



realtimepublishers.com[™]

The Definitive Guide[™] To

Controlling Malware, Spyware, Phishing, and Spam

McAfee[®]
Proven Security[™]

Dan Sullivan

Introduction to Realtimepublishers

by Don Jones, Series Editor

For several years, now, Realtime has produced dozens and dozens of high-quality books that just happen to be delivered in electronic format—at no cost to you, the reader. We’ve made this unique publishing model work through the generous support and cooperation of our sponsors, who agree to bear each book’s production expenses for the benefit of our readers.

Although we’ve always offered our publications to you for free, don’t think for a moment that quality is anything less than our top priority. My job is to make sure that our books are as good as—and in most cases better than—any printed book that would cost you \$40 or more. Our electronic publishing model offers several advantages over printed books: You receive chapters literally as fast as our authors produce them (hence the “realtime” aspect of our model), and we can update chapters to reflect the latest changes in technology.

I want to point out that our books are by no means paid advertisements or white papers. We’re an independent publishing company, and an important aspect of my job is to make sure that our authors are free to voice their expertise and opinions without reservation or restriction. We maintain complete editorial control of our publications, and I’m proud that we’ve produced so many quality books over the past years.

I want to extend an invitation to visit us at <http://nexus.realtimepublishers.com>, especially if you’ve received this publication from a friend or colleague. We have a wide variety of additional books on a range of topics, and you’re sure to find something that’s of interest to you—and it won’t cost you a thing. We hope you’ll continue to come to Realtime for your educational needs far into the future.

Until then, enjoy.

Don Jones

Introduction to Realtimepublishers.....	i
Chapter 1: Overview of Preventing Malware, Spyware, Spam, and Phishing Scams.....	1
Proliferation of Internet Content and the Roles of Internet Content in Organizations	2
The Internet: A New Kind of Infrastructure	2
Organizational Responsibilities for Securing the IT Environment.....	4
Protection of Information Assets	4
Systems Infrastructure	5
Core Applications and Supporting Software	5
Protecting Data.....	9
Efficient Operations	10
Non-Threatening Work Environment.....	11
Protecting Against Loss of Controlled Information.....	11
Regulatory Compliance	12
Compliance and Security	13
Summary of Organizational Responsibilities	13
Threats Posed by Internet Content.....	14
Viruses, Worms, Trojan Horses, and Other Malware.....	14
Computer Viruses	14
Worms.....	16
Trojan Horses.....	16
Additional Forms of Malware.....	17
Spyware.....	17
Phishing.....	18
Spam	19
Undesirable Web Sites	19
Countermeasures to Threats.....	20
Content Filtering.....	20
URL Filtering.....	20
Antivirus Software	21
Anti-Spyware	21
Spam Management.....	21
Countermeasure Implementations	22
Summary	22

Chapter 2: Organizational Responsibilities for Protecting the Network from Internet Attacks	23
Protecting Employees	23
Harassment in the Workplace	23
Offensive Material in the Workplace.....	26
Protecting Information Assets.....	26
Preventing Virus and Other Malware Attacks	27
Infection Rate.....	27
Server Downtime	28
Recovery Time.....	29
Recovery Cost.....	30
Other Factors and Emerging Trends	30
Malicious Use of Computers	32
Spamming with Zombies	32
Committing Click Fraud with Zombies	35
DDoS Attacks	36
Preventing Wasted Bandwidth and Storage.....	39
Spam and Unnecessary Storage	40
Protecting Customers	41
Preventing Disclosure of Confidential Information.....	42
Protecting Customers from Identity Theft	42
Protecting Stakeholders	43
Preventing Non-Business Web Activity	44
Complying with Regulation.....	44
Avoiding the Cost of Security Breaches.....	45
Summary	45
Chapter 3: Viruses, Worms, and Blended Threats.....	46
Evolution of Viruses and Countermeasures.....	46
The Early Days of Viruses.....	47
Beyond Annoyance: The Proliferation of Destructive Viruses	48
Wiping Out Hard Drives—CIH Virus	48
Virus Programming for the Masses 1: Macro Viruses.....	48
Virus Programming for the Masses 2: Virus Generators.....	50
Evolving Threats, Evolving Countermeasures	51

Detecting Viruses.....	51
Radical Evolution—Polymorphic and Metamorphic Viruses	53
Detecting Complex Viruses	55
State of Virus Detection.....	55
Trends in Virus Evolution.....	56
Worms and Vulnerabilities	57
Early Worms	57
Implementation Techniques and Consequences	57
Sobig	58
MyDoom.....	58
Sdbot	59
SQL Slammer.....	60
Increasing Malevolence	62
The New Frontier—Blended Threats.....	63
Multiple Methods of Attack.....	63
Multiple Methods of Transmission.....	64
Multiple Methods of Control	64
Summary	65
Chapter 4: Spyware and other Potentially Unwanted Programs.....	66
Varieties of PUPs.....	67
PUPs.....	67
Adware.....	68
Spyware.....	69
Keyloggers	69
Password Stealers.....	71
Tracking Cookies	72
Defining Spyware	76
Installation Methods and Effects	77
Concealment	77
Injection	77
Payload Types.....	78
Spyware and other PUP Behaviors	79
File Scanning	79

Reading Cookies	79
Changing Browser Settings.....	80
Installing Browser Help Objects	81
Malware vs. Malware.....	82
Impact of Spyware	82
Reduced Computing Performance	82
Performance Effects of Monitoring	83
Number of PUPs	83
Loss of Proprietary and Confidential Information.....	84
Help Desk Costs.....	84
Summary	85
Chapter 5: Phishing and Identity Theft.....	86
Anatomy of Phishing Scams	86
Social Engineering and Phishing	87
Direct Questioning	87
Incremental Accumulation of Information	88
Physical Access.....	88
Reverse Social Engineering	88
Internet-Based Social Engineering	89
Phishing Trap.....	93
Limits of Social Engineering-Based Phishing	94
Malware-Based Phishing Techniques.....	95
Botnets and Phishing Attacks	95
Malicious Software for Phishing	97
Exploiting Browser Vulnerabilities	99
DNS Attacks	100
Economics of Phishing and Identity Theft.....	102
Impact on Consumers	104
Phishing and Identity Theft Countermeasures.....	105
Business Countermeasures.....	105
Consumer Countermeasures	106
Summary	106
Chapter 6 Spam in the Enterprise	107

Email Operations and Spam Techniques	107
Early Spam and Reactions to It.....	107
The Basics Steps in Email Operations	108
Email Clients.....	108
Email Servers	109
Vulnerabilities in Email Infrastructure	110
Economics of Spamming	112
Costs and Revenues of Spamming.....	113
Negative Externalities of Spam	114
Costs Incurred by Others	114
Distorting the Supply/Demand Balance.....	115
Correcting for Negative Externalities: Government Regulation	116
Beneficiaries of Spam.....	119
Spam Management.....	120
Detection and Determination	121
Integrity Analysis.....	121
Heuristic Detection	121
Content Filtering	122
Blacklists and Whitelists.....	122
Self-Tuning	123
Bayesian Filtering	123
DNS Block Lists	124
Actions in Response to Spam	125
Blacklist and Whitelist Management	125
Quarantine Access, Search, and Management.....	126
Evaluating Anti-Spam Systems	126
Catch Rates	127
False Positives.....	127
Manageability and Reporting.....	127
Summary	128
Chapter 7 Technologies for Securing Information and IT Assets	129
Content-Specific Technologies for Information Security.....	129
Email Content Filtering.....	130

The Evolution of Email Functionality and Vulnerabilities.....	130
Harassment and Hostile Workplace Issues	132
Information Leakage and the Loss of Intellectual Property.....	133
Disclosure of Private Information.....	134
Email Content-Filtering Techniques.....	135
Multiple Techniques for Email Content Filtering.....	144
Filtering Spam.....	144
Policies and Actions for Filtered Content.....	145
Web Browsing and URL Blocking.....	145
Multi-Layered Security: Defense in Depth.....	147
Gateway Defenses.....	147
Network Security Measures.....	147
Desktop Security Measures.....	148
Summary.....	148
Chapter 8: Implementation Issues in Securing Internet Content	149
Choosing a Secure Content Mechanism	149
Comprehensive Coverage	150
Threats in Email Messages	150
Threats in HTTP Traffic	151
Threats in Instant Messages.....	152
Threats from File Transfers	152
Secure and Reliable Platforms for Content Security	153
Support Open Standards	154
Easily Customized Black Lists and White Lists.....	155
Managing Black Lists	156
Managing White Lists.....	156
Align with Organizational Structures and Line of Business Needs.....	157
Provide Adequate Reporting.....	158
Application Status.....	158
System Performance	159
Significant Event Logs.....	160
Configuration and Update Status	160
Compensating for Simple Bypass Techniques	161

Key Features of a Secure Content System	162
Implementation Options of Secure Content Systems	162
Implementation Option 1: Secure Content Server Appliance.....	162
Standardized Hardware	162
Standardized Software	163
Advantages of Appliances	164
Implementation Option 2: Software Applications for Secure Content Management	165
Implementation Option 3: Outsourced Service.....	166
Choosing Among the Options.....	167
Management Issues in Secure Content Management	168
Appropriate Use	168
Auditing and Reporting.....	169
Quarantining and Deleting Content	169
Black List and White List Maintenance.....	169
Capacity Planning	169
Best Practices in Secure Content Management	169

Copyright Statement

© 2006 Realtimepublishers.com, Inc. All rights reserved. This site contains materials that have been created, developed, or commissioned by, and published with the permission of, Realtimepublishers.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 Realtimepublishers.com, Inc or its web site sponsors. In no event shall Realtimepublishers.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.

Realtimepublishers.com and the Realtimepublishers 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 Realtimepublishers.com, please contact us via e-mail at info@realtimepublishers.com.

[**Editor's Note:** This eBook was downloaded from Realtime Nexus—The Digital Library. All leading technology guides from Realtimedpublishers can be found at <http://nexus.realtimedpublishers.com>.]


Chapter 1: Overview of Preventing Malware, Spyware, Spam, and Phishing Scams

The Internet is a double-edged sword. It is fundamental infrastructure for contemporary businesses and organizations—the Internet has evolved from research tool to the basis for more efficient production and better distribution of information. We have all benefited from new services such as virtual marketplaces and online comparison shopping to instant access to wide ranges of information from a single search engine. However, while realizing these benefits, we have also opened ourselves and our systems to a number of threats.

The Internet is home to malicious programs that can steal, destroy, and make data inaccessible; spyware that ignores social conventions of privacy to track Internet users' activities online; phishing scams that bring the art of the con artist to new threatening levels; and spam, the inevitable electronic counterpart to direct mail that requires so little investment even miniscule response rates justify its use. Consider just a few examples:

- Phishers have claimed to represent Citibank, SunTrust, and Bank of America with emails to customers notifying them of alleged attempts to log on to online accounts from foreign countries or the need to update customer information.
- Brilliant Digital Entertainment has embedded spyware in the popular Kazaa file sharing program to track users' online activities as well as to add infected PCs to a distributed network controlled by the company.
- Spammers claiming to be relatives of deposed government officials in politically unstable countries promise millions of dollars in return for an advance fee to help the alleged victim flee their country. These are known as 419 frauds after the section of the Nigerian criminal code that makes such spam messages illegal.

Imagination is the only limit on the types of fraud and misappropriation of computing resources that arise on the Internet.

 For more details about these and other threats, see “Stealth P2P network hides inside Kazaa” at <http://news.com.com/2100-1023-873181.html>, the anti-phishing Web site at <http://www.millersmiles.co.uk/>, and the spam archive at <http://www.spamarchive.org/>. Expert Law has a good layman description of email frauds at http://www.expertlaw.com/library/consumer/spam_email_fraud2.html.

Proliferation of Internet Content and the Roles of Internet Content in Organizations

Although Internet threats abound, the value of standardized, cross-organization, distributed computing is so great that organizations find it too compelling to ignore. For example, many services we take for granted are built on Internet services:

- **Email**—Email is now a standard means of communications within and between organizations. Its advantages—such as low cost, broad and rapid distribution, and electronic copies that can be stored for long periods of time—make it a superior method of communications compared with telephones and postal mail.
- **E-commerce**—Businesses and organizations find what they need faster and with greater options because of online catalogs and ordering, customer self-services, virtual marketplaces, comparison shopping sites, and online consumer reviews.
- **Workflow**—Complex business processes are routinely broken down into smaller steps, and workflow software allows greater control and efficiency than possible in the past. Some of the benefits provided by workflow software include:
 - Embedded business logic that can automatically control the flow of information and tasks
 - The ability to distribute work tasks anywhere in the world, which allows businesses to find the optimal combination of cost and quality
 - Process management and reporting, which provide managers with data on the status of production throughout complex business operations

The Internet has become the foundation of services and operations to the point that we can think of it as a utility, like electric and water services, but with some challenging differences.


The Internet: A New Kind of Infrastructure


Clearly the Internet has demonstrated its value as well as its potential dangers. As with other kinds of infrastructure, we need to set up the Internet to maximize benefits and minimize dangers. Take electricity for example. Does anyone think twice about plugging an appliance into an outlet? We assume the electricity will be available at the proper voltage and amps to run the appliance without damaging it. Most of us know the basics of how power is generated and transmitted, and some are familiar with how power companies adjust the load on the electrical grid to keep meeting demand without overloading the system. Except for engineers in the industry though, the vast majority of electric users do not know the subtle nuances and technical details of how that supply and demand balance is met. Nonetheless, we can all work a toaster. The electrical system in fully industrialized countries has matured to the point where we can use it reliably and consistently without having to worry about a lot of technical details.

Similarly, while most Internet users understand the basics of networking, they do not have to concern themselves with the idiosyncrasies of protocols, routers, and servers. Unlike the electric grid, however, the Internet cannot be trusted to function as expected in all cases. If the Internet were used as benignly envisioned by most, users would be fine. Unfortunately, electronic utopias are no more real than social ones. The analogy between the electric grid and the Internet breaks down in two key ways.

- **Openness**—First, unlike the electric grid, which has relatively few producers able to substantially alter its state, the Internet is open to all.
- **Complexity**—Second, the network is more complex; the Internet communicates instructions and data around the world, it does not just transmit energy in a narrowly controlled manner.

This combination of openness and complexity is the underlying reason for both the Internet's benefits and its threats.

 If you are not convinced a single person could substantially affect the Internet, read about the effects of SQL Slammer, a malicious program known as a worm that spread so rapidly that Internet traffic was effectively shut down within minutes of the program's release. For more information about this worm, see CNN's "Computer Worm Grounds Flights; Blocks ATMs" at <http://www.cnn.com/2003/TECH/internet/01/25/internet.attack/> and CNET's "Slammer Attacks May Become Way of Life for Net" at http://news.com.com/Damage+control/2009-1001_3-983540.html.

 For a high-level summary of how the Internet and related technologies have changed businesses and organizations, see Thomas Friedman's *The World is Flat: A Brief History of the Twenty-First Century*, (Farrar, Straus and Giroux, 2005).

This guide explores some of the most pressing threats to businesses and organizations from the Internet along with best practices for addressing them. Chapter 1 presents an overview of the problem and includes:

- Organizational responsibilities for securing the IT environment
- Threats posed by Internet content
- Countermeasures to threats
- Countermeasure implementations

Chapter 2 examines what organizations can do to protect employees, customers, stakeholders, and information assets from the malicious elements on the Internet. In Chapter 3, the topic turns to viruses, worms, and an especially virulent combination of malware known as blended threats. Chapter 4 addresses how to control spyware in the enterprise with detailed explanations of different types of spyware, how it works, and what can be done to control it. Phishing and identity theft are covered in Chapter 5. This chapter includes a discussion of the structure of a phishing attack, its objectives, the economics of phishing, and, of course, how to reduce the incidents of successful phishing attacks. In Chapter 6, the focus is on spam with an emphasis on managing spam and evaluating the effectiveness of anti-spam systems. Chapter 7 takes a comprehensive look at technologies for securing users and information assets from the threats described throughout the guide, then describes best practices for controlling threats from Web and email content. Finally, Chapter 8 presents an overview of some of the implementation options available for controlling malware, spyware, spam, and phishing. The chapter concludes with best practices for a comprehensive approach to securing an internal network from external, Internet threats. The goal of this book is to provide you with the tools needed to leverage the benefits of the Internet with the most safety and reliability possible.

Organizational Responsibilities for Securing the IT Environment

Every organization that uses the Internet has some responsibility for protecting itself from the potential damage to information assets and harm to users. The main areas of concern are:

- Protection of information assets
- Efficient operations
- Non-threatening work environment
- Protecting against loss of controlled information
- Regulatory compliance

Some of these concerns are relevant to all organizations, such as protection of information assets, while others, such as regulatory compliance, will require varying levels of attention depending upon the specific nature of your business.

Protection of Information Assets


One of the most obvious concerns to business users of the Internet is protecting their information assets, in particular:

- Systems infrastructure
- Core applications and supporting software
- Data

Each type of asset has its own particular set of needs with regards to preserving the integrity of the asset.

Systems Infrastructure

Systems infrastructure consists of the servers, desktops, laptops, mobile devices, and networking hardware that comprise the organization's information systems hardware. The greatest threats to hardware are physical: damage from fire, flood, storm, and other natural disasters. For the most part, these threats are best dealt with through appropriate disaster recovery and business continuity planning. The main threats from the Internet target the software running on these machines and the data stored on them.

 For more information about disaster recovery and business continuity planning, see "The Disaster Recovery Guide" at <http://www.disaster-recovery-guide.com/> and the SANS Infosec Reading Room on disaster recovery at <http://www.sans.org/rr/whitepapers/recovery/>. For up-to-date news on severe weather warnings, see the United States National Oceanic and Atmospheric Administration (NOAA) National Warning Web site at <http://iwin.nws.noaa.gov/iwin/nationalwarnings.html>.

Core Applications and Supporting Software


Core applications are programs that support an organization's primary activities. They are as diverse as the businesses, governments, and other organizations that use them. For example:

- An enterprise resource planning (ERP) system controls a manufacture's inventory, production, and distribution operations
- Claims processing systems are used by insurers to store, review, and pay claims
- Sales force automation systems track contacts, proposals, contracts, and other material needed by sales teams
- Emergency responders use computer-aided dispatch systems to manage police, fire, and ambulance services

When core applications are down, the organization's ability to function is severely hampered. Large businesses and organizations can often support contingency servers with backup applications that can be quickly brought online in the event of a failure of the primary systems. For small and mid-sized businesses, fully functional contingency servers may be too costly to justify. In those cases, it is even more imperative to minimize the threats to those core applications.

Threats to Core Applications

Major threats to core applications are not necessarily directed against those applications themselves. There are certainly cases in which it would be worth an attacker's effort to target a large application with few implementations (for example, military and intelligence systems or electronic check and credit card clearing house applications). Although the number of targeted attacks against specific businesses, such as the major attack on the credit card transaction processor, CardSystems Solutions (<http://www.computerworld.com/databasetopics/data/story/0,10801,102631,00.html>), can be expected to increase, many current attacks target supporting software at the operating system (OS) and networking level. These attacks are not against the core application itself (for example, the ERP system), but they can effectively damage and disrupt the core application as if it had been a direct attack.


 For more information about a targeted attack to a core application, see “Hackers Using Targeted Attacks to Steal Firms' Customer Data” at <http://www.ccnmag.com/news.php?id=3633>.

Threats to Supporting Systems

Vulnerabilities in OSs and network software are widely known and, in some cases, easily detected. Hacking tools, such as vulnerability scanners, are readily available on the Internet, along with other tools, such as root kits, which allow even beginner hackers to take control of servers, desktops, and other devices connected to the Internet. In fact, most of the top vulnerabilities in Windows and UNIX environments, target OS and networking systems, including:

- Web servers and services
- Web browsers
- File sharing
- Mail clients
- BIND domain name service
- Simple Network Management Protocol (SNMP)

Databases, which are ubiquitous in enterprise-level applications, are so complex that they too suffer from security vulnerabilities. Widely used databases—including Microsoft SQL Server, Oracle, MySQL and PostgreSQL—have all had vulnerabilities that allow the databases to be compromised.

 For details about these and other top vulnerabilities, see the SANS' list of top vulnerabilities, “The Twenty Most Critical Internet Security Vulnerabilities” at <http://www.sans.org/top20/>.

When systems are connected to the Internet, they are exposed to threats that exploit vulnerabilities in core applications and supporting software. It is essential that systems administrators and IT managers keep applications and OSs patched to minimize the chance of a security breach.

In addition, IT staff responsible for infrastructure must understand that patching vulnerabilities and subscribing to OS update services is not enough to secure systems. First, a vulnerability may be discovered and exploited by an attacker before it is known to the application developers who could correct the flaw with a patch.

In addition, the process of testing and deploying patches can be time consuming, especially in large organizations. Change control procedures must be followed when applying patches just as they are when implementing any other change to software. These procedures typically include determining whether the patch is necessary (there is no need to apply a patch for a system service that is not used), whether the patch functions correctly in your environment (many systems administrators have applied OS service packs only to find they break critical applications), and how to deploy the patch (deployment must be prioritized so that the most critical of the vulnerable servers are patched first).

Addressing vulnerabilities is one piece of a broader security profile that must be in place to protect information assets. Access controls (to prevent disgruntled employees from damaging applications from within, for example) and Internet content management (such as antivirus, anti-spam, anti-spyware, and anti-phishing software) are also essential elements.

Beyond Vulnerability Management

Vulnerabilities in systems can only be exploited if a program or a person gains access to the vulnerable system. Some methods of entry are:

- Attaching viruses to email messages that are passed through firewalls directly to email servers
- Launching worms from ActiveX components running within an Internet Explorer (IE) browser
- Probing for open ports on firewalls and exploiting a known vulnerability (SQL Slammer spread this way)
- Using a virus or leaving a malicious program on a computer that then listens for instructions for a specific chat room or other two-way communication method

In an ideal world, these various forms of malware would never make it past network perimeter defenses. The real world is too complicated. Some ports have to be left open on firewalls, some applications depend on browser-based applets, and email systems are designed to move and distribute large quantities of content across secure boundaries. In addition to blocking some network traffic at the perimeter, you must also examine the content that pass through the perimeter to ensure it is not carrying a malicious payload.

Where Is the Network Perimeter?

The concept of a network perimeter is changing as we deploy ever more sophisticated distributed applications. Web services, for example, is a framework for building applications that take advantage of computing services on any accessible server that is using standard Internet protocols. Consider an online book seller who takes orders using a local order management system that calculates state- and country-specific sales tax using a third-party Web service. The system also determines shipping costs and delivery dates using a Web service provided by the book seller's shipping company. Where is the book seller's network perimeter?

The purchaser is certainly outside the perimeter; other than the occasional book orders, this person's system has no long-term, well-defined links to the seller's systems. Technically, Web services from third parties are outside the control of the book seller and therefore outside the network perimeter. Yet, these services are essential to the business operations of the book seller.

The Web services are outside the network perimeter are still within the sphere of business operations of the book seller. The concept of a network perimeter is still useful from a network engineering and maintenance perspective; from a business and operational perspective, the boundaries between inside and outside the organization are becoming more fluid (see Figure 1.1)

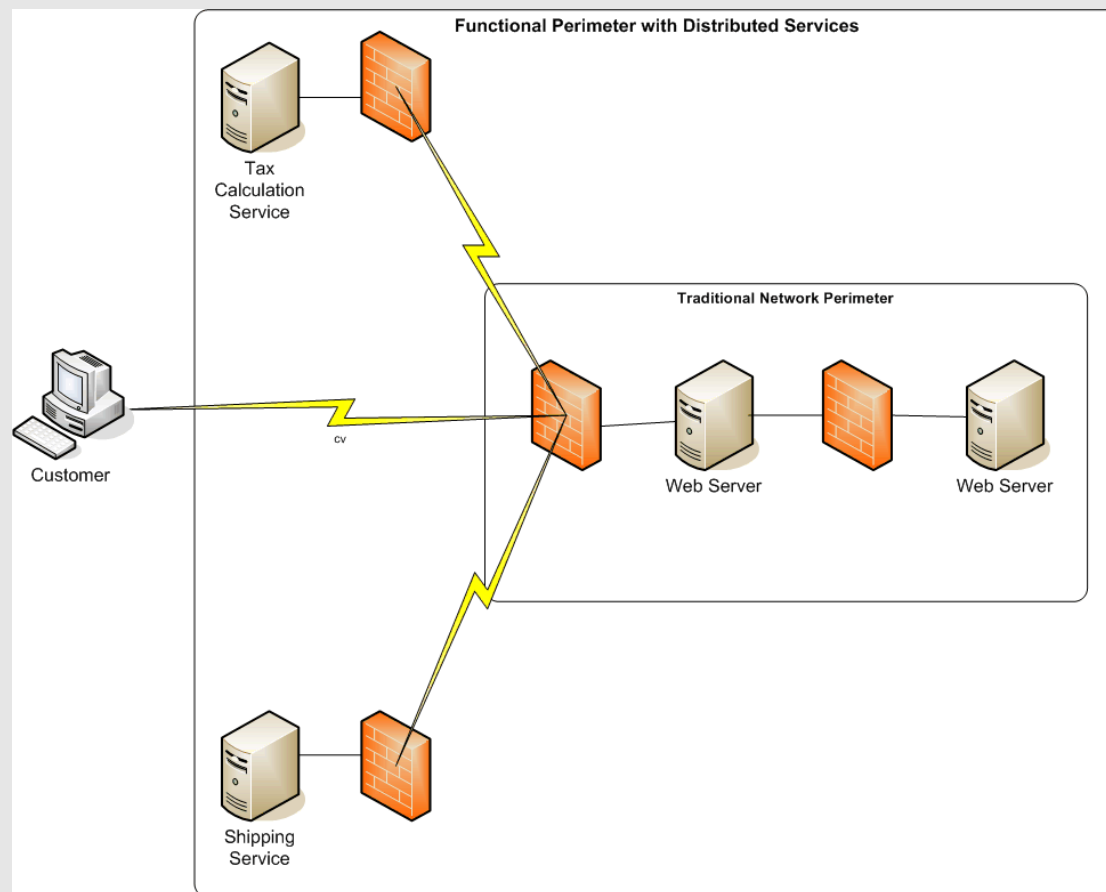


Figure 1.1: The concept of a network perimeter is not as clear cut as it was before the advent of Web services.

Protecting Data

In addition to protecting system infrastructures and core applications, organizations must protect the integrity and confidentiality of their data. Data lost as a result of systems failure or even malicious acts can be restored, albeit perhaps only partially, when sound backup and recovery practices are in place. Although small and midsized businesses and organizations may not have the level of failover recovery found in large enterprises, everyone should have effective backup and recovery procedures.

A more difficult problem to address is when a disgruntled employee or an outside attacker has tampered with data. In these cases, there may be no obvious, immediate signs of tampering. The tampering may not have a consistent pattern. In the worst case scenario, the only way to ensure the integrity of the data is to roll back to a point in time known to have reliable data and recreate all changes since that point.


A third type of data protection problem is extremely difficult to recover from—the loss of confidential data. Reports of stolen credit card and related identity information from major credit card processing services have made clear the level of this problem. In some cases, simply canceling credit cards and re-issuing new ones can solve the problem. In cases of identity theft, it can take individuals several months to resolve issues related to the theft.

Consequences of Identity Theft

With the right type of personal information, an identity thief can wreck havoc on someone's financial business. Some potential problems are:

- Tampering with existing accounts
- Unauthorized charges to credit cards
- Theft of funds from checking accounts
- Opening of credit cards using the stolen identity
- Incorrect entries on credit reports
- Applying for driver's licenses using the stolen identity
- Filing for bankruptcy using the stolen identity
- Making investments using the stolen identity

According to a 2003 survey by the United States Federal Trade Commission (FTC), consumers lost \$5 billion and businesses \$47.6 billion in out-of-pocket expenses due to identity theft. The vast majority of identity thefts have been the result of theft of information from offline sources, such as paper records and postal mail. Recently, though, sophisticated keylogging programs have been discovered that can collect password and logon information, disable the Windows firewall, and modify system files to prevent access to security vendors' sites so that victims cannot update their antivirus and other security software. Expect the proportion of identity thefts originating online to grow. For more information about identity theft, see the United State Federal Trade Commissions site on identity theft at <http://www.consumer.gov/idtheft/>.

 See the section "Regulatory Compliance" later in the chapter for more information about data confidentiality.

Losing confidential business data—such as sales lists, proposals, product designs, legal briefs, and strategy documents—can devastate a business. Imagine a small startup with a proprietary method for producing a low-cost, high-quality wireless device with applications in medicine, telecommunications, and public safety. Suppose an attacker compromises a server with information about the process and posts details on the Internet.

Protecting information assets, including hardware, software, and data, is a multifaceted problem. In the case of natural disasters, the best you can do is to implement plans that allow for rapid recovery when they do happen. In the case of malicious acts, you can utilize best practices in information security management to detect, isolate, and eliminate those threats.

Efficient Operations

The need for efficient operations has been around at least as long as free markets have existed. The advent of information technology (IT) has produced more tools with which to redesign and reengineer operations; it also brings with it the potential for adverse affects on efficiency.

Consider a few examples:

- Email has streamlined communications within and across organizations; however, this tool has also brought large volumes of spam, phishing attacks, and increased storage requirements due to poor document management practices (for example, attaching a large file and sending it to dozens of recipients instead of emailing a link to the file).
- Internet browsers are becoming the new client desktop as well as providing access to stores of information on the Internet. It is also the stepping stone to time-wasting sites such as online casinos.
- Computers that have been compromised by malware may contain programs that launch Denial of Service (DoS) attacks. In these attacks, compromised computers flood Web servers with bogus network traffic in an attempt to overwhelm the Web site. In addition to shutting down the target of the attack, the compromised machines are wasting bandwidth and computing cycles on the compromised network.

Making the most efficient use of computing and networking resources requires a combination of technology—such as antivirus applications, firewalls, and content filtering—and policies about appropriate use of information assets, such as using business systems for business operations only.

 In the past few years, there has been heated debate within the business community about IT efficiency and competitiveness. The discussion here does not delve into those broader issues and instead examines efficiency from a security management perspective. For more information about the broader debate, a good starting point is Nicholas Carr's "Does IT Matter?" at <http://www.nicholasgcarr.com/doesitmatter.html> and John Soat's response to Carr in "Book Review: IT's Transformational Effect: The Real Point" at <http://www.informationweek.com/story/showArticle.jhtml?articleID=51201618>.

Non-Threatening Work Environment

Employers are responsible for ensuring a non-threatening work environment to employees. It should be no surprise that the pervasive reach of the Internet has touched this aspect of organizations along with so many others.

In a non-threatening work environment, employees and contractors are not subject to unwelcome harassment or exposed to offensive material. Today, that harassment can take the form of

- Offensive emails sent by employees to others in the organization
- The display of inappropriate material in a colleague's Web browser
- Offensive postings on an intranet discussion group

Employers cannot control the behavior of employees, but they can utilize a combination of policies and technology to define and enforce legitimate boundaries of discourse in an organization.

Protecting Against Loss of Controlled Information

In spite of the information overload experienced in today's business world, information is an increasingly valuable asset. From a security perspective, there are three types of controlled information that warrant particular attention:

- Intellectual property
- Proprietary data
- Private information

For some organizations, the most valuable assets are not their buildings, equipment, and other tangible assets but their intellectual property. Proprietary processing methods, deep knowledge of their customer's business needs, and trade secrets about designs of products are valuable targets for attackers and thieves.

Proprietary data includes information about business strategies, customer proposals, budget details, and operational reporting. Like intellectual property, this information would be especially useful to competitors. In some cases, proprietary data is eventually made public—for example, a public company may announce a major contract with a new customer; if the information were obtained prior to public disclosure, it could be used to influence illegal investment decisions.

Private information is data about customers, employees, and others who provide personal information to an organization. Loss of private information can damage the reputation of a business and, in some cases, lead to eventual business failure. CardSystems Solutions, a credit card transaction clearing house, lost the business of Visa and American Express shortly after a major security breach occurred resulting in the exposure of 40 million identities.

Although protecting all three types of controlled information is the responsibility of any business, the need to protect such private information is coming more under the direction of government regulation.

Regulatory Compliance

Regulation from multiple governments concerning individual privacy and the integrity of public companies is a fact many IT organizations must address. The growing public awareness for the loss of privacy in the information age has occurred at roughly the same time as several high-profile investor fraud cases, such as Enron and WorldCom. Not surprisingly, regulations about privacy and business integrity have emerged in response.

The first thing to understand about these regulations is that there are too many from too many sources to mistake the new regulatory environment as a passing fad. For example, consider some of the governments that have passed regulations that impact IT operations.

Privacy regulations include:


- State of California, United States passed State Bill 1386, a law directing companies and government agencies to inform California residents of any unauthorized disclosure of personal information.
- The United States passed the Health Insurance Portability and Accountability Act (HIPAA), which dictates how personal medical information is used and shared. It also establishes security standards for healthcare providers to ensure private data is adequately protected.
- The Gramm-Leach-Bliley Act includes privacy and security regulations that govern how banks collect private information, how consumers may opt-out of information sharing programs, and directives for ensuring unauthorized changes are not made to private information.
- The Australian Federal Privacy Act defines principals for the collection and use of personal information of Australian citizens by businesses and governments.
- The European Union (EU) Directive 95/46/EC defines regulations about how personal data is collected, stored, shared, and updated.
- Canada has passed the Personal Information Protection and Electronic Documents Act (PIPEDA), defining standards for protecting personal data.

Business integrity regulations include:

- The Sarbanes-Oxley Act, which is perhaps the most well-known government regulation, was issued in response to recent corporate and investment scandals. This regulation broadly covers the governance of public companies. Its impact on IT operations focuses primarily on ensuring the integrity of financial reports, ensuring internal procedures are appropriate to guarantee data integrity, and reporting material changes in a company's operations.
- Title 21, Code of Federal Regulations, Part 11 (21 CFR Part 11) was created by the United States Food and Drug Administration (FDA) to define proper controls on the use of electronic signatures and electronic documents in the pharmaceuticals industry.
- Basel II was created by the Bank for International Settlements, an international organization designed to promote cooperation in international banking, to ensure banks properly manage and report credit risks.

Compliance and Security

Although government regulations are varied in context, they share a common requirement—protecting data from unauthorized access and changes. Rather than try to create policies, procedures, and technical solutions to individual regulations, a common security framework that addresses access controls, identity management, and content filtering, combined with other information security tools and techniques can solve the problem.

 For more information about best practices in IT compliance, see the Information Systems Audit and Control Association Web site at <http://www.isaca.org/>.

Summary of Organizational Responsibilities

Businesses, governments, and other organizations that use IT share many common responsibilities. First, they must protect information assets, including the hardware, software, and network infrastructure that constitutes their IT infrastructures. Just as importantly, the information housed within that environment must be protected. Second, they must ensure efficient operations for the benefit of stakeholders. Just as finances and personnel must be managed effectively, so too do information assets. Third, employers are responsible for preserving a non-threatening environment for employees. Widespread use of Internet technologies introduces new venues for harassing activities and events. Fourth, organizations must ensure the integrity of controlled information, including private customer data as well as proprietary information. Finally, businesses and some government agencies must comply with specific regulations governing privacy and business integrity.

Sound security management practices are essential to meeting these obligations. Although no single security practice or tool will address all threats, the proper combination of them will. The emphasis of this guide is on a set of Internet-based threats that will now be addressed.

Threats Posed by Internet Content

There are many threats to information security on the Internet, including attacks that render servers unavailable for legitimate operations, attacks that steal information by “eavesdropping” on other’s communications, and attacks that use the transmission of content as a mechanism for attacks. The most prevalent problems associated with Internet content are:

- Viruses, worms, Trojan horses, and other malware
- Spyware
- Phishing
- Spam
- Undesirable Web sites

Each of these presents distinct problems to the integrity of IT operations. Later chapters will delve into the details of these topics; the following sections provide a brief overview.

Viruses, Worms, Trojan Horses, and Other Malware

Malicious software is a well-known problem for which there are many solutions. Unfortunately, the problem is constantly evolving in response to defensive measures deployed by Internet users. One of the earliest forms of malicious software, the virus, is still around but has changed radically since the early days of PC adoption when viruses spread by copying themselves to floppy disks. In addition, new forms of malicious programs, including worms, Trojan horses, and blended threats endanger IT assets. The first step to understanding how to prevent damage from these malicious programs is to understand what they are and how they work.

Computer Viruses

Computer viruses are the best known type of malicious software and have been a problem since before the Internet’s widespread adoption. Computer viruses are programming code that attach themselves to other programs, perform destructive tasks such as deleting files, and make copies of themselves. A variation on the traditional virus is the macro virus.

Macro viruses attach themselves to data files, such as word processing documents and emails, rather than executable programs. These viruses take advantage of embedded programming systems, such as Visual Basic Script (VBScript), which allow documents to execute custom code.

Traditional Virus Detection

Traditional viruses and macro viruses are relatively easy to detect. Once a virus is discovered, antivirus developers can identify unique patterns within the program code that identify the virus, much like a fingerprint can uniquely identify a person. This identifying pattern is known as a signature. Antivirus software consists of an antivirus engine, which analyzes content such as files on a disk or data transmitted over a network, and compares that content to signatures in the tool's virus library. Antivirus vendors are constantly updating their library of signatures and making them available for download.

Mutating Viruses

Not to be outdone by antivirus developers, virus writers developed a radically new technique for creating viruses that do not have distinct signatures. Known as mutating viruses, these malicious programs copy themselves and continue to perform malicious tasks, but unlike traditional viruses, they do not make identical copies. Instead, they introduce useless instructions in the code and rearrange the order of instructions in every copy. In this manner, there is no distinctive pattern, like a fingerprint, to use for identification.

For example, a traditional virus might employ a sequence of instructions, known as the payload, such as:

- Read data from register 1 into memory location A
- Set interrupt 21
- Write "Infected by XYZ" in COMMAND.COM at location B.
- ...

A mutating virus will use a piece of code known as a mutation engine to add useless commands and rearrange the order of instructions not dependent on each other, creating something like the following:

- Write 1+1 to memory location B
- Read data from register 1 into memory location A
- Read data from register 1 into memory location B
- Set interrupt 21
- Write 0-0 to memory location C
- Write "Infected by XYZ" in COMMAND.COM at location B.
- ...


The result is that the net effect of the virus is the same, but it appears to be a different program from the one identified by the virus signature for the first set of instructions. The emergence of mutating viruses forced antivirus developers to change strategies and move from identifying viruses by their structure to identifying viruses by their behavior.

Worms

Worms are similar to viruses in that they are self-replicating, malicious programs. Unlike viruses, worms are fully functional programs; they do not need to attach themselves to an executable or document file to spread. Worms often exploit mail and file transfer services available to compromised machines to propagate.

- The Melissa worm demonstrated the effectiveness of mass emailing of malware in 1999 when it spread around the world in 2 days. The worm emailed itself to the top 50 or 100 (depending on the version) names in the infected user's Outlook Address book.
- The Sasser worm emerged in April 2004 and exploited open ports in firewalls to spread. Sasser used FTP to download malicious payload, then replicated using the same protocol.
- SQL Slammer exploited a vulnerability in the Microsoft database, SQL Server, and reached 75,000 victims in 10 minutes in January 2003. SQL Slammer is a simple worm that generates random Internet Protocol (IP) addresses and copies itself to those addresses. If the host with that address happens to be running an unpatched version of SQL Server, it becomes infected and begins to spread the infection.

As worms do not depend on other programs to act as carriers to spread, they can replicate rapidly. Viruses typically activate when the infected executable is run or infected document is open; worms are not slowed by waiting for a user to act on a program or file.

 For more information about Melissa, Sasser, SQL Slammer, and other worms and viruses, see the McAfee Virus Information Library at <http://vil.mcafeesecurity.com/vil/>.

Trojan Horses

Trojan horses are malicious programs that appear useful and benign. Unlike viruses and worms, Trojan horses do not replicate themselves. Instead, a programmer either includes a piece of malicious code in an otherwise useful program (for example, a disk utility) or disguises the program as something completely different, such as an electronic greeting card. In either case, once the file is downloaded and executed, the malicious code can execute. Trojan horses are used for a variety of tasks, including:

- Deleting files
- Corrupting data
- Using computing resources of the compromised machine, for example, to conduct spam or phishing mass emailings or to launch DoS attacks
- Capturing identity information, such as usernames and passwords, with keylogging programs
- Downloading adware

Viruses, worms, and Trojan horses are some of the oldest and most common forms of malware, but there are others as well.

Additional Forms of Malware

Malicious code, or malware, has evolved into multiple forms. Viruses and worms are well known because they spread so rapidly, but other forms of malware can be just as—and in some cases more—dangerous than either. Other forms of malware include:

- Keyloggers, which are programs that record a user's keystrokes and transmits them to a server on which usernames, passwords, credit card numbers, and other useful information can be captured.
- URL Injection programs change a user's URLs to make it appear that a user is browsing to a site, such as Amazon.com, from an affiliate site.
- Backdoor programs allow attackers to gain control of a computer without the knowledge or authorization of its user. These programs are sometimes used to launch DoS attacks and mass mailings of spam and phishing lures. Computers compromised by backdoors are often called zombies.
- Rootkits are the most dangerous form of malware. These are sets of programs used by attackers to maintain control of compromised computers and erase traces of their activities. As rootkits can modify system files and disguise changes, the only way to ensure a rootkit is removed is to format disk drives and reinstall the OS, applications, and data.
- Blended threats are programs that package several types of malware, often including worms, keyloggers, and even email, file transfer, and other communication programs.

Another form of malware that has grown into a serious threat to both corporate and home computer users is spyware.

Spyware

Spyware is a form of malicious software that captures information about users or otherwise takes control of system resources without the user's knowledge or consent. Spyware emerged in the late 1990s and had reached epidemic levels by 2004. In that year, a survey by America Online and National Cyber-Security Alliance found that 80 percent of the computers surveyed had some form of spyware on them, and there were an average of 93 pieces of spyware on each machine. The vast majority of users, 89 percent, did not know these programs were on their computers.

 For details on the AOL/NCSA spyware survey, see http://www.staysafeonline.info/news/safety_study_v04.pdf.

The most dangerous effects of spyware include:

- Loss of privacy and identity theft
- Decreased system performance
- System crashes as a result of poorly designed and written software
- Disabled security software, such as antivirus and firewalls, which leaves systems vulnerable to other malware infections

Less destructive but still problematic effects of spyware include:


- Displaying unwanted pop-up advertisements
- Changing search engine results
- Changing host files and other networking files causing users to unintentionally navigate to spyware-promoted sites

Spyware is spreads as Trojan horses, such as in Internet toolbar add-ons, and through the ability to manipulate browser vulnerabilities, especially in IE.

Phishing

Phishing is a relatively new method for age-old con schemes. The purpose of phishing is to trick users into disclosing valuable information such as credit card numbers, Social Security numbers, driver license numbers, and other personal information. In the most audacious scams, phishers attempt to convince victims to send money to a “charity” or “investment opportunity.”

A phishing scam begins with a lure, such as a request to update account information or verify account numbers. Financial institutions are commonly used in these phishing lures. Recent scams have targeted eBay, PayPal, Citizens Bank, Ameritrade, Bank of America, and Comcast.

 For examples of phishing lures, see the Anti-Phishing Working Group’s Phishing Archive Web site at http://www.antiphishing.org/phishing_archive.html.

Lures typically contain links to bogus sites where users are prompted for identifying information. Phishers will often use HTML and scripts from legitimate sites to make their bogus versions appear legitimate. Ways to identify a phishing scam include:

- Misspelled URLs, for example www.bankofamericas.com instead of www.bankofamerica.com
- Ungrammatical or unusual greetings, such as “Dear PayPal” or “Dear Value Customer”
- Random strings of characters in the message—these are used to trick spam and phishing filters that look for specific patterns of text in messages

Phishing lures are just one type spam that has created many problems for email administrators and users.

Spam

Spam is unsolicited, unwanted email. It's an obvious problem for users with inboxes that are cluttered with junk email, but that is just the tip of the iceberg. Spam creates several problems for email and network administrators:

- The need for additional storage space to store unwanted, unsolicited emails
- The threat of malware or phishing scams being carried by spam message
- Taxed bandwidth, especially when delivering messages to client's over slow network connections
- Lost time and productivity cleaning up spam

As with other threats, systems administrators have responded with spam management practices, such as quarantining suspected spam and blocking emails from known spamming sources. Although these methods reduce the eventual number of spam messages that make their way to user inboxes, these practices require additional technology that, in turn, requires maintenance and change control processes. Spam is not going away, so the best approach to managing it is to deploy a spam management system that accurately identifies spam while minimizing the demands on systems and network administrators.

Undesirable Web Sites

Businesses and other organizations commonly have policies about the proper use of IT resources, including Internet browsing. Outside of highly secure government and business networks, few employers are likely to be concerned with an employee occasionally checking the latest news or stock quote. Problems arise when employees use company resources for offensive or time-wasting activities.

One of the responsibilities of organizations in general is to provide a non-threatening work environment. This requirement includes ensuring that offensive material is not circulated or stored within the organization's network, email systems are not used to transmit offensive material, and employees do not download such material. Creating a threatening environment is only one of the problems posed by employee use of the Web.

Time-wasting Web sites are easy to find. At least hundreds of gambling sites, online games, political blogs, personal ad sites, music and video sites, and online shopping are readily available on the Web. In addition to reducing productivity, visiting such sites can expose the user's system to spyware, adware, Trojan horses, and other types of malware. File sharing services are a commonly used means of distributing malware. The major threats from Internet content—malware, spyware, phishing, spam, and the use of undesirable Web sites—must be addressed using a combination of technology and user training.

Countermeasures to Threats

The majority of this chapter has focused on high-level goals for managing information resources and the threats to those resources. The topic can become discouraging when one understands how many different ways there are to compromise systems, commit fraud, and steal personal and proprietary information. Fortunately, for each threat, there is a combination of technologies and procedures (including user education) that can significantly reduce these threats. The key technologies are:

- Content filtering
- URL filtering
- Antivirus software
- Anti-spyware
- Spam management

Much of this guide focuses on these technologies and how to apply them most effectively.

Content Filtering

Content filtering is a process in which network traffic is scanned for patterns that indicate dangerous or offensive content. Content targeted by these scans are blocked from transmission. Content filtering is used for email, Web browsing (HTTP), instant messaging, and chat and other text-based traffic.

Content filters use libraries of words and phrases that are typically found in target content. For example, phrases such as “live dealers,” “blackjack,” “payouts,” and “poker rooms” are indicative of an online casino. Similarly, organizations can specify terminology used in their business to identify proprietary information and prevent it from being transmitted outside the organization.

URL Filtering

URL filtering is similar to content filtering in that content is blocked; however, rather than focusing on a particular piece of content, URL filtering blocks all transmission to or from specific sites. This technique is especially useful for blocking established sites that are not relevant to business operations. For example, employers can prevent employees from checking personal email by blocking mail.yahoo.com, www.hotmail.com, and similar sites.

