Ransomware Detection & Response Catalogue

Version 1.0

Date 23.09.2025

Purpose	This document provides a structured catalogue of detection signals and investigation guidelines. It is designed to support the Security Operations Center (SOC) and Incident Response (IR) teams during alert triage, investigation, and remediation.
Scope	Covers detections related to ransomware, credential abuse, privilege escalation, and exfiltration. Intended for SOC analysts, incident responders, and security engineers.
How to Use This Catalogue	Use the Table of Contents to quickly navigate to the relevant detection signal. Each entry provides the signal name and category, the query summary and detection logic, indicators of compromise (IoCs), context and response guidance
Disclaimer	This catalogue is a living document. Queries, thresholds, and response steps may evolve as the threat landscape changes. Always validate detections with contextual evidence before taking remediation actions. Use in conjunction with official SOC procedures, escalation matrix, and runbooks.

Contents

[Mercuria] - Ransomware Stage 1 - Suspicious Execution

This analytics rule identifies suspicious execution behaviors that are commonly associated with the early stage of ransomware or malware activity.

Goal: The launch of a first malicious code.

Aggregation: Events are grouped into 15-minute time windows per device.

1) SUSPICIOUS POWERSHELL EXECUTION

Why / how attackers use it:

Attackers use PowerShell to download and execute code directly in memory, leaving no (or minimal) artifacts on disk. Commands like Invoke-WebRequest, DownloadString, and IEX allow them to retrieve a remote script and execute it immediately, useful for quickly deploying a payload, installing a loader, or launching an exfiltration chain.

What the KQL detects:

- powershell.exe processes where the command line contains "DownloadString", "Invoke-WebRequest", "IEX", or "Invoke-Expression". Use to pull and execute remote stagers.
- Requires at least 2 events in 15 minutes

IOCs:

- Process: powershell.exe, pwsh.exe
- Command-line tokens: Invoke-WebRequest, DownloadString, IEX, Invoke-Expression, -EncodedCommand, -NoProfile, -WindowStyle Hidden, Start-Process -Verb RunAs
- Network: HTTP(S) URLs in command lines
- Paths: user-writable paths in commands (\AppData\, \Temp\, \Downloads\)
- Parent process not signed and unusual parent
- Recurrent short bursts of such commands from same account/device

Remediation if malicious:

- Isolate the affected endpoint from the network (quarantine VLAN or remove network access).
- Collect forensic artifacts: full process command lines, PowerShell transcription/logs, Sysmon/EVENT logs, and memory capture if possible.

- Run a full EDR/AV scan and remove confirmed malicious artifacts.
- Block the observed URLs/domains and associated file hashes at the perimeter and on EDR.
- If malware confirmed, restore from a known-good backup and review containment timeline.

FP exclusions :

- Excludes benign sources that create FP: “chocolatey.org” and “github.com/chrisant996”
- Excludes if the command contains “tvcpkg-init.cmd”

Likelihood of Malicious intent: Medium 🟠

MITRE mapping: Command and Scripting Interpreter — PowerShell (T1059.001).

2) Suspicious rundll32.exe Execution

Why / how attackers use it:

rundll32.exe is a legitimate Windows tool that can execute DLL functions. Attackers abuse it to load malicious DLLs (often stored in user-writable folders) or to run crafted command strings that hide intent. Because rundll32.exe is signed and normal, its misuse blends in.

What the KQL detects:

- rundll32.exe where the command line is very short (<20) or doesn't contain .dll.
- Or where the command includes sensitive/user-writable paths or the initiating process is unsigned/invalid.

IOCs:

- Process: rundll32.exe executing with arguments that don’t include a clear .dll path or have extremely short/obfuscated args
- Command lines referencing user-writable paths (AppData, Temp, Downloads, ...)
- Unusual parent process (unsigned or non-standard launcher)
- Unexpected child behavior (network connections, process creation) following rundll32 launch

Remediation if malicious:

- Quarantine the device and capture full process and command-line telemetry.
- Identify and collect the DLL or payload referenced by the command, compute hashes and check TI.
- Block execution of the identified DLL/hash; consider AppLocker/Code Integrity rules for prevention.
- Reimage the endpoint if persistent or unknown backdoors are found.

FP exclusions: Excludes benign sources that create FP: MSI/custom action patterns.

Likelihood of Malicious intent: Low 🟡

MITRE mapping: Signed Binary Proxy Execution / Execution via DLL (T1218 family).

3) Executable Dropped in Sensitive Folders + TI match

Why / how attackers use it:

An executable that appears in sensitive OS folders (Windows\Temp, Tasks, System32\Tasks, ProgramData, etc.) and matches a known-bad hash from threat intelligence is a strong sign of malware being staged to run automatically or persist. Attackers place executables in these locations because they often have elevated access or are executed by scheduled tasks/services.

What the KQL detects: Catch all the files created in a sensitive path where the file SHA256 matches an active TI Indicators file hash.

IOCs:

- Created file in sensitive system path
- File SHA256 (or other hash) that matches threat-intel feed entries with high confidence.
- Initiating process that created the file (cmd, msiexec, installer, or unknown)

Remediation if malicious

- Immediately isolate the host and collect the file.
- Identify and stop any scheduled tasks, services, or autoruns pointing to the dropped executable. Disable them and take screenshots/logs.
- Remove the malicious file and related persistence mechanisms.
- Scan the environment for the same hash or related indicators and block at endpoint/AV and network perimeters.
- If compromise is confirmed, perform credential resets for impacted accounts and perform deeper lateral movement investigation. Reimage if necessary.

Sensitive path list: C:\Windows\Temp, C:\Windows\SystemTemp, C:\PerfLogs, \Windows\Tasks\, \Windows\System32\Tasks\, \Windows\System32\spool\drivers\, \Windows\SysWOW64\spool\drivers\, \Users\Public\, \Users\Default\, \AppData\Local\Temp\, \AppData\LocalLow\, \AppData\Roaming\, \OneDrive\, \Downloads\, \AppData\Local\Microsoft\Windows\INetCache\, \AppData\Local\Microsoft\Windows\Temporary Internet Files\, \$Recycle.Bin\

Likelihood of Malicious intent: Medium

MITRE mapping: Impact / Execution / Ingress Tool Transfer (T1105) and Malware installation techniques.

