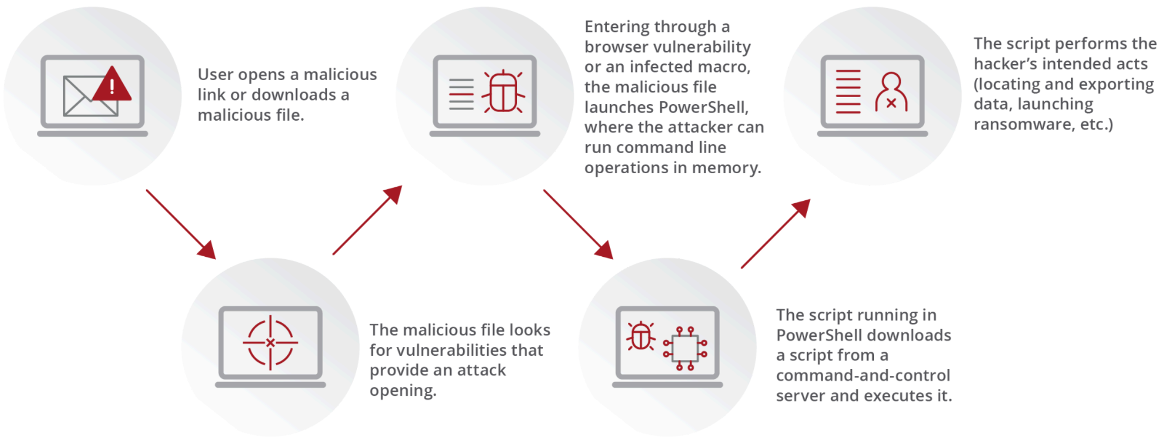
Fileless Malware and Hunting Techniques Using KQL Queries

1. Introduction

Fileless malware is a type of cyberattack that runs **directly in a computer's memory** instead of being stored on the **hard drive**. It **does not use traditional malicious files**, making it harder for antivirus software to detect. Instead, it **hijacks trusted system tools** like **PowerShell, WMI (Windows Management Instrumentation), or MSHTA (Microsoft HTML Application Host)** to execute malicious activities.

Since it **leaves no files**, it avoids detection by traditional security scans. However, security teams can still **track its behavior** by monitoring **suspicious activities**, like unusual command execution or unauthorized data access.This document provides an in-depth overview of fileless malware, common attack techniques, and effective hunting strategies using **Kusto Query Language (KQL)** in **Microsoft Sentinel** and **Microsoft Defender**.



FILELESS MALWARE ATTACK PROCESS

User opens a malicious link or downloads a malicious file.

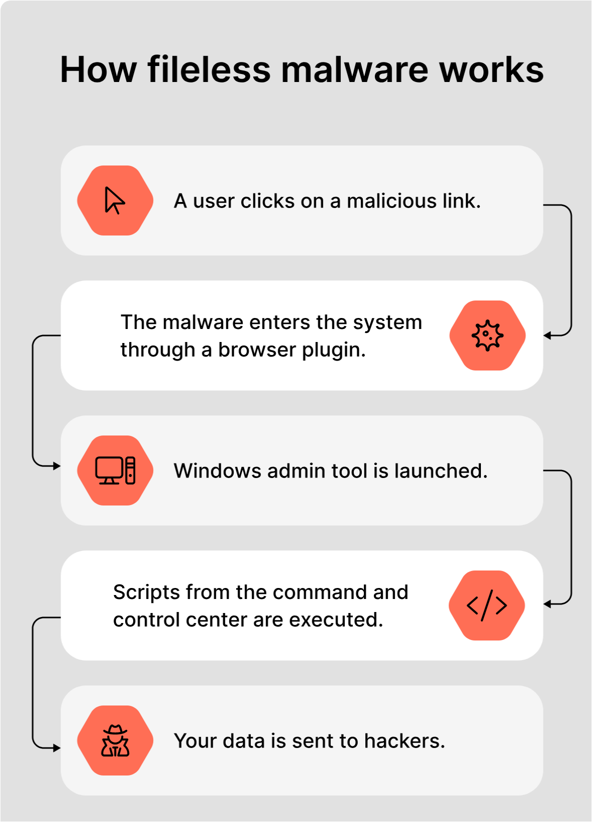
The malicious file looks for vulnerabilities that provide an attack opening.

Entering through a browser vulnerability or an infected macro, the malicious file launches PowerShell, where the attacker can run command line operations in memory.

The script running in PowerShell downloads a script from a command-and-control server and executes it.

