Trusteer

Stepping Up the Battle Against Advanced Threats

White Paper

*new threats, **new thinking***

Trusteer

Table of Contents

Introduction

The primary approaches used to fight cybercrime over the past several years simply aren't effective. Despite losing some of these battles, we can still win the war. However, a new approach is needed. Trusteer has pioneered a new cybercrime prevention approach that provides unparalleled protection against spear-phishing, drive-by downloads and advanced, information-stealing malware, which enable targeted attacks, with no management load for IT staff or disruption to end users.

Targeting the End Users

Cybercriminals use all sorts of tricks and social engineering tactics to fool employees, compromise their machines and corporate accounts, and gain a foothold in the enterprise network. The most common techniques used today are phishing and watering hole attacks.

Phishing and Spear-phishing

In a phishing attack, cybercriminals send users a message in an attempt to lure users to perform an action that will result in malware infection, credentials theft, or both. The message can be sent in the form of an email, Instant Message (IM), Facebook message, or a Twitter message (a.k.a. tweet). Depending on their type, messages can contain either a weaponized document or a link to a malicious website. In a spear-phishing attack, the attacker uses the same tools, only instead of the "shotgun" approach, the attacker personalizes the message and targets specific users. This is because a personalized message is often more convincing and trusted by the users.

The message may contain:

Weaponized attachments: a weaponized document (e.g. Word, Excel, or PDF) contains hidden malicious code. When the document is opened and the content is rendered by the application, the hidden code executes and exploits a vulnerability to download malware onto the user's PC.

Links to malicious sites: can lead users to two types of malicious sites:

- **Phishing sites:** these sites, which are designed to steal users' credentials (username and password), try to mimic the look and feel of legitimate websites, such as online banking and e-commerce websites and even Google Apps[1]. When a user accesses a phishing website and tries to log in, the credentials are sent to the attacker who can use them to log into the user's account and steal information, funds or both.
- **Exploit sites:** these sites contain a hidden malicious applet or code that exploits a browser (or browser plug-in) vulnerability to silently download malware to the user's PC. The user does not need to initiate the download and in most cases is unaware of the download taking place. This type of download is called a drive-by download.

Watering Hole Attacks

In a watering hole attack, cybercriminals compromise a legitimate website that is routinely accessed by a specific type or group of users. The compromised website becomes an exploit site and infects its visitors with malware. In a recent watering hole attack, a mobile application development site was compromised to serve up malware to the site visitors. As a result, developers from companies like Facebook, Apple and Microsoft were infected with a RAT (Remote Access Trojan). Although the perception may be that watering hole attacks cast a wider net, they are still highly targeted in nature.

1 http://theonion.github.io/blog/2013/05/08/how-the-syrian-electronic-army-hacked-the-onion/

The Three Lost Battles

Over the years, organizations have used various tools and techniques to block cyber-attacks and prevent attackers from gaining access into the enterprise network. It is now clear that three of these techniques have failed.

User Education: Explaining the Do's and Don'ts

Education and awareness programs are continuously developed to train employees to recognize common phishing and spear-phishing attack tactics, and the proper use of external content. The belief is that with proper education it is possible to reduce the risk of successful phishing attacks occurring through human error.

Despite the time and resources that organizations have put into training users, user education has failed to mitigate the risk. Training programs that explain the dangers of opening untrusted external content and clicking on suspicious links have not prevented users from doing so on a daily basis. This is not only due to enterprise users being naïve or careless. Users open untrusted content and click on suspicious links because phishing schemes can be very convincing. Attackers use information gained through social engineering to personalize the spear-phishing messages and convince the targeted user that the message is legitimate.