4) MSHTA remote content execution

Why / how attackers use it:

mshta.exe can execute HTML Application (HTA) content directly. Attackers host malicious HTA pages and run them via mshta to execute script and drop payloads, often used in phishing campaigns because clicking a single URL or opening an attachment can trigger the flow.

What the KQL detects: mshta.exe with command line that contains http/https.

IOCs:

- Process: mshta.exe invoked with an http:// or https:// URL as the argument
- Immediate subsequent network connections to suspicious domains or C2 hosts
- Child processes launched from mshta (powershell, wscript, cmd, rundll32)
- Small/obfuscated HTA content referenced in URLs

Remediation if malicious

- Quarantine endpoint and capture process tree and network flows initiated by mshta.
- Block the hosting domain(s) and any associated IPs at the proxy/firewall.
- Retrieve the HTA content (if still accessible) for analysis and add indicators to blocklists.
- Remove payloads and related persistence.

Likelihood of Malicious intent: High 🔴

MITRE mapping: Signed Binary Proxy Execution (mshta) / Remote File Execution (T1218 family + T1105).

5) REGSVR32 REMOTE SCRIPTLET (SQUIBLYDOO)

Why / how attackers use it:

regsvr32.exe can load scriptlets or remote code via COM. Attackers use it (the "Squiblydoo" technique) to download and execute scripts over HTTP while appearing as legitimate Windows activity. This allows code execution without writing an EXE to disk and can bypass some application allow-lists.

What the KQL detects: regsvr32.exe where command line has any ("scrobj.dll", "/i:", "/i ") and the command line contains an http/https URL.

IOCs:

- Process: regsvr32.exe launched with /i: or with scrobj.dll and an HTTP/HTTPS URL
- Network calls to URLs from the regsvr32 command line
- Unusual parent process (unsigned or non-standard launcher)
- Subsequent script interpreter activity (wscript/cscript/powershell)

Remediation if malicious

- Quarantine host.
- Add the observed URL and any returned payload hashes to blocklists.
- Remove launched scriptlets and any persistence they created.

Likelihood of Malicious intent: High 🔴

MITRE mapping: Signed Binary Proxy Execution (regsvr32) / Remote Script Execution (T1218 family).

6) RUNDLL32 HTML/JS EXECUTION ABUSE

Why / how attackers use it:

Attackers can abuse rundll32.exe together with mshtml.dll (the Windows HTML engine) to execute HTML/JS payloads in memory. This technique allows malicious scripts to run without saving script files to disk, making detection harder.

What the KQL detects: rundll32.exe when the command line contains ("mshtml.dll,RunHTMLApplication", "javascript:")

IOCs:

- rundll32.exe command lines containing mshtml.dll,RunHTMLApplication or javascript: tokens
- Rapid child process creation or network traffic after such invocations
- Execution originating from user-writable folders or from browser contexts not normally associated with rundll32

Remediation if malicious:

- Quarantine and collect process chains; analyze the invoked HTML/JS payload if retrievable.
- Block the origin domain(s) and any payload hashes.
- Reimage if unknown persistent implant is discovered.

Likelihood of Malicious intent: High 🔴

MITRE mapping: Signed Binary Proxy Execution (rundll32) / Code Injection / Execution via HTML application (T1218 family).

7) RUNDLL32 INF INSTALL SUSPICIOUS

Why / how attackers use it:

rundll32.exe can call advpack.dll or setupapi.dll to run INF installation sections. Attackers sometimes craft INF files (driver or installer scripts) to install malicious drivers or persistence routines via this path, avoiding obvious executable drops.

What the KQL detects: Where the file rundll32.exe is used with and the command line has any ("advpack.dll,LaunchINFSection","setupapi.dll, InstallHinfSection") calling advpack/setupapi INF install routines

IOCs:

- rundll32.exe command lines with advpack.dll,LaunchINFSection or setupapi.dll,InstallHinfSection
- INF file paths referenced from poor/non-standard locations (user profile or temporary folders)
- New drivers or services installed shortly after the INF execution
- Unexpected parent process (unsigned installers, downloaded setup stubs)

Remediation if malicious:

- Isolate the host and gather INF file(s) and the invocation command-line for analysis.
- Remove any installed drivers/services that are malicious; collect driver files for TI/hashing.

- ❖ block hosting domain and remove the installer.
- ❖ If many hosts show similar INF installs, perform broader containment and credential review.

FP exclusions: Exclude several benign exclusions for known good installers (msiexec valid-signed, NVIDIA setup.exe cases, netskope EPDLP, known INF names, android emulator driver INF)

Likelihood of Malicious intent: Low 🟡

MITRE mapping: Execution via Installer/Installation (T1543/T1218 depending on technique).

8) CERTUTIL/BITS SUSPICIOUS STAGING

Why / how attackers use it:

certutil.exe and bitsadmin.exe are native tools used to fetch and decode content. Attackers use them to download encoder/decoder payloads, to fetch stagers, or to exfiltrate data (BITS can upload). Because they're signed system utilities, their usage can be missed unless the command line is monitored.

What the KQL detects: certutil.exe or bitsadmin.exe files where the command line contains ("-urlcache", "-decode", "/transfer", "/addfile")

IOCs:

- ❖ certutil.exe commands with -urlcache, -split, -decode and URLs in arguments
- ❖ bitsadmin.exe commands with /transfer, /addfile, /upload, /download and external URLs or UNC paths
- ❖ Use of base64 decoding or temporary decoded files created around the same time
- ❖ Parent process not typical for administrative downloads (e.g., explorer.exe launching certutil)

Remediation if malicious:

- ❖ Quarantine host and collect certutil / bitsadmin command lines, related files, and network logs.
- ❖ Block the domains and IPs used for staging or exfil at the proxy/firewall.
- ❖ Remove or quarantine downloaded payloads.
- ❖ Rotate credentials if data exfiltration is suspected.

FP exclusions: Excludes certutil -decode that does not involve http(s) (which are benign local decode operations.

Likelihood of Malicious intent: Medium 🟠 or High 🔴 (if external URL OR binary target OR unsigned parent)

MITRE mapping: Ingress Tool Transfer (T1105), Signed Binary Proxy Execution for LOLBIN use.

