Making Everything Easier!™

Trusteer Special Edition

Stopping Zero-Day Exploits

FOR DUMMIES®

A Wiley Brand

Learn:

- About the dangers of zero-day threats
- How Stateful Application Control protects against zero-day threats

Compliments of

Trusteer
an IBM Company

Peter H. Gregory, CISA, CISSP, CRISC

Stopping Zero-Day Exploits

FOR DUMMIES®
A Wiley Brand

Trusteer Special Edition

by Peter H. Gregory, CISA, CISSP, CRISC

With contributions from:

Dana Tamir
Director of Enterprise Security
Trusteer, an IBM Company

FOR DUMMIES®
A Wiley Brand

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

ISBN 978-1-118-75850-2 (pbk); ISBN 978-1-118-75990-5 (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
Custom Publishing Project Specialist: Michael Sullivan
Project Coordinator: Melissa Cossell

Contents

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 technology called Stateful Application Control, which provides 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 of us 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 explores a new technology 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 Trusteer, an IBM company, 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 abilities 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
- Scraping 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 code 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 malware that give 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 attacker 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

Every criminal knows 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 their immediate attention and action — such as logging into a fake website. The fake website can be a phishing website designed to steal their 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 code 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.

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. And the bring your own computer (BYOC) trend is making it impossible to ensure patch deployment on the growing number of unmanaged computers plugging into the network. 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 also uses advanced techniques to avoid detection and also to sabotage antivirus and patch installation programs.

Juggling Security Trade-Offs

Security is important — but so is the business. Security should enable the growth of the business and user productivity, not hinder it.

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, which are discussed in this section.

Balancing usability and security

Usability refers to the relative ease at which a system can be used. Plainly stated, a system is no longer considered usable if the system's users are unwilling or unable to expend the effort to utilize the system's security controls, or if the system becomes so overburdened that it becomes effectively unusable. Examples of usability issues include:

- **Excessive machine resources spent on security:** There is nothing as annoying as an antivirus performing a whole-disk scan consuming all system resources while you're trying to do other work. (And when antivirus is so much less effective than it used to be, what's the point anyway?)
- **Connectivity restrictions:** Good ways to prevent data leakage may include preventing the use of USB attached external storage, email, printing, or saving files to a local filesystem. Certainly these controls may help, but they will probably also hinder an employee's ability to get any work done.
- **Never-ending pop-up messages asking the user if actions should be allowed or blocked:** Most users don't know and don't care. They just want to do their jobs. To most users, these pop-up messages are just annoying, and they click through them regardless of the circumstances.

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 code* — code 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. What is *zero day vulnerability* to us 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 one or more 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 platform:** 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.

It should be no surprise that Java is a new favorite target of security researchers of both the black and white hat sort.

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.

Dealing with BYOC: The elephant in the room

As if all of the other developments in this section didn't provide enough challenges for IT and security managers, the seemingly unstoppable BYOC (bring your own computer) trend, where end users are allowed to use personally owned computers for conducting business, instead of organization-provided and managed computers, adds considerable complications to stopping all exploits. Because personally owned computers aren't managed by IT departments, there is no control over the software installed on such computers, the patch levels, or the security controls protecting these computers. In addition, these machines travel in and out of the secure corporate networks. If not properly protected, they can get infected while outside of the network, and introduce the infection after they reconnect.

Managing security on endpoint systems, even in mature organizations, is especially difficult, in part because the attack surface is different on every individual endpoint. Users often have the ability to change at least some of their endpoints' security settings, as well as install software programs and browser plug-ins. But with BYOC, things can quickly get out of control, which is why these computers require special considerations.

Using Weaponized Content and Watering Hole Attacks

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 stealthy 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 a 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.

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 visiting user's 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 done 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 spearphishing 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 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.

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.

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 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. The exploit code takes advantage used to deliver the malware of vulnerability in an application to silently download malware on the user machine.

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*. This helps the attacker begin the attack on the organization that will result in data exfiltration.

Information-stealing malware

Once malware has established itself on an endpoint, it will begin to carry out its mission: stealing information and relaying that stolen information back to the malware's owner. We discuss techniques used in this section.