On June 28th 2013, the FBI issued a **warning** about the rise of spear-phishing attacks, saying, *"Often, the emails contain accurate information about victims obtained via a previous intrusion or from data posted on social networking sites, blogs, or other websites. This information adds a veneer of legitimacy to the message, increasing the chances the victims will open the email and respond as directed."*

A recent **Trusteer blog** entitled **Twitter Malware: Spreading More Than Just Ideas**, described how a new malware used fake Twitter messages and shortened URLs to spread more malware among Twitter users. In this case, the malware gained access to the victim's Twitter account and created malicious tweets containing URLs. Followers of those accounts received a tweet from the trusted source, with a link that led them to an exploit site, and infected their endpoints with malware. Because the malware compromised trusted Twitter accounts, and used shortened URLs (which mask the real URL), it was very difficult for users to identify such messages and links as malicious.

Additionally, user education can't protect users against the fairly recent phenomenon of watering hole attacks. In a watering hole attack, the attacker compromises a legitimate website and turns it into an exploit site. Compromising websites that employees need in order to perform their jobs is devious, as companies cannot block users from visiting these sites. It is practically impossible to train employees to avoid watering hole websites, as no one knows which sites have been compromised, and banning employees from accessing trusted sites required for their work is counter-productive.

The bottom line: as long as employees are dependent on online information, spear-phishing and watering hole attacks will remain a threat leading to credentials theft and malware infections.

Eliminating Vulnerabilities: Patching and Secure Code Development

Vulnerabilities in endpoint applications introduce significant risk to an enterprise, as they can be exploited[2] by cybercriminals to silently download malware onto the user's device.

Timely application of patches is critical for preventing the exploitation of known endpoint application vulnerabilities. However, failure to keep up with software patches is one of the most common challenges identified by security and IT professionals. New patches are released daily, making it difficult for even the most experienced system administrators to ensure proper deployment in a timely manner.

Some of the major attacks in the past few years have targeted known vulnerabilities for which patches existed. One example is a **recent targeted attack** on users in Vietnam, India, China, Taiwan and possibly other countries. In this attack, weaponized Word documents were sent to victims in spear-phishing emails. The weaponized documents contained an exploit targeting known vulnerabilities in installations of Microsoft Office (CVE-2012-0158 and CVE-2012-1856). A patch for these vulnerabilities was provided by Microsoft in 2012 as part of the MS12-027 and MS12-060 security bulletins. Despite being relatively old, these Word vulnerabilities continue to be exploited in targeted attacks. In this case, the exploit was used to download 'KeyBoy' – a malicious backdoor program that steals credentials stored in Internet Explorer and Mozilla Firefox and installs a keylogger that steals credentials entered into Google Chrome. The backdoor program also allows the attackers to get detailed information about the compromised computers, browse their directories, and download or upload files from and to them. In addition, the malware can be used to open a Windows command shell on the infected computers that, in turn, can be used remotely to execute Windows commands.

Clearly, the timely patching of application vulnerabilities is important, but it is not enough. The increasing frequency and sophistication of zero-day attacks has highlighted the need for more proactive measures. Zero-day vulnerabilities, which are software vulnerabilities unknown to the vendor, are a top concern of security practitioners since no patches exist. Attackers who are aware of the vulnerability can quickly develop a zero-day exploit, a piece of code that exploits an unknown vulnerability to silently download malware onto the user's PC, and embed the exploit in a webpage or email attachment. Users who access the malicious webpage or open the weaponized document cannot prevent (or even see) the exploitation of the vulnerability. As a result, they are infected with malware. As long as the vulnerability isn't patched, users will continue to be infected with malware.

A recent example of a **targeted attack that exploited a zero-day vulnerability** resulted in the infection of users in over 37 countries with the infamous "Poison Ivy" Trojan. The attackers used a watering hole attack, in which they compromised a US Department of Labor website that is regularly visited by American

2 Vulnerability exploitation is the process in which a cybercriminal takes advantage of a weakness in an application to change its intended behavior.

government employees and contractors in the nuclear research sector, and Europeans working in the defense, security and aerospace industries. Visitors to the website were redirected to another site where malicious code targeted those using the Internet Explorer browser. The code exploited a zero-day (unknown) vulnerability in Microsoft Internet Explorer 8 (IE8) to download the Trojan to the victims' endpoints. The attackers may have used the Trojan to collect sensitive military information on behalf of a nation-state. Microsoft was only able to provide a patch for this vulnerability a month after the attack was discovered.