[Mercuria] - Ransomware Stage 2 - Persistence

This analytics rule detects suspicious persistence techniques commonly leveraged by ransomware and advanced malware families to maintain execution after reboot or logon. .

Goal: Maintain access after reboot or logout.

Aggregation: Events are grouped into 15-minute time windows per device.

1) Scheduled Tasks – Suspicious Creation

Why / how attackers use it:

Adversaries abuse scheduled tasks to achieve automatic execution of malicious code at logon, reboot, or on timed intervals. This persistence mechanism is reliable, stealthy, and often blends with legitimate IT operations. Tasks can:

- Launch payloads directly from disk or user-writable folders
- Repeatedly pull second-stage malware from remote servers (HTTP, UNC, FTP)

- ❖ Run obfuscated scripts or LoLBins to evade detections

This ensures the malware survives reboots and re-establishes control without requiring user interaction.

What the KQL detects:

- ❖ Where the file is schtasks.exe and the command line has /create.
- ❖ Flags tasks that reference LOLBins (PowerShell, wscript, cscript, mshta, rundll32, regsvr32, bitsadmin, certutil, curl, ftp.exe)
  - OR encoded flags (-enc, -encodedcommand)
  - OR URLs (http(s), UNC)
  - OR script extensions (.ps1 .bat .cmd .vbs .js)
  - OR user-writable target paths (\appdata\, \users\public\, \programdata\, \temp\) OR privileged run modes (/ru system, /rl highest)
  - OR sensitive triggers (onlogon, onstart, onidle) and short intervals (≤5 minutes).

IOCs:

- ❖ schtasks /create with: powershell, wscript/cscript, mshta, rundll32, regsvr32, bitsadmin, certutil, curl, ftp.exe, -enc, -w hidden, URLs, UNC paths
- ❖ /ru SYSTEM, /rl HIGHEST, /sc onlogon|onstart|onidle, minute intervals ≤ 5 (/sc minute /mo 1..5)
- ❖ /tr pointing to user-writable paths (\AppData\, \Users\Public\, \ProgramData\, \Temp\) or scripts (.ps1 .bat .cmd .vbs .js)

Remediation if malicious:

- ❖ Isolate the host, export the task XML and full command.
- ❖ Disable the task, collect referenced file(s)/URL(s) and compute hashes.
- ❖ Remove payloads and persistence, block hashes/domains.
- ❖ Hunt for the same task name or command across the estate; reset creds if compromise suspected.

FP exclusions: Exclude if the command refers to TeamViewer Rollback task, Office Subscription Heartbeat, HP Firmware Installer, AMD Link update XMLs, PBCCRCPassGuardXInput.

Likelihood of Malicious intent: Medium 🟠

MITRE mapping: T1053.005 (Scheduled Task).

2) System Restore Disabled

Why / how attackers use it:

Disabling System Restore removes the built-in recovery capability of Windows, making it harder for defenders to roll back encrypted or tampered systems. Ransomware families frequently disable restore points early in their kill chain to maximize impact and ensure victims cannot easily recover without paying the ransom.

What the KQL detects: schtasks.exe or regedit.exe with command lines has all ("SystemRestore", "disable", "Change")

IOCs:

- schtasks.exe or regedit.exe commands changing SystemRestore state (disable/change)
- Admin/SYSTEM context doing restore policy changes close to other suspicious activity

Remediation if malicious:

- Isolate host, revert System Restore settings and re-enable restore points.
- Block the parent command pattern, scan for additional policy tampering.

Likelihood of Malicious intent: High 🔴

MITRE mapping: T1562 (Impair Defenses).

3) Service Created via sc.exe

Why / how attackers use it:

Creating a custom Windows service provides persistent and privileged execution. Once installed, the service launches automatically at boot with SYSTEM privileges, giving attackers both durability and elevated access. This method is often combined with dropping payloads in user-writable or network paths to bypass application whitelisting.

What the KQL detects: sc.exe where command line has all ("create", ".exe")

IOCs:

- sc create <name> binPath= "<...>\something.exe"
- binPath in user-writable folders (\AppData\, \ProgramData\, \Temp\, network/HTTP)
- Unsigned parent/installer creating the service or new service set to auto start

FP exclusions: Excludes specific HP analytics services when launched by tainstaller.

Likelihood of Malicious intent: 🟠 Medium or 🔴 High (High if unsigned parent or user-writable/remote path detected)

MITRE mapping: T1543.003 (Create or Modify System Process: Windows Service).

4) PowerShell Service Persistence (New/Set-Service)

Why / how attackers use it:

Attackers leverage PowerShell to script the creation or modification of Windows services. This technique is attractive because:

- It blends into legitimate administrative activity
- It can be replicated across many endpoints quickly
- It supports obfuscation via -EncodedCommand or IEX

This makes it a flexible persistence mechanism during ransomware staging.

What the KQL detects: powershell.exe where command line has any ("New-Service", "Set-Service")

IOCs:

- PowerShell with New-Service or Set-Service pointing to suspicious paths or scripts
- Services set to Automatic/DelayedAutoStart right after a suspicious file drop

Remediation if malicious:

- Quarantine and list new/changed services and binaries.
- Remove or revert malicious service entries; delete payloads after collection.

FP exclusions: None

Likelihood of Malicious intent: 🟠 Medium or 🔴 High (high if -EncodedCommand or -enc is present OR matching one of the typical obfuscation/execution functions (frombase64string, Invoke-Expression, IEX))

MITRE mapping: T1543.003, T1059.001 (PowerShell).

5) WMI Permanent Subscription (WMIC)

Why / how attackers use it:

WMI permanent event subscriptions allow stealthy, event-driven persistence. Instead of launching malware on startup, the attacker configures WMI to trigger their payload when specific system events occur (e.g., user logon, process start). This technique is hard to detect because it leaves no visible startup item, and execution is proxied via wmiprvse.exe.

What the KQL detects: wmic.exe where command line has any ("/namespace:\\\\root\\subscription", "/CREATE")

IOCs:

- wmic /namespace:\\root\subscription /CREATE ...
- New __EventFilter, CommandLineEventConsumer, and __FilterToConsumerBinding objects on disk/registry
- Follow-on execution by wmiprvse.exe

Remediation if malicious

- Isolate, enumerate and export WMI subscriptions in root\subscription.
- Remove malicious filters/consumers/bindings and collect referenced payloads.