Malware developers have created many techniques for stealing data from an endpoint system. A few of these techniques 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.

Several iterations of compromising systems and exfiltrating data may be required before the attacker's targeted information has been obtained. For instance, the first compromised machine may have provided login credentials, which enables the attacker to gain access to other systems from which more information can be obtained to help the attacker determine the actual location of the data the attacker wishes to steal.

A single compromised machine can provide the attacker with access to other systems, to corporate networks, and eventually to corporate resources that contain valuable data.

Dodging the honeypot

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.

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.

Exfiltrating stolen information

When malware has obtained information, it next needs to send it back to the malware operator, the attacker.

The methods used by attackers to exfiltrate information depend on both what is available to the attacker and which method is least likely to be detected. The choices include HTTP, HTTPS, FTP, email, and Internet relay chat (IRC). If the attacker believes the target organization has a data leakage prevention (DLP) system, for example, then the attacker may choose to encrypt the information so that the exfiltration will not be detected.

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 computers. 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).
- **Data exfiltration:** An installed malware program can periodically send stolen data (login credentials, sensitive data, or whatever the adversaries are targeting) back to the adversary's central computer. This is the whole point of most malware.
- **Malware software updates:** Yes, even malware writers sometimes want to improve their programs, so why not?
- **Dirty work:** Adversaries can direct infected machines to do all sorts of things, such as host phishing sites, relay spam, or participate in distributed denial of service (DDoS) attacks.

In order to communicate with the attacker, malware will attempt to open an external communication channel. The simplest way to do that is by opening a direct communication channel. But direct communication channels are highly visible. Host-based security controls, such as personal firewalls, can easily identify that these are new, potentially malicious communication channels and block them.

To evade detection and bypass such security controls, the malware will attempt to hide its communication. A popular technique for hiding external communication is by compromising legitimate 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* 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 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.

The usage of such evasion techniques makes it very difficult to identify the malicious communication channel. Because it 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

Discovering Stateful Application Control

In This Chapter

- Getting a handle on Stateful Application Control
- Understanding how Stateful Application Control stops zero-day exploits and prevents malicious data exfiltration
- Choosing deployment options for endpoint protection
- Examining the impact on end users
- 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 of the traditional means for combating it have proven practically useless. Ignorance is not bliss.

But all is not lost, and there is a solution to the otherwise dreadful state companies are in regarding the malware wars. Keep reading to learn about Stateful Application Control and why it is one of the most promising developments in a long time.

This chapter discusses Stateful Application Control in general and also discusses Trusteer Apex, which offers the technology.

Introducing Stateful Application Control

Stateful Application Control is a new approach to preventing the execution of malware delivered by exploits and curbing advanced malware from compromising user endpoints. Stateful Application Control analyzes the application state to determine what an application is doing and why it is doing it. Using this approach, it's possible to accurately determine whether the application's action is legitimate — and block unauthorized, malicious files that were downloaded via illegitimate actions (exploits).

Through years of research, Trusteer has found that legitimate application actions create known application states. By analyzing the application state, it is possible to understand the context of an action, to figure out *why* it is taking place. For example, by understanding the application state, it is possible to determine that the application is writing a file to the file system because the user requested to Save File.

Stateful Application Control quickly and automatically identifies invalid application states that don't match known legitimate application actions. This enables accurate detection of exploits and protection against malware delivered via exploitation of known and unknown application vulnerabilities.

Application states don't change very often — even when an application is patched or upgraded from one version to another. A solution that is based on monitoring and validating the application state is inherently stable and accurate, requiring fewer updates and resulting in fewer false-positives. On the rare occasion when an application has a new, legitimate state, Trusteer Apex immediately notices the new state and applies an automated process to update the solution with the new application state.

For a visualization of how Stateful Application Control works, see Figure 4-1.

