Making Everything Easier!™

IBM Trusteer 2nd Edition

Stopping Zero-Day Exploits

FOR DUMMIES®

A Wiley Brand

Learn:

- About the dangers of advanced malware and zero-day threats
- How to break the threat life cycle and protect user endpoints

Compliments of

IBM

Peter H. Gregory, CISA, CISSP, CRISC

Stopping Zero-Day Exploits

FOR DUMMIES®
A Wiley Brand

IBM Trusteer 2nd Edition

by Peter H. Gregory, CISA, CISSP, CRISC

With contributions from:

Dana Tamir
Trusteer Director of Enterprise Security, IBM

FOR DUMMIES®
A Wiley Brand

Stopping Zero-Day Exploits For Dummies®, IBM Trusteer 2nd Edition
Published by
John Wiley & Sons, Inc.
111 River St.
Hoboken, NJ 07030-5774
www.wiley.com

ISBN 978-1-119-03864-1 (pbk); ISBN 978-1-119-03781-1 (ebk)

Manufactured in the United States of America

10 9 8 7 6 5 4 3 2 1

Publisher's Acknowledgments

Some of the people who helped bring this book to market include the following:

Project Editor: Jennifer Bingham
Acquisitions Editor: Amy Fandrei
Editorial Manager: Rev Mengle
Business Development Representative: Sue Blessing
Project Coordinator: Melissa Cossell

Introduction

Zero-day malware attacks and advanced persistent threats (APTs) are growing, serious threats to organizations. Cybercriminal organizations seem to be more motivated (and more skilled) every day.

Malware's advanced evasion techniques are making detection solutions ineffective for preventing infections. Advanced information-stealing malware utilizes ever-advancing techniques for exploiting application vulnerabilities, infecting targeted endpoints, and stealing information.

Most security experts today agree that threat detection is no longer the answer. Traditional detection systems are declining in effectiveness. Antimalware programs block only a minority of malware. Despite improvements in endpoint deployment tools and patch management processes, most organizations still take weeks or longer to deploy critical security patches. And cybercriminals continually develop new methods for bypassing detection rules.

This book discusses zero-day exploits and additional threats that are used to compromise enterprise endpoints and enable APTs and targeted attacks. It describes a promising new approach that uses multilayered defenses and breaks the threat life cycle at strategic chokepoints to provide effective yet transparent protection to enterprise endpoints.

About This Book

Malware and zero-day threats, which enable targeted attacks and advanced persistent threats, have advanced so quickly that most people are unaware of their stealthy and potent techniques. That's why the first three chapters describe today's malware problem in lurid detail (but this book is safe to keep around the house even if you have children).

The next chapters explore a new approach used for blocking exploits and malware infections and preventing malware from compromising user endpoints. And no *For Dummies* book is complete without a top-ten chapter; here we explore many considerations to keep in mind when exploring advanced threat prevention solutions.

This book was written with and for IBM Security Trusteer and also covers some of its technology.

Icons Used in this Book

Icons are used throughout this book to call attention to material worth noting in a special way. Here is a list of the icons along with a description of what each means.

Some points bear repeating, and others bear remembering. When you see this icon, take special note of what you're about to read.

Watch out! This information tells you to steer clear of things that may leave you vulnerable, cost you big bucks, suck your time, or be bad practices.

This icon indicates technical information that is probably most interesting to technology planners and architects.

If you see a Tip icon, pay attention — you're about to find out how to save some aggravation and time.

Chapter 1

Examining the Threat Environment

In This Chapter

- Understanding the players and motivations behind cybercrime
- Getting the point about advanced persistent threats
- Why zero-day attacks are a growing concern
- Balancing security, usability, and manageability

Millions of organizations and billions of people are connected to the Internet in order to conduct business, communicate, find information, and buy and sell goods and services of every kind.

But the criminal element is also participating in the global Internet, skimming away profits and leaving victim organizations in its wake. The nature of this crime has changed dramatically over the years. Today, hackers in criminal organizations use advanced methods to exploit vulnerabilities in end-user applications as a means to silently download malware and gain a foothold within corporate networks. This results in significantly higher risks for organizations that conduct business on the Internet, and whose employees use computers to access the Internet.

Cybercriminal organizations develop *zero-day exploits* — malicious pieces of software that take advantage of unknown, zero-day vulnerabilities for which patches don't exist. These zero-day vulnerabilities are a favorite target for attackers because many machines are affected and no protection is available. Organizations are concerned about this because zero-day exploits are quite harmful and often utilized for

infecting user machines with advanced, information-stealing, remotely controlled malware. In this chapter, we discuss the threat environment; zero-day exploits are covered in detail in Chapter 2.

On the Internet, adversaries can't be seen. Everyone knows they're out there because the airwaves are full of news about new breaches and break-ins. There are adversaries and victims, but connecting the dots in between is often difficult. Adversaries are willing to take great risks because catching them, or even knowing who or where they are, is difficult.

Understanding Cybercriminals and Their Motivations

Why do cybercriminals and adversaries attack organizations? Their motivations fit into a number of categories:

- **Financial gain:** Safe to say that many organized cybercriminal organizations are in it for the money — directly or indirectly. According to the U.S. Treasury Department, worldwide cybercrime surpassed drug trafficking as the largest source of criminal revenue.
- **Industrial espionage:** Organizations spy on each other to steal industrial secrets, for their own advantage, or to blunt the capabilities of their competitors.
- **Political espionage:** Nations and nation-states continue to spy on each other, and they always will. Breaking into computers is just the latest technique available.
- **Military:** Like political espionage, competing military organizations want to know more about their military adversaries, and they have added cyberattacks as another means to gain needed intelligence or to sabotage military or industrial facilities.
- **Activism and hacktivism:** A lot of cyberattacks are aimed at disabling the online capabilities of organizations that the attackers disagree with on some social or ideological level.

Exploits are a popular method for infecting target machines with malware because they operate silently, without user assistance or awareness. If a target system has a vulnerability

that the exploit is designed to attack, the exploit will successfully install whatever malware the attacker has chosen.

The primary risk of exploits and malware to organizations is this: Malware is used to steal information and gain control over employee machines, leading to a data breach. The methods that malware uses to compromise a machine are varied, including:

- Stealing credentials
- Logging keystrokes
- Grabbing data from browser and application screens
- Exfiltrating documents, emails, and other information resources directly from the infected machine
- Providing remote access for an attacker who wishes to directly examine target systems and networks

A brief history of cybercrime

The very early Internet was an environment where research ideas were exchanged. In the beginning, there was little to steal of monetary worth. But even at that time individuals were creating malware, primarily in the form of computer viruses. The first viruses were code fragments that attached themselves to .exe files and floppy disc boot sectors. They did little harm other than to simply propagate from one computer to another via the only means available for exchanging data: floppy disks.

Innovations in computing and networking brought new capabilities for sharing information, and new types of information to share. And — you guessed it — virus writers were there. Macro viruses could be embedded in word processing documents and spreadsheet files, and viruses and Trojan horses could be sent via the new communication tool — electronic mail.

In the mid-1980s, computers were networked together within — and among — organizations. The first widely spread Internet worm, the Morris Worm, was written and set loose in 1988. It may have infected about 10 percent of all computers on the Internet. Although the Morris Worm was highly disruptive to the usability of the Internet, it didn't disrupt business overall, because few businesses used the Internet for business transactions.

In the 1990s, the popularity of the Internet spread well past research institutions and computing professionals to include a sizeable portion of nontechnical private citizens.

(continued)

(continued)

Personal computers increased in popularity, and Internet connectivity expanded exponentially. This fertile ground was irresistible to creators of viruses, worms, and Trojan horses who found it entertaining to disrupt and remotely control large numbers of computers around the world.