URL filtering software uses *white lists* and *black lists* to identify sites that have content allowed within an organization and those which do not, respectively. With rapid change in Web site registrations, it is difficult if not impossible to maintain a compressive list of disallowed sites. However, even with that limitation, URL filtering in combination with other technologies can help reduce the threats of malicious or inappropriate material entering or leaving a network.

Antivirus Software

Antivirus software is widely used on desktops; it can also be deployed at the enterprise level on email servers or other servers to scan content as it enters or leaves an organization's intranet. As noted earlier, antivirus programs use both static pattern-recognition techniques to identify known viruses and behavioral techniques to analyze the way a program operates to determine whether it is a virus.

Anti-Spyware

Anti-spyware systems operate in two ways—by filtering and blocking spyware before it can be installed or by detecting it and removing it after it has been installed. Like antivirus programs, anti-spyware systems detects tell-tale signs of spyware, such as tracking cookies used in Internet browsers, entries in Windows registries, or known spyware programs found on a system. It can also operate as a content filtering system, scanning incoming network traffic looking for similar indicators that a remote site is trying to install spyware on a local machine.

Spam Management

In some ways, managing spam should be easy; after all these are just email messages. They are not as complex as a mutating virus or a blended threat. It is a challenge, though, because distinguishing spam from legitimate email is not straightforward for several reasons. First, spammers know that businesses and Internet service providers (ISPs) use spam management tools to block their mass emailings. They carefully craft spam messages to minimize the use of tell-tale characteristics:

- Hide recipient's email address in the BCC: section of the header
- Improper or empty TO: address
- Large number of recipients in the TO:, CC:, or BCC: fields
- X-mailer field contains names of known spamming software
- Unusual use of HTML, such as an excessive number of comments to hide spam text
- Offer phrases such as "Click here now," "free," "earn money," and "limited time offer"

In addition, spam management software should not block legitimate email. Just because an email has a large number of recipients or a few questionable phrases does not necessarily mean it is spam. No anti-spam filter will ever be 100 percent accurate, but spam management vendors try to minimize the number of false positive hits (that is identifying a legitimate email as spam) while also minimizing the number of false negatives (that is identifying a spam message as legitimate email).

A combination of countermeasures is essential to ensuring the content flow in and out of an organization is properly managed to minimize threats to the organization, its employees, and its information assets.

Countermeasure Implementations


Countermeasures are implemented in three ways:

- Software applications
- Services
- Appliances

Software applications, as classified here, are countermeasures that execute within an intranet and perform antivirus, anti-spam, anti-phishing, and other content filtering operations. As with other software applications, they are installed and maintained by systems administrators.

Services are software applications that provide the same type of services but do so from an Internet accessible server outside an organization's intranet. These services may use similar software to that which is deployed within an intranet, but it is maintained and updated by the software vendor or a third party.

The third way to implement countermeasures is with an appliance. An appliance is a network device that is easily configured and performs countermeasure tasks with minimal maintenance or system administration. This ease of deployment, use, and administration is a major benefit of this method of delivering counter measures.

 More details of the advantages and disadvantages of these countermeasure implementations will be discussed in future chapters.

Summary

Businesses and organizations depend on the Internet to function, yet this critical piece of infrastructure comes with significant threats. Some of the most pressing for today's IT professionals are viruses, worms, and other malware; phishing scams; spam; and the inappropriate use of IT resources, especially Internet access. The remaining chapters will examine technologies and practices available today to help mitigate these threats and allow organizations to focus on the core business.

Chapter 2: Organizational Responsibilities for Protecting the Network from Internet Attacks

Any computer linked to the Internet is potentially subject to a variety of threats. These threats range from less-malicious port scans to disruptive and costly DoS attacks, virus infections, and theft of information. Damage can easily extend beyond a single compromised system.

SQL Slammer disrupted Internet operations around the globe because SQL Server administrators did not patch a known vulnerability. The problem was likely compounded by the fact that some users of Microsoft SQL Server Desktop Edition (MSDE), which is used for persistent storage in some desktop applications, may not have known they were running a version of SQL Server.

Clearly, protecting information assets begins with knowing which systems are in place and how they function; but organizational responsibilities extend to a wide array of challenges, including:

- Protecting employees
- Protecting information assets
- Protecting customers
- Protecting stakeholders

This chapter examines a variety of threats to organizations and describes how to use secure content technologies to manage those threats and their adverse consequences.

Protecting Employees

Employers want to protect employees from disruptions in their work processes. Businesses, government agencies, and other organizations cannot accomplish their objectives if their employees are left idle by downed systems and disrupted services. Although these are real and costly concerns, they are not the only threats to employees. Employees expect and have a right to work in non-threatening conditions. The Internet has created new means for perpetrating the old threats of harassment and hostile work environments. Besides attending to infrastructure protection, employers today are rightfully concerned with protecting their employees.

Harassment in the Workplace


Harassment in the workplace has gained attention in the past several decades, both in the United States and the European Union. Employers are more aware of the problem and many have established practices and procedures to protect employees. Harassment can be physical or psychological in form; it can be constituted by a single event or a series of incidents. In the context of this guide, the question addressed is, How can employers protect employees from harassment and hostile work environment conditions that make use of information technology (IT)?

Recent years have seen the emergence of harassment and hostile work environment incidents in which the use of company computer systems has played a central role. Some of the more well-known court cases in this area include:

- In *Smyth v. The Pillsbury Company*, an employee, Michael A. Smyth, was dismissed for threatening to kill members of the sales management staff in an email to his supervisor sent over the company's email system. (914 F. Supp. 97 (E.D. Pa. 1996))
- In *Bourke v. Nissan Motor Co*, two individuals responsible for establishing an email were dismissed for sending personal messages with inappropriate sexual humor. (YC 003979 Cal. Super. Ct., L.A. County 1991, affirmed by the Court of Appeals in 1993)
- In *May v. Teleservice Resources, Inc.* (WL 222906 (N.D. Tex. 1997)) a manager was demoted to an entry-level position after an email was intercepted that was critical of the company's cultural diversity program.

As technology is playing an integral role in harassment incidents, organizations are faced with questions of how to respond and balance competing interests, such as how to balance the right to be free from harassment with other employees' perceived privacy rights with regards to electronic transmissions. The American Management Association 2003 Survey on email rules, policies, and practices depicts the lack of a single approach to controlling the content of email and the way employees use email. The findings included:

- 52 percent of companies surveyed use some form of email monitoring and enforce email policies with disciplinary actions
- 22 percent of companies surveyed had terminated an employee for violating an email policy
- 75 percent of surveyed companies have formal policies, but fewer than 50 percent of those companies provide training on those policies
- 90 percent of email users receive personal email at work; for most, personal email is less than 10 percent of their total email
- 14 percent of companies surveyed have been ordered by a court or regulatory agency to produce employee email
- 5 percent of companies have been involved in a lawsuit triggered by email use





 These statistics are found in American Management Association, 2003 Email Rules, Policies and Practices Survey, available at <http://www.epolicyinstitute.com/survey/survey.pdf>.

Clearly, companies are responding to the organizational and legal demands brought on by the increased use and importance of email. Establishing formal policies and monitoring compliance are widely practiced. At the same time, not all companies using policies and monitoring train their users on these topics. Employees also continue to use email for personal use although, at least a significant number of them, have been trained on proper use policies. (An earlier survey cited in the *Monthly Labor Review* found both employers and employees comfortable with some personal use of email.)

Are these findings inconsistent? On the one hand, employers want to control how email is used. In addition, the court cases cited earlier demonstrate that employers are willing to resort to litigation to defend their ability to maintain that control. On the other hand, a large majority of employees continue to use email for personal use—action presumably not allowed by most policies. Rather than demonstrating an inconsistency in findings, these results reflect the complexity of the subject.

Email systems should be used only for business use; at the same time, employers recognize the occasional use of email for personal use is not detrimental to the organization. This behavior falls into the same category of making a personal call from a work phone. The goal of email monitoring should be to identify and isolate those instances of improper use that threaten the welfare of other employees or the organization as a whole.

Harassment is a problem that extends well beyond Internet and email use, as is another problem that has co-opted IT—offensive material in the workplace.

-  Workplace harassment is complex subject with both organizational and legal dimensions. For examples of the breadth and depth of the issue see:
-  Paul Buchanan's "The Evolving Understanding of Workplace Harassment and Employer Liability: Implications for Recent Supreme Court Decisions Under Title VII" at http://www.law.wfu.edu/prebuilt/LR_v34n1_Buchanan.pdf
-  Virttorio Di Martino "Preventing Violence and Harassment in the Workplace" <http://www.eurofound.eu.int/publications/files/EF02109EN.pdf>
-  Charles J. Muhl "Workplace Email and Internet Use" (*Monthly Labor Review*, February 2003) at http://www.findarticles.com/p/articles/mi_m1153/is_2_126/ai_100729675

Offensive Material in the Workplace

Closely related to the problem of harassment is offensive material in the workplace. Unlike harassment, which is intentional behavior to intimidate or harm another, offensive material is not necessarily brought into the workplace to harm. For example, two friends could share material which they find humorous and others find offensive. This occurred in the *Bourke v. Nissan Motor Co.* case cited earlier.

The Internet has not created a new problem in this area; it has created a new method for conducting the problematic behavior. It would seem that employers must now educate users on what should be obvious. Elizabeth du Fresne, labor and employment attorney, summed up the problem: “I don’t understand why [employees] think they can send racial and sexist jokes via email or download pornography at work. Why they don’t understand that the same rules of life apply when they get such material from the Internet and pass it on to others in the workplace, I don’t know” (Source: “Ethics in the Workplace—I” at <http://www.humanlinks.com/manres/ethics1.htm>).

The following list highlights best practices in preventing email from being used for harassment and the distribution of offensive material:

- Define clear policies governing email use—state the types of behaviors (for example, threats, use of inappropriate language, and so on) that are not tolerated
- Educate users about policies
- Filter email content to reduce the chance that harassing messages are successfully transmitted
- Inform users that email is monitored and messages may be blocked
- Although legal issues abound in this area, common sense about public and professional behavior can address many of the remaining issues

Protecting Information Assets

IT professionals are keenly aware of the need to keep operational information systems functioning. Some systems must be available 24-hours a day, 7 days a week with only rare downtimes for maintenance. Administrators trade pagers for “on-duty” times with each other so that someone is available to respond immediately if a system goes down. Business users and executive sponsors are dictating strict service level agreements (SLAs) not only for system availability but also for processing windows. Key operational reports and business intelligence reports must be ready at the start of business each day in some cases. In addition to typical maintenance and change management procedures, systems administrators must secure content that enters a corporate network to reduce the chances of several serious threats, including:


- Virus attacks
- Malicious use of computers
- Wasted bandwidth and storage

Preventing Virus and Other Malware Attacks

The threat of viruses, worms, Trojan Horses, and other types of malware is not news. The change in the severity of attacks and the length of time needed to recover, though, is a trend worth noting.

ICSA Labs, an independent research and certification testing center for information security conducts annual surveys on the prevalence and cost of viruses and other malware. The results of the most recent survey (2004) depict disturbing trends:

- 3.9 million virus encounters on 900,000 desktops, servers, and perimeter gateways in 2004; this statistic reflects 392 encounters per 1000 machines per month (an encounter is any time a virus is found and dealt with; it does not necessarily constitute an infection)
- 6 percent increase in virus disasters (25 or more PCs infected with the same virus at the same time) from the prior year; the increase was from 31 percent to 37 percent of respondents
- Recovery time increased 25 percent or 7 days from the prior year
- Cost of recovery, on average, was \$130,000
- 91 percent of respondents felt 2004 was somewhat or much worse than 2003, which had been to that point the worst year on record

 For complete details about the ICSA Labs survey, see “ICSA Labs 10th Annual Computer Virus Prevalence Survey” at http://www.trusecure.com/cgi-bin/ct_download.cgi?ESCD=w0206&file=VPS2004.pdf.

The signs of increasing threats of viruses and related malware span several different measures.

Infection Rate

The increase in the number of infections is in spite of improved deployment of antivirus software (see Figure 2.1). 98 percent of respondents reported use of full-time background antivirus software on desktops, up from 89 percent in 2003. The use of antivirus scanning within email gateways also rose, reaching 96 percent from a 94 percent rate in 2003.

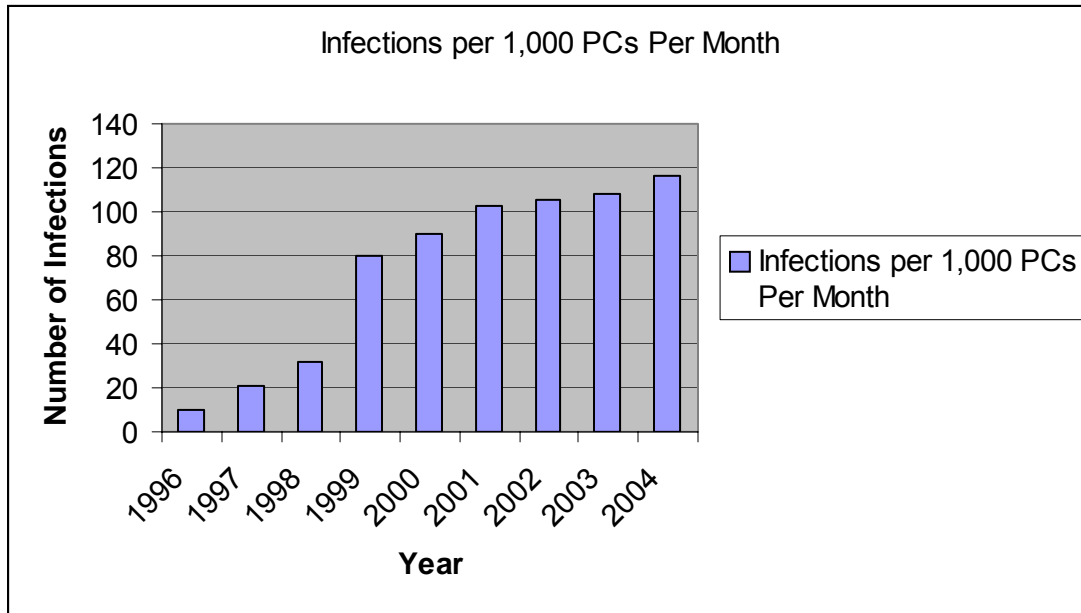


Figure 2.1: The number of infections per month continues to increase (Source: ICSA Labs 10th Annual Computer Virus Prevalence Survey).

Firewall and proxy servers continue to have relatively low protection rates of 50 percent and 60 percent, respectively. As the survey authors noted “perimeter protection provides a critical layer of protection and is a necessary component for a complete corporate virus protection strategy.”

Server Downtime

Servers were included in 95 percent of the virus disasters reported. The average downtime for infected servers was 23 hours with 80 percent reporting recovery within 20 hours (see Figure 2.2).

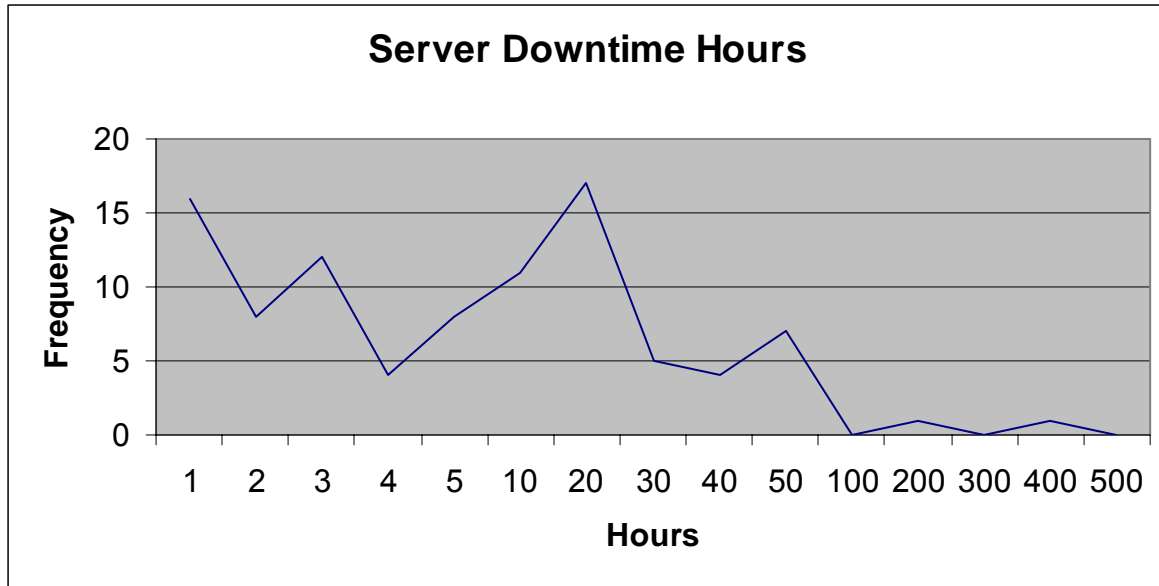


Figure 2.2: Server downtime is typically less than 20 hours per infection (Source: ICSA Labs 10th Annual Computer Virus Prevalence Survey).

The survey notes an upward trend in the time required to disinfect and recover from virus infections.

Recovery Time

ICSA Labs has found that companies are facing more virus incidents—requiring more personnel, resources, and time. The average recovery time in 2003 was 24 person days; in 2004, that figure rose to 31 days (see Figure 2.3).

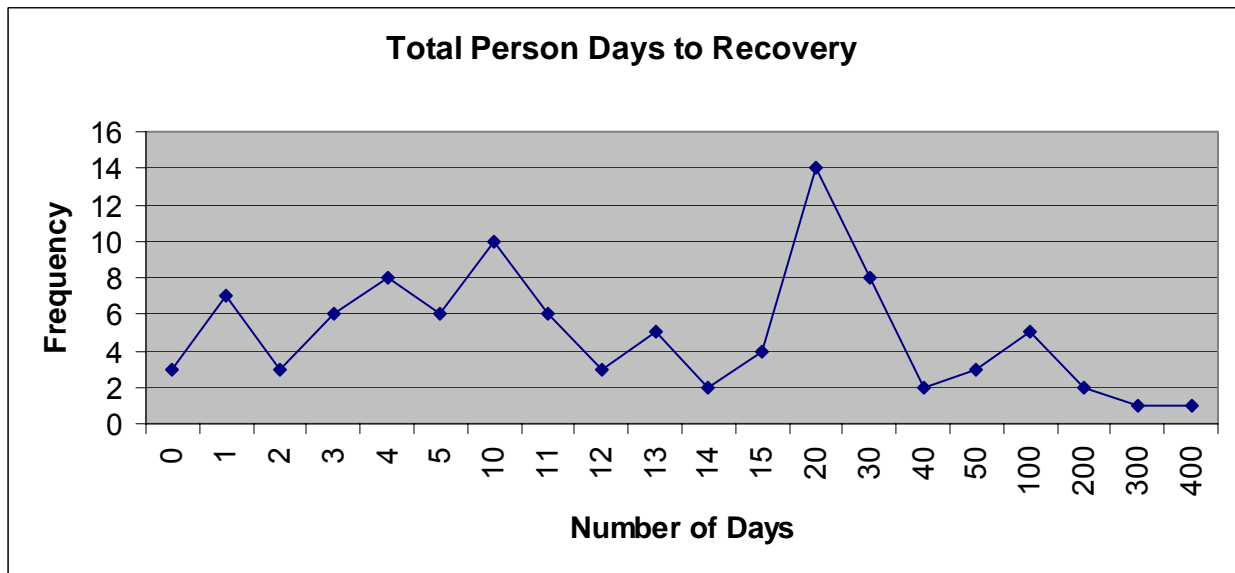


Figure 2.3: On average, 31 person days are now required to recover from an infection (Source: ICSA Labs 10th Annual Computer Virus Prevalence Survey).

Recovery Cost

As Figure 2.4 illustrates, the average cost to recover from a virus disaster in 2004 was \$130,000, up from an average of \$99,900 in 2003.

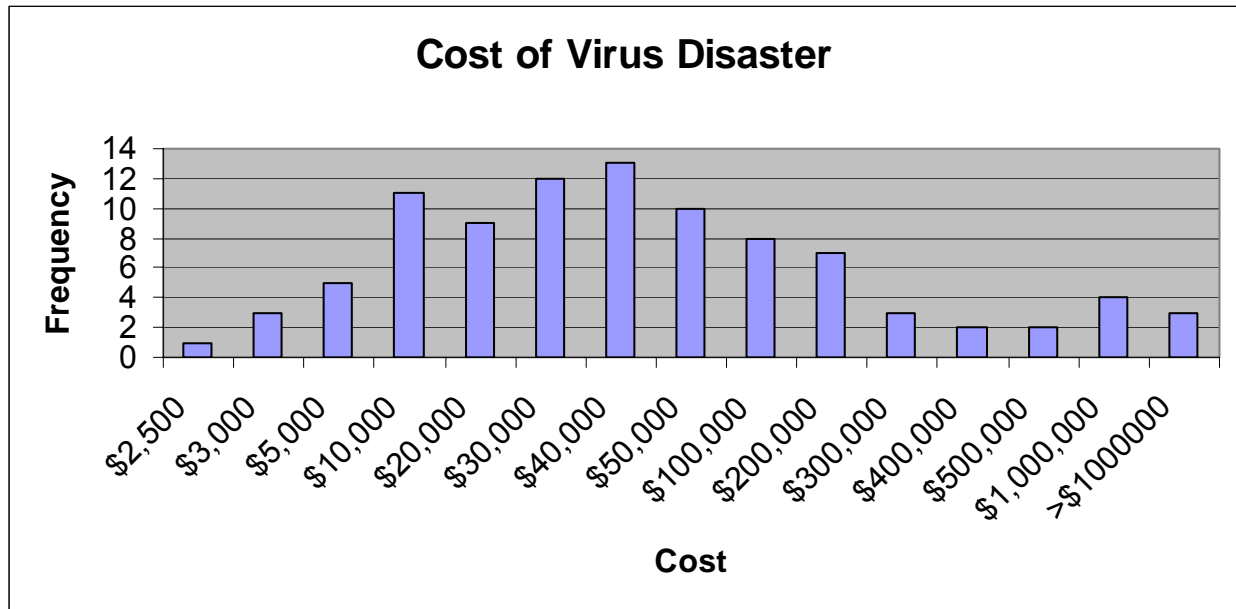


Figure 2.4: The average cost of virus recovery includes both a relatively low number of low cost and high cost recoveries (Source: ICSA Labs 10th Annual Computer Virus Prevalence Survey).

It should be noted, the survey authors estimate that respondents have historically underestimated the cost of recovery by a factor of 7 to 10 times.

Other Factors and Emerging Trends

In addition to the quantitative measures, the survey authors noted other disturbing trends, including:

- Virus writers and spammers are joining forces or at least adopting each others techniques.
- Viruses that successfully infect systems are using their own email engines for propagating themselves and distributing spam.
- Virus families are surviving longer in part due to rapid deployment of variants.

Although these additional factors make virus protection more difficult, the pattern is not new. Security professionals have long dealt with attackers and cyber-criminals who adapt to improved countermeasures. When early antivirus software successfully used signatures to detect and isolate viruses, virus writers tried encryption and then successfully created mutating viruses to thwart signature-based systems. Antivirus designers responded with a new class of antivirus detection methods based on behavioral analysis. Some emerging variations on traditional attacking and malware distributions include:

- Spammers and virus writers are sharing techniques, so it is not surprising to see antivirus and anti-spam technologies leveraging each other's strengths to counter the ever-evolving nature of virus threats.
- The use of malware is also spreading into other forms of cyber crime.
 - The Turkish authorities, for example, are investigating 16 suspects involved in credit card fraud who have links to the Zotob worm outbreak (Source: Jaikumar Vijayan "Zotob Case May Lead to Credit Card Arrests" <http://www.computerworld.com/securitytopics/security/story/0,10801,104388,00.html>)
 - Incidence of extortion of online gaming sites with threats of DoS attacks to indicate a linking of organized crime and attackers (Source: Jack M. Germain, "Global Extortion: Online Gambling and Organized Hacking", <http://www.technewsworld.com/story/33171.html>)
 - In 2000, a variation of the Love Letter worm was used to gain access to passwords at one Swiss and at least two United States banks (Source: Phil Williams, CERT Coordination Center, "Organized Crime and Cyber-crime: Implications for Business" <http://www.cert.org/archive/pdf/cybercrime-business.pdf>)
- Hacker wars, in which competing individuals or groups modify their own or each other's viruses and worms to gain control of compromised computers

There is no reason to doubt that these emerging trends and evolving threats will be dealt with as earlier threats have been addressed. At the same time, the widespread use of desktop antivirus software has not been enough to halt the increasing threat of viruses, malware, and the emerging threats from organized crime.

Malicious Use of Computers

Some malware is destructive; it may be designed to demonstrate an attacker's abilities or to disrupt services, but it does not have any other purpose. Other malware is designed with the clear intention of gaining control of infected machines and using those systems' computing and network resources for economic gain or to cause service disruptions beyond the compromised machine. Some of the primary reasons an attacker would want to gain control of a computer include:

- Send spam
- Commit click fraud
- Launch DoS attacks

The first two items clearly have economic gain as an objective; disruptive attacks such as DoS attacks, may or may not be economically motivated. Computers that have been compromised are known as zombie computers, or zombies for short.

Spamming with Zombies

Spammers are in business and have the same concerns as other businesses—minimizing costs and maximizing revenues. They can minimize their costs by using some one else's computers and bandwidth to distribute email. Fortunately for the spammers, that same technique helps them maximize revenues. By increasing the number of zombie computers under their control, they can increase the amount of spam they generate, which, in turn, leads to higher revenues (see Figure 2.5). There are more than economic incentives as well.

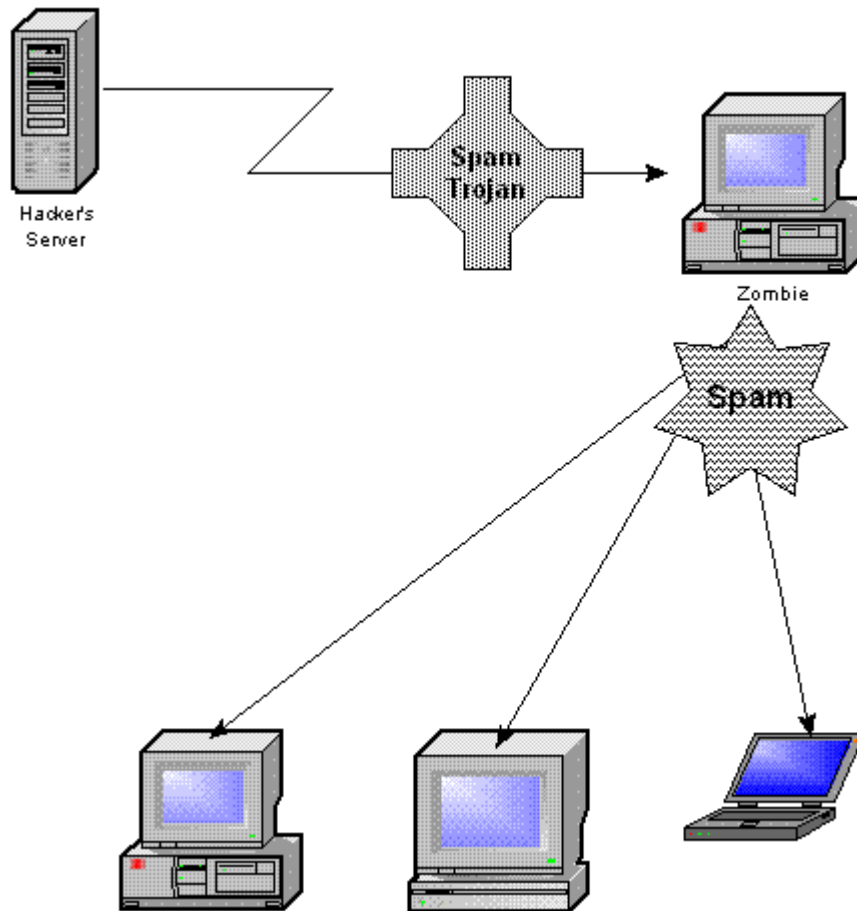


Figure 2.5: PCs are first compromised by a spam Trojan Horse, virus, or other malware; the infected machines then become email servers for spam distribution.

Spammers violate the law when they send mass mailings of unwanted, unsolicited emails. In the United States, the federal government and 38 states have passed anti-spam legislation. The European Union, Australia, and other countries have also passed laws regulating spam. Needless to say, spammers have a lot of incentive to mask their identities. Zombies can help with that.

For links to the text of anti-spam laws, see <http://www.spamlaws.com/>.

Using zombies also helps spammers avoid domain black lists such as DNS Providers Blacklist and real-time spam blacklist services (RBLs). As the mass mailings are spread out to a large number of computers on different domains, spammers can send large volumes of email without necessarily exceeding thresholds that would trigger their classification as a spammer.

If someone were to trace the origin of a piece of spam, it would lead to the compromised computer, not to the spammer. For example, Tom Spring, a writer for *PC World*, traced spam messages he received back to a number of legitimate businesses and organizations, including a financial planning company in New York, a medical services company in Kansas, a nursing home in Ontario, and a university in Beijing (Source: Tom Spring, "Slaying Spam-Spewing Zombie PCs" at <http://www.pcworld.com/news/article/0,aid,121381,00.asp>). Spammers are using techniques developed by attackers for distributed processing, such as distributed DoS (DDoS) attacks, to capture computing resources while hiding their identities.

Responding to Spam Zombies

The problem of zombies distributing spam has become so prevalent that email and network systems administrators are heeding calls to do more. Countermeasures are being deployed at two levels:

- Internet service provider (ISP) level
- Corporate network level

Some have argued that ISPs must do more to stop spam. As spammers develop more sophisticated techniques, it is becoming less likely that average users can keep up with the technical knowledge required to protect their computers. One anti-spam measure that has been advocated at the ISP level is blocking outbound traffic on port 25, the port used for email. There are advantages and disadvantages to this approach.


Normally, when a user sends an email, the message is sent via port 25 to an email server at an ISP or corporate email server. The email server then relays the message to the target recipient. This process works well for those of us sending moderate amounts of email and not trying to hide our email activity. Spam programs that infect zombie computers typically use their own email engine and bypass the ISP or corporate email server and send mail directly to the recipient. In response, many in the industry have argued that blocking port 25 can reduce the impact of zombies sending spam.

The advantages of this approach include:

- Most users that send large volumes of email via port 25 are not aware if they are spamming and would likely welcome the preventative measure
- Users with a legitimate need to use port 25 could be granted access if they agree to reasonable terms of use (for example, no mass mailings)
- Blocking port 25 shifts some of the responsibility for controlling spam closer to the point of origin and away from the recipient

There are disadvantages to blocking port 25, including the fact that completely blocking port 25 is probably not practical and additional administrative overhead will be incurred managing exceptions, and blocking port 25 at the perimeter will still allow spamming within the organizational network.

There is also the defeatist argument that spammers will just find away around port 25 blocks. If this problem follows the pattern of threat-countermeasure-response with revised threat that has characterized information security since its inception, you can certainly expect some workaround to this countermeasure. However, anti-spam researchers and practitioners will develop a countermeasure for the spammer's response. And so the cycle will continue.

 See Larry Seltzer's "Shutting Down the Internet Highway to Hell" at <http://www.eweek.com/article2/0,1759,1784276,00.asp> and Joe St. Sauver's "Spam Zombies and Inbound Flows to Compromised Customer Systems" <http://darkwing.uoregon.edu/~joe/zombies.pdf> for a discussion from both sides of the port 25 blocking debate.


Anti-spam measures can be deployed within the corporate network as well:

- **Personal firewalls**—Although contributing to the overall solution of reducing spam, each of these options serves a different purpose. Personal firewalls examine network traffic to and from a desktop or laptop. A personal firewall brings the same type of protection that traditionally has been found at the perimeter, including blocking ports and filtering by protocol. In addition, personal firewalls can provide ease of use features for non-technical users, such as pop-ups indicating a program is trying to access the Internet. If the user is not sure whether the program should be accessing the Internet, for example a spam Trojan Horse, its traffic can be blocked by the firewall.
- **Content filtering**—In cases in which spam reaches the corporate network, either as incoming email or as outgoing messages from a zombie, a content filtering appliance can block the traffic.
- **Desktop anti-spam software**—Desktop anti-spam software, like desktop antivirus software, is designed to protect a single computer. Desktop protection is especially important for mobile devices that are not always protected by network-based content filtering.

The combination of personal firewalls, content filtering, and desktop anti-spam software are one example of a defense-in-depth strategy that provides multiple types of countermeasures.

Committing Click Fraud with Zombies


Online advertising is an important source of revenues for Web publishers and search engines such as Google and Yahoo. Publishers typically charge advertisers for each click on an ad displayed by the publisher. This setup can create an incentive to fraudulently increase the number of clicks on an advertisement. Click fraud is committed in one of two ways: either a human clicks on ad links or a script programmatically simulates a click. As with spam delivery, fraudulent clicks are easier to hide if they are coming from a large number of compromised machines.

 The Times of India documented a case of organized click fraud in which a fraud group hired staff whose job was to click online ads for \$0.18 to \$0.25 per click; for more information, see <http://timesofindia.indiatimes.com/articleshow/msid-654822,curpg-1.cms>.

A variation on click fraud, known as impression fraud, is emerging. In this scheme, the defrauder temporarily terminates online ads with a search engine. He then proceeds to query the search engine in such a way that competitors' ads are displayed. The defrauder's script does not click the competitors' ads, so the competitors' click-through rates drop. Lower click-through rates can adversely impact the competitors' ad position or reduce the cost the defrauder must pay for a particular position.

Search engines and online advertisers are implementing anti-click fraud technologies, but this type of incident provides yet another example of how the infrastructure of e-commerce can be exploited for fraud. A key component of some of the schemes is the ability to distribute programs that commit the fraud using zombie computers, thus allowing the perpetrators to hide their identities.

Spamming and click fraud have clear economic motives. The economic benefit of exploiting compromised machines is not always so obvious.

 For more information about click fraud and related schemes, see Jessie Stricchiola's "Lost Per Click: Search Advertising & Click Fraud" at <http://searchenginewatch.com/searchday/article.php/3387581>.

DDoS Attacks

A DoS attack occurs when a malicious program sends excessive network traffic to a server in an effort to consume the server's key resources so that the server cannot respond to legitimate requests. A DDoS attack originates from multiple machines. A simple form of this type of attack is known as SYN Flooding.

For example, two devices on the Internet begin communication with a three-step process. The initiating device sends a synchronization packet (SYN) to the recipient. The recipient machine responds with a synchronization acknowledgment packet (SYN/ACK) if a service is available on the port or a reset packet (RST) if the port is closed. The initiating machine would then respond with an acknowledgement packet (ACK) after receiving the SYN/ACK packet, and communications would proceed (see Figure 2.6).

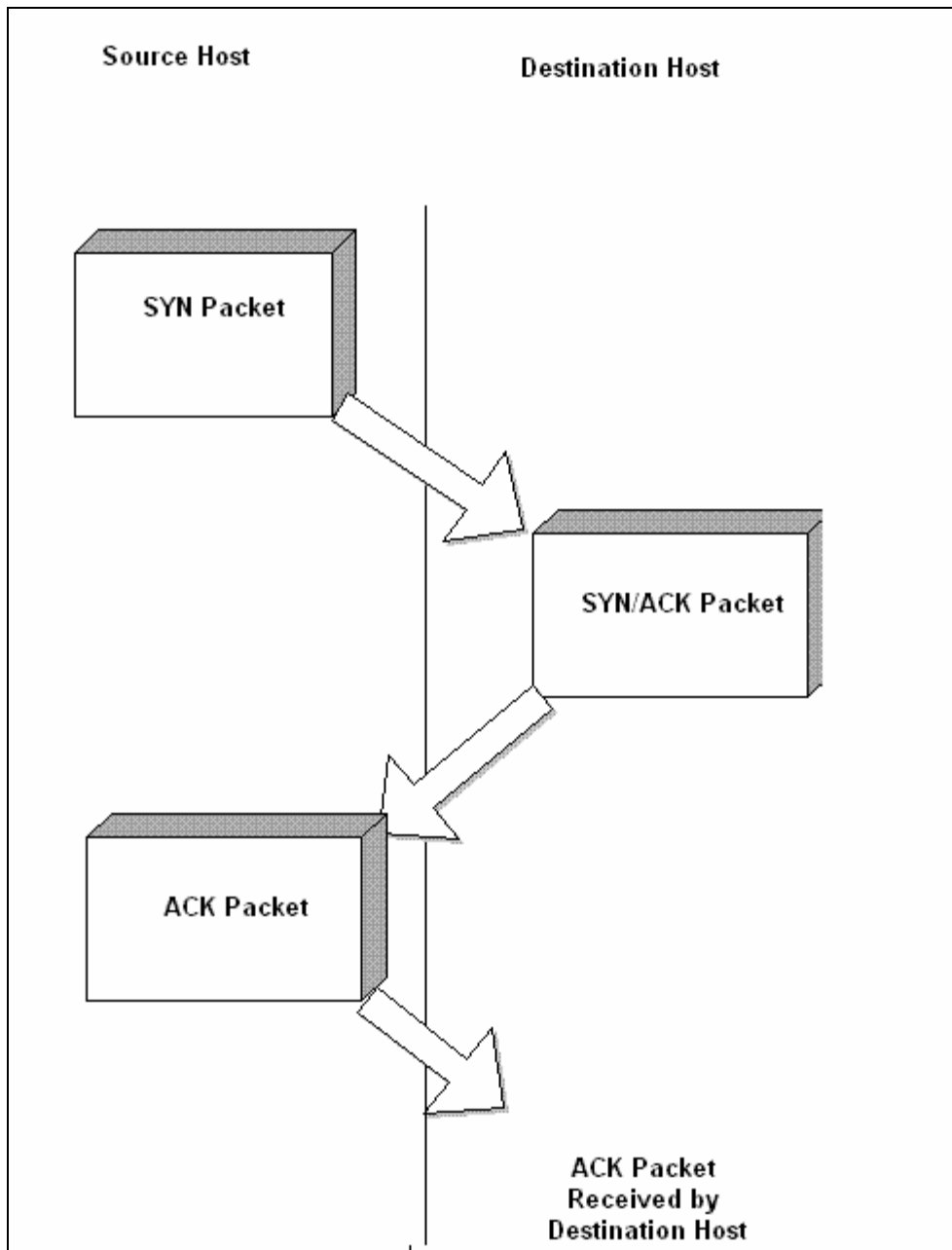


Figure 2.6: TCP communications begins with a three-way handshake.

When a SYN packet arrives, it contains the return address of the sender of the packet. The destination host uses this address to send the SYN/ACK packet. In addition to sending the packet, the destination host allocates a connection for its pool of network connections to this session. There is a limit to the number of connections maintained, and once those connections are used, new requests (SYN packets from other hosts) are refused. When sessions terminate, connections are freed and available for use by other hosts.

SYN Flooding exploits the fact that the destination host trusts the source host to provide its legitimate address. Instead, the attacking program uses a false, typically random, address. The destination host then sends the reply packet (SYN/ACK) to a false address and waits for a response (see Figure 2.7). While the destination waits for the ACK, a connection is tied up and will stay that way until the session is killed (the time to wait is system dependent, it may be a few minutes).

An attacker can easily initiate enough sessions to consume all the connections in the pool and continue to initiate new sessions as connections become available. Again, compromised computers can play a key role. A zombie with a DoS program can be the source of SYN packets that flood a destination host.

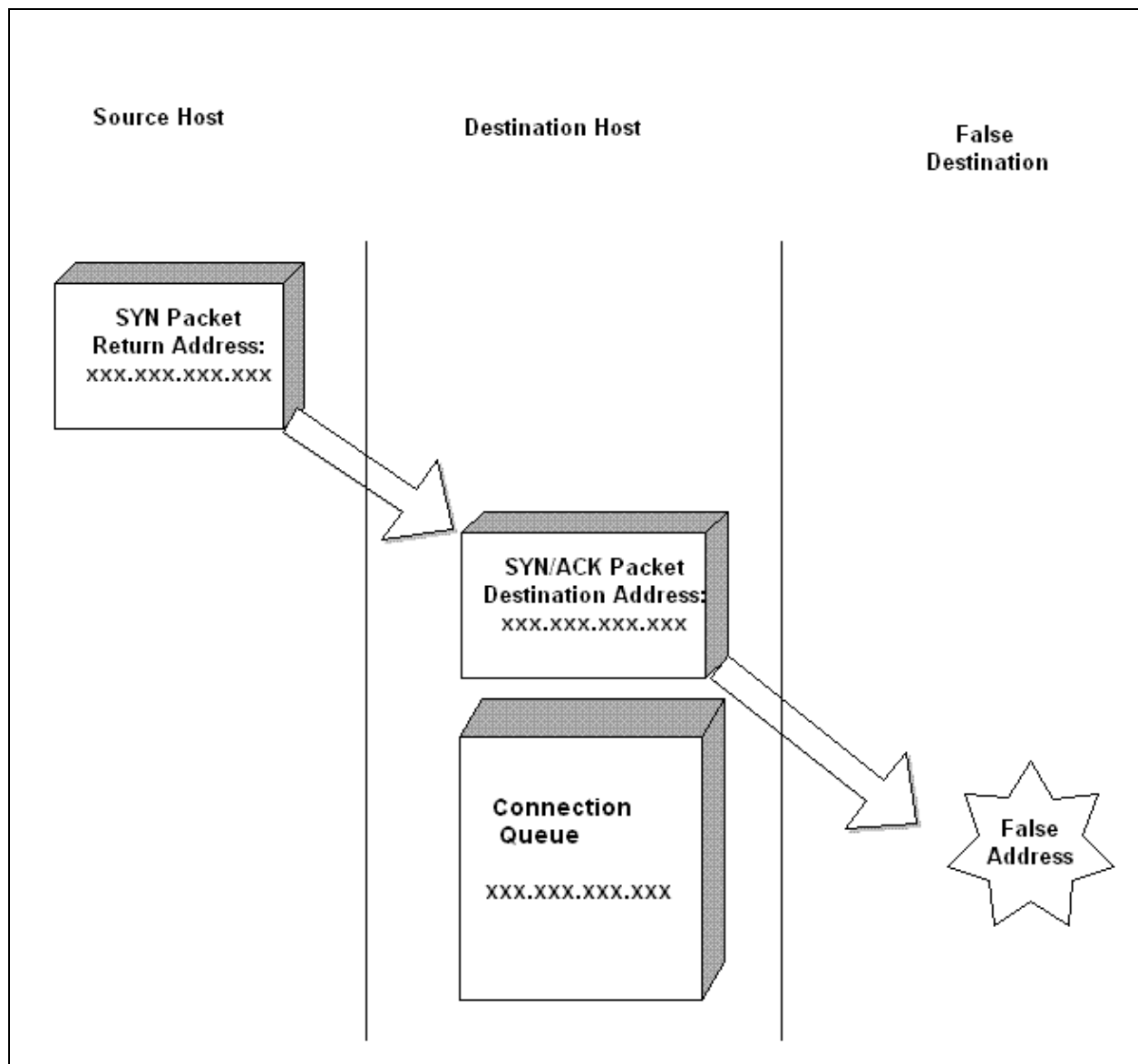


Figure 2.7: SYN Floods fill the connection queue while the destination host waits for acknowledgments that will never arrive.

DoS attacks can be prevented by denying perpetrators devices for launching those attacks and blocking traffic on corporate networks that could be used as part of DDoS attack. Ideally, a DDoS program would never make it to a corporate device. Network-based content scanning and desktop antivirus programs can detect and quarantine these malware programs along with others. In addition, firewalls can be configured to prevent outgoing traffic with false IP addresses.

 For more techniques for preventing DoS attacks, see the SANS Institute's step-by-step guide to preventing these types of attacks at <http://www.sans.org/dosstep/>.

The malicious use of compromised computers is a serious threat to computer and information security. Spamming, click fraud, and DoS attacks are three examples of how compromised computers can be exploited. Other types of attacks are likely to emerge that exploit vulnerable servers and desktops. Clearly, the need to scan incoming and outgoing content will not diminish and will likely increase in the future. In addition to the malicious use of computers, information assets are threatened with wasted resources.

Preventing Wasted Bandwidth and Storage

Another threat to corporate and organizational information assets is wasted bandwidth and storage. Small and midsized businesses may pay between \$400 and \$550 per month for a T-1 (1.544 megabits/second) Internet connection; while large organizations can pay between \$4000 and \$16,000 per month for a T-3 (43.232 megabits/second) line (Source: Infobahn, <http://www.infobahn.com/research-information.htm>). Malware and spammers can hit an organization coming and going, literally.

Incoming spam consumes bandwidth, and the longer the spam travels before it is detected, the more network bandwidth is wasted (see Figure 2.8). Ideally, ISPs would block spam near its point of origin. The next best option is to block spam as it reaches a corporate network. Network appliances positioned just inside of a firewall can meet this objective. The next point to catch spam is at the email server, but this method taxes the email server with additional work. The last chance to catch spam is at the desktop, but at that point, there is no savings on wasted bandwidth.

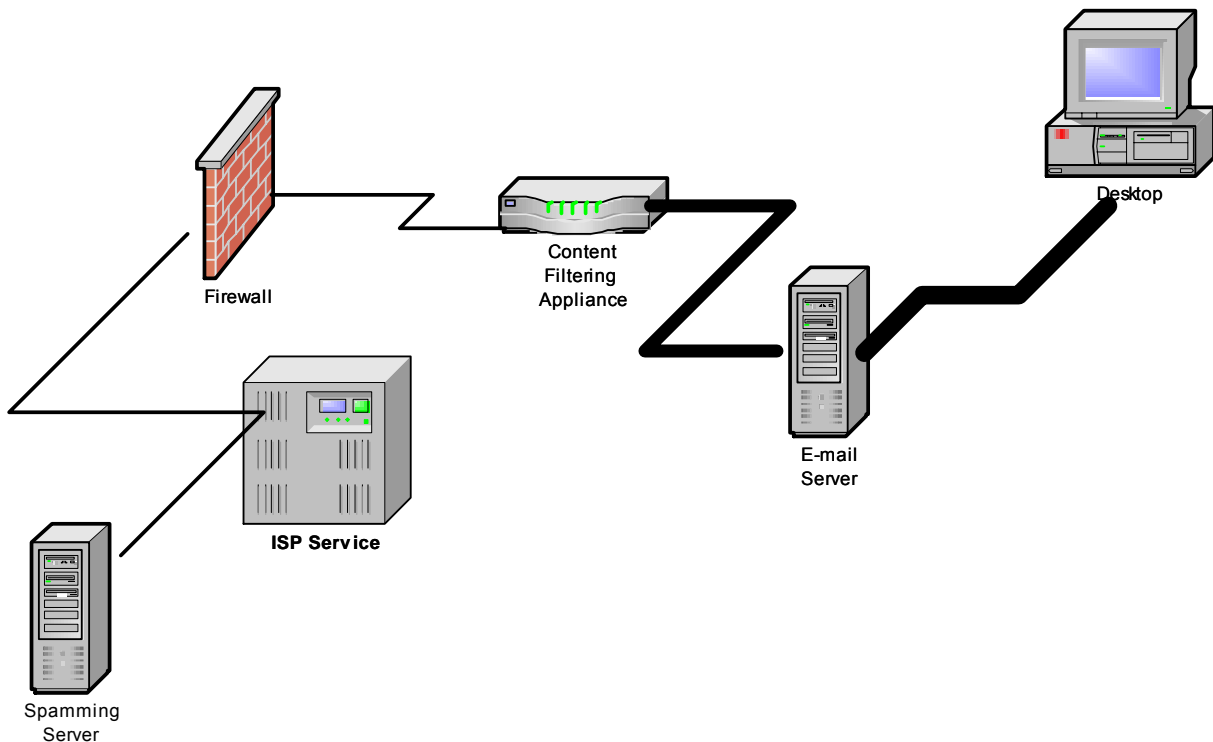


Figure 2.8: *The longer spam travels from its point of origin to the recipient, the more bandwidth is wasted.*

An internal device infected with a spam distribution program can generate substantial volumes of outgoing mail messages that consume bandwidth as well. Malware that implements DDoS attacks or click through fraud or spreads malware (such as SQL Slammer) also consumes bandwidth unnecessarily.

