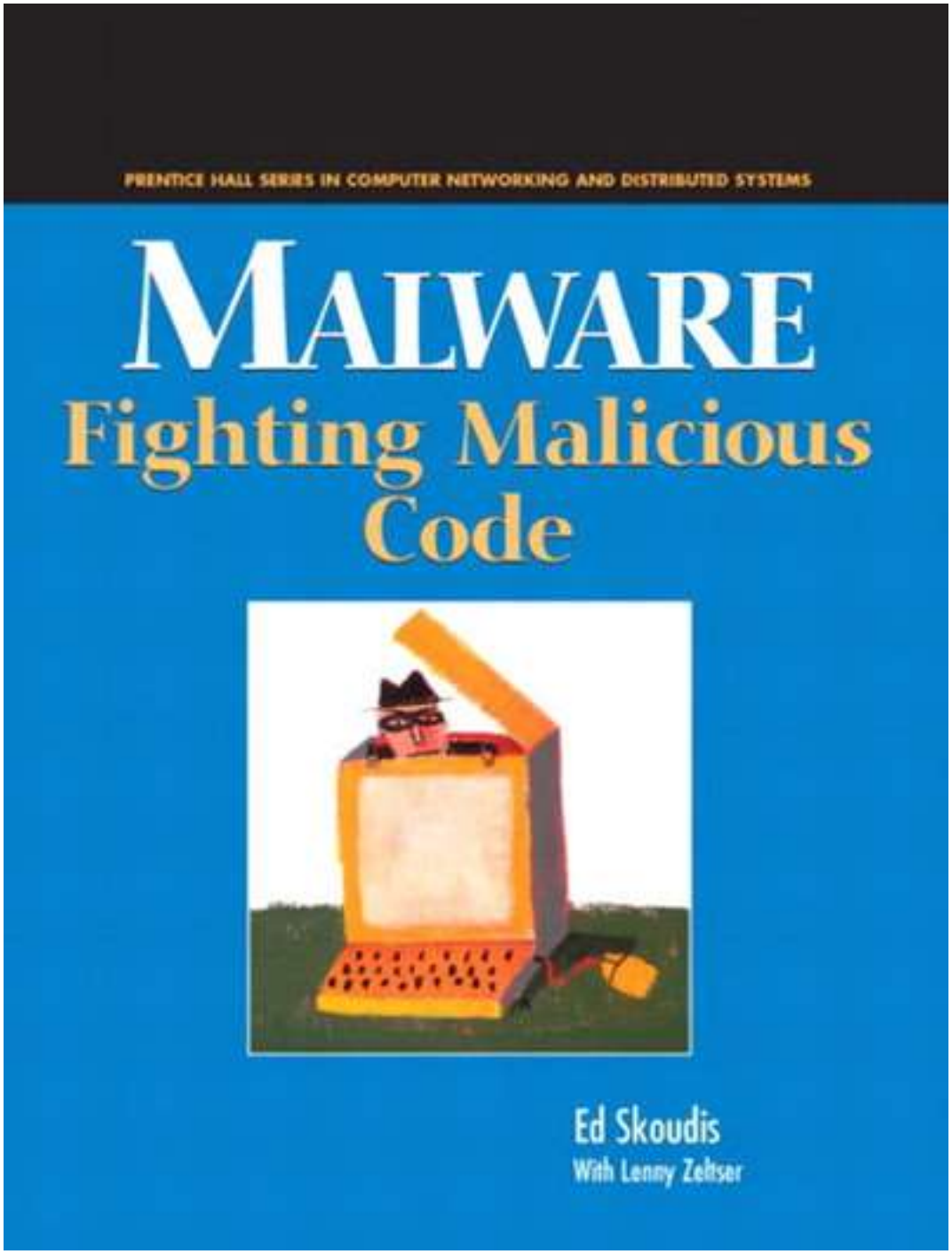
O'REILLY

BOOK

Malware: Fighting Malicious Code

★★★★★ 1 review

By Ed Skoudis, Lenny Zeltser



Trend Micro, Inc.
EX1015-1

Continue

TIME TO COMPLETE:
18h 40m

TOPICS:
Malware

PUBLISHED BY:
Pearson

PUBLICATION DATE:
November 2003

PRINT LENGTH:
672 pages

- Reveals how attackers install malicious code and how they evade detection
- Shows how you can defeat their schemes and keep your computers and network safe!
- Details viruses, worms, backdoors, Trojan horses, RootKits, and other threats
- Explains how to handle today's threats, with an eye on handling the threats to come

"This is a truly outstanding book-enormous technical wealth and beautifully written."
—Warwick Ford

"Ed does it again, piercing the veil of mystery surrounding many of the more technical aspects of computer security!"
—Harlan Carvey, CISSP

"This book is entertaining and informative, while justifiably scaring you. Luckily it also tells you how to protect yourself, but makes you realize it's going to be a permanent spy-vs-spy struggle."
—Radia Perlman, Distinguished Engineer, Sun Microsystems

Keep control of your systems out of the hands of unknown attackers

Ignoring the threat of malware is one of the most reckless things you can do in today's increasingly hostile computing environment. Malware is malicious code planted on your computer, and it can give the attacker a truly alarming degree of control over your system, network, and data-all

Trend Micro, Inc.
EX1015-2

without your knowledge! Written for computer pros and savvy home users by computer security expert Edward Skoudis, Malware: Fighting Malicious Code covers everything you need to know about malware, and how to defeat it!

