THE ATTACKER'S MINDSET

Insights into real-world attack campaigns

Introduction

In this paper, we will examine the attacker's mindset, first with insights into the motivations and high-level methods shaping today's threat landscape and accelerating the need for endpoint security technologies that can prevent – not just detect – known and unknown threats. We will then go on to examine a series of real-world attack campaigns, revealing attackers' individual steps and tactics for everything from distribution to installation and execution of a malicious payload. Finally, we will show how Palo Alto Networks® Traps™ advanced endpoint protection breaks the attack lifecycle for each attack, often at multiple points, delivering a level of protection far beyond the capabilities of legacy antivirus and more modern endpoint detection and response solutions – including against zero-day threats.

No One Is Immune

The motivations and high-level game plans of today's threat actors go a long way in explaining the current threat landscape, not to mention the relative ineffectiveness of existing endpoint protection technologies.

The Attacker's Mindset

Motivation: A handful of years ago, attackers did what they did to build a reputation, notoriety and even fame – but that's no longer the case. With few exceptions, such as hacktivism, the primary motivation behind most of today's attacks is simply money. Whether financial gain for the attacker (e.g., via ransomware or the theft of salable data) or financial loss for the target (e.g., via theft/exposure of proprietary information or destruction of digital property), it all comes down to dollars and cents.

There's even a lucrative threats-as-a-service market offering threat tools, such as botnets and ransomware kits, for rent. Here, the creators profit by selling their wares to others instead of conducting attacks themselves.

It's also important to acknowledge the role digital currencies, such as bitcoin, have played in this progression. Like fuel on a fire, they have served as an accelerant for the shift to money as a motivator, as well as the volume and sophistication of threats being generated, by making it trivially easy to convert effort (e.g., designing a new threat) and output (e.g., stolen data) into real money.

Plan of attack: We'll dive deeper into attacker tactics shortly, but for now, we remain at a high level. The shift to money as a motivator brought with it a fundamental change in most attackers' general approach – from rattling doors, exhibiting obvious impact and leaving calling cards to avoiding getting caught at all. Indeed, the watchwords for attackers today are success and stealth.

Today's attackers pay more attention to detail to ensure a payday, including a greater emphasis on avoiding detection to effectively prolong the productive duration of a successful attack and maximize its financial return. As we'll see shortly, concrete manifestations of these high-level objectives include the use of "safety checks" and compartmentalization at multiple points in an attack.

Results: These changes to attacker motivation and strategy bring us to a threat landscape characterized by high volumes, growing sophistication and greater successes, as demonstrated by these statistics:

- **Attacks in general:** According to the 2017 Cyberthreat Defense Report, nearly 80 percent of organizations were affected by a successful cyberattack in 2016, with a full third breached at least six times in the span of a year.1
- **Ransomware:** More than 60 percent of organizations were affected by ransomware in 2016, with nearly a third of these acquiescing to the associated ransom demand.[2] In addition, Cybersecurity Ventures® predicts ransomware damage costs will exceed $5 billion in 2017, up more than 1,500 percent from 2015.[3]
- **Malware in general:** According to AV-TEST, there were approximately 127 million new malware samples in 2016. With nearly 38 million new samples identified in the second quarter alone, 2017 appears to be on a similar trajectory.[4] This equates to a new malware variant appearing every four seconds.

Endpoints on the Front Lines

Although attacks can target practically any part of your organization's infrastructure, endpoints, in particular, are proving to be a major battle ground. Several IT trends are elevating the importance of more effective security at the endpoint. One example is the steady growth of user mobility across all industries, which is significantly increasing the quantity and diversity of endpoints requiring support.

A second example is the rising use of cloud services. Software- and infrastructure-as-a-service offerings are increasingly being pursued as ways to trim costs and keep pace with aggressive competitors. However, these services inherently remove points of enterprise control, thereby increasing the importance of those that remain, such as endpoints.