Spam and Unnecessary Storage

Like bandwidth, storage is a resource that can easily be consumed by spam. Unlike bandwidth, the waste of storage space continues through time. The problem may be compounded by email policies:

- Spam is stored in an email server and may be duplicated on the recipient's local drive.
- Depending on an organization's email retention policy, spam may be backed up onto long-term storage.
- In highly regulated or litigious industries, archives may be kept for long periods of time, sometimes in costly, off-site storage facilities.

Once a piece of email reaches an email server, the message becomes subject to email policies that must address a wide range of issues, such as auditing and compliance. Courts may demand that litigants produce emails during the discovery phase of a trial. These considerations can lead some organizations to back up and keep emails, including spam, for long periods of time. Again, as with wasted network bandwidth, it is best to detect and remove spam as early as possible in the transmission process. Ideally, spam will never reach the recipient's email server; that would save both the recipient and the company time, money, and management headaches.

Information assets must obviously be protected; what is less obvious is the ways in which malware and spam can harm those assets. In the early days of PC use and widespread Internet adoption, viruses would be created and distributed just to cause destruction. Today, economic incentives, from boarder-line legal spam to organized criminal activities, are exploiting vulnerabilities in technology and business processes and driving innovative threats. The trend will continue.

Protecting Customers

Organizations carry varying degrees of regulated responsibility in protecting customer information. Healthcare providers are under some of the strictest regulations to keep protected health information confidential. Financial institutions are also subject to regulation as well as market pressures to maintain adequate security for customer data.

No business wants to be the next company to make headlines with a security breach leaving customer information vulnerable. Some recent examples include:

- ChoicePoint, a credit information vendor, allowed unauthorized access to 145,000 customers' addresses and Social Security numbers
- 300,000 individuals may have had their personal information stolen from Lexis-Nexis
- UPS lost a box of CitiGroup's computer tapes with personal information on 3.9 million people
- CardSystem Solutions exposed 40 million credit card customers to fraud and identity theft; the breach cost the company two of its major customers, American Express and Visa

The key obligations of businesses and government agencies with personal information are twofold:

- Prevent the disclosure of confidential information
- Prevent identity theft that results from disclosed information

Accomplishing these objectives is not a trivial task in today's interconnected environment.

Preventing Disclosure of Confidential Information

What is confidential information? In the absence of regulations, such as the Health Insurance Portability and Accountability Act (HIPAA), that explicitly define confidential information, the question is difficult to answer. One customer may not mind a business or non-profit sharing their name and address with similar or related organizations while another might consider it a breach of his or her privacy. Opt-in programs are widely employed to allow customers to decide themselves how their information is used.

Opt-in and opt-out programs allow businesses to categorize their customers and use their data appropriately. In practical terms, though, there are a number of security issues that still present problems. Customer data is typically consolidated into a single database; it is not separated into opt-in and opt-out databases. Confidential information is intermingled with non-confidential information. Practically speaking, administrators need to protect a database according to the most sensitive information in it. In the case of opt-in and opt-out data in the same database, it all must be treated as confidential. This means:

- Access controls must be in place to limit users' ability to view and change the data
- Audit controls should be in place to track all changes
- Measures must be in place to preserve the confidentiality of data in transit
- Database servers and application servers with access to data must be configured securely

These measures, in addition to other security measures such as vulnerability testing and perimeter defenses, are essential to preventing identity theft.

Protecting Customers from Identity Theft

Identity theft is difficult to prevent when confidential information moves beyond the managed network. For example, when a customer uses an ATM at the bank, the customer is reasonably assured that no malicious programs are running on the machine. What about a public access machine in the local library or in a hotel business center? Chances are good that some type of spyware, keylogger, or video frame grabber may be running on the machine.

Regardless of the defenses and countermeasures put in place in a controlled network, once the data leaves that network, it is subject to disclosure. Security procedures and techniques deployed within a corporate network, including content scanning for malware, can reduce the chance of identity theft and disclosure of confidential information. However, such measures are not enough.

Users—whether employees, contractors, business partners, or customers—should be made aware of vulnerabilities in Internet-based applications. This argument is not in favor of excessive detail about technical vulnerabilities (for example, the database listener module used by Java Database Connectivity—JDBC—is vulnerable to a buffer overflow attack) but for education about best practices such as:

- Clearing buffer caches
- Running antivirus and anti-spyware programs
- Avoiding Internet utilities that could contain Trojan Horses
- Not using public access devices to transmit confidential information such as bank account numbers
- Being conscious of social engineering ploys to disclose confidential information

Identity theft is a problem exacerbated by poor information security but it is not caused by poor security. Preventing identity theft will require a combination of technical and non-technical countermeasures.

Protecting Stakeholders

The last area that organizations must consider with regard to their responsibilities in information management is organizations' stakeholders. Stakeholders can be owners, staff, business partners, and others who contribute to and stand to benefit from the organization's activities. Although there are many ways in which to protect stakeholders interests, the following list highlights three common examples:

- Preventing non-business Web activity
- Complying with regulation
- Avoiding the cost of recovering from security breaches

This list is certainly not comprehensive but is sufficient to demonstrate the scope of the issue facing businesses, government agencies, and other organizations.

Preventing Non-Business Web Activity

Non-business Web activity, beyond a reasonable amount, is a productivity drain. Like the occasional personal phone call at work, a quick check of the weather or last night's sports scores is commonplace and expected in the workplace. Non-business activity becomes a problem when it extends beyond the bounds of reasonable practices to include:

- Protracted periods of browsing. No one can reasonably expect to come into work and sit at their desk reading the paper for an hour to be tolerated as reasonable behavior. Similarly, an organization's can expect their staff to refrain from perusing an assortment of news sources or spending hours at an online casino.
- Downloading large files for personal use. Businesses and large organizations have large-capacity Internet connections to meet their needs. This connection should not have to be dedicated to downloading music files, video clips, and other forms of entertainment for employees who have slower connections at home.
- Bringing offensive material into the organization. This topic was addressed earlier in the section on protecting employees from hostile working conditions.
- Using corporate assets without permission. For example, downloading a peer-to-peer file-sharing client onto a company computer for sharing non-work related files.

The Internet is a powerful tool for getting work done efficiently but it can also be a distraction. The long-term interests of stakeholders dictate that some policies and procedures must be in place to balance the benefits of the technology with the unwanted costs of non-business-related activity.

Complying with Regulation

Compliance is a hot topic. Scandals of the last decade have demonstrated that not all businesses conduct themselves according to implicitly agreed upon manners and the rules must be explicitly stated. The Sarbanes-Oxley Act, Graham-Leach-Bliley Act, HIPAA, and a host of other regulations now dictate more requirements than previously imposed on organizations. Many of these regulations have implications for the way information systems are deployed, utilized, and managed.

The details of these and other well-known regulations vary, but several basic principals apply across regulations:

- Information must be accurate
- Changes to information must be done according to established procedures
- In many cases, information is confidential and must be treated as such
- Organizations must be able to demonstrate compliance with these regulations

From the perspective of information security, data must be protected from unauthorized tampering (for example, viruses and other malware should not be able to change details of a person's credit history), unauthorized disclosure (for example, millions of credit card holders should not have their information transmitted to an unauthorized destination), and audit logs must be tamperproof (that is, an attacker should not be able to install a root kit to hide the changes he or she made to a system).

Avoiding the Cost of Security Breaches

Recovering from a virus infection or cleaning up Trojan Horses that are generating spam can be costly. These efforts take the time of IT professionals, many of whom are already working at capacity on production operations or new development projects. Even more difficult to quantify is the damage to corporate image and brand when a breach is well publicized. The case of CardSystem Solutions, a credit card transaction processor, is one of the most telling examples of just how much damage a single breach can cost.

Stakeholders have wide and varied interests. From protecting the production capacity of an organization to complying with government regulation to protecting corporate image in the marketplace, stakeholders' interest encompasses the interest of the organizations in general.

Summary

Corporate networks connected to the Internet are capable of delivering great value to organizations. They are also subject to numerous, constantly evolving threats. Organizations have responsibilities to their employees, customers, stakeholders, and themselves to protect their information assets. Fortunately, the tools are available to meet these challenges, as we'll explore throughout the rest of this guide.

Chapter 3: Viruses, Worms, and Blended Threats

Viruses, worms, and blended threats are all examples of malicious code collectively known as *malware*. Malicious programs have existed since (at least) the early 1980s with the advent of personal computing. Since then, viruses, worms, and related programs have evolved rapidly, often in response to new opportunities presented by advances in networking or application features. Other times, virus writers are forced to adapt to avoid detection by ever more sophisticated detection techniques and countermeasures.

This chapter examines the history of some of the most common types of malware: viruses and worms. Both types of malware can succeed only when they can replicate and spread without detection. Much of the effort needed to deploy a virus goes into disguising itself to avoid detection. Worms similarly try to hide themselves, but variants exist that have opted to remain in the open and propagate rapidly and in large numbers to survive and spread. There is no single programming technique or stealth strategy deployed by these prominent forms of malware; rather, like their biological namesakes, they have adopted and survived using a variety of techniques.

In addition to using multiple methods to ensure their survival, malicious programs have evolved to become more than a single virus or worm and are now often a collection of multiple pieces of malware operating together to compromise computing platforms. These multiple-threat programs, known as blended threats, are common today. This trend is driven, in part, by emerging uses of malware. The motives for writing and deploying malware have also changed over the past two decades as the economic dimension of malware has emerged to provide one of the most powerful incentives for creating malicious code.

Evolution of Viruses and Countermeasures

Technically, a virus is a program that can replicate itself using another program running on a host. These programs are usually also malicious, carrying out destructive acts, including:

- Deleting files
- Corrupting files
- Destabilizing OS runtime environments
- Vandalizing platforms (for example, displaying messages about the virus compromise)
- Spreading spam
- Capturing personally identifying information

This list is by no means an exhaustive inventory of malicious acts that viruses can perpetrate, but it is representative of two notable facets of malware:

- The potential loss due to a virus infection ranges from a minor annoyance, as is the case with simple vandalism, to significant loss or disruption in operations, as in the case of stealing personal information.
- The incentives of virus writing were once limited to proving one's coding prowess and earning the respect of other attackers, but now include economic gain from identity theft and other fraudulent acts.

To understand both the technical evolution of viruses and the changing social and economic incentives for their development, it is worth starting at the beginning with computer viruses.

The Early Days of Viruses


Early viruses were, by today's standards, simplistic and benign. They would infect the boot sector of a floppy disk and cause minor damage, such as occupying small amounts of storage space and displaying annoying messages. The basic goal was to get the virus loaded into memory when the legitimate program was executed and then to infect other programs.

For example, the Elk Cloner virus, written in 1982 by a 15-year-old high school student is generally regarded as the first computer virus outside a research institution (where predecessors of viruses and worms were experimented with since the 1960s). The virus spread by sharing floppy disks. Once activated, the virus would print a poem about "Cloner" affecting the user's machine.

The Brain virus, the first known virus to infect IBM-compatible computers, was designed to prevent the pirating of the author's software. Infected machines would display the message:

"Welcome to the Dungeon (c) 1986 Basit * Amjad (pvt) Ltd. BRAIN COMPUTER SERVICES 730 NIZAM BLOCK ALLAMA IQBAL TOWN LAHORE-PAKISTAN PHONE: 430791,443248,280530. Beware of this VIRUS.... Contact us for vaccination..."

The purported intent was to prevent users from spreading illegal copies of their program.

 For more information about the Brain virus and the author's reaction to the unintended notoriety, see <http://www.brain.net.pk/aboutus.htm>.

These viruses had few options for spreading. Floppy disk sharing is a well-known method, but bulletin boards were also used. Games and other programs shared on bulletin boards could quickly spread viruses depending on the popularity of the infected program.

Elk Cloner and Brain are examples of parasitic viruses; they used hosts to replicate but did not damage the host systems. Things turned decidedly ugly in the 1990s with the 10,000 DOS-based viruses in existence by 1996, including some truly malevolent viruses.

Beyond Annoyance: The Proliferation of Destructive Viruses

The late 1990s witnessed the wide spread of destructive viruses that propagated more rapidly than early viruses.

Wiping Out Hard Drives—CIH Virus

A destructive virus known as CIH spread through Asia in 1998 after a hard-disk manufacturer shipped a firmware update infected with the virus. The virus overwrites the first 1024 kilobytes of the boot drive, often damaging the partition table; in some PCs, another part of the virus could overwrite the BIOS. The virus took advantage of the fact that the Portable Executable File Format used in the Windows 9x OS often left files with blank space in executable files. The malicious code was essentially hidden in this empty area of the file so as not to change the file size.

Coding a program such as CIH requires in-depth knowledge of OSs, file systems, and systems programming, as well as the ability to code in low-level programming languages, such as assembly language. When viruses were written in assembly language, there was a natural barrier to entry; not every virus programmer want-to-be could design and implement malicious code. Unfortunately, like other technical advances, evolutionary features of desktop applications could be used for both productive and destructive ends.

Virus Programming for the Masses 1: Macro Viruses

Until the late 1990s, most viruses spread by attaching themselves to executable programs. By that time, however, advances in desktop applications were providing new methods of spreading (technically, these are known as *vectors*). Visual Basic for Applications (VBA) is a simple programming language embedded in desktop applications such as Microsoft Word and Microsoft Excel to allow users to program macros. Although the intended use was to automate repetitive tasks and improve productivity, virus writers discovered that the lax security features, ease of programming, and rich set of commands made VBA an ideal vector for the next generation of viruses.

Some macro viruses are trivially simple. The Word macro virus, BREEDER, for example, checks a document or template as it is opened, and if it does not have an AutoOpen module, it replicates itself to the file. VONANBUG is another, more malicious, macro virus. When this one executes, it disables Word menu functions, including the ability to view macros. If the user were to press Alt-F11 to view the macro, it would delete files with extensions that indicated graphic, sound, archive, and Web files. In addition to infecting open document files, it infects the template file, normal.dot.

Macro viruses can infect any desktop application that supports macros, not just Microsoft Word. Several macro viruses infect Microsoft Excel with trigger infections that use Auto_Open routines. The TRASHER.D macro virus, for instance, attempts to delete files in antivirus program directories, and if the day of the month is the 1st, 9th, or 23rd, the program will modify the autoexec.bat file to format the hard drive.

Macro viruses and related macro worms are easy to develop, but application developers have included some security features to limit the risk of these threats. Microsoft applications, for example, allow users to specify a trust level for running macros (see Figure 3.1). In some cases, a high trust level will prevent legitimate macros from running. By lowering the trust threshold to medium, a user can choose which macros to run and which to terminate. This setup is a reasonable balance for some users; others may be unfamiliar with the nature of macros and allow any macro to run when prompted for a decision about executing the embedded code. The flexibility of this type of security feature is certainly welcome but does not eliminate the need to scan all documents to ensure malicious code is not reaching the desktop.

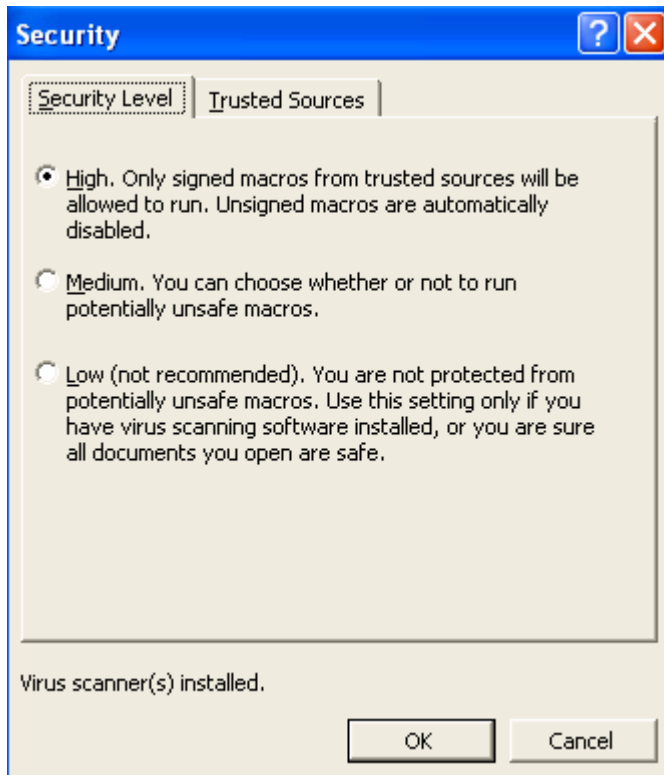


Figure 3.1: Desktop applications, such as Microsoft Word, are now employing more security controls to limit the risk of macro viruses.

Macro viruses still require some basic programming skill and understanding of the underlying OS. The bar for creating viruses has been lowered even further with the advent of virus generators.

Virus Programming for the Masses 2: Virus Generators

Virus generators are programs that create viruses according to a user's specification. These tools, which are freely available on the Internet, enable virtually anyone with the skill to download a file and work with a graphical user interface (GUI) to create a virus. At first glance, the availability of this software is frightening. Fortunately, the code created by these generators is easily detected by antivirus software. As long as there is widespread use of antivirus programs, the threat from "script kiddies" armed with virus generators is controllable.

Computer viruses have evolved to take on a variety of forms from handcrafted assembly language routines to generated viruses. Virus writers have exploited new technologies, such as application macros, to spread their malware; at the same time, antivirus developers have created more effective countermeasures to keep pace with the malware writers. As the McAfee World Virus Map in Figure 3.2 shows, virus infections are a global phenomenon—and not one that is likely to disappear soon.

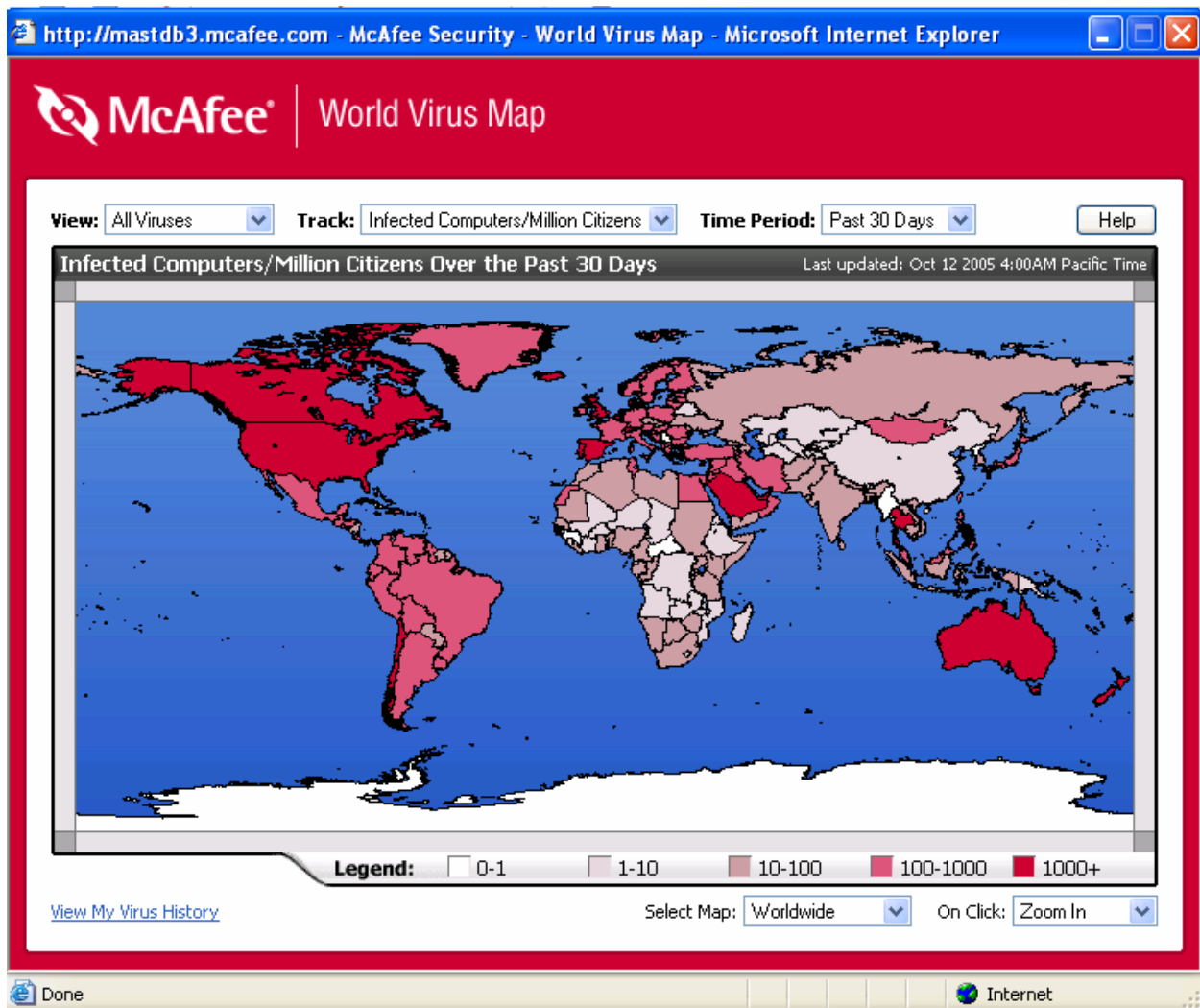


Figure 3.2: Computer viruses are a global problem; North America, Europe, and Australia are most infected.

Evolving Threats, Evolving Countermeasures

Computer security is often described as a cat and mouse game. Hackers, virus writers, and cyber-vandals exploit vulnerabilities in systems; developers and systems administrators fix flaws in software, change configurations, or deploy safeguards such as anti-virus software. The attackers go back to the drawing board (or keyboard in this case) to find other vulnerabilities that are exploited, causing the security professionals to respond...and so it goes. Although this wording presents a simplistic picture of computer security (security professionals do much more than wait around to respond to threats), it provides a useful framework for understanding the evolution of antivirus countermeasures.

Detecting Viruses

Viruses must remain undetected long enough to propagate in order to succeed in whatever goal they have been designed for. In the early days of DOS-based PCs unprotected with antivirus software, the task was relatively straightforward. If a virus could infect a file or the boot sector on the disk, not take up too much room in RAM, and not change the size of an infected file, chances were good that the virus would replicate. The widespread use of antivirus software changed the playing field.

Early Signature-Based Detection

Early antivirus programs were essentially string matching programs. When a virus was discovered, it was analyzed to determine the sequence of machine instructions that constituted the program. Antivirus developers would find a unique sequence of instructions that were always present in the virus (and presumably not in non-virus files). That sequence constituted the virus signature that was added to the database of signatures maintained by the antivirus program.

During a virus scan, the antivirus program would open all files and run a pattern-matching algorithm over the file. If a signature was detected in the file, it was infected but otherwise was virus free. Needless to say, this process made virus detection relatively easy (at least by today's standards) until the number of viruses grew to outpace the ability of early programs to effectively scan all files.

Antivirus designers responded by taking advantage of some characteristics of viruses to improve scanning performance, including:

- Most early viruses were relatively small, using less than 8KB of code
- Viruses tended to be located at the beginning or end of a file
- Viruses infect at the entry point of a program, which is the first instruction executed by a program; by analyzing the entry point instruction, virus scanners could detect where the flow of control is transferred, which, in the case of an infected file, would be the location of the virus code

These observations allowed antivirus designers to vastly improve performance over full-file scanning as well as limit the length of scans, focusing on entry points and other techniques derived from the analysis of typical virus structure and function.

Although it is almost a decade old, Carey Nachenberg's paper "Computer Virus Anti-Virus Coevolution" is still an excellent overview of the early approaches to virus detection. The paper is available at http://crypto.stanford.edu/cs155/virus_antivirus_coevolution.pdf.

With antivirus scanning keeping up with the volume of viruses, virus writers had to turn to new techniques to avoid detection.

Avoiding Detection with Encryption

The next step in the evolution of viruses was the introduction of encryption routines. This functionality allowed viruses to hide themselves until they were executed. The process works as follows:

- The virus writer creates a virus and embeds encryption and decryption routines.
- When the virus replicates, it encrypts the virus using a different encryption key from the previous generations. This method allows each replicated version to appear different from the other versions.
- The replicated virus continues to carry with it the decryption routine that is executed when the virus is activated.

Traditional virus detection signatures did not work at first because the viruses were encrypted while stored in a file and only decrypted when executed. Fortunately, as Figure 3.3 shows, the decryption routine, although small, was still unencrypted and could be used to identify viruses.

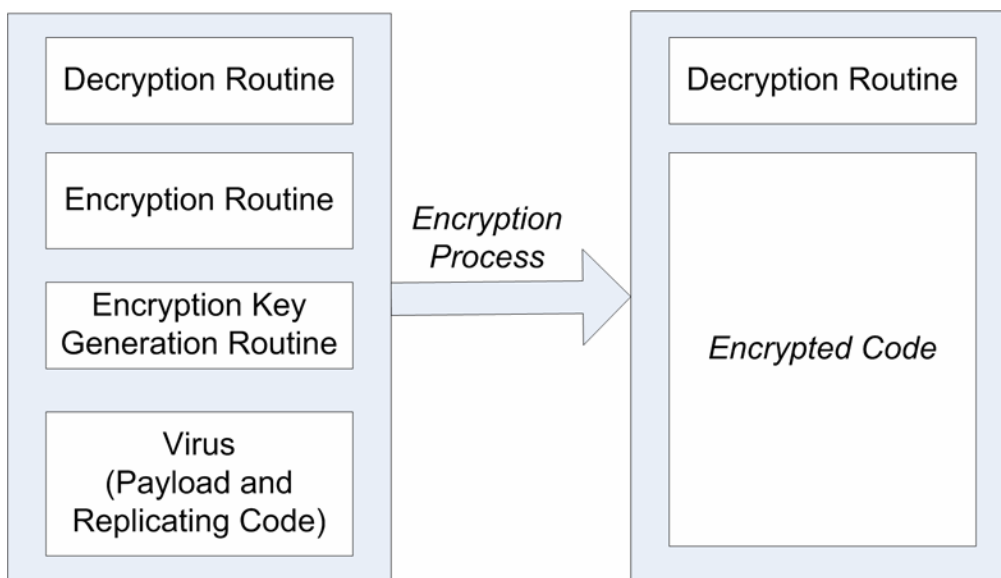


Figure 3.3: Even with encryption, enough of the virus program remains unencrypted to allow for signature-based detection.

The one drawback of this approach is that decryption routines can be quite short. Short patterns are more likely to be found in files than long patterns. For example, "mai" is more likely to be found in a randomly selected word document than "virus infected email." The short patterns led to an increase in false positives, which was easily countered by a technique known as x-raying in which the virus is decrypted and the contents examined for known virus signatures. The simple encryption techniques used with viruses made the decryption process relatively straightforward.

Radical Evolution—Polymorphic and Metamorphic Viruses

The basic signature-based model of virus detection worked for years, even withstanding challenges in the volume of viruses and the use of encryption to hide virus code. The most radical change up to that point in the virus/antivirus co-evolution came in 1992 with the introduction of mutation engines.

As the decryption routine was the Achilles heel of virus writers, they needed a way to mask that code. Encryption would not work because the routine must be in its decrypted form to work. Encryption was not used to keep the virus confidential (the usual reason for encrypting) but to prevent it from being identified by signature-based antivirus detection. An alternative method of masking the code was to change the sequence of instructions in such a way that still maintained the essential functionality of the virus. Mutation engines consist of three parts: a disassembler, a reverse engineering module, and a transformation module (see Figure 3.4).

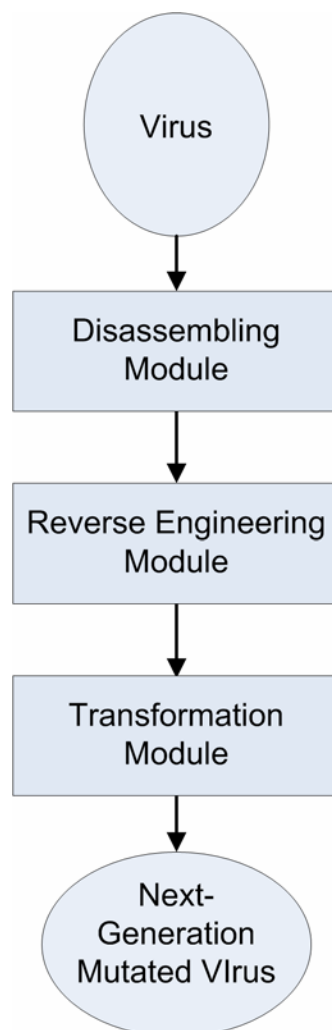



Figure 3.4: A mutation engine uses a three-stage process to analyze a virus and transform it into a functionally equivalent but structurally different next-generation version of the virus.

The job of the disassembler is to identify the programming instructions and data in a virus so that critical pieces of data (such as the length of an offset or memory reference locations) are not changed. The reverse engineering module identifies logical blocks of code for the transformation module to manipulate, and the transformation module maps single instructions or blocks of instructions into functionally equivalent code.

 Reverse engineering and transformation techniques are beyond the scope of this chapter. For more details about the structure and functioning of mutation engines and a case study of the Win32.Evol metamorphic virus, see “Are Metamorphic Viruses Really Invincible?” at <http://web.cacs.louisiana.edu/~arun/papers/invincible-complete.pdf>.

Although writing a mutation engine is difficult, the basic function of the mutation engine is simple: introduce useless instructions to fool signature-matching programs. These useless instructions could include:

- Adding zero to a number
- Multiplying by one
- Checking a condition that is always true or always false
- Shifting a bit pattern in one direction and then back in the other direction

Polymorphic viruses still encrypt the virus and the encryption routine, but, in addition, they introduce random useless instructions that scramble the decryption routine. The result is that there is no longer even a small fraction of code for antivirus signature-based detectors to latch onto.

Virus writers now have two techniques for hiding their code: encryption and mutation engines. These can be applied in different ways and result in a few different types of mutating viruses, categorized as

- Oligomorphic virus—A virus that changes its encryptor but not its decryptor
- Polymorphic virus—A virus that changes its encryptor as well as its decryptor
- Metamorphic virus—A virus that changes its encryptor, decryptor, and its *payload*, which is the destructive part of the virus

Metamorphic viruses are especially challenging—even if a decryptor was scrambled, once the virus was decrypted (virus encryption is generally not as difficult to crack as encryption techniques used to keep information confidential), the antivirus program could scan for known fragments of virus code. Metamorphic viruses change the payload of a virus by introducing useless instructions or substituting one set of instructions for another set of functionally equivalent instructions. For example, rather than multiplying a number by four, the code would multiply the number by two twice. Antivirus programs cannot realistically check for all possible combinations of malicious code and useless instructions, so signature-based detections are not sufficient to detect these threats.

Detecting Complex Viruses

There are a few general techniques for detecting stealth viruses. They are all based on the premise that for a virus to cause damage, it must act on the infected system in some way. Granted, this statement is somewhat like saying that water is wet, but what is important is not so much this first principal as what you can derive from it, such as

- Viruses may change files on the compromised host
- Viruses will make system calls
- Viruses will have patterns of execution different from non-virus programs

Starting with these premises, antivirus designers have developed techniques that complement signature-based detection. Static analysis is the process of analyzing the properties of a program without actually executing it. Compiler designers have used static analysis for generations to optimize compiled code, and similar techniques can be used to detect suspicious patterns of system calls and other indicators of malicious code.

Dynamic analysis techniques execute suspicious code in an emulator and monitor the behavior of the program. Again, suspicious system calls or attempts to intercept system calls can indicate a virus.

Integrity checking techniques examine files for unexpected changes. For example, a checksum is calculated for all executable files and stored for future reference. Periodically, checksums are recalculated and compared with the original checksum. If they are different, the file has changed.

Needless to say, virus writers have responded to these additional detection techniques as well. Responses include:

- Tricking integrity-checking programs to prevent them from detecting changes in checksums by intercepting file I/O and opening uninfected versions of files
- Disguising system calls
- Detecting monitoring programs and changing program behavior in response
- Detecting when the virus is executing within a dynamic analysis emulator and not decrypting code

Again, another iteration of the cat and mouse game is underway.

State of Virus Detection

Virus writers have circumvented signature-based detection with a variety of stealth techniques and have found ways to avoid dynamic, behavior-based detection. However, antivirus researchers are exploiting the techniques used by polymorphic and metamorphic viruses to detect the malware.

New detection techniques are based on the general methods—rather than the specific instructions—used by viruses to mutate. Mutation engines depend upon program analysis techniques similar to techniques used by compiler writers. Both mutation engines and compilers map one program into a functionally equivalent but structurally different representation. These techniques have theoretical limitations that force them to “show their hand” so to speak.

Exploiting those limitations is not a trivial task and detection techniques are constrained by theoretical limits as well. For example, a classic unsolvable problem in computer science is writing a program that determines whether another program terminates. Known as the Halting Problem, this idea is highly relevant for techniques for analyzing program behavior to determine whether a program is a virus. Virus writers, for example, can include long-running loops in their code knowing that antivirus programs will not generally be able to determine whether a running program will go on forever or eventually stop. Eventually, a behavior analysis program will have to stop a simulated run and decide whether the program is a virus.


Trends in Virus Evolution

Viruses have evolved along a number of dimensions. They have become progressively more difficult to detect. At first, encryption was used to hide the most obvious identifying patterns of viruses. Next, mutations were introduced to limit the usefulness of signature-based detection. When antivirus designers deployed countermeasures to the radically new mutating viruses, virus writers adopted additional techniques—most notably, attacks on countermeasures, such as hiding system calls, trapping system calls, and manipulating files to hide telltale signs of changes to files.

Both virus and antivirus developers are now working within the theoretical limits of the current static and dynamic analysis techniques. What does the future of virus development look like? It is impossible to say exactly how the continuing cat and mouse scenario will unfold, but the next stages of the virus evolution might realize a number of changes including:

- More sophisticated combination of existing techniques
- More targeted attacks on antivirus programs and other countermeasures
- Exploits of yet-unknown opportunities in new platforms, including 64-bit CPUs and OSs

There are examples of each of these in existence today. Combining different types of malware is common and discussed in detail in the later section on blended threats. Just as antivirus developers study viruses, virus writers as a group have made it a goal to know their enemy and there is no reason to think this trend will end. Viruses targeting 64-bit platforms—such as the proof-of-concept virus, W64/Rugrat—have been discovered. Nonetheless, viruses are not invincible.

 For more information about the W64/Rugrat virus, see http://vil.nai.com/vil/content/v_125990.htm.

There are theoretical limits of computation to constrain virus writers. Although it may be challenging, antivirus designers will always have that as a firm foundation to work from, even if those constraints put limits on their work as well. As threatening as they are, viruses are not the only form of malware that must be controlled.

Worms and Vulnerabilities

Viruses depend upon other programs to function. Like their biological namesake, computer viruses lack some basic machinery to fully function by themselves. Computer worms are like viruses in that they propagate and typically carry malicious payloads; unlike viruses, they do not depend upon other programs to operate.


Worms are fully functional programs that propagate by exploiting vulnerabilities in network software and applications. Two early examples are the Morris Worm and the Melissa worm.

Early Worms

Worms have existed since at least 1978 when Xerox PARC researchers invented the first-known computer worm. The Morris Worm, released in 1988 by a computer science graduate student at Cornell, was the first worm to spread widely on the Internet. The worm took advantage of bugs in three programs distributed with the BSD UNIX OS: sendmail, fingerd, and rsh/rexec.

In 1999, the fastest spreading virus to that date infected email systems around the globe: Melissa. The original version of the virus spread as a macro within Microsoft Word documents that were attached to email. When a recipient of the infected message opened the attachment, the macro virus automatically executed. The virus emailed itself to the first 50 names in the victim's Microsoft Outlook address book. Once activated, the worm would also infect other Word documents on the disk. Variations on the original Melissa worm varied the subject line, emailed 100 instead of 50 contacts, and deleted critical files, including `c:\command.com` and `c:\io.sys`.

Like viruses, worms have evolved from their early days to more sophisticated and threatening forms. Studying the evolution of viruses is instructive for understanding the challenges of detecting malware; similarly, studying examples of worms is instructive for understanding vulnerabilities in systems as well as the emerging incentives for writing malware.

 See Bob Page's "A Report on the Internet Worm" for an analysis of the Morris Worm. The paper is at <http://www.ee.ryerson.ca:8080/~elf/hack/iworm.html>.


Implementation Techniques and Consequences

Worms use a variety of techniques for spreading. Sometimes they email themselves, take advantage of low-level Internet protocols, or exploit a bug in network or application services. The most common transmission methods are

- Email
- Vulnerabilities in Internet protocols or applications, such as IIS
- Instant messaging systems
- Internet Relay Chat (IRC)
- Peer-to-peer file-sharing systems

The consequences of these implementations can vary as well from nuisance, to a Denial of Service (DoS) attack, to a commandeering of computing resources. Some instructive examples of worms are:

- Sobig
- MyDoom
- Sdbot
- SQL Slammer

 For details about these and other worms and viruses, see the McAfee Virus Information Library at <http://vil.nai.com/vil/>.

Sobig

Sobig is a worm that spreads through email and infects Microsoft Windows OSs. The worm spreads as an attachment to an email message and infects a compromised computer. There were a number of variants of the Sobig worm. After copying itself to a computer, the worm would do the following:

- Copy itself as an .exe file to the Windows system directory
- Create a data file in the Windows system directory
- Add an entry to the Windows registry causing the program to run when Windows starts
- Attempt to copy itself to network shares but failed due to a bug in the program
- Email itself to other victims using a randomly selected email address found on the infected system in the From field of the email.

Variants of Sobig used their own Simple Mail Transfer Protocol (SMTP) program to send themselves to other targets. The payload of the worm was programmed to contact specific Internet addresses and install additional programs including a backdoor that could be used by the worm writer to commandeer the compromised host's resources for spamming or to steal confidential information.

Rather than just leave a Trojan horse program on the infected machine, Sobig attempted to add machines to a collection of network relays that could be used as a distributed computing service. Sobig is illustrative of how worms can spread by email and install Trojans for later use by the worm writer.

MyDoom

MyDoom, which appeared in January 2004, was another mass mailing worm similar to Sobig in a number of ways, including spreading via email, spoofing email sender information, and setting up back doors for use by the worm programmer. It also expanded beyond the functions found in its predecessor, Sobig, and in the process became the fastest-spreading mass-mailing worm up to that time.

The worm spreads using an attachment that resends itself to email addresses found on the compromised machine. It also copies itself to folders used by peer-to-peer networks, such as KaZaA, allowing it to spread through file-sharing programs as well.

The original version used one payload to establish a backdoor on TCP port 3127 and allow the worm writer to take control of the compromised hosts. A second payload was included to launch DoS attacks against the SCO Group, a UNIX vendor, but did not always function correctly. A later version added a DoS attack against Microsoft. MyDoom is an example of a fast-spreading worm that uses two vectors of transmission, email and peer-to-peer file sharing.

Sdbot

The Sdbot worm appeared in February 2005 and spreads through instant messaging and takes advantage of a buffer overflow vulnerability in the Windows remote procedure call (RPC) interface that allows the attacker to run programs on the compromised host. It also uses a vulnerability in the Local Security Authority Subsystem Service (LSASS) to execute arbitrary code. Once on a machine, the worm copies itself in several ways:

- Insecure network shares
- Poorly secured SQL Server database instances
- Poorly secured MySQL database instances

Once on a machine, the worm installs itself in the Windows system directory and changes the system registry to run the program on startup. The payload of this virus includes a *bot*, or software agent, that can download and execute programs as directed by the worm's writer or other controller. McAfee identified a number of remote access functions that can be performed by the worm, including:

- View, modify registry data
- Browse filesystem
- Browse, terminate, start processes
- Upload, download, delete, modify, execute files
- Run FTP server
- Run HTTP proxy
- Run SOCKS proxy
- Manipulate (add, remove, modify) shares on victim
- Log keystrokes
- Launch DoS attack from victim machine
- Perform network scans (locate other vulnerable machines)

With these functions, the attacker can effectively take control of a machine and use it for a number of illegitimate uses, including stealing personal information and sending spam. One of the most dangerous types of programs that an attacker can download is a rootkit. Rootkits are sets of tools that allow an attacker to control a computer while disguising its actions. Rootkits are extremely difficult to detect; the best solution is to prevent them from reaching computers in the first place.

Sdbot is illustrative of the use of IRC and instant messaging systems to propagate. Like other worms, Sdbot leaves backdoor programs and Trojans, creating the opportunity for further exploitation by the attacker.

SQL Slammer

SQL Slammer is probably one of the best known Internet worms. In January 2003, a small 376-byte program effectively shutdown large segments of the Internet by flooding the network with useless traffic. The damage caused by some worms, such as Sdbot and MyDoom, is caused by their relative sophistication. In the case of SQL Slammer, the damage was due to several factors, including:

- The ability to exploit a single, widespread vulnerability in Microsoft SQL Server
- The speed of the protocol used, which was the User Datagram Protocol (UDP)
- The inherent trust (lack of authentication) in the protocol that was exploited

Database Vulnerability

The worm took advantage of two vulnerabilities in the SQL Server Resolution Service, a service designed to ensure database requests are sent to the correct instance of SQL Server when multiple instances are running on a single machine. Like other buffer overflow exploits, this one takes advantage of the fact that the SQL Server Resolution Service expects a piece of data to be at most a certain length (in this case, a 16-byte database name). The worm author crafted a packet of information with data longer than 16-bytes, causing a buffer overflow. (Actually, there were two different overflows, one overwrote the *heap*, a data storage area; the other overwrote the *stack*, the instruction storage area.)

The SQL Server Resolution Service did not detect an error but simply continued to execute; however, now it was executing code written by an attacker instead of Microsoft developers. The program was simple:

- Determine the number of milliseconds since the server was booted
- Use the number of milliseconds as a destination IP address in a UDP packet
- Copy the program to the data segment of the UDP packet
- Repeat

Of course, randomly generating IP addresses will generate some that do not resolve to actual servers and many that are not running SQL Server. However, the speed at which the program was able to execute and speed of UDP made this simple but rapid attack quite effective.

UDP—The Minimalist Protocol

UDP is as simple a network protocol as one will find. Figure 3.5 shows the structure of a UDP Packet.

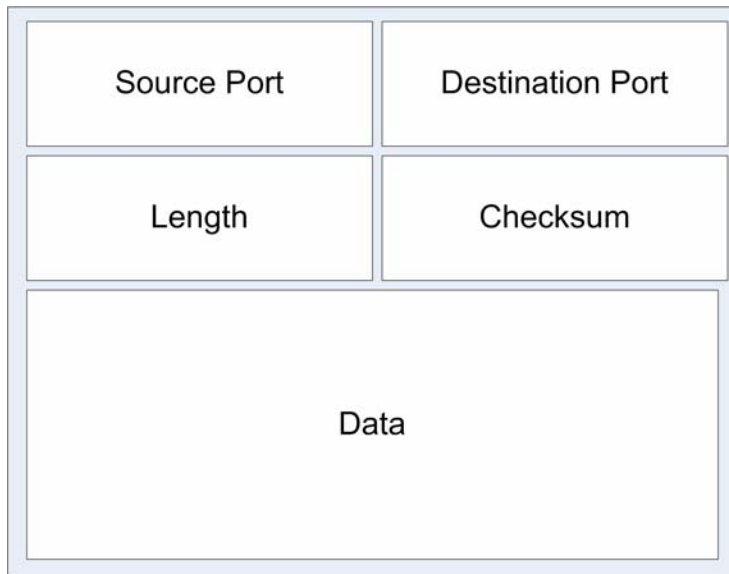


Figure 3.5: The UDP packet contains only address information, basic data integrity data, and the data itself.

UDP is one of the two network-level protocols used by the Internet. The other, TCP, is much more complex. When data must be reliably delivered from source to destination in the correct order and with some control over the speed at which packets are sent, then TCP is used. This protocol provides more guarantees about the information delivered but at the cost of significantly higher overhead.

UDP is used when packets can be sent but acknowledgement is not required. Packets can arrive out of order without causing problems for the recipient. In fact, packets may not be delivered at all, but services that use this protocol have a higher tolerance for failure than those that use TCP. Programs that use UDP are typically network services, such as:

- Domain Names Service (DNS)
- Dynamic Host Configuration Protocol (DHCP)
- Simple Network Management Protocol (SNMP)
- SQL Server Resolution Service
- Routing Information Protocol (RIP)

When programs use these services, they exchange small amounts of information and, if necessary, can request again and resend information without adversely affecting the source and target applications. The simplicity of UDP is also its weakness. The protocol has no flow control mechanism, so a network can be flooded with UDP packets to the point that other services cannot function. This is what happened with SQL Slammer.

Trusting Protocols

Another factor that affects both TCP and UDP is that they inherently trust the source of a packet. There is no authentication of a server with these network-level protocols (there is with other protocols, such as SSL/TSL). In the case of UDP, there is no mechanism to slow or discard packets that are clearly flooding the network. Furthermore, it is not at all clear that these services should be at the network protocol layer. High volumes of traffic from a single source are better controlled by routers and intrusion prevention systems than at the lower levels of the network.


 For a detailed description of the impact of the SQL Slammer worm, see Paul Boutin's "Slammed: An Inside View of the Worm that Crashed the Internet in 15 Minutes" at <http://www.wired.com/wired/archive/11.07/slammer.html>.

