F

Fileless Attacks: Techniques, Detection, and Response

A fileless attack relies mainly on memory, trusted tools, or data stores such as the registry instead of a conventional malicious executable on disk.

PUBLISHER
Gridinsoft Support

UPDATED
Oct 8, 2026

READ TIME
3 min

CONTENTS

In this guide

01 What makes an attack fileless?
02 Common fileless techniques
03 Why file scanning alone is insufficient
04 Detection priorities
05 What to do after detection
06 Prevention
07 Source

Quick answer: A fileless attack executes malicious activity mainly through memory, trusted system tools, scripts, or data stores such as the registry instead of relying on a conventional malware executable saved to disk. “Fileless” does not mean no files are ever involved, no evidence exists, or the attack cannot be detected.

01 What makes an attack fileless?

Fileless is a description of how code is delivered, executed, or persisted, not one malware family. A phishing document may be a real file but launch PowerShell that loads code directly into memory. Another threat may store encoded instructions in the Windows registry or WMI repository and use a legitimate component to execute them later. The useful question is where each stage resides and which behaviors it performs.

Microsoft distinguishes several models: memory-only payloads, fileless persistence in repositories such as WMI, and abuse of legitimate tools. Some attacks are only partially fileless because an initial document, shortcut, script, or loader still touches disk.

02 Common fileless techniques

- **Script interpreters:** PowerShell, Windows Script Host, shell, or other interpreters download, decode, or execute commands.
- **Memory injection:** code runs inside another process or loads reflectively without a normal installed executable.
- **Registry or WMI persistence:** instructions are stored in multipurpose system databases and triggered at startup or by an event.
- **Living off the land:** trusted administration utilities perform discovery, credential access, transfer, or execution.
- **Document active content:** macros or embedded scripts start a chain when the user opens a file and enables content.

03 Why file scanning alone is insufficient

A signature scanner may never see a stable malicious file, and an attacker can encode commands or use signed tools already present. However, the operating system still records process creation, command lines, script scanning, registry changes, authentication, memory activity, DNS requests, and network connections when appropriate telemetry is enabled.

Microsoft notes that technologies such as the Antimalware Scan Interface, behavior monitoring, memory scanning, and boot protection can inspect fileless threats. Central collection is important because an attacker with endpoint control may clear local evidence.

04 Detection priorities

1. Capture process ancestry and full command lines for interpreters, office applications, browsers, remote tools, and system utilities.
2. Monitor PowerShell and script content according to platform guidance, with protected centralized logs and appropriate privacy controls.
3. Alert on unusual WMI subscriptions, registry autoruns, scheduled tasks, services, and file-association handlers.
4. Correlate credential access, privilege escalation, lateral movement, and rare outbound destinations instead of alerting on a tool name alone.
5. Use memory acquisition and forensic analysis when volatile payloads or injected code are suspected.

PowerShell, WMI, and administrative tools are legitimate. Blocking them globally can break operations; allow approved use, constrain privilege, and detect abnormal sequences and users.

05 What to do after detection

Isolate the endpoint while preserving volatile evidence if the response plan requires it. Record the alert, process tree, scripts, registry or WMI persistence, identity activity, network connections, and delivery artifact. Hunt other endpoints for the same chain. Revoke exposed sessions and credentials from a clean device, remove the initial access path, and rebuild high-value systems when integrity cannot be established.

A reboot may remove memory-only code but can also destroy evidence; persistence may simply reload it. Coordinate reboot timing with the incident-response team.

06 Prevention

Patch supported systems and applications, block internet-originated macros, use least privilege, restrict script interpreters and remote administration to approved users, and apply application control. Enable endpoint behavioral protection, protected logging, network segmentation, phishing-resistant MFA, and tested backups. Measure whether the team can investigate script and memory activity before an incident.

07 Source

Definitions, attack models, and detection technologies follow Microsoft’s fileless-threat guidance.

NEED MORE HELP?

We are here when the guide is not enough.

Visit the Gridinsoft Support Center for current product guidance or submit a support request with the details of your case.

Open this guide online at support.gridinsoft.com