This book devotes a full chapter to each type of malware-viruses, worms, malicious code delivered through Web browsers and e-mail clients, backdoors, Trojan horses, user-level RootKits, and kernel-level manipulation. You'll learn about the characteristics and methods of attack, evolutionary trends, and how to defend against each type of attack. Real-world examples of malware attacks help you translate thought into action, and a special defender's toolbox chapter shows how to build your own inexpensive code analysis lab to investigate new malware specimens on your own. Throughout, Skoudis' clear, engaging style makes the material approachable and enjoyable to learn. This book includes:

- Solutions and examples that cover both UNIX® and Windows®
- Practical, time-tested, real-world actions you can take to secure your systems
- Instructions for building your own inexpensive malware code analysis lab so you can get familiar with attack and defensive tools harmlessly!

Malware: Fighting Malicious Code is intended for system administrators, network personnel, security personnel, savvy home computer users, and anyone else interested in keeping their systems safe from attackers.

About the Publisher

Pearson Publishing is an established educational publisher, software developer, and provider of training systems. Specialising in mobile learning, it helps raise achievement, improve staff performance and reduce costs.

More about Pearson

Resources

Purchase book

Trend Micro, Inc.
EX1015-3

Copyright

A CIP catalog reference for this book can be obtained from the Library of Congress

Editorial/Production Supervision: MetroVoice Publishing Services

Executive Editor: Mary Franz

Editorial Assistant: Noreen Regina

Marketing Manager: Chanda Leary-Coutu

Manufacturing Manager: Maura Zaldivar

Cover Designer: Talar Agasgan

Cover Design Director: Jerry Votta

Series Designer: Gail Cocker-Bogusz

Full-Service Project Manager: Anne R. Garcia

© 2004 by Pearson Education, Inc.

Publishing as Prentice Hall Professional Technical Reference

Upper Saddle River, New Jersey 07458

Prentice Hall PTR offers excellent discounts on this book when ordered in quantity for bulk purchases or special sales. For more information, please contact: U.S. Corporate and Government Sales, 1-800-382-3419, **corpsales@pearsontechgroup.com**. For sales outside of the U.S., please contact: International Sales, 1-317-581-3793, **international@pearsontechgroup.com**.

Trend Micro, Inc.
EX1015-4

All company and product names mentioned herein are the trademarks or registered trademarks of their respective owners.

All rights reserved. No part of this book may be reproduced, in any form or by any means, without permission in writing from the publisher.

Printed in the United States of America

First Printing

Pearson Education LTD.
Pearson Education Australia PTY, Limited
Pearson Education Singapore, Pte. Ltd.
Pearson Education North Asia Ltd.
Pearson Education Canada, Ltd.
Pearson Educación de Mexico, S.A. de C.V.
Pearson Education—Japan
Pearson Education Malaysia, Pte. Ltd.

Dedication

To the four Js...

Trend Micro, Inc.
EX1015-5

Defending against Viruses

Until now, this chapter has focused on analyzing virus threats. It's time to turn our attention to ways in which we can counteract these threats. After all, understanding the threat and defending against it is what this book is all about. In general, protecting our systems against malicious software requires a layered approach to security. The diversity of malware and the inventiveness of its authors make it likely that a specimen will find a way around one particular defense mechanism. There is no single tool that will reliably block all malware attacks. However, employing several protective measures will ensure that if one of the mechanisms is bypassed, the other ones still have a chance of stopping the infection. It's a classic belt-and-suspenders approach. If someone cuts your suspenders, you'll still have a belt to hold up your pants. With this mindset, we'll discuss several mechanisms critical to defending against viruses and keeping your pants on, including antivirus software, configuration hardening, and user education.

Throughout this book, each chapter presents you with recommendations best suited for dealing with the particular malware threat discussed in that chapter. As you read about these defensive techniques, keep in mind that some of them apply to more than just one type of malicious software. For example, antivirus tools are important for fighting viruses, as well as for catching known worm and Trojan horse specimens. In the remainder of this chapter, we'll look at these defenses with a virus defender's mindset. We'll address some of these same tools later on in subsequent chapters, but then we'll focus on them in the context of worms, Trojan horses, RootKits, and other types of malware.

Antivirus Software

Antivirus software is one of the most widely adopted security mechanisms in use today. Even the stingiest of chief information officers (CIOs) will probably admit that not installing antivirus software would likely be

Trend Micro, Inc.
EX1015-6

violating due care principles that have become commonplace in modern computing environments.

When it comes to deploying antivirus software at home, there aren't that many different types of devices where we can install these programs. In the typical household, if you apply antivirus software to each home machine, you're in pretty good shape. The environment that supports a business tends to be more complex, though, and usually offers more installation options. When deciding where to deploy antivirus software in an organization, consider these infrastructure components that can act as gateways for viruses trying to reach potential hosts:

- User Workstations: As users double-click on e-mail attachments or download files from the Web, they are likely to encounter malware that will target their systems. Therefore, it is critical to have antivirus software running on workstations, both desktop and laptop models
- File Servers: A file server acts as a central repository for users' files, and is a great place to centrally detect and eradicate malicious code. Therefore, it is a good idea to run antivirus software on your file servers.
- Mail Servers: A mail server acts as a hub for mail processing within an organization, and is a great place to scan for malicious e-mail attachments before they reach end users. Installing antivirus software on such servers allows you to compensate for the possibility that it might be disabled on user workstations, or that the users' virus signatures are outdated.
- Application Servers: An application server typically runs network-based applications that implement certain business tasks, and its file system is not directly accessed by end users. System administrators are often cautious about installing antivirus software on such servers because it might interfere with the operation of the system's core application. If this applies to you, you may forego installing antivirus software on these servers, but you should still take other protective measures, such as configuration hardening.