Secure Coding Practices

Increased awareness of the risk introduced by application vulnerabilities, specifically zero-day vulnerabilities, have accelerated the introduction of secure coding/programing initiatives. Secure coding guidelines developed by organizations such as **SANS**, **OWASP** and **CERT** promote the concept of "secure applications by design." These guidelines support solution architects and developers in their efforts to conceive, develop, acquire, operate, and maintain hardened applications, and minimize software vulnerabilities. Secure coding has been getting significant attention over the last few years, and training programs designed to educate developers about the importance of secure coding, and practices that should be followed, have been introduced in many organizations. So far these important initiatives have not eliminated vulnerabilities. In fact, in 2012 the number of publicly reported software vulnerabilities jumped by 26 percent, the biggest increase in five years[3]. While this does not mean that secure coding initiatives have failed, it does suggest there is still a long way to go.

Nowadays, security professionals must assume that vulnerabilities exist in Internet-facing software installed on user endpoints and that these vulnerabilities can be exploited to silently download malware and infect their machines. Waiting for a patch is simply not enough.

Malware Detection: Blacklisting and Behavior Analysis

It was once the industry's "dirty little secret," but today security experts agree: even when anti-virus applications work perfectly, they still fail to block sophisticated malware attacks.

Anti-virus solutions, which first appeared in the late 1980s, use several blacklisting methods to identify viruses and other malware:

- **Signature-based detection**: comparing the contents of a file to a dictionary of known virus/malware signatures
- **Heuristic-based detection**: comparing the heuristics of a file to a dictionary of known malware heuristics
- **Behavior-based detection**: comparing the behaviors of the file in a virtual sandbox test environment to known malware behaviors

3 http://www.darkreading.com/vulnerability/lessons-learned-from-a-decade-of-vulnera/240148896

Most anti-virus solutions are host-based, scanning the host file system for known malicious files. However, due to the performance impact on user endpoints, some anti-virus and malware detection vendors have moved the detection process to network appliances, but the detection methods remain unchanged.

Many users still trust that malware detection solutions will protect their enterprise endpoints and home computers against advanced threats. In the current threat landscape, however, malware detection solutions fall short. The recent breach into the **New York Times** demonstrates that blacklisting detection methods used by anti-virus vendors (in this case, Symantec) cannot prevent attackers from gaining entry into the corporate network: According to the New York Times, in 2012 Chinese hackers persistently attacked there systems for over 4 months, infiltrated computer systems and stole reporters' and employees' credentials. The attackers infected user endpoints with malware and stole the corporate passwords of every Times employee to gain access to the personal computers of 53 employees, most of them outside the Times's newsroom.

According to Mandiant, the data breach response firm hired by the Times, out of the 45 different pieces of malware planted on NYT systems over the course of the attack, only one was spotted by the Symantec antivirus software the Times used. The other 44 were only found during Mandiant's post-breach investigation months later.

Today's hackers are creating new malware faster than anti-virus vendors can blacklist them. AV-Test, a research institute that tests anti-virus products, says it registers **more than 200,000 new kinds of viruses** every day. In addition, attackers use polymorphic code to continuously mutate malware and evade anti-virus detection.

Anti-virus vendors have introduced behavior-based detection methods to better identify new, unknown malicious files and battle polymorphic code evasion techniques. This approach emulates unknown, untrusted file execution in a sandbox environment (typically on a network appliance) and, based on the file behavior, determines if the file is malicious or benign. Naturally, attackers have responded with techniques to evade these detection solutions as well. By designing malware that 'sleeps' for hours or days, or waits for a mouse-click, the malware can avoid detection in synthetic sandbox environments. Once the malware gets to the user endpoint, where mouse-click events are abundant, it will compromise the endpoint.