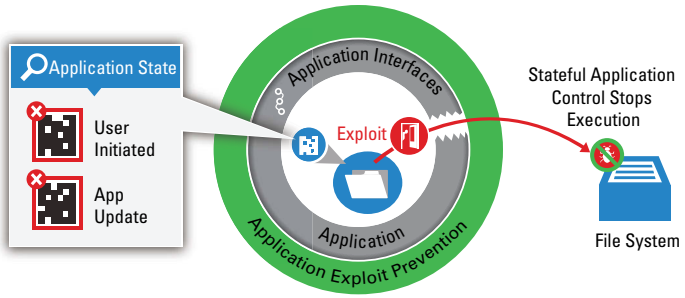

Figure 4-1: How Stateful Application Control stops execution of malware delivered by exploits.

Stopping Zero-Day Exploits and Targeted Attacks

Stateful Application Control is an effective means for blocking zero-day exploits and other advanced persistent threats (APTs) that are able to fly under the radar. This is because Stateful Application Control doesn't rely on malware detection and doesn't require a daily update of malware signatures or malicious behaviors. It doesn't need any prior information about the threat, its source, or the malware it's trying to download. Because it doesn't attempt to detect the threat, but rather validate the application state, Stateful Application Control is able to accurately block threats, whether known or unknown.

Here's how it works: Trusteer Apex monitors the state of application whenever it is performing sensitive operations like writing to the file system or opening a communications channel. When the application uses an application interface, Stateful Application Control is triggered to validate the currently observed application state against all known application states. 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 Trusteer Apex prevents the downloaded file from executing and compromising the machine.

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.

Stateful Application Control protects organizations from malware by preventing the execution of malware delivered through exploitation of vulnerabilities in endpoint applications.

This protects organizations from 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. Unlike malware detection solutions, Trusteer Apex doesn't require advanced information about the threat in order to identify and block it.

To further explain how Stateful Application Control prevents compromise by malware delivered via 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.

New legitimate application states are introduced infrequently, but when they do change, Trusteer provides the new application state so users and security administrators don't need to create the updates themselves, and don't need to develop special expertise around the creation of application states. Automated updates are directly sent to all protected endpoints.

Application whitelisting's failed promise

Because blacklisting of known malicious signatures (antivirus solutions) has proven ineffective against advanced threats, a different and completely opposite approach emerged. Instead of blacklisting all known-bad and continuously changing malware (you already know why this is no longer viable), you create a whitelist of all known good software. The thinking went that with millions of new varieties of malware emerging every week, whitelisting trusted files would be far easier to keep up with. It was believed that the list of valid applications used in an enterprise environment would be far smaller and more stable than the list of malicious files, but this assumption proved to be incorrect.

The problem is, there are so many good pieces of software that the list is in the billions and climbing daily. Every software program used by employees must be included in the list — every home-grown application, every patch, and every update.

An organization that wants to implement a whitelisting solution must first verify that all software files used by employees are included in the whitelist. The organization needs to create processes to define how and when new software and software patches are added to the whitelist. This continuous effort becomes a huge burden on IT. In addition, users who need new software installations and updates can become frustrated with a solution that restricts these updates. The number of helpdesk tickets grows, and the load on the helpdesk significantly increases.

And don't you just wonder how long it would take malware writers to get their programs whitelisted, defeating the whole thing? This has already happened.

Preventing Data Exfiltration

Stateful Application Control provides a *defense in depth* approach to blocking malware. What this means is that Stateful Application Control blocks the initial execution of malware if its intrusion was through exploitation of application vulnerabilities (the *grappling hook*). In addition, Stateful Application Control blocks malicious command and control (C&C) communication and the actual theft of data (exfiltration). Any attempt by malware to transmit stolen data away from the victim machine is blocked. You might think of this

like a security guard in a bank who prevents bank robbers from entering the bank, and also prevents bank robbers from leaving the bank.

In case this is sounding a little too abstract, let me explain with an example:

> Malware has successfully infiltrated a user's computer. Now it needs to get further instructions from its operator. To do that, the malware needs to establish an external communication channel. Because malware should not be allowed to communicate externally, this communication channel must be blocked.

It is easiest for the malware to open a direct communication channel from the infected machine to the Internet, and use it for communicating with the attacker, typically via a C&C server. However, because direct communication channels are highly visible, it is quite easy to block them, and most personal firewalls will be able to identify these channels and block them.