Trend Micro, Inc.
EX1015-7

- Border Firewalls: A firewall located on the border of your network can often be configured to integrate with an antivirus server for scanning e-mail or Web-browsing traffic as it enters and leaves the organization's network. Catching malware at this choke point, before it further infiltrates your infrastructure, is a powerful weapon against malicious code.
- Handhelds: These lightweight devices often take the form of personal digital assistants (PDAs). As handheld vendors add wireless and other networking capabilities to these devices, and as the PDAs' processing and memory capacity increase, they will become a more likely target for malware. Although not many specimens have targeted handhelds so far, keep an eye on the evolution of this threat vector, and install antivirus software on PDAs when the risk of infection justifies the cost of deployment.

Depending on the complexity of your infrastructure and on your budget, you might not be able to install antivirus software at all these locations. That's okay, as long as you combine antivirus software that you do deploy with other methods of defending against malware that we discuss a bit later in this section. But please do yourself a favor—at least install antivirus software on user workstations, file servers, and mail servers.

Now that we've seen where you can install this software, let's focus on how it works. To allow you to make the most of your antivirus software, we'll discuss the strengths and weaknesses of the techniques antivirus software uses to detect malicious code, namely signatures, heuristics, and integrity verification.

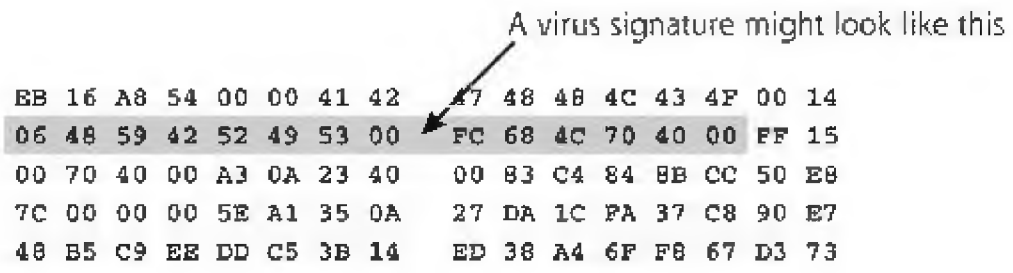
Virus Signatures

One of the simplest and most popular ways in which antivirus software detects malicious code is through the use of virus signatures. The antivirus vendors collect malware specimens and “fingerprint” them. Thousands of signatures are gathered together in a database for use in an antivirus scanner. The database of such signatures is distributed to sys-

Trend Micro, Inc.
EX1015-8

tems that require protection. When scanning files for malicious code, antivirus software compares the current file to its signature set and determines whether the file matches a signature of a known malware specimen. This process is depicted in **Figure 2.9**, which shows a file segment represented by hexadecimal characters, along with a sequence of bytes that a signature-based detector might recognize as a pattern that belongs to a virus.

Figure 2.9. Signature-based detectors look for familiar patterns in files to identify known malware specimens.



Antivirus software might attempt to locate familiar malware patterns on the fly, as the user accesses files on the protected system. The user can specify that all files should be scanned for malicious code in this manner; considering the variety of infection techniques, this is often the preferred configuration. As a more efficient but less thorough alternative, the user can require that only file types most likely to harbor viruses, such as .EXE, .COM, .DOC, and so on, be scanned. In environments where real-time scanning is not acceptable for performance reasons, users can manually request a scan by pointing the antivirus program to the files that need to be examined.

One of the biggest challenges to this signature-based method of malware detection is that antivirus software needs to include a signature for the virus to discover it on the victim's system. This means that antivirus vendors strive to collect new virus samples, develop patterns that fingerprint them, and distribute signature updates to the customers as quickly as possible. This is also the main reason it is so important to routinely update

Trend Micro, Inc.
EX1015-9

virus definitions on machines protected by antivirus software, downloading the latest signatures as often as once a day.

Luckily, modern antivirus software allows users to retrieve signature updates over the Internet without manual interaction. Symantec's Norton AntiVirus, for example, comes with a utility called LiveUpdate, shown in **Figure 2.10**. Users can schedule LiveUpdate to run automatically, or they can run the program on demand to download and install the latest virus patterns. Enterprise-centric versions of antivirus software give an organization additional control over how signature updates are distributed to its systems. For instance, Symantec's central management console allows administrators to define the update schedule, and offers the ability to monitor the effectiveness of signature deployment and virus protection mechanisms.

Figure 2.10. LiveUpdate, which comes with Norton AntiVirus, allows users to retrieve the latest virus signatures over the Internet.



Still, even with rapid and frequent updates, one of the weaknesses of the signature-matching approach is that it is always playing catch-up with

Trend Micro, Inc.
EX1015-10

malware authors as they release brand new, or even slightly modified, specimens into the wild. Additionally, a bad guy might create a custom virus, keeping it close to the vest before releasing it against a particular target. Without a widespread release, antivirus software developers cannot create a signature until the virus has been deployed against its target, which might be too late to stop major damage. Another significant disadvantage of this signature-based detection technique is that, in its pure form, it cannot identify malicious code designed to automatically change itself as it propagates, thereby modifying itself so that it doesn't match any signatures. We will look at such antidetection tricks in the "**Virus Self-Preservation Techniques**" section later in the chapter. As you can imagine, if a virus can continuously alter its code, then antivirus vendors will have a hard time devising a reliable signature for it.

Heuristics

Consider a situation in which you were tasked with identifying all world-class international spies that you might meet, but you did not know what they actually looked like. You could approach this challenge by first developing a matrix that listed known spy attributes and assigned points to them based on how strongly they indicate a spy. Your list might look something like this:

- Wears a stylish suit or a tuxedo (70 points).
- Survives catastrophes and other improbable situations (30 points).
- Drives a slick car (80 points).
- Never has a bad hair day (58 points).