On top of everything else is the broadly accepted truth that users are the weakest link in most organizations' security defenses. Users' susceptibility to phishing scams, tendency to click on dubious links and willingness to install/run suspicious applications makes endpoints the most likely points of entry for ransomware, malware and other threats.

Failing to Report for Duty

With endpoints squarely on the front lines for many of today's attacks, enterprises must establish and maintain effective endpoint defenses. However, many endpoint security technologies are ineffective against modern threats.

Approaches that rely primarily on signatures and behavior-monitoring techniques are ineffective against the volume and diversity of sophisticated threats, including ransomware and other types of malware. Other "next-gen" products – those classified as "endpoint detection and response," or EDR – have also struggled to effectively prevent endpoints from compromise. EDR solutions focus on reducing damage by collecting and analyzing data from endpoints in hopes of identifying anomalous activities. The result remains the same, however, as they cannot effectively prevent cyber breaches from happening in the first place.

Consider ransomware attacks, such as WannaCry or Petya. A product that relies on EDR will identify that you've been hit – a fact that will already be painfully obvious if normal operations have turned upside down, and salespeople and rank-and-file employees no longer have programmatic access to client accounts or other business-critical data. An EDR solution might also expose details about when and how an attack occurred, or how far the infection has spread, but this is a case of too little, too late. Furthermore, it takes substantial effort and expertise to process, analyze and interpret all the telemetry such a solution collects.

Today's enterprises need a more effective approach to endpoint security – one that delivers prevention first to stop both known and unknown threats in their tracks.

Introducing Traps Advanced Endpoint Protection

Palo Alto Networks Traps focuses on preventing threats before they compromise endpoints. Leveraging multiple methods of prevention against malware and exploits, Traps effectively blocks known and unknown threats at multiple points in the attack lifecycle. In the remainder of this paper, we will illuminate the power of this approach by examining several real-world attack campaigns and showing how Traps can thwart them, even as attackers continue to evolve their tactics.

Terminology

Before proceeding, it's important to define a few terms.

An attack campaign is a set of activities carried out by threat actors using specific techniques for a particular purpose, such as stealing financial information or targeting a certain business sector.[5] The distinction is the set of techniques involved, for example, to land a malicious payload on a target system.

An attack lifecycle is the sequence of steps (techniques) an attacker uses to achieve a given goal. Different campaigns are likely to have different attack lifecycles. For that matter, a campaign's lifecycle may change over time to prolong the campaign's period of usefulness/productivity. Defenders should strive to uncover the specific steps of a campaign's attack lifecycle in order to reveal available opportunities to prevent associated breaches.

Fundamentally different from malware, exploits are weaponized data files or content designed to leverage vulnerabilities in legitimate applications or operating systems to execute malicious code. When such a file is opened, the malicious code exploits the software's processes for its own purposes, including executing further malicious code. Because this type of attack is difficult to distinguish from normal application behavior, it typically bypasses legacy antivirus, whitelisting and other endpoint security technologies. To be truly effective, an endpoint security offering must be able to prevent both malware (i.e., self-contained, malicious executable files) and exploits.

Last but not least, an exploit kit is a collection of multiple exploits an attacker can use to enhance a campaign's chances of success by having more than one way to compromise a given endpoint.

Campaign #1: Exploit Kit-Based Attacks

Exploit kit-based attacks have been around for many years, and EITest is a classic example. The key to its longevity has been its evolution over time. After originally using the Angler exploit kit in 2014, EITest later changed to the RIG kit. Its attack lifecycle was tweaked at one point (in response to Unit 42, the Palo Alto Networks threat intelligence team, burning some of its assets). Its lifecycle – fairly typical of attacks that leverage exploit kits – is as follows:

Stage 1: Malicious JavaScript, or JS, is made to run in a victim's browser in one of two ways:

a) Malvertising, wherein an attacker buys ad space that appears on legitimate webpages. When the iFrame that constitutes the ad is filled by the attacker's site, the content includes the attacker's malicious JS code. Clicking on thead – or in some instances, merely visiting the affected page – deposits this code into the victim's browser.

b) Taking advantage of unchanged, default admin accounts or unpatched software to compromise a legitimate website and inject the malicious JS. Accompanying tactics may include deleting local logs (to wipe suspicious activity) and configuring a blacklist for security vendors' sites (so the compromised site won't send a copy of the malicious JS).