By the early 2000s, entirely new industries centered on the Internet and computing. The Internet was not just about transforming brick-and-mortar activities into their electronic counterparts, but advances in information technology created new waves of goods and services unavailable prior to the Internet. Governments and organizations have changed themselves to such an extent that they are utterly dependent on the Internet to conduct many or all of their core operations.

Improvements in malware have kept pace with the Internet's growing complexity. With every new feature, protocol, and service that's available on the Internet, malware is one step behind (and sometimes one step ahead!) with tools and techniques to exploit known and unknown weaknesses.

Advanced Malware Enables APTs and Targeted Attacks

Often, a hacker develops a precise attack objective — usually targeting a specific organization that holds valuable data or wealth. In this section, we explore the techniques adversaries use to penetrate an organization.

After an adversary chooses an organization as its target, the objective the adversary seeks lies in one or more computer systems in the organization. The target organization may have defenses in place that make a direct, frontal attack practically impossible. Instead, many adversaries choose to penetrate an organization by enlisting the unknowing help of its personnel. A typical attack campaign may proceed through these steps:

1. **Reconnaissance:** Here, the adversary organization gathers information about its target, typically through social engineering and publicly available information.
2. **Attack planning and tools development:** Armed with some specifics about the organization and its employees, the adversary begins to plan its attack and

tailors tools and techniques specific to the particular campaign. Often these tools include the development of messages and websites meant to resemble sites used by personnel in the target organization.

3. **Grappling hook:** The adversary launches its initial attack, often using a spear-phishing message targeted at personnel and containing a *weaponized attachment* or using a *watering hole attack* by weaponizing legitimate sites that people in the target organization frequently visit. The goal of the attacker is to get one or more employees of the target organization to open the weaponized file or visit the weaponized website in order to infect their computers with remote control malware. Successful infections will give the adversary the means to continue its campaign. Typically, the malware is a remote access Trojan (RAT) that gives the adversary remote control of the victim's computer or information-stealing malware that can steal the user's credentials, payment card information, emails, documents, and more.

The term *weaponized* simply means that the file or website contains malicious content known as an *exploit,* which is discussed in detail in Chapter 2.

4. **Internal reconnaissance:** Now that the adversary has control of one or more of the target organization's workstations, it can use those workstations to conduct internal reconnaissance. This may involve monitoring email messages, observing network traffic, or mapping the internal network in order to discover the location of servers containing the ultimate objective (typically stealing money, stealing information, or disrupting site operations). This additional knowledge may require the development of more tools. Some of the internal reconnaissance will include observing to see whether the initial break-in was noticed or not.

5. **Final compromise:** Armed with the necessary knowledge and tools, the adversary is ready to launch its primary compromise, which may range from significant data breach and IP theft to sabotaging a target system.

6. **Covering the tracks:** The attacker will take a series of operations designed to cover its tracks so that the attack (or at least its source) will remain unknown to the organization.

This entire campaign may range in length from several weeks to a year or longer. The larger the potential reward, the stealthier the attacker will need to be in order to avoid detection and yet successfully reach its objective.

Developing advanced malware

As organizations place more capabilities for conducting business on the Internet, hackers have kept pace by developing new ways to attack governments, corporations, and citizens by stealing valuable information.

Some of the malware innovations that have developed over time include the following:

- **Remote access Trojans (RATs):** These are a type of malware that gives an attacker the ability to remotely access and control the target system (without a user's knowledge) at any time. The purpose may be to observe the user's actions, access data on the user's system, or to use the system as a jumping-off point to find and compromise other systems in an organization.
- **Information stealing:** This is malware that is specifically designed to steal login credentials, payment card information, or other sensitive data from high-value applications such as corporate applications and online banking applications.
- **Botnets:** Here, attackers remotely control large numbers (sometimes into the hundreds of thousands and beyond) of compromised machines and use them to relay spam or attack target organizations in a distributed denial of service attack.

Exploits are a popular method for getting malware onto target systems. Exploits are the tools often used to get into a system, and malware is the code that provides remote access, enables data theft and exfiltration, or is used to deliver spam or attack other systems.

Using spear phishing and social engineering for delivering advanced malware

All criminals know that the best way to steal something from someone is to convince a target to trust them. In the context of the Internet, attackers use social engineering to convince users to trust the content they provide. For example:

- **Fake security and news alerts:** A message claiming to come from an email service provider, online banking or payment site, news website, or online merchant will try to convince the recipient that some security-related or breaking news matter requires his or her immediate attention and action — such as logging into a fake website. The fake website can be a phishing website designed to steal the user's credentials or a malicious exploit site.
- **Transaction notifications:** A message claiming to come from a bank or merchant tells the recipient about a fictitious transaction that has just taken place. The message may contain a weaponized attachment or a link to a malicious phishing site/exploit site indicating that it will provide the user more information about the transaction or the ability to cancel it.
- **Government notices:** A message claiming to be from a government agency tells recipients about some urgent matter requiring their attention. This could be a bill from a tax collector, law enforcement, or almost any other agency.

Bottom line: The attacker convinces the target that the attacker is actually a trusted party. This targeting of personnel in order to obtain access to information is known as *social engineering*. The specific techniques used include:

- **Phishing and spear-phishing attacks:** Here, attackers create realistic-looking email messages or websites, hoping to trick recipients into performing something that will install malware on the target's machine, such as:
    - **Open a weaponized attachment.** Typically, such a document or program contains malicious code that downloads malware to the user's machine, enabling the adversary to steal information,

capture keystrokes using malware called a *key logger,* or give the adversary remote control of the computer.

- **Visit an exploit site.** Here, the website contains exploits that may infect vulnerable systems with malware.
- **Visit a phishing site.** The website is a visually convincing copy of an actual website, such as an online banking website. When the victim logs in to the phishing site, the victim types in login credentials, believing he's visiting the genuine site. The attacker can then use those credentials to steal money or information from the victim.

Phishing and spear phishing are basically the same except that a spear-phishing attack is targeting specific people or a specific organization. Further, the malware payloads delivered may target known technologies in use in the organization.

- **Whaling:** This is a phishing attack that targets high-value people in an organization — usually executives. Like spear phishing, the contents of the phish will be specific to the targeted audience.
- **Watering hole:** These attacks also result from social engineering and are used for delivering malware. Watering holes are websites that serve a specific community that the attacker is interested in, for example, a website used by certain professionals or a website that provides services to specific individuals. By compromising these websites and turning them into exploit sites, the attacker is able to directly infect the targeted website visitors with malware.

Zero-day exploits

Cyberattackers with talented resources can create a custom grappling hook to penetrate an organization by creating content that exploits a vulnerability unknown to the public. This is known as a zero-day exploit, and is especially dangerous because no patch is available for mitigating these vulnerabilities. Using such vulnerabilities to download unknown, never-seen-before zero-day malware enables the attacker to bypass traditional security controls that simply can't detect it. Zero-day exploits are explored fully in Chapter 2.

Spear phishing is on the rise. Why? Because it works. Spear phishing is an effective method for adversaries to get their malware into a target organization, as a first step in an advanced persistent threat (APT) attack that works toward stealing targeted information or sabotaging a critical operation.

Massively Distributed APT Malware

A growing trend seen over the last few years is the use of massively distributed malware, originally designed for financial fraud, for targeting nonfinancial organizations in APT-style attacks. An APT-style attack targets organizations in search of valuable information or intellectual property — or to sabotage the organization's systems. There are a few dozen malware families that were initially created to steal money from financial targets, like banks. These include the infamous Zeus, SpyEye, Shylock, and many others. Over time, malware developers extended the capabilities of these malware families, and added advanced evasion techniques, turning them into sophisticated APT tools that are used to target organizations in general. Massive distribution campaigns are used to infect as many user PCs as possible. After the PC is infected, the malware communicates with its C&C to get information about new targets.