In order to evade detection and enable malware communication, malware creators have developed sophisticated evasion techniques to bypass these controls. Advanced malware will use more sophisticated techniques to communicate with its operator. Knowing that some security systems will prevent unknown programs from opening direct communications, malware will attempt to use a legitimate program to communicate. As we explain in Chapter 3, advanced malware can, for example, compromise other legitimate processes to hide its malicious communication:

> The malware will launch a process that is allowed to communicate externally, for example, Internet Explorer (IE). When the process launches, the malware will freeze the process, inject code into the process, replacing existing code with malicious code, and resume the process. There is now a process that looks to the operating system like an IE process, but it is actually only a shell of that process running malicious code. It looks like any other IE process (although if you look carefully, there is no relevant IE window visible on the screen). Because there is no indication that this process is a compromised process, security controls, like personal firewalls, allow it to communicate externally.

Why does the malware use a browser for communications instead of communicating directly? There's one very good reason: It removes the risk of having security controls identify outbound communications from an unknown program and block it. But if the process looks legitimate, who is going to block it?

Another evasion technique often used together with legitimate process compromise is the use of legitimate websites for C&C communication. Because security controls can't identify compromised processes, they try to determine whether the communication channel is malicious based on its destination. If it communicates with a known C&C site, it should be blocked. To evade detection, malware can communicate with C&Cs over legitimate sites like user forums and Google Docs. Because these are legitimate sites, and it is impossible to distinguish between legitimate and malicious communication to these sites, this traffic isn't blocked.

Stateful Application Control blocks the malware's attempts to establish direct communication channels. It also identifies that the malware is attempting to compromise legitimate processes and blocks it. This prevents the malware from being able to hide its communication channels and freely communicate with the attacker to exfiltrate data. Because Stateful Application Control has deep visibility into malicious code operations on the machine itself, it is able to accurately block the malware at an early stage, preventing it from using the previously mentioned evasion techniques.

Trusteer Apex blocks attempts by malware to open direct external C&C communications. It also prevents malware from compromising other legitimate processes for hiding external communication. When C&C communication is blocked, information-stealing malware is rendered useless.

Protecting Corporate Credentials

Adversaries are after corporate credentials because they provide the keys to the kingdom. With stolen credentials, adversaries can simply log in to corporate systems; this will not be seen as unauthorized entry but legitimate access.

There are a few ways that attackers gain corporate passwords:

- **Using key loggers to steal credentials off the user's machine.** Many malware variants include key-logging functions that enable the attacker to grab users' credentials.
- **Using phishing sites.** These are fake sites that look like legitimate sites (like online banking sites or Google Apps login pages), convincing the users to enter their credentials. For more on this, see Chapter 1.
- **Hacking into public consumer websites and/or social networks and stealing the user database.** Cyberattackers know that people don't want to remember a lot of different passwords. Instead, many people remember one or two good passwords and use them on as many different sites as possible. Users will tend to use their one, good, complex password on all of their personal and business sites, not knowing that there is an inherent risk in doing this. If a password table from one website is stolen and successfully decrypted, its hashes cracked, or its login credentials compromised, adversaries will try those stolen credentials on other sites, and often meet with success.

Trusteer Apex includes security layers specifically designed to protect corporate login credentials against theft and exposure:

- One part of the solution prevents key loggers from capturing the user's credentials. It does so by obfuscating users' keystrokes. Any malware that is intercepting keystrokes is going to be reading obfuscated keystrokes, which will do the attacker no good.
- The second part of the solution prevents users from exposing their corporate credentials on phishing sites. Trusteer Apex allows users to enter their corporate credentials only on preapproved valid corporate websites. If Trusteer Apex recognizes that the user isn't on an approved corporate website, it prevents the submission of the corporate credentials. So if the user thinks he is accessing a corporate site but it's actually a phishing site designed to look like the corporate site, he won't be allowed to log in.
- The third part of the solution prevents users from reusing their corporate credentials on consumer sites and social networks: Because these sites aren't on the list of

approved corporate sites, users can't use their corporate credentials to log in. They will have to use different credentials for such sites, preventing the risk of corporate credentials exposure via a third-party website hack.