Stage 2: The injected JS completes preliminary fingerprinting of the endpoint, looking for signs of sandboxing or the presence of threat hunting tools (e.g., Wireshark®, IDA). Because the JS has a limited set of privileges/capabilities based on its context – for example, it can't read the registry or create/delete items from the local file system at this stage – the victim isn't considered breached at this point.

Tactic: To hide its meaning, malicious JS is often highly obfuscated (e.g., mixed with gibberish code). However, because obfuscation can raise flags for threat researchers, EITest shifted away from it.

Stage 3: The JS downloads the landing page or attack page code if the target is determined to be clean. This code first conducts additional fingerprinting to identify vulnerable software packages on the target system. A lack of vulnerable software will trigger serving of a legitimate page and termination of the attack. Otherwise, the landing page will download an applicable exploit from the exploit kit being used (e.g., Angler, RIG).

Tactic: A campaign can use a one-time token to allow an instance of the landing page code to download an exploit. Without a valid token, security researchers cannot get the attacker's site to send them copies of exploits for analysis.

Stage 4: The exploit takes advantage of a vulnerability to compromise the target system. Invariably, this process involves using a small, relatively fixed set of methods to circumvent inherent defenses of the target operating system – for example, by chaining return-oriented programming, or ROP, techniques to defeat Data Execution Prevention, or DEP.

Stage 5: The running exploit transfers control to a shellcode module that, unlike the malicious JS described earlier, has complete access to the client operating system. Shellcode is not constrained to the JavaScript virtual machine. At this point, the target system is breached.

Stage 6: The shellcode downloads the malware payload, puts it in the local file system and triggers its execution, typically using a one-time token. Sample payloads associated with EITest include Cerber (ransomware), CryptoMix (ransomware), GootKit (information stealing) and Chthonic (banking Trojan).

Tactic: Compartmentalization is used throughout, as evidenced by the separation into stages. For example, not combining the attack code of the payload directly into the shellcode limits "damage" – if researchers burn one component, the other remains unaffected; provides the flexibility of using different payloads over time; and enables persistence – the shellcode only resides in memory, whereas the payload resides in the local file system.