The list could go on, but you get the idea. If the sum of all points for the individual exceeds a certain value, you might decide that he or she is probably a spy without ever seeing this particular spy before. Then, you can ask for a ride in the slick car.

Realizing the limitations of signature-based detection methods, antivirus vendors have devised similar ways in which they can detect previously

Trend Micro, Inc.
EX1015-11

unseen viruses that exhibit certain behavioral and structural characteristics. Symantec, for instance, calls this feature of its Norton AntiVirus product Bloodhound. A heuristics-based detection engine scans the file for features frequently seen in viruses, such as these:

- Attempts to access the boot sector.
- Attempts to locate all documents in a current directory.
- Attempts to write to an EXE file.
- Attempts to delete hard drive contents.

As the heuristics scanner examines the file, it usually assigns a weight to each virus-like feature it encounters. If the file's total weight exceeds a certain threshold, then the scanner considers it malicious code. If the scanner's developer sets the threshold too low, then the user could be overwhelmed with false alarms. On the other hand, if the threshold is set too high, or if virus-like features are not properly identified, then the detector will miss too many viruses. Either way, the user's protection is limited unless the sensitivity is set just right.

This technique would not be very helpful if antivirus software was able to detect malware only after the virus exhibited malicious behavior such as infecting programs or deleting files. If that were the case, you might get a warning from the antivirus software that says, “Your system has just been completely undermined by a virus! Have a nice day.” Although this is certainly interesting information, you need to get the warning before the malware has its way with your machine. The trick is to parse the suspicious file in a way that allows antivirus software to estimate what actions would be performed if the virus actually has a chance to execute. This analysis must occur before the code runs. Antivirus software accomplishes this goal by attempting to emulate the processor that would have executed the potentially malicious program. In the case of executables compiled for Intel x86 machines, this approach calls for emulating key features of the x86 processor. In the case of VBScript macros embedded into Microsoft Office documents, this approach requires emulating basic functionality of the VBScript processing engine.

Trend Micro, Inc.
EX1015-12

Considering the difficulty of reliably emulating a processor, heuristic detection approaches are far from foolproof. It is especially challenging to assess the effects of macro-based viruses, because their structure and possible execution flows are much less predictable than those of compiled executables. As a result, virus scanners do not rely on heuristics as the sole approach to detecting viruses—they also use the good old signature technique, and sometimes they also employ the integrity verification method described next.

Integrity Verification

When defending against viruses, we are dealing with creatures that modify their host programs as they spread. Therefore, one way to detect the presence of a virus is to discover files that have been unexpectedly modified. The integrity verification process aims to achieve this goal by following these steps:

1. While the machine is in a pristine state, compute fingerprints (in the form of checksums or cryptographic hashes) of files that need to be monitored, and record them in a baseline database.

2. When scanning the file system for suspicious modifications, compute fingerprints of monitored files and compare the values to those in the baseline.

3. If unexplained differences between the current state and the baseline are detected, issue an alert.

There are several commercial and free applications that are dedicated to implementing such integrity verification procedures. The most famous of these tools is probably Tripwire (available at **www.tripwire.com**), which has been capable of detecting unauthorized changes to the file system since it was first released in 1992. Tripwire and other software of this

Trend Micro, Inc.
EX1015-13

type are not virus checkers per se—such programs aim at alerting administrators of suspicious changes to the machine's state regardless of whether the attack was performed by malware or was executed through some other channel. We'll discuss these file integrity checking tools in more detail in **Chapters 6** and 7 when we analyze Trojan horses and RootKits.

Integrity verification approaches can also be used by antivirus software, although vendors are rarely forthcoming about the extent to which they have implemented such mechanisms. Sophos AntiVirus is known to use checksums to help determine whether a file needs to be examined more carefully via other detection methods. When scanning a file, Sophos AntiVirus computes the file's checksum and compares it to the value calculated earlier. If the checksums do not match, then there is a chance that the file was infected, and the antivirus program might need to examine it more thoroughly **[23]**.

An antivirus product trying to make the most of integrity verification techniques is likely to be selective about the portions of the file that are fingerprinted for baseline comparisons. For example, it could be okay for the contents of a Microsoft Word document to change when the user edits its text; however it is far less common for the macros embedded in the document to be modified. Therefore, antivirus software might be more suspicious of changes detected in the macros section of the document.

The main limitation of the integrity verification method is that it detects the infection only after it occurs. However, it is a useful addition to the toolkit consisting of approaches that look for signatures of known malware specimens and those that use heuristics to detect harmful code. Unfortunately, even antivirus software that implements each of these detection techniques will not be able to catch all malware that comes our way. To add a belt to our antivirus suspenders, we can use configuration hardening, which offers additional protection against malware attacks.

Configuration Hardening

Trend Micro, Inc.
EX1015-14

Configuration hardening is a powerful defense against viruses because it focuses on making the environment less likely to be infected, as well as on impeding the spread of viruses should infection occur. This defensive technique typically incorporates the following security goals that work in concert with each other:

- The principle of least privilege dictates that access to data and programs should be limited to those files that the user explicitly requires to accomplish business tasks. Sometimes, the principle of least privilege is abbreviated POLP, and affectionately pronounced "polyp."
- Minimizing the number of active components involves disabling functionality that the system does not need to serve its business purpose.