Likelihood of Malicious intent: High 🔴

MITRE mapping: T1546.003 (Event Triggered Execution: WMI Event Subscription).

6) Scheduled Task via PowerShell Cmdlets

Why / how attackers use it:

Instead of using schtasks.exe, adversaries use PowerShell cmdlets like Register-ScheduledTask to set up tasks programmatically inside scripts. This approach:

- Avoids detection rules focused only on schtasks.exe
- Simplifies persistence when malware is delivered via PowerShell payloads
- Provides flexibility in defining triggers and execution contexts

What the KQL detects: powershell.exe where command line has any ("Register-ScheduledTask", "New-ScheduledTask", "New-ScheduledTaskAction", "New-ScheduledTaskTrigger")

IOCs:

- PowerShell with Register-ScheduledTask, New-ScheduledTask
- Actions executing script interpreters, LoLBins, or files in user-writable paths
- Triggers: at logon/startup, frequent minute intervals

Remediation if malicious

- Quarantine, export created tasks/task XML and disable them.
- Collect and remove referenced payloads after analysis.

Likelihood of Malicious intent: Medium 🟠

MITRE mapping : T1053.005.

7) IFEO Debugger Hijack (reg.exe)

Why / how attackers use it:

By modifying the Image File Execution Options (IFEO) registry key, attackers hijack the Debugger field of legitimate binaries (e.g., explorer.exe). This ensures that whenever the target binary is launched, the attacker's malware executes instead. This technique grants stealthy persistence and abuse of trusted system processes for malicious code execution.

What is the KQL detecting: reg.exe where command line has "add "and target "...\Windows NT\CurrentVersion\Image File Execution Options\<EXE>\Debugger"

IOCs:

- Registry writes to HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\<target>\Debugger
- Debugger pointing to unexpected EXE or script in \AppData\ / \Temp\ / network path
- Immediate unexpected process launches when the target app is run

Remediation if malicious:

- Isolate, export the IFEO keys and remove malicious Debugger values.
- Delete referenced binaries after collection and monitor for re-set attempts.

Likelihood of Malicious intent: High 🔴

MITRE mapping: T1546.012 (Event Triggered Execution: IFEO).

8) Winlogon Shell/Userinit Modified (reg.exe)

Why / how attackers use it:

Winlogon registry keys control what executables run during logon. By altering Shell or Userinit, attackers force Windows to launch their payload every time a user logs in. This provides persistence at both system and user level. Such modifications are highly impactful since they ensure execution at a sensitive OS initialization stage.

What the KQL detects: reg.exe where command line has "add " and target "...\Software\Microsoft\Windows NT\CurrentVersion\Winlogon\ (Shell|Userinit)"

IOCs:

- Registry writes to HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\Shell or Userinit
- Values referencing unexpected executables, scripts, or non-default entries
- New files placed shortly before the registry change

Remediation if malicious

- Quarantine, export the Winlogon keys and restore defaults (e.g., explorer.exe, userinit.exe,).
- Remove referenced.
- Perform full compromise assessment, rotate creds if needed.

Likelihood of Malicious intent: 🟠 Medium or 🔴 High (high if use a writtable path OR-EncodedCommand or -enc is present OR matching one of the typical obfuscation/execution functions (frombase64string, Invoke-Expression, IEX))

MITRE mapping: T1547.004 (Boot or Logon Autostart Execution: Winlogon Helper DLL/Keys).

9) Startup LNK via PowerShell (COM CreateShortcut)

Why / how attackers use it:

Dropping .lnk files into Startup folders is a simple but effective persistence mechanism. The shortcut silently points to malicious binaries or scripts, guaranteeing execution whenever the user logs in. Attackers prefer this technique because it:

- Requires no privileges beyond write access to the Startup folder
- Can disguise payloads as normal application shortcuts
- Survives reboots and user logouts reliably

What the KQL detects: powershell.exe with command line matching WScript.Shell.*CreateShortcut.*\.lnk.

IOCs:

- ❖ PowerShell using WScript.Shell → CreateShortcut and writing .lnk files
- ❖ Shortcuts created under user or all-users Startup paths pointing to scripts/LoLBins
- ❖ .lnk targets in \AppData\, \Temp\, \ProgramData\, or containing hidden/encoded arguments

Remediation if malicious:

- ❖ Quarantine, collect the .lnk file and its target, and compute hashes.
- ❖ Delete malicious shortcuts, remove payloads and block the hash/command pattern (if necessary).

Likelihood of Malicious intent: High 🔴

MITRE mapping: T1547.009 (Boot or Logon Autostart: Shortcut Modification/Startup Items).

[Mercuria] - Ransomware Stage 3 - Privilege Escalation

This analytics rule detects suspicious privilege-escalation techniques frequently leveraged by ransomware operators and post-exploitation frameworks to obtain admin/SYSTEM permissions before impact.

Goal: Switch to an administrator or SYSTEM account.

Aggregation: Events are grouped into 15-minute time windows per device (and 1-day for runas)

1) RUNAS.EXE USED FOR PRIVILEGE ESCALATION

Why / how attackers use it:

Attackers abuse runas.exe or PowerShell's Start-Process -Verb RunAs to execute payloads as another account with higher privileges. This allows them to:

- ❖ Escalate from a compromised user to an admin context
- ❖ Execute scripts and LoLBins directly from writable paths
- ❖ Bypass some EDR detections by leveraging native Windows elevation tools

Chaining runas with obfuscated scripts provides reliable escalation during ransomware staging.

What the KQL detects:

- runas.exe where line command has “runas”
- OR “PowerShell” with “Start-Process -Verb runAs “
- OR “cmd.exe” with “-Verb runAs”
- Where the command targets scripts OR LOLBins OR writable path
- ≥2 events within 24h

Targets scripts list: ps1, bat, cmd, vbs, js, psm1

LOLBins list: powershell, cmd, wscript, cscript, regsvr32, mshta

Writable path list: \AppData\, \Temp\, \Roaming\, \Downloads\

IOCs:

- runas.exe / PowerShell -Verb RunAs commands invoking LOLBins
- References to writable list
- Encoded/hidden flags (e.g., -EncodedCommand, -WindowStyle hidden)
- Multiple runas-like events within 24h on the same device

Remediation if malicious:

- Isolate the host.
- Capture process list, command line, user context and parent process and collect payloads and compute hashes.
- Disable the elevated process if running, revoke or rotate any credentials used and check for scheduled re-execution.