The use of massively distributed malware means that attackers don't need to spear phish targets, or design custom malware. Instead, they use mass distribution techniques to infect as many PCs as possible. These malware distribution campaigns can use malicious email attachments, drive-by downloads, watering hole attacks, and social engineering schemes to infect millions of PCs around the world.

Studying the Three Lost Battles

Organizations have developed and adopted many tools, techniques, and processes in order to resist malware and cyberattacks. Some of these techniques have been more or less successful, but these successes have only served to motivate attackers to develop even better attack techniques to bypass our improving defenses (the "persistent" in advanced persistent threats).

If you think of the current era of cyberwarfare as a world war being fought on many fronts, three of these fronts, once thought reliable, have not proven effective at stopping advanced threats.

User education

Organizations call it *security awareness training* — a means for training internal personnel on the rules of safe computing in order to avoid and resist attacks such as phishing and spear phishing.

Unfortunately, you're bound to have a few people who, through haste, ignorance, forgetfulness, poor judgment, or just plain curiosity, will open a phishing or spear-phishing message that results in a successful compromise of the user's workstation. These days, even Internet-savvy employees can be fooled by the highly sophisticated phishing attacks that have been developed. And in turn this can lead to a successful cybercriminal campaign that may not be detected for a long time, if ever.

Of course, user education is useless against watering hole attacks that leverage websites that users access on a regular basis.

Avoiding vulnerabilities

Most phishing, spear phishing, drive-by downloads, and watering hole attacks depend on victim computers lacking essential security patches. Without these patches, user workstations may be vulnerable to exploits that are able to take complete control of a user's system without his knowledge.

Effective patch management consists of the timely deployment of security patches. But often there are so many security patches issued by software vendors in a given month that even well-funded security teams have trouble keeping up. There are also many vulnerabilities for which a patch isn't available — either because the vendor simply hasn't developed one, or because the vendor is unaware of the vulnerability. There is no way to defend against a zero-day vulnerability.

Malware detection

Many organizations have relied on traditional malware detection solutions such as antivirus to detect and remove malware from user endpoints. However, the advances in malware and evasion techniques have made traditional security controls far less effective.

A recent example that demonstrated the incompetency of traditional antivirus was shown in the 2013 *New York Times* breach, where the antivirus solution (provided by one of the leading vendors) detected only 1 out of 45 malicious files on employees' machines.

A technique called *polymorphism* allows malware to use a different, unique signature on each targeted system. This alone can make signature-based antivirus programs defenseless against these attacks. Today's malware uses advanced techniques to avoid detection and also to sabotage antivirus and patch installation programs.

Key Challenges in Advanced Endpoint Protection Solutions

Facing the current threat landscape, organizations struggle to implement effective controls to address the multiple attack vectors used to target users and gain access to sensitive resources. Products that address emerging threats are mostly offered as stand-alone solutions focused on a single threat vector. As a result, organizations are left with the inconvenient choice between the need to implement multiple endpoint solutions, an operationally untenable option, or remain with insufficient control gaps.

In addition, most organizations are lacking skilled resources needed for implementing and managing complex security controls. According to an IBM Trusteer-commissioned Ponemon study on advanced persistent threats, targeted attacks are the greatest threat organizations are facing, costing them on average $9.4 million in brand equity alone. Only 31 percent of respondents in that study say adequate resources are available to prevent, detect, and contain these threats.

In the IT and security profession, folks are constantly required to balance security against many other needs. Aside from scarce resources, overly tight security controls introduce other problems.

Managing IT Security Overhead

If a solution is difficult to deploy and requires continuous updates and administration, the total cost of ownership increases.

IT will have to invest a lot of professional resources in order to make the solution run properly, instead of enabling the growth of the business. If the security overhead is too high, there is a possibility that some of the security will be removed, or the solution will be deployed in a different way that requires less security.

Chapter 2

Understanding Zero-Day and Other Exploits

In This Chapter

- Understanding the nature of zero-day threats
- Exploring attacks against Java and other user applications
- Learning about weaponized content and watering hole attacks
- Taking a look at the vulnerability window

Data security professionals are adept at borrowing terms from common speech and assigning new meanings to them. Two such terms discussed in this chapter are *zero-day* and *exploit.* This chapter explains these concepts in detail.

Exploits are a common way to silently infect user endpoints with malware (for more on this, see Chapter 1). The malware will be used to gather information and enable an advanced persistent threat (APT) attack. This poses a significant threat to organizations, because a type of exploit called a zero-day exploit can be extremely difficult to detect.

Unfolding the Zero-Day Threat

A *zero-day vulnerability* is an unknown vulnerability in an application or a computer operating system. Software programs are full of vulnerabilities that are waiting to be discovered. The more complex a system is, the likelier it is that there will be more vulnerabilities, and that more of them will be serious.

A *zero-day exploit* is a never-before-seen code that exploits a zero-day vulnerability. A *zero-day threat* is a new threat that exploits a zero-day vulnerability and/or a zero-day exploit. The timelines of zero-day threats have changed dramatically over the years.

Technically, the *time to exploitation* is the time between the discovery of a vulnerability and the realization of threats that could exploit it. Often, due to extensive research, the black hats know about vulnerabilities before the software vendor (sometimes the vendor knows the vulnerability exists, but doesn't develop a patch right away). This provides exploit developers ample time to develop the *exploit* — content designed to exploit the vulnerability and alter the designed behavior of the application. By the time the application vendor releases a patch and makes it available, it is likely that cyberattackers are already exploiting the vulnerability.

Zero days from when?

The term *zero-day exploit* generally refers to a new threat that has never been seen before and has only now been discovered and investigated. But there's more to it than meets the eye.

The typical life cycle for a vulnerability begins well before public announcements of its existence. It is quite common for a software or hardware manufacturer to wait several months or even years after being notified of a vulnerability before publishing a patch for it.

The agreed-on protocol is to notify a vendor and give it a reasonable period of time to fix the vulnerability, and permit the vendor to control the publicity. However, there are many who don't play by these rules: If hackers discover a vulnerability, they may either develop an exploit on their own or sell information about the vulnerability to the highest bidder.

Although malicious black hats actively engage in research in hopes of finding new vulnerabilities, white hats do the same in hopes they will discover vulnerabilities first and notify software vendors before such vulnerabilities are exploited with zero-day exploits. Something most people would think is a *zero day vulnerability* may sometimes be related to a vulnerability that has been known for weeks, months, or more.

The Holy Grail for malware developers and other troublemakers is writing exploit code that exploits a zero-day vulnerability that isn't publicly known. With a zero-day vulnerability, no known patch or fix is available, so attackers are able to cause maximum damage while organizations with the vulnerable software have little or no defense against it.

It's best to assume that exploits for any newly published security patch have already been developed and are being used.

Exploiting Java and Other Vulnerable Applications

Today's endpoints have a plethora of built-in and add-on software components. These software components often include vulnerabilities. Some are a result of coding mistakes, some result from negligence, and some might be there due to design flaws. When these vulnerabilities are found in popular end-user applications, in a format that allows hackers to exploit them, these vulnerabilities become dangerous.

A few common characteristics are shared among exploitable applications:

- **They have vulnerabilities:** These weaknesses can allow a hacker to write exploit code that alters the designed behavior of the application.
- **They receive external content:** The attacker needs a way to deliver the exploit code to the machine. The easiest way is to hide the code within external content that the user receives. Such content can be email attachments, HTML content on web pages, and more.
- **They're commonly used by end users:** It's easier for the attacker to develop and deliver an exploit via a common application that can be found on most user machines, than to design and deliver an exploit for a unique, custom application.

The following sections contain some examples of exploited applications.

