# Fileless Malware

Jagir Shastri, B.Tech CSE (CS),

ICT Ganpat University, Ahmedabad

## Abstract

With the coming of anti-cybersecurity measures, the threat has changed, especially from malware to malware. Fileless malware does not use the usual utility to perform its functions. Therefore, it does not use the file system, thus avoiding the signature-based detection system. Fileless malware attacks are a disaster for any company because of their persistence, and the ability to evade any anti-virus solutions. A malware program uses the power of operating systems, trusted tools to accomplish its malicious purpose. To analyze such a malicious computer program, security experts use forensic tools to track the attacker, and the attacker may use anti-forensics tools to clear their tracks. This survey conducted a comprehensive analysis of the unprofessional computer-free program and their access methods found in the textbooks. Introducing a process model for handling file-free malware attacks in the response process. Finally, specific points in the proposed process model are identified, and the associated challenges are highlighted.

## Introduction

Throughout the history of malicious programs, there has been one thing that has not changed, the development of malware. Someone had to upgrade so that no anti-virus (AV) software could detect its presence in the system.

This malware is capable of staying in the main memory of the system undetected, making minor changes to the file system. This strategy has become a malware or Fileless program.

If a system is identified as being at risk of a malicious program or malware, the first thing a forensics expert will do is to detect malicious file running in background. However, in this case, it is not possible because the fileless malware does not reside in the file system, it is an operating in memory.

Attackers are heavily involved in testing the exploits for pre-installed software such as flash player, web browser, PDF viewer and Microsoft office to exploit and upload excutable to ram without touching the local file.

In Windows Operating Systems, two tools are also very powerful Powershell and wmi. The PowerShell, is a very flexible system shell and a writing platform for

the attacker to provide all the features in different stages of intervention. As such, it can also be used to bypass anti-virus detection, retention persistence or data leakage. Example: In 2016, a group of hackers sneaked into the DNC (Democratic National Committee) with fileless malware. In this case, PowerShell and WMI (Windows Management Instrumentation) were used as attack vectors

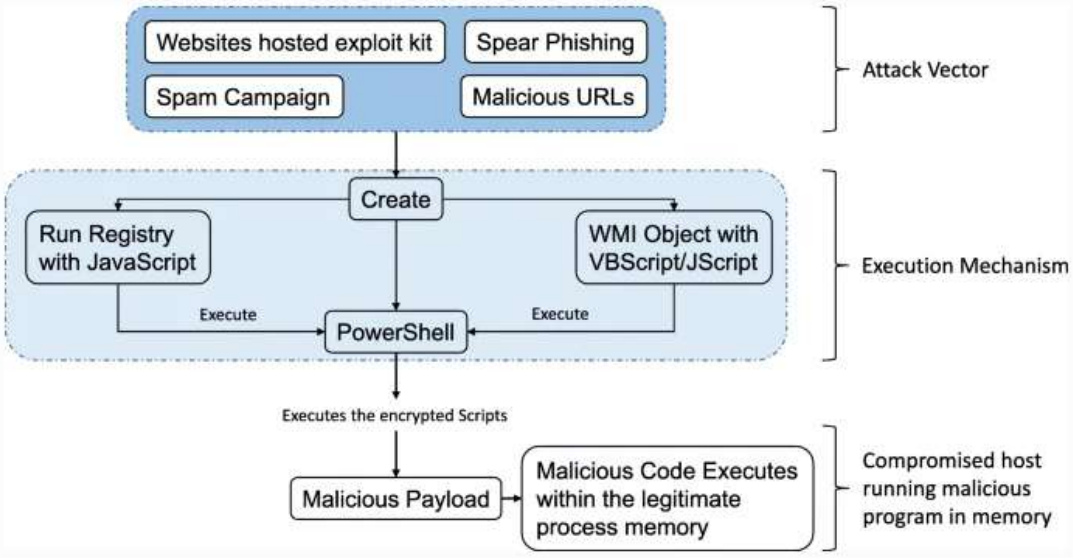
Attacker uses a tool like PowerShell to link attacks with the help of existing tools such as meterpreter, SET (Social Engineering Toolkit), or Metasploit Framework including a wide range of modules already built in and ready for use for the purpose. planning more attacks

Fileless malware attacks on organizations or individuals targeting the targeted system avoid downloading malicious and malicious files usually on disk; instead, it uses the power of web exploitation, macros, documents, or * **trusted management tools *(Show)**

There are no limits to what kind of attacks can be caused by fileless malware.