Unseen Vulnerabilities—Lack of Knowledge

The impact of SQL Slammer did not have to happen. Microsoft published a patch for the buffer overflow vulnerabilities in the SQL Server Resolution Service 6 months before the attack. Obviously, there were tens of thousands of unpatched servers in on the Internet. Some systems administrators may not have known about the patch and others may have known about it and choose not to apply it.

This reaction might sound irresponsible to some, but as any seasoned systems administrator knows, applying a patch can break as well as fix. Organizations that depend on SQL Server, or any production application for that fact, for essential services should not apply patches without testing them in a controlled environment. In addition to ignorance about the existence of the vulnerability and the patch, there are some users that probably did not know they were running a version of SQL Server known as Microsoft SQL Server Desktop Engine (MSDE), which is licensed free of charge with some applications.

SQL Slammer is illustrative of the fact that worms need not be complex to cause significant damage. It also illustrates the existence of inherent vulnerabilities in basic Internet protocols as well as enterprise applications.

 Databases should never be placed unprotected on the Internet. Database servers should at least be behind a firewall. A better option is to make the database server accessible only to an application server that is also protected behind a firewall.

Increasing Malevolence

Worms are becoming more destructive. Early worms, such as the Morris Worm, disrupted Internet services, but systems administrators were able to recover without long-term damage. Worms such as Sdbot and SQL Slammer show that malware developers are effectively exploiting vulnerabilities throughout the Internet infrastructure and in application services. Like viruses, worms are becoming more destructive. As if that were not enough to keep antivirus developers on their toes, malware developers have combined techniques to form a new type of malware—the blended threat.

The New Frontier—Blended Threats

Blended threats are malware that use multiple techniques to spread and cause damage. A blended threat may contain several of the following:

- Virus
- Worm that propagates by multiple methods
- Trojan horses
- Keylogger
- Frame grabber
- SMTP email engine
- Rootkits
- Backdoors

Any one of these types of malware can cause significant damage; in combination, the potential is much greater. Blended threats were launched as early as 2001 with the release of the Code Red and Nimda worms. Code Red launched Distributed DoS (DDoS) attacks, changed registry settings, created network shares, and spread via email and vulnerable IIS Web servers. Nimda had similar characteristics and even went so far as to attack backdoors left by Code Red II. Some of the most notable characteristics of blended threats are their

- Methods of attack
- Methods of transmission
- Methods of control

Multiple Methods of Attack

Blended threats use multiple methods of attack. This behavior increases the chance of success (that is, from the malware developers point of view) when deployed into environments with countermeasures in place. For example, laptops with up-to-date antivirus software may not be vulnerable to a virus infection from a polymorphic virus but it could be vulnerable to a worm that exploits a vulnerability in the Microsoft SQL Server Desktop Engine.

Attacks can come in the form of:

- Deleting files
- Changing registry settings
- Infecting executable files
- Deploying backdoors for later use
- Disabling antivirus software

Although vandalism malware is still a threat (for example, defacing a Web site), stealing information and computing resources are the real objectives of these attacks. Identity theft is a growing problem because identities have replaced cash as the object of criminal desire. Times have changed from when famed bank robber Willie Sutton targeted banks because “that’s where the money is.” The location and form of the money has changed; it is now in computers in the form of personally identifiable information.

In addition to personal information, just having access to the bandwidth and computing cycles of a compromised machine has economic value. Anti-spam legislation makes it difficult to carry out spamming activities in the open with fixed servers and known IP addresses. Instead, spammers turn to attackers who have commandeered collections of PCs through the use of backdoors and rootkits. Need to deliver a million pieces of spam? Find the right chat room, and you can rent a set of compromised machines to do the mailing for you. Attacks are changing and they are reflecting changes in the underlying motives of the attackers. Economics are driving malware.

Multiple Methods of Transmission

Blended threats may include virus-like components, but they propagate without the use of other programs; in that way, they are a form of a worm. The Klez family of worms, for example, spreads by taking advantage of a bug in Microsoft Outlook’s Automatic Execution of Embedded MIME Type, which does not require any action on the part of the user. The malware then spreads using its own email engine after collecting email addresses from the compromised host. It also propagates by copying itself to network shares.

Typical transmission methods include:

- Employing virus-like file infections
- Copying to network shares
- Copying to peer-to-peer network shares
- Exploiting vulnerabilities in Web servers or other system applications
- Using email


Once they have transmitted and attacked, blended threats do not necessarily shutdown or cease their malicious activity.

Multiple Methods of Control

As noted earlier, the motives for hacking and attacking are changing. Economics is an established incentive in the world of malware. One of the ways cybercriminals can realize a return on their investment in malware development is by gaining control of a large number of computers.

Zombies, or computers that are available for control by attackers, are essentially free resources for malware developers. Once a worm has reached a PC and installed a backdoor program, the program can download updates, install additional malware, or even launch a mass mailing of spam. Computationally intensive tasks, such as cracking encryption keys, could also be parceled out to zombies.

Blended threats combine the techniques learned over decades of development and evolution of viruses and worms and now apply them to ever-more damaging activities. The world of attacking is changing; it is no longer the stereotypical lone hacker breaking into a system to demonstrate his or her technical prowess. Organized crime has discovered that money is not just in banks anymore.

 See “Hacker Hunters: An Elite Task Force Takes on the Dark Side of Computing” for an example of how one cybercrime group, ShadowCrew, was broken by the United States Secret Service and FBI. The story is available at http://www.businessweek.com/magazine/content/05_22/b3935001_mz001.htm.

Summary

Viruses began as the intellectual brainchild of computer science researchers in the 1960s and were rediscovered by teenagers with early PCs. Worms appeared later but followed a similar trajectory of evolution. Today, we’re confronted with a host of well-known techniques for creating, deploying, hiding, and modifying malicious programs. We are also becoming painfully aware of the vulnerabilities that exist in the trusted code that runs our organizations and serve us in home use. The combination of the two is challenging to say the least.


Fortunately, there are countermeasures. Antivirus and anti-worm software has co-evolved along with threats. Improved security practices and system patching procedures can minimize vulnerabilities. Controlling malware is not a project that can be completed; we will always have to defend against it.

Chapter 4: Spyware and other Potentially Unwanted Programs

Spyware is a type of Potentially Unwanted Program (PUP) that monitors users' online behavior as well as performs other tasks, which this chapter will explore. As with other forms of malware, the use of spyware has increased and studies have shown it affects large numbers of Web users. The pervasiveness of spyware is not limited to a particular segment of the population or to particular types of Web users; it is a problem for home users as well as businesses and other organizations that support large numbers of users.


In the case of home users, a 2004 survey by America Online (AOL) found that 80 percent of the systems surveyed contained at least one known spyware program. (It also found that 20 percent of those systems hosted a virus). Compounding the problem is a lack of understanding about the issue.

In the AOL survey, two-thirds of respondents felt their computer was safe from online threats. A 2005 survey from the Pew Internet and American Life Project found similar confidence in users' ability to stop potentially unwanted programs. The Pew survey found that 61 percent of home users felt very or somewhat confident that they could keep malware, as well as spyware, off their computers.

 For more details about the AOL and Pew surveys, see "AOL Survey Finds Rampant Online Threats, Clueless Users" at <http://www.computerworld.com/securitytopics/security/story/0,10801,96918,00.html> and "Spyware Survey" at http://www.pewinternet.org/pdfs/PIP_Spyware_MayJune05_Qs.pdf.

The spread of spyware in corporate environments is comparable to that found in home users. Leading anti-spyware vendors are finding increasing numbers of spyware getting installed on users' systems. For example:

- The McAfee Anti-Virus Emergency Response Team (AVERT) Labs witnessed a 12 percent rise in spyware incidents from the first quarter to the second quarter of 2005 and a 60 percent increase since the first quarter of 2004.
- According to a WebRoot survey of enterprise clients, 88 percent of enterprise computers had instances of spyware on them in 2005.
- The problem is also manifesting itself at the Help desk. Twenty percent of support calls to Dell in late 2004 were spyware related, which is an increase of 12 percent from earlier in the year.

 For more information about the state of spyware in corporate environments, see "McAfee AVERT Reports Top 10 Threats for 2004 and Advises on Future Threats and Trends" at http://www.mcafee.com/us/about/press/mcafee_enterprise/2005/20050103_093758.htm, "Spyware: A Customer Relations Problem" at <http://www.technewsworld.com/story/40419.html>, "Dell Spyware Decision Spurs New Trend" at <http://www.crbuyer.com/story/37668.html>, and "First State of Spyware Report Shows Bad Guys Winning" at <http://www.technewsworld.com/story/42844.html>.

Fortunately, there appears to be more awareness of PUPs in organizations than among home users. According to an *Information Security* survey, countering spyware is a top security priority for the next 18 months for more than two-thirds of respondents.

This chapter will delve into the details of PUPs—how they threaten information security and what can be done about it. The discussion is divided into three areas:

- Varieties of PUPs
- PUP behaviors
- Impact of PUPs

Let's begin with a look at examples of PUPs before developing a more formal definition.

Varieties of PUPs

PUP is a general term for a broad class of programs that include:

- Adware
- Spyware
- Monitoring programs
- Keyloggers
- Password stealers
- Tracking cookies

Although these types of PUPs are distinct, some features may overlap. Just as viruses and worms have evolved into multifaceted threats, known as blended threats, a single spyware application may contain multiple components. To better understand these various types of programs, let's look at an overview of the broader class of PUPs.

PUPs

The Internet is a distributed computing platform that allows relatively easy access to a wide array of information and services. It also provides a direct line into vulnerable systems that can become infected with a variety of PUPs. PUPs are programs written for a seemingly legitimate purpose that alter the security or privacy posture of a computer on which it is installed. PUPs may be distributed as standalone programs or may be included with popular types of applications and file types. For example, PUPs can be:

- Peer-to-peer file-sharing clients
- Screensavers
- Utilities, such as clock synchronization programs
- Browser toolbars


These vectors, or methods of distribution, can be used to carry a number of different types of PUPs.

Adware

Adware is a common type of PUP whose primary function is to derive advertising revenue for a third party. Adware can deliver pop-up and pop-under advertisements and banner ads in addition to tracking Web activities. Widely deployed adware programs include:

- Gator
- 180SearchAssistant
- BonziBuddy
- GAIN

Gator from Claria Corporation is one of the best-known programs in the area of “online behavioral marketing,” in which the online habits of users are tracked and used to profile user interests and target advertising to those interests. If users knowingly agree to allow the installation and use of monitoring programs on their computers, the application may be categorized as adware.

 No one likes to read End User License Agreements. They are long, legalistic, and often incomplete; but they are worth reading if you are downloading a freeware or shareware program that may be used to install adware. This type of download can include just about any utility or popular program, but not all free programs are spyware. When in doubt, read the End User Agreement or pass on the program and do not download it.


In addition to the behavioral marketing monitoring programs, homepage hijackers also fall into the adware category of PUPs. These programs modify home page settings to redirect browsers to a particular page, thus boosting the number of hits on that page. In general, manually resetting the home page property in a browser only works temporarily because the home page hijackers will change it back again.

One of the ways home page hijackers work is by adding an entry to the Run registry keys in Windows. In Windows NT 4.0, Windows 2000 (Win2K), Windows XP, and Windows Server 2003 (WS2K3) the keys are:

- HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run
- HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run

Every time the system starts, the commands listed in the \Run registry setting are executed. The home page hijacker inserts a command into the entry that causes the home page setting to be reset. Home page hijackers can be overcome by deleting those registry entries.

In other cases, hackers have exploited a vulnerability in Internet Explorer (IE) whereby they use HTML Application files (.hta) to install ActiveX components when a user browses a site. When an .hta-based home page hijacker infects a system, users can search for and remove any unauthorized .hta files to clean up the problem.

 HTML Application is a Microsoft proprietary protocol and works only with IE. Other browsers, such as Mozilla Firefox and Opera, are not vulnerable to this threat. Microsoft has provided a patch for this vulnerability.


Of course, home page hijackers can enter a computer as part of the payload in a blended threat and use injection methods commonly seen by viruses as well.

Spyware

Spyware is software that gathers personally identifiable information about a computer's user and delivers that information to a third party. Spyware is often distributed with adware, but may also be downloaded while browsing sites that download unwanted software along with Web content.

In some cases, software distributors openly admit to deploying data collection software or programs that utilize computing resources on client machines. Companies, such as Brilliant Digital Entertainment, have openly admitted to downloading programs for the purpose of using computing and storage resources of users of the company's program.

For example, in their 2002 SEC filing, the company described an installer that would be downloaded along with the popular Kazaa peer-to-peer file-sharing system client as "components [that] help facilitate the delivery of files—ad banners, music files, documents, software files, etc.—across the network. Apart from facilitating Altnet SecureInstall connectivity, the Installer is a full-fledged software installation system, with key features including file compression, file patching, and file encryption." Spyware can be a component in a more sophisticated distributed application.

 See the full text of Brilliant Digital Entertainment's 2002 SEC 10KSB filing at <http://www.sec.gov/Archives/edgar/data/1022844/000101143802000252/0001011438-02-000252.txt>.

Once a spyware program is on a computer, there are a number of methods for stealing information. One of the most dangerous is the use of keyloggers.

Keyloggers

Keyloggers are programs that record every keystroke, allowing the user of the keylogger to acquire usernames, passwords, account numbers, and other identifying information (see Figure 4.1). Because the keyboard is the primary vehicle for user data entry, virtually no application is safe from this eavesdropping mechanism. In addition, data does not even have to be saved. An email can be typed but never sent, a word processing document edited and then discarded, or a username and password entered but then cancelled—the information could still be captured by a keylogger program.

Applications that demand high security have tried countermeasures such as visual keyboards to circumvent keyloggers. With visual keyboards, a display of a keyboard appears on the screen and users click the mouse over images of the keys. As we could have expected, keyloggers have evolved to include more sophisticated features, such as capturing screenshots and recording the names of windows and Web sites visited along with timestamps for these activities.

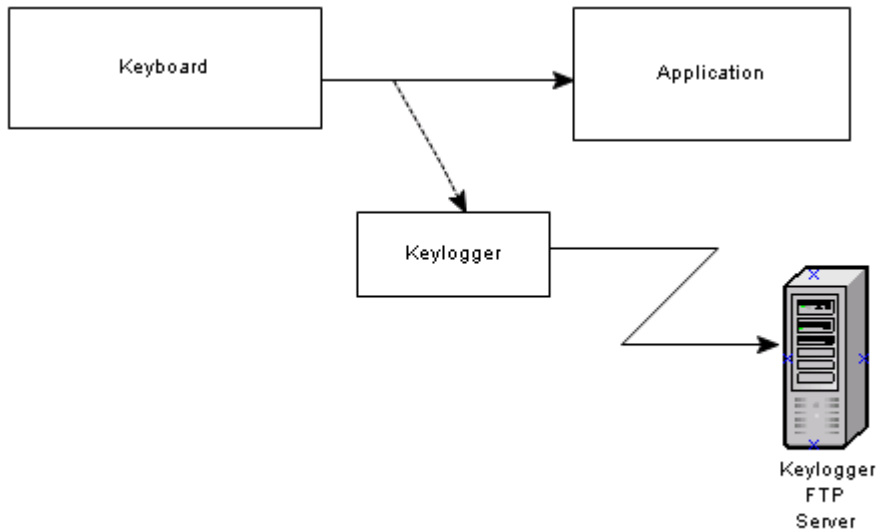



Figure 4.1: Keyloggers intercept communications between the keyboard and applications and send collected information to attacker-controlled servers, such as FTP or other collection servers.

Unlike other PUPs, not all keyloggers are used by attackers:

- Commercial keyloggers are sold to consumers as tools for monitoring children's and other family members' activities online
- Keyloggers are used by software developers to help diagnose software bugs
- The FBI has disclosed that it has developed and used a keylogging program known as Magic Lantern to retrieve encryption passcodes of suspects

 For more information about the FBI's disclosure on Magic Lantern, see "FBI Software Cracks Encryption Wall" at <http://www.msnbc.com/news/660096.asp?0na=x21017M32&cp1=1>.

 It should be noted that the keyloggers used by spyware are software-based devices. Hardware devices have also been developed as well. When they are installed between the keyboard cable and computer keyboard port, they capture and record keystrokes.

In some cases, PUPs are not designed to steal information but to use others' computing resources. One of the simplest PUPs of this type drives up the number of visits to a Web site.

Password Stealers

Password stealers are programs that read the text of passwords transmitted to applications and send them to a site where they are collected by the attacker. Hiding the characters in a password, as Figure 4.2 shows, is common but ineffective against these programs because the programs do not depend on reading what is displayed in the box, only what is transmitted from the client to the application.



Figure 4.2: Masking the characters in passwords does not prevent password stealers, which detect characters as they are sent from the keyboard to the application.

Password stealers are specialized forms of keyloggers. Both types use hooks in the operating system (OS) to detect keyboard events and record keystrokes.

Password stealers are evolving in a cat-and-mouse fashion. To avoid keyboard vulnerabilities (such as the SetWindowsHookEx function described shortly), some applications have deployed visual keyboards. Rather than type on the keyboard, users click on an image of a keyboard on the screen. The input program then uses the position of the mouse click to determine which character was selected.

Although visual keyboards avoid the problem of intercepted key strokes, they have similar limitations. In particular:

- Mouse events can be intercepted as are key strokes
- Screenshots can be captured making a recording of the mouse position over the visual keyboard.

Password stealers are a substantial threat to information security. Comprehensive policies on using strong, difficult-to-crack passwords, changing passwords frequently, and not sharing passwords are all moot once a password stealer has captured a username and password. One way to detect unwanted programs that startup automatically is to use a startup program reporting tool, as Figure 4.3 shows.

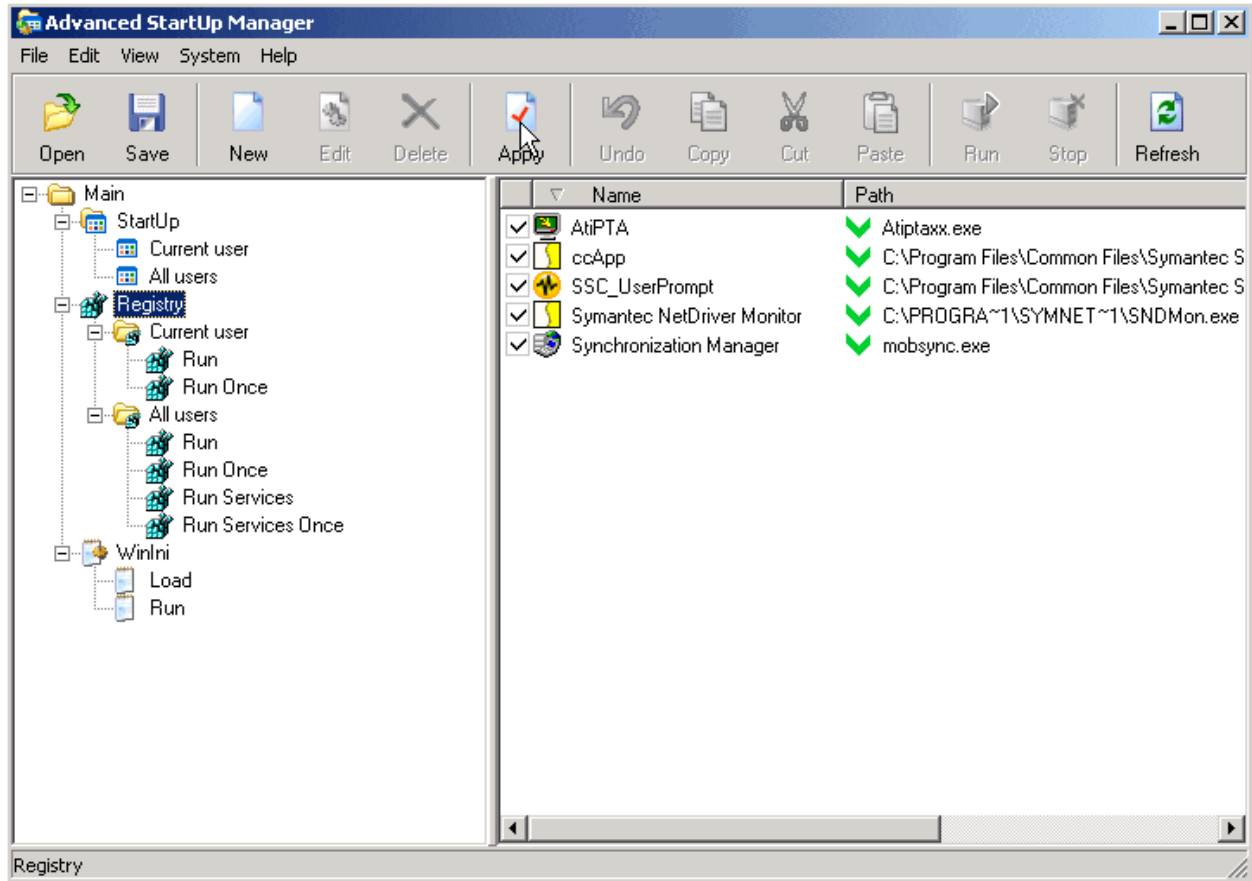


Figure 4.3: Utilities, such as a startup program reporting tool like Advanced Startup Manager, can help to identify programs set to auto start in the registry.

⚠ Be aware that some complex malware, known as root kits, can hide the existence of running processes. Less sophisticated malware can be identified with utilities.

Tracking Cookies

Cookies are text files used by browsers to record information locally. Cookies are used for a number of reasons, including:

- Personalizing preferences
- Remembering passwords
- Keeping track of a program state in Web applications
- Tracking user activities online

The first of these are legitimate uses of cookies; the final reason, tracking activities, can have legitimate uses but not always.

Personal Preferences

Personal preference cookies allow users to customize the look and feel of sites. For example:

- News sites may allow readers to rearrange the layout of news sections and can record those preferences using cookies.
- Web portals use cookies to record physical address information allowing the portals to display weather and other topics of local interest.
- E-commerce sites use cookies to remember account numbers, language preferences, and other information about cookies.

Typically, personal preference cookies are used for the convenience of the site visitors and do not collect personal information.



When in doubt about the information tracked in personal preference cookies, consult a site's privacy policy.

Remembering Passwords

Cookies are also used to remember passwords for Web sites and Web-based applications. Online newspapers, for example, often ask readers to create accounts and provide basic demographic information. For convenience, many readers will have the site save their login information.



Saving passwords in cookies is obviously a security risk and should only be used if the disclosure of the password would not compromise private information. Passwords to online bank accounts, brokerage accounts, health insurance plans, or other sites dealing with financial or confidential information should never be saved in cookies. When in doubt, do not select any option to save passwords.

Tracking Program State

One of the advantages of HTTP, the protocol underlying Web pages and many Web application interfaces, is its simplicity. It is also one of its drawbacks. A common problem in any application design is keeping track of what the user has done and where the user is in the overall process of the application. This is known as maintaining “state.”

When programmers developed applications for mainframes or client/server applications, maintaining state was relatively easy. The program would have a single channel of communication between the user and the application.

When applications moved to the Web, they no longer had a single channel of communication when using HTTP. Every message sent from a Web server or a browser was independent of other messages. Developers use cookies to save information in the browser so that it can re-use that information in later messages. For example, if a user adds an item to a shopping cart, it can be stored in a cookie and read when the user decides to check-out. The information is not lost if the user decides to browse for other items.

Tracking User Activity Online

Tracking user activity online is a commonly used technique by online advertisers. Regardless of whether the user approves, the purpose is to record events, such as visits to particular sites, and analyze patterns in usage.

This information is then used to target ads to users with particular interests. For example, if someone browses sites on hiking, mountain biking, and national parks, that person is a likely target for backpacking products. Studies have demonstrated that targeting ads based on behavioral profiling is more effective than non-targeted online advertisement, so tracking online user activity is not likely to diminish.

 For details of one study on the behavioral marketing's impact on revenue, see "Behavioral Targeting Study Reveals CPM Lift" at <http://www.clickz.com/news/article.php/3396431>.

Keeping track of cookies, and just knowing what is on a computer, can be a daunting task. Browsers have a number of tools for managing cookies, including privacy settings. A few simple steps can give users better control and information about cookies:

- Selectively disabling cookies from particular sites
- Setting browser security options to disable cookies in certain circumstances
- Using browser audit tools to review information collected and maintained by browsers

For example, Mozilla Firefox provides the ability to remove specific cookies and disallow future cookies from being set from the sites that have had their cookies removed (see Figure 4.4).

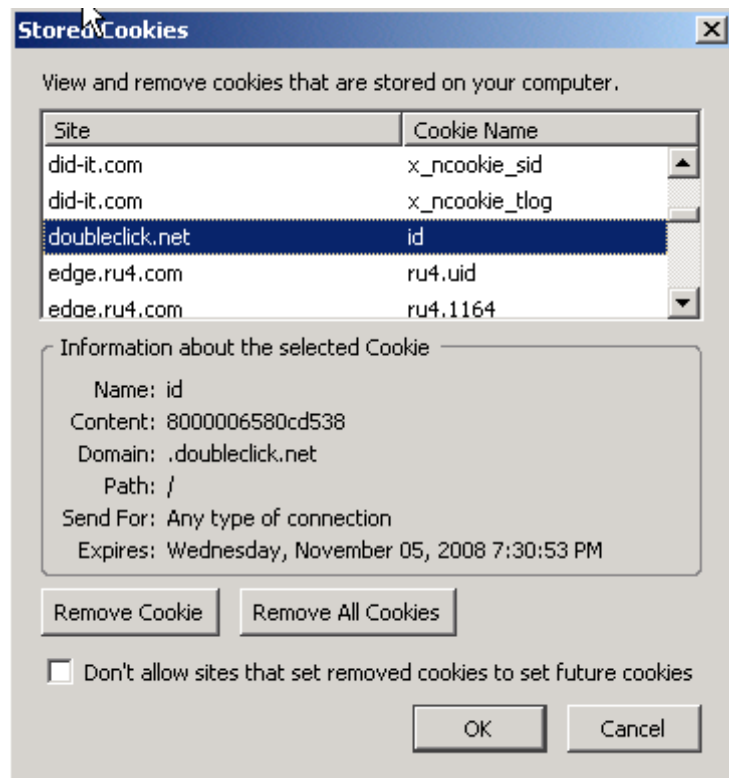


Figure 4.4: Mozilla Firefox allows users to view details of cookies as well as to remove and control the resetting of cookies.

IE provides general levels of cookie control and allows users to define custom configurations for handling cookies from first party as well as third-party sites (see Figure 4.5). A first-party site is one intentionally visited by a user; third-party sites are sites that the user did not directly navigate to but were contacted on behalf of the user by the first-party site. Third-party sites include Web-tracking sites and behavioral-monitoring sites.

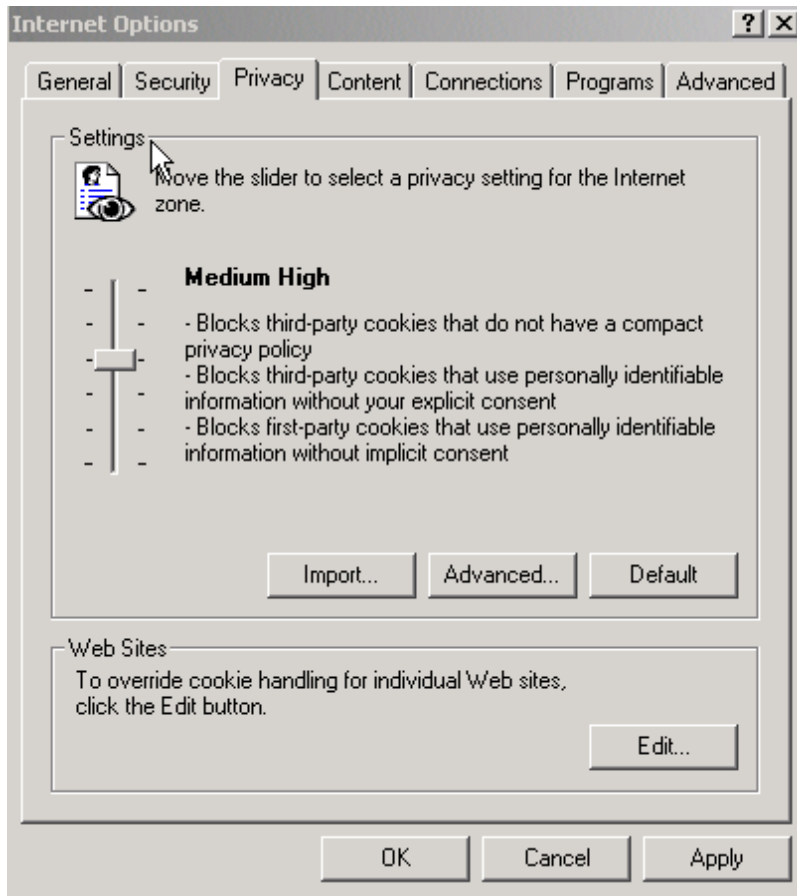



Figure 4.5: IE allows users to choose from predefined sets of cookie controls or to configure their own set through the Advanced option.

In addition to browser settings, third-party tools are widely available for examining cookies, browser history, caches, and related information stored in browsers.

It should be assumed, for security purposes, that any information stored in a browser is accessible to spyware, particularly tracking software. Cache and cookie audit tools (such as the one that Figure 4.6 shows) can be used to monitor information maintained by browsers. Other types of cookie-management utilities include cookie-removal tools, and real-time cookie blocking.

 Cookie management features vary by browser, so features in Mozilla Firefox may not be available in IE and vice versa.

URL	Filename	Hits	Size (Bytes)	Last Modified	Last Access
Cookie.dsullivan@burstnet.com/	\dsullivan@burstnet[2].txt	2	148	7/28/2005 10:23:48 ...	7/28/2005 10:23:48 ...
Cookie.dsullivan@ca.com/	\dsullivan@ca[1].txt	6	134	5/4/2005 7:19:29 PM	10/6/2005 5:47:12 P...
Cookie.dsullivan@centport.net/	\dsullivan@centport[1].txt	10	88	6/13/2005 7:09:56 P...	11/3/2005 6:57:21 A...
Cookie.dsullivan@chatboards.ebay.com/	\dsullivan@chatboards.ebay[1].txt	2	104	5/11/2005 4:00:21 P...	5/11/2005 4:00:29 P...
Cookie.dsullivan@cheats.gamespot.com/	\dsullivan@cheats.gamespot[2].txt	9	226	8/27/2005 10:42:32 ...	8/27/2005 10:42:32 ...
Cookie.dsullivan@choicehotels.com/ires	\dsullivan@ires[2].txt	4	258	7/31/2005 11:06:09 ...	8/9/2005 7:07:34 AM
Cookie.dsullivan@cisco.com/	\dsullivan@cisco[1].txt	1	97	9/5/2005 11:02:39 A...	9/5/2005 11:02:39 A...
Cookie.dsullivan@citi.bridgetrack.com/	\dsullivan@citi.bridgetrack[1].txt	34	1280	8/5/2005 5:01:17 PM	8/5/2005 5:01:17 PM
Cookie.dsullivan@click.theonion.com/	\dsullivan@click.theonion[1].txt	11	178	10/12/2005 3:52:46 ...	10/12/2005 3:52:46 ...
Cookie.dsullivan@cnn.122.2o7.net/	\dsullivan@cnn.122.2o7[1].txt	7	131	10/22/2005 7:47:06 ...	10/22/2005 7:47:06 ...
Cookie.dsullivan@cnn.com/	\dsullivan@cnn[2].txt	7	177	10/22/2005 7:47:05 ...	10/22/2005 7:47:05 ...
Cookie.dsullivan@cnaudience.com/	\dsullivan@cnaudience[1].txt	2	102	7/22/2005 2:17:40 P...	8/5/2005 4:50:48 PM
Cookie.dsullivan@com.com/	\dsullivan@com[1].txt	13	311	10/28/2005 1:33:55 ...	11/3/2005 6:56:16 A...
Cookie.dsullivan@commonservices.novartis...	\dsullivan@commonservices.novartis[1].txt	1	95	6/18/2005 12:20:55 ...	6/18/2005 12:20:55 ...
Cookie.dsullivan@creativeby.viewpoint.com/	\dsullivan@creativeby.viewpoint[1].txt	7	152	7/22/2005 9:28:04 P...	7/22/2005 9:28:04 P...
Cookie.dsullivan@data.coremetrics.com/	\dsullivan@data.coremetrics[1].txt	2	101	8/12/2005 11:06:23 ...	8/12/2005 11:31:55 ...
Cookie.dsullivan@dealtime.com/	\dsullivan@dealtime[1].txt	1	100	4/28/2005 10:52:49 ...	4/28/2005 10:52:49 ...
Cookie.dsullivan@dell.com/	\dsullivan@dell[1].txt	2	81	4/2/2005 7:32:56 AM	4/13/2005 4:20:02 P...
Cookie.dsullivan@delta.com/	\dsullivan@delta[1].txt	5	80	4/12/2005 9:21:35 P...	8/25/2005 10:40:47 ...
Cookie.dsullivan@did-it.com/	\dsullivan@did-it[2].txt	2	285	8/6/2005 10:03:10 A...	8/6/2005 10:03:10 A...
Cookie.dsullivan@dist.belnk.com/	\dsullivan@dist.belnk[2].txt	2	186	7/17/2005 8:06:42 A...	7/17/2005 8:06:42 A...
Cookie.dsullivan@dogpile.com/	\dsullivan@dogpile[1].txt	1	100	6/26/2005 2:52:05 P...	6/26/2005 2:52:05 P...
Cookie.dsullivan@doubleclick.net/	\dsullivan@doubleclick[1].txt	17	83	8/3/2005 9:47:58 PM	10/22/2005 7:46:59 ...
Cookie.dsullivan@ebay.com/	\dsullivan@ebay[2].txt	195	1065	5/11/2005 4:01:13 P...	8/27/2005 8:12:48 A...
Cookie.dsullivan@edge.ru4.com/	\dsullivan@edge.ru4[1].txt	33	1286	6/27/2005 2:04:10 P...	7/10/2005 9:08:02 A...
Cookie.dsullivan@ehg-airtran.hitbox.com/	\dsullivan@ehg-airtran.hitbox[1].txt	16	1068	8/4/2005 1:27:58 PM	8/4/2005 1:27:58 PM
Cookie.dsullivan@epinions.com/	\dsullivan@epinions[1].txt	1	99	5/6/2005 3:50:48 PM	5/6/2005 3:50:48 PM

Figure 4.6: Cookies are not easily segregated by function. Personalizing cookies as well as tracking cookies are not distinguished by browsers. (Screenshot of STG Cache Audit, a shareware utility available at <http://www.stgsys.com/audit.asp>.)

Defining Spyware

Perhaps the most well known form of PUP is spyware. This guide follows the criteria used by McAfee's anti-spyware researchers Prabhat K. Singh, Fraser Howard, and Joe Telfatici in their article "How Dare You Call it Spyware" published in the *Virus Bulletin*, December 2004 to define spyware.

📖 "How Dare You Call it Spyware" and related research, is available online at <http://www.virusbtn.com>.

Singh, Fraser, and Telfatici categorize spyware, and malware in general, according to six criteria:

- Installation methods and effects
- Concealment
- Injection
- Payload

The combination of structural and functional attributes provides a reasonable approach that should minimize otherwise unhelpful debates about what constitutes spyware and other PUPs.

Installation Methods and Effects


Installation programs change a device by installing code in such a way that it executes each time the system is restarted or when some predefined system event occurs. The program may be run by several methods, including:

- Adding a system service that starts when the computer is rebooted
- Employing COM objects that store startup information in the system registry
- Using Browser Helper Objects in IE
- Adding an entry to one of the two registry settings that control the automatic execution of programs during system startup
- Using network-level intercepts that redirect traffic to spyware-related sites

Each of these techniques changes either a system configuration or the functioning of a core component of the OS, usually without the user's knowledge.

Concealment

PUPs use many methods to hide their existence. The goal, of course, is to prevent detection and removal from a machine. Virus writers have created a number of techniques for changing the structure of a program without changing its behavior, which can be adopted for spyware.

 For more information about concealment techniques, see Chapter 3.


In addition to masking code, spyware writers may change registry entries to avoid detection. COM objects, for example, use globally unique identifiers (GUIDs), which are long strings of characters. These can be randomly changed on each installation to prevent anti-spyware programs from matching registry entries based on those character strings.

Injection

Injection is the process of inserting code into an executable object, such as another program. Some ways to accomplish this are:

- Inserting registry entries in such a way as to force a dynamic link library (DLL) to load in memory when the system starts
- Using the Windows system function SetWindowsHookEx function, which allows applications to monitor system events, such as keyboard events
- Injecting an execution thread in another process, such as IE

Spyware can use any of these techniques but the last is usually used.

 OS hooks are powerful tools for programmers but are dangerous on vulnerable systems. Hooks are designed to intercept messages between components, such as the keyboard and an application, and carry out additional activities. Hooks in the Windows OSs allow programmers to carry out tasks on keyboard and mouse events when the foreground task is not utilizing system resources and at other times as well. For technical details on Windows OS hooks, see the Microsoft Developers Network at <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/winui/winui/windowsuserinterface/windowing/hooks.asp>.

Payload Types

Payload is the part of a PUPs that carries out the program's core function, such as doing something to the host computer, communicating data to third parties, or receiving commands from third parties. Although much of the code of a virus, worm, or spyware program may be designed to control installation and concealment, the payload is the portion that delivers the core functions intended by the spyware writer.

Host payloads are program components that change the behavior of the host. This change can include:

- Displaying banner ads
- Intercepting and redirecting URLs
- Consuming computer and storage resources
- Monitoring Web browsing activities

Network payloads transmit information to and from the host computer. Viruses, worms, Trojan horses, and blended threats may establish connections to servers controlled by their program's developers to download updated payloads, receive instructions to carry out particular tasks—such as distributing spam, or initiate a Denial of Service (DoS) attack on a third-party site. Spyware typically sends information about user behaviors and activities back to a central server. Table 4.1 provides a summary of malware characteristics displayed by spyware.

Malware Characteristic	Use in Spyware
Installation methods and effects	COM objects, Browser Help Objects, and network intercepts
Survey techniques	Not used in spyware
Replication methods	Not used in spyware
Concealment	Use techniques developed for viruses; randomly generate GUIDs
Injection	Often inject code into IE process
Payload type	Both host- and network-based payloads

Table 4.1: Spyware can be identified using the criteria defined by McAfee's malware researchers.

With an understanding of criteria used for classifying spyware according to its structure and function, you can turn your attention to more specific spyware behaviors.

Spyware and other PUP Behaviors

PUPs can conceal themselves and carry an undesirable payload. Beyond the fact that their presence constitutes a risk, there are specific behaviors that are common to PUPs:

- Monitoring keystrokes
- Scanning files
- Reading cookies
- Changing browser and registry settings
- Installing Browser Helper Objects
- Capturing screenshots

Keystroke monitoring has already been discussed; the remaining behaviors are addressed in the following sections.

File Scanning

Once in place on a computer, spyware can set about its work of collecting information. This task is often done by monitoring activities but can also be accomplished by scanning existing data for sensitive information including:

- Usernames and passwords
- Account numbers
- Personally identifying information, such as Social Security numbers
- Potentially proprietary files, such as computer aided design files

File scanning is the general process of searching for information but can also be more targeted as with cookie reading.

Reading Cookies

Cookies are tied to a single Web site. Only the Web site that places the cookie is able to read it from within an HTML page. This setup allows Web site designers to maintain the privacy and integrity of their cookies. The obvious irony notwithstanding, it also serves to protect the interest of users who cannot have their browsing habits analyzed simply by reading all cookies on a computer (advertisers have found other ways; see the sidebar “Large-Scale Web Activity Monitoring”).

For spyware/adware writers, cookies are an obvious wealth of information. Users’ interests are documented in cookies. Spyware authors do not even have to know the structure of a cookie, just knowing the names of Web sites allows for profiling. Web directories such as Google Directory, Yahoo Directory, and the Open Directory Project hierarchically organize myriad Web sites so that spyware writers do not even need to maintain their own database of Web site categories for profiling.

Large-Scale Web Activity Monitoring

Spyware is not the only way to monitor users' online activity; in fact, advertising companies such as DoubleClick have been quite successful in tracking users without the use of spyware. Rather than place an unwanted program on a user's computer, Web advertisers work with Web sites to deploy links to small, transparent image files (1 × 1 pixel) hosted by the advertiser. These links are embedded in target pages on a Web site. When a browser downloads one of those pages from the target Web site, the transparent image is downloaded from the advertiser's server. When that happens, the advertiser is able to capture information about the browser downloading the image.

Changing Browser Settings

Another common activity for spyware is to change browser settings. Security settings and configurations are often targeted. Security settings in browsers control several aspects of browsing behavior, including:

- History tracking
- Saved form information
- Saved passwords
- Cookie controls (see Figure 4.7)
- Enabling and disabling Java, JavaScript, and ActiveX controls
- Allowing Web applications to install software on the local computer

Spyware may change feature settings that would not allow it to function, such as running ActiveX controls or other plug-ins. A common problem with spyware is that it changes browser settings, such as when homepage hijacking occurs (described earlier).

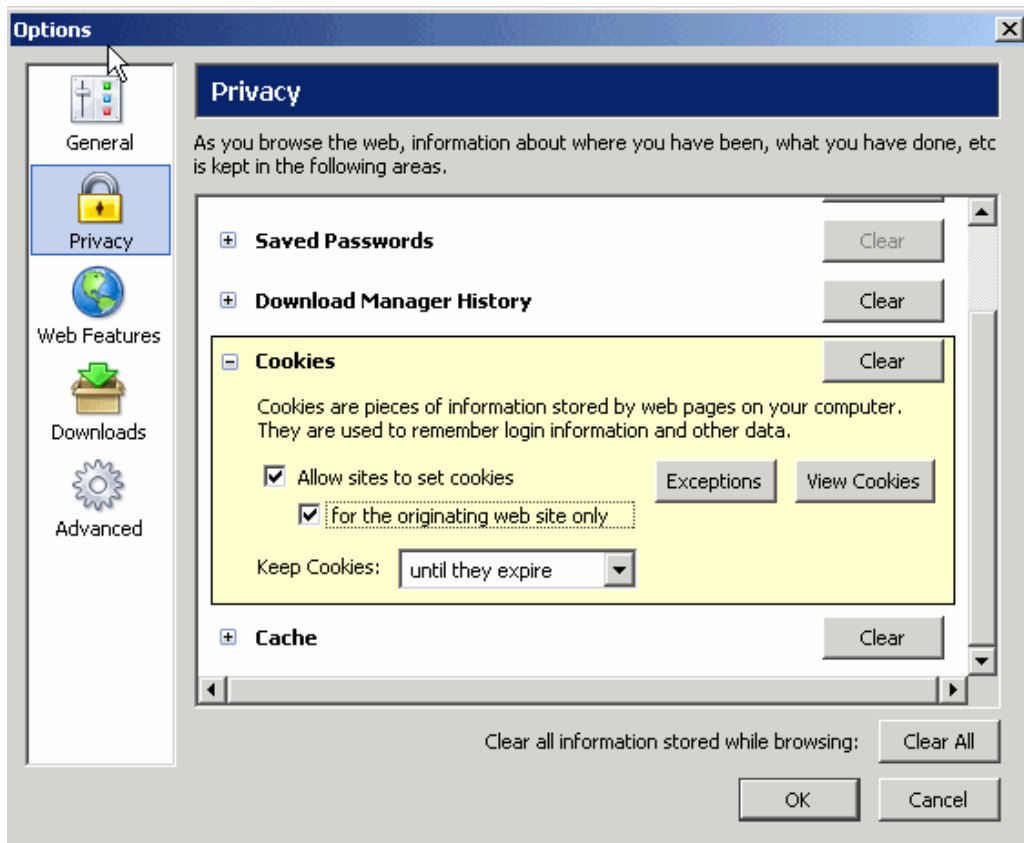



Figure 4.7: Firefox provides a mechanism to control the use of third-party cookies by allowing users to specify that cookies are allowed only for the originating Web site.

Installing Browser Help Objects

Browser Help Objects are plug-ins designed to extend the functionality of browsers. Like so many security vulnerabilities, this one started as a good idea, which has been exploited. Typical behaviors of Browser Help Objects include:

- Installing unwanted search assistant toolbars
- Displaying banner ads
- Redirecting homepages
- Displaying pop-up and pop-behind ads

In addition to exhibiting these unwanted behaviors, Browser Help Objects can be difficult to remove and sometime requires editing the Windows registry.

 It should be emphasized that not all Browser Help Objects are currently considered spyware. Google Toolbar and the Adobe Acrobat reader plug-in for IE are implemented as Browser Help Objects.

Malware vs. Malware

In addition to the other behaviors typical of PUPs, recent trends in malware behavior include commandeering other malware. Some examples noted by malware researchers at McAfee include:

- Adware that removes other adware
- Viruses that remove other viruses
- Blended threats that distribute spyware
- Bot armies commandeered by other malware
- Malware shutting down firewalls and antivirus software
- Malware changing browser security settings

Corporate and home computers are becoming something of prizes in what amounts to a cyber turf battle. Compromised computers are valuable resources and attackers are becoming more creative in their methods for capturing and maintaining at least some control over them.

Impact of Spyware

The impact of spyware on business operations falls into three general categories:

- Reduced computing performance
- Loss of proprietary and confidential information
- Help desk and recovery costs

Reduced Computing Performance

Hardware vendors promote the speed of their processors, data buses, disk drives, and just about every other component within a PC. This promotion is not surprising, when so many users are putting more demands on their systems. Applications are more complex and users are running more of them. It is not unusual for a user to have multiple programs open at once, including:

- Email
- Instant messaging
- Office applications, such as word processing and spreadsheets
- Online meeting/conferencing software
- Web browsers
- Database applications
- Image editing programs

Memory-intensive programs can fill RAM to capacity, which forces paging of data to disk, slowing response times. Computing-intensive programs, such as computer aided design, can consume available CPU cycles. Large downloads can consume bandwidth. The result is a sluggish system and an unhappy user.

Spyware, like other programs, will consume resources as well. Although writing data to files and copying files back to a central server can use some disk and bandwidth resources, one especially problematic practice of spyware is the way in which it monitors activities.

Performance Effects of Monitoring

Windows OSs provide the ability to trigger specialized processing when a particular event occurs, such as a key being typed or a window moved. (This action was discussed earlier with regard to keylogging.) These specialized triggers are known as hooks.

Hooks have legitimate uses but they slow processing even when designed carefully. For example, when a window is opened on the desktop, messages are sent among various components of the OS to create a window data structure and display it on the screen. Imagine if a spyware program inserted a hook to intercept window open messages so that every time a window was opened, the following tasks were executed:

- Determine the application opening the window.
- If the application is Outlook Excel or Word then
 - Record the time of the window open event in a log file
 - Capture a screenshot of the window
 - Save the screenshot to a file

This one program alone could slow a system by consuming memory and writing large amounts of data to the disk. This type of video frame grabbing is one way to steal proprietary information and conduct industrial espionage. In addition to the overhead of monitoring, the number of spyware applications on a computer will influence the overall impact.