The script performs the hacker's intended acts (locating and exporting data, launching ransomware, etc.)

2. Understanding Fileless Malware


How fileless malware works
A user clicks on a malicious link.
The malware enters the system through a browser plugin.
Windows admin tool is launched.
Scripts from the command and control center are executed.
Your data is sent to hackers.

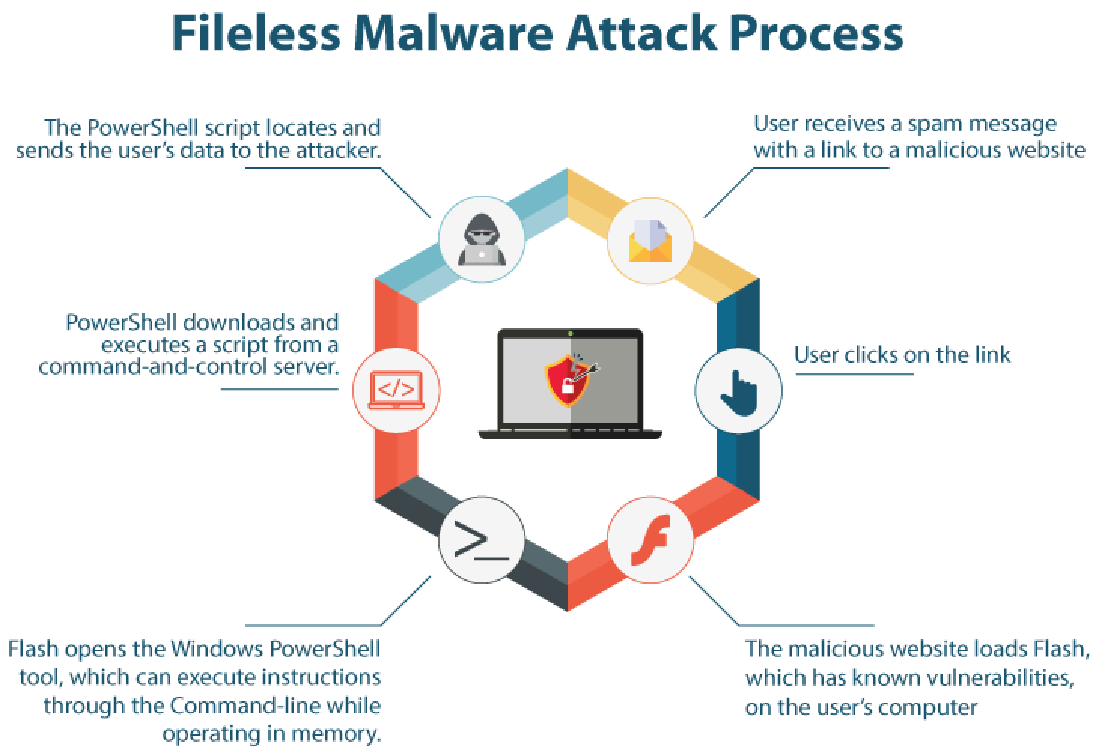
2.1 What is Fileless Malware?

Unlike traditional malware, which relies on executable files, fileless malware:

- Runs entirely in memory
- Uses built-in system tools (e.g., PowerShell, WMI, cmd.exe)
- Leaves minimal forensic evidence
- Evades traditional signature-based detection

2.2 Common Fileless Malware Techniques

- **Memory-Only Attacks:** Malware resides in RAM, leaving no files on disk.
- **Living off the Land (LotL) Attacks:** Utilizes legitimate system tools like PowerShell or WMI.
- **Registry-Based Attacks:** Stores malicious scripts in the Windows Registry for persistence.
- **Process Injection:** Injects malicious code into legitimate processes (e.g., svchost.exe).
- **Scheduled Task Persistence:** Creates scheduled tasks for recurring execution.


Fileless Malware Attack Process
User receives a spam message with a link to a malicious website
User clicks on the link
The malicious website loads Flash, which has known vulnerabilities, on the user's computer
Flash opens the Windows PowerShell tool, which can execute instructions through the Command-line while operating in memory.
PowerShell downloads and executes a script from a command-and-control server.
The PowerShell script locates and sends the user's data to the attacker.

3. Fileless Malware Hunting Using KQL Queries

KQL queries help security analysts detect and investigate fileless malware activities in **Microsoft Sentinel** and **Defender** logs.

3.1 Detecting Suspicious PowerShell Execution

PowerShell is commonly exploited for fileless malware attacks. This query identifies suspicious commands used for execution.