FP exclusions: Excludes admin consoles (mmc, gpedit.msc, services.msc, etc.) and benign tools (Chocolatey, WSL install).

Likelihood of Malicious intent: Medium

MITRE mapping: T1548 (Abuse Elevation Control Mechanism); T1059.001 (PowerShell) when applicable.

2) UAC Bypass via LOLBins

Why / how attackers use it:

User Account Control (UAC) is designed to limit unauthorized privilege escalation, but attackers exploit trusted Windows binaries (LOLBins) such as fodhelper.exe, eventvwr.exe, or sdclt.exe to bypass UAC prompts. These binaries can launch payloads silently with elevated privileges by abusing:

- Registry hijacks
- Misconfigured handlers
- Malicious arguments pointing to attacker-controlled paths
  This lets ransomware run elevated code without alerting the user.

What the KQL detects: Executions of those LOLBins

IOCs:

- Execution of fodhelper.exe, eventvwr.exe, sdclt.exe, computerdefaults.exe with unusual args or pointing to user-writable paths.
- Parent process not microsoft-signed or unusual parent.
- Process command lines referencing \AppData\, \Temp\, UNC or HTTP(S) paths.
- Multiple such hits or follow-on activity like payload launches.

Remediation if malicious:

- Isolate the host and gather the offending binary/command lines and parent processes.
- Revert any registry or scheduled changes used in the bypass chain.
- Remove payload artifacts and block known malicious patterns/hashes.

FP exclusions: Exclusions for known-good fodhelper -Embedding via svchost, clean eventvwr launches, signed parents, and typical viewer use. Surfaces non-Microsoft/unsigned parents and user-writable args.

Likelihood of Malicious intent: Medium 🟠

MITRE mapping: T1548.002 (Bypass User Account Control); (secondary) T1218 (Signed Binary Proxy Execution).

3) Privilege Escalation via Local Admin Group

Why / how attackers use it:

By adding accounts to the Local Administrators group, attackers gain persistent administrative control over the endpoint. Even if primary credentials are revoked, a backdoor account ensures continued access. This technique is frequently used in ransomware operations to guarantee elevated rights for deploying payloads, creating services, or disabling defenses.

What is the KQL detecting: ("powershell.exe", "net.exe", "net1.exe") where command line has all ("administrators", "add") or only "Add-LocalGroupMember"

IOCs:

- net user / net localgroup administrators /add or Add-LocalGroupMember commands.
- Parent processes: suspicious or unsigned tools.
- Immediate follow-on actions that require admin rights (service creation, scheduled tasks, service binary drops).

Remediation if malicious:

- Isolate the host and capture the local group state (who was added, when).
- Remove unauthorized accounts and rotate credentials for any accounts suspected to be compromised.

Likelihood of Malicious intent: High 🔴

MITRE mapping: T1098 (Account Manipulation).

4) PowerShell UAC Bypass Script Detected

Why / how attackers use it:

Attackers leverage PowerShell-based UAC bypass techniques because they are easy to script, automate, and replicate across large environments. With commands like Invoke-*UAC or Start-Process -Verb RunAs, adversaries can escalate privileges without relying on binaries like runas.exe. Combined with obfuscation (-EncodedCommand, iex, iwr), these scripts provide a stealthy, repeatable escalation path.

What the KQL detects:

- ("powershell.exe","pwsh.exe") where command line has "invoke-*uac"
- OR has "bypassuac" and "invoke"
- OR has "Start-Process" and "-verb runas"
- Where there is minimum one abuse indicators
- Where the process parent is not signed OR doesn't come from Microsoft OR in ("cmd.exe","wscript.exe","cscript.exe")

Abuse indicators list: -encodedcommand, \\appdata\\, \\temp\\, \\downloads\\, iex , iwr , irm , downloadstring

IOCs:

- PowerShell commands containing Invoke-*UAC, bypassuac, Start-Process … -Verb RunAs.
- Have some abusive indicators
- Non-Microsoft signed parent or use of cmd, wscript as parents.

Remediation if malicious:

- Quarantine the host, collect the PowerShell command lines and any downloaded payload.
- Block the payload hashes and indicators, disable offending user sessions.

FP exclusions: Excludes common dev/installer strings (Chocolatey/winget/dev installers)

Likelihood of Malicious intent: Medium

MITRE mapping: T1548.002 (Bypass User Account Control); T1059.001 (PowerShell).

5) Credential Dumping Indicators (LSASS/Mimikatz)

Why / how attackers use it:

Stealing credentials from LSASS memory is a critical step for lateral movement and privilege escalation. Tools like Mimikatz or procdump.exe extract plaintext passwords, NTLM hashes, and Kerberos tickets. With these, attackers can:

- Impersonate admins and domain accounts
- Move laterally across the network
- Escalate privileges without needing further exploits

This makes LSASS dumping one of the most common and effective escalation techniques in ransomware intrusions.

What the KQL detects:

- procdump.exe where command line has “lsass” and has any (" -ma ", " -mm ", " -r ", " -o ")
- OR rundll32 where command line has “comsvcs.dll” and “MiniDump”
- OR ("powershell.exe","pwsh.exe","cmd.exe") where command line has “Invoke-Mimikatz/sekurlsa::/privilege::debug” OR binaries named mimikatz.

IOCs:

- procdump.exe targeting lsass with full dump flags (-ma, -mm, -o).
- rundll32 comsvcs.dll, MiniDump command patterns.
- PowerShell / CMD lines with Invoke-Mimikatz, sekurlsa::, privilege::debug, or binaries named mimikatz.
- Sudden creation of large .dmp files, or suspicious access to lsass.exe.

Remediation if malicious:

- Isolation of impacted hosts, stop the offending process and collect memory/dump artifacts for analysis.
- invalidate credentials: force password resets/rotate keys for impacted accounts, require re-authentication for privileged accounts, revoke long-lived tokens.
- Hunt and identify lateral movement using harvested creds, consider mandatory credential rotation/conditional access.

