What Is Fileless Malware?

Fileless malware is a type of malicious software that uses legitimate programs to infect a computer. It does not rely on files and leaves no footprint, making it challenging to detect and remove. Modern adversaries know the strategies organizations use to try to block their attacks, and they're crafting increasingly sophisticated, targeted malware to evade defenses. It's a race against time, as the most effective hacking techniques are usually the newest ones. Fileless malware has been effective in evading all but the most sophisticated security solutions.

Fileless malware emerged in 2017 as a mainstream type of attack, but many of these attack methods have been around for a while. Frodo, Number of the Beast, and The Dark Avenger were all early examples of this type of malware. More recent, high-profile fileless attacks include the hack of the Democratic National Committee and the Equifax breach.

What makes fileless infections so insidious is also what makes them so effective. There are claims that fileless malware is "undetectable." This isn't literally true, it just means that fileless attacks are often undetectable by antivirus, whitelisting, and other traditional endpoint security solutions. In fact, the Ponemon Institute claims that fileless attacks are 10 times more likely to succeed than file-based attacks.

How does a fileless attack happen?

Fileless attacks fall into the broader category of low-observable characteristics (LOC) attacks, a type of stealth attack that evades detection by most security solutions and impacts forensic analysis efforts. While not considered a traditional virus, fileless malware does work in a similar way—it operates in memory. Without being stored in a file or installed directly on a machine, fileless infections go straight into memory and the malicious content never touches the hard drive. Many LOC attacks take advantage of Microsoft Windows PowerShell, a legitimate and useful tool used by administrators for task automation and configuration management. PowerShell consists of a command-line shell and associated scripting language, providing adversaries with access to just about everything and anything in Windows.

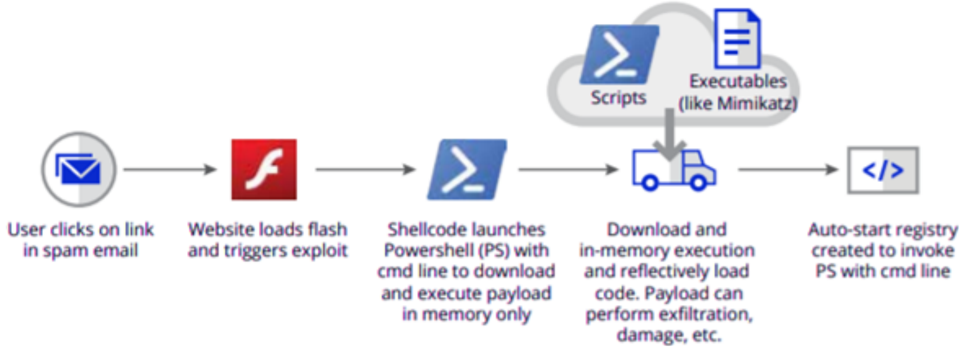
Scripts
Executables (like Mimikatz)
User clicks on link in spam email
Website loads flash and triggers exploit
Shellcode launches Powershell (PS) with cmd line to download and execute payload in memory only
Download and in-memory execution and reflectively load code. Payload can perform exfiltration, damage, etc.
Auto-start registry created to invoke PS with cmd line

Figure 1. Example of a fileless attack kill chain.

The figure above illustrates how a fileless attack can happen. Like most advanced attacks today, fileless attacks often use social engineering to get users to click on a link or an attachment in a phishing email. Fileless attacks are typically used for lateral movement, meaning they make their way from one device to the next with the objective of gaining access rights to valuable data across the enterprise network. To avoid suspicion, fileless malware gets into the inner recesses of trusted, whitelisted applications (like PowerShell and Windows script host executables such as wscript.exe and cscript.exe) or the operating system to initiate malicious processes. These attacks abuse the trust model used by security applications to not monitor whitelisted programs.

What is important to notice in the above scenario is that the hacker did not have to figure out how to sneak a malicious program past antivirus and malware defense. Most automated sensors cannot detect command line alterations. A trained analyst can identify these scripts, but often doesn't know where to look in the first place.

How can you defend against fileless attacks?

As the cybersecurity industry gets more sophisticated in closing off exploits, the lifespan of fileless attacks gets shorter and shorter. One way to defend against fileless infections is simply keeping your software up to date. This especially includes Microsoft applications, and the launch of the Microsoft 365 suite includes enhanced security measures. Microsoft has also upgraded its Windows Defender package to detect irregular activity from PowerShell.

The real key to successfully counteracting fileless attacks is an integrated approach that addresses the entire threat lifecycle. By having a multi-layered defense, you gain an advantage over attackers by being able to investigate every phase of a campaign before, during, and after an attack.

Two things are especially important:

- The ability to see and measure what's happening: discovering the techniques used by the attack, monitoring activities in PowerShell or other scripting engines, accessing aggregated threat data, and gaining visibility into user activities.
- The ability to control the state of the targeted system: halting arbitrary processes, remediating processes that are part of the attack, and isolating infected devices.

Successfully interrupting fileless attacks requires a holistic approach that can scale up and rapidly cascade appropriate actions where and when they are called for.

How to protect against specific types of fileless threats?

Our research teams include over 250 researchers around the world. With Trellix, you will get a team of experts who analyze suspicious objects and behaviors for malicious threats and develop tools that directly block different variants of fileless threats. We've released several signatures that block different variants of fileless threats. These include:

Fileless threat: Reflective self injection
Reflective loading refers to loading a portable executable (PE) from memory rather than from disk. A crafted function/script can reflectively load portable executable without getting registered as a loaded module in the process and hence can perform actions without leaving footprints. PowerShell is one of the most widely used applications to execute these crafted scripts. This event indicates a fileless attack where a PowerShell script tries to inject a PE into the PowerShell process itself.

Fileless threat: Reflective EXE self injection
Reflective loading refers to loading a PE from memory rather than from disk. A crafted function/script can reflectively load an executable (EXE) without getting registered as a loaded module in the process and hence can perform actions without leaving footprints. PowerShell is one of the most widely used applications to execute these crafted scripts. This event indicates a fileless attack where a PowerShell script tries to inject an EXE into the PowerShell process itself.

Fileless threat: Reflective DLL remote injection
Reflective loading refers to loading a PE from memory rather than from disk. A crafted function/script can reflectively load a DLL without getting registered as a loaded module in the process and hence can perform actions without

About
Why Trellix?
About Us
Explore Products
Leadership
Careers

News and Events
Newsroom
Press Releases
Blogs
Webinars
Events

Resources
Security Awareness
Resource Library
Training and Education
Communication Preferences
Trellix Store

Support
Support
Customer Success Plans
Downloads
Product Documentation

Trellix
Contact Us

Privacy | Legal | Terms of Service

More Ransomware Articles