Explanation: This query identifies PowerShell scripts executing suspicious commands like `Invoke-Expression`, which attackers commonly use to run malicious code.

```
DeviceProcessEvents

| where InitiatingProcessFileName == "powershell.exe"

| where ProcessCommandLine contains "Invoke-Expression" or

      ProcessCommandLine contains "IEX" or

      ProcessCommandLine contains "downloadstring"
```

```
| project Timestamp, DeviceName, FileName, ProcessCommandLine, InitiatingProcessFileName

| order by Timestamp desc
```

3.2 Detecting WMI-Based Attacks

WMI (`wmiprvse.exe`) is often used for remote execution and persistence.

Explanation: Attackers use `wmiprvse.exe` to execute remote commands and maintain persistence.

```
DeviceProcessEvents

| where InitiatingProcessFileName == "wmiprvse.exe"

| where ProcessCommandLine contains "execute"

| project Timestamp, DeviceName, FileName, ProcessCommandLine
```

3.3 Hunting Memory-Only Malware via LOLBins

Attackers use legitimate system tools (`rundll32.exe`, `mshta.exe`, `regsvr32.exe`) for execution.

Explanation: This query detects processes executing scripts (`.js`, `.vbs`) or making HTTP requests, which may indicate malware.

```
DeviceProcessEvents

| where InitiatingProcessFileName in ("rundll32.exe", "mshta.exe", "regsvr32.exe")

| where ProcessCommandLine contains "http" or ProcessCommandLine contains ".js" or ProcessCommandLine contains ".vbs"

| project Timestamp, DeviceName, FileName, ProcessCommandLine
```

4. Ransomware-Based Fileless Attacks

4.1 Fileless Ransomware Execution via PowerShell

Some ransomware strains leverage PowerShell to encrypt files directly in memory.

```
DeviceProcessEvents
```

```
| where InitiatingProcessFileName == "powershell.exe"

| where ProcessCommandLine contains "-Enc" or ProcessCommandLine contains "AES" or ProcessCommandLine contains "Encrypt"

| project Timestamp, DeviceName, FileName, ProcessCommandLine
```

Real-World Example: Some ransomware strains like **Powersniff** and **FIN7** have used PowerShell for in-memory encryption.

5. Credential Dumping Techniques and Detection

5.1 Detecting Mimikatz-Like Activity (LSASS Dumping)

```
DeviceProcessEvents

| where FileName in ("rundll32.exe", "powershell.exe", "cmd.exe")

| where ProcessCommandLine contains "sekurlsa::logonpasswords" or ProcessCommandLine contains "lsass"

| project Timestamp, DeviceName, FileName, ProcessCommandLine
```

5.2 Monitoring Suspicious LSASS Access

```
DeviceProcessEvents

| where FileName == "taskmgr.exe" or FileName == "procdump.exe"

| where ProcessCommandLine contains "lsass"

| project Timestamp, DeviceName, FileName, ProcessCommandLine
```

6. Best Practices for Detecting and Preventing Fileless Malware

To effectively defend against fileless malware, organizations should implement the following best practices:

6.1 Logging and Monitoring

- **Enable Script Block Logging** for PowerShell.
- **Monitor process execution logs** in Microsoft Sentinel and Defender.
- **Track parent-child process relationships** to identify unusual command executions.

6.2 Restricting Execution

- **Limit PowerShell and WMI access** to administrative users only.
- **Disable unused Windows scripting components** (e.g., `mshta.exe`, `wscript.exe`).
- **Use Attack Surface Reduction (ASR) rules** to block script-based attacks.

6.3 Threat Intelligence and Automated Alerts

- **Ingest threat intelligence feeds** to detect known indicators of compromise (IoCs).
- **Set up automated alerts** for KQL queries detecting fileless malware activities.
- **Regularly scan for unauthorized registry modifications and scheduled tasks.**

6.4 Automated Response with Sentinel Playbooks

- **Use Microsoft Sentinel Playbooks** to trigger automatic responses when malicious activity is detected.
- **Integrate with SOAR tools** to enable automated threat containment and remediation.

7. Conclusion

Fileless malware is a sophisticated and evasive threat that leverages built-in system tools to execute malicious actions without traditional files. Security teams must leverage proactive threat-hunting techniques using **KQL queries** in **Microsoft Sentinel** and **Microsoft Defender** to detect and mitigate these attacks effectively. Implementing best practices such as **script monitoring, process analysis, and threat intelligence integration** will enhance an organization's defense against fileless malware threats.