These security goals are key aspects of setting up reliable defenses against all types of computer attacks, whether they involve malicious code or not. There are entire books and comprehensive courses dedicated to locking down the configuration of individual operating systems and applications. However, if you distill much of the information associated with hardening an operating system's configuration, a few particularly relevant recommendations pop out in the context of stopping virus infections:

- If you have global administrative privileges, perform your day-to-day tasks while logged into a regular, unprivileged account. Then, use tools such as `su` and `Runas.exe` to perform tasks that require superuser rights. Never, ever, ever surf the Web or read e-mail while logged in as a root user on UNIX or any user in the Administrators' group on Windows. You're just asking for trouble if you surf or read e-mail in this way, because all malware inside your browser or mail reader will run with superuser privileges.
- Disable or remove unnecessary services and tools that were installed as part of the default operating system image on your workstations and servers.
- Use file systems that allow you to restrict access to sensitive files—in Windows environments, migrate from FAT to the NTFS file system. NTFS is far more secure than the relatively very weak FAT. Whereas

Trend Micro, Inc.
EX1015-15

Windows allows you to choose between a really weak and more secure file system, most UNIX file systems include some built-in security capabilities.

- Configure group membership for user accounts and enforce access restrictions in a way that prevents a virus that executes with the privileges of a single user from infecting all files on your file server.
- Use free tools such as Bastille Linux (**www.bastille-linux.org**), JASS (**www.sun.com/software/security/jass**), and Windows Security Templates to automate the implementation of numerous other hardening recommendations. We'll discuss Bastille and some specific Windows security templates in more detail in **Chapter 7**.
- Use assessment tools, such as those freely available from the Center for Internet Security (CIS; available at **www.cisecurity.org**), to compare the configuration of your systems with benchmarks based on established best practices. This CIS assessment tool is also covered in more detail in **Chapter 7**.

In addition to those high-level suggestions, there are a few application-specific security measures that warrant special mention. These are the defensive mechanisms built into Microsoft Office to help protect documents against infection. Because Office is so widely used and so often attacked, we need to address these very specific recommendations. If implemented properly, they can be surprisingly effective at diminishing the threat associated with macro viruses.

Although Microsoft Word has supported macros since version 2.0, the reality is that most of the legitimate documents that we exchange do not contain macros. After slowly realizing this a few years after releasing Word 2.0, Microsoft enhanced the program with a warning such as the one shown in **Figure 2.11**, which alerts the user when a document contains macros.

Figure 2.11. Microsoft Word can warn the user if the document that is being opened contains macros.

Trend Micro, Inc.
EX1015-16



The major problem with presenting users with a warning like this is that they might click Enable Macros without thinking of the consequences, thus triggering malicious code embedded in a document. Luckily, the default installation of Microsoft Office XP and later versions is set up to silently disable untrusted macros, without presenting the user with the warning at all. This level of trust is based on digitally signed code, and can be configured by the user in the settings of the program. Microsoft Office will process such macros based on the security level defined by the screen shown in **Figure 2.12**. You can access this feature from the Tools ➤ Macro ➤ Security menu.

Figure 2.12. Microsoft Office processes macros according to the security level established by the user.

Trend Micro, Inc.
EX1015-17



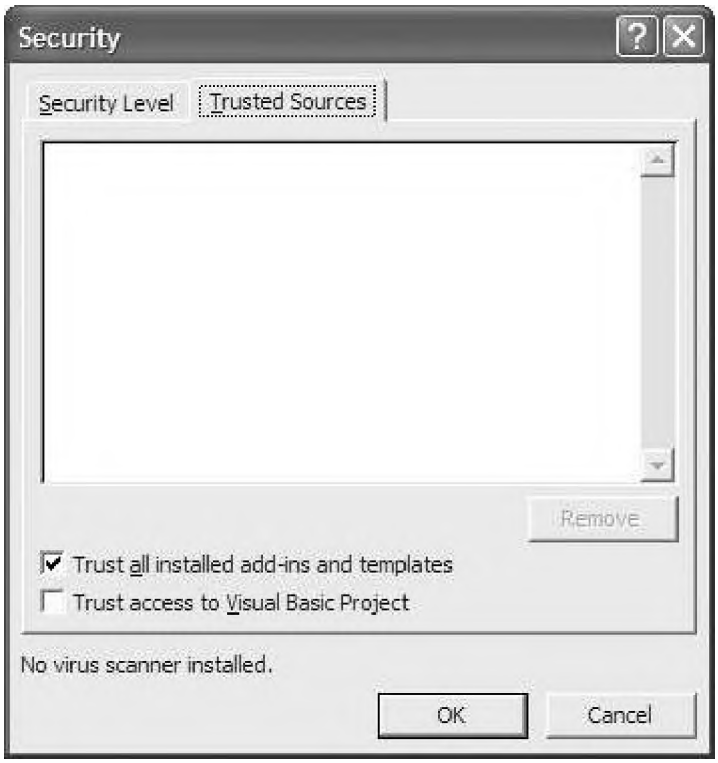
When the security level is set to High, even if the user opens a document that contains malicious code, the macros will be silently disabled and the virus will lie dormant. For macros to be executed in this configuration, they have to be digitally signed by a trusted source, which is unlikely to occur in the case of a virus. This is definitely the recommended setting for most organizations, and I am happy that it has become the default configuration for Microsoft Office.

If the security level is set to Medium, the user will be presented with the alert that you saw in **Figure 2.11**, unless the macro is digitally signed by a trusted source. I sincerely hope you will not have to consider setting the security level to Low, because in this configuration Microsoft Office will run all macros embedded in the document without prior warning. Such a configuration is very dangerous, akin to running backwards with scissors up and down stairs, while blindfolded and chewing gum.