Java

Java is a top target for hackers looking for a way to break into endpoint systems. The Java software platform, which is present on practically every device in the world, has many critical vulnerabilities and exploits. Oracle has been criticized as being slow to respond in a timely manner to these vulnerabilities.

Some of Java's troublesome issues that make it a favored target for adversaries include:

- **Enterprise stickiness:** Most organizations rely heavily on business applications that require Java. Changing to other business applications that don't require Java may be prohibitively expensive.
- **Older version lock-in:** Many software applications are bundled with older versions of Java that contain, in some cases, hundreds of exploitable vulnerabilities. Rather ironically, one of the examples is a well-known endpoint security tool that requires an old, vulnerable version of Java.
- **Multiple platforms:** The Java software platform has been implemented on dozens of operating systems. Code written for Java will run on every system with the Java Virtual Machine (JVM). This means that exploits may be able to affect a vast install base.
- **Open source:** Java is open source software. Its source code is available for black-hat and white-hat researchers alike. And although white-hat researchers will try to make sure that vulnerabilities they find are fixed, black-hat researchers will secretly create exploits for vulnerabilities they find.

Java vulnerabilities can allow native exploits. A *native exploit* results in running native shell code. This is accomplished by techniques such as buffer overflow, use-after-free, and more.

In addition, Java vulnerabilities can enable execution of malicious Java applications.

Malicious Java applications are difficult to detect because of their delivery and execution modes: Java applications are delivered in a Java archive file called a *JAR,* which is basically a zip format. In a rogue Java application, the JAR contains malware that is encrypted, which makes it difficult to identify. In addition, malicious Java applications execute within the JVM (Java virtual machine), which is part of the Java runtime environment (JRE), a term coined by Oracle to describe Java's execution environment. This might sound complex, but many of you use the JRE every day when you visit your favorite websites. When the JVM executes sensitive activities, like making changes to the registry, or writing files to the file system, it's unknown if the code executing is trusted or not, so it is difficult to apply effective security controls without disrupting legitimate operations.

When a malicious Java application is executed, it targets vulnerabilities in the JVM and JRE to gain elevated privileges. With elevated privileges on the machine, malicious activities seem legitimate at the OS level. This means that unlike native exploits, malicious Java applications completely bypass the native OS-level protections.

Protecting against malicious Java applications is difficult. It's a major security gap that current security controls struggle to address.

Browsers and other targeted applications

Browsers, media players, and document viewers like Adobe Acrobat and Microsoft Word account for the majority of human interaction between computers and the Internet. Like Java, browsers, browser plug-ins, media players, and document viewers are the subject of vulnerability research and numerous zero-day exploits. They're software programs with a long history of highly critical security defects that are often exploited, resulting in partial or complete compromise of the endpoint.

The best-known media players and document readers with a long history of exploits include Adobe Flash Player, Adobe Shockwave Player, and Adobe Reader.

Malware developers are more likely to attack popular programs found on user endpoints in order to increase their success rates and maximize their return on investment.

Using Weaponized Content and Drive-by Downloads

The goal of today's cyberattacker is to perform stealthy attacks. The longer the breach remains undetected, the deeper the attacker can penetrate the organization. Exploits delivered via weaponized content and watering hole attacks enable silent downloads of malware on user machines and a stealthy entry point. The following sections discuss popular stealth-attack vectors that you should fear.

Weaponized content

The term *weaponized content* refers to a document, attachment, or a link to a website that contains hidden exploit code. A weaponized document or attachment can be, for example, a Word or PDF document, an Excel spreadsheet, or an Adobe Flash object. These files may include hidden exploit code that executes when the content is opened by the viewer application (for instance, when a weaponized Word file is opened by a vulnerable Word application).

A website containing weaponized content is known as an *exploit site.* An exploit site is a website that contains hidden exploit code. This can be a malicious site, created by an attacker, or a legitimate site that has been compromised and injected with exploit code. When the user browses to the website, it exploits a browser or browser plug-in vulnerability to download malware to the user's endpoint.

Spear-phishing attacks

Weaponized content is often delivered via a spear-phishing email convincing the user to open an attachment or click on a URL for an exploit site. Spear phishing requires the attacker to design a convincing message that would be trusted and opened by the user. This isn't a simple task. But by investing in social engineering and personalized messages, attackers find ways to gain a user's trust. This reflects a significant scaling up of the potency and sophistication of malware attacks.

Drive-by downloads

A drive-by download is a malware infection caused by exploitation of a browser vulnerability. When a user uses a vulnerable browser to access a website containing an exploit, the exploit can start a process that takes advantage of the vulnerability and downloads malware onto the user machine.

Watering hole attacks

In a *watering hole attack,* attackers target legitimate websites that are frequently visited by personnel in the target organization. Attackers compromise one of those websites, turning it into an exploit site by arming it with exploit code designed to take advantage of browser or browser plug-in vulnerabilities and download malware onto a visiting machine.

The term *watering hole attack* comes from the technique used by predators that wait for their prey to visit a watering hole. Instead of chasing the prey or luring it into traps, the predator simply waits at a place where the prey will go.

Because the compromised site is a legitimate site, often one needed by the employees, it is impractical for the organization to block access to this site.

A watering hole attack is a viable alternative to a phishing or spear-phishing attack if attackers believe that the target organization's personnel will be more resistant to phishing attacks.

Spear phishing, explained in Chapter 1, is a popular way to get weaponized content to targeted users. Whether through weaponized attachments or exploit sites, specially crafted messages to the targeted organization's personnel will often give attackers at least a few compromised systems from which to expand their attacks.

Understanding the Vulnerability Window

Security and risk managers are concerned with the *vulnerability window,* which is the span of time through the phases from discovery to full protection:

- **Discovery:** A vulnerability can be discovered by a software vendor or by a white-hat security researcher who notifies the vendor. That is the best-case scenario because it allows the vendor to start working on a patch immediately. But often the discovery is made by malicious players who prefer to take advantage of the vulnerability to promote their own agendas. In that case, the vendor might discover the vulnerability well after the exploit has been active in the wild.
- **Exploit development:** As soon as a malicious player knows about the vulnerability, exploit code is developed to exploit the vulnerability, and permit control or compromise of a target system.
- **Zero-day exploit:** Development is complete, and the exploit is ready to be used in an attack. The exploit is planted in a weaponized attachment or an exploit site.
- **Zero-day attack phase:** The exploit is delivered via spear-phishing emails, weaponized attachments, or exploit sites, and actively compromises user machines.
- **Vulnerability becomes publicly known:** As a result of the live attack, the public becomes aware of the vulnerability, forcing the vendor to issue a patch. In some cases, the patch can be provided in a matter of days. In other cases, it can take months for the vendor to issue a patch.

- **Software patch availability:** The software vendor has completed development of the patch and makes it available to the public.
- **Software patch deployment:** Organizations and end users begin to deploy the software patch to affected systems. Note that it can be weeks, months, or even years before patches are deployed to all systems.
- **Full protection:** Only after the patch has been deployed is the affected system fully protected against the exploit.

The presence of traditional signature-based security controls only protects endpoints against *known* malware because they must be familiar with the malware in order to recognize it (a signature of the malware must be available). Further, with advances in polymorphic malware, signature-based protection is becoming less and less effective.

As long as you have unpatched vulnerabilities, known or unknown, the machine is vulnerable to exploits.

The RSA zero-day attack

RSA, the Security Division of EMC, long known for its SecureID security token, was successfully compromised in early 2011. The breach was enabled by a weaponized document: an Excel spreadsheet that contained a hidden exploit.

The attacker used a spear-phishing message that was sent to HR employees at RSA. The message, which contained a weaponized Excel spreadsheet, claimed to be from a business partner. One or more staff members at RSA opened the Excel file, which planted Poison Ivy, a remote access Trojan (RAT) on their computers. This initial compromise gave attackers the grappling hook they required to carry out additional reconnaissance on RSA to find the location of valuable information about the SecureID product.