Trusteer Apex notifies users when they're not allowed to use corporate credentials and sends an alert to IT security about these events. In case a new corporate website needs to be approved for user login, security administrators can easily add the site to the approved corporate website list with a click of a button.

Looking at the Deployment Options

A powerful tool like Stateful Application Control would be useless if there was no easy way to get the software onto endpoints. Trusteer thought of this and has designed the solution to make it as easy as possible to get it installed onto all endpoints — managed and unmanaged. The options work like this:

- **Deployment for Managed Endpoints.** Trusteer Apex software can be pushed to all managed endpoints using common enterprise software distribution tools. Many different enterprise deployment tools can be used, including Microsoft SMS or IBM Endpoint Manager.
- **Deployment for Unmanaged Devices.** Organizations can place a detection snippet on corporate websites and SSL VPN login pages to ensure that the machine from which the user is trying to log in is protected by Trusteer Apex, before the access is granted. If Trusteer Apex isn't present, the detection snippet will pop up a message that requires users to download and install the software. The user will have to download the software agent in order to proceed, a process that takes only a couple of minutes and doesn't require machine reboot. After Trusteer Apex is installed and running, users will be allowed to access the network or company site they need.

The software agent technology can be installed on a wide variety of platforms and is compatible with other software applications that are already running on the machine. This has

been proven on tens of millions of protected machines around the world already running the software agent.

Ease of installation for unmanaged devices can be a boon for organizations struggling with BYOC (bring your own computer). Trusteer Apex protects them all.

Managing Zero-Day Risk and Endpoint Protection

Trusteer Apex includes centralized management for all protected endpoints, including managed and unmanaged endpoints (think BYOC here). The Trusteer Apex dashboard is depicted in Figure 4-2. This allows an organization to

- **View all agent statuses from a centralized console (for both managed and unmanaged endpoints).** This helps a security manager to quickly spot agents having trouble of any kind.
- **Analyze security events.** The events dashboard can give widespread visibility into blocked malware delivered by exploit events or exfiltration events of malware attempting to communicate externally.
- **Manage security policies.** Centralized management of configuration security policies across all enterprise endpoints.

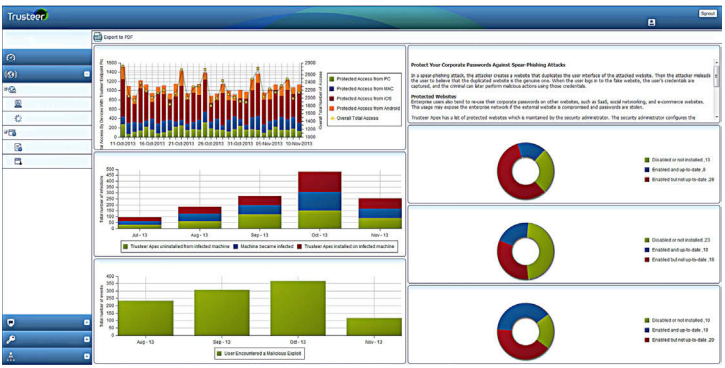

Figure 4-2: The Trusteer Apex dashboard provides enterprise security visibility.

Understanding End-User Impact

Stateful Application Control is designed to be lightweight and unintrusive, while transparently protecting user endpoints and preventing the exploitation of application weaknesses and data exfiltration.

Proven on millions of endpoints, this technology doesn't interfere with business applications or other security software products such as antivirus, HIPs (host-based intrusion prevention systems), website filtering, application whitelisting, or firewalls.

Stateful Application Control is an automated technology that is completely transparent to the user. It doesn't run intensive scans on the user machine so it doesn't impact system resource availability. Users are never asked to make decisions regarding unknown files (such as permitting access to resources), relieving them from guesswork and the need to make uninformed decisions. After all, most enterprise employees aren't security experts and don't know how to deal with security alerts.

IT security professionals are provided with all security event details including forensic data and an optional sample download so they can conduct security investigations without disrupting end users.