There is an additional layer of defense built into Microsoft Office to help prevent macro-based infections. It is enabled by default in Office XP and later versions, and is activated as long as the Trust Access to Visual Basic Project check box is unchecked, as shown in **Figure 2.13**. Most macro

viruses copy themselves from the current host to another document through the use of commands that Microsoft Office implements in its Visual Basic Project (VBProject) object. This security setting tells Office to block access to the VBProject object, thereby disabling the macro commands typically used to copy virus code to new documents. Without being able to copy its code, the virus cannot spread. To ensure that add-ins and templates distributed as part of Microsoft Office continue to work, the Trust all installed add-ins and templates check box is enabled by default, a pretty reasonable setting for the default installation.

Figure 2.13. Not trusting access to Visual Basic Project helps prevent macro viruses from infecting documents on the system.



For enterprisewide management in organizations that have deployed Microsoft's Active Directory, you can even use Group Policy to centrally define these settings, and prevent users in your organization from tinkering with them. To accomplish this task, you will need to obtain Microsoft Office template files that come with the Office Resource Kit, a separate set of software you can purchase. After loading the templates into Group Policy Editor and browsing to the appropriate section of the template, you

Trend Micro, Inc.
EX1015-19

will have the ability to establish macro restrictions for each Microsoft Office application.

The techniques we've discussed so far go a long way in dealing with the virus problem. However, regardless of the number of technical measures you take to protect data, human beings who use the system could remain a weak link in your security infrastructure. You could implement beautiful and perfect technical security measures, if such perfection exists. However, careless users could accidentally undermine all of your security, unless you take the time to educate them about their role in counteracting virus threats.

User Education

More often than not, end users of our systems activate viruses simply because they don't know any better. You need to help them help you, and educate your users about the importance of protecting data. Tell them about the techniques that malicious software uses to spread and help them understand what they can do to prevent and detect malware infections. With this in mind, I suggest incorporating advice like this into your security awareness training materials and policies:

- Do not attempt to disable defensive mechanisms such as antivirus software or macro security settings when something doesn't seem to work. Instead, contact the help desk or system administrators so that they can help you resolve the problem in an efficient manner without compromising security.
- Be cautious of attachments that are not documents with which you routinely work. In particular, do not open executable files, even if they come from your friend and claim to be a cute little game.
- Do not download and install programs from external sources, even if they are simple utilities you might be able to install on your own. Sources for obtaining software should be approved by the system administrator in advance.

Trend Micro, Inc.
EX1015-20

- Do not connect your own systems to the organization's network. Only corporate-approved and managed systems should be interconnected with the organization's network. This includes personal laptops and home computers connecting over the company's virtual private network (VPN).
- Learn to recognize signs of a virus infection, such as sluggish performance, system crashes, bounced e-mail, and anti-virus warnings. Alert the system administrator if you observe suspicious behavior or if you receive files that you believe might be infected. When in doubt, ask for help.
- Do not forward virus warnings to your friends and colleagues as soon as you receive them. Check with Web sites such as **www.truthorfiction.com** and **hoaxbusters.ciac.org** to check if the warning is a hoax, and contact the help desk or system administrator if you are still concerned about the alert.

Antivirus software, configuration hardening, and user education are some of the essential tools for fighting virus infections, especially when combined with other malware defense techniques we'll discuss throughout the remainder of the book. However, as hard as we try to block and detect malicious code, malware authors try to bypass our security mechanisms. It is a never-ending race, as we'll see in the next section.

Trend Micro, Inc.
EX1015-21

Additional Defenses against Malicious Mobile Code

While examining the risks associated with malicious mobile code, we've also looked at the applicable approaches to mitigating them. Before we look at some additional defensive measures, here's a high-level overview of the most critical protective mechanisms that we have covered so far:

- Surf the Internet and read email from a non-superuser account (i.e., not a root or administrator).
- Stay aware of vulnerability and patch announcements for browser and e-mail software that you use.
- Apply relevant patches or workarounds in a timely manner.
- Be mindful while visiting rogue Web sites that might attempt embedding XSS exploits in hyperlinks.
- Be mindful of clicking URLs in e-mail messages that might attempt embedding XSS exploits in hyperlinks.
- Do not execute ActiveX controls, whether signed or not signed, unless you trust their author with access to your system.
- Do not execute signed Java applets unless you trust their author with access to your system.
- Remember that there is no such thing as “trust once,” when it comes to ActiveX controls or Java applets, because a malicious program can grant itself perpetual trust once it has access.
- Consider disabling support for HTML rendering in your e-mail software if you don't really need it.
- Disable support for third-party cookies in your browser.
- Disable support for mobile code that you do not require in your browser and e-mail software.
- Enforce the appropriate restrictions in your browser and e-mail software for mobile code that you cannot fully disable.

That's quite a huge and exhausting list! Sadly, the complexities of modern mobile code implementations in browsers and e-mail readers are im-

Trend Micro, Inc.
EX1015-22

mense, making these defenses necessary. Beyond this list, there are three more defense mechanisms that can come in handy in the fight against malicious mobile code: antivirus software, behavior monitors, and anti-spyware tools.