As a result, some of the valuable information about the SecureID token product was stolen, which could have given attackers the ability to successfully break into an organization using the SecureID product.

The RSA break-in is sure to become a classic textbook zero-day attack.

Looking at Remnant Risk from Unpatched Vulnerabilities

Zero-day attacks have attracted a lot of attention from security professionals, researchers, and organizations, and rightly so, for it is difficult to defend against them. However, some of the successful computer intrusions come not from exploiting zero-day vulnerabilities but from successful exploitation of known vulnerabilities on unpatched systems.

Some of the most famous malware attacks, including Nimda, SQL Slammer, and Zeus, exploited known vulnerabilities that had patches readily available for one to six months or longer. Nearly 60 percent of vulnerabilities exploited by hackers today are two years old. These attacks are still successful because many organizations are failing to apply security patches even years after their availability.

Chapter 3

Endpoint Compromise, External Communication, and Data Exfiltration

In This Chapter

- Understanding information-stealing malware
- Examining malicious command and control communication

In Chapter 2, we discuss *exploits* — one of the main entry points for malware that enables a security breach. By using weaponized content and hidden exploits, attackers silently download malware onto user endpoints. In this chapter, we describe how malware is used for gaining access to sensitive data and/or full control over the compromised system, and enabling the attacker to successfully breach the organization.

Information-Stealing Malware and Credentials Theft

Nearly everyone has heard the cliché, *information is the new currency,* and we're here to tell you that it's the gospel truth. And, like any currency, many folks are looking to find and steal it.

Grappling the hook

To successfully compromise a system in order to steal information, the malware needs to be executed. In Chapter 2, we

discuss the ways in which an exploit code can be used to deliver the malware to a target system.

Once the malware is delivered to a target system, it starts executing; this is the starting point that will allow the attacker to locate and exfiltrate data. The rest of this chapter discusses how malware is able to do this.

This first piece of malware that executes successfully on an endpoint is often called the *grappling hook* or the *beachhead.* Next, the malware will establish a communication channel with its operator, to receive operational commands. This helps the attacker begin the attack on the organization that will result in data exfiltration.

Information-stealing malware

After malware has achieved persistency on the endpoint, it will begin to carry out its mission: establish a communication channel to enable the attacker to gain access to data or control the endpoint. Stealing information and relaying it back to the attacker is the ultimate goal. We discuss techniques used for establishing communication channels and stealing information in this section.

Malware developers have created many techniques for stealing data from an endpoint system. After the malware is active on the machine, the attacker can choose which malicious functions to operate. A few typical malicious functions that can be operated remotely include:

- **Key logging:** Malware can intercept keystrokes from the system's keyboard drivers.
- **Screen capturing:** Malware can grab images of the screen on the endpoint, at timed intervals or when specific events occur, such as when bank account or credit card numbers are being displayed. Malware can also capture video recordings of the screen.
- **Form grabbing:** Malware can acquire information on web forms inside the user's browser. The most desirable information includes, for example, login credentials, credit card numbers, and bank account numbers.

- **Network eavesdropping:** Malware can eavesdrop on the target system's network communications, grabbing what it wants when it sees it. But malware can also turn a targeted system into a network sniffer and capture network communications from other systems on the same local network.
- **File system:** Malware can search the target machine's file system, looking for patterns, key words, or whatever is of interest to the attacker.
- **Remote control:** Some malware can provide the attacker full remote control over the machine.

Dodging the sandbox

A favorite technique used by anti-malware vendors for detecting new malware is to set up virtual sandbox systems that resemble poorly protected end-user systems. The idea is this: Malware will successfully infect these systems, and researchers will carefully examine the malware to see how it works so they can build new defenses for it. But adversaries have a few tricks up their sleeves, and they use these tricks to evade detection. Some examples include:

- **Watching for real human interaction.** Malware can watch for keyboard and mouse activity to see whether it has infected a real user's machine or an automated synthetic environment. If the latter, there will be no mouse or keyboard activity, and the malware will simply not activate.
- **Verifying whether it's running on a virtual machine.** Some malware variants analyze the infected machine's registry keys and other system settings in order to understand if it's running on a virtual machine or a physical machine. If it identified that it is running on a virtual machine, it will stop its functioning in order to evade being classified as malware.
- **Sleeping for a while.** The malware can just go to sleep for hours or days, after which time the sandbox analysis is completed. Because no malicious activity was detected during the analysis time frame, the sandbox may believe there is no malware to be detected.

In order to enable the malware operator to turn on any of these functions, a communication channel must first be established. Without establishing a communication channel, the malware can't register with the attacker and notify that a new machine is available. Without the communication channel, the attacker can't send functional commands to operate the malware. In other words, without a communication channel, the malware is useless.

When malware has implanted itself in a target system, the techniques available for stealing data are almost limitless. If the information is there, it can be found and exfiltrated. When malware has obtained information, it next needs to send it back to the malware operator, the attacker.

Malware C&C Communication

Command and control communications (known as C&C) refers to the interchange that takes place between installed malware programs and their central management systems. There are several uses for C&C:

- **Remote access and control:** This permits the adversary to illicitly access the infected system, explore its programs and data, and control the system. Think of the helpdesk getting into your system to help you, only this is being driven by the bad guys (and gals).
- **Operational instructions:** Adversaries can direct infected machines to do all sorts of things, such as grab user credentials, relay spam, or participate in distributed denial of service (DDoS) attacks.
- **Data theft and exfiltration:** An installed malware program can periodically send stolen data (login credentials, sensitive documents, or whatever the adversaries are targeting) back to the adversary's central system. This is the whole point of most malware.
- **Malware software updates:** Yes, even malware writers sometimes want to improve their programs, so why not?

In order to communicate with the attacker, malware will open an external communication channel. The simplest way to do that is by opening a direct communication channel. Direct communication channels are the simplest way for a malware to communicate with its operator.

A more sophisticated way, specifically designed to evade detection and bypass security controls, enables the malware to hide its communication. A popular technique for hiding external communication is by compromising legitimate applications and using their communication channels. For example, the malware will launch a legitimate browser process, like Microsoft Internet Explorer (IE). While the process is starting, the malware freezes the process and injects malicious code into it, replacing the legitimate code. When the process is resumed, all that is left is a *shell* of a process that looks legitimate — in fact it looks just like any other IE browser process, but it is actually not a regular process. There isn't even an IE window interface shown on the machine. This is because this is now a malicious process used for data exfiltration.

To evade detection by network security solutions that look for communication with known C&Cs, malware can also communicate with C&Cs over legitimate websites like Google Docs and user forums. After all, organizations will not block user communications with legitimate websites.

The usage of such evasion techniques makes it very difficult to identify the malicious communication channel. Because a compromised process looks like a regular process that is allowed to communicate externally, personal firewalls and network security solutions are unable to identify it as malicious. And it allows the attacker to freely communicate with the malware, operate it, and use it for data exfiltration.

These evasion techniques are less likely to be detected and blocked because they resemble legitimate traffic.

Chapter 4

Breaking the Threat Life Cycle

In This Chapter

- Following the threat life cycle
- Understanding the strategic chokepoints
- Choosing a multilayered approach for endpoint protection
- Examining the impact on IT
- Reviewing some other benefits

If you opened this book and turned right to this page, then you need to just trust us when we tell you that today's malware is bad — so bad that all the traditional means for combating it have proven practically useless. Ignorance isn't bliss.

But all is not lost, and there is a solution to the otherwise dreadful state companies are in regarding the malware wars.

