of a hardship than being without the telephone.


Malware is now gleaning private personal and corporate information from computers. It is also replicating by simply sending itself via email to email client address book entries, or even worse, by exploiting some form of vulnerability on huge numbers of Internet-connected hosts. More and more systems are now being exposed to remote access vulnerabilities (specifically, remote access Trojans—RATs) that place administrator-level backdoors for future access. In addition, newer malware attacks are bypassing existing security mechanisms such as antivirus software and personal firewalls. To top it all off, malware is being programmed cleverly enough that evidence that the infection occurred is covered up.

Modern malware exploits the natural trusting tendency of human beings (called social engineering) to con people into opening malicious attachments. This vulnerability is one of the greatest that organizations face with regard to email and IT security. The following email subject lines and message bodies are from four famous viruses and worms and provide evidence of why people are tempted to open them:

- Anna Kournikova Worm
Email subject line: Here you have, ;o)
Email body: Hi: Check This!
- Melissa Virus
Email subject line: Important Message From *sender's name*
Email body: Here is that document you asked for ... don't show anyone else ;-)
- MyLife Worm (aka Bill Clinton worm)
Email subject line: bill caricature
Email body: Hiiiiii How are youuuuuuuuu? look to bill caricature it's vvvery verrrry ffffunny :-) :-) i promise you will love it? Ok buy
=====No Virus Found=====
MCAFEE.COM


- Palyh Worm (aka support@microsoft.com worm)
Email sender: support@microsoft.com
Email subject line: *varies*
Email body: All information is in the attached file

The proper mix of technology, policies, procedures, and user awareness can help ward off these attacks.

 I discuss user awareness issues to consider later in this chapter.

There are other reasons for the increased use of email as a malware propagation mechanism:

- Malware creation tools for writing malicious programs are readily available for experts and script-kiddies (beginner hackers)
- Less complex skills are needed to create malware than were required in the past
- Most organizations don't know how to quantify their malware losses, so they just ignore the possibility of a problem until one occurs
- The number of protocol, application, and OS vulnerabilities is increasing
- There is an increasing number of vulnerable hosts connecting to the Internet


 According to CERT, 99 percent of intrusions result from exploitation of known vulnerabilities or configuration errors where countermeasures were available.

At the time of this writing, Kaspersky Labs had determined that 89.1 percent of all malware was network worms, and 95.6 percent of these worms are email worms. These numbers certainly show where our priorities need to be. We'll discuss some tips and techniques for preventing malware outbreaks shortly.

Adding insult to injury, cleaning up after today's malware infections is not quite the same as it used to be. It's no longer as simple as letting your antivirus software clean infected files. Often it requires invasive tasks such as

- Taking servers offline for extended periods of time for cleanup and investigation
- Restoring or rebuilding corrupted databases
- Reinstalling and reconfiguring everything from scratch
- Ultimately being forced to implement various security policies, procedures, and technologies that could have prevented the outbreak in the first place

According to ICSA Labs, when factoring in all the indirect costs, malware outbreak costs for the average organization can total as much as \$500,000.

 You can use various Internet databases to learn more about malware based on name, type, effect, and so on. The following sites are some of my favorites:


Symantec's Virus Encyclopedia Search at <http://securityresponse.symantec.com/avcenter/vinfodb.html>

Trend Micro's Virus Encyclopedia Search at <http://www.trendmicro.com/vinfo/virusencyclo>

McAfee Virus Search at <http://www.mcafee.com/anti-virus/default.asp>

Computer Associates' Virus Information Center at <http://www3.ca.com/virusinfo/browse.aspx>

Malicious developers are not only developing malware that spreads quickly to millions of computers, they are developing quite a sense of humor. For instance, there is a Linux-based worm called the Cheese worm. This worm “attacks” systems that have been previously compromised via another vulnerability, patches them, and moves on. An Outlook-based worm called ProLin (Pro Linux) goes as far as actually promoting Linux. Among other file manipulations, it displays the message “change at least now to LINUX.” Imagine the spam opportunities that exist here!

 I'll cover spam in detail in Chapter 3.

In addition to the technical enablers surrounding malware, there are some key non-technical factors that enhance the propagation of malware. Some of the factors that supplement the propagation of malware include:

- Emerging peer to peer (P2P) usage is enabling the connection of huge numbers of systems to the Internet

 I'll discuss P2P in more detail later in this chapter.

- Marketing techniques are being used to lure gullible people into opening attachments (called social engineering, as I mentioned earlier)
- Communications barriers among various countries around the world limit the ability to spread word of malware attacks
- Time of day—cleverly crafted malware attacks come in the early morning hours when the majority of people are not available to respond quickly

Speaking of time of day, are you aware of which malware attacks might affect you and your organization today? Thanks to several antivirus vendors, we have access to virus calendars that display which attacks are expected on any given day. Figure 2.1 shows a screen capture of Symantec's graphical calendar, which you can find at <http://www.symantec.ca/avcenter/calendar>. Perhaps one day we'll even have the opportunity to synchronize these calendars with our Outlook, Organizer, and so on calendars so that we won't miss a beat!