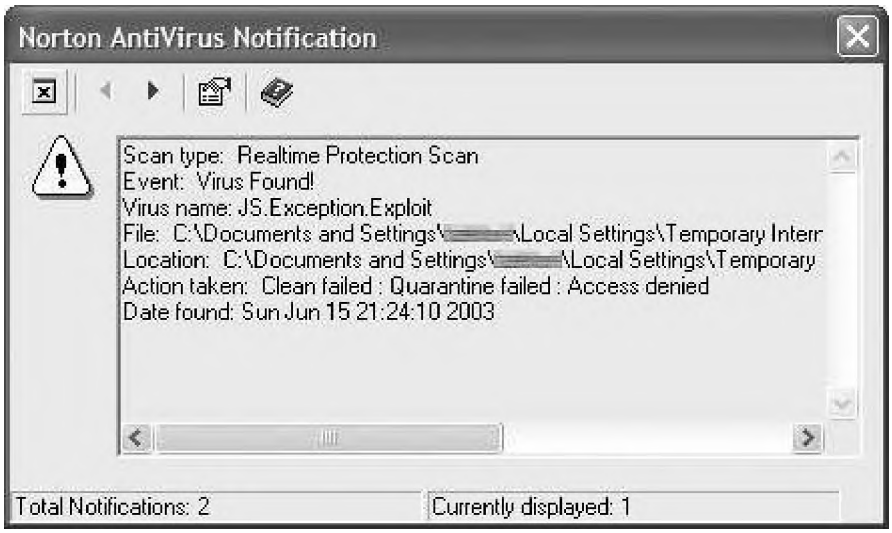
Antivirus Software

We already looked at antivirus software in **Chapter 2** when discussing computer viruses, but those same defensive ideas also apply to malicious mobile code. Current versions of most antivirus packages are able to automatically scan the contents of a Web page before your browser has a chance to process it. This means that they might be able to capture malicious mobile code before it even gets to the browser by matching various malicious mobile code signatures. That's the good news. The bad news is that there are so many variations of attacks based on mobile code that very few of them are actually detected by antivirus software. Creating effective signatures for the multitude of different attack vectors would hog enormous resources on the system, and still not cover every single malicious mobile code possibility.

Antivirus software is able to recognize some of the malicious code examples that are included in widely circulated vulnerability advisories, such as those posted to the Bugtraq mailing list. For example, **Figure 4.17** shows you a screenshot of Norton AntiVirus blocking a script embedded in a malicious Web page. More detailed analysis of the file that triggered this warning revealed that it contained a variation of the Scriptlet.Typelib attack. Even if the user's browser wasn't patched, this layer of defense would block the attack. You should definitely take advantage of the protection that antivirus software offers. However, given the relative lack of effective signatures, antivirus tools shouldn't be your sole defense against malicious mobile code. They should only be used in coordination with the other defenses covered throughout this chapter.

Figure 4.17. Antivirus software can detect some of the more common malicious mobile code specimens.

Trend Micro, Inc.
EX1015-23



Behavior-Monitoring Software

Instead of relying on signatures to detect known malware specimens, behavior blockers keep an eye out for suspicious actions to recognize and block behavior typically associated with malicious code. For example, attempts to perform the following actions could suggest that the program is malicious:

- Writing to sensitive portions of the registry
- Creating a file on a system directory
- Overwriting browser settings
- Adding a browser plug-in

One tool that uses a behavior-based technique to protect workstations against malicious mobile code is Finjan SurfinGuard, which you can purchase from **www.finjan.com**. SurfinGuard runs in the background on your machine, creating a tight sandbox around programs and browser scripts that would otherwise have unrestrained access to the system's resources. For example, **Figure 4.18** shows a warning that SurfinGuard popped up when a malicious site managed to push a malicious ActiveX control to a system in my lab. The control slipped through Internet Explorer's security settings because I purposefully misconfigured them—this resembles an environment where a careless user agreed to install an untrusted ActiveX control, or where the browser contained an unpatched vulnerability.

Trend Micro, Inc.
EX1015-24

Figure 4.18. SurfinGuard can intercept and block actions taken by suspicious executables.



My browser downloaded the ActiveX control and attempted to execute a program named Loader.exe. Fortunately, SurfinGuard alerted me about this activity, giving me an option to block the action or to continue monitoring the suspicious executable. Without this layer of protection, my slightly misconfigured workstation would fall victim to the malicious ActiveX control.

Similar behavior-blocking functionality exists as part of the Tiny Personal Firewall (TPF) product sold by Tiny Software at **www.tinysoftware.com**. TPF allows the user to categorize programs into groups based on their trustworthiness, with each group subject to different restrictions. When protecting the misconfigured machine that I described in the previous example, TPF offered to execute the suspicious program within a sandbox appropriate for the Restricted Applications group, as you can see in **Figure 4.19**.

Figure 4.19. Tiny Personal Firewall is able to intercept actions taken by an unfamiliar application and restrict them according to the appropriate group profile.

Trend Micro, Inc.
EX1015-25



Behavior-monitoring tools such as SurfinGuard and TPF help protect workstations against malicious mobile code, because they can often recognize a new malware specimen without relying on signatures that might not even exist for many mobile specimens. In addition to supplementing traditional antivirus software, behavior monitors play well with tools that specifically target spyware programs.

Antispyware Tools

Beyond antivirus and behavior-monitoring software, there are numerous free or inexpensive utilities on the market that specialize in the defense against spyware on Windows operating systems. These nifty tools really come in handy in defending against the predatory practices of spyware and some aggressive advertisers. You already saw one such utility, BHODemon, earlier in this chapter. We used it to detect and remove unwanted BHOs that plugged themselves into Internet Explorer. Although free, BHODemon is a relatively limited tool that requires a user to recognize plug-ins that don't belong in their browser.

Ad-aware is a popular antispyware tool that is more full-featured and user-friendly than BHODemon. You can download Ad-aware free for non-commercial use from **www.lavasoftusa.com**. Ad-aware can recognize the presence of a large number of known spyware tools, cookies, and browser-hijacking programs by examining the list of currently running processes, as well as by scanning the machine's file system and the registry. **Figure 4.20** shows you the summary of Ad-aware scan results on the systems where I installed the Go!Zilla download manager, which is notorious for containing spyware components.