Another method for detecting malware is by monitoring outbound traffic for data exfiltration. Such solutions look for communication with known command and control servers (C&Cs) or network behavior profiles that indicate malicious communication. However, attackers have also devised evasion techniques to bypass these solutions. These include usage of legitimate sites like **social networks** and **Google Docs** as proxies to hide the malicious traffic, and usage of custom communication protocols. We've also seen malware authors incorporate 'sideways' P2P communication so there is no one set of addresses that can be blocked.

Simply put, malware detection rules will be bypassed. Every time a new detection method is developed, hackers study its blacklisting rules and develop new evasion techniques. It is clear that anti-virus cannot prevail against today's advanced malware. Despite the fact that security experts agree that anti-virus solutions fall short, many organizations still continue to buy and implement anti-virus solutions in order to adhere to compliance regulations such as SOX, PCI-DSS, GLBA, HIPAA and more, which explicitly require implementation of an anti-virus solution.

The Importance of Application Control

Application Control: Harder to Evade, But Difficult to Deploy and Maintain

Application control technologies are used to ensure that only approved applications and their associated executables are permitted to run on the endpoint. The most well-known application control approach is based on file whitelisting. File whitelisting is the opposite of the blacklisting approach used by anti-virus applications, essentially using a list of approved/certified files, instead of a list of known malicious files. It is more difficult for malware to bypass this control because whitelisting does not use detection rules. That said, file whitelisting solutions introduce a different challenge - the setup and maintenance of the whitelist. The whitelist must contain every file and application that any user in the organization might use.

Consider the fact that the average endpoint contains 20,000 executable files. The whitelist must include every update, every patch and every executable required for internally-developed applications. As a result, the whitelist that an organization must maintain becomes complex and difficult to manage. For example, Kaspersky's whitelisting solution uses a database that contains **over 700 million unique files** and this database is constantly updated by the company and by more than 300 partner software vendors. Another vendor, Bit9, uses a database called the Global Software Registry (GSR) which contains **over 8 billion records** with hundreds of thousands of new files added every week. Due to this complexity, large enterprises are struggling to implement and maintain enterprise-wide deployments, especially for dynamic, internet-facing endpoints, leaving user endpoints vulnerable to malware infections.

A different application control approach isolates application tasks by executing the tasks in a virtual environment. When an isolated application is compromised, the threat remains inside the virtual environment and does not infect the underlying host. This can be a very strong security control, but it also introduces many challenges. Primarily, end user applications are not designed to run in isolation. On the contrary, applications are designed to inter-operate. Think about the simple copy-paste capability which allows users to copy content from one application (like the browser) and paste it into another application (like a Word document). Applications are also designed to interact with the underlyng host, such as when saving files to the file system, printing files, etc. These functions require the definition of special policies and the installation of special drivers to enable basic business workflow without impacting user productivity. This becomes very challenging in large enterprise environments that consist of a variety of users, endpoint platforms and applications, and could potentially harm legacy applications.

Stateful Application Control: Next-generation Advanced Malware Protection

Since user education, vulnerability elimination, malware detection and application control approaches provide scarce protection against advanced malware, a new approach is needed. Organizations need a solution that is not dependent on the end user, patch availability or malware detection methods and that is easy to implement and maintain across all enterprise endpoints.

Trusteer Apex applies a new, ground-breaking approach to advanced malware protection: **Stateful Application Control**. By analyzing what the application is doing and why it is doing it, it automatically and accurately determines if an application action is legitimate or malicious. Using this approach, Trusteer Apex is able to block vulnerability exploitation and silent malware downloads. It also blocks malware communication and data exfiltration which enables the attacker to gain a foothold in the network. Moreover, Trusteer's Stateful Application Control enables automated enterprise malware protection that maximizes security while simplifying deployment and minimizing management overhead.