## Theory Of Fileless Malware

**Fig. 1**




Infection flow of fileless malware

A malicious malware attack does not download malicious files or run on any disk to compromise systems. Attacker simply exploits the vulnerable application to inject malicious code directly into the main memory. Attacker can also use trusted and widely used programs, i.e., Microsoft office or Windows OS management tools such as PowerShell and WMI to process scripts and upload malicious codes directly to dynamic memory.

Attackers have taken advantage of two powerful Windows operating systems Windows Management Instrumentation and PowerShell to use their vulnerability to make the attack invisible to AV solutions.

The life cycle of a fileless computer program works in three stages. First, attack the vector, which has the means by which the attacker targets their victims. Second, the approach to this initial malicious code may attempt to create persistence or WMI object via VBScript / JScript to use the PowerShell instance. Thirdly, PowerShell can continue to use malicious software on legitimate process memory directly without discarding any files in the file system making its target compromise its system/server/endpoint.

## Types of Fileless Malware

There are three primary categories of fileless malware attacks.

- **Windows registry manipulation**
- **Memory code injection**
- **Script-based techniques**

### Windows registry manipulation

Windows registry manipulation involves the use of a malicious file or link that, when clicked on, uses a normal Windows process to write and execute fileless code into the registry.

Examples of this include Kovter and Powelike, which can transform your infected system into a click bot by connecting with websites and click-through ads.

- **Persistence mechanism -** JavaScript code is added into the registry and is executed by a legitimate Windows file, mshta.exe, via WMI instead of mshtml.dll:

HKLM\path{\Software\Microsoft\Windows\Current Version\Run}.

### Memory code injection

Memory code injection techniques involve hiding malicious code in the memory of legitimate applications. While processes that are critical to Windows activity are running, this malware distributes and reinjects itself into these processes.

These Filessless malware use the permission given to a specific application to run its own malicious code. Fileless malware attacks use legitimate Windows

programs like PowerShell and MWI, so commands executed by these default programs are assumed to be legitimate and safe. they look like a program that's supposed to be running. This can be tricky for companies. You can't ban employees from using these programs as you could with other potentially malicious programs, because they're often integrated into daily operations.

**Script-based techniques**

Script-based techniques may not be completely fileless, but they can be hard to detect.

Two examples are SamSam ransomware and Operation Cobalt Kitty. Both are malware attacks that used techniques of common fileless malware attacks

## Possible ways to protect an organization against fileless malware, and what to look out for?

There is no full proof solution to protect an organization against fileless malware.

The only thing as a security researcher we can do is to build best teams for a SOC (security operation center) and monitor any unwanted/unwilling/unidentified process/service running or any external communication is going on by our endpoint.