Number of PUPs

The second factor in the performance impacts of spyware and other PUPs is the number of these programs running on a system. A single program may have negligible impact on performance, but when the number climbs to half a dozen or more, the drain on system resources can become noticeable.

Loss of Proprietary and Confidential Information

Another impact of spyware is the loss of proprietary or confidential information. Proprietary business information—such as sales plans, customer lists, and design documents—can be easily copied to a remote FTP server by spyware or other malware without the knowledge of the owner. Even when malware is detected, it may not be clear what information is compromised.

Another concern is the loss of personal information, especially Social Security numbers, account numbers, and other information that can be used for identity theft or online fraud. Online transactions are not always insured against fraud, so recovering lost funds can be challenging in the best of cases.

Know Your Online Agreements

Many people believe that if someone steals their credit card information and makes fraudulent charges or forges checks in their name, their liability is limited. Such is not always the case and it certainly is not necessarily the case with online transactions. *USA Today* reported two enlightening examples of online fraud: one involved a commercial bank account and the other a personal retirement account.

The first case involved a business owner who lost \$20,000 after a cyber criminal managed to transfer more than \$90,000 from the businessman's account into a third-party account. A cash withdrawal of \$20,000 was made the next day before the funds could be restored. It was later learned that a Trojan horse program called Coreflood had captured and transmitted the businessman's account name and password to the thief or an accomplice. The bank refuses to compensate for the loss claiming its safeguards were in place and since this was a business account, it is governed by the Uniform Commercial Code, which limits online service provider liability.

In the second case, a man discovered a thief was in the process of selling \$60,000 worth of stock from his online brokerage account. The man alerted the brokerage firm, which stopped the trade. The victim was later informed that the brokerage's actions were a "one-time courtesy" and that in the future he would be responsible for trades made from his account.

The moral of both stories is to know the details of your agreements with online service providers. And beware, consumer protections are not always available to businesses. For more information, see "Cyber Crooks Break into Online Accounts with Ease" at


http://www.usatoday.com/money/industries/technology/2005-11-02-cybercrime-online-accounts_x.htm.

Help Desk Costs

Help desks often bear the brunt of spyware's impact. According to a recent Computerworld survey:

- 83 percent of respondents reported desktop support and performance issues
- 50 percent of respondents reported increased Help desk activity due to spyware
- 79 percent of respondents reported spyware incidents significant enough to involve an IT department response
- When enterprise anti-spyware solutions were implemented in one company, the number of Help desk calls dropped by 30 percent
- In another company, the installation of anti-spyware software and Windows upgrades virtually eliminated spyware-related Help desk calls

As noted earlier, the computer manufacturer Dell reported that spyware accounted for as much as 20 percent of its Help desk calls in late 2004.

 For more information about the Computerworld spyware survey, see “Spy Stoppers Fight Back” at <http://www.computerworld.com/securitytopics/security/story/0,10801,105764,00.html>.

The proliferation of spyware has reached the point at which organizations are observing quantifiable impacts on their system performance as well as their Help desk costs.

Summary

PUPs such as adware and spyware clearly demonstrate the contrasting interests of Web users. On one side, you have PUP developers who realize the potential economic gain from targeted advertising, commandeering computing resources and network bandwidth, and artificially inflating page hits. On the other side, you have large numbers of corporate and home users whose computers are running multiple spyware applications that reduce their system performance, tamper with system configurations, hijack applications, and create vectors for potential information theft.

Like other malware, PUPs are best dealt with using a layered defense. Network devices, such as content-filtering appliances, in conjunction with desktop anti-spyware programs can minimize the impact of spyware. This topic will be addressed in further detail later in this guide; the next chapter will focus on another threat to Web users—phishing scams.

Chapter 5: Phishing and Identity Theft

Some of the most challenging security problems are based on people's behavior more than on device or application vulnerabilities. The term *phishing* has come into use to describe techniques for tricking individuals into disclosing confidential information, such as account numbers, Social Security numbers, or financial data. The practice of conning information and money is certainly not new, but like so many other operations, the Internet has changed how it is done. Email and bogus Web sites are now tools in the con men's toolboxes. With personal information in hand, criminals masquerade as the victim and withdraw money from bank accounts, sell investments, and transfer funds. Another troubling and increasing related problem is identity theft.

Identity theft occurs when a perpetrator uses a victim's identity for financial gain. Pretending to be someone else to secure loans, acquire telecommunications services, or apply for credit cards are common objectives. Identity thieves can get personal information in a number of ways, from sorting through trash looking for account statements, paycheck stubs, or other financial documents ("dumpster diving") to tricking the victim to reveal details through phishing scams.

This chapter will examine the nature of phishing scams with an emphasis on

- Anatomy of phishing scams
- Economics of phishing
- Countermeasures to phishing

In the course of the discussion, we will also explore how phishing is evolving in the all too common game of cat-and-mouse played by security professionals and cyber criminals. The chapter will also include a discussion of identity theft and its relation to phishing.

Anatomy of Phishing Scams

Phishing has evolved from a traditional social engineering operation to include malicious software for stealing credentials and network attacks to redirect traffic from legitimate sites. In general terms, the goal of phishing is to steal credentials. Three techniques are used:

- Tricking individuals to disclose information
- Deploying malware to capture user credentials and account information
- Attacking the Domain Name Services (DNS)

Each method presents a different set of challenges and requires particular countermeasures. As Figure 5.1 shows, these attacks target different parts of the computing infrastructure, including users, computers, and the network.

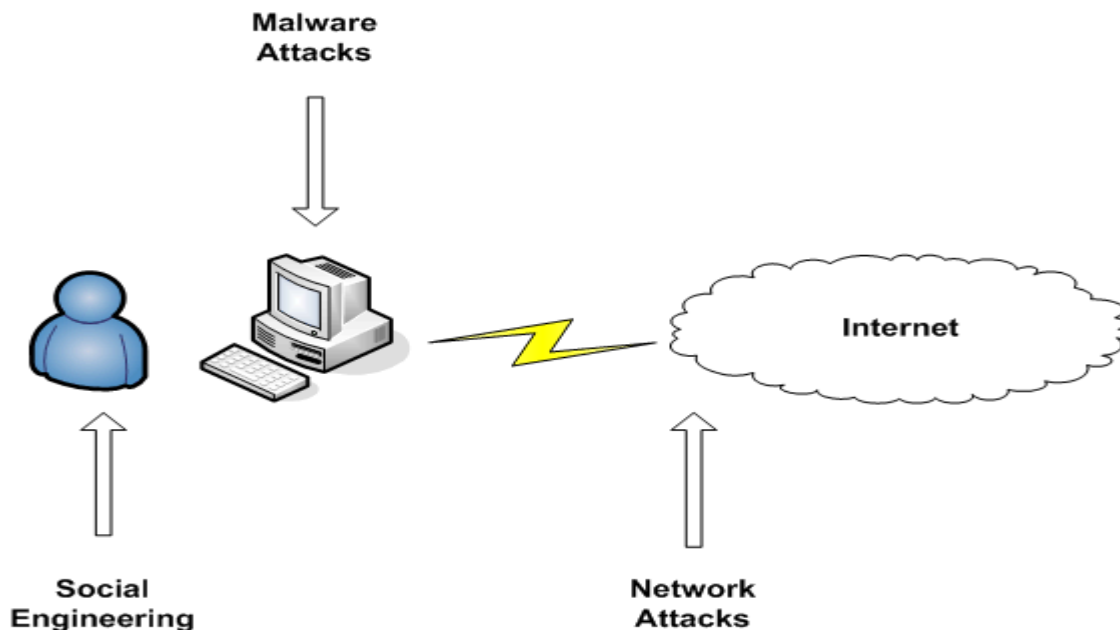


Figure 5.1: Phishing schemes use social engineering, malware, and network attacks to collect user information.

Social Engineering and Phishing

Social engineering is the practice of tricking individuals into disclosing confidential information or collaborating in a malicious act. There are many ways to perform social engineering:

- Direct questioning
- Incremental collection of information
- Using physical access
- Reverse social engineering
- Internet-based social engineering

Direct Questioning

The simplest and most direct phishing method is to simply ask people for information. Of course, virtually no one would give out the PIN number of their bank card to a stranger that walks up to them on the street and asks for it. There is not perceived trust in that situation; there is probably a fair amount of justified distrust. Consider another example.

An employee is working at her desk and receives a call from Bob Johnson in the Help desk support center. Bob introduces himself and goes on to say that there have been network problems and proceeds to ask a series of questions about network performance, access to shared drives, and any problems she might have with email. He then says he needs to verify that her account is correctly executing the latest login script so he'll need her username and password. At this point, Bob has probably built up some degree of trust with the employee and if the request seems reasonable, the employee might give out the information.

When employees and users are too savvy to fall for direct questioning other techniques are utilized.

Incremental Accumulation of Information

Directly asking for a username and password is too blatant in many circumstances; a better approach is to acquire information incrementally. For example, a perpetrator could call the human resources department to get the names of key personnel, such as the name of the CIO or head of PC support. From there, the perpetrator could call the office of the key person claiming to be from a software vendor checking on licenses. In the process, the caller can acquire the names of key pieces of software used in the organization. This process can continue and although the perpetrator may never get a username and password, they may get enough information to leverage other techniques, such as gaining physical access.

Physical Access


Gaining physical access to a site can open a number of opportunities for someone conducting social engineering; it enables the attacker to:

- Watch users as they type usernames and passwords
- Claim to be from another office with the need to “borrow” a PC to check email
- Pretend to be from a vendor and on-site to correct a configuration problem with an application
- Scan wireless networks for unencrypted or weakly encrypted information

Physical or logical access to a network can enable another form of social engineering, known as reverse social engineering.

Reverse Social Engineering

Reverse social engineering plays on our natural inclination to appreciate help from others. In this con, the perpetrator disrupts operations on a network, server, or desktop (which implies that the person already had some logical or physical access to the systems). The perpetrator then arrives claiming to understand the problem and have the solution (which of course, he or she does). When the problem is fixed, the perpetrator is the hero of the day and users are inclined to reciprocate or engage in conversation that would then lead to a disclosure.

 Social engineering does not change with technology trends. See, for example Ira S. Winkler and Brian Dealy's 1995 paper "Information Security Technology? ... Don't Rely on It: A Case Study in Social Engineering" for a discussion of simple and rapid techniques used to acquire access information in a number of financial services companies. The paper is available at http://www.usenix.org/publications/library/proceedings/security95/full_papers/winkler.ps.

Internet-Based Social Engineering

Phishing, Internet-based social engineering, utilizes emails and bogus Web sites to lure victims. Like other forms of social engineering, this technique depends on establishing trust and then convincing the victim to share information. Phishing scams based on social engineering use a two-part scam: the lure and the trap.

The lure in phishing is usually an email message that appears to be a request for some information or action from a legitimate business. eBay, PayPal, and banks are commonly portrayed as the senders. To get people's attention, phishers use subject lines that often raise users' concern. The following list highlights example messages lines (Source: the Anti-Phishing Working Group):

- Update Your Wells Fargo Access Online Information
- New email address added to your eBay account
- Credit Union Services: update your account records
- Credit Card Declined Notice
- Receipt of Your Payment to DELL

In general, phishers depend on three types of lures:

- Those based on fear
- Those that promise rewards
- Those that work with a victim's expectation of legitimate email

The fear-based phishing scams are probably the most prevalent judging from examples found in phishing databases, such as the Anti-Phishing Working Group's Archive.

 The Anti-Phishing Working Group's Archive is available at http://www.antiphishing.org/phishing_archive.html.

Lure Number 1: Fear

Phishers have taken an approach sometimes seen in technology sales—the appeal to fear, uncertainty, and doubt (FUD). In fear-oriented phishing scams, the subject line needs to generate enough concern to get the victim to read through message as the message tries to establish trust through a combination of visual cues and text.

For example, one scam purporting to be from eBay uses the eBay logo and links to eBay services such as search, discussion boards, and help. The message headline states that the user's credit/debit card must be updated immediately followed by an ironic warning that the recipient's eBay account has recently been accessed from a foreign IP address. It goes on to provide an IP address from which the access attempt occurred along with the ISP host. The message even advises the reader to "Please save this fraud alert ID for your reference." The cumulative effect of well-placed visual cues, such as logos, and well-written, official-looking text can be effective enough to get readers to take the next step, and click through to a bogus Web site, then provide their identifying information.

Lure Number 2: Something for (Virtually) Nothing

The appeal of something for nothing, or very little, is bound to get someone's attention. Phishing scams have used the allure of prizes, gift cards, and other rewards to entice readers to click through to phishing sites. Just as some phishing scams can be very sophisticated and likely to catch even some wary readers, other scams are so poorly done it is difficult to imagine the scheme is very effective (see Figure 5.2).



Figure 5.2: An example of both a “something-for-nothing” lure and a poorly crafted phishing message.

The subject line “Target Customer #500849687” is not particularly promising, but when the reader does open the message, he or she finds a simple one-line enticement to receive a \$500 shopping card just for visiting a Web site. Given that this message is purportedly from Target, it is hard to imagine why the company would use their competitor's name in the URL listed in the message, <http://www.redistrainer.com/walmart>.

The target URL listed in the bottom of the browser window shows what looks like a generated URL. This URL could be unique to the this message so that the phisher can link a click-through back to this email address even if the reader does not fill in any information at the bogus Web site.

Another telltale sign of mass generated phishing messages is that the basic form is repeated but the phisher impersonates a different business. Figure 5.3 shows another phishing message received less than 90 minutes earlier by the same recipient. The content is the same except the retailer's name has changed and the URL no longer mentions a competitor.

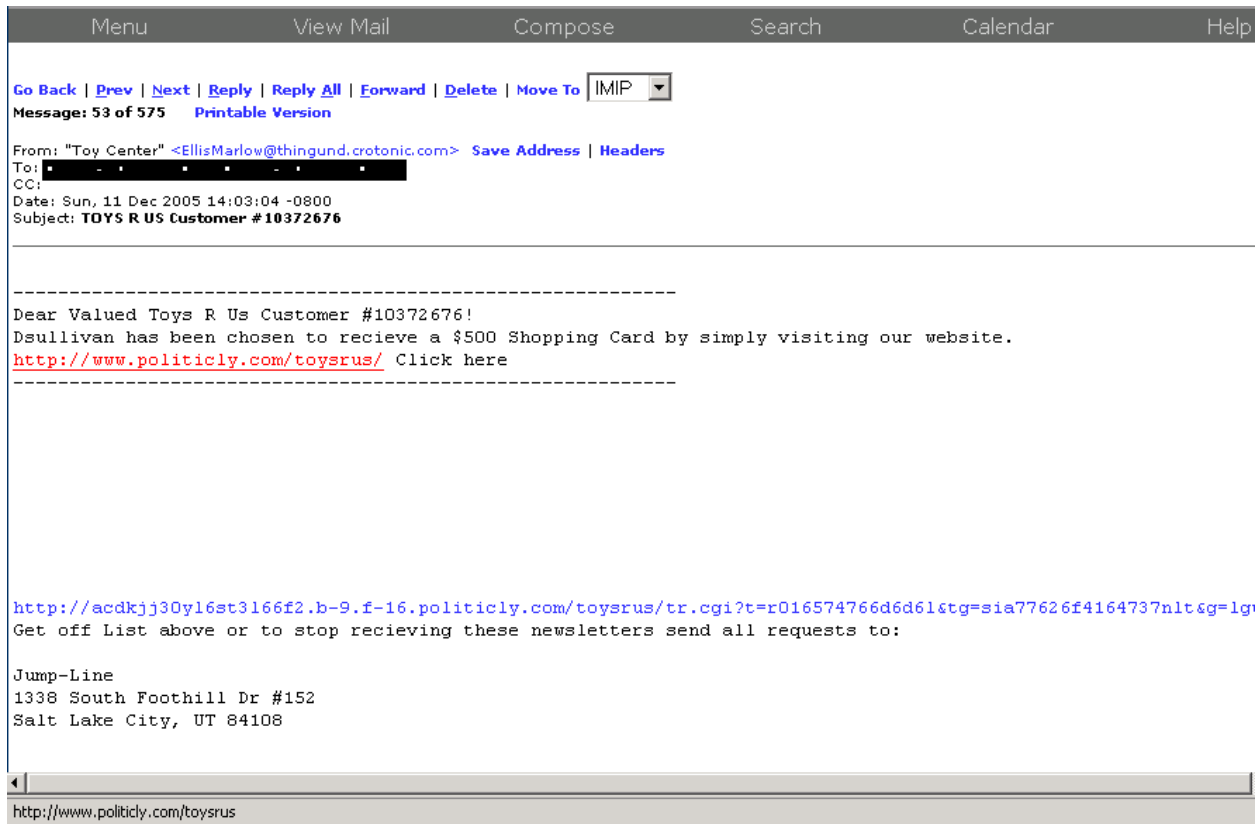


Figure 5.3: Mass-produced phishing messages with little variations are easy to spot.

These phishing examples represent one end of the phishing spectrum: mass generation, multiple versions, poorly edited, few if any graphics to provide visual indications of a legitimate site. As with spam, the cost of creating these phishing messages is so low, a positive return on investment can be realized if only an extremely small number of readers fall for the lure. At the other end of the spectrum are carefully crafted messages that take into account the readers' expectations for legitimate email.

Lure Number 3: Expectations of Legitimate Email

A more difficult-to-prevent form of phishing is known as context-aware phishing. (This discussion is based on the work on Markus Jakobsson of Indiana University.) This form of phishing is targeted to a specific audience, such as eBay customers that bid on particular types of products, unlike the mass emailed fear- and enticement-style scams. The messages are designed to appear to be an expected message, such as an announcement that the recipient is a winner of an eBay auction. This format increases the likelihood that a reader will respond to the message.

Context-aware phishing is a three step process:

- Identity linking
- Victim selection
- Context-aware password phishing

In the first step, identity linking, the phisher links identity information of a user, such as an eBay user name, with public identity information, such as an email address. There are three forms of identity linking:

- Inside-out linking, which starts with identity information used inside an application and links it to outside information
- Outside-in linking, which starts with external information (for example, email) and links to internal information
- Epidemic linking, in which accounts compromised by one of the other two accounts are exploited to find other username/email pairs

In some cases, linking is trivial because application names are the same as the user's email address. In other cases, the way an application works is exploited to get inside or outside identities. For example, a bidder on eBay can request the email address of a seller after making a bid for an item. When questions are asked of eBay users, if they do not respond using the anonymous response method provided, phishers are given users' email addresses.

If a phisher has an email address but needs a user account, the phisher could send a bogus message purportedly from eBay requesting that users provide their eBay user IDs. The message might specifically say a password is not needed and in fact should never be shared with anyone.

Once the phisher has a set of email addresses and application identifiers, he or she can move on to victim selection. During the victim selection step, the phisher will select application users who are plausible targets. For example, the phisher could monitor eBay items to determine the highest bidder at any time and select all users listed at any time. This step requires knowledge of the target application, such as an eBay auction, and the ability to cull specific information from such data-rich systems. One of the advantages of services, such as eBay, is that they build the trust of their users by sharing so much information about the histories and evaluations of buyers and sellers. It is, however, also one of the aspects of such sites that phishers can exploit.

The final stage is the password phishing step. Once the victims have been identified, they are sent an email in response to their bids, such as a congratulation message saying they have won an auction. The email will contain a link to a phisher-controlled site that looks like the real eBay site and the user will be prompted for his or her user ID and password. Since the user is expecting, or at least not surprised, to find a message from eBay, this type of attack has a higher probability of succeeding than cruder phishing attacks.

 For more information about context-aware phishing and methods of prevention see Markus Jakobsson's "Modeling and Preventing Phishing Attacks" at http://www.informatics.indiana.edu/markus/papers/phishing_jakobsson.pdf.

The phishing lure is designed to get victims to click through to a phisher-controlled site. Phishers will use fear and enticement as well as more sophisticated attack methods. Some phishing schemes are so poorly executed they are easy to spot; others, like context-aware phishing, are challenging even for the cautious recipients. For whatever reason, the recipient clicks through; once they reach the bogus Web site controlled by the phisher, they are operating within the phishing trap.

Phishing Trap

The phishing trap is the Web site that appears legitimate but is controlled by the phisher. This trap is where the phisher collects victim's information. Once in hand, the phisher can then go on to use the victim's credentials to gain access to the victim's accounts, funds, and other resources. The full phishing process is depicted in Figure 5.4.

Once at the site, there may be additional cues to indicate the site is not legitimate. These can include:

- Suspicious-looking URLs
- Use of HTTP instead of HTTPS protocols
- Text with grammatical errors
- Asking too many personal questions for a simple update lure

Phishers use a variety of techniques to mask their sites, such as using IP addresses rather than domain names or using punctuation in a URL to create one similar to a legitimate Web site but still distinct. Hexadecimal encoding can also be used to replace literal characters with their numeric encoding. In still other more-difficult-to-detect cases that have been demonstrated but not necessarily used yet, character sets can be mixed so that most letters in a URL are in ASCII while Cyrillic characters are used for letters similar in both character sets (such as a, c, p, and t).

Another indication of an illegitimate site is the use of the non-secure HTTP protocol. Few if any legitimate businesses ask customers to provide confidential information, such as account numbers, without using Secure HTTP protocol (SSL encryption). Secure Web sites have URLs that begin with `https://` rather than `http://`.

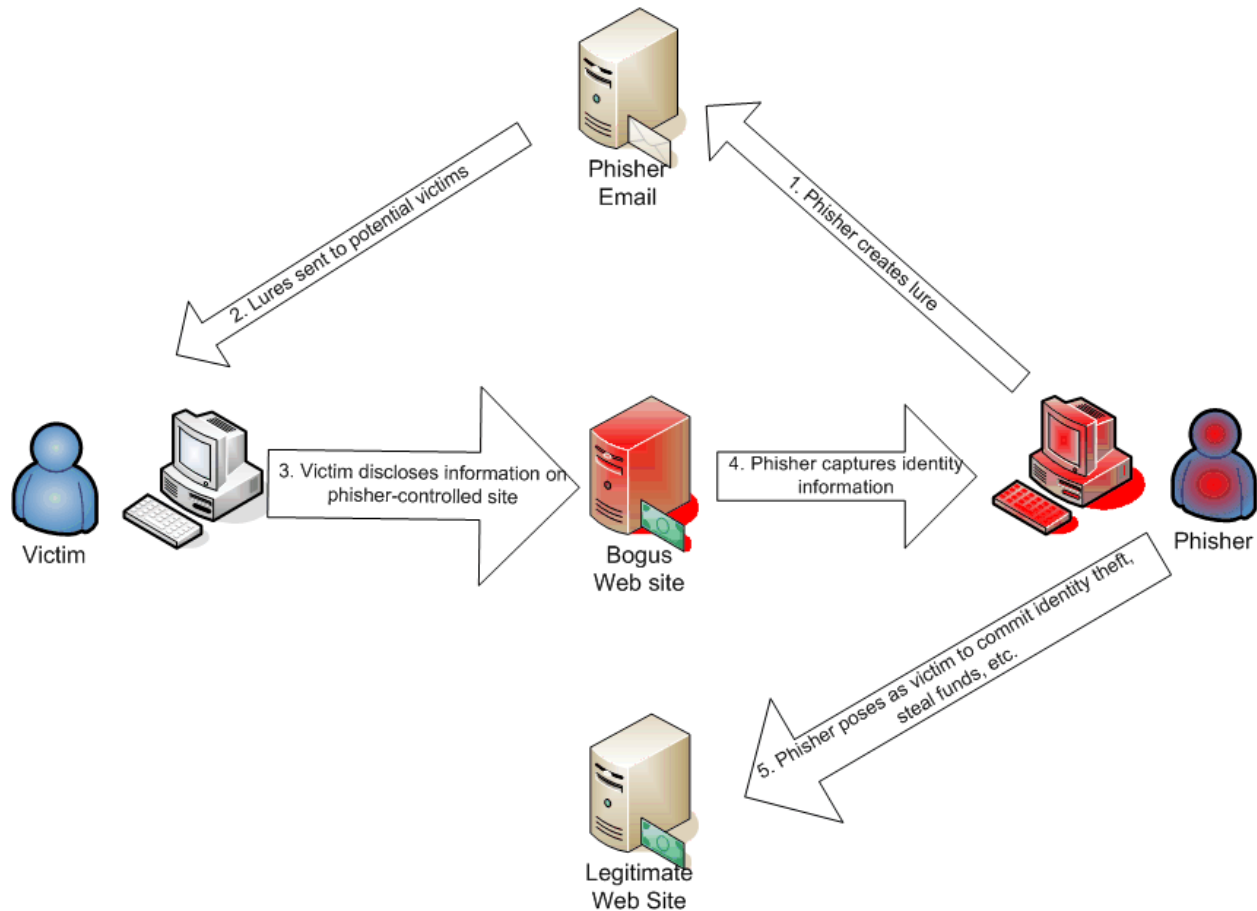


Figure 5.4: Phishing is a multi-step attack that, when successful, results in the attacker gaining access to the victim's identity, funds, and other resources.

Other indications of bogus sites are ungrammatical text and poor editing. When large numbers of bogus sites are put up at once, the quality of the content may reflect that speed. The email examples mentioned earlier that referenced Wal-Mart in an email purportedly from Target demonstrates that this lack of quality can happen even in the lure.

Limits of Social Engineering-Based Phishing

Social engineering-based phishing is becoming more sophisticated with techniques such as context-aware phishing, but awareness of the problem will limit the effectiveness of less-sophisticated approaches. Some common suggestions to prevent phishing attacks are becoming better known:

- Do not open emails from unknown senders
- Do not click on links in emails
- Type in URLs instead of clicking through from an email
- Examine URLs more closely for variations in business names or suspicious-looking addresses

As users become more careful, phishers are turning to technology-based methods to acquire personal information—especially the use of malware.

Malware-Based Phishing Techniques

In addition to traditional social engineering approaches, some phishers are turning to malware to collect personal information. If we were to argue about definitions, this type of malware might be considered spyware and not a phishing attack. However, some researchers, such as Jason Milletary of the CERT Coordination Center, argue that some spyware should be dealt with as phishing attacks. Milletary makes three points:

- Phishers/spyware developers often use some type of social engineering to trick users into downloading the malicious software onto users' systems.
- Common techniques and tools are used to distribute phishing emails and malware.
- Countermeasures deployed to stem threats still leave other avenues of attack, so it is important to understand and address the broad range of techniques that criminals have at their disposal.

 For more information about this topic, see Jason Milletary's "Technical Trends in Phishing Attacks" at http://www.cert.org/archive/pdf/Phishing_trends.pdf.

This gray area between phishing attacks and spyware includes three sub-areas that require distinct countermeasures:

- Use of botnets
- Malicious software installed on victims' machines
- Exploitation of browser vulnerabilities

These techniques are used in other information security threats, such as viruses, worms, and spam. Fortunately for the rest of us, that means the countermeasures deployed to combat one type of threat may address others at the same time.

Botnets and Phishing Attacks

Botnets are sets of compromised computers that can be controlled from a single source (see Figure 5.5). Usually, a malicious piece of software is downloaded to a victim's computer, performs basic installation steps, then communicates with or waits for communication from a command system. The infected machine (the "bot") and command system can communicate over instant messaging, Internet chat protocols, peer-to-peer protocols, or other protocols.

Botnets are used as distributed computing platforms for performing tasks for the controller of the botnet. Some common uses are:

- Distributing spam
- Launching social-engineering-based phishing attacks
- Launching Distributed Denial of Service attacks (DDOS)
- Deploying additional malware, such as Trojan horses, to collect passwords and account information

In one of the more sophisticated botnet attacks to date, a group of three bot programs worked in conjunction to infect a large number of systems. The attack began with a malware program named Glieder-AK, which infects systems and opens backdoors for other programs to exploit. The second step in the attack came from the Fantibag Trojan, which disabled security features and prevented compromised machines from contacting antivirus vendors for updates. The attacks concluded when Mitglieder was downloaded, which opened further backdoors and left the system under the control of the botnet.

 The multi-Trojan attack is discussed in more detail in *The Register* "Hackers Plot to Create Massive Botnet" at http://www.theregister.co.uk/2005/06/03/malware_blitz/.

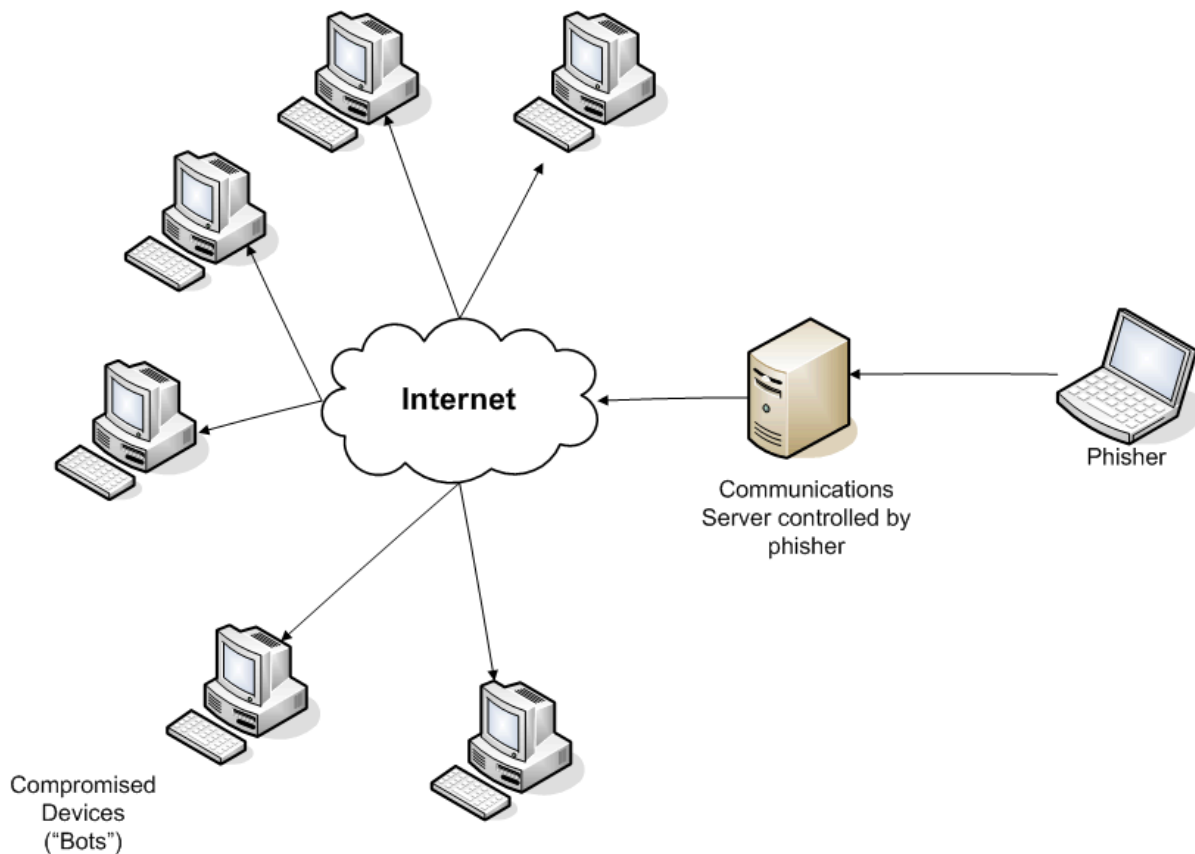


Figure 5.5: Botnets are computers infected with malware that provides for remote control of the compromised host.

Botnets are tools of cyber criminals; the botnet itself is not typically the end goal, rather they are platforms for deploying and controlling other malware.

Malicious Software for Phishing

The goal of the phisher is to collect information about a user that can be exploited for financial gain. Several kinds of malware are in use for this purpose; in general, these programs are referred to as Trojan horses or Trojans for short.


A Trojan is a program that appears to be for one purpose but is actually carrying out a malicious task instead of or in addition to its purported purpose. The malicious tasks include:

- Monitoring keystrokes
- Copying video frames
- Changing security settings
- Harvesting information from caches and other data sources

Keystroke monitoring uses a program or hardware device to capture each keystroke as it is typed. This technique has many legitimate uses, from evaluating human-computer interactions and testing software to law enforcement. It is also a well-known method for capturing user IDs, account numbers, and passwords.

One way to avoid using the keyboard to enter passwords is to use a virtual keyboard displayed on the screen. Users then mouse over characters on the virtual keyboard and click to enter the character. This method effectively avoids the potential for keyboard monitors to capture the confidential information. As you would expect, when security professionals and systems developers deploy a countermeasure, another avenue of attack is opened by hackers. In this case, it is the use of video-frame grabbers.

The purpose of a video frame grabber is to make a copy of the image that appears on a computer monitor at any time (such as when a mouse is clicked). Malware can use features of the operating system (OS) to intercept messages from the mouse or keyboard to detect a particular type of event and then take some action, such as copy the contents of the video buffer.

 The Windows Win32 API implements a number of hooks for programmers to intercept messages between objects, such as windows, the mouse, and the keyboard. For more information, see “Win32 Hooks” at http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnwui/html/msdn_hooks32.asp.

Trojans used in phishing and spyware attacks may also change security settings on the infected system. For example, if a malware program infects a computer and executes in the context of an account with administrative privileges, it can change registry and browser settings that define access controls. (See Figure 5.6 for examples of Microsoft Internet Explorer—IE—security settings and their options.)

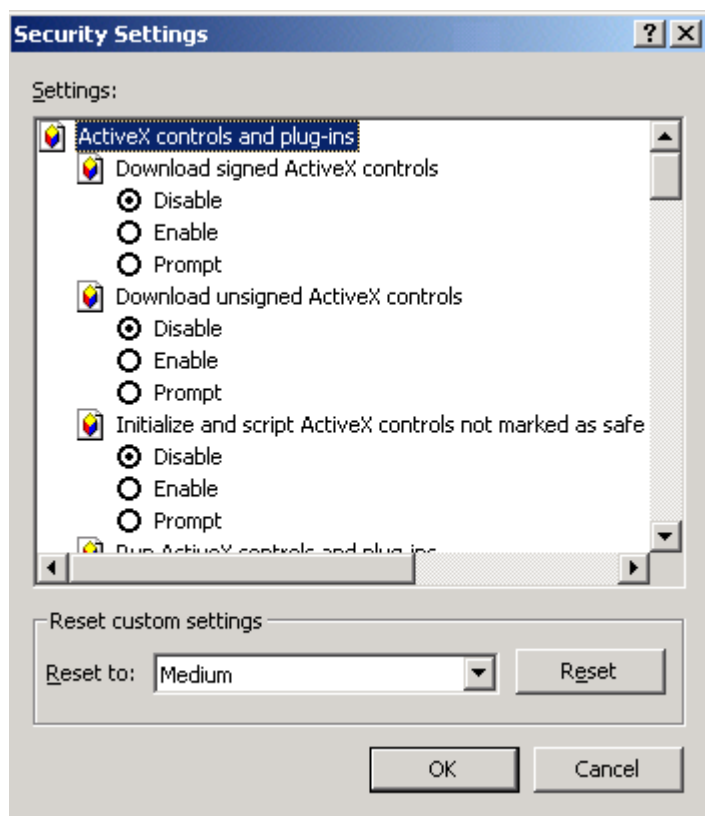


Figure 5.6: Once a computer is infected, malware can further reduce the security of the compromised machine by changing security settings, such as IE settings.

Once malware is on a system, it can capture information that has been left after its legitimate use has finished. For example, browser caches improve performance and ease of use by maintaining recently used information. Trojans can use this information to capture identifying information such as email accounts, user names, and recently visited Web sites.

This information alone may not be enough to compromise the user's financial accounts, but it can provide seed information for a successful context-aware phishing attack. For example, if a user recently made a hotel reservation, a phisher could send a bogus email stating that an error occurred in processing the reservation and the user needs to re-enter his or her credit card information.

Malicious software provides a number of methods for phishers to conduct their attacks. Like social engineering-based phishing, malware phishing can be used to incrementally collect information that is combined with details captured from multiple methods. A single piece of malware does not have to implement a full-blown phishing attack to be effective. This fact is made apparent when considering the impact of browser vulnerabilities.

Exploiting Browser Vulnerabilities

Several vulnerabilities in popular Web browsers, particularly Microsoft IE and Mozilla Firefox, provide avenues for phishing attacks or at least help to obfuscate the instances of attacks. Past vulnerabilities have included (these have all been fixed with patches or in new versions):

- A bug in Mozilla Firefox truncates the status bar display when the mouse is moved over it and the href referenced in the URL contains a %00. (For details, see https://bugzilla.mozilla.org/show_bug.cgi?id=228176.)
- A bug in Mozilla Firefox allows a malicious page to appear encrypted with SSL and present the certificate of another site. (For details, see https://bugzilla.mozilla.org/show_bug.cgi?id=240053.)
- A bug in IE causes requests to some objects to be mishandled, which allows attackers to execute arbitrary code with the privileges of the user running IE. (For details, see <http://www.kb.cert.org/vuls/id/887861>.)
- A bug in IE DHTML allows attackers to exploit an ActiveX control that can download malicious code, read cookies, and change the content of Web pages. (For details, see <http://www.kb.cert.org/vuls/id/356600>.)

Although fixes are available for these vulnerabilities, any unpatched versions would still be vulnerable. It is important to note that vulnerabilities range from relatively low-impact problems, such as the bug in Mozilla Firefox that truncates a status bar display, to severe vulnerabilities that can allow arbitrary code to execute with the privileges of the user logged in to the system. (This is enough to give pause to systems administrators who surf the Web from privileged accounts).

One should also note that these are just examples of browser vulnerabilities that are known to exist. Others exist and have been patched, and we are likely to find others in the future. As the number of features in browsers increase, so do the opportunities for attackers to exploit those features.

Malware-based phishing techniques range from sophisticated networks of bots that can launch mass phishing and spam email campaigns to Trojans targeted to gathering information from a single device to Web browser components that exploit vulnerabilities in browsers to conduct phishing attacks. Both social engineering attacks and malware phishing attacks present more than enough challenges to systems administrators and security managers; unfortunately, there is still one more category of phishing attacks they must address.

DNS Attacks

The last form of phishing attack addressed in this chapter targets the infrastructure of the Internet. DNS is a protocol used to map easy-to-remember names of sites, such as McAfee.com, to the IP address of those sites, such as 216.49.81.129. Resolving an IP address from a domain name takes several steps (see Figure 5.7):

- A client requests the location of the server that has the address mapping of a site, such as MySite.com.
- A name receives the request and responds with the address of the domain's primary and secondary domain name server.
- If available, the primary domain server returns the IP address of the domain to the client making the request; otherwise, the secondary domain sends the information.
- Once the browser has the IP address, it requests the Web resource (for example, an HTML page) from the target site.
- The target site responds to the request by sending the resource to the client.

The relationship depends on implied trust. If the name server does not point to the correct primary and secondary domain servers, the client won't get the proper IP address. If the primary or secondary domains servers have inaccurate information, then again, the client will not find the site they are looking for. In the case of phishing attacks, the result is worse: the client is redirected to a bogus site.

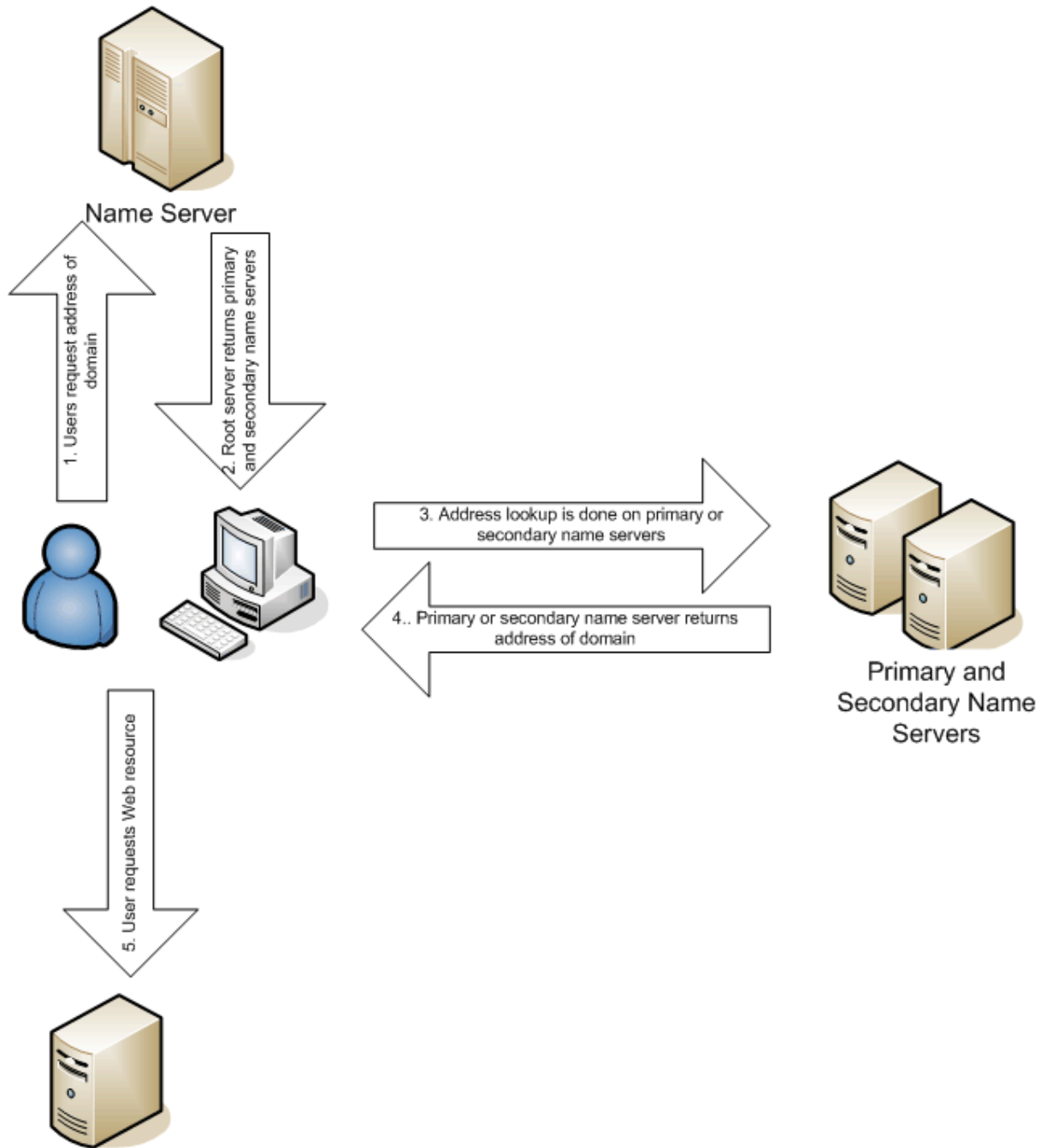


Figure 5.7: DNS is compromised when inaccurate information is loaded into name servers.

Corrupting DNS is known as *DNS poisoning* and is a long-known and understood problem. It is also known as *pharming* in the context of phishing attacks. For the phisher, there are definite advantages to DNS poisoning over other methods. Most importantly, large numbers of potential victims can be trapped by corrupting a single server rather than luring victims through emails or infecting client devices with malware.

Another potential problem with DNS is the use of wildcards in DNS entries. Originally intended to manage mistyped URLs, wildcards have been exploited to lure victims to bogus Web sites.

 The case of a DNS wildcard phishing scam against Barclay's bank in the UK is described in *InformationWeek* "Phishers Turn to DNS Wildcards, Cache Poisoning" at <http://informationweek.com/story/showArticle.jhtml?articleID=60407745>.

Both DNS poisoning and wildcard vulnerabilities stem from the underlying trust assumed in the DNS protocol. A new protocol, DNS Security Extension (DNSSEC), uses digital certificates to authenticate parties involved in the exchange of DNS information. However, servers that do not use this protocol are still vulnerable to some DNS poisoning attacks. Some DNS servers, such as Berkeley Internet Name Domain (BIND), have added countermeasures that operate with the DNS protocol, such as ignoring messages unrelated to a query.

 For more information about DNSSEC, see <http://www.dnssec.net/>. For details about BIND security, see <http://www.isc.org/index.pl?sw/bind/>.

There are limits to social engineering and malware-based attacks and network attacks, such as DNS poisoning, are additional tools for the phisher. These attacks also demonstrate the overlap in techniques used by virus writers, spammers, phishers, and cyber criminals. Regardless of the technique, the objectives of phishers are the same: collect personal and confidential information that can be used for identity theft and financial gain.

Economics of Phishing and Identity Theft

The economics of phishing and identity theft can be considered from the perspective of both the perpetrator and the victim. The economic benefit of phishing to the perpetrators is apparent from recent trends in the number of phishing attacks. Figure 5.8 provides recent statistics about phishing attacks. Like spamming, phishing attacks are low-cost endeavors, so even extremely low response rates can make the effort worth the expense.

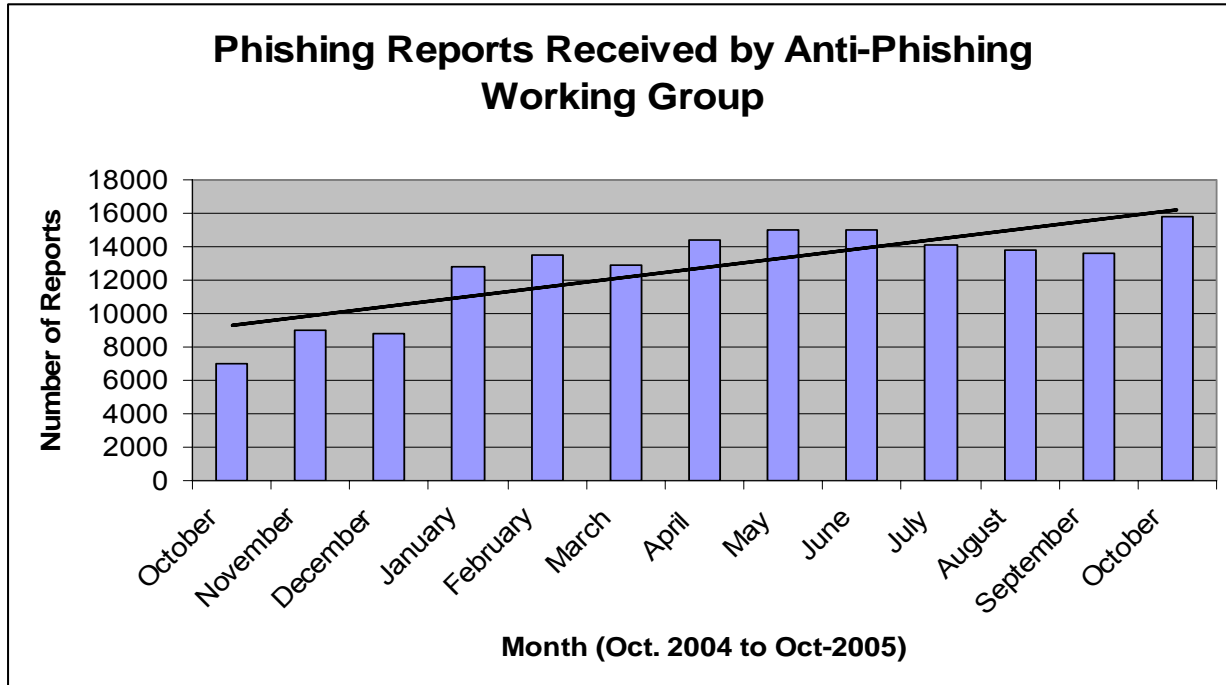


Figure 5.8: The number of distinct phishing reports received by the Anti-Phishing Working Group shows a trend upwards (Source: Phishing Activities Trend Report, October 2005, http://antiphishing.org/apwg_phishing_activity_report_oct_05.pdf).