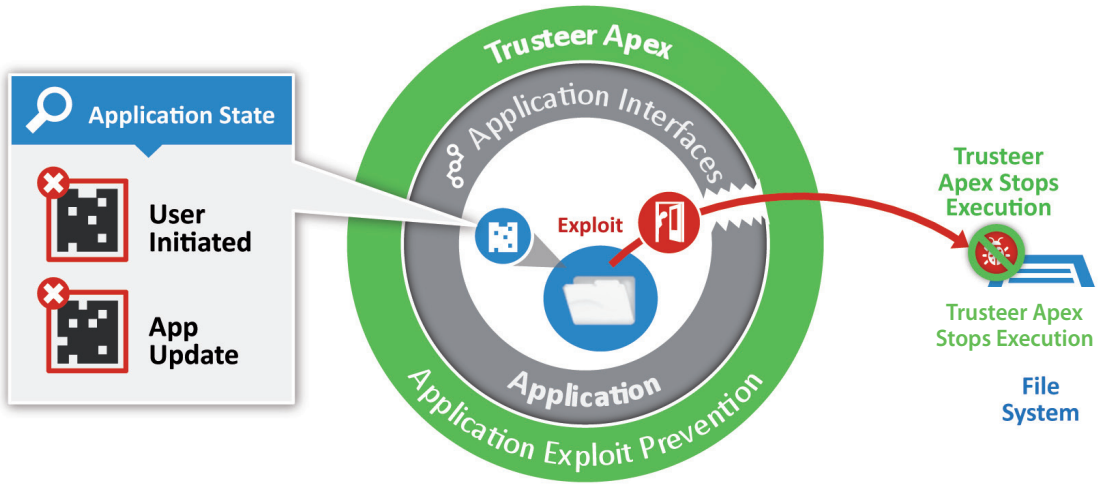

Figure 1: Stop Application Actions with Unknown State

Unlike other security controls, Stateful Application Control does not try to identify malicious files or control their execution. Instead, it stops the silent download of malware via vulnerability exploitation by validating the state of the application during sensitive functions. For example, when an application downloads a file for a legitimate reason (e.g. when the user selects the 'Save As' option from the application menu) a specific "application state" is created (the state of the memory and kernel level processes). By analyzing the application states during normal operations, Stateful Application Control maps the legitimate application states of the targeted applications (i.e., browsers, Adobe, Flash, Java) when these applications write to the file system. These mapped application states provide the context of the action, allowing an accurate determination of why the file was downloaded.

Stateful Application Control uses this map to validate the application's operations; when the application downloads a file, the control verifies that a known, legitimate application state was created and if so, it allows the action to continue. But, if the application downloads a file as a result of an exploit, an unknown application state is created; one that doesn't match any of the mapped application states. In that case, the file is stopped so that it can do no harm.

One of the main advantages of Stateful Application Control is that it stops the exploitation process no matter what vulnerability is being exploited. It doesn't matter if the vulnerability is known or unknown (zero-day), what kind of malware it is trying to download to the endpoint, the malware's source or its destination. As soon as an unknown application state is created, the exploitation process is stopped and the downloaded file is stopped. This makes Stateful Application Control a powerful exploit prevention solution which can stop any type of exploit and is not susceptible to evasion.

Stateful Application Control allows for more stable, effective, and manageable endpoint security than traditional application control approaches because it is focused on exploitable applications for which the legitimate application states are few and relatively static. This reduces the maintenance required compared to other application control approaches that must inspect and manage a multitude of files.
The key to implementing Stateful Application Control is making it highly manageable - so that it requires no end user intervention and minimal IT staff involvement. This can only be accomplished through a sizeable network of endpoints that enables new, legitimate application states to be detected, and immediately pushed out to all protected endpoints via the cloud.