Figure 4.20. Ad-aware scans the system for signs of known spyware and browser-hijacking programs.



Clicking Next displays a detailed listing of the detected problems, a description of how the user can eliminate the threats, and links to obtain additional information about them. For instance, one of the problems that Ad-aware detected on my Go!Zilla machine warned me about the `mmod.exe` process running the system, showing the following details:

```
Vendor:      EzuLa
Category:      Data Miner
Object Type:   Process
```


Trend Micro, Inc.
EX1015-27

```
Size:          -
Location:      c:\program files\ezula\mmod.exe
Last Activity: 6-15-03 4:00:00 AM
Risk Level:    High
Comment:
Description:   Thiefware. Inserts its own yellow links
               on the website you are visiting.
```

Although the description that Ad-aware presented in this case is not very thorough, it alludes to EzuLa's functionality that inserts yellow hyperlinks on Web pages whose contents match advertisement-related keywords. The commercial version of Ad-aware adds support for real-time blocking of known spyware objects, as well as suppressing pop-ups, ActiveX controls, browser hijacks, and other activity often associated with malicious mobile code.

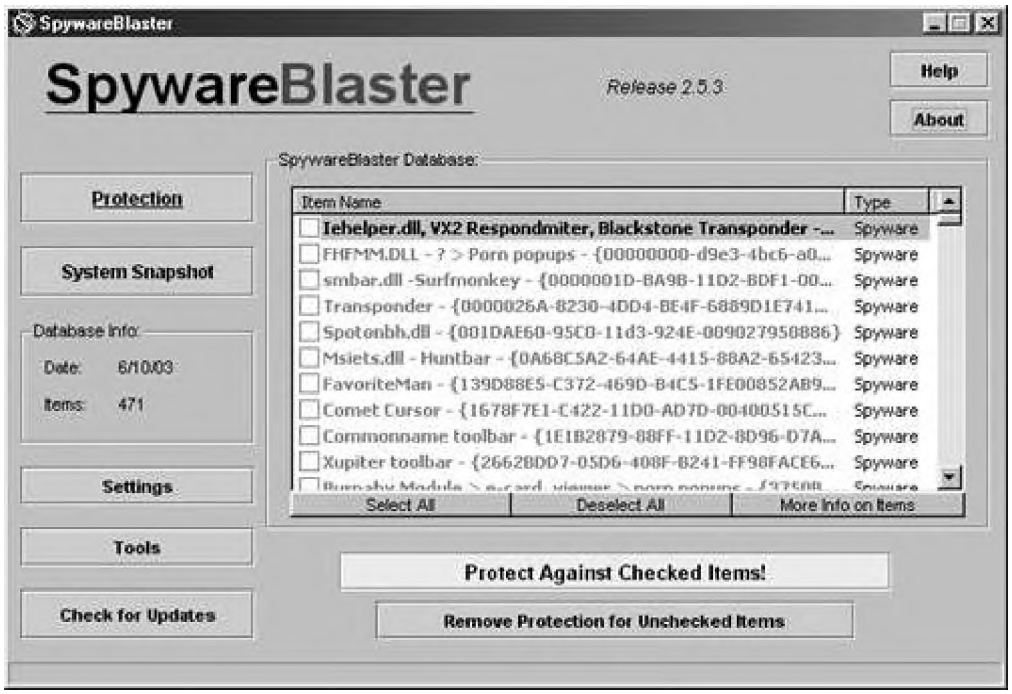
If you need a program with many of the features of the Ad-aware freeware without the restrictions on commercial use, take a look at Spybot—Search & Destroy. You can download this program for free from **http://security.kolla.de**. Although it is not quite as user friendly as Ad-aware, Spybot comes with several useful tools for no extra charge. For example, Spybot can register itself as an Internet Explorer BHO, thereby recognizing and intercepting known spyware installers before they get to the system. It can also generate a list of all currently installed BHOs, show you what processes are running on the system, and let you control which programs are started during boot time.

For a simple but effective approach to preventing Internet Explorer from activating malicious ActiveX controls, take a look at SpywareBlaster at **www.wilderssecurity.net**. Instead of detecting spyware, SpywareBlaster sets the kill bit for all malicious ActiveX controls that it knows of. As we discussed earlier in the chapter, Internet Explorer just won't run an ActiveX control with its kill bit set. **Figure 4.21** shows the program's screen where the user can decide which ActiveX controls to block. As SpywareBlaster reminds its users, this is the list of objects that the pro-

Trend Micro, Inc.
EX1015-28

gram can protect against, not a list of malicious objects currently installed on the machine. That certainly is an important distinction!

Figure 4.21. SpywareBlaster can set the kill bit for malicious ActiveX controls to prevent Internet Explorer from activating them.



Another useful feature of SpywareBlaster is the ability to take a snapshot of the system's registry settings that are frequently targeted by spyware and browser-hijacking software. Having a snapshot of your system in its pristine state will allow you to easily restore those settings if malicious software modifies them.

This whole antispyware genre is growing rapidly, indicating the magnitude of this problem. In no particular order, other solid antispyware tools available on both a free and commercial basis include the following:

- Spy Sweeper: **www.webroot.com**
- PestPatrol: **www.pestpatrol.com**
- SpyStopper: **www.itcompany.com**

Trend Micro, Inc.
EX1015-29

Keep in mind that, just like antivirus software, antispyware programs usually rely on signatures to detect malicious mobile code. They typically include an easy way of retrieving the latest signature definition database over the Internet, so be sure to take advantage of this feature, keeping your signature base regularly updated. By keeping your signatures up to date, you'll be able to protect yourself against the constant march forward of malicious mobile code.

Trend Micro, Inc.
EX1015-30