Another measure of phishing activity is the number of brands that are highjacked per month (see Figure 5.9). Although useful, this measure may be under reported. Analysts have noted a trend to target smaller businesses and to distribute smaller numbers of emails. This phishing method is known as spear phishing. The likely objective is to “fly under the radar” and remain undetected. There is no way of measuring the success of those attempts.

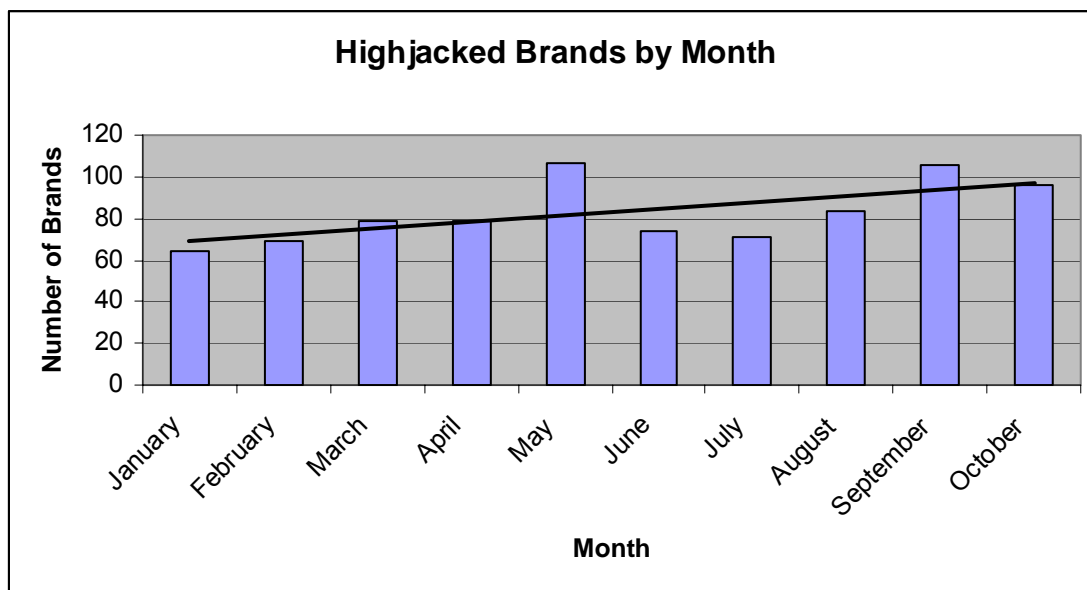



Figure 5.9: Tracking the number of highjacked brands is another measure of phishing activity.

In addition to those creating and launching the attacks, there are attackers who earn money renting their botnets for attacks. Attackers earn income by launching DDoS attacks. According to *New Scientist*, attackers with control of botnets are charging between \$500 and \$1500 per attack or \$1 to \$40 per bot for a smaller network (Source: Celeste Bieber “How Zombie Networks Fuel Cybercrime” at <http://www.newscientist.com/article.ns?id=dn6616>).

The cost to individuals and businesses is difficult to calculate, and the estimates vary widely. One reported estimate from *The Industry Standard* puts the cost of phishing as high as \$1.2 billion dollars; others estimate it to be as low as \$137 million.


 For more information about the estimate of phishing costs, see Linda Rosencrance “TECF Aims to Fight Online Fraud” at <http://www.thestandard.com/article.php?story=20040617173836939> and Greg Goth’s “Phishing Attacks Rising but Dollar Losses Down” at <http://csdl2.computer.org/comp/mags/sp/2005/01/j1008.pdf>.

Impact on Consumers

Those that are victims of phishing attacks risk identity theft and theft of funds. In addition, recovering from identity theft can take years. According to one study by the Identity Theft Resource Center (ITRC), victims spent on average 600 hours over a period of years recovering. The same study found that victims continue to deal with the effects of identity theft even after the initial recovery period. The impacts include:

- Increased insurance or credit card fees
- Inability to find a job
- Need to pay higher interest rates
- Difficulties with collection agencies

Although difficult to estimate, the study concluded that businesses lose between \$40,000 and \$92,000 per stolen identity.

 These and other statistics on identity theft are available at <http://www.idtheftcenter.org/facts.shtml>.

Phishing and identity theft are just two forms of fraud that businesses have to address. Citing research from Gartner, the Bank Administration Institute (BAI) reports both an increase in identity theft and a shift in the tactics of phishers:

- Rather than focusing on widely recognized, large financial institutions such as Bank of America, phishers are targeting smaller community banks.
- Thieves are using business accounts, rather than consumer accounts, because screening for the former accounts is not as effective as consumer account screening.
- Organized crime is finding identity theft a lucrative and less risky operation than some traditional areas of organized crime.

 The source of for these points is Clint Swift and Karen Epper Hoffman “Fraud Looms Large” at BAI Online at <http://www.bai.org/bankingstrategies/2004-jul-aug/fraud/>.

The impacts extend beyond the cost of losses due to fraud and the expense of countermeasures deployed by businesses and individuals. Fear of phishing will likely curtail the use of online services to the detriment of both the consumer and businesses. Consumers lose flexibility while businesses will incur higher costs as those services shift to higher cost operations such as call centers.

Phishing and Identity Theft Countermeasures

Countermeasures are available to both businesses and individuals to help minimize the problems of phishing and identity theft.

Business Countermeasures

Businesses can implement a number of countermeasures to the threats of phishing, both to prevent the use of their business identity in a phishing scam and to prevent their information infrastructure from being used by phishers.

In the first case, business logos and Web copy are used to trick individuals into believing a bogus site is actually associated with a legitimate institution. Phishers use a number of tricks, including linking to images and linking to parts of the legitimate site, such as a help or FAQ page. Systems administrators should monitor network traffic for suspicious activity. For example, a business could identify a phishing attack in progress if it detects a spike in traffic to download a particular JPEG.

In other cases, desktops and servers within an organization's network could be compromised and become part of a botnet. Regular use of desktop antivirus and inbound content filtering at the gateway can minimize the likelihood of this occurrence. If a machine were compromised, content filtering on outbound traffic and firewalls could detect and block suspicious activity, such as communications with an IRC chat room. Fortunately, the countermeasures in place for antivirus, anti-spam, and content filtering are capable of countering many of the tasks involved in phishing attacks.

Consumer Countermeasures

The first step individuals should take is to become better educated about identity theft, phishing, and related threats to online business. The United States Federal Trade Commission (FTC) provides the following advice to consumers:

- Do not reply to pop-up ads that ask for personal information or click through on links in pop-up ads.
- Use antivirus software and personal firewalls to prevent malware infections and to control communications between a PC and a phisher's system.
- Do not use email to transmit financial information.
- Review credit card and bank statements for suspicious charges.
- Exercise caution when opening email attachments, regardless of the sender.
- Report phishing scams to the FTC at spam@uce.gov and to the business or organization presented in the phishing email.
- Victims of identity theft should file a complaint with the FTC at <http://www.consumer.gov/idtheft>.



Individuals should also understand their online agreements with financial service providers—not all accounts are insured against fraud. For example, funds lost due to bank fraud against a personal checking account are generally restored by the bank. The regulations that protect consumer accounts do not necessarily apply to business accounts. Similarly, investment accounts and their financial instruments may not be insured and the investor may bear the risk of cyber crime. See some relevant examples at http://www.usatoday.com/money/industries/technology/2005-11-02-cybercrime-online-accounts_x.htm.

Summary


Phishing and identity theft are one of several threats to online commerce. Confidence schemes are not new, and banks have had to deal with fraud for as long as they have existed. The mechanics of these criminal activities is changing and requires new methods to address them.

Phishers are resorting to a combination of traditional social engineering techniques coupled with mass emails to reach a wide audience. As users become more aware of the phishing problem and educated about social engineering techniques, phishers have become more resourceful. They now use malware and botnets, similar to those used by attackers and spammers, to lure victims and collect account and password information without user interaction. More sophisticated social engineering techniques, such as context-aware phishing, and more stealthy operations, such as spear phishing, make these activities more difficult to detect.

Countermeasures are available to reduce the threat of phishing. Many of the information security measures already deployed in organizations—such as antivirus software, firewalls, and content filtering—are readily leveraged to combat phishing attacks and the potential use of an organization's systems in phishing scams. The driver behind phishing of course is economic gain. The low cost of launching phishing attacks, the relative low risk, at least when compared with other crimes, and the high returns imply this problem will not be fading away.

Chapter 6 Spam in the Enterprise

Spam, or unwanted and unsolicited email, in the enterprise unnecessarily taxes IT resources. Unlike its kin, phishing scams, spam itself is not a direct threat to security; rather the damage it causes is the result of the fact that it consumes network bandwidth and storage as well as wastes employees' time. As part of broader compliance initiatives, companies may be required to archive all email messages for extended periods of time, so even if spam is deleted by end users, it could continue to consume storage for years to come.

 For more information about phishing, see Chapter 5.

This chapter begins by examining the basic operations of mass emailing and discussing how spammers exploit weaknesses in email protocols. Next, it addresses the economics of spam and the attempts to control spam through legislation. Although helpful, legislation has not stopped spam and likely will not. Technology is therefore crucial to managing spam. This chapter includes a review of spam management techniques and concludes with some guidelines for evaluating anti-spam systems.

Email Operations and Spam Techniques

When unsolicited mass mailings and newsgroup postings began, spammers sent messages the way any other user would send messages. There was no attempt to hide its source or pretend to be something other than an advertisement. By the early 1990s, spammers began using programs to automatically post messages to multiple Usenet newsgroups. Today, legislation, such as the CAN-SPAM Act in the United States, attempts to curb spam and has prompted spammers to use techniques that hide their identities and the origins of their messages. These techniques depend on exploiting a combination of vulnerabilities in email protocols and insufficiently protected computers on the Internet. The following discussion examines early examples of spam to demonstrate that this phenomenon is not new.

Early Spam and Reactions to It

Spam is so prevalent today it is difficult to imagine an email user becoming upset about receiving an unsolicited message. During the early years of the Internet, and while its predecessor the Arpanet was in use, small groups of users generally adhered to a common understanding of the proper use of distributed applications such as email, listservs, and newsgroups. (And, in the case of the Arpanet, which was run by the United States military, there were formal rules governing its use as well.) When a user violated this understanding, there was usually a substantial negative response by others in the user community.

For example, in 1978, a Digital Equipment Corporation (DEC) sales representative sent an email to a large number of Arpanet email addresses encouraging recipients to attend a presentation on a new line of computers from DEC. This act was viewed as a flagrant violation of Arpanet regulations as well as common practice. It generated a strong anti-advertising response from other users.

 For the original message and some of the reactions to this early spam, see Brad Templeton's "Reaction to the DEC Spam of 1978" <http://www.templetons.com/brad/spamreact.html>.

Another early spam episode stemmed from a posting from two lawyers at the Canter & Siegel law firm to about 6000 newsgroups. (Newsgroups allow users to subscribe to and receive messages about a particular topic.) In the posting, the attorneys offered their services to anyone who wished to participate in a United States government lottery of "green card" work permits for foreign nationals. Newsgroup users shared a common understanding that newsgroup postings should be relevant—even relevant commercial announcements were allowed. This posting would have been appropriate on newsgroups about work permits, visas, and labor law, but appearing in completely unrelated newsgroups created a storm of protest.

Canter and Siegel received 20,000 inflammatory emails as well as numerous junk faxes. At least two Internet service providers (ISPs) terminated their service because of the volume of network traffic they were generating. Still, they, like today's spammers, were undeterred. The *New Scientist* magazine quotes Canter and Siegel's spammer's manifesto dismissing a common standard for using Internet resources:

The only laws and rules with which you should concern yourself are those passed by the country, state, and city in which you truly live. The only ethics you should adopt as you pursue wealth on the (information superhighway) are those dictated by the religious faith you have chosen to follow and your own good conscience (Source: Charles Arthur, "A Spammer in the Networks," *New Scientist*, November 1994 as found at <http://www.kkc.net/cs/new-sci1.txt>).

The spammer's credo as argued by Canter and Siegel directly contradicted the attitude of many early Internet users. As legitimate business use of the Internet has evolved, we have witnessed the rise of legitimate online advertising and a general acceptance of it. Spam, however, is still seen as largely an inappropriate use of Internet resources.

To realize the benefits of mass emailing without bearing the consequences, spammers have turned to exploiting vulnerabilities in the email system. Before discussing those vulnerabilities in detail, a brief discussion of email operations is in order.

The Basics Steps in Email Operations

Internet email systems are composed of two types of programs: clients and servers, sometimes called user agents (UA) and message transfer agents (MTAs), respectively (see Figure 6.1).

Email Clients

Client programs allow users to create, send, and manage email messages; Microsoft Outlook, Eudora, and Netscape Mail are examples of email clients. They typically provide features to organize messages into folders, offer track lists of email addresses, enable rules for categorizing messages, and provide related functions. Another core function is to coordinate the sending and receiving of messages from email servers.

Email Servers

A mail server is a transfer agent—it moves messages from the sender to the receiver's email client. Mail servers run the Simple Mail Transfer Protocol (SMTP), which listens for messages on TCP port 25. When a message arrives, the mail server examines the header information to determine the recipient, for example someuser@abc.com. In this example, the email should be sent to the mail server at domain abc.com. To do so requires mapping from the domain name to an IP address, so a domain name services (DNS) server is queried.



Figure 6.1: Mail clients and servers work in conjunction to deliver mail from senders to receivers.

DNS servers maintain several mapping records. The ones associated with email servers, MX records, identify the IP addresses of the server that should receive email messages. As Figure 6.1 shows, once the sender's email server has the IP address from the Mail eXchanger (MX) record, the message can be sent on to the recipient's email server.


In some cases, the message passes through multiple email servers. For example, when a user's account is set to forward messages, the server will send the message to the mail server in the domain specified by the forwarding address. Another case when multiple mail servers are used is when mail servers are configured as *open relays*. In this configuration, the server accepts and transfers messages on behalf of any user. Open relay stems from methods of transfer used when constant, high-bandwidth Internet connections were not commonly available. Today, open relay servers are abused by spammers and are therefore not recommended. Open relays are just one of a number of vulnerabilities found in today's email infrastructure.

Vulnerabilities in Email Infrastructure

Vulnerabilities in the email system occur in both client applications and with email protocols. In the case of email clients, vulnerabilities emerge as additional features are added to make client applications more functional and integrated with other desktop applications. The vulnerabilities in SMTP are the result of the simplicity of this protocol, which has been widely used for decades.

Email Client Vulnerabilities

SMTP and early email clients were designed for sending and receiving only text-based emails. Email clients have become more feature-rich as well, supporting scripting languages, address books, and integration with other desktop applications. Although certainly useful, these additional functions have also introduced vulnerabilities into mail clients that have been exploited by viruses, worms, and other forms of malware.

 For more information about worms, viruses, and malware in general, see Chapter 3.

One technique used by spammers exploits scripting languages supported by email clients to collect email addresses from address books. Spammers (and malware developers) can use this technique to either harvest addresses or propagate messages from the victim's account. Other vulnerabilities in email systems result from design choices in email protocols.

Protocol Vulnerabilities

A number of vulnerabilities in SMTP stem from the fact that it trusts participants in message exchanges. The open relay configuration is an example. The purpose was to provide a transfer service for others when it was not always practical to send messages directly between any two email servers. The underlying assumption was that anyone sending messages through an open relay had a legitimate purpose and was not abusing the service to the point that it hampered others.

In the case of client-to-server transmissions, a client can send a message purportedly from a user, which, in fact, is sent by someone else. A service extension to SMTP, SMTP-AUTH, has been developed that provides a mechanism to force users to authenticate before sending messages through the server. Many email servers and clients now support this extension; Figure 6.2 shows a typical type of client configuration dialog box.

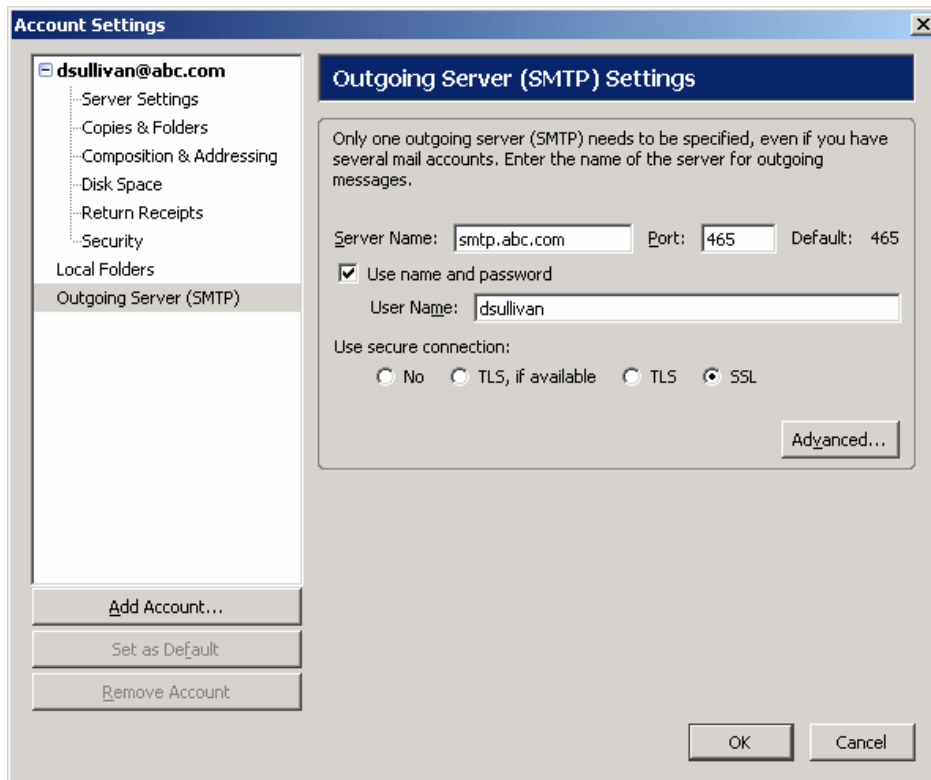


Figure 6.2: Mail servers can be configured to authenticate before allowing client-to-server communications. This example is from the Mozilla Thunderbird email client.


 For more information about SMTP-AUTH, see the specifications at <http://www.faqs.org/rfcs/rfc2554.html>.

Another example of a trust assumption is that servers do not verify the origin of a message. The sending server can put any origin address in the message and send it. The receiving server can then take one of several actions:

- Accept this address as correct and continue to handle the message
- Determine whether the origin address is the same as the machine sending the message; if not, reject the message
- Determine whether the origin address is the same as the machine sending the message; if not, insert the IP address of the sending machine

The first option allows spammers to substitute fake addresses (*spoofing*) and hide the true identity of the sender. The last option, appending the IP address of the sending machine, at least provides some path information if someone were to trace back the origin of a message. Of course, when dynamic IP addressing is used, which is common, one must know both the IP address and the time the message was sent to determine the origin.

Additional protocols have been proposed for authenticating message transmissions between clients and servers as well as between servers and servers. These proposals include DomainKeys Identified Mail (DKIM) and the DKIM Sender Signing Policy.

 For more information about DKIM, see <http://www.ietf.org/internet-drafts/draft-allman-dkim-base-01.txt>. For details about DKIM Sender Signing Policy, see <http://www.ietf.org/internet-drafts/draft-allman-dkim-ssp-01.txt>.

Hiding the Origins of Spam

The cumulative effects of vulnerabilities in the email infrastructure allow spammers to hide behind several methods:

- Faking the sender's address
- Using throwaway email accounts
- Using throwaway domains
- Relaying through third parties
- Using zombies

Authentication methods can help to reduce the problem of fake sender addresses, and increasing the cost of domains can help control the use of throwaway domains. Better security on email servers (for example, eliminating open relays) and other computers can limit the use of third-party relays and zombies.

To summarize, the vulnerabilities in email infrastructure stem from feature-rich but vulnerable clients and inherent weaknesses in SMTP. Vulnerable clients can be a source of email addresses for spammers and the SMTP weaknesses allow spammers to fake origin addresses as well as other header information.

Economics of Spamming

Spamming is a lucrative business. Take Christopher Smith, for example. In May, 2005, United States federal authorities shut down his spamming operation, Xpress Pharmacy Direct, and seized \$1.8 million in luxury cars, two homes, and more than \$1 million in cash from Smith and his associates, according to the Associated Press.

 For more information about the Christopher Smith case, see "Feds: Spamming Made Millions for Dropout" at http://www.sptimes.com/2005/09/12/Technology/Feds__Spamming_made_m.shtml.

Anecdotes such as this highlight the profits to be made in spamming; however, to understand how such a generally unwanted operation can be so successful, one must look into the economics of spamming. The economics of spam can be divided into a number of topics:

- Costs and revenues of spamming
- Distorted costs, or negative externalities, of spamming
- Beneficiaries of spamming

Costs and Revenues of Spamming

Common sense tells us that businesses will continue to operate as long as their revenues are greater than their costs. In addition, the more the revenues exceed the costs, the more likely others will want to get into the business. Economists describe our common sense understandings more precisely with the concepts of supply and demand.

The basic idea is that businesses will provide products and services as long as they can sell their inventory at a price that exceeds the cost of producing the product or service. Simultaneously, customers will continue to buy the products and services as long as the cost does not exceed the perceived value. When prices are low, more customers will come into the market; when prices are high, more producers will come into the market. The logic of the market drives the supply of a product or service to an equilibrium point with the demand (see Figure 6.3). (At least in theory.)

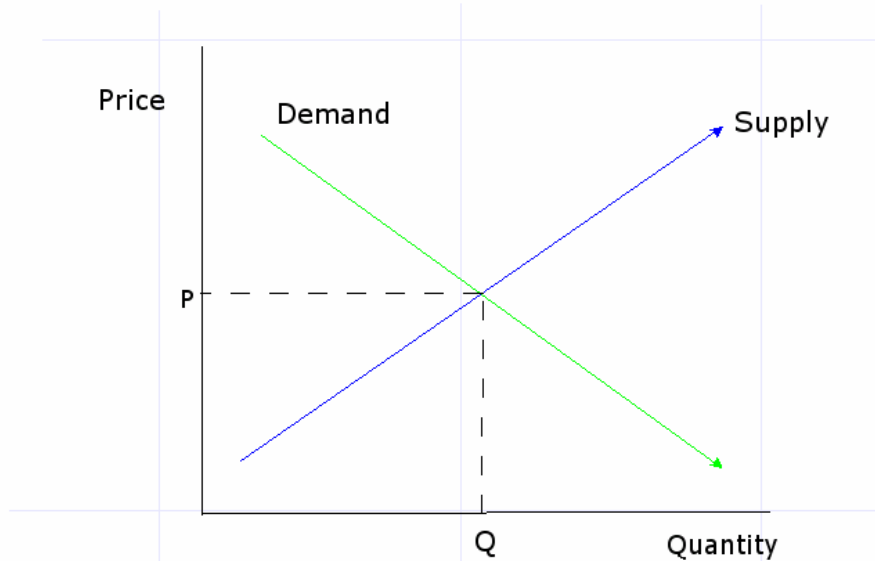



Figure 6.3: The quantity of a product in the marketplace should balance with the demand for it.

So if the supply and demand should balance, why is there so much unwanted spam? First, the providers of a product, such as spam, will generate a profit once the fixed cost of their computer equipment, Internet connections, software, and so on plus the cost of sending each additional email message (marginal costs) is paid for. The fixed costs are relatively low—even a desktop PC can generate large volumes of email messages. The cost of generating one more email once the basic equipment and network services are in place is virtually nothing. The result is that the low cost of spam allows spammers to supply large volumes of spam even when the price is low.

A second factor is that even though a spammer might send out 1,000,000 messages, only a small fraction, for example 100 recipients, might respond. Those 100 respondents actually pay the price that covers the cost and profit for the spammer. Although most consumers do not respond to any spam, a small number of consumers responding to even a fraction of all spam they receive can sustain the economic motivations for spamming. According to a 2005 survey by the Pew Internet and American Life Project, 6 percent of respondents claimed to have made purchases in response to unsolicited emails and 13 percent have responded to emails that they later discovered were fraudulent. Such high response rates are not likely to dissuade any would-be spammers from trying their hand at such easy money.

 The full Pew Internet and American Life Project survey report is available at http://www.pewinternet.org/pdfs/PIP_Spam_Ap05.pdf.

The market model works well when buyers and sellers bear all the costs and receive all the benefits of a transaction. Such is not the case with spam.

Negative Externalities of Spam

Economists use the term *externalities* when there are either costs or benefits that are shared by others outside of a transaction. For example, when a steel mill releases emissions into the air, the operation is “free” for the steel mill operators. Those living near the mill pay the price of additional air pollution. The situation is similar with spam. Those who do not want unsolicited email pay the price of having to deal with spam.


Costs Incurred by Others

Businesses and other organizations incur several costs related to spam:

- Wasted bandwidth
- Load on email servers
- Disk and archival storage
- Anti-spam applications
- Employee time

Statistics on the amount of spam are difficult to collect accurately and vary but some estimate that more than 67 percent of all emails are actually spam. This level of network traffic can drive up the cost of bandwidth for both businesses and their ISPs.


Spam also places additional computational and storage load on email servers. With increasing legal precedents and regulations governing electronic communications, some organizations are storing large volumes of emails for extended periods of time. If an organization does not want to risk deleting legitimate email, it might choose to store and archive all emails, including spam. The additional storage and archival costs are born by the recipients of spam, not the senders.

 Legitimate email is also known as “ham” when distinguishing it from unsolicited, unwanted email, spam.

Ideally, spam will never reach a user's inbox. An important element of many IT organizations' strategies for controlling spam includes the use of anti-spam applications. These include both software and network appliances that scan emails before they reach the mail server or, at least, the end user. The cost of acquiring, maintaining, and managing these systems is another cost incurred by those outside of the business transactions of the spam sender.

Another cost that is difficult to quantify is the value of lost employee productivity. Managing spam is annoying, and if too much reaches the inbox, can become a time drain on end users.

As much of the true cost of spam is paid by someone other than the spammer or the person responding to the spam, the economic balance of spamming is therefore distorted.

 You can find a useful anti-spam ROI calculator at <http://www.mcafeesecurity.com/us/products/tools/roi/spam.asp>.

Distorting the Supply/Demand Balance

As depicted in Figure 6.3, free markets will balance themselves so that suppliers will come into the market in just the right number to meet the demand for a product at a particular price. This principal describes the behavior of suppliers in aggregate. Individual suppliers will enter or leave the market depending on the profit they can earn. The suppliers' profits are determined by a combination of their fixed costs, marginal costs, and marginal revenues.

Consider the decision-making processes of a spammer. To get started, the spammer needs a PC, a network connection, and some software. He or she does not have to pay any of the costs incurred by the recipients of spam, so the normal cost information provided by markets is distorted—a fact that has a direct impact on the spammer's decision making.

To maximize profits, suppliers will continue to provide a product as long as the marginal revenue earned is greater than the marginal cost. As Figure 6.4 shows, when the cost to the spammer does not reflect the full cost of spam, the spammer will produce a large quantity of spam (shown as the Distorted Supply line). If the spammer had to incur the full cost of spam, including the costs now incurred by unwilling recipients, the supply would be much less and possibly non-existent (shown as the Undistorted Supply line).

The distortion in the supply of spam is that the market does not convey the proper cost signals to the supplier. The result is the negative impact on email users who are not customers of the spammers. This kind of negative impact, or externality, has occurred before.

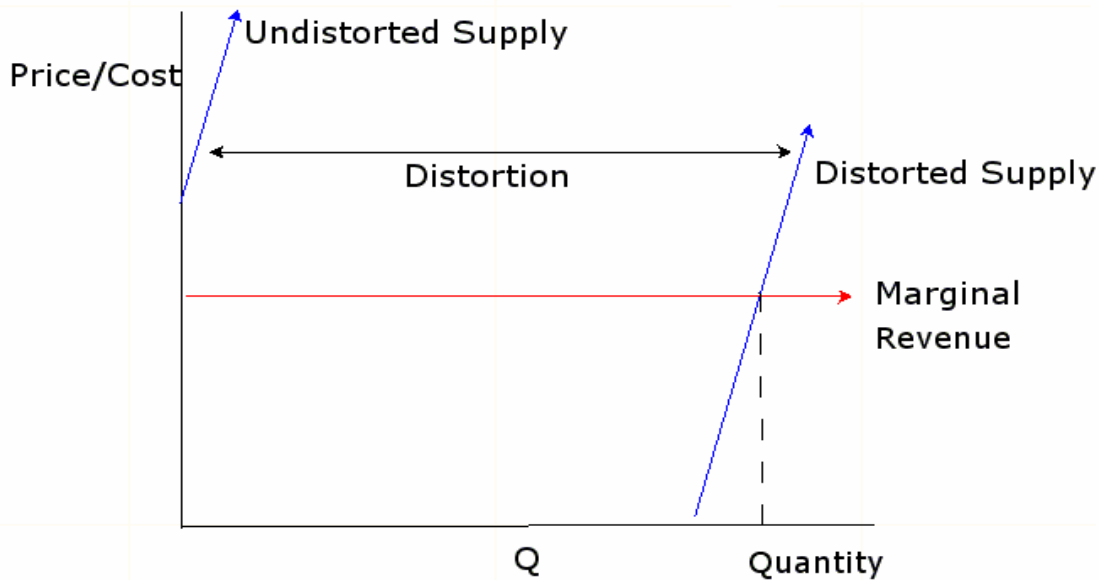



Figure 6.4: Without the impact of the true cost of spam on the suppliers, the quantity of spam generated is greater than what the market equilibrium would support.

Correcting for Negative Externalities: Government Regulation

Pollution is a classic example of a negative externality. The market does not have a mechanism to force polluters to incur the cost of pollution. From the perspective of a polluter, there is no cost to the bottom line for polluting and so no reason to stop or limit it. Governments compensate for this type of market inefficiency by imposing a direct cost on polluters through regulations, such as fines for exceeding pollution limits. In an effort to apply this tactic to spammers, many countries have established laws regulating spam, such as the Controlling the Assault of Non-Solicited Pornography and Marketing (CAN-SPAM) Act enacted in the United States in 2003.

 The United States, the European Union, and some of its member states as well as other countries have passed anti-spam legislation. For links to information about these laws, see <http://www.spamlaws.com/>.

The CAN-SPAM Act

The CAN-SPAM Act was passed because spam had moved well beyond being a mere inconvenience or annoyance to a threat to the continued utility of electronic communications. The United States Congress also found that a number of states had passed anti-spam laws but found that the different penalties, standards, and requirements were not effectively addressing the problem.

The CAN-SPAM Act prohibits a number of actions commonly used by spammers:


- Using false or misleading transmission information
- Using deceptive subject headings
- Including false return addresses
- Sending additional commercial mail after recipient objects
- Using address harvesting and dictionary attacks
- Using automated methods to create email accounts for sending spam
- Relaying email through unauthorized access

The law also requires that advertisers provide an opt-out method for recipients as well as a valid physical address of the sender and a clear indication that the message is an advertisement. The legislation includes fines of as much as \$11,000 for violations of each provision. The law also allows for criminal prosecution in cases in which spammers

- Make unauthorized use of another's computer to send spam
- Relay messages through multiple mail servers to hide the messages' origin
- Falsify header information
- Register for multiple email accounts or domain names using false identities
- Falsely represent themselves as owners of IP addresses

Researchers studying the effects of CAN-SPAM have found that the law does help consumers and ISPs block unwanted, unsolicited emails when the senders comply with the law, but technical measures are required when spammers do not comply. Even the effectiveness of technical countermeasures is limited when spammers take actions to evade detection (Source: Matt Bishop, "Spam and the CAN-SPAM Act"

<http://www.ftc.gov/reports/canspam05/bishoprpt.pdf>).

 A summary of the CAN-SPAM Act's requirements for commercial emailers is available from the United States Federal Trade Commission (FTC) at <http://www.ftc.gov/bcp/online/pubs/buspubs/canspam.htm>. The full text of the law is available at http://frwebgate.access.gpo.gov/cgi-bin/getdoc.cgi?dbname=108_cong_public_laws&docid=f:publ187.108.pdf. An FTC report on the effectiveness of the act is available at <http://www.ftc.gov/reports/canspam05/051220canspamrpt.pdf>.

Effectiveness of CAN-SPAM

Measuring the effectiveness of CAN-SPAM is difficult. Surveys and research reports do not always agree. According to a report from the United States FTC, since CAN-SPAM went into effect, the spam problem has improved. These improvements are attributable to a number of factors:

- Spam volume has declined, along with consumer frustration, as legitimate mass emailers comply with the law
- Anti-spam technologies have improved and are widely deployed
- The adoption of domain authentication protocols that verify the source of the message is the same as the source listed in the message header

Federal prosecutors are also using the legislation to bring criminal charges against spammers. In the first year, more than 20 cases were brought under the CAN-SPAM Act. In one case, a federal grand jury indicted spammers involved in a large-scale international spam operation that used several techniques to hide the origin of their messages:

- Sending from computers with IP addresses in the Netherlands with domains registered in the Indian Ocean island state of Mauritius
- Using fake addresses in the From line
- Remotely controlling servers in the Netherlands from systems in the United States

In addition, overseas companies were established to disguise their operations and overseas bank accounts were established to facilitate the laundering of profits from the spamming venture (Source: Department of Justice Press Release “Three Defendants Indicted, Fourth Pleads Guilty In Takedown Of Major International Spam Operation,” August 25, 2005—http://www.usdoj.gov/opa/pr/2005/August/05_crm_431.htm).

Although the FTC report argues for definite improvements, the Pew Internet & American Life Project finds mixed success in controlling spam. A 2005 report on spam finds:

- 28 percent of email users with personal email accounts report receiving more spam than a year ago; 22 percent report receiving less
- 21 percent of email users with work email accounts report receiving more spam than a year ago; 16 percent report receiving less

It also reported some improvements, for example:

- 53 percent of email users say spam makes them less trusting of email, an improvement over the 62 percent the prior year
- 22 percent of email users have reduced their use of email due to spam, an improvement over the 29 percent the prior year
- 67 percent of email users claim that spam has made their online experience unpleasant or annoying compared with 77 percent the prior year

 For full details on the Pew Internet & American Life Project report, see http://www.pewinternet.org/PPF/r/155/report_display.asp.

Even with the cooperation of legitimate emailers and the ability of regulators to charge violators, spam continues. Regulators recognize that legislation and prosecution are not enough to stem spamming, and technology will continue to play a central role in efforts to control unsolicited emailing. Before addressing the technical aspects of controlling spam, there is one more element of the economics of spamming that should be addressed.

Cutting Costs and Avoiding Prosecution: The Role of Botnets in Spam

When regulators work with ISPs, together they can trace the source of spam to its origin and find the perpetrators—at least that used to be the case. One can still trace the path of spam through mail servers across the network, but doing so today often leads not to a spammer's server but a compromised system owned by an unwilling participant in a spam operation. Using someone else's computer to distribute spam has two key advantages: it saves the cost of hardware and it makes it more difficult to trace the source of spam.

Spammers have adopted techniques used by malware developers, such as installing malicious programs known as Trojan Horses on victims' computers. The spammer communicates with the Trojan Horse through IRC, FTP, or some other communication protocol, sending instructions and data as needed to direct the distribution of spam. By infecting large numbers of computers, the spammer can create and control a network of "zombies" that perform the bulk of the work for mass mailings. For more technical details on botnets, see John Kristoff's presentation "Botnets" at <http://www.nanog.org/mtg-0410/pdf/kristoff.pdf>.

Beneficiaries of Spam

The obvious beneficiaries of spam include the spammers themselves. When consumers respond to spam, spammers benefit from the profit on any sales transactions but also from the fact that they have the address of a responder. Email addresses of spam responders can command a premium when selling email lists. Unfortunately, this is not the extent of businesses that profit from spam.

In addition to selling products, spammers can make money by generating leads for high-value products such as mortgages. For example, a technology correspondent for MSNBC responded to an unsolicited email with an enticing subject line about low interest mortgages. Within days, he received four inquiries from legitimate mortgage institutions.

The path from the spammer to the lenders passed through a lead generator in the United States who claimed to have purchased the information "from someone else, who in turn bought it from someone else, who in turn bought it from an emailer based in China" (Source: Bob Sullivan, "Who Profits from Spam?" at <http://msnbc.msn.com/id/3078642/>). The lead generator in the United States sold the information to another party who then sold it to yet another party before it was finally sold to a mortgage provider (see Figure 6.5).

When customers complain, legitimate businesses can track down offending lead generators and affiliates—incurring yet another additional cost to businesses directly related to spam. It also requires consumers to take the time to file complaints and provide details so that the business can address the problem. Needless to say, it is often not worth the consumer's time to file the complaint in the first place. The result is that these grey market operations can launder email addresses and lead information providing yet another line of revenue for spammers.

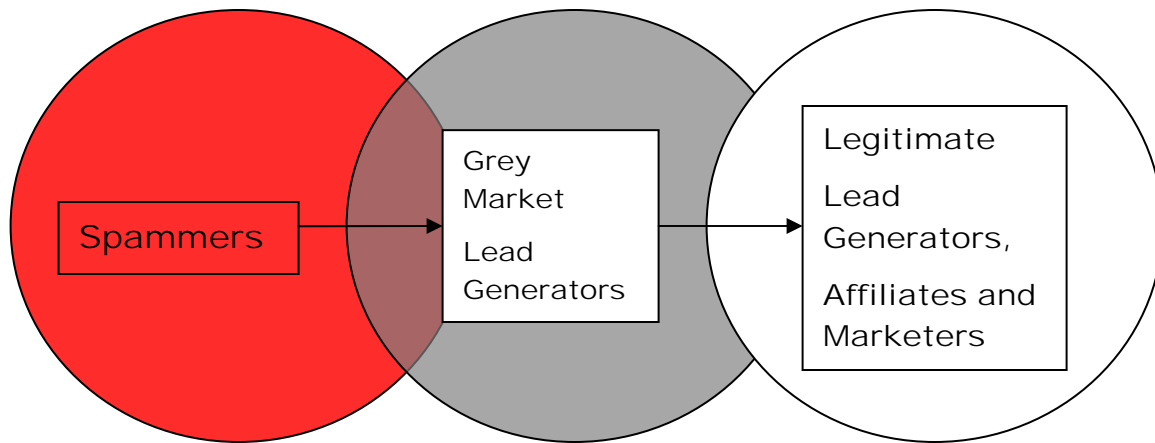


Figure 6.5: Legitimate businesses may ultimately use leads generated from spammers that use a network of grey market lead generators to mask the original source of the leads.

Clearly the economics of spam hold the potential for profit, especially when one is willing to circumvent regulations. Government intervention is certainly helpful—it will at least allow legitimate mass email operators to operate within defined boundaries. Regulations alone will not control spam, and technical measures are required to adequately address the problem.

Spam Management

Just as regulation is not enough to control spam, no single technical measure will completely control spam. The process of identifying and disposing of spam consists of four operations:

- Detection and determination
- Actions in response to spam
- Managing detection methods
- Managing quarantined messages

Together, these tasks constitute the technical aspects of spam management.

Detection and Determination

The first step in spam management processes is to detect suspect messages and determine whether they are actually legitimate messages or unwanted, unsolicited email. Ideally, this process occurs before messages reach the recipient's email server and thus eliminate unnecessary load on the server. The process should operate on all messages; if multiple email servers are in place, the spam detection and determination operations should occur on traffic streams to all of them.

Once email traffic has been intercepted or redirected to a spam detection mechanism, there are several methods for determining whether a message is spam. The common techniques are:

- Integrity analysis
- Heuristic detection
- Content filtering
- Blacklists and whitelists
- Self-tuning
- Bayesian filtering
- DNS block lists

These techniques each have advantages and disadvantages, but together their complementary strengths provide an effective detection mechanism.

Integrity Analysis

Integrity analysis examines the structural characteristics of email messages to determine whether they may be spam. The header, the layout of the messages, and the overall organization of the message can provide clues about the status of a message. For example, a header may have an invalid time zone or a date far earlier than the current date. The body of the message might contain a single line of text in upper case with many whitespace characters and end with an exclamation mark followed by a short paragraph of text and a URL preceded by "Click here." This type of detection is a specific form of the more general process of using heuristics to detect spam.

Heuristic Detection

Heuristics, or rules of thumb, are often used to craft detection rules. Rules are typically of the form:

```
If <some property of the email>  
Then spam score = spam_score + <confidence factor>
```

The condition, "some property of the email," can be any pattern that is indicative of spam. For example, the presence of a spam-tool name in the header or the use of upper-case letters in the subject line or body of the message.

The confidence factor is a measure of how well the pattern in the condition indicates spam. Some patterns are very often associated with spam, such as a URL with the word “remove” in it and a subject line that contains a unique identifier. These would qualify as high-confidence factors. Other patterns often found in spam, such as color-coded HTML text is also found in legitimate email, so the confidence factor would be lower.

The spam score is the sum of confidence factors of all rules that are true for the message. When a message’s spam score exceeds a predefined threshold, the message is categorized as spam.

One advantage of heuristic rules is that they allow for custom coding of filters that can take into account the location of a pattern (for example, in the header versus the body) and thus can make finer distinctions than just looking for a particular pattern of characters anywhere in a message. Another advantage is that different combinations of rules can detect a wide variety of spam. One spam message might be detected because of the use of colors and keywords while another is detected because of an invalid date header and a suspicious phrase such as “As seen on national TV!” At the same time, the confidence factors and thresholds can be set to ensure that no single rule can trigger the classification of a message as spam. They can also be adjusted to minimize the likelihood of identifying legitimate mail as spam.

Perhaps the biggest disadvantage of heuristic rules is the time it takes to develop them. Designers have to take into account how the condition in the rule might detect legitimate email and how the rule interacts with other rules. Unlike virus detection, which can use a single signature to identify a particular piece of malware, spam detection is done with a set of rules so that full rule sets must be developed and tested before they are released.


Content Filtering

Content filtering uses lists of words and phrases that indicate spam or offensive material that is banned within an organization. Content filtering is an efficient technique for identifying blatantly offensive messages but it lacks the ability to distinguish legitimate uses of a term from inappropriate uses. Content filtering lists are generally limited so as not to mistakenly categorize legitimate email based on the use of terms that have both appropriate and inappropriate uses in a business context.

A specialized type of content filtering involves domain name reputation technology. The domain name reputation technique examines the URLs within each email message and blocks those messages that contain links to malicious or spammer Web sites. The technique works to block spam, phishing messages, and messages containing links to malicious payloads such as spyware. The accuracy of this technique can be very high but is dependent on having reliable and up-to-date lists of suspicious domains. Such lists are very costly to maintain and are typically beyond the reach of all but the largest and best-funded anti-spam vendors.

Blacklists and Whitelists

Blacklists and whitelists are essentially lists of known spammers and known non-spammers, respectively. The advantage of these lists is that they allow the spam management software to quickly categorize messages based on information in the header. More computationally intensive operations, such as applying heuristic rules and calculating spam scores, are not necessary.

 The Open Directory Project (<http://dmoz.org>) maintains a list of blacklist providers at <http://dmoz.org/Computers/Internet/Abuse/Spam/Blacklists/>.

Self-Tuning

The techniques discussed so far are generalized to apply to all email users. For example, if an email user is categorized as a spammer on a blacklist, that user is considered a spammer for everyone. With self-tuning techniques, anti-spam applications can adjust their categorization rules to take into account the type of email individual users or organizations receive. When a message is sent from a user with a history of sending legitimate emails, the spam score assigned by heuristic rules can be adjusted based on patterns derived by self-tuning.

Bayesian Filtering

Bayesian filtering is a mathematical method for using the probabilities of a word, phrase, or symbol being found in a spam message. Unlike content filtering, which is based on a list of commonly used words and phrases compiled by a human, Bayesian filtering methods analyze large numbers of spam and legitimate mail to calculate precise measures of a word's likelihood of being found in spam.

Without going into too many details of the math behind Bayesian filtering, we can still outline the basic points. First, Bayesian filtering uses probabilities that are measured between 0 and 1 that indicates the likelihood of a particular hypothesis; 0 indicates with certainty that the hypothesis is false, 1 indicates with certainty that the hypothesis is true. If a weather forecaster says there is a 0.5 probability of rain, it is just as likely to rain as not to rain. Probabilities calculated for Bayesian filtering are different from scores used in heuristic rules. Scores are intuitive measures assigned to rules by individuals; they may be adjusted to improve accuracy when used with other rules. Probabilities are calculated using a specific formula.

Second, Bayesian filtering uses information from both example spam and legitimate mail. Consider the word "free." It is used in both spam and legitimate email. In a sample of 10,000 legitimate emails, "free" may appear 200 times; the same word might appear 1000 times in a sample of 10,000 spam messages. Clearly the word "free" is a good indicator of spam, but how good? How should you weight your belief that an email with that word is actually spam? A Bayesian formula, known as Naïve-Bayes, can tell you the answer (see Figure 6.6).

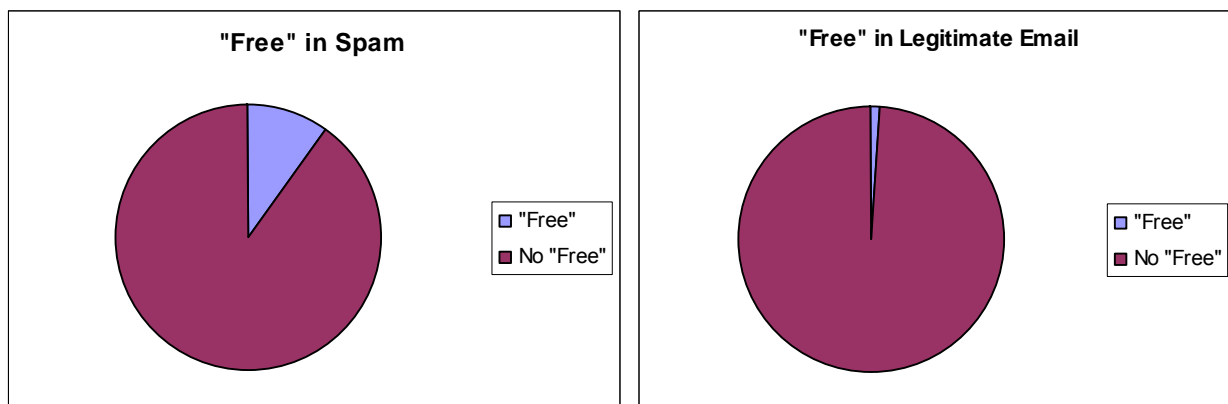



Figure 6.6: Bayesian filtering takes into account the frequency with which words appear in both spam and legitimate emails.