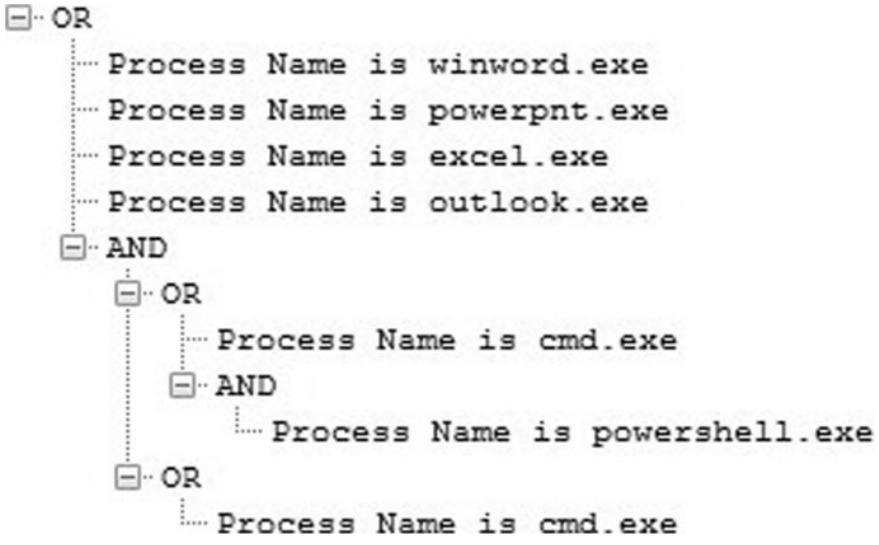
It is important to identify the main sources of information such as network traffic, network connectivity(i.e external connectivity), and suspicious modification of certain Windows registry keys. In addition, the Windows event log, alerts specific indicators that may suggest a malicious action. Some of the important events need to be adequately monitored, such as:

- Event ID 4688: The system needs to monitor all the newly created processes whose parent process is PowerShell.
- Event ID 10148: This event is responsible for listening to the specific IP and Port for WS-Management related requests.
- Event ID 7040: If the service has been changing to auto-start from disabled/demand start of the Windows Remote Management (WS-Management).
- Event ID 4103 and 4104: These both Event id's gives us a better view of Powershell related activity.
  The first(4103) gives us module logging and records the pipeline execution details of PowerShell such as command portion of script.

The other(4104) is a script block logging this records the entire content of the PoerShell script and blocks of code running in PowerShell.

## Detection by rule-based

In addition, the detection of such programs, which trigger cmd.exe or powershell.exe, may be malicious. Therefore, the detection method may work with a specific rules defined that can distinguish between a malicious process and a malicious process.




## Challenges In Detecting Fileless Malware

Fileless malware already presents a significant problem, and it is gaining further popularity among attackers because it is undetectable by traditional file-based prevention and detection systems. Since, there are no files written on the disk, when the malware is persisting exclusively through process memory and registry files. The malicious process is not accessible without doing in-memory analysis because the source code of the process is not available. The fileless attacks can evade these AV tools without triggering alarms

## Conclusion

Security defenders should focus more on detecting and preventing fileLess malware attacks. Attackers use a legitimate app to achieve their malicious motives. As such, PowerShell and WMI can also be used to bypass signature systems based on signature, maintaining persistence or filtering data makes it difficult to detect malicious activity. In this paper, we have distinguished most of the available diagnostic methods and types of malware, targeted in the real world.

The rapid increase of these malware attacks has increased this problem. The massive international outcry against SWIFT (Society for Worldwide Interbank Financial Telecommunication) served as a signal that critical infrastructure and global financial systems will continue to be driven by this kind of complex attack.

## References

[1] A Review on Fileless Malware Analysis Techniques. May 2020 International Journal of Engineering and Technical Research V9(05)

https://www.researchgate.net/publication/341870307_A_Review_on_Fileless_Malware_Analysis_Techniques

[2] Phase Bot - A Fileless Rootkit (Part 1) (2014). https://www.malwaretech.com/2014/12/phase-bot-fileless-rootki.html

[3] Gorelik, M., Moshailov, R.: Fileless Malware: Attack Trend Exposed (2017). http://blog.morphisec.com/fileless-malware-attack-trend-exposed Accessed 2018-05-02

[4] Living Off The Land Binaries And Scripts - (LOLBins and LOLScripts) (2019). https://github.com/LOLBAS-Project/LOLBAS

[5] Fileless Malware - A Behavioural Analysis Of Kovter Persistence (2016). https://airbus-cyber-security.com/fileless-malware-behavioural-analysis-kovter-persistence/

# Plagiarism Report