When security controls begin to impede end-user productivity, users figure out ways to go around the controls. Or remove them from their machines.

Boosting Protection with Real-Time Threat Intelligence

Trusteer has an extensive lab that researches and provides threat intelligence. In part, this intelligence comes from endpoints protected by Trusteer.

Trusteer's dashboard provides real-time status on advanced threats that are knocking on your door. Malware intrusions and extrusions are blocked in real time. Trusteer Apex automatically updates all protected endpoints with new legiti-

mate application states and new advanced threats. Trusteer researchers continuously analyze feeds provided by tens of millions of protected endpoints around the world. Automated updates are pushed to all protected endpoints as soon as they're available.

Because Trusteer provides the updates, IT security professional don't need to continuously update rules or policies, and no in-house expertise in needed for ensuring 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.

Trusteer Apex updates aren't blacklist updates like antivirus or intrusion prevention systems; instead they're relatively infrequent application state updates. Trusteer Apex updates aren't required to block new zero-day exploits.

Leveraging Web-Based Management

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 Trusteer manages the entire management infrastructure so you don't have to. With rising data center costs and IT departments being stretched razor thin, Trusteer's hosted solution in your data center is *zero footprint*.

Chapter 5

Top Ten Considerations for Effective Advanced Threat Protection

In This Chapter

- Stopping all known and unknown exploits
- Accurately blocking exfiltration
- Protecting endpoints without affecting end users

If you're considering advanced threat protection, the ten most important considerations for selecting a solution are found right here in this chapter.

Ability to Stop Malware Delivered by Exploits

An advanced threat solution should be able to stop the execution of malware even if it was silently delivered by an exploit. The malware should be stopped regardless of the application vulnerability being exploited to deliver it, the malware type, its source, or its destination. The solution should be able to protect against malware delivered by both zero-day and known exploits.

Not Detection-Based, Not Dependent on Patch Availability

Advanced threat protection should not rely on detection of threats because many of them are unknown. Daily updates of new malware characteristics should not be required to block new unknown zero-day threats. Instead, a solution should block zero-day threats based on the identification of invalid application actions that don't have a known context.

Accurate Exfiltration Prevention

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 threat protection solutions should prevent malware communications and data exfiltration, regardless of evasion techniques.

To prevent compromise, advanced threat protection solutions need to prevent data exfiltration on infected machines.

Protection of 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.

The primary considerations for protecting enterprise credentials are

- Protection against key-logger malware that targets employee login credentials
- Protection against phishing attacks that target employees with fake websites
- Preventing corporate credentials reuse on noncorporate sites to reduce the risk of password exposure

Minimal Impact on Users (Usability, Performance)

The best kind of protective software is the kind that is transparent to the users, which is to say that users are completely unaware of it. Everyone knows that users just want to get their work done: They aren't going to do any of the security-related chores you ask of them.

Don't rely on end users to make security decisions like "Should this application be allowed to change these registry keys and save these files to the file system?" Most users aren't experts in data security, nor should they need to be. In most cases, users can't understand the meaning of technical security alerts and therefore can't make the right decision.

Coverage for All User Platforms

In most organizations, multiple endpoint platforms are in use by employees. In addition, all the flavors of bring your own computer (BYOC) are resulting in many different types of unmanaged endpoint platforms. Make sure that any advanced threat protection solution you consider can protect all the platforms in use by employees in your organization.

Deployable on All Endpoints (Managed and Unmanaged)

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

For unmanaged endpoints (we're talking BYOC here), easily installed code snippets on VPN and web-based login pages will quickly get Stateful Application Control installed on all endpoints.

Minimal Ongoing Maintenance (Automated)

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.

Scales to Protect All Enterprise Employees

An effective advanced threat protection system is going to be successful only if it works in any size organization, whether 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.

Leverages 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.

Stop zero-day exploits with Stateful Application Control!

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 how Stateful Application Control is the most promising development yet to defeat 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 Stateful Application Control — the most promising new technology to 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 Stateful Application Control protects managed and unmanaged endpoints**

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: 978-1-118-75850-2
Not for resale