Another factor of Bayesian filtering is that it can adjust and adapt as the number of examples of spam and legitimate email grows. For example, email sent by a user as well as email sent from email users on an organization's whitelist can be considered illustrations of legitimate email. Messages identified as spam by email users or deleted without opening may be considered spam and used to adjust the spam probabilities of words occurring in those messages.

 For more details about the math behind Bayesian filtering, see http://en.wikipedia.org/wiki/Naive_Bayes_classifier.

Spammers try to avoid detection using several techniques:

- Using long stories that skew the statistical measures and lessen the overall probability measure that a message is spam
- Inserting randomly selected words from a dictionary to again skew the spam measure
- Adding unrelated information, such as the text from a news story, into the message.

Fortunately, these techniques that can fool a Bayesian filter can be detected by other spam-detection techniques such as heuristic rules. Spammers may be able to get around one anti-spam measure, but it is difficult to fool multiple techniques simultaneously.

 For a discussion on the limits of Bayesian filtering, see William S. Yerazunis' "The Spam-Filtering Accuracy and How to Get Past It" at http://crm114.sourceforge.net/Plateau_Paper.pdf.

DNS Block Lists

DNS block lists use the IP address of the sender to identify known spammers. DNS blocks lists are publicly available on the Internet so that email administrators or anti-spam vendors do not need to maintain individual lists. Of course, like other commonly maintained resources, the quality can vary from poor to good. It is not uncommon for DNS block lists to contain the addresses of non-spammers (false positives).

A combination of detection techniques, from examining the structure of a message to analyzing word frequencies, can provide highly accurate means of detecting spam without generating unacceptable numbers of false positives. Of course, once spam has been identified, the anti-spam system must decide what to do with it.

Actions in Response to Spam

Once a message has been categorized as spam, there are generally three actions the anti-spam system can take:

- Delete the message automatically
- Quarantine the message
- Tag the message as spam and deliver it

The appropriate action will depend on user preferences and the level of confidence that the message is truly spam.

When a message is deemed to be spam with certainty—for example, the message is from an address on a blacklist, the message may be deleted without user intervention. Addresses included on blacklists are used only when all email from those sources should be blocked. Email administrators may also configure an anti-spam system to automatically delete any message with a spam score above a particular threshold. The drawback of deleting messages is that in the case of a false positive, the recipient has no way to know the message ever existed or to review it before disposing of it.

Quarantining is an alternative to deleting messages in which messages are isolated. The recipient's inbox is not cluttered with spam, but messages are not permanently deleted either. Typically, the messages are preserved in an isolated folder and a list of quarantined messages is sent to the recipient. The recipient can retrieve any of the quarantined messages or delete them if they are actually spam. Messages in quarantine are usually purged after a predefined period of time (such as 30 days) if no other action is taken on them by the recipient.

The third option is to tag an email message and send it on to the recipient. For example, the prefix spam could be added to the subject line to highlight the fact that the message is likely spam.


The appropriate action will depend on a combination of factors, including tolerance for false positives and the amount of spam received. Low tolerance for false positives calls for quarantining or tagging. Receiving large volumes of spam argues for automatic deleting. When both factors are in effect, users and administrators must find their own suitable balance.

Blacklist and Whitelist Management

As previously noted, blacklists are used to define addresses of known spammers and whitelists are used to define addresses of known legitimate emailers. These lists are not static and can change frequently as the needs of the organization and individual email users, as well as the behaviors of spammers, change. These lists must be updated, for example, when

- Spammers change domains and begin sending spam from a new source
- Businesses acquire new customers or business partners
- Email users find themselves on mailing lists they do not want to be on
- Legitimate emailers have been added to blacklists by mistake

Both blacklists and whitelists should be managed globally and individually. Global lists apply to all messages sent to a server. Email administrators are responsible for maintaining these. As it is difficult, if not impossible, for a single organization to track all known spammers, many email administrators use publicly available real-time blacklists.

 The Spamhaus Block List (<http://www.spamhaus.org/sbl/index.lasso>) and SORBOS (<http://www.nl.sorbs.net/>) are two popular blacklists. For a comparison report on a number of blacklist sources, see the latest report at http://www.sdsc.edu/~jeff/spam/Blacklists_Compared.html.

Anti-spam systems should also allow individual users to configure personalized blacklists and whitelists. Server-level blacklists should catch many spammers, so the personalized blacklist can target lower-volume spammers and phishers. In addition to managing blacklists and whitelists, email administrators and users also need to manage the quarantine process.

Quarantine Access, Search, and Management

Quarantining messages can help to counter the effects of false positives but it does impose additional management tasks on administrators and users. Administrators have to take into consideration a number of factors, including:

- How much disk space to allocate to quarantine folders
- How long to retain quarantined messages before deleting them
- How to adjust retention policies for different types of users

For users, the issues tend to center around access and search. How often is the user notified about quarantined messages and how is the information delivered? A single message once a day listing all quarantined messages works well in many cases. Users may also want immediate access to the quarantine folder through their email client. With long retention periods, the number of spam messages in a quarantine folder will grow and users will need search tools to sort and filter quarantined messages as they look for legitimate messages that may have been incorrectly categorized as spam. The ability to manage quarantined messages is just one of the considerations one should take into account when evaluating anti-spam systems.

Evaluating Anti-Spam Systems

There are a number of anti-spam systems available today, both from vendors and open source projects. Choosing among these options can be a difficult task, so it is important to focus on key features:

- Catch rate
- False positive rate
- Manageability and reporting

In addition to these, there are, of course, the ever-present concerns about integration and reliability that come with any enterprise-scale application.

Catch Rates

An anti-spam system's catch rate is a measure of how well it detects spam. As noted earlier, spammers can often trick one detection method at a time but it is difficult to fool multiple methods simultaneously. For high catch rates, consider systems that use multiple techniques, especially Bayesian filtering, heuristic rules, and domain name reputation technology.

False Positives

False positives, or legitimate email categorized as spam, are a serious problem for anti-spam systems. It is generally preferable to allow some spam through rather than risk blocking legitimate email. False positive rates can be minimized with the use of whitelists, heuristic rules, and self-tuning. Systems that allow individual users to customize whitelists and train Bayesian filters can also help reduce the chances of generating false positives.

Manageability and Reporting

Manageability is one of those general characteristics exhibited by well-designed systems. In the case of anti-spam systems, there are three areas administrators should consider:

- Flexibility in configuration for different sets of users
- Ease of updating blacklists and whitelists
- Ease of quarantine access and management

Different sets of users will require different policies governing their use of email. For example, some departments, such as legal affairs or human resources, may have no tolerance for false positives. They may need all suspect messages with high spam scores quarantined and those with moderately high scores labeled and delivered to the recipient. Other accounts, such as a general customer service email account, may receive a great deal of spam because the email address is published on the company Web site. In that case, messages with moderate or high spam scores may be automatically deleted while low scoring messages are quarantined for short periods of time, such as a day or two, before being deleted.

Email administrators should be able to edit their whitelists easily. Email users should also have the ability to specify custom lists based on their own email patterns.

Finally, both administrators and users should be able to review quarantined messages, transfer them to inboxes, and delete them as necessary. Additional searching, sorting, and filtering features will help when large numbers of messages are quarantined.

Anti-spam systems should also provide administrative reports that allow managers to track the volume of email messages analyzed, the number of spam messages detected, the volume of storage used for quarantine, and other key indicators of the performance of the system.

Spam management depends upon technical and regulatory measures. Although regulations help to define the boundaries of appropriate mass emailing and allow legitimate marketers to stay within the law, they cannot prevent determined spammers from flooding inboxes with unsolicited, unwanted email. Technical solutions, especially multi-tiered methods for spam detection, are the key to controlling the impact of spam on email infrastructure and end users.

Summary

Spam is nothing new. One of the earliest examples of spam dates back to the late 1970s, and by the 1990s, spamming email and listserv systems was growing in automation and sophistication. Today, spammers exploit vulnerabilities in email infrastructures, leverage compromised hosts (“zombies”), and adapt their messages to avoid detection from a number of anti-spam techniques. Although spamming is illegal in many countries, the problem continues. The economic benefits of spam are based on the fact that even very low response rates can generate enough revenue to more than cover the direct costs to spammers. This, of course, does not cover all the costs, because, like pollution, the cost of spam is shared by many, not just those that create the problem.

Anti-spam systems employ effective and highly accurate methods for detecting spam. A combination of several methods provides the best defense against the adaptive nature of spammers. In addition to accuracy, manageability and reporting are key considerations when selecting an appropriate anti-spam system for an organization.

Chapter 7 Technologies for Securing Information and IT Assets

Throughout, this guide has have examined threats to an organization's ability to protect the integrity and confidentiality of its information. Some of the most troubling threats today include:

- Viruses, worms, and other forms of malware
- Spyware that monitors, gathers, and steals information about users
- Phishing scams and identity theft
- Spam that taxes network and email service resources
- Employee behavior with IT resources that violates regulations and company policies

Techniques and technologies for controlling these threats is the focus of this chapter. The chapter begins with a focus on content-specific measures for mitigating the impact of these threats. The chapter concludes with a discussion of how a multilayered defense strategy can effectively provide adequate levels of security for an organization while maintaining necessary levels of usability and performance.

Content-Specific Technologies for Information Security

Content-specific technologies for information security address the specific issues that arise when preventing external threats from inflicting damage using network-based communication services while addressing the threats from inside the organization. Three representative technologies are considered:

- Email content filtering
- Email spam filtering
- URL blocking

Email content filtering uses a variety of techniques to identify malicious or inappropriate content and prevent it from entering or leaving an organization's network. Email spam filtering is a specialized form of email content filtering that uses additional techniques specific to the practice of spamming. The final technology, URL blocking, is a relatively simple but highly useful technique to prevent access to Web sites and resources that are known to contain problematic content.

Email Content Filtering

Email content filtering has emerged as a necessary element of information security because of a number of inappropriate uses of email. Sometimes the inappropriate use originates from the outside, but such is not always the case. Inappropriate uses of email systems by employees, contractors, and other insiders have also contributed to the necessity of filtering email.

Four commonly recognized threats can be addressed with the use of email filtering:

- Malicious programs transmitted through email messages
- Offensive material sent through email, creating a hostile work environment
- Loss of intellectual property
- Disclosure of private information

These problems are not solely limited to email services, for example the problem of a hostile work place existed long before email. Email, however, has changed the ways in which these threats are realized and the speed and breadth with which they can damage an organization.

The Evolution of Email Functionality and Vulnerabilities

Email is a channel for moving staggering amounts of information in and out of an organization. In the early days of email adoption, few controls were needed. Users were primarily academics, researchers, students and government contractors and employees working with the early Internet or its predecessor, the Arpanet. Then email evolved, both technically and with respect to its user base.

On the technical side, advances such as the Multipurpose Internet Mail Extensions (MIME) standard allowed users to attach documents to their text messages. Users were no longer limited to sending basic text messages; documents, images, and executable files could be sent as well. Email clients also added functionality, supporting address books with email addresses of other users, supporting macros to automate routine tasks, and offering improved interface functions, such as the ability to display HTML.

Unfortunately, as is often the case with software, greater functionality leads to more vulnerabilities (see Figure 7.1). Hackers and malware developers have been able to exploit these vulnerabilities, transmitting viruses over email, which, from the virus writers' perspective, must have been a great improvement over transferring the virus by floppy disk, which was the old transmission method. Sometimes email functionality was combined with vulnerabilities in desktop applications, such as Microsoft Word, to create viruses such as the Melissa and I Love You viruses.

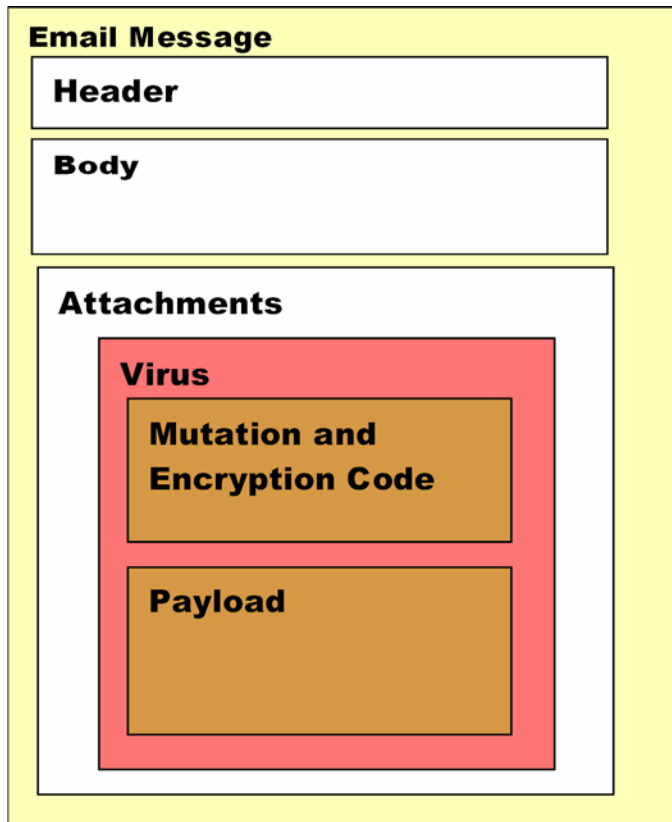


Figure 7.1: Improvements in email functionality, such as attachments, have become mechanisms for propagating malicious software.

For more information about these viruses, see the McAfee Virus Information Library. Details of the Melissa virus are available at http://vil.nai.com/vil/content/v_10132.htm; for details about the I Love You virus, see http://vil.nai.com/vil/content/v_98617.htm.

Today, the simple fact is that email content cannot be trusted. This reality is unfortunate because the vast majority of users have no intention of inflicting harm by propagating malware or overwhelming email servers with spam. At the same time, the evolution of email to this point is not surprising. The potential for malicious behavior in cyberspace mirrors the same potential in other areas. Brick-and-mortar stores deploy anti-theft devices (euphemistically called “inventory control devices”) at exits to deter shoplifting; airports employ complex screening procedures to reduce risks to travelers. As email cannot be trusted, it must be examined and, if found to be malicious or problematic in some way, dealt with according to a well-defined procedure.

Common Email-Based Threats

Common forms of malicious software that can be transmitted through email include:

- Viruses—Malicious programs that propagate by using other programs. Some viruses purposefully change characteristics with each generation to avoid detection by antivirus software; antivirus systems, however, have changed detection techniques to counter those mutating schemes. Viruses may also encrypt themselves to avoid detection. This practice leaves them vulnerable to detection by looking for decryption routines with the body of the virus unless the decryption code is changed using a mutating technique.
- Worms—Malicious programs that propagate on their own. These programs typically take advantage of a vulnerability in an application, such as an email client. Mutating and encryption techniques used in viruses may also be used with worms.
- Trojan horses—These programs appear to do one thing, such as add a toolbar to a Web browser, and in fact also carry out malicious activities such as recording keystrokes and capturing passwords.
- Keyloggers—These programs intercept keyboard events and record keys typed by a user. The information collected by keyloggers may be copied to a server controlled by a hacker who then uses the information to compromise the computer, conduct identity theft, or perform some other form of theft or fraud.

Email systems allow outsiders to introduce malicious software into an organization; it is also used by insiders in ways contrary to the intention of the organizations that provide the system.


Harassment and Hostile Workplace Issues

A hostile workplace is defined as an environment in which the harassment fundamentally alters the conditions of a workplace. The concept of a hostile workplace arose out of litigation related to sexual harassment in the United States Supreme Court case, *Meritor Savings Bank v. Vinson*, which found sexual harassment prohibited by Title VII of the Civil Rights Act of 1964. As technology in the workplace has changed, it is not surprising that the considerations for a hostile workplace have changed to include technology, especially email.

Emails have been central to cases of litigation, dismissal, and employee discipline for well over a decade:

- In 1995, Chevron Corporation was ordered to pay \$2.2 million to female employees to settle a case involving inappropriate emails.
- In 2002, six employees of the State of Washington Labor Department were terminated for excessive use of the state-provided email system for personal use. One former employee was chided for emails that contained “shockingly explicit, vulgar, and very offensive” language (Source: <http://www.gigalaw.com/articles/2003-all/towns-2003-03-all.html>).
- In 2006, a Chesterfield Township, Michigan detective was suspended and demoted for sending pornographic material through email to coworkers (Source: <http://www.freep.com/apps/pbcs.dll/article?AID=/20060208/NEWS04/602080469/1001/NEWS>).

Preventing the distribution of offensive material is a key driver to the adoption of content-filtering systems.


 For more information about email and its role in hostile workplace considerations, see Douglas M. Towns, “E-Harassment in the Workplace” at <http://www.gigalaw.com/articles/2003-all/towns-2003-03-all.html>.

Another issue facing commercial organizations is the loss of intellectual property and other confidential or sensitive information through information leakage.

Information Leakage and the Loss of Intellectual Property

Information leakage occurs when proprietary information is intentionally or inadvertently released to individuals outside the organization. The intentional theft of information is on the rise. According to the 2005 Computer Security Institute/FBI Computer Crime and Security Survey, unauthorized access to information and theft of proprietary information rose significantly over previous years.


The motivations for intentional theft can range from financial incentives, such as stealing a competitors’ customer list, to revenge by a disgruntled employee. In one recent case, a former Honeywell employee released payroll and other personal information about 19,000 Honeywell employees on the Internet; how the perpetrator procured the information has not been disclosed (Source: <http://www.computerworld.com/securitytopics/security/story/0,10801,108434,00.html>).

 For the full report on the CSI/FBI 2005 Computer Crime and Security Survey, see http://www.gocsi.com/forms/fbi/csi_fbi_survey.jhtml. For a disturbingly long list of incidents involving the release of confidential and private information, see ComputerWorld’s Special Coverage: Data Security Breaches at <http://www.computerworld.com/news/special/pages/0,10911,3888,00.html>.

Unintentional disclosures can be damaging as well. Typically, these disclosures result from one of several root causes:

- Not understanding the full range of information stored in a document or database
- Ignorance of organization policy
- Not following security procedures

For example, many users of Microsoft Office might not be aware that metadata about the author and file location may be stored in the document. In addition, “deleted” information may remain in the file but not displayed by Word.

 AntiWord, a free Microsoft Word reader, maps Word formatted documents to text and includes a feature to include information hidden in Word. This tool can be used to determine whether sensitive information is inadvertently stored in a Word document. AntiWord is available at <http://www.winfield.demon.nl/>.

In other cases, employees and contractors might not be aware of company policies governing sensitive information. For example, a marketing company making a proposal to a prospective client might be asked to provide references and examples of prior work. The salesperson may decide to send a sample from another client without first asking permission from the client, violating company policy. This act could leave the marketing firm liable to the customer whose information has been disclosed.

In the worst case, employees and contractors may be aware of company policies but violate them anyway. For example, an employee may be working with a vendor to create a demonstration of a new document management system proposed for the research and development department. The vendor asks for sample documents in order to create a realistic demonstration. Although the employee knows such documents cannot be sent outside the company, he or she emails them anyway, assuming they will just be used for the demonstration. Of course, once the documents leave the boundaries of the company's network, the company will have no control over their distribution and use.

In some cases, the disclosure of information does not involve proprietary information about a company but private information about its customers.

Disclosure of Private Information

Even when employees know the rules, there are times the rules are violated. Consider an employee rushing to meet a deadline to deliver a list of customer contacts for a marketing campaign for a bank. The contact information includes sensitive demographic data, including household income as well as account numbers. According to company policy, the data must be encrypted before being transmitted outside the corporate network, but the employee decides to “cut a corner” to save time and sends unencrypted data outside the company. If this data were intercepted, the company could be liable for violating privacy regulations.

Privacy regulations have been created by a number of government bodies and cover a range of topics:

- The Health Insurance Portability and Accountability Act (HIPAA) of 1996 specifies regulations for maintaining the privacy of protected healthcare information.
- The Gramm-Leach-Bliley Act (GLBA), formally known as the Financial Modernization Act of 1999, includes provisions to protect consumers' private financial information.
- California law, State Bill 1386 (SB 1386), addresses the threat of identity theft and fraud by requiring notification when a California resident's identifying information (such as Social Security number or driver's license number) is disclosed.
- The Safe Harbor framework, negotiated by the United States Department of Commerce and the European Commission, meets requirements of the European Commission's Directive on Data Protection, which defines standards for privacy protection for European Union citizens' data.

The intentional or unintentional disclosure of private information as well as other risks—such as receiving malicious programs through email, the exchange of offensive material on email, and the loss of intellectual property—constitutes compelling reasons to filter content. The next section examines several methods for addressing these risks.

Email Content-Filtering Techniques

Email content-filtering techniques range from the relatively simple to the complex. The simplest techniques use word matching to determine whether a message contains inappropriate content. At the other end of the spectrum, sophisticated antivirus detection techniques can scan for large numbers of known viruses using efficient pattern-matching methods.

Email content-filtering techniques can be categorized into several groups:

- Term matching
- Regular expression matching
- Matching in context
- Malware scanning

Each of these techniques provides varying levels of protection at equally varying levels of complexity. (That is, with respect to the way these programs work, not necessarily with the complexity of installation and administration). They all, however, fit into the stream of email processing in similar ways—following the pattern that Figure 7.2 shows. (Malware scanning does not use a dictionary, but a database of patterns; the overall structure is the same, however.)

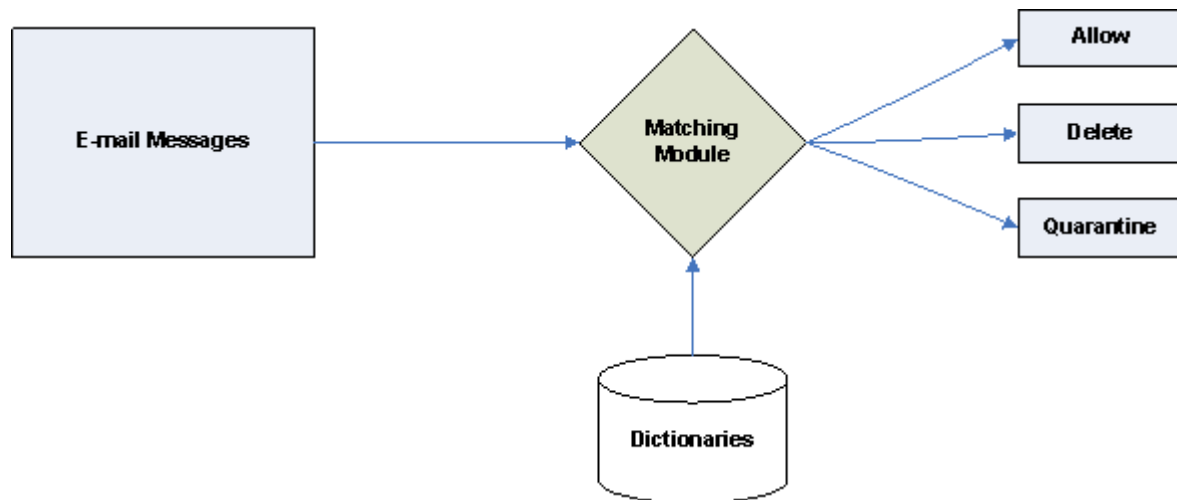


Figure 7.2: Email content scanning analyzes each message for particular patterns and carries out particular actions based on policies defined by email administrators.

Term Matching

Term matching is a process of scanning the contents of a message for particular words or terms that are deemed inappropriate. Terms are maintained in one or more dictionaries. The contents of a message are compared with the contents of the dictionaries to determine whether the message should be blocked. Although the process is relatively simple, there are a number of variations that vendors may use to improve the speed and accuracy of term matching. Because the methods used in commercial programs are proprietary, this discussion will focus on general techniques that may or may not be used in any particular system.

These techniques all determine whether a particular set of terms is present in a message. The term-matching program would then apply a calculation to determine the relative weight of the banned terms prior to making a decision to allow the message through or to handle the message in some other way. For example, if the term “gamble” is included in a dictionary of controlled terms, one occurrence of the term may not warrant blocking the message. One can easily imagine a message legitimately using that term in a metaphorical way:

```
From: Mary Jones <mjones@mycompany.com>
To: Kevin Johnson <kjohnson@mycompany.com>
CC:
Subject: Next Sales Call with Acme
```

Kevin,

I'm concerned about your approach on the next sales call with Acme. We can't gamble with this one - we need it to make this quarter's numbers. Call me to discuss.

Mary

However, in the context of other related terms also found in the dictionary, the cumulative effective of the presence of multiple terms can indicate inappropriate content:

```
From: gamble-while-at-work@onlinecasinoatwork.com
To: Kevin Johnson <kjohnson@mycompany.com>
CC:
Subject: 24-hour Online Casino

Blackjack! 30 kinds of Poker! American and European Roulette!
Slots and more!

Don't wait - Gamble while at work. Join us at
www.onlinecasinoatwork.com
```

Matching large numbers of terms against high volumes of emails requires efficient processing. Some techniques are:

- Dictionary lookups
- String matching
- Word stemming

Dictionary Matching

Dictionary lookups are simple but can be inefficient. This technique requires that a message be divided into lexical tokens as Figure 7.3 shows.

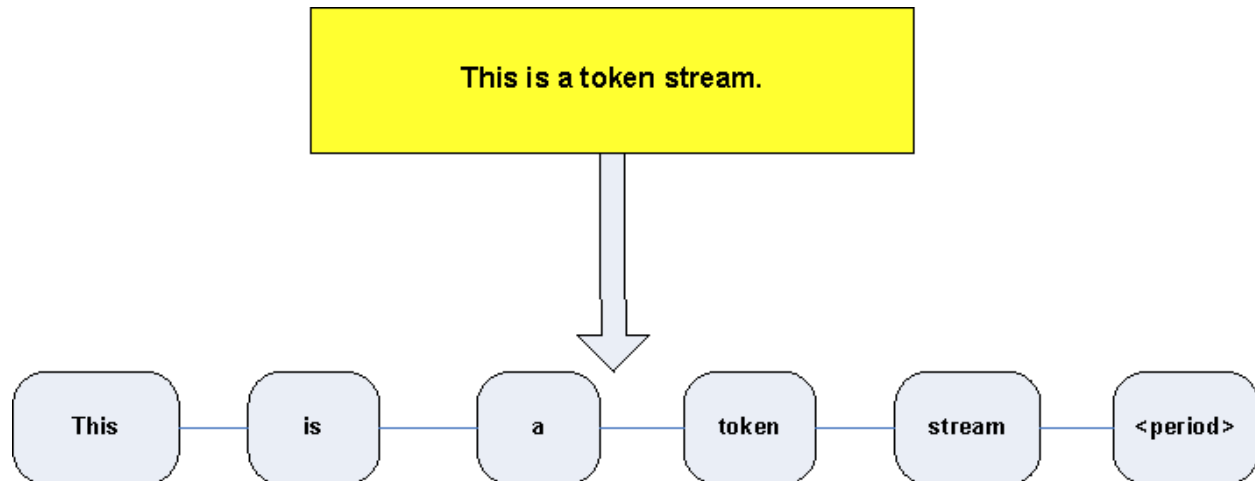


Figure 7.3: Lexical scanning divides a stream of text into individual tokens consisting of single words or punctuation marks that are further analyzed by pattern-matching techniques.

Each token is then compared with the contents of a dictionary and, if the word is present, the process takes that term into account in the cumulative measure of the message.

The process of looking up terms sounds simple, and it is, but when working with the volumes of text found even in small and mid-sized organizations, the time required to perform so many lookups can quickly add up. Without delving into too many technical details, consider three lookup methods.

- **Linear lookup**—Compare each term in the message with each entry in the dictionary. On average, the time to lookup a term in the dictionary is proportional to the length of the dictionary. This method is highly inefficient and impractical.
- **Binary lookup**—This approach uses a divide-and-conquer technique that allows terms to be found in logarithmic time. For example, if it takes 1 unit of time to search a list of 32 words, it will take 2 units of time to search 64 words, 3 units of time to search 128 words, 4 units of time to search 256 words, and so on. Although much better than a linear lookup, this approach can be impractical for large dictionaries.
- **Hash lookup**—This technique calculates a numeric value for each term that is used as an index to the dictionary. This technique requires about the same amount of time to lookup terms regardless of the size of the list. (Actually, the time can increase at some points when the dictionaries get very large, but in many cases, it is safe to assume a constant amount of time to perform the lookup).

The binary lookup works well with short lists and hash lookups work well with moderate-sized and large dictionaries. However, even the fastest of these techniques may not be optimal in all situations. Other techniques, based on string matching, may be more appropriate.

String Matching

In many cases, problems can be solved more efficiently by exploiting patterns in data. For example, if “gamble” and “gambling” are both words in a content-filtering dictionary, you can take advantage of the fact that five letters, “gambl,” appear in both and in the same order. One way to do so is to treat the content of the email message not as separate words but as a string of characters.

Rather than look up the work “gamble” in the dictionary, a content-scanning program could use the following rule:

Look for a “g” followed by an “a” followed by an “m” followed by a “b” followed by an “l” followed by either an “e” or an “i,” which is then followed by an “n” followed by a “g.”

This type of character-by-character pattern matching is quite an undertaking, so there are a number of highly efficient algorithms for pattern matching. One advantage of this approach is that it saves space in dictionary storage as well as provides a rapid method for matching terms. As Figure 7.4 shows, the terms “gamble” and “gambling” can be represented in a combined representation using a directed graph.

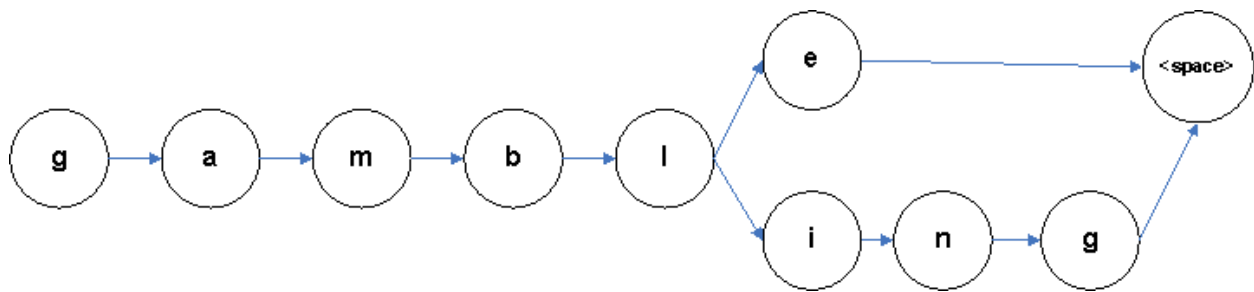



Figure 7.4: Multiple words can be efficiently stored and matched using a set of nodes and arcs.

 The formal term for this representation is a finite state machines (FSM). For more information about FSMs and string matching see <http://www.cs.princeton.edu/introcs/73fsa/>.

Word Stemming

Even with efficient storage and processing, the number of words in a dictionary can grow rapidly. For example, if an email administrator wants to block content related to gambling, the dictionary should include “gamble,” “gamble,” “gambled,” and “gambling.” A better approach is to use word stems, or roots of words, for matching. By using stemming, a single dictionary entry can be used to detect multiple forms of a word.

Limits of Term Matching

Term matching in its various forms is the most basic method for analyzing content. Although this technique is relatively simple to implement, there are considerable drawbacks. Term matching does not work well with misspelled words. In the examples earlier, the word “Gamlbe” would not be detected as a controlled term. Similarly, spammers commonly use punctuation marks to disguise words that readily trigger detection. Instead of using “Low Mortgage Rates” in a subject line, spammers might use “Low Mort!gagge Rat-es.” Trying to capture all possible variations and misspellings is impractical.

One method of addressing this problem is to use a *soundex* or similar function. The soundex function computes a string of characters for a word based on the sounds of the characters that make up the word.

How to Compute a Soundex

The soundex for a word is calculated as follows:

1. Capitalize all letters in a word and remove punctuation marks.
2. Keep the first letter of the word.
3. Remove all occurrences of A, E, I, O, U, H, W, and Y.
4. Replace each of the following letters with the corresponding number:
 - 1 = B, F, P, V
 - 2 = C, G, J, K, Q, S, X, Z
 - 3 = D, T
 - 4 = L
 - 5 = M, N
 - 6 = R
5. Remove all pairs of digits that appear next to each other.
6. Pad the string with 0s if necessary to create a four-character string, and return the first four characters, which should be the first letter of the original word followed by three digits. For example, the word “gamble” and “gambll” would both yield “G514” but the misspelling “gamlbe” yields “G541.”

A number of variations to the soundex algorithm have been developed as well as related algorithms.

Source: Soundex, Wikipedia, <http://en.wikipedia.org/wiki/Soundex>. For variations on the soundex algorithm, see “How To: Understanding Classic Soundex Algorithms” at <http://www.creativyst.com/Doc/Articles/SoundEx1/SoundEx1.htm>.

Even with the soundex algorithm, term matching as a content filter can generate a high number of false positives; that is, messages are identified as banned when in fact they are legitimate messages. To counter this problem, weighting schemes must be carefully crafted to include enough occurrences of controlled words to minimize false positives. This, however, introduces the chance of an unacceptable number of false negatives (messages that should be blocked but are categorized as legitimate). Alternatives to simple term matching, including regular expression matching and matching in context, seek to minimize these problems.


Regular Expression Matching

Regular expressions are patterns that can match multiple strings. Patterns are made up of letters, digits, and special pattern characters. The “*” character is a special pattern character that matches any number of characters. For example, rather than having a dictionary with the words “gamble,” “gambles,” “gambling,” and “gambled,” a regular expression pattern such as “gambl*” could be used to match them all.

Other commonly used special patterns include:

- [0-9] matches a digit
- [a-zA-Z] matches a letter
- \s matches a whitespace character, such as spaces and tabs
- \b matches a word boundary
- {n,m} used to modify another pattern, specifying that the pattern must occur at least *n* times but not more than *m* times

There are many other special patterns used in regular expressions that allow developers to craft precisely targeted patterns.

 For examples of regular expressions for matching strings, numbers, dates, email addresses, and other useful patterns, see The Regular Expression Library at <http://www.regexp.com/Default.aspx>.

An advantage of regular expressions over term matching is that regular expressions allow for matching more complex patterns of multiple terms. For example, a simple pattern such as

```
"Texas\s[hH]old*\s[pP]oker"
```

would match against “Texas Holdem Poker,” “Texas Hold’m Poker,” and similar variations. There would be no risk of having a legitimate message mistakenly categorized as a blocked message because it had both the terms “Texas” and “hold” in it, which might occur with a simple term-matching approach. Regular expressions are also useful for detecting well-defined numeric patterns, such as Social Security numbers, credit card numbers, bank routing codes, and so on.

A drawback of regular expressions is that, as the earlier example indicates, they can be cryptic and difficult to write. There is also additional processing time required to match regular expressions, and care must be taken when writing regular expressions to ensure that they are crafted efficiently.

Another limitation is that efficient regular expressions do not try to span too much text or capture too many patterns in a single expression. Thus, they work well for capturing the local context of a term (for example “Texas” near “poker”) but not for capturing the global context of a message. Additional techniques are needed for that.

Matching in Context

If a human were given the task of categorizing messages, the person would not be tripped up by minor misspellings, added punctuation marks, or controlled terms used in appropriate context. One of the reasons is that a human reader evaluates messages based on the broad context of the message, not just a limited string of characters within the message.

To improve the quality of email filtering, software must take into account the global context of a message. This is done in a number of ways:

- Rule-based analysis
- Categorization based on terms
- Categorization based on n-grams

Rule-based analysis complements categorization schemes and may be used in conjunction to improve performance.

Rule-Based Analysis

Rule-based analysis uses a series of IF-THEN rules that can examine both the structural elements and content of a message. Structural elements are characteristics of the message that include:

- The sender of the message
- The number of recipients
- The size of the message
- The number and types of attachments

These elements can be used to craft rules to allow or block particular messages. For example:

- If the sender is a member of the legal department, a message is allowed regardless of the content.
- If the number of recipients is greater than 10, block the message.
- If the file extension of a file does not match the MIME type detected by examining the contents of the file, quarantine the message.

Rules can also be based on the contents of a message. These rules could use a series of terms and Boolean operators to detect patterns indicative of inappropriate content. The Boolean operators could include AND, OR, NOT, and NEAR. Example rules in this category are:

- If “Texas” NEAR “poker,” block message
- If “breast” AND NOT “cancer,” block message
- If “gamble” AND “slot” AND “online,” block message

Rules that operate on the content of the message suffer from the same drawbacks as term-based matching; that is, misspellings and added punctuation marks can result in mistakes in categorizing messages. However, structural rules provide a mechanism for controlling filtering based on global properties and other characteristics not tied to terms used in the body of messages. To avoid the limits of targeted rule-based systems, email content can also be categorized by looking at the entire set of terms used in a message.

Categorization Based on Terms

Earlier, this chapter discussed the use of terms and dictionary lookups to determine whether a message should be allowed or managed in some other way (for example, deleted, quarantined, and so on). In that scheme, the scanning system was examining individual terms and determining whether the collective set of controlled terms in the message warranted blocking the message. Although limitations of this approach are well known, the use of terms for categorization is not without merit, you simply need to adjust how you use the terms.

To improve on simple term lookups, you can use categorization techniques that use *all* terms in a message to determine the classification of a message. The object is to measure both the frequency with which terms are used and the context in which they are used.

Suppose when an email message is scanned, it can be categorized into two groups: allowed or blocked. Also assume you have a large number of samples of messages that should be allowed as well as a large sample of messages that should be blocked. These messages can be used with algorithms known as *supervised learning algorithms* to train a scanning program with statistical measures for categorizing email messages.

A well-known and widely used method for categorizing email content, as well as other categorization tasks, is Naïve Bayes, also referred to as Bayesian categorization (see Figure 7.5). This algorithm uses examples to determine the frequency with which certain characteristics (such as terms) appear in relation to the likelihood that the message is in a particular category (such as allowed or blocked). For example, if the term “gamble” appears frequently in messages that are categorized as blocked and infrequently in example messages that are categorized as allowed, then a new message with that term is more likely to be classified as blocked.


 The details of Naïve Bayes are beyond the scope of this chapter. For more information about this approach, see http://en.wikipedia.org/wiki/Naive_Bayes.



Figure 7.5: Categorization algorithms, such as Naive Bayes, group messages based on shared characteristics with previously categorized messages.

Classification algorithms, such as Naïve Bayes, can use just about any characteristic of the objects they are classifying. (There are some restrictions related to the independence of these characteristics, but we will assume the restrictions are met for our purposes). When working with email messages, you can, of course, use terms, but then you run into the same problems: misspellings and added punctuation marks within words can throw off your measures. Fortunately, when working with messages with high error rates, you can use a simple variation of the algorithm to compensate for the problem.

Categorization Based on N-Grams

An n-gram is a sequence of characters extracted from another set of characters. The “n” in n-gram indicates the number of characters extracted. For example, “cat,” “ate,” and “teg” are all 3-grams. To determine the set of n-grams for a piece of text:

1. Start with the first letter of the text.
2. Select n characters, and output as an n-gram.
3. Move to the next character.
4. Repeat steps 2 and 3 until the end of the text is reached.


For example, the text

Don't wait - Gamble while at work.

Yields the following 3-grams:

"Don"	"on'"	"n't"	"t "	"t w"
" wa"	"wai"	"ait"	"it "	"t -"
" - "	"- G"	" Ga"	"Gam"	"amb"
"mbl"	"ble"	"le "	"e w"	" wh"
"whi"	"hil"	"ile"	"le "	"e a"
" at"	"at "	"t w"	" wo"	"wor"
"ork"				


The advantage of this approach is that the characteristics used are smaller than terms, so a minor difference in a term, such as a transposed pair of characters, will change some n-grams generated from that word, but not all of them. Similarly, with the use of punctuation marks in words, some n-grams will be different from those generated from the correct spelling, but many will be the same. N-grams help to dilute the influence of errors such as misspellings because much of the information about the word or term is reflected in a number of n-grams.

 For more information about how N-grams are used for categorization, see William B. Cavnar, John M. Trenkle “N-Gram-Based Text Categorization” at http://citeseer.ist.psu.edu/rd/43754390%2C68861%2C1%2C0.25%2CDownload/http://citeseer.ist.psu.edu/cache/papers/cs/810/http:zSzzSzwwww.info.unicaen.frzSz%7EgiquetzSzclassifzSzcvnar_trenkle_ngram.pdf/n-gram-based-text.pdf.

Filtering emails based on the content of messages can use a combination of term matching, structure-based rules, and categorization methods. When the task turns to detecting malware, specialized approaches are required.

Malware Scanning

Detecting malicious software in content streams, such as email, is particularly challenging. Certainly, spammers try to mask the contents of their messages in such a way as to avoid detection, but virus and worm writers have created far more sophisticated techniques for avoiding detection. Conventional text-scanning technologies, such as term matching and Naïve Bayes categorization, are not sufficient to combat the latest generation of malware. Two general approaches are used to detect malware: signature-based detection and behavior-based detection.

 This section briefly describes highlights of malware scanning; for details about the evolution of malware and countermeasures, see Chapter 3.

Signature-based detection uses a set of patterns that act like fingerprints to uniquely identify malicious programs. Antivirus vendors are constantly adding to their signature databases to keep up with emerging viruses, worms, and related malware. Once a signature has been identified for a malicious program, the code can be readily blocked at the network or desktop levels. As described in Chapter 3, virus writers have used code-changing techniques to avoid detection by signature-based systems. Antivirus researchers and developers have responded with another type of detection method, known as behavior-based detection.

In behavior-based detection, an antivirus program looks for specific behaviors likely to be found in malicious code but not as likely in legitimate code. These include making particular types of system calls and changing certain registry keys. Behavior-based detection may also simulate an execution of a program to determine the kinds of actions carried out by the program. Like other areas of email content filtering, malware detection is best performed with a combination of complimentary techniques.

Multiple Techniques for Email Content Filtering

Filtering email presents two opposing challenges: the process must be comprehensive and it must be efficient. If the process is not comprehensive, email that should be blocked will be allowed through; however, if the scanning is too strict, legitimate emails may be blocked. Finding the right balance between blocking banned content and not blocking allowed messages can require a combination of techniques, each with distinct strengths. Regardless of which techniques are used, they must be efficient. Even small and midsized organizations might have to manage large volumes of emails, so performance is also a primary concern. A specialized version of email scanning that often combines multiple techniques is anti-spam systems.

Filtering Spam

Spam requires a special case of email filtering. Email administrators have to deal with large volumes of unwanted, unsolicited email. In addition, spammers are constantly trying to evade detection, so anti-spam developers are engaged in a similar cat-and-mouse game that antivirus developers find themselves in. Spam is also an example of where multiple techniques work better than any single technique, in most cases. Although a spammer may be able to avoid detection by a Naïve Bayes filter by including long strings of nonsense text at the end of a message, a rule-based detection system or regular expression pattern matching might detect such tricks.

Policies and Actions for Filtered Content

After a message has been categorized as spam, offensive material, or otherwise banned content, the question arises, what to do with it? This is where policies are needed. Policies are sets of rules that define the action taken by the scanning system in response to a event, such as the detection of spam. Responses may include:

- Deleting an outbound message deemed in violation of company policy
- Sending a notification to the sender that a message was deleted because it violated company policy
- Quarantining an inbound message categorized as spam
- Deleting an inbound message categorized as offensive or otherwise inappropriate
- Re-routing a message deemed offensive or otherwise inappropriate to an administrator's queue for review and further action

Ideally, policy enforcement would be supported with robust reporting to allow administrators to track trends in spam, offensive content, and the types of actions taken in response. Of course, not all potentially offensive and inappropriate material enters the organization through email systems. Comprehensive content security must also include the ability to block inappropriate browsing on the Web.

Web Browsing and URL Blocking

Most organizations do not concern themselves with employees who spend a few minutes checking weather forecasts, traffic conditions, or reading the news online during breaks. There are, however, plenty of time wasting, inappropriate, and patently offensive sites on the Internet that should never be accessed from company computers. For those cases, URL blocking is a reasonable mechanism for controlling access to those sites.

As Figure 7.6 shows, when a browser makes a request to retrieve a URL, a URL blocking program first checks the URL against a database of banned sites. If the requested URL is in the database, the request is not sent on and, typically, a message is displayed notifying the user that the site requested is not allowed. If the requested URL is not on the list, it continues to be processed.

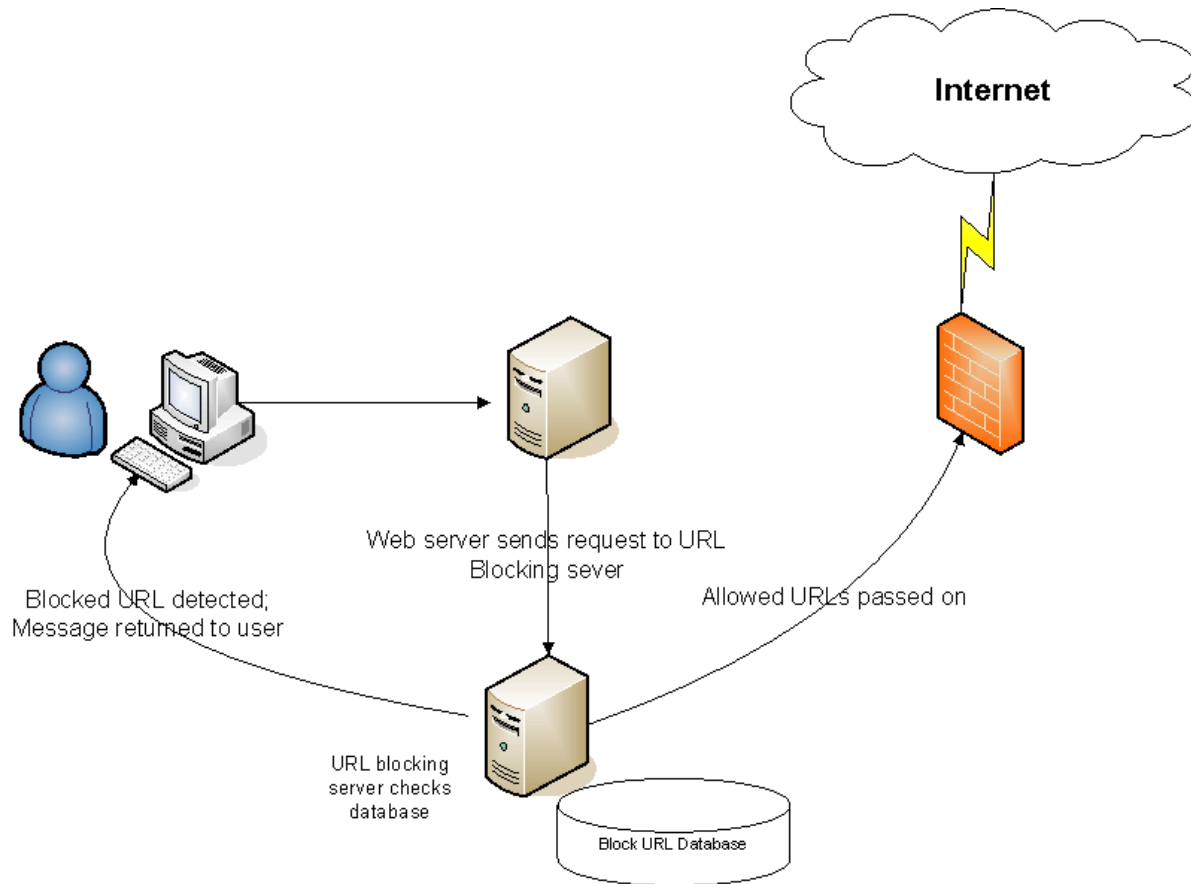


Figure 7.6: URL blocking intercepts requests for Internet resources and compares them with lists of banned sites.

