Windows Forensics: Text Search Tools (grep & awk Equivalents)

When you're hunting through Windows logs, memory dumps or evidence files, you need quick ways to find strings or patterns. Two built-in tools stand out:

1. **findstr** — the classic CMD utility
2. **Select-String** — PowerShell's powerful, object-oriented matcher

---

1. findstr (CMD)

- **Purpose**: Search files for text or simple regex patterns.
- **Syntax**:
- findstr [options] "pattern" [file(s)]
- Common Flags:

/S Recursively search subdirectories

/I Case-insensitive

/M Print only filenames with matches

/R Treat pattern as a regular expression

Example: Search all .log files under C:\Evidence for "ErrorCode123" (ignore case, list only filenames):

batch Copy findstr /S /I /M "ErrorCode123" C:\Evidence*.log 2. Select-String (PowerShell) Purpose: Advanced pattern matching with full regex, context, and object output.

Syntax:

powershell Copy Select-String -Path -Pattern [options] Key Parameters:

-Path Files, wildcard support (C:\Dir*.txt)

-Pattern Regex or literal string

-CaseSensitive:$false (or -CaseSensitive)

-List Show only first match per file

-Context Number of lines of context (-Context 2,1)

Example: Recursively search .log files for “UserLoginFailed” and display filename, line number, and matched text:

powershell Copy Get-ChildItem -Recurse -Path C:\Evidence -Include *.log | Select-String -Pattern 'UserLoginFailed' | Format-Table Filename, LineNumber, Line When to Use Which? Use findstr for quick, script-friendly searches in CMD environments.

Use Select-String when you need full regex, context lines, or want to further process results in PowerShell.

ProTip: Pipe Select-String output into other cmdlets (e.g. Export-Csv, Where-Object) for richer forensic workflows.

Copy

| **Atom Bombing** | Writes malicious code into NTFS “Alternate Data Streams” (ADS) then uses AtomTable to execute from memory. | | **DLL Search Order Hijacking** | Place a rogue DLL in the same folder as a legitimate EXE to trick Windows into loading the malicious DLL. | | **Process Hollowing** | Spawn a benign process in suspended state, unmap its memory, write in shellcode, then resume. | | **Thread Execution Hijacking** | Inject a new thread into a running process using CreateRemoteThread() and point it at malicious shellcode. | | **Drydex Malware Case Study** | Drydex uses a combination:

1. Drops a loader via phishing.
2. Uses process hollowing to inject its payload into svchost.exe.
3. Employs thread hijacking to maintain persistence in memory. |

Rootkit Paradox:
A rootkit hides by intercepting OS calls—but must run code in kernel or userland, which leaves detectable hooks.
The more it hides, the more unpredictable behavior flags it.

🛠️ Microsoft’s Memory Forensics Contributions

- **AVML (Acquire Volatile Memory for Linux):**
  A userland, static-binary tool that captures live memory on Linux

via /proc/kcore or /dev/mem without on-target build infrastructure :contentReference[oaicite:0]{index=0}.

- **Project Freta:**
  A cloud-based service for automated Linux memory analysis—identifies injected code, hidden processes, and gives a jump-start on forensic triage :contentReference[oaicite:1]{index=1}.

🔍 **Key Memory Artifacts to Recover**

- **Running Processes:** Use Volatility's pslist/psscan plugins to enumerate active and terminated processes.
- **Network Connections:** netscan plugin reveals in-memory TCP/UDP sockets not logged elsewhere.
- **Loaded Drivers:** modscan uncovers kernel modules that may not appear in driverquery.
- **Carved Metadata Structures:** Use yarascan or malfind for dumped PE headers and injected code sections.
- **Packet Buffers:** connscan and sockscan can reconstruct in-flight packets from memory.

** 📚 Further Reading & Tools**

- **Volatility 3 Documentation:** https://volatility3.org/
- **AVML on GitHub:** https://github.com/microsoft/avml :contentReference[oaicite:2]{index=2}
- **Project Freta:** https://learn.microsoft.com/security/research/project-freta/how-to-capture-an-image :contentReference[oaicite:3]{index=3}
- **Atomic Canary (Atom Bombing explanation):** https://medium.com/@dustinrue/atomic-cmd-attack-…
- **Process Hollowing Demo:** https://attack.mitre.org/techniques/T1093/ | Atom Bombing |

Writes malicious code into NTFS “Alternate Data Streams” (ADS) then uses AtomTable to execute from memory. | | DLL Search Order Hijacking | Place a rogue DLL in the same folder as a legitimate EXE to trick Windows into loading the malicious DLL. | | Process Hollowing | Spawn a benign process in suspended state, unmap its memory, write in shellcode, then resume. | | Thread Execution Hijacking | Inject a new thread into a running process using CreateRemoteThread() and point it at malicious shellcode. | | Drydex Malware Case Study | Drydex uses a combination: Drops a loader via phishing. Uses process hollowing to inject its payload into svchost.exe. Employs thread hijacking to maintain persistence in memory. |

Rootkit Paradox:
A rootkit hides by intercepting OS calls—but must run code in kernel or userland, which leaves detectable hooks.
The more it hides, the more unpredictable behavior flags it.

🛠️ Microsoft’s Memory Forensics Contributions AVML (Acquire Volatile Memory for Linux):
A userland, static-binary tool that captures live memory on Linux via /proc/kcore or /dev/mem without on-target build infrastructure :contentReference[oaicite:0]{index=0}. Project Freta:
A cloud-based service for automated Linux memory analysis—identifies injected code, hidden processes, and gives a jump-start on forensic triage :contentReference[oaicite:1]{index=1}. 🔍 Key Memory Artifacts to Recover Running Processes: Use Volatility’s pslist/psscan plugins to enumerate active and terminated processes. Network Connections: netscan plugin reveals in-memory TCP/UDP sockets not logged elsewhere. Loaded Drivers: modscan uncovers kernel modules that may not appear in driverquery. Carved Metadata Structures: Use yarascan or malfind for dumped PE headers and injected code sections. Packet Buffers: connscan and sockscan can reconstruct in-flight packets from memory. 📚 Further Reading & Tools Volatility 3 Documentation: https://volatility3.org/ AVML on GitHub: https://github.com/microsoft/avml :contentReference[oaicite:2]{index=2} Project Freta: https://learn.microsoft.com/security/research/project-freta/how-to-capture-an-image :contentReference[oaicite:3]{index=3} Atomic Canary (Atom Bombing explanation): https://medium.com/@dustinrue/atomic-cmd-attack-… Process Hollowing Demo: https://attack.mitre.org/techniques/T1093/ Memory forensics turns RAM into your most honest witness. Combine it with disk artifacts to reconstruct the full story of an attack.

Memory forensics turns RAM into your most honest witness. Combine it with disk artifacts to reconstruct the full story of an attack.