Virus Calendar

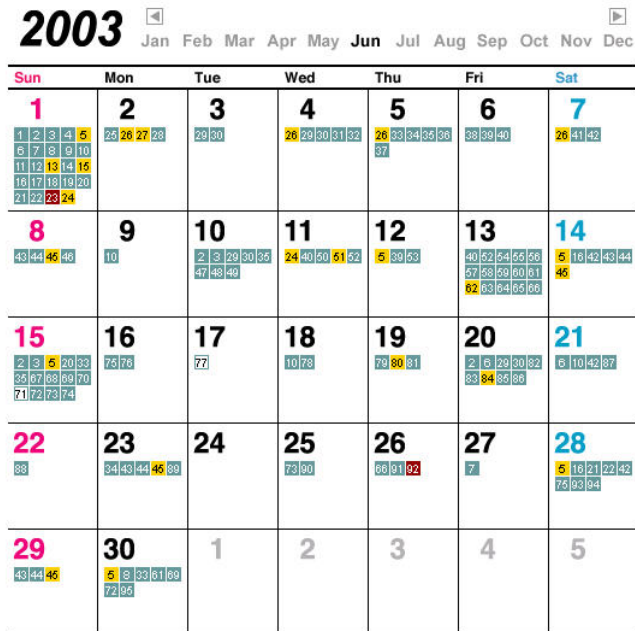



Figure 2.1: Symantec's Security Response virus calendar.

Where Malware Is Headed

What we've seen thus far is only the tip of the destructive iceberg. As long as programs have the ability to bypass security controls and do basically anything on a computer system, and as long as applications are not designed with security in mind, we will continue to have malware problems. Regarding future attacks, there's one huge thing we have against us—time. Time is systems administrators' biggest foe and malware developers' best friend. Malware developers know that their code must be written, tested, and launched as soon as possible after a vulnerability has been detected and made public.


 There are ongoing debates as to whether security vulnerabilities should be made public before fixes are available—I'll discuss this debate in more detail in Chapter 5.

Fighting malware attacks is usually a case of too little, too late no matter how quickly antivirus software vendors can get new signature updates and patches out. This situation is only going to get worse. The time it takes for software and antivirus vendors to develop a fix increases the attackers' chances of inflicting the maximum amount of harm on hosts that haven't yet been patched or protected. I believe that we'll see more malware that takes advantage of this time factor by using new techniques to spread across the Internet, infect systems, and avoid detection long before new signatures can be added to antivirus software and patches can be developed. There's even a chance of serious attacks targeting specific hosts or users—and that may be just around the corner. What were once zero-day exploits might just become personal zero-hour and zero-minute devastations.

 Malware infection times are only going to decrease!


In addition, while we will still see social engineering exploits that take advantage of naturally trusting human beings, malware is going to become less dependent on the human factor and more dependent on the fact that network administrators and users are not properly patching their systems. Building on this vulnerability, malware developers will likely increase the damaging effects of attacks because the developers know that there is a good chance that no preventative measures have been put in place, and even worse, no response measures have been documented or tested in most organizations.

The more complex attacks that are still to come will render standalone malware solutions—such as signature-based antivirus software and personal firewalls—ineffective. Email servers, or at least gateways, with built-in firewall and intrusion detection/prevention functionality will become more important. In addition, organizations will become more reliant on hardened email relay servers that sit in front of the actual email server in order to protect their systems.

 Relying on standalone malware protection tools in the future might be bad for your IT systems' health. A layered security infrastructure will be essential.

There are already software development tools that are giving standard antivirus software a tough time and might eventually render them useless. For instance, there are executable packing tools available that let malware developers restructure the binary layout of malware without affecting its ability to work as designed. In addition, there are various binding tools available that let these developers combine legitimate programs with malware—allowing them to look like harmless programs while at the same time wreaking havoc on users' computers and the network. All of these tools combined with the endless signature possibilities will create too many unique signatures for antivirus software to keep up with.

In the future, there will be less of a distinction between client and server attacks. We will see more “unified” attacks that affect the majority of computer systems at once. In addition, attacks will increasingly affect all forms of electronic communication including cell phones and instant messaging (IM). We will need to rely more on behavior blocking, sandboxes, and access controls for protection. Real-time monitoring of new processes and files on our systems will need to be built-in to applications and perhaps even OSs. Some of these features are already included in a few host-based intrusion detection system (IDS) products that exist today. Potential future features include better security for address books and email addresses to prevent malware from finding email addresses and exploiting them.