Likelihood of Malicious intent: High 🔴

MITRE mapping: T1003.001 (OS Credential Dumping: LSASS Memory); T1003 (OS Credential Dumping).

6) CMSTP UAC Bypass

Why / how attackers use it:

cmstp.exe is a signed Microsoft binary that can process specially crafted .INF files. Attackers exploit this by hosting or dropping malicious INF files in writable paths or remote shares, then executing them silently with flags (/s /ni /ns). This enables privilege escalation without UAC prompts, while leveraging a trusted Windows binary to avoid detection.

What the KQL detects: : cmstp.exe where command line has any has_any ("/s", "/ni", "/ns") AND “.inf” AND source has any (AppData/Temp or http(s))

IOCs:

- cmstp.exe executed with silent flags (/s, /ni, /ns) referencing .inf files in \AppData\, \Temp\ or remote http(s) locations.
- Parent processes unsigned or unusual.
- Follow-on privileged actions or new services installed after cmstp run.

Remediation if malicious

- Isolate host; collect the .inf file and cmstp command line.
- Remove any files installed/created by the INF and block the source URLs/hashes.

Likelihood of Malicious intent: High 🔴

MITRE mapping: T1218 (Signed Binary Proxy Execution: CMSTP); T1548.002 (Bypass UAC).

7) UAC Settings Tampering (reg.exe)

Why / how attackers use it:

By modifying UAC registry keys (e.g., EnableLUA, ConsentPromptBehaviorAdmin), attackers weaken or disable Windows elevation prompts. This ensures that any subsequent admin action runs silently, making privilege escalation frictionless and stealthy. It also reduces the likelihood of a user noticing unusual behavior during ransomware execution.

What the KQL detects:

- reg.exe where command line has “add”
- where the targets are UAC prompts under \Policies\System\ with any (" /d "," /v ")

IOCs:

- reg.exe add modifying HKLM\...\Policies\System\EnableLUA, ConsentPromptBehaviorAdmin, PromptOnSecureDesktop.
- Commands altering these values (/v /d in reg calls).

Remediation if malicious:

- Isolate and export the registry key values (before/after).
- Re-enable secure UAC settings to default and restore keys and audit for further tampering.

Likelihood of Malicious intent: High 🔴

MITRE mapping: T1112 (Modify Registry); T1562.001 (Impair Defenses: Disable/Modify Security Controls).

8) Odbcconf LOLBin Elevation

Why / how attackers use it:

odbcconf.exe can register DLLs or configuration payloads through /A {...} arguments. Adversaries exploit this to load malicious DLLs from writable paths, achieving execution with elevated rights. Since odbcconf.exe is a trusted, signed binary, abusing it provides both privilege escalation and defense evasion by blending into legitimate activity.

What the KQL detects: "odbcconf.exe" where the command line has any " /A " AND "{}" and arguments referencing user-writable paths ("\AppData\", "\Temp\", "\downloads\") OR files (.dll/.rsp/.ini/.txt).

IOCs:

- odbcconf.exe /A {...} with arguments referencing user-writable paths (\AppData\, \Temp\, \downloads\")) or files like .dll, .rsp, .ini.
- Unsigned or non-Microsoft parent process invoking odbcconf.
- Subsequent DLL loads or service installs.

Remediation if malicious:

- Quarantine the host and capture odbcconf command lines and referenced files.
- Remove malicious DLLs and rollback any ODBC or driver registrations.
- Block the offending file hashes and restrict odbcconf.exe usage if possible.

Likelihood of Malicious intent: 🟠 Medium or 🔴 High (high if the command loads a .dll/.ini/.rsp or any commands from a user-writable path (\AppData\, \Temp\, \Downloads\) AND the initiating process is unsigned)

MITRE mapping: T1218 (Signed Binary Proxy Execution).

9) Built-in Administrator Enabled / Password Set

Why / how attackers use it:

Enabling the disabled built-in Administrator account or setting a known password gives adversaries a reliable and privileged backdoor. Unlike normal user accounts, this account:

- Has unrestricted privileges
- Is often overlooked in monitoring
- Can be reused for RDP or lateral login

This ensures persistence and guarantees local admin rights for deploying ransomware payloads.

What the KQL detects: ("net.exe","net1.exe","powershell.exe","pwsh.exe") where commands enabling Administrator, setting its password, or creating local users; excludes /domain and /delete.

IOCs:

- net user administrator /active:yes or net user administrator <password>; PowerShell Enable-LocalUser / Set-LocalUser targeting Administrator.
- Follow-up remote logins or admin-level actions with the built-in account.

Remediation if malicious

- Isolate host.
- Re-disable built-in admin if not required
- Rotate passwords for impacted accounts, check RDP/remote sessions, and hunt for lateral use.

FP exclusions: Excludes the command that contains ("/domain", "/delete")

Likelihood of Malicious intent: Medium 🟠

MITRE mapping: T1136.001 (Create Account: Local Account); T1098 (Account Manipulation); (secondary) T1078 (Valid Accounts).

10) SHADOW-CREDENTIALS TOOL EXECUTED

Why / how attackers use it:

Attackers abuse shadow credentials (KeyCredentialLink objects in AD) to associate new authentication material (keys/certificates) with an existing account. This allows them to authenticate without the real password, providing stealthy persistence and privilege escalation. Tools like Whisker automate this process, enabling long-lived account impersonation resistant to password resets.

What the KQL detects:

- Where cmd has any ("shadowcredentials","shadowcredentials","shadowcreds","msds-keycredentiallink","keycredentiallink","keycredential", "whisker")
- And folder or cmd has writable path ("\\downloads\\", "\\appdata\\", "\\temp\\", "\\roaming\\", "\\users\\public\\", "\\programdata\\")

IOCs:

- Execution of tools or commands containing shadow-credentials, shadowcredentials, msds-keycredentiallink, keycredentiallink, keycredential, or tool names like whisker in user-writable paths.
- Process located in writable path or unsigned parents.
- Rapid use following token theft or AD CS abuse.

Remediation if malicious

- Isolate host
- Remove created shadow-credential entries and any backing key/certificate objects.
- Revoke/replace certificates and keys associated with affected accounts
- Rotate secrets and reset affected account credentials.