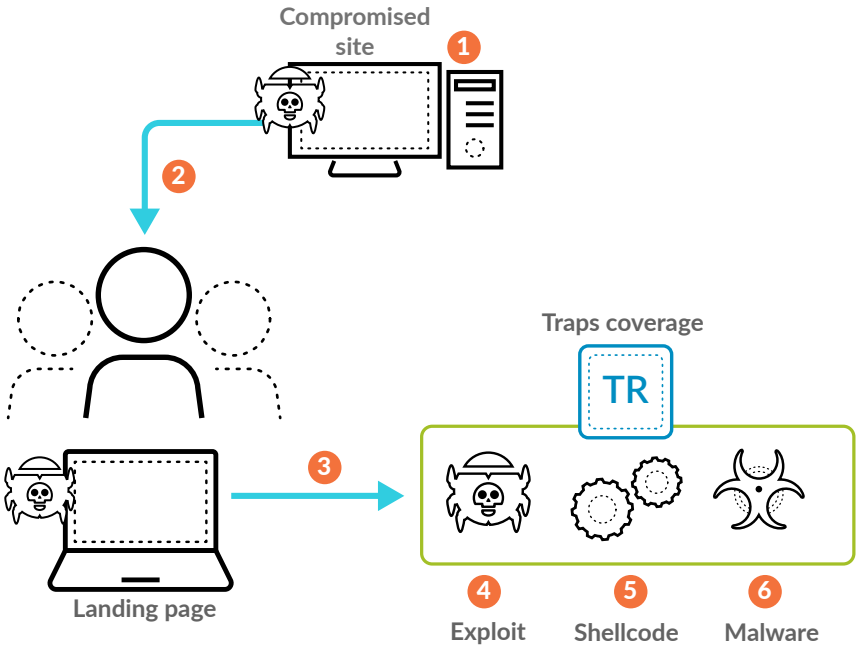

Figure 1: EITest attack lifecycle with Traps coverage

Traps Versus Exploit Kit-Based Attacks

Traps prevents exploit kit-based attacks like EITest by disrupting Stages 3 through 6 of the attack lifecycle (see Figure 1). Specifically, the multiple methods of prevention Traps employs include:

Pre-exploitation protection (Stage 3): By blocking the vulnerability profiling ("fingerprinting") attempts these campaigns often use, Traps stops many exploits at the earliest stage of operation.

Technique-based exploitation prevention (Stages 4 and 5): Traps recognizes and blocks the set of methods, or patterns, upon which all exploits rely. The result is a small-footprint, low-processing way to prevent known and zero-day exploits. There is also a module for blocking suspicious shellcode behavior and activities.

Kernel exploitation prevention (Stage 5): If the campaign uses kernel exploits, Traps will prevent exploits that leverage related vulnerabilities to allow processes to run with the escalated system-level privileges obtained from another process.

In the unlikely scenario that an exploit reaches Stage 6, Traps also features an extensive set of malware prevention capabilities, which we will cover in the next section.

Campaign #2: Banking Trojan Attacks

An example of a classic banking Trojan comes from the Ursnif malware family. Although it can also be distributed via web exploit kit, the following scenario involves spam-delivered malicious attachments. The attack lifecycle – typical of banking Trojans in general – is as follows:

Stage 1: A spambot sends emails with Word attachments that contain malicious macros.

Stage 2: Social engineering embedded in the message convinces the user to "allow macros" when the operating system presents a pop-up meant to keep arbitrary documents from running arbitrary code.

Stage 3: The malicious VBScripts begin with "safety checks" to ensure they've landed on a suitable target. For example, if the current username is "sandbox," the recent docs directory is empty and/or the filename for the document includes only hex characters, the target system is likely involved in threat research, and the attack will be aborted.

Stage 4: The malicious VBScripts download the malware payload, put it in the local file system and trigger its execution. To help avoid detection, the download might use an encrypted channel.

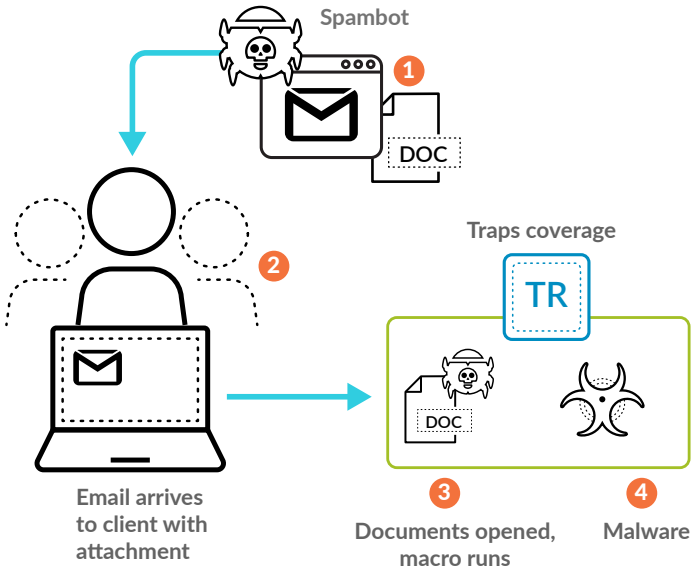

Figure 2: Ursnif spam macro attack lifecycle with Traps coverage

Traps Versus Banking Trojans

Traps prevents banking Trojan campaigns, such as Ursnif, that can utilize malicious macros by disrupting Stages 3 and 4 of the related attack lifecycle (see Figure 2). Specifically, Traps multi-method malware prevention includes:

WildFire threat intelligence: Traps prevents previously seen malware and macros by leveraging Palo Alto Networks WildFire® cloud-based threat analysis service. For any unknown executable file or macro it encounters, Traps submits a corresponding hash to WildFire. If WildFire returns a malicious verdict, Traps automatically blocks and remediates the associated file or macro.