 Without proper integrity checks in place, future malware attacks could take advantage of major vendor support Web sites and distributed patching systems. These attacks could modify patches, antivirus engines, and antivirus signature files that will be downloaded onto thousands of systems before anyone even notices. This type of attack has already occurred in the open source arena—proprietary software vendors might be the next target.

Other future attacks will come by the way of email protocol exploitation. There are already many known TCP/IP-based vulnerabilities. As new protocols are developed, weaknesses will be found and documented and, ultimately, exploited. Current anomaly-based detection systems will be enhanced to run on both servers and clients—and possibly even perimeter devices such as routers and firewalls—to monitor protocol behavior and ensure that the protocol specifications are being followed.

How Malware Affects Email

The best way to protect your systems from malware outbreaks is to understand how malware can successfully cause problems. In this section, we will take a high-level look at some common ways that email-based malware achieves its goals. Figure 2.2 shows the essential ingredients of malware. Any malware—virus, worm, Trojan horse, and so on—depends on these minimum requirements.

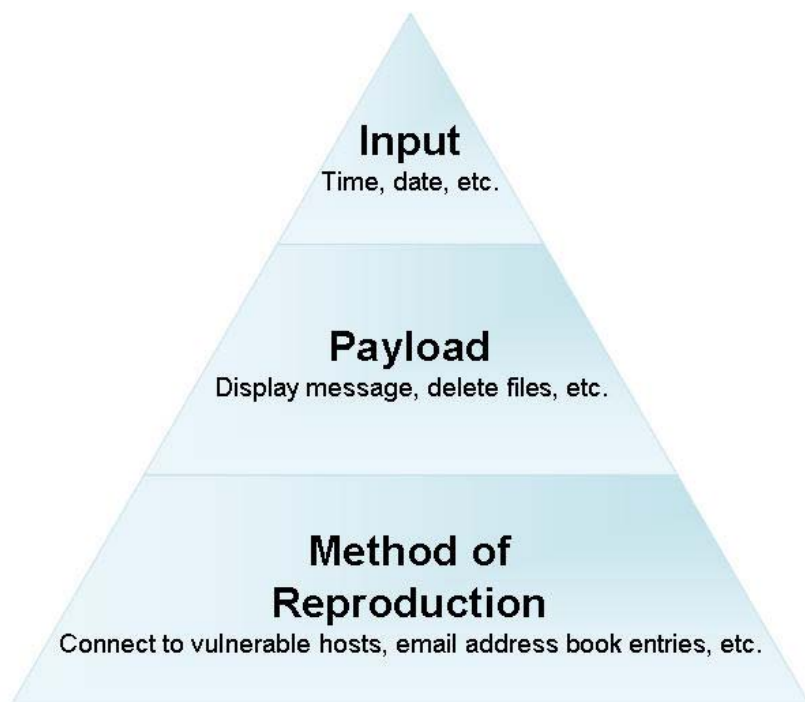


Figure 2.2: Essential ingredients of malware.

In addition to these basic elements of malware, we know the following factors to be true:

- The Internet is an open network making even the most far-reaching attack possible
- Our information systems are becoming more and more complex—to the point that many developers don't understand how everything works in their own applications
- TCP/IP and most of the communications protocols in its suite were not designed with security in mind
- Software is typically designed without security in mind
- Protective measures to keep malicious users from getting caught are becoming more sophisticated
- Human beings are naturally trusting

These factors combined with the following myriad vulnerabilities are a lethal combination hindering secure email usage:

- Vulnerable multimedia Multipurpose Internet Mail Extensions (MIME) email attachments are becoming more common
- Email clients have HTML-rendering engines that allow them to easily execute malicious scripts and programs
- Newer malware attacks are able to affect multiple OSs (for example, the W32.Winux virus can infect both Windows and Linux)



Although malware exists that can attack multiple platforms, most malware is written for a specific OS or application or to exploit a specific vulnerability in order to spread more rapidly and affect targeted hosts.

- Various services—such as email, Web browsers, and Web servers—running on the same machine can all contribute to the problem or be affected at the same time
- Polymorphic malware can change appearance using various encryption keys and generating new malicious code in real-time, making detection very difficult
- Metamorphic malware can change behavior during propagation, making prevention and detection that much more difficult even with more sophisticated behavior-based detection technologies
- Malware, specifically worms, are now being preprogrammed before being launched to attack specific systems that have been found to have a specific vulnerability—increasing the rate of infection exponentially
- The payload of malware is getting smaller and more efficient



Payload is defined as the malicious data contained in malware that executes to create the ultimate (usually negative) effect.

For example, the Slammer/Sapphire worm had a tiny payload (376 bytes) that could easily fit into one User Datagram Protocol (UDP) packet. In comparison, the Code Red worm had a payload of around 4000 bytes and the Nimbda payload was around 37,000 bytes. Based on general network physics, the larger the packet required for transmission, the slower the infection will be.



Combine small payload, UDP, and millions of vulnerable systems, and a worm such as Slammer can cause some serious damage!




There is one vulnerability that deserves a little more coverage—file attachments in emails. There is a common myth that .exe files are the only types of files that can cause problems. Many users who believe this myth open other file types without fear of infection—only to become infected. Any type of executable file can contain malicious code (including, but certainly not limited to, common file extensions such as .bat, .pif, .com, .scr, .vbs, and even the very familiar .doc and .xls files used in Microsoft Word and Excel). Most antivirus software provides a listing of common executables that you can refer to. Just know, and train your users, that executable files can be encrypted, compressed, or otherwise disguised as other harmless files in an attempt to bypass antivirus software and fool users into thinking a file is benign.

 Check out The File Extension Source at <http://filext.com> for information about more than 14,000 file extension types.

Other Types of Email-Affecting Malware

In addition to viruses, worms, and Trojan horses, there are a few other types of malware affecting email security that deserve mentioning:

- Logic bombs—Programs that are set to perform malicious acts upon a certain triggering event, usually a certain time or date

 There have been several high-profile logic bombs:

UBS PaineWebber allegedly spent \$3M fixing the damages caused by a logic bomb planted by a former network administrator

An ex-Omega Engineering network administrator planted a logic bomb that erased all company data and put a halt to the organization's manufacturing that resulted in 80 employees being laid off and \$10M in damages

- ActiveX controls—Microsoft-based applications that could potentially cause harm through email
- Java applets—Programs written in Sun Microsystems's programming language that could potentially cause harm through email
- JavaScript—Programs written in Netscape's scripting language that could potentially cause harm through email
- Keyboard loggers—Programs loaded into memory, often under the Trojan horse guise, in stealth mode (so as not to be seen or detected) that log all keystrokes that are sometimes sent to remote computers for further analysis
- Network analyzers—Programs or utilities that are used to capture and analyze packets of data going across a network
- Password crackers—Programs that can be installed or run, often under the Trojan horse guise, to crack local or remote system passwords
- Rootkits—A set of administrator-level programs that allow for the creation of backdoors, remote control, network analysis, and more on UNIX and Linux systems

- Spyware—Programs downloaded (oftentimes unintentionally) that run in the background and spy on user’s Internet habits or, worse, send out confidential, personal, or corporate information
- VBScript—Microsoft-based scaled-down version of the Visual Basic programming language that could potentially cause harm through email

Most modern antivirus, behavior monitoring, personal firewall, and spyware applications when working in conjunction with each other can detect and prevent these malware applications from running or causing harm. A layered host-based defense strategy similar to that which Figure 2.3 shows should be deployed to ensure this type of protection.

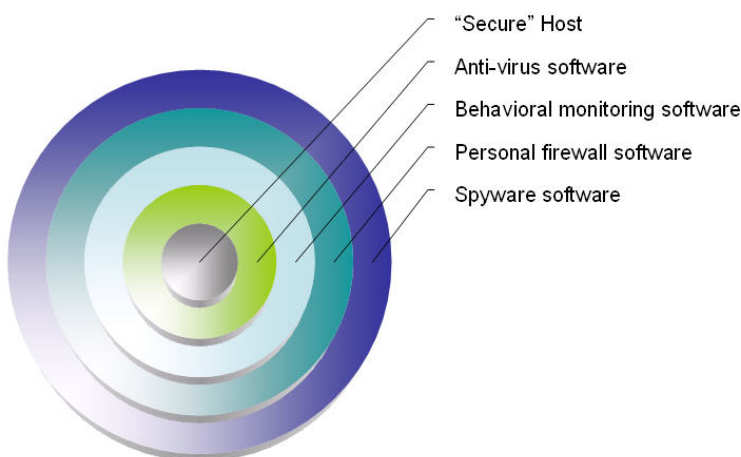


Figure 2.3: Multiple layers of an ideally protected host.

A Word About P2P Networks

Sure, P2P is not the same as email, and email is not going to be replaced by P2P. However, P2P’s communication methodologies are being integrated into newer groupware/messaging platforms that directly relate to email. Given this integration, I thought it might be of value to at least introduce some of the high-level security concerns with P2P so that you can ensure secure integration of email and P2P technologies in the future.

 P2P is a mindset—a way of working rather than a specific technology.