This chapter discusses a multilayered approach for endpoint protection and how IBM Security Trusteer Apex Advanced Malware Protection uses this approach to break the threat life cycle at strategic chokepoints.

Explaining the Threat Life Cycle

The threat life cycle (see Figure 4-1) can vary between attacks, but in general, it goes through the following common attack phases:

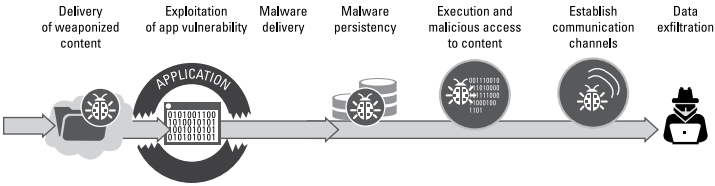

Figure 4-1: The threat life cycle: Infecting the user machine with advanced malware and exfiltrating data.

1. **Delivery of weaponized content:** An attacker creates *weaponized* content (a malicious attachment, web page, or something of that nature) that contains an *exploit* (a piece of content that takes advantage of a vulnerability in an application to change its behavior). The weaponized content can be delivered to the user by attaching it to a *spear-phishing email* (a specially crafted email designed to target the user), or sending a user a message with a link to an *exploit site* (a website that contains the exploit).
2. **Exploitation of an application vulnerability:** When the user opens the weaponized content with the targeted application, the embedded exploit alters the application's behavior. For example, it can cause a buffer overflow. The exploit causes the initiation of a chain of events that can result in malware delivery.
3. **Malware delivery:** The exploit chain can end up with malware delivered to the endpoint. Unlike a regular file download that is requested and approved by the user, this malware delivery is a silent process that takes place in the background, without the user's knowledge. When this process is cause by visiting an exploit website, it is known as a *drive-by download.*
4. **Malware persistency:** Once the malware reaches the endpoint, it strives to achieve *persistency.* This means that it will install itself to the system to ensure it stays there even after the machine is restarted.
5. **Execution and malicious access to content:** Now the malware can start executing. It can gather documents and files from the infected machine, capture user keystrokes, capture screenshots, and more. This information will be sent back to the command and control server after external communication is established.

6. **Establish communication channels:** Advanced malware has to establish an external communication channel in order to get operational commands from its operator. Such communication channels are also needed if the attacker wants to gain remote control over the machine or to exfiltrate data.
7. **Data exfiltration:** This is the final phase in which the attacker can access data and exfiltrate it, not just from the compromised machine, but from the network as well.

Understanding the Strategic Chokepoints

For years, companies have been trying to break the threat life cycle and prevent malware infections. But success has been limited. Cybercriminals continuously develop new evasion techniques to bypass detection methods. Even new detection methods are quickly bypassed. Detection methods used by popular antivirus solutions, for example, are no longer useful against these threats. This is because they try to identify threats by checking files against a blacklist of known malicious files. But the rate in which new malicious files are created today makes it impossible to keep up. In addition, the use of polymorphic engines makes it difficult for antivirus software to recognize the offending code because it constantly mutates.

Other solutions, like those that use network sandboxing, are trying to detect known malicious behaviors, and are bypassed by making changes to the threat's behavior, or by detecting the synthetic environments used for testing these threats.

In order to break the threat life cycle, you can no longer rely solely on threat detection. You have to consider that sophisticated evasion techniques are continuously improved by malware developers. In fact, there is a good chance that every detection method you use will eventually be bypassed. So, in addition to threat detection, you should look at new ways to break the attack life cycle.

Through extensive research, IBM has identified specific phases of the threat life cycle where cybercriminals have relatively few options to execute their malicious content (see Figure 4-2). IBM has termed these *strategic chokepoints*. By tightly controlling these chokepoints at the operating system level, IBM breaks the kill chain and prevents the attack.

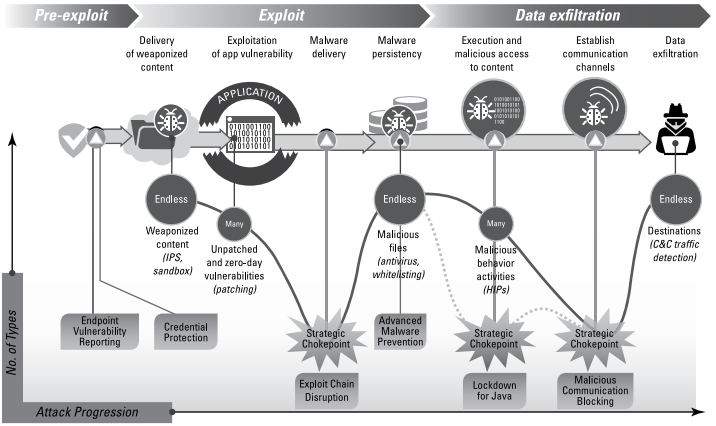

Figure 4-2: The strategic chokepoints in the attack life cycle.

The following sections go into detail on the three strategic chokepoints IBM Trusteer Apex addresses today.

Exploit Chain Disruption

Cybercriminals exploit end-user application vulnerabilities in order to silently download malware on the machine. This is a complex, multiphased process that ends up with the malware delivered to the endpoint. The phase in which the malware is delivered to the machine can be executed in a limited number of ways and is therefore, a strategic chokepoint.

To disrupt the exploit chain, IBM Trusteer Apex monitors the application state when the application is performing sensitive operations. The application state indicates not only what the application is doing, but also why it is doing it. For example, the application state can indicate that a file download is a result of a user request (Save As). Every application action creates a unique application state. This allows IBM Trusteer Apex to quickly validate the behavior of the application. However,

when an application state isn't indicative of a known, legitimate behavior, it means that an exploit is taking place.

To further explain how IBM Trusteer Apex disrupts the exploit chain and prevents malware infections resulting from exploits, the following examples illustrate the difference between legitimate actions that create valid application states and exploits that alter the behavior of the application, creating unknown and invalid application states:

- If a user uses a browser to access a website and download a file, that is a legitimate action that creates a known application state. However, if the website contains hidden exploit code that exploits an unpatched vulnerability in a browser to perform a drive-by download, an unknown, invalid application state is created.
- If a user works with Adobe Acrobat, and the application updates itself by writing files to the file system, a legitimate application state is created. But if a user opens a PDF document that he received via email, and the document contains hidden exploit code, the exploit code alters the behavior of the application to download malicious files to the file system. In this case, an unknown, invalid application state is created.

By monitoring the application state, IBM Trusteer Apex is able to identify when an application is exploited without needing any prior information about the exploit or weaponized content used to deliver it. By disrupting the exploit chain and preventing malware delivery, IBM Trusteer Apex effectively prevents malware infections that result from exploits.

Lockdown for Java

Java poses a great threat to enterprises and is so widely used that it has become a prime target for hackers. Vulnerabilities in Java applications can not only be exploited, but malware written in Java is especially difficult to detect and block.

When dealing with malware written in Java, or rogue Java applications, delivered as a JAR (Java archive file) that contains malware, different security measures are needed. Because rogue Java applications execute within the JVM (Java Virtual Machine), which is part of the Java Runtime

Environment (JRE), it isn't clear which JAR is actually executing. The first step for enabling effective security controls is to identify which JAR is executing. Then, you need to validate that its behavior is legitimate.

The second strategic chokepoint that IBM Trusteer Apex focuses on is based on the fact that excluding a small number of known, trusted applications, most Java applications don't execute sensitive actions, like writing to the file system or making changes to the registry. IBM Trusteer Apex prevents the execution of malicious Java applications by assessing application trust and activity risk, and blocking untrusted apps from doing high-risk activities. By controlling the behavior of Java applications, and preventing untrusted Java applications from executing sensitive actions, IBM Trusteer Apex can prevent malicious Java code from executing. As long as unknown Java applications are executing nonsensitive actions (for example, display to screen, or calculations), they are allowed to execute without disturbance. But as soon as a sensitive action is taken by an unknown, untrusted Java application, the application is blocked and quarantined.