The database of URLs may contain both white lists and black lists. White lists are known sites that users are allowed to access. The most restrictive form of URL blocking allows access only to sites on white lists. Black lists are compilations of URLs of known sites that are not allowed to users. Because the Web changes constantly and Web sites can easily move content from one domain to another, to be effective, URL blocking databases must be kept up to date in near real time. Vendors and some open-source projects maintain real-time black lists that are used in URL blocking systems. It is impractical for a single business to implement a full black list.

Securing content in an enterprise is a multifaceted challenge. Systems and network administrators must counter malicious software, inappropriate content, and the potential loss of proprietary and private information. Meeting these challenges is best done by implementing a general security practice known as defense in depth.

Multi-Layered Security: Defense in Depth

The defense-in-depth strategy acknowledges the fact that no single security technique or system can address all potential threats. For example, a properly configured firewall can block probes and attacks from outside the network, but someone with access to the internal servers and applications of a network will not be stopped by the firewall. You can broadly categorize security measures as applying at three levels:

- Gateway
- Network
- Desktop

Security at these three levels can provide a comprehensive, multi-layered defense.

Gateway Defenses

The purpose of gateway defenses is to prevent unwanted content from entering and to prevent controlled content from leaving a network. The first line of defense at this point is a network firewall. Firewalls perform packet filtering, monitor the state of communications between systems inside and outside the network, and, in some cases, provide basic access control and other security functions.

Just within the network perimeter is an ideal position for content filtering. At this point, inbound traffic has cleared the preliminary line of defense provided by the firewall; it is also the last point before the firewall to analyze outgoing traffic. A well-formed email message, for example, will be allowed through a firewall configured for SMTP traffic even though it might contain spam or malicious software. A content-filtering network appliance configured for anti-spam scanning or antivirus scanning could block the message before it reaches the email server.

Gateway defenses are barriers between the internal network and the Internet. Within the network, additional measures are required.

Network Security Measures

Within the network, security measures include two key activities—locking down servers and applications and detecting anomalous behavior within the network. Once an intruder has gained access to a network, either by breaching perimeter defenses or through legitimate access to the network for other purposes, the intruder can attack servers and the applications that run on them. Targets include:

- File servers containing proprietary documents
- Web servers that can be compromised and used for other purposes
- Database servers with customer information

Locking down these servers is essential. This task entails shutting down any unnecessary services, removing applications that are not essential (for example, production servers should never have compilers installed), ensuring that critical patches are installed, and ensuring that the operating system (OS) and applications are properly configured. Host intrusion detection systems should be used on critical servers to detect any unauthorized changes to system files or key data sources.


Identity management and access controls are also central to network security. Users should be given the minimum access required to do their jobs, and access to information should be granted on a need-to-know basis. Networks can be monitored with intrusion prevention systems to detect anomalous behavior, such as unusually high levels of SMTP traffic from several desktop devices—which could indicate compromised machines being used as bots for spamming.

Network security is a broad and deep discipline. Effectively securing content depends on well-defended network services.

Desktop Security Measures

The final layer of defense is at the desktop. Although gateway defenses and network security measures can provide countermeasures to many threats, some malicious code or damaging attack may reach the desktop. Such is especially the case for mobile devices that are not always connected to the network and therefore not continuously protected. At a minimum, desktops should be protected with

- Antivirus software
- Anti-spyware
- Personal firewall
- Automatic OS updates
- Vulnerability scanners for desktop systems, such as Microsoft Baseline Analyzer

 The Microsoft Baseline Analyzer can identify many known vulnerabilities, missing patches, and insecure configurations in Windows OSs. The tool is freely available at <http://www.microsoft.com/technet/security/tools/mbsahome.mspx>.

A multi-layered security scheme that includes gateway defenses, such as content scanning, as well as network and desktop security measures, is a critical element of effective security management practices.

Summary

There are a multitude of threats on the Internet. Businesses and organizations that leverage the benefits of the Internet must also contend with the risks. Effective content security requires the ability to scan content that enters and leaves a network to ensure that inappropriate, offensive, proprietary, or private information is not disseminated against organizational policies. Fortunately, a wide range of techniques, from basic term matching to sophisticated pattern matching, are available to systems administrators and managers responsible for securing an organization's content.

Chapter 8: Implementation Issues in Securing Internet Content

Securing Internet content in an enterprise is now a basic element of the broader information security practices of organizations. The Internet is now woven into the fabric of business much like telephones and shipping services; it is difficult to imagine doing business without it. At the same time, with the benefits of the Internet come the downsides: viruses, spam, phishing messages, potentially unwanted programs (PUPs), and time wasted browsing and downloading offensive material.

Throughout, this guide has examined the responsibilities of organizations to protect the integrity of their information and infrastructure, specific threats to that mission, and technologies for combating those threats. This chapter continues the discussion started in Chapter 7 about technologies for securing Internet content with an examination of the implementation issues associated with applying those technologies.

The particular topics addressed include:

- Criteria for choosing a secure content mechanism
- Benefits and drawbacks of implementation approaches
- Management issues in securing Internet content
- Best practices in securing Internet content

Let's begin with a discussion of the core features that a secure content system should support.

Choosing a Secure Content Mechanism

Ideally, a secure content mechanism will possess several characteristics:

- Provide comprehensive coverage against common threats
- Reside in a secure and reliable platform
- Support open standards
- Offer easily customized black lists and white lists
- Align with organizational structures and line of business needs
- Provide adequate reporting
- Prevent simple bypass techniques

Comprehensive Coverage

Network and systems administrators are well acquainted with common threats that use Internet content transmissions as a way into an enterprise: viruses, worms, Trojan horses, keyloggers, video frame grabbers, spam, phishing messages, and PUPs (Figure 8.1).

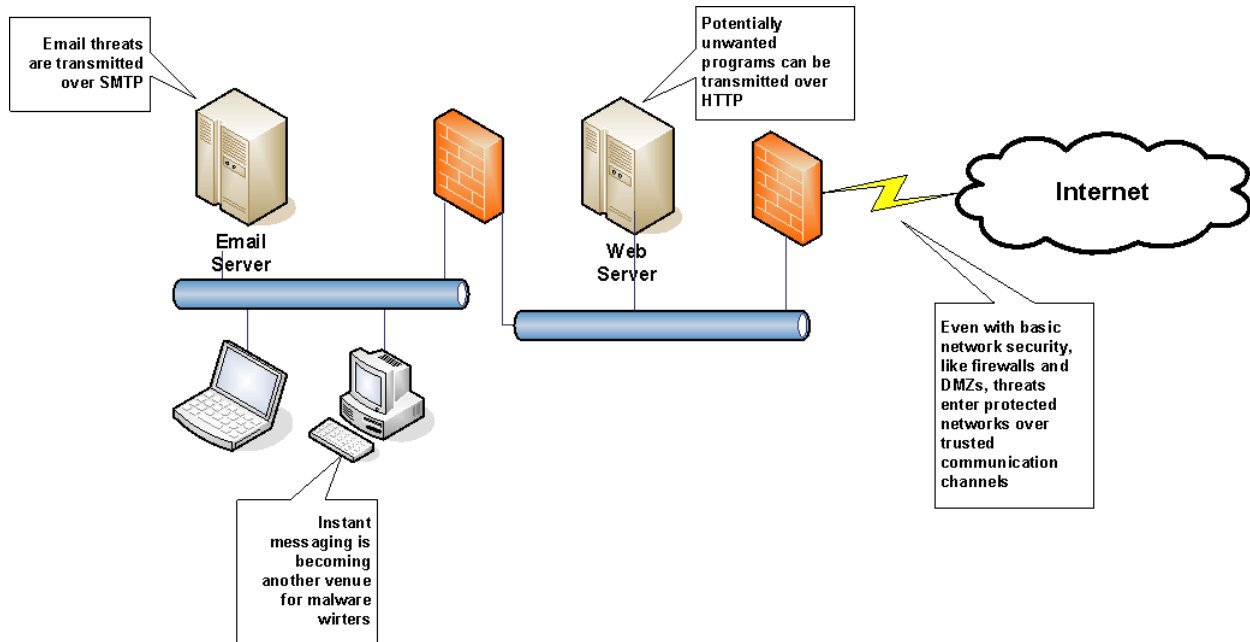



Figure 8.1: Content-based threats can endanger a secure network over trusted communications channels.

The problem of dealing with this breadth of threats is compounded by the fact that the threats can use multiple vectors, or means of entering a private network, including:

- Email messages
- HTTP communications
- Instant messaging protocols
- File transfers

Threats in Email Messages

Viruses and worms are probably the threats most commonly associated with email. Well-known email viruses and worms—such as Melissa, SoBig, and MyDoom—are carried in email messages over the Simple Mail Transport Protocol (SMTP). In addition to malicious programs, emails can carry banned content, such as MP3 files, video clips, and other large, non-business-related information into an organization.

 For details about viruses, worms, and other malicious programs, see the McAfee Virus Information Library at <http://vil.nai.com>.

Threats in HTTP Traffic

The complexity of communications over HTTP is growing as applications become more sophisticated. With the growing complexity of applications come new ways to transmit unwanted content:

- Spyware can be downloaded without a user's knowledge or explicit permission from Web sites when a user browses to that site (see Figure 8.2).
- Users may unintentionally install a Trojan horse program when installing what they assume to be a legitimate application.
- Even basic browser functionality, such as cookies, is now used in elaborate mechanisms for tracking users' behavior across multiple sites.

Because both SMTP and HTTP are fundamental protocols to core Internet applications, network administrators cannot simply block them at the firewall. Rather, the content that is carried on these protocols must be analyzed to prevent unintended uses. Other protocols, such as those used for instant messaging, are not as commonly required as HTTP and SMTP; however, in cases in which instant messaging is a required network service, secure content filters must work with those protocols as well.

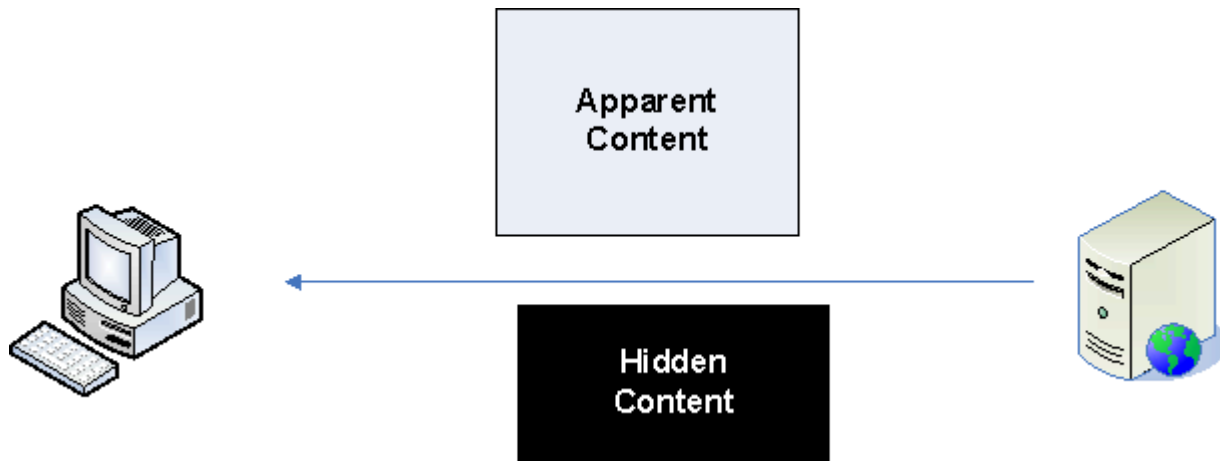


Figure 8.2: In addition to the apparent content that is downloaded when browsing the Web, other content, such as tracking cookies, may be downloaded without a user's knowledge or permission.

Threats in Instant Messages

As more mail traffic is scanned for viruses and similar malicious software, some malware writers are turning to instant messaging systems as a means for deploying their code. Some of the past threats to instant messaging include:

- The Kelvir worm struck the MSN Messenger tool in 2005. Kelvir displayed a Web site link in instant messaging chat sessions that when clicked would download a Trojan horse from the Web site.
- A similar worm, known as Lamo, worm struck the AOL instant messaging service in early 2006. Again, users were enticed to click a link that would have infected their machines.
- The Ocabot-F, similar to the others, enticed users to click a link, which then installed malware. In this case, once installed, the malware would send messages to all parties listed in the victim's instant messaging address book.

One option for administrators is to ban instant messaging from the enterprise network. This method would deny a valuable communication tool legitimately used in some organizations. In addition, some instant messaging clients allow users to use TCP port 80, used for HTTP traffic, bypassing administrator's attempts to prevent IM traffic.

Threats from File Transfers

Peer-to-peer, often referred to as P2P, file sharing has many advantages for groups collaborating or sharing resources, but the tools themselves may be problematic. For example, BitTorrent is a popular open source application for sharing programs as well as audio and video files. Because it is an open source program with freely available code and a liberal distribution license, BitTorrent has been repackaged—some distributors include spyware and possibly other PUPs in the repackaged version.

 For other file sharing programs that may include PUPs, see “Clean and Infected File Sharing Programs” at <http://www.spywareinfo.com/articles/p2p/>.

Besides the file sharing tools, the content that is transmitted may contain unwanted content. For example, sharing and storing MP3 files on company servers and desktops can consume large volumes of storage as well as network bandwidth.

A comprehensive, secure content filter should be able to detect malicious and other unwanted programs in emails, HTTP traffic, across peer-to-peer file sharing, and within instant messaging traffic. As the list indicates, there is no single “vulnerable” protocol that must be protected; any protocol and its associated applications are potentially vulnerable to misuse. The secure content mechanism, in addition to being comprehensive, must also be protected from tampering.

Secure and Reliable Platforms for Content Security

Secure content devices must be hardened to minimize the chances that the system is not compromised. Like other servers, a secure content device will run an operating system (OS) and multiple services. Each of these is a source of potential vulnerabilities that must be addressed. Common issues that must be considered to maintain the security of a secure content device include:

- Patching the OS
- Patching system applications
- Shutting down unnecessary services
- Protecting access control information, such as password files
- Reviewing audit logs

One of the advantages of using an appliance for secure content management is that the system is configured securely when shipped and maintained through automatic updates and patches. For those running a secure content application on a device managed internally, see the sidebar “Hardening Servers: Using Bastille to Lockdown Linux.”

Hardening Servers: Using Bastille to Lockdown Linux

Locking down, or hardening, an OS requires in-depth knowledge of system services, vulnerabilities, and attack methods. Fortunately, freely available tools, such as the Bastille Hardening program, can assist systems administrators with this task. The tool works interactively with systems administrators by prompting with questions about the servers use and the administrator’s needs. The tool then builds policies based on those needs. Bastille also supports an assessment mode that generates a report about available security configurations and changed settings.

Bastille analyzes and configures:

- Patches
- File permissions
- Account security
- Network and other daemons
- Sendmail
- Domain names
- Printing

The value of tools such as Bastille is twofold: first, it tightens the security of a server; second, it educates systems administrators about key OS security issues.

 For more information about the Bastille Hardening tool, see <http://www.bastille-linux.org/index.html>.

It is often said of IT that the only constant is change. With that in mind, a secure content system should support a wide variety of open communications standards to ensure that as users’ needs and usage patterns change, content remains protected.

Support Open Standards

The Internet is built on a number of open standards. To understand the complexities of securing content on the Internet, it helps to reference a standard model for internetworking services known as the Open Standards Interconnection (OSI) model. The model consists of seven layers, as Figure 8.3 shows.

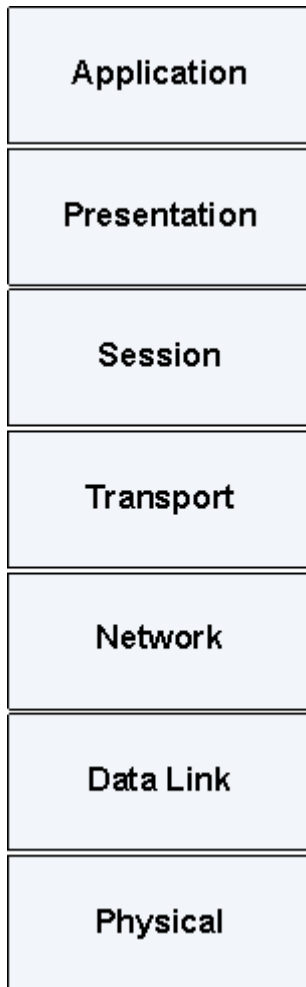


Figure 8.3: The OSI model highlights how higher-level networking services build on lower-level services.

The bottom layer, the physical layer, is responsible for converting bits into electrical signals and controlling certain physical aspects of the data. The data link converts messages into bits, controls how a computer accesses the network (for example, with collision detection for Ethernet), and converts frames between different types of networks. The networking layer provides logical internetworking services, routing, and addressing services. For the most part, secure content applications can ignore much of the complexity at these levels; information about operations at these levels is hidden by the protocols that run at the upper levels.

The Internet Protocol (IP) operates at the networking layer and underlies most of the protocols that are used by secure content applications. The transport layer manages transmission, message segmentation, integrity checking, flow control, and, if used in the protocol, sequencing. Both TCP and UDP function at this level. The session layer manages connections between applications. The presentation layer manages the translation of data into standard formats such as ASCII, EBCDIC, JPEG, and MPEG.

The top division, the application layer, provides the services that most users think of when they think of networking. Some of the common applications at this level include:

- SMTP
- File transfer protocol (FTP)
- HTTP
- Simple Network Management Protocol (SNMP)

Much of the focus of secure content management is on the application layer.



Instant messaging also falls into the application category but there is not yet a single protocol that is commonly used across instant messaging services; a number have been proposed—including Session Initiation Protocol (SIP), SIP for Instant Messaging and Presence Leverage (SIMPLE), and Application Exchange (APEX)—but have not been widely adopted. Thus, to work across different protocols, users need an instant messaging client that supports multiple protocols, as is the case with iChat, Gaim, and Trillian, or they must use a server, such as Jabber, that uses software bridges to provide support for multiple instant messaging protocols. The Jabber protocol, known as the Extensible Messaging and Presence Protocol (XMPP) is now an Internet standard.



For more information about Extensible Messaging and Presence Protocol (XMPP) as an Internet standard, see <http://www.ietf.org/rfc/rfc3920.txt>.

The services provided by the lower layers of the networking model are used in a variety of content transmission applications. Although email, hypertext, and file transfers have been popular tools for a long time (at least in terms of Internet history), and instant messaging has emerged as a valuable business tool, it is likely that other applications and new protocols, such as XMPP, will become commonly used communications tools. When that occurs, secure content applications that have worked well with other applications should readily adopt to protect the new means of communication. In addition to broadly advantageous characteristics such as comprehensive coverage, secure platform, and extensibility, secure content systems should have characteristics that support customizable functions that meet specific organizational needs.

Easily Customized Black Lists and White Lists

White lists and black lists are used to allow content from certain sources into an organization and to block content from other organizations.

Managing Black Lists

Secure content applications generally ship with defined black lists of known sites that provide online gambling, contain sexually explicit content, incite or promote illegal activities, contain illegal material, and promote hate speech or other patently offensive behavior. There may be occasions in which sites may need to be added or removed from a black list.

What is generally inappropriate for most business activity will have legitimate uses in some cases. For example, journalists and academic researchers may need access to sites professing hate speech, while the staff of a creative marketing firm might need access to lingerie catalogs online when developing a campaign for a fashion industry client. A “one-size-fits-all” approach rarely works in IT and secure content management is no exception.

Web sites are easily changed, so keeping up with all the potentially inappropriate sites is virtually impossible. Automated tools for collecting the URLs of sites containing offensive content and user-submitted sites to Internet repositories of black lists help, but they are not guaranteed to find all problematic sites. Network administrators should have the ability to easily add URLs to the black list as needed. In addition to keeping track of which sites should be blocked, administrators do not want to block legitimate communications.

Managing White Lists

White lists are used to define sources of content that are allowed to transmit information to the organization. Manually constructing white lists can be time consuming, but application-specific techniques can speed the process.

Lists of contacts maintained in email clients are obvious sources for white list content. If a person is listed in an employee’s contact list, it is reasonable to assume that the person has legitimate business with the organization and should have his or her email sent to the recipient without blocking.

Web server log files can be another source of information for building URL white lists. For example, by analyzing Web server logs, administrators may be able to determine that the majority of Web traffic at their site is generated by a few hundred sites. By reviewing the list of most popular sites, an administrator can construct a white list of known popular and verified sites.

Black lists and white lists are well-established tools in the area of secure content management. The ease of customization, along with the size of the database of black listed URLs, is an important consideration when choosing a secure content system.

 For more information about white lists and black lists, see Chapter 7.

Just as organizations will have differing needs with regard to black lists and white lists, they will have varying needs in terms of their organizational structure.

Align with Organizational Structures and Line of Business Needs

In this chapter and throughout this guide, the terms *system manager* and *network manager* have been used to refer to IT personnel responsible for network services and application management. There has been little, if any, distinction regarding the scope of these managers' responsibilities.

In practice, particularly in large organizations, there are many systems and network managers. Some are responsible for managing email servers, others maintain Web servers, and still others install, configure, and monitor database servers. Each of these has a role in securing content. As Figure 8.4 shows, administrators responsible for maintaining email systems may belong to different parts of the IT organization than those that control the Web servers. Securing email traffic (SMTP content) and Web traffic (HTTP content) should fall under the control of their respective departments or groups.

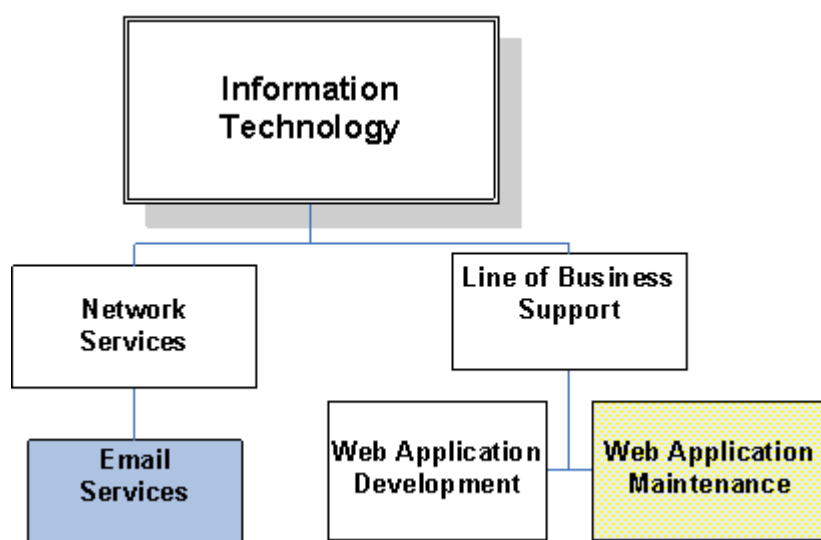


Figure 8.4: Secure content systems should be managed along the same organizational lines as IT responsibilities.

When choosing a secure content system, it is useful to partition the responsibilities of the secure content application along organizational lines. For example, one set of administrators may be responsible for mail servers, another is responsible for Web servers, and yet another group manages firewall and router configurations. The email managers may want full control over the secure content application that analyzes SMTP traffic, while the Web server managers expect total control over HTTP traffic; they both need to coordinate to some degree with firewall and router managers.

Provide Adequate Reporting

It is not enough for a secure content system to block targeted material; it must also be able to report to systems administrators about the status of the application as well as measure the state of the network traffic. At the least, secure content systems should report on:

- Application status
- System performance
- Significant event logs
- Configuration and update status

These are the core attributes systems administrators will need to monitor.

Application Status

The application status is essentially a snapshot of the state of content processing (see Figure 8.5). The overall status of the system consists of measures about each protocol that is analyzed by the application. Protocol measures examples include:

- Number of viruses detected
- Number of email messages deferred
- Number of email messages quarantined
- Reason for quarantine, such as virus detected or spam detected
- Volume of traffic processed in a given period of time

These measures are closely related to another set of key measures, system performance.

Technical Support | Submit a Sample | Virus Information Library | Remote Access Web Interface | About The SCM appliance | Resources | Help To

McAfee® Secure Internet Gateway v4.2 (3200)

192.168.10.40

Monitor
Policy
Configure
Update
E-Mail
System
Network
Troubleshoot

Home
Show Quick Help

System Status

Reset Counters Settings

Protocol Status (Counters Reset Mar 15 2006 04:53:07)

SMTP Viruses Detected	From inside 0 : From outside 0	E-Mails Deferred	104889
POP3 Viruses Detected	From inside 0 : From outside 0	Viruses Quarantined	0
HTTP Viruses Detected	From inside 0 : From outside 0	Content Quarantined	0
FTP Viruses Detected	From inside 0 : From outside 0	SMTP Spam And Phish Detected	0
ICAP Viruses Detected	From inside 0 : From outside 0	SMTP Spam And Phish Blocked	0
Total Viruses Detected	0	SMTP Spam Blocked By RBL	0
Total Unwanted Programs Detected	0	POP3 E-mails Received	0 / Hour (0 received)
SMTP E-mails Received	77.8 / Hour (104890 received)	ICAP Traffic	Not Available
HTTP Traffic	0.011 / Hour (15259393 bytes received)	URLs Blocked	4
FTP Traffic	0 / Hour (0 bytes received)		

Dashboard

SMTP Health	●	Scanning Partition Used	● 6% (7239.3 MB / 50
POP3 Health	●	Logging Partition Used	● 6% (13766.1 MB / 4
HTTP Health	●	Quarantine Partition Used	● 6% (14034 MB / 31
FTP Health	●	Deferred Partition Used	● 6% (19141.6 MB / 4
ICAP Health	●	Var Partition Used	● 11% (2422.3 MB / 4
Memory Swap Rate	● 0 /sec	Web Acceleration Health	●
Load Average	● 0.10	Web Throughput Utilization	● 0.00%
Processors Used	● 1%		

Apply All Changes Cancel All Changes Print Logoff

Figure 8.5: A secure content application should provide aggregate reporting on system status to allow administrators to quickly identify problem areas.

System Performance

System performance monitoring should allow administrators to select measures of interest, such as number of email messages analyzed, number of spam messages blocked, and number of viruses detected. Ideally, administrators will be able to specify how often the measures are updated and how they are displayed (for example, in table or chart form).

Significant Event Logs

Secure content systems analyze multiple protocols and, in the course of those functions, create events that systems administrators should be aware of:

- Events triggered by management events, such as logging in to the appliance
- Software update events
- Failures to communicate between clients and servers
- Events created when email messages are deferred, quarantined, or deleted
- Events created when a virus, spam, spyware, or other PUP is detected
- URL blocking events

Reports on significant events should be available on an on-demand basis or reported automatically, for example, as an email message to an administrator. Systems administrators should have the ability to configure the reporting mechanism to best suit their needs.

Configuration and Update Status

Another area of reporting focuses on system configuration. At any time, systems administrators should be able to review key configuration parameters, such as the date and time of the last software update. Changing system parameters should also trigger events that are tracked in the system event log. Configuration information, such as port settings (see Figure 8.6), should also be readily available.


Advanced - Connection Settings			
Intercept Ports	Intercept Ports	Add	Ports on which to intercept SMTP traffic
	110	Modify	
		Delete	
Listen Ports	Listen Ports	Add	Ports on which to listen for e-mail messages
	110	Modify	
		Delete	

Figure 8.6: Configuration settings, such as the ports with intercepted traffic, should be logically organized for easy access by systems administrators.

Effective reporting on content analysis, the results of that analysis, and the configuration of the secure content system is a fundamental requirement for systems administrators. Reporting on application status, system performance, significant events and configuration status covers a basic set of reporting requirements for system managers.

Features such as reporting, adaptability, and comprehensive coverage of threats are obvious aspects along which one might compare different secure content systems. Another characteristic that is not as often discussed is the way the systems may be bypassed and thwarted.

Compensating for Simple Bypass Techniques

 This section discusses techniques that can bypass some secure content applications. The objective is not to explain how to bypass but to point out to network and systems administrators the limits of secure content systems. In this section, the content tries to strike a balance between informing administrators of potential weaknesses and disclosing information that could be readily used to bypass such systems. For this purpose, high-level descriptions of techniques are provided but detailed procedures are not.

The history of information security is riddled with examples of countermeasures put in place to neutralize threats only to have those countermeasures rendered ineffective. For example, when signature-based antivirus programs were made available, virus writers responded by encrypting viruses. Antivirus developers then deployed tools to detect encrypted viruses, and virus writers again responded with even more sophisticated techniques, including the development of polymorphic viruses. The lesson is not that deploying countermeasures is futile but that even with countermeasures in place, you must be aware of ways determined individuals can circumvent those countermeasures.

Methods for bypassing secure content mechanisms include:

- Using IP addresses instead of domain names
- Using anonymous proxy sites
- Passing banned pages through a Web service provider
- Remotely controlling a device off the network with a remote control package

These are simple techniques that have equally simple countermeasures. For example, lists of banned sites should include both IP addresses as well as their domain names. Similarly, popular anonymous proxy sites can be banned along with other categories of unwanted content.

In the case of Web service providers, such as aggregators or search engines, it may be possible to access banned content. For example, to access the home page for [Realtimepublishers.com](http://www.realtimepublishers.com), one could browse to:

<http://www.realtimepublishers.com>

but if that were blocked, one might be able to retrieve the cached version from Google using the following URL:

<http://72.14.203.104/search?q=cache:UjoNwhAmuLcJ:www.realtimepublishers.com/+realtimepublishers&hl=en&gl=us&ct=clnk&cd=1&client=firefox-a>

Secure content devices could undermine this technique by scanning the full length of a URL looking for well-formed embedded URLs passed as parameters to a site. This countermeasure would work with other Web service providers, such as translation services and redirectors.

With enough time, money, and expertise, most security systems can be circumvented. The goal is not to find a secure content tool that can never be bypassed but to find one that adequately addresses the threats present in an organization without costing more than the value of the resources protected. When choosing a secure content application, network and systems administrators should ensure that simple countermeasures cannot undermine the effectiveness of the system.

Key Features of a Secure Content System

Up to this point, this chapter has summarized the key features one should look for in a secure content system, including comprehensive coverage, a secure and reliable platform, support for open standards, easy customization, the ability to align technical deployment along organizational structures, adequate reporting, and countermeasures to bypass techniques. These are essentially the “whats” of secure content systems; the next section will examine the “how” or implementation options of such applications.

Implementation Options of Secure Content Systems


The market is currently providing three types of secure content systems:

- Appliances
- Software applications
- Services

Each has advantages and drawbacks.

Implementation Option 1: Secure Content Server Appliance

A server appliance is a bundled solution that includes hardware and software designed for ease of installation and maintenance. A key advantage of a server appliance is that standardized hardware and software is configured in an optimal way prior to shipping. System and network administrators do not have to learn and implement lengthy installation and configuration procedures.

 This discussion focuses on server appliances, which should not be confused with network appliances, also known as thin clients. Network appliances are low-cost personal computers (PCs) with minimal hardware (for example, no disk drive or CD drive). Software is downloaded from the network as needed.

Standardized Hardware

In the case of a secure content server appliance, the system would be configured with enough processing power, memory, and secondary storage to meet specific performance ranges. For example, an entry-level secure content appliance might use:

- A mid-range Celeron processor
- 512MB RAM
- A 60GB to 100GB disk drive
- Dual high-speed network interfaces

This configuration could not process high volumes of traffic and would not provide reliability features needed in a high volume, mission-critical environment. Higher-performance appliances would use server-scale processors and improve reliability using components such as:

- One or more Intel Xeon-class processor
- 1GB to 4GB RAM
- Hot-swappable SCSI drives in a RAID configuration
- Dual power supplies
- Dual high-speed network interfaces

Appliance servers typically also have standard video interfaces, USB ports, CD or DVD drives, and other equipment that does not directly impact the content processing performance of the device but are used for monitoring, management, and maintenance. Standard rack-mountable chassis make appliances easy to add to existing hardware environments. In addition to standardizing hardware, appliance servers also standardize software.

Standardized Software

Standardizing software can significantly reduce the time and resource required to manage systems. Of course, there are different levels of standardization. At one level, an IT department might select a family of software as a standard. One company might standardize on Microsoft software but run Windows 2000 (Win2K), Windows XP, and Windows Server 2003 (WS2K3). Another organization might standardize on Linux for servers but run Red Hat, SUSE and Debian. Although similar, the OSs with the Windows and Linux categories can be different enough to require slightly different configurations, different patches, and may suffer from different security vulnerabilities. For a systems manager, these differences add up to managing multiple types of implementations in spite of similarities.

At another level of software standardization, an organization could select one OS and one set of software applications. For example, a Microsoft shop might standardize on WS2K3, SQL Server 2005, and Internet Information Services (IIS) 6.0. Even in this scenario, different hardware and varying application requirements can lead to different configurations. One server may require an FTP server to run while another does not; one server might run applications needing one version of .NET components while another server needs a different set. Again, even though the organization has standardized on software, no two servers are guaranteed to be configured identically.

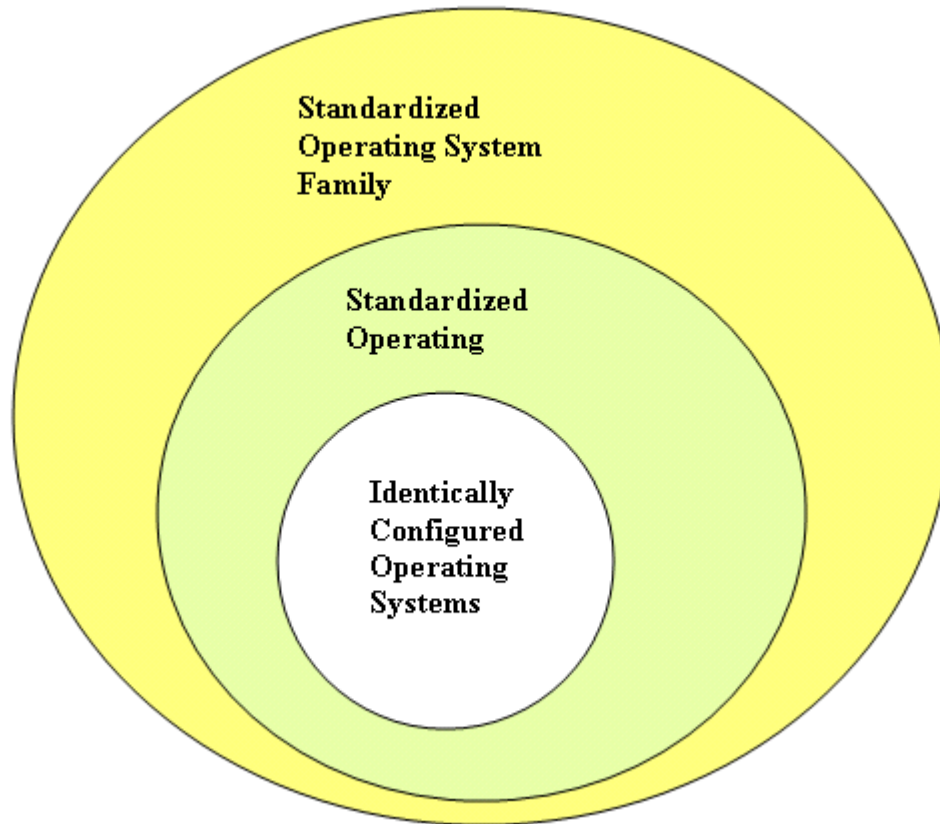


Figure 8.8: Standardizing software can mean several things; the more similar the instances of software installations are, the greater the benefits of standardization.

Another level of standardization is when the software installed on different machines is identical. This setup is virtually impossible unless the hardware and OS are identical. For example, a secure content appliance is a mass produced product; the vendor will use the same processor, video controller, disk drive, and other hardware components. These appliances can then use the same drivers for these components, run the same OS and applications—such as anti-spam filtering software—and be maintained at the same patch level. Standardizing on hardware and software results in significant benefits to appliance users.

Advantages of Appliances

The benefits of an appliance help both vendors and appliance users. Vendors do not have to troubleshoot problems in unusual or rarely used configurations, there is no need to support multiple OSs, and the core application, in this case secure content applications, can be thoroughly tested on a single target platform.

Appliance users benefit from that fact that the appliance is preconfigured at the factory, so minimal configuration is required. The hardware installation is minimal; typically, it is a matter of installing a rack-mounted device. Software maintenance and patching can be automated by the vendor, easing the workload of administrators. Finally, as network traffic increases or if network architecture changes and additional appliances are needed, identically configured devices can be added to the network without adding significantly to maintenance.

Implementation Option 2: Software Applications for Secure Content Management

Securing content depends on the use of multiple applications, including antivirus, anti-spam, anti-phishing, and URL blocking software and other programs. With software at the heart of securing content, one option is to deploy secure content applications on existing hardware.

One advantage of this approach is that it combines multiple functions on a single hardware platform. Systems administrators have fewer machines to manage and there is potentially less capital cost.

There are two primary disadvantages to a software-on-shared-server approach. The first is that because the application is not running on a dedicated server, performance may vary. For example, if a content filtering application were run on the same hardware as a Web server, when the demand for Web access is high, both the Web server and the content filter would have large workloads and would compete for the same computing resources. If the content filtering application shared a server with an application with peak demand periods—for example, a data exchange server that replicated data to remote offices several times a day—performance of the content filter would vary according to the other application's schedule.

The second disadvantage is that there may be conflicts in software dependencies between applications. For example, one application may require a particular patch to a shared library to correct a bug, but that same patch is incompatible with the secure content application. In another case, the optimal system parameter for one application is not optimal for other applications. As a result, systems administrators have to try to balance the requirements of each application with the relative value of each application to the organization. Some questions systems managers will confront include:

- How do I prioritize processes from different applications? Should they all be equal or should one application have a higher priority over another?
- If multiple software library versions are required, how should they be organized?
- If two applications have different administrators and both require root privilege, is it acceptable for each administrator to have potential control over the other application?
- How can upgrades be scheduled for one application to minimize the down time of the other applications?
- If additional capacity is required, should one application move to another server?

Servers that support multiple applications certainly have their role in IT. They are ideal in development environments, in remote offices that do not warrant multiple servers, and—when virtualization is available—in some cases in which dedicated servers (or at least virtual servers) are needed.

Does Hardware Virtualization Provide the Best of Both Worlds?

Virtualization is a method for making a single hardware platform appear as if it were several different platforms. This has traditionally been accomplished using software such as VMware (<http://www.vmware.com/>), but hardware vendors are adding support for virtualization to processors. (See for example, an introduction to Intel's virtualization technology at <http://www.intel.com/cd/ids/developer/asmo-na/eng/218153.htm?page=4>.)

Virtualization technology solves problems related to software dependencies because applications with conflicting configuration requirements can be run in different virtual machines. Unfortunately, virtualization does not solve the problems related to sharing computing resources among multiple applications.

In addition to appliances and software-only solutions, some enterprises may consider service providers for content filtering.

Implementation Option 3: Outsourced Service

Rather than install an appliance or software on a non-dedicated server, organizations may consider using a service that provides secure content filtering. This option is especially appropriate when email services are hosted by a third party or if an organization generates small to moderate amounts of Internet content.

A basic email filtering service works by routing all inbound and outbound email through a third-party's content filter mechanism. This approach shares several characteristics with an appliance-based approach, especially minimal maintenance. With well-defined and enforced SLAs, outsourced providers can meet the performance demands of an organization. As one of the service provider's core competencies is content filtering, one would expect that they would keep URL filtering black lists up to date, provide for white lists, maintain secure servers, and deploy the latest antivirus systems.

A disadvantage of services is that content has to be sent to a third party site before it is sent on to its ultimate destination. If the content filtering appliance or server is down, content will not go through. Because the server is not managed internally, the IT department cannot respond directly to the problem. Business continuity planning will have to take into account this potential disruption and formulate options, such as allowing email through without filtering or switching to a backup third-party service.

Specialized services, such as email-only content filtering provided with email hosting services, will not protect other content sent via ftp, HTTP and other protocols. Finally, services may use a single secure content appliance or other server to filter traffic from multiple customer sites. Performance may vary unless strict SLAs are in place. Organizations with modest amounts of content and a tolerance for some degree of disruption and possible swings in performance may be best suited for a service-oriented approach to secure content filtering.

Choosing Among the Options

It should be no surprise to anyone in IT that there is no single secure content option that is best for every situation. The following lists summarize the advantages and disadvantages of each of the three approaches.

Secure Content Appliance Server

Advantages

- Known performance
- Stability
- Easily scaled
- Minimal software (fewer potential vulnerabilities)
- Consistent performance
- Automated maintenance

Disadvantages

- No control over software installed on appliance

Software Applications

Advantages

- Able to use existing, underutilized servers

Disadvantages

- Performance can vary
- Must balance conflicting dependencies with other software
- Must lock down and maintain security of server OS
- Internal staff responsible for upgrades, patches, and so on

Secure Content Service

Advantages

- No additional servers on site
- Management is outsourced

Disadvantages

- Performance may vary
- Service may not filter all protocols used to transfer content
- Additional complexity of business continuity planning

In general, secure content appliances provide a balance between the minimal maintenance provided by service providers and the flexibility of installing software on existing hardware. Secure content services are appropriate for small organizations; software on shared servers is best for organizations with the staff and resources to maintain both the content filtering system and the security of the OS. Secure content appliances will well serve other organizations. Regardless of which secure content method is used, there are a number of management issues that will have to be addressed.

What About Filtering at the Client?

We have not discussed client-based filtering because, as a primary solution, it is not appropriate for enterprise-scale networks. Client-based filtering is popular with home PC users who are concerned about malicious software, PUPs, and inappropriate content.

Desktop antivirus software, personal firewalls, and similar products are certainly appropriate for enterprise use; they just do not constitute the full solution to the secure content problem. These applications play an important role in the “defense-in-depth” strategy used in security conscious organizations. For example, laptops may be protected by network-based services when connected to the corporate network, but when accessing the Internet from employee’s homes or at coffee shop, they must be protected with client-based applications.

For enterprise network managers, desktop solutions present two problems. First, they can be bypassed or turned off by determined users. Second, deploying and maintaining a large number of desktops is more challenging than updating a single appliance. For example, even if a systems administrator could push a patch or antivirus signature file to every client on the network, the administrator still needs to account for devices not connected at that time and deliver the new software when they connect to the network. Of course, by that time, the device could already be infected with the malware the patch was designed to counter.

Management Issues in Secure Content Management

When securing content, IT managers should consider:

- Appropriate use
- Auditing and reporting
- Quarantining and deleting content
- Black list and white list maintenance
- Capacity planning

Management decisions about these areas should be formalized in policies and procedures.

Appropriate Use

Appropriate use policies define what employees, contractors, business partners, and other users of IT resources are allowed to do in general terms. (Appropriate use does not delve into detailed access control issues, such as which users have write access to the Human Resources database). These policies should describe what types of Web browsing are allowed, what are legitimate uses of email, the boundaries of personal use of IT resources, and the consequences of not adhering to the policies.

Auditing and Reporting

Systems administrators should be aware of significant events, such as a spike in virus-ridden emails or a large number of attempts to access a blocked site. They should also establish regular procedures for reviewing secure content filtering logs to stay abreast of patterns in usage and detect emerging problems. In addition, industry-specific regulations may require further auditing and reporting procedures.

Quarantining and Deleting Content


When questionable content is detected in the email stream, Web traffic, or other protocol, the content should be isolated by either sending the content to a quarantine area, deleting the content, or blocking further access to a problematic Web site. System managers should establish policies for how long to keep quarantined content and how to notify users if their messages or other content has been blocked, quarantined, or deleted.

Black List and White List Maintenance

Black lists are often updated automatically by secure content vendors, but organization may add their own set of sites to those lists that are publicly available. They may also want to allow access to sites that are typically blocked on black lists. In the case of white lists, it is important for organizations to track which sources are essentially allowed a free pass to send content to their networks and update that list regularly.

Capacity Planning

As the number of users grows and new Internet-based services are provided and adopted, the volume of content passing through an organization's network will grow. While planning for additional servers, routers, firewalls, and other network equipment, network administrators should plan for additional appliances, servers, or services to meet the demand.

 A comprehensive security plan is based on well-defined policies. For help developing these policies, see the SANS Institute's Security Policy Project at <http://www.sans.org/resources/policies/>.

Best Practices in Secure Content Management

The purpose of this guide has been to describe the problems facing organizations when dealing with Internet content, and outline solutions to those problems. This chapter concludes with a brief overview of some best practices to incorporate with your secure content strategy.

First, deploy comprehensive content filtering. Even when client-side applications are used to prevent infections by viruses, worms, and other malware, these tools complement, they do not replace, network-based content filtering. Comprehensive filtering includes blocking malicious software, blocking spam and phishing attempts, and preventing access to inappropriate Web content.

Second, organizations should develop and enforce secure content policies. Users should be aware of what is considered appropriate use and the consequences of not following those policies. Policies should be reviewed periodically and adjusted to changing requirements.

Next, secure content administrators should align filtering practices with specific organizational goals, including:

- Protection of information assets
- Efficient operations
- Maintaining a non-threatening work environment
- Protecting against loss of intellectual property, proprietary data, and private information
- Maintaining regulatory compliance

The combination of comprehensive coverage and well-defined and enforced policies aligned with broader organizational goals and responsibilities underlie successful secure content practices.

Download Additional eBooks from Realtime Nexus!

Realtime Nexus—The Digital Library provides world-class expert resources that IT professionals depend on to learn about the newest technologies. If you found this eBook to be informative, we encourage you to download more of our industry-leading technology eBooks and video guides at Realtime Nexus. Please visit <http://nexus.realtimepublishers.com>.