P2P is a form of decentralized computing that takes advantage of previously unused resources such as processing power, storage space, and content available on the Internet and private networks. It’s changing the way we communicate through improved file sharing and IM. It’s helping to lower IT hardware, software, and administration costs by eliminating the need for centralized servers, mass storage devices, and centralized network administration. However, it’s not without its security drawbacks. The following is a list of P2P security issues you should consider before testing or deploying P2P in your organization:

- Provides yet another entry point into corporate networks
- Introduces more points of failure
- Introduces new vulnerabilities that current antivirus and other malware-protection software may not address

- Puts a great deal of security responsibility in the hands of end users
- Is a great platform for truly distributed denial of service (DoS) attacks
- Is a huge malware threat that could lead to malicious users controlling thousands of computers at a time—including your own network computers
- Some P2P programs may appear to be legit but are instead Trojan horse programs that can be transferred onto your local network
- Computer/network/email configuration information can be gathered by remote malicious users
- Poorly written P2P programs can introduce security vulnerabilities or cause computers to crash creating system availability issues
- Some legitimate P2P applications have “security” features built in that automatically update software upon user logon; this functionality is not good if the updates haven’t been tested first
- P2P can use up valuable network bandwidth and storage space creating DoS attacks
- Messaging can have insecure authentication that could allow someone to impersonate you or your users
- Loss of control over what leaves the network
- You have to create new policies and procedures and implement new technologies to deal with these P2P security issues

Preventing Malware Outbreaks

Unfortunately, many organizations don’t think about having to fight off viruses until they’re faced with an attack. Many organizations have no centralized malware management system, which leads to updates not being properly installed, users disabling their protection, and more. Users want freedom to send and receive emails, so it’s up to the IT/security staff to balance security with convenience. A general lack of technology is bad, but a lack of policies, procedures, and security incident response plans can be even worse.

 I’ll cover email security incident response in Chapter 5.

Having solid policies, procedures, and incident response plans are key elements in any successful information security program. It’s impossible to plug all the potential security holes in your software and on your network—your information systems will never be 100 percent secure. However, there are some general best practices that you can follow to help prevent malware outbreaks and keep your systems up and running. In the following sections, I’ve divided the tips into three areas:

- General software protection
- Client malware protection
- Server and perimeter malware protection

I’ve also included some coverage about security policies and user awareness.

Keep in mind that these best practice lists are not intended to be comprehensive, as certain factors such as the type of applications and email system you're using, the size of your organization, and so on will ultimately dictate what is specifically needed for your situation. Be sure to test any specific settings in a non-production environment to ensure system compatibility and stability before you roll them out.

General Software Protection

- Ensure that your malware-protection mechanisms are in line with your security policies
- Be wary that malware-protection products can provide a false sense of security
- Be aware that malware-protection products have a reputation of limiting what a user can do and using up a lot of memory and processor power—these factors can have negative side effects such as slowing down computers and decreasing user productivity
- Know that current malware detection applications do not have the ability to assess and patch new vulnerabilities; doing so requires an adequate patch-management system be in place. Future patch management products may merge with antivirus and pest control software to provide more in-depth protection.
- Make sure that you can trust the source of malware protection and information
- Configure your antivirus software to automatically check for engine and signature updates in real time; if your software won't do so, request this feature from your vendor or consider switching products (there's no reason this feature shouldn't be present and you shouldn't have to be burdened with performing the updates)
- Test your software using the EICAR test virus; although this method isn't comprehensive, it provides a good starting place. Antivirus software searches for the following EICAR string:

```
X5O!P%@AP[4PZX54(P^7CC)7}$EICAR STANDARD-ANTIVIRUS-TEST-FILE!$H+H*
```

 You can download the EICAR test file from http://www.eicar.org/anti_virus_test_file.htm.

- Remember that antivirus software only works by identifying known virus signatures and therefore cannot protect against signatures it doesn't know about
- Layer your malware protection—run malware protection software on your clients, servers, and perimeter devices whenever possible; if possible, run a different virus engine at each layer
- When running multiple antivirus scanning engines, be sure to reboot between scans to eliminate the chance of errors
- Take advantage of the newer applications that employ a sandbox approach to malware analysis—this solution analyzes code in a safe area independently of any application and determines whether the code is safe

☞ Want to see a demo malware “attack” that may blow you away and make you think twice about just how vulnerable your systems are? Check out the *executable malicious file attachment demo* from Finjan Software at http://www.finjan.com/mcrc/sec_test.cfm. It’s a safe and harmless way to show you that typical antivirus and personal firewall software may not be providing adequate protection.

- Be aware that not all JavaScript, Java applets, and ActiveX controls are malicious—in fact, the majority are just fine to run on your systems. Thus, you don’t need to disable JavaScript, Java, or ActiveX controls in your applications just for the sake of security. If other security controls are properly configured, antivirus software is kept current, users are aware of security issues to look out for, and you have another layer of security such as behavior-based monitoring on your systems, your systems should be pretty secure. Remember to balance security and convenience.
- Combine antivirus software with other technologies such as pest/spyware software, file integrity-checking software, and behavior-monitoring software
- Remember that encryption won’t protect your systems from malware—encryption will just help malware sneak past antivirus software and other protection mechanisms
- Consider purchasing ICSA Labs (or similar) certified antivirus products