Local analysis via machine learning[6]: This method delivers a verdict for any unknown executable file before it can run. Traps examines hundreds of the corresponding file's characteristics in a fraction of a second, without reliance on signatures, scanning or behavioral analysis.

WildFire inspection and analysis: In parallel with local analysis, unknown executable files are automatically submitted to WildFire for more detailed analysis. WildFire goes beyond ordinary sandboxing approaches used to detect unknown threats by combining the benefits of four independent techniques for high-fidelity, evasion-resistant discovery: dynamic analysis, static analysis, machine learning and bare metal analysis. When new malware is found, WildFire automatically creates and shares a new prevention control with Traps in as few as five minutes, without the need for human intervention.

WildFire is the world's largest distributed sensor system focused on identifying and preventing unknown threats, with more than 20,500 enterprise, government and service provider customers contributing to the collective immunity of all users.

Malicious process control: Traps delivers fine-grained control over the launching of legitimate applications and processes (such as script engines and command shells) that can be used for malicious activities. For example, Traps can prevent Internet Explorer® from launching a specific script interpretation engine as a child process–a common technique of ransomware.

Campaign #3: Mac Attacks

With more than 90 percent of enterprises using Mac® endpoints in some capacity, it is not surprising to see them increasingly coming under attack.[7] Malware, in particular, is becoming a vexing issue – the first quarter of 2017 alone saw the introduction of more new samples of Mac malware than emerged over all of 2016.[8]

KeRanger was the first fully functional ransomware targeting Mac OS X®. Discovered by Palo Alto Networks in March 2016, KeRanger was distributed by infected versions of the installer for the Transmission BitTorrent® client. Because these installers were signed with valid Mac app development certificates, they were able to bypass Gatekeeper – a native protection mechanism that automatically prevents the execution of improperly signed apps.

Traps defeats Mac malware like KeRanger by uploading every Mach-O file it encounters to WildFire for analysis. For those instances where a malicious verdict is returned, new protections are distributed to all Traps clients in as few as five minutes. Traps also enhances standard Gatekeeper functionality by providing options to block all child processes or only allow those with equivalent or higher-level signatures than their parent processes.

Conclusion

The majority of traditional and modern endpoint security products cannot keep pace with the rapid rate at which new and increasingly sophisticated threats are emerging, and commonly do not focus on prevention. In contrast, the multiple methods of prevention employed by of Palo Alto Networks Traps stop modern threats at multiple points along their respective attack lifecycles. The result is an unmatched capability for preventing known and unknown ransomware, malware, and exploits before they're able to compromise an organization's endpoints.

For more information about Palo Alto Networks Traps advanced endpoint protection, visit: https://www.paloaltonetworks.com/products/secure-the-endpoint/traps

1. CyberEdge Group, 2017 Cyberthreat Defense Report.
2. ibid.
3. https://cybersecurityventures.com/ransomware-damage-report-2017-5-billion/
4. https://www.av-test.org/en/statistics/malware/
5. https://stixproject.github.io/documentation/idioms/campaign-v-actors/
6. For more information on how Traps takes advantage of innovative machine learning technology, see Machine Learning and Endpoint Security – Separating Hype from Value
7. Jamf, 2016 Survey: Managing Apple Devices in the Enterprise.
8. Jai Vijayan/Dark Reading, Ransomware, Mac Malware Dominate Q1 Threat Landscape, April 2017.

3000 Tannery Way
Santa Clara, CA 95054

Main: +1.408.753.4000
Sales: +1.866.320.4788
Support: +1.866.898.9087

www.paloaltonetworks.com

the-attacker-mindset-wp-013118