Malicious communication prevention

Advanced malware must have an open communication channel with its operator. Without it, the malware can't receive operational commands or send out stolen data. IBM Trusteer research has shown that the ways in which the malware can establish external communication channels are few and limited. Therefore, this is the third strategic chokepoint. By preventing unknown applications from establishing external communication channels, IBM Trusteer Apex blocks malicious communication. IBM Trusteer Apex also prevents unknown applications from compromising known legitimate applications, and using their communication channels for hiding malicious communications. This means that even if a machine is infected with malware, the malware can't communicate with its operator, and thus is rendered useless.

Stopping zero-day exploits

Zero-day exploits occur during the time that elapses from the initial discovery of the vulnerability in an application until a patch or another mitigation is developed and published. This time period is known as the *vulnerability window.* Once a patch is available, a vulnerability is no longer a *zero-day exploit.*

In many cases, though not always, new zero-day vulnerabilities are discovered by malicious actors and not by the application vendor. As long as the vulnerability remains unknown, in zero-day status, malicious parties can develop exploits without disturbance. And they're likely to be successful in their attacks because the threat isn't known.

Stopping zero-day exploits is therefore a tricky task. It requires a method that doesn't rely on malware detection and doesn't require a daily update of malware signatures or malicious behaviors. It requires a method that doesn't rely on prior information about the threat, its source, or the malware it's trying to download. Instead, there is a need to identify invalid behaviors caused by exploits, and break the exploit chain to prevent the malware infection.

Here's how it works: IBM Trusteer Apex monitors the state of applications whenever applications perform sensitive operations like writing to the file system or opening a communications channel. When the application executes a sensitive action, IBM Trusteer Apex is triggered to validate the currently observed application state against all known application states. The application state indicates not only what the application is doing (its behavior) but also why it is doing it. For example, it can tell us if a file is downloaded to the file system because the user selected Save As or if it is part of an application update process.

As long as the application state matches a known legitimate application state, which means that the context of the operation is known, the application is allowed to proceed with the operation. However, if the application's state doesn't match any of the valid application states, as happens when an exploit takes place, then IBM Trusteer Apex disrupts the exploit chain and prevents the malware delivery to the endpoint. It also generates an alert to notify the user and the security administrator that an exploit attempt has been detected and the file it downloaded was blocked.

By disrupting the exploit chain, it is possible to prevent malware infections that result from exploiting virtually all known and unknown exploits, whether a patch is available or not. This is an especially effective defense against zero-day attacks and APTs, which rely on flying under the radar of malware detection solutions, like antivirus and intrusion detection systems.

The Essentials of a Multilayered Approach for Endpoint Security

Technologies that address emerging threats are mostly offered as stand-alone solutions focused on a single threat vector. As a result, organizations are left to choose between creating multiple stand-alone endpoint clients (an operationally untenable option) or continuing to grapple with insufficient controls that leave significant security gaps and expose them to the risk of a breach.

In addition, IT security organizations face operational challenges that result from the limited availability of highly skilled security professionals who are needed for implementing and maintaining complex security controls.

To address multiple attack vectors and emerging threats, IBM Trusteer Apex offers multilayered defenses integrated into a single software client. In addition to the strategic chokepoints discussed in the previous section, the defense layers that focus on strategic chokepoints are further integrated with additional defenses designed to address other important phases of the attack. We discuss those in the following sections.

Application Vulnerability Status Reports

Vulnerable unpatched end-user applications expose the enterprise to exploitation risk. The continuous need to apply application patches — in many cases urgent, critical patches — puts organizations in a never-ending rat race. And even that isn't enough to prevent exploitation of zero-day vulnerabilities for which a patch doesn't exist.

IBM Trusteer Apex includes a Vulnerability Status Report that lists installations of vulnerable applications like Java and Adobe Acrobat, describes known vulnerabilities associated with them, and provides further details about each vulnerability. The report enables security professionals to make informed decisions to either patch or remove vulnerable applications (if it is possible to patch or remove them).

Corporate Credential Protection

Many cyberattacks involve the use of stolen corporate credentials. According to the 2013 Verizon Data Breach Investigations Report, 76 percent of network intrusions exploited weak or stolen credentials. This fact isn't surprising. After all, if you have someone's login credentials, you've got the keys to the kingdom.

To address this threat, organizations require the use of complex passwords. Complex passwords may be harder to guess, but they can still be compromised through credentials phishing attacks. In addition, password reuse is a major problem, especially when corporate passwords are reused on noncorporate websites, like social networks and e-commerce sites. The lack of enforcement of password reuse policies is a big security gap that organizations are struggling to deal with.

To protect credentials from phishing attacks and prevent password reuse, IBM Trusteer Apex verifies that corporate credentials are used only on approved corporate sites. If a user is lured to a phishing site and attempts to provide the corporate credentials for logging in, a message will pop up and alert the user that this isn't a corporate website. The same will happen if the user tries to reuse the credentials on any other nonapproved website, social network, or e-commerce site. The user is granted a period of time to change the password before use is blocked. This easy-to-implement protection enables organizations to address an important security gap and ensure the proper use of corporate credentials.

Malware detection and mitigation

When enterprise endpoints are already infected with malware, the malware must be detected and removed. Two groups of malicious files need to be dealt with.

Massively Distributed Malware

Originally designed for financial fraud, malware families like Zeus, SpyEye, and Citadel have been massively distributed around the world. Millions of machines around the world are already infected with variants of these malware families. Over the years, these malware families have been enhanced with

advanced capabilities and evasion techniques turning them into sophisticated APT tools that can be used to target organizations in APT-style attacks.

Legacy threats

Many endpoints are still infected by known malicious files. This is because traditional security controls can no longer keep up to date with the rapidly growing numbers of known malicious files. The independent security firm AV-Test claims it registers over 220,000 new malicious programs every day. This means that maintaining an up-to-date blacklist of known malicious files would be darned near impossible. In addition, solutions that scan the file system searching for known malicious files, like antivirus solutions, consume many system resources and impact the systems' availability.

To detect and mitigate malware infections, Trusteer Apex provides:

- **Prevention and mitigation for massively distributed APT malware:** Apex detects, mitigates, and remediates massively distributed malware infections by identifying new and existing installations and removing the threat from the machine.
- **Cloud-based file inspection:** Apex uses consolidated information from over 20 AV engines to provide legacy protection from known malware. If a file is detected as known malware, Trusteer Apex Advanced Malware Protection effectively prevents the file from executing and compromising the machine. The feature offers operational simplicity, without requiring lengthy signature file update processes that impact network and user productivity.

The integration of controls at strategic chokepoints, and defense layers that address other attack stages, enables Apex to provide powerful protection against both unknown, zero-day threats and known malware, without impacting user productivity.

Boosting Protection with Real-Time Threat Intelligence and Security Services

With the threat landscape changing so rapidly, enterprises struggle to keep up with dynamic and rapidly evolving threats intruding on their networks. Enterprise security professionals and risk officers face many daunting challenges. Making informed, intelligent decisions is harder than ever.

To help organizations deal with emerging threats and security incidents, IBM Trusteer Apex deployments are backed by IBM's security services, which provide ongoing support to customers. IBM's security services dramatically improve the customer's ability to face advanced threats and targeted attacks.

IBM Security Trusteer's research labs and expert team of malware researchers work in cooperation with IBM Security labs and the IBM X-force team to continuously analyze the latest security threats and targeted attacks. Threat research and intelligence data is based on dynamic security feeds provided by over 100 million protected endpoints around the world. The combined vulnerability database is one of the largest in the industry with over 70,000 vulnerabilities categorized. Threat research and intelligence is translated into security updates that are automatically sent to protected endpoints.