Likelihood of Malicious intent: High 🔴

MITRE mapping: T1556.004 – Modify Authentication Process: Add Credentials (Shadow Credentials), T1098 – Account Manipulation

11) LOCAL ACCOUNT CREATION (NET/NET1/POWERSHELL)

Why / how attackers use it:

Creating new local accounts gives attackers persistent footholds, especially useful when domain accounts are monitored or rotated. These accounts can later be escalated to local admin group membership, allowing:

- RDP or SMB access

- Privileged execution of ransomware loaders
- Backup access in case domain creds are lost

This technique is widely used for backdoor persistence in ransomware playbooks.

What the KQL detects: ("net.exe","net1.exe") where cmd matches "net user <name> <pwd?> /add" AND cmd has not "/domain" OR ("powershell.exe","pwsh.exe") where cmd has any ("New-LocalUser" "net user <name> <pwd?> /add")

IOCs:

- net user <name> /add or PowerShell New-LocalUser commands that create local accounts (excluding /domain).
- Newly created local accounts followed by password set or admin group changes.
- Unsigned parent processes or unusual install paths.

Remediation if malicious:

- Quarantine, enumerate and document newly created local accounts.
- Remove unauthorized accounts and audit for any uses (remote logon events).
- Inspect for related persistence (task/service) and rotate local credentials if needed.

Likelihood of Malicious intent: High 🔴

MITRE mapping: T1556.004 – Modify Authentication Process: Add Credentials (Shadow Credentials); T1098 – Account Manipulation

12) AD CS ABUSE VIA CERTIPY (REQUEST/AUTH/SHADOW/ENUM)

Why / how attackers use it:

Abuse Active Directory Certificate Services (AD CS) to issue valid Kerberos/AD certs for accounts they control or impersonate (ESC1/ESC8/ESC11, etc.). A minted PFX or user/computer certificate enables "Pass-the-Certificate" auth to domain resources (Kerberos PKINIT/Smartcard logon) endpoint-agnostic, long-lived, MFA-bypassing in many environments.

Certipy automates enumeration/vuln discovery/request/abuse providing priv-esc and stealthy persistence without dumping passwords/tickets.

What the KQL detects: Process ("certipy.exe", "certipy2.exe", "python.exe") where the command line has "certipy" AND cmd has any ("req", "auth", "cert", "find, "shadow", "adcs").

Abusive flags list: -ca, -template, -pfx, -hashes, -target, -upn, -dc-ip, -username, -password, -kerberos

IOCs:

- certipy or Python invocations with certipy parameters: req, auth, find, shadow, adcs
- Parent unsigned or non-Microsoft process launching Certipy.
- Presence of abusive flags (request or pfx export, UPN templating).

Remediation if malicious

- ❖ Isolate the host, capture Certipy commands/output and any generated certificates/PFX files.
- ❖ Revoke suspicious certificates immediately and change/disable any accounts targeted.
- ❖ Audit AD CS templates and harden by removing dangerous template permissions (enrollment rights), enable enrollment approvals, and monitor certificate requests.
- ❖ Hunt for lateral use of issued certificates (SMB/LDAP/Kerberos service ticket impersonation)

FP exclusions: Exclude if parent unsigned or parent not in Microsoft

Likelihood of Malicious intent: Medium 🟠 or 🔴 (high if request/authentication or shadow)

MITRE ATT&CK mapping: T1649 – Steal or Forge Authentication Certificates (core technique); T1550 – Use Alternate Authentication Material (Pass-the-Certificate via issued PFX/PKINIT); T1078 – Valid Accounts (resulting certificate grants valid auth/persistence).

[Mercuria] - Ransomware Stage 4 - Defense Evasion

This analytics rule detects defense-evasion techniques frequently used by ransomware and post-exploitation tooling to blind telemetry, remove recovery options, and conceal artifacts prior to encryption/impact.

Goal: Neutralize protections and evade detection.

Aggregation: Events are grouped in 15 minutes bins per device

1)EXCESSIVE USE OF TASKKILL.EXE (TARGETED)

Why / how attackers use it:

Attackers pre-empt defenses by force-killing AV/EDR, backup and database agents so encryption or data theft can proceed uninterrupted. They often run bursts of taskkill /F /IM … against a curated list of protective/backup processes, sometimes chained inside scripts to re-issue kills if services respawn, or run them right before shadow-copy deletion to block recovery.

What the KQL detects: taskkill.exe runs with /F and /IM (or an .exe token) that target high-risk processes (AV/EDR/backup/DB/VSS listed in High Risk Kill Tokens)

High Risk Token list:

- AV / EDR: "msmpeng.exe","mpcmdrun.exe","nissrv.exe","mssense.exe","senseir.exe","sensece.exe", "csfalconservice.exe","csfalconcontainer.exe","csagent.exe", "sentinelagent.exe","sentinelservicehost.exe","sentinelhelperservice.exe","parity.exe","repux.exe","cbdefensewsc.exe", "ccsvchst.exe","rtvscan.exe","smc.exe","avp.exe","ksde.exe","tmlisten.exe","ntrtscan.exe", "savservice.exe","sophosfilescanner.exe","sophosfs.exe","hmpalert.exe","mfemms.exe","mcshield.exe", "masvc.exe","mfevtps.exe","ekrn.exe","bdservicehost.exe","epsecurityservice.exe","cylancesvc.exe",
- Backup/DR: "veeamagent.exe","veeam.endpoint.service.exe","veeamdeploymentsvc.exe", "trueimageservice.exe","acronisagentservice.exe","acronisactiveprotectionservice.exe", "cvd.exe","cvfwd.exe","bpbkar32.exe","bpcd.exe","nbdisco.exe", "beremote.exe","bengine.exe","bemgr.exe","arcserveudpservice.exe",
- DB & Windows backup: "sqlservr.exe","sqlbrowser.exe","sqlwriter.exe","mysqld.exe","mariadbd.exe","postgres.exe", "pg_ctl.exe","oracle.exe","tnslsnr.exe","vssvc.exe","wbengine.exe"]);

IOCs:

- taskkill /F with /IM or explicit *.exe names that match AV/EDR/backup/DB processes
- Bursts of multiple distinct taskkill commands in minutes.
- Unsigned/suspicious parent.