Client Malware Protection


- Start with at least a minimum level of antivirus protection—any client computer without antivirus software is an accident waiting to happen
- Before purchasing any standalone version of malware-protection software, determine whether you will benefit more from a network version—network versions may provide you with centralized administration features that could pay for themselves in the first week of use
- Harden clients based on industry best practices
- Enable real-time virus protection to ensure that floppy diskettes, CD-ROMs, DVD-ROMs, and Universal Serial Bus (USB) drives are scanned upon access
- Enable email protection to ensure that both inbound and outbound emails are scanned
- Perform full scans at least weekly but focus more so on ensuring that real-time protection is always enabled
- Scan all files, not just executables
- Turn on heuristics protection, which can detect strange behavior and potential commands within software that should be blocked; this type of protection was initially used to analyze macros and is now common in most antivirus applications
- Test new and unfamiliar software on an isolated system before deploying across a network—you never know what malware could be present and what it could do
- Don’t rely on signed ActiveX controls to be secure; the signing only means that you know where the code came from, not what it’s going to do


-  For more information about ActiveX security, see the CERT Results of the Security in ActiveX Workshop document at http://www.cert.org/reports/activeX_report.pdf.
-  For more information about Java security, see Sun Microsystems' Frequently Asked Questions—Java Security at <http://java.sun.com/sfaq>.


- Enable automatic software updates
- Force a full scan when virus signatures are updated
- Do not open unsolicited attachments and encourage your users not to
- If possible, limit the number of authorized users on each computer
- Apply patches as soon as they are released and you have successfully tested them
- Subscribe to one or more of the following security/virus alert services to stay abreast of new security vulnerabilities and malware threats:
 - CERT—http://www.cert.org/contact_cert/certmaillist.html
 - SANS—<http://www.sans.org/newsletters>
 - SOPHOS—<http://www.sophos.com/virusinfo/notifications>
 - Microsoft—<http://register.microsoft.com/regsys/pic.asp>
- Make the normal.dot default Word template file read-only
- Enable macro security on your Microsoft Office applications
- Create and use email filtering rules in email clients to filter suspicious emails
- Write-protect floppy disks when using them in other computer systems
- Make a boot disk/emergency repair disk (ERD)
- Make backups of program executable files; consider creating disk images of a clean installation
- Disable *Hide file extensions for known file types* in Windows Explorer so that you (and your users) can see the full extension of files
- Do not turn off real-time antivirus protection—the 5 percent or so computer speed loss you may regain is not worth being unprotected! If you need to turn it off temporarily to burn a CD-ROM or DVD-ROM, be sure to remember to re-enable it!
- Make it a habit to back up the critical OS files, including the Windows registry, when performing data backups
- Microsoft Outlook Express and Outlook contain a preview function—disable this function
- Disable Outlook Express if it's not being used
- Use personal firewall/IDS software
- For PDAs, ensure that malware-protection software is installed and is loaded on both the PDA and the local computer when synching or beaming to/from another PDA





- Use a non-administrator or root-equivalent account whenever possible
- Disable Windows Script Host (WSH)
- Consider buying the appropriate licenses for users to be able to install antivirus software on their home computers if they will be connecting into the network remotely
- Disable computers from booting from the floppy drive
- If using IE, Microsoft Outlook, or Outlook Express, run the Windows Update built-in to Windows or at <http://windowsupdate.microsoft.com> to ensure that the latest Outlook and IE patches have been applied
- Ignore advice to disable HTML in your email applications as doing so is not useful (neither is forcing people to use the .rtf file format instead of .doc in Word and the .csv file format instead of .xls in Excel)

 For a free online virus scan, check out one of the following tools:

 McAfee FreeScan <http://www.mcafee.com/myapps/mfs/default.asp>

 Panda ActiveScan http://www.pandasoftware.com/activescan/com/activescan_principal.htm

 Symantec Security Check <http://security.symantec.com>

 Trend Micro HouseCall <http://housecall.trendmicro.com>

Server and Perimeter Malware Protection

- Make a boot disk/ERD
- Harden servers based on industry best practices such as the National Security Agency's Security Recommendation Guides (<http://www.nsa.gov/snac/index.html>), the National Institute for Standards and Technology's (NIST's) 800 Series publications (<http://csrc.nist.gov/publications/nistpubs>), and SANS Step-by-Step guides (<http://store.sans.org>)
- Look for antivirus solutions that hook directly into your server's email software, which can provide greater protection than standard SMTP, POP3, and IMAP4 protection
- Disable booting from the floppy drive
- Enforce protection on the actual email server or perimeter device (such as a firewall), preventing malware from reaching end users
- Scan messages as they are received on the server—both inbound and outbound and on each message transfer agent (MTA)
- Move virus defense as far out to the perimeter of your network as possible; malware-protection solutions based solely at the client level are too risky—you want to get the protection mechanisms as far away from users as possible
- Limit attachment sizes to eliminate questionable attachments
- Limit the number of incoming messages per minute

- Scan message body contents of all emails to look for specific programming codes (such as classid) that shouldn't be in typical emails
- Restrict executable files in email attachments if necessary
- Educate your users as much as possible
- If worse comes to worst, consider blocking well-known malware attachments
- Configure your server and/or antivirus software to send a page or email when suspicious activity is detected
- Maintain backups for at least a month in case a virus is detected and files are corrupted or lost so that you can go back and find good copies

 If you're running sendmail—the popular UNIX email server—on a system without antivirus protection, check out the virus-killing script on TechTV's Web site that can help protect your computers (<http://www.techtv.com/screensavers/answerstips/jump/0,24331,3317403,00.html>).

Malware Policies


Year after year in numerous industry surveys, the majority of organizations run malware-protection software yet those same organizations continue to have malware outbreaks. The primary reason for this scenario is the lack of policies and the failure to enforce existing policies. There are several key policy considerations when dealing with malware protection for your email systems. The first of these is to make sure that you clearly define your malware policies and educate *all* of your end users. After that, you must issue appropriate warnings if policies are violated. Keep in mind that your main goal is to influence behavior.

Keep malware-protection measures including software updates, hoax analysis, and incident response procedures out of the hands of users whenever possible. They don't want to deal with these issues and you don't want them to deal with these issues. Don't simply block all file attachments and expect your users to sit back and not question this policy. File attachments are one of the greatest benefits of email! If you just disallow attachments, people will find a way around it (such as Web-based email, IM, or even the good old floppy or CD-ROM method). Similarly, why allow executables through your email system only if they are compressed? There's not much of a difference because they will be uncompressed eventually anyway—perhaps on a system that doesn't have proper malware protection! Remember that malicious executables can still get in even if you attempt to block them. I can't stress the security vs. convenience issue enough. It *is* possible to strike a good balance.

Make sure that your malware-protective measures are always loaded on your computer systems. Perhaps, most important, always have policies that require telecommuters, auditors, consultants, temporary workers, and so on to have current malware protection (at least antivirus software) on their systems before they *ever* connect to your network. This area is one of the most often overlooked yet simplest ways for a network to be infected with malware.

The following list provides areas that malware-related policies in your organization should consider:

- Personal use of email systems
- Using malware protection software
- Creating secure email passwords
- Downloading and installing software
- Dealing with virus hoaxes
- Forwarding emails
- Web-based email access responsibilities

 I'll cover policies and user awareness in-depth in Chapter 6.

User Awareness

The human factor is the weakest link in any information security program. Email security is more than a technical issue, so remember to educate upper management and your end users. You also need to remember to train your staff properly. Malware is receiving mainstream media coverage ranging from the local news to popular sitcoms to the big screen. Between this coverage and the dependence upon email as a corporate tool, users are somewhat aware of the potential security threats and vulnerabilities associated with email, but there's still widespread ignorance about the overall risks. There could be an entire book dedicated to the subject of user awareness.

Although the following suggestions might contradict traditional methods of user awareness in the arena of malware, they are a result of my experiences of what works in the real world. Everything that I've touched on in this chapter—from the successful propagation of sophisticated email worms to the opening of macro virus—relies on one thing: the lack of awareness on the part of the end user. Humans are trusting by nature and easily tricked. You've got to explain common methods of social engineering to your users. Antivirus and other malware-protection software can give a false sense of security. Make sure that your users know that there is no such thing as absolute security. Tell them what they need to be looking out for and which questions they should ask of you and others that approach them about computer access. Also, outline consequences of a social engineering or malware attack—these consequences should be clearly stated in your company's malware policies.

Rather than simply telling users not to open email attachments (you simply cannot force the responsibility of malware protection onto users), explain what to look for. It's not their end users' fault if their systems become infected by email-based malware that doesn't even require them to open their email—much less an attachment. In addition, if there is no malicious intent, end users should not be held responsible for innocently replying to or forwarding an email that came from someone they know (as was the case with the Melissa virus). Users cannot and should not make security decisions based on what a pop-up window is asking, such as those that Figure 2.4 shows.




Figure 2.4: Empowering users is good—giving them the decision-making authority to answer these types of questions is not good.

Panic during a malware outbreak often leads to poor decision making, unnecessary downtime, and lost productivity. Most users either don't know to watch out for malicious attachments or trust everything more than they should. Awareness can help minimize potential outbreaks. However, educating users is no simple task—education has to be ongoing.

Dealing with Hoaxes

Have you ever gotten an email from your friend stating that if you don't delete a certain file on your system or turn your computer off immediately that you'll be infected by a nasty virus? I think we all have. These are known as virus hoaxes—the spam of the malware world. We've all seen them come and go and come yet again no matter how many times you encourage people to stop sending them. Hoaxes are basically junk emails containing “alerts” and “warnings” about mythical viruses, Trojan horses, and the like that must be heeded immediately—or at least that's what the originator wants you to believe.

There have been several studies done to determine the cost and effect of hoaxes on corporate email users. Similar to spam, most seem to agree that each hoax email costs organizations an average of \$1 per instance in lost productivity and system resource usage. This cost can add up to a substantial amount of money, especially for larger organizations.

 According to CIAC, people spend more time trying to debunk hoaxes than dealing with legitimate virus and Trojan horse incidents!

Hoaxes are never accurate and only serve to waste time and take up processing power and disk space on your email servers and client workstations. So why do people send hoaxes? Some send them to harass people or see how far the messages will go. Others do so to try to damage someone else's reputation or make themselves feel important. Like malware developers, spammers, and other computer miscreants, perhaps one might conclude that these hoaxsters have too much time on their hands!

There are a few ways to deal with hoaxes. First and foremost, you must have a policy, procedure, and consequences regarding end users forwarding these junk emails. Your policy could simply be to not forward potential hoax emails to anyone inside or outside the organization or, better yet, require that they forward them to one person only (most likely the systems administrator) so that you can determine whether they are legitimate. Your procedures could include the following:

- Contact the point person first
- Check the PGP or other digital signature inside the alert to determine its authenticity
- Check <http://www.vmyths.com> or another reliable hoax site to determine whether you're dealing with a hoax

From a technical perspective, to help enforce your policy, you could deploy content filtering on emails to filter the junk based on keywords. Whatever you choose to do about hoaxes, just do something. The hoax problem is getting worse every year and perhaps with enough awareness training and policy enforcement, we can get it under control.

 For more hoax information, also check out <http://www.symantec.com/avcenter/hoax.html> and <http://hoaxbusters.ciac.org>.

Malware Tools and Resources

For more information about malware, check out the following resources. In addition, I've included a list of tools, research and response sites, and vendors who offer malware protection solutions.

Resources

- AVERT Technical White Papers at <http://vil.nai.com/VIL/white-paper.asp>
- Java Security Hotlist links at <http://www.cigital.com/javasecurity/complete.html>
- Kaspersky Lab Virus Alerts Mail List at <http://www.viruslist.com/eng/maillist.html>
- McAfee Dispatch - FREE e-mail newsletter at <http://dispatch.mcafee.com>
- Reinforcing Dialog-Based Security at <http://www.usafa.af.mil/dfcs/papers/mcc/ieeesmc2001.pdf>
- Security Tradeoffs: Java vs. ActiveX at <http://www.cs.princeton.edu/sip/java-vs-activex.html>
- SOPHOS Computer Viruses Demystified at http://www.sophos.com/sophos/docs/eng/comviro/viru_ben.pdf
- Virus Bulletin at <http://www.virus-bulletin.com>
- Virus Bulletin—Independent Anti-Virus Advice at <http://www.virusbtn.com/index.xml>
- VirusTalk Computer Virus Forums at <http://www.virust.com>

Tools

- Central Command Vexira Antivirus Rescue Disk System at http://www.centralcommand.com/rescue_disk.html
- McAfee Virus Removal Tools at http://www.mcafee.com/anti-virus/virus_removal/default.asp
- Symantec Security Tool List at <http://securityresponse.symantec.com/avcenter/tools.list.html>

Research and Reporting Sites

- Anti-Virus Information Exchange Network at <http://www.avien.org>
- Computer Emergency Response Team (CERT) Coordination Center at <http://www.cert.org>
- ICSA Labs at <http://www.icsalabs.com/html/communities/antivirus>
- Symantec Security Response Virus Submission site at <http://securityresponse.symantec.com/avcenter/submit.html>

Vendors Offering Malware Protection Solutions

- Antigen
- AvoCon
- BorderWare
- Brightmail
- Central Command
- CipherTrust
- Clearswift
- Computer Associates
- ESET
- F-Secure
- GeCADSoftware
- GFI Software
- IceWarp
- Kaspersky
- McAfee
- MessageLabs
- MicroWorld Technologies
- NetIQ
- Norman
- Panda Software
- PrivateExpress
- Singlefin
- Sophos
- Symantec
- ZixIt

Summary

In this chapter on malware, we examined the history of malware, its current realities, and where it is headed. I also covered malware's negative effects on email and P2P networks, as well as how malware outbreaks can be avoided—from software protection to user awareness. I listed a substantial collection of malware tools and resources at the end of the chapter that can be used in the fight to protect your OSs and network against the ever-growing problem of viruses, worms, and other malware. Keep in mind that this list is not a comprehensive listing of tools and resources, but some of my favorites and a sampling of what's available.

Coming up in Chapter 3, we'll enter the world of spam. We'll explore what spam is about, tricks and tools that spammers use, and some interesting spam statistics. We'll also delve into the security implications of spam, how to eliminate *most* of it, how to track it, and which tools and resources are valuable in your quest to minimize its impact on your organization.