IBM Security Trusteer researchers continuously analyze security event feeds from around the world. Automated updates are pushed to all protected endpoints as soon as they're available. Exploits and malware intrusions are blocked in real time. Because Trusteer provides the updates, IT security professional don't need to continuously update rules or policies, and no in-house expertise is needed to ensure the solution is up to date against the latest threats. This allows the organization to focus its resources on IT projects that support the core business.

Organizations can offload the analysis of potentially suspicious activity to the IBM Security Trusteer threat analysis service, which helps organizations assess suspicious activities

and provide protection recommendations. IBM security services analyze the organization's specific threats and help the organization take action on them.

Deployment and Management Considerations

IBM Trusteer Apex provides unique, multilayered defenses to help you deal with the multitude of threat vectors. Plus, it's easy to deploy, manage, and maintain. It leverages in-depth technical expertise and low level visibility into application execution paths, to apply accurate and effective controls on strategic chokepoints and prevent malicious code execution.

IBM Trusteer Apex provides automated threat analysis capability to prevent attacks. Because it's a single solution, you don't have to deal with many disparate point solutions — each needing its own infrastructure, management expertise, and maintenance. Because the product is easy to manage and maintain, it helps the Chief Security Officer and the IT security team be more resourceful and effective.

IBM Trusteer Apex is a hosted solution, which means there are no in-house servers or appliances to install and manage. The beauty of a hosted solution is that IBM manages the entire management infrastructure so you don't have to. With rising data center costs and IT departments being stretched razor thin, IBM's hosted solution in your data center is *zero footprint*.

IBM Trusteer Apex is part of the IBM Threat Protection System. This system is designed to address enterprise security challenges by discovering new problems and stopping them before they impact your business. The security approach is contextually aware and integrated with a network protection solution, security intelligence and analysis, and security information and event management systems. It uses the very latest techniques in next-generation prevention, comprehensive detection, and automated response capabilities.

Chapter 5

Top Ten Protection Considerations

In This Chapter

- Ten protection considerations for selecting zero-day threat protection solutions

If you're considering zero-day exploit disruption and advanced malware protection, the ten most important considerations for selecting a solution are found right here in this chapter.

Defense in Depth and Layered Security

You should never rely on a single defense mechanism. Even the latest, most sophisticated security controls may be circumvented. The more defense layers that are in place, the better chance you have against emerging threats and targeted attacks. A layered approach to security can be implemented at any level, including the endpoint. It should include multiple types of security measures, each protecting against a different vector for attack. Be sure to implement layered security in a way that is manageable and doesn't impair the users.

Leverage Global Attack Intelligence

Your vendor must provide threat intelligence. It's best to have this intelligence based on millions of client machines monitoring threats throughout the world. Intelligence from these systems

gives monitoring threats data about the latest threats. Research teams use this data to ensure that the solution effectively blocks all threats, without impacting user performance or productivity.

Accurate Real-Time Threat Protection

With attacks growing more sophisticated than ever, your organization needs to block attacks as they happen. In fact, real-time prevention remains essential to any successful strategy for stopping advanced threats from penetrating the organization. It can be the key for protecting critical business assets — including endpoints, servers, and applications — from malicious attacks.

Stop Known and Unknown Zero-Day Threats

An advanced threat solution should be able to stop attacks even if they utilize a zero-day threat. This means that effective protections would break the exploit chain even if the exploited vulnerability is an unknown, and no patches or other mitigations are available. Your solution should be able to identify malicious activity and block it, even if the source is a never-before-seen file or application. Daily updates of new malware characteristics should not be required to block new, unknown zero-day threats.

Preventing Endpoint Compromise and Data Exfiltration

After malware infects an endpoint, it's going to establish an external communication channel to enable communication with the attacker. Later, this communication channel will be

used for data exfiltration. Advanced malware protection solutions should prevent malware communications and data exfiltration, regardless of evasion techniques.

To prevent compromise, advanced threat protection solutions need to prevent malware from establishing external communication channels on infected machines.

Protect Corporate Credentials

Stolen credentials are an adversary's favorite way to break into a valuable environment. Finding a bag of keys is the next best thing to finding a bag of money.

Two primary considerations for protecting enterprise credentials include protecting against phishing attacks that target employees with fake websites and preventing corporate credential reuse on noncorporate sites to reduce the risk of password exposure.

Understand Endpoint Vulnerability Status

Vulnerable, unpatched end-user applications expose the enterprise to exploitation risk. The continuous need to apply application patches, in many cases urgent critical patches, puts organizations in a never-ending rat race. And even continued effort to install patches isn't enough to prevent exploitation of zero-day vulnerabilities for which a patch doesn't exist.

Visibility into the enterprise risk posture resulting from vulnerable applications enables security professional to make informed decisions to either patch or remove vulnerable applications (if it is possible to patch or remove them).

Even if you have the latest version installed, and all patches applied, the application can contain vulnerabilities. These unknown vulnerabilities are often exploited by zero-day threats!

Simple and Easy Deployment Options

If your organization has enterprise endpoint management systems such as IBM Endpoint Manager or WSUS, great! Then you'll want to be able to push silent, no-reboot installs of your chosen advanced threat protection software to all your endpoint systems.

In addition, you can use detection snippets to ensure that all endpoints are protected before they access critical enterprise resources. These easily installed code snippets can be placed on a VPN login or any login of a web-based enterprise application, including Software as a Service applications. The detection snippet will quickly ensure that all endpoints are properly protected.

Minimize IT Costs and the Need for Professional Resources

IT departments are overworked. The last thing an IT department wants to hear is that yet another security tool needs constant care and feeding. That's a fail. Further, most organizations don't have in-house expertise to research the latest threats and ensure that the solution is up to date.

Instead, today's IT organizations (and businesses) need solutions that require minimal maintenance and have the vendor manage automatic updates, alerts, infrastructure, and everything else that makes it work.

Scalable Protection for All Enterprise Endpoints

An effective advanced threat protection system will be successful only if it works in any size of organization, whether there are 100 employees or several million. Your environment will constantly grow and change; whether you have office-based employees, remote offices, traveling employees, or all of the above — every employee endpoint needs to be protected.

Stop zero-day exploits and targeted attacks with a multilayered approach!

Zero-day and other exploits are used to infect your systems with advanced malware and steal valuable data. This book explains how they work and why traditional defenses such as antivirus, patching, and security awareness training are ineffective to stop them. Find out why a multilayered approach is needed to break the threat life cycle, prevent zero-day exploits, and protect your most valuable information.

- ***Explore zero-day exploits* — *how they're used to break into an organization***
- ***Understand why antivirus and patching fail to protect you* — *zero-day exploits are always one step ahead***
- ***Discover data exfiltration* — *how intruders steal and remove your data***
- ***Look at a multilayered approach* — *break the threat life cycle in multiple places and stop zero-day exploits***

Open the book and find:

- **How zero-day exploits enable silent downloads of malware on user machines**
- **Why antivirus and patching can't stop zero-day exploits**
- **The techniques intruders are using to get information out of your endpoints**
- **How to protect corporate login credentials**
- **How a multilayered approach is used to break the threat life cycle and stop zero-day threats**

Go to Dummies.com®
for videos, step-by-step examples, how-to articles, or to shop!

Peter H. Gregory is the security and risk manager for a global retail organization, an adjunct university instructor, and the author of over thirty books on security and emerging technologies.

FOR DUMMIES®
A Wiley Brand

ISBN 978-1-119-03864-1
Not for resale

WILEY END USER LICENSE AGREEMENT

Go to www.wiley.com/go/eula to access Wiley’s ebook EULA.