Stateful Application Control enables the following capabilities:

- **Application Exploit Prevention:** Trusteer Apex blocks malicious code embedded in web pages and business documents from exploiting zero-day or unpatched vulnerabilities in client applications and installing malware on the endpoint.
- **Data Exfiltration Prevention:** Trusteer Apex restricts untrusted files from executing sensitive operations that are potentially malicious. For example, tampering with other application processes to hide communication traffic to a command and control center. Untrusted files are sent to Trusteer for analysis and are either approved or removed from the endpoint.
- **Ease of Deployment and Automated Management:** Trusteer Apex can be deployed within days, over tens of thousands of endpoints, both managed and unmanaged, and is specifically designed to support large and complex environments. No learning period is required and no initial or ongoing configuration is necessary.

Protecting Enterprise Credentials

In the beginning of this paper we explained that attackers often target corporate employee credentials. Compromised credentials allow the attacker to gain fraudulent access to the corporate network and resources. Attackers can gain corporate credentials by stealing them off user endpoints, using keyloggers, or by stealing credentials on the Internet. This can be done by using phishing sites (like a fake GoogleApps login page), or by stealing the user database of public websites; since employees often reuse their corporate credentials on public consumer sites like e-Bay and Amazon, or social networks, the site's user database can provide the attacker with valuable credentials. This can result in a data breach that has significant impact on the corporate business.

To secure enterprise credentials against key-logger and phishing attacks, and prevent exposure through public sites' user databases, Trusteer added the following protections:

a. Obfuscating keystrokes on the endpoint, preventing key-loggers from capturing the actual keystrokes.

b. Validating that corporate credentials are used online only to log into approved enterprise web applications. This control prevents users from submitting their credentials on phishing sites. It also prevents the employees from reusing their corporate credentials on public consumer websites (such as ebay, Sony, Amazon) and social networks (such as Facebook, Twitter, LinkedIn).

Conclusion

The endpoint has become the path of least resistance for cybercriminals and hackers to get a foothold into enterprise networks. Advanced information-stealing malware is the main tool that enables APTs and targeted attacks on enterprises. Traditional methods such as user education, vulnerability patching and malware detection have failed to protect enterprises against the current threat landscape. Attackers continuously develop sophisticated tactics and evasion techniques to bypass the latest protection methods, requiring the security industry to find a different approach to malware protection.

Trusteer Apex leverages Stateful Application Control, a ground-breaking technology that provides effective protection against advanced malware by stopping exploitation of vulnerabilities and silent malware downloads. Because it does not rely on users' judgement, patch availability or malware detection rules, it is effective even against unknown, zero-day threats.

Trusteer Apex includes a data exfiltration prevention layer that prevents malware from communicating with command and control servers (C&Cs) and exfiltrating data. It also includes specific features to protect enterprise credentials against theft and exposure.

Delivered as a light software agent, Trusteer Apex is easily deployed on both managed and unmanaged endpoints. It transparently runs on the endpoint, protecting against advanced malware, without impacting the endpoint performance or user experience. Automation of the solution, enabled by its Stateful Application Control technology, ensures it requires minimal ongoing maintenance. Automated updates are provided by Trusteer's research lab and delivered directly to the agents wherever they are. Centralized management and reporting enable Trusteer to provide a simple and cost effective solution to a growing problem.

Security professionals who are concerned about the growing frequency and sophistication of threats targeting employee endpoints now have a solution that accurately protects endpoints against advanced threats, yet is easy to deploy and manage in a dynamic user environment.

Trusteer

About Trusteer

Boston-based Trusteer is the leading provider of endpoint cybercrime prevention solutions that protect organizations against spear-phishing, and advanced malware that enable targeted attacks and data breaches. Hundreds of organizations and millions of end users rely on Trusteer to protect their managed and unmanaged endpoints from online threats and advanced information-stealing malware. For more information please visit: **www.trusteer.com**.

Trusteer Inc.
545 Boylston Street, 5th Floor
Boston, MA 02116
T: +1 (866) 496-6139
T: +1 (617) 606-7755
info@trusteer.com

trusteer.com

Trusteer