Remediation if malicious

- Immediately isolate host and stop remaining kill scripts.
- Restore/repair AV/EDR services and collect artifacts (commands, parent PIDs, hashes).
- Validate backups are intact and reachable and consider restoring impacted services.

FP exclusions: The query explicitly excludes calls that only use /PID (common benign admin operations).

Likelihood of Malicious intent: High 🔴

MITRE mapping: T1562.001 (Defensive Evasion: Disable/Modify Tools) / T1490 (Impact — Data Destruction) Signed-binary proxy execution patterns.

2)REPEATED USE OF NET STOP

Why / how attackers use it:

net stop is a blunt instrument to turn off services fast (logging, VSS, backup, EDR helpers). It's noisy but effective during hands-on operations and prior to ransomware execution. Adversaries script multiple net stop <svc> calls in quick succession to disable protections, then move immediately to wiping or encryption.

What the KQL detects: Multiple "net.exe" where command line has "stop" (count >1 within 5 min).

IOCs:

- Several net stop <service> commands close together, especially for AV/backup/logging services
- Parent is script, PowerShell or unsigned binary
- Happens before shadow-copy deletion or file IO bursts

Remediation if malicious

- Quarantine the endpoint.
- Re-enable and lock the affected services (set Start=Automatic, enforce service recovery)
- Investigate who executed commands; rotate credentials if compromised.

FP exclusions: Excludes many known benign services (audio, WSL, Docker, some vendor agents like Nexthink).

Likelihood of Malicious intent: Medium 🟠

MITRE mapping: T1489 (Impact — Service Stop) / T1497 (Impair Defenses).

3)SERVICES DISABLED USING SC CONFIG

Why / how attackers use it:

Changing service startup to disabled ensures defenses don't come back after reboot. It's a persistence-of-evasion step: even if SOC restarts a host, critical protections (e.g., Windefend/VSS/backup agents) remain off, buying the attacker time to stage, exfiltrate, or re-launch encryption.

What the KQL detects:

- Detect command line that have all ("sc", "config", "disabled")
- Signal escalates to High if any high-risk services match

High risk services flag list: winmgmt, pangps, windefend, sense, wdnissvc, mbamservice, savservice, sepmastersvc, sentinelagent, csagent, csfalconservice, cylancesvc, veeamagent, veeam.endpoint. service, sqlwriter, sqlservr, vss, wbengine

IOCs:

- sc config <service> start= disabled targeting high-value services
- Many different services disabled in a burst
- High-risk services match
- Unsigned or unusual parent

Remediation if malicious:

- Isolate
- Revert service configuration (start= auto), start services, enable tamper protection.
- Review services.msc and registry HKLM\SYSTEM\CurrentControlSet\Services
- Block responsible tool.

FP exclusions: Excludes some benign services ("ccmexec","wwan firmware flash service","wwan","fbwwanfilterservice","thingsmatrixagentservice")

Likelihood of Malicious intent: Medium 🟠 or High 🔴 (high if risk match)

MITRE mapping: T1489 (Impact — Service Stop) / T1562 (Impair Defenses).

4)EVENT LOGS CLEARED VIA WEVTUTIL

Why / how attackers use it:

Clearing logs erases investigative breadcrumbs (Privilege Escalation, Tooling, Execution traces). Attackers iterate wevtutil cl across key channels (Security, System, PowerShell, Microsoft-Windows-*), often after disabling auditing and before exfil/encryption, to delay detection and frustrate IR timelines.

What the KQL detects:

- Detect command line that have all ("wevtutil", "cl").
- Triggers when many logs are cleared (LogClearCount > 3).

IOCs:

- Multiple wevtutil cl <logname> commands (e.g., Security, System, PowerShell).
- Unsigned/suspicious parent process.

Remediation if malicious

- Quarantine immediately, export remaining logs (forwarded/central collectors) and EDR telemetry.
- Enforce log forwarding and locked permissions and monitor for wevtutil abuse.

Likelihood of Malicious intent: High 🔴

MITRE mapping: T1070.001 (Indicator Removal: Clear Windows Event Logs).

5)VOLUME JOURNAL DELETION VIA FSUTIL

Why / how attackers use it:

Deleting the NTFS USN journal cripples forensic reconstruction of file operations (no delta history), making it harder to map what was touched or recovered. Ransomware and wipers pair this with shadow-copy deletion to break both recovery and evidence.

What the KQL detects: "fsutil.exe" where command line has all ("usn", "deletejournal")

IOCs:

- fsutil usn deletejournal /d (or variants) on system volumes.
- Privileged console/PowerShell launching fsutil.
- Temporal proximity to encryption or mass file actions.

Remediation if malicious:

- Isolate host.

- ❖ Capture volume metadata and EDR traces and review recent file operations from central sensors.
- ❖ Disable local admin where not needed.

Likelihood of Malicious intent: High 🔴

MITRE mapping: T1070.004 (Indicator Removal — File/Artifact Removal).

6)Shadow Copy Deletion (vssadmin/diskshadow/wmic)

Why / how attackers use it:

Shadow copies are the fastest path to recovery; removing them maximizes impact and increases pressure to pay. Adversaries use multiple methods (vssadmin delete, DiskShadow scripts, wmic shadowcopy delete) to survive allowlists and ensure no restore points remain.

What the KQL detects: vssadmin delete shadows, diskshadow scripts invoking deletion, or wmic shadowcopy delete/call delete patterns.

IOCs:

- ❖ vssadmin delete shadows, diskshadow -s <script>, or wmic shadowcopy delete.
- ❖ Often accompanied by service stops (VSS, SQLWriter, wbengine) and backup app kills.
- ❖ Unsigned parent or batch/PowerShell driver.

Remediation if malicious

- ❖ Isolate host and halt encryption processes if active.
- ❖ Verify off-host backups and plan restore path.
- ❖ Block the commands via policy, enable backup tamper protection and review VSS provider status.

Likelihood of Malicious intent: High 🔴

MITRE mapping: T1490 (Impact — Inhibit System Recovery) / T1070 (Indicator Removal).

7)File Hiding via attrib.exe

Why / how attackers use it:

Marking payloads/staging files as Hidden (and sometimes System/Read-only) reduces casual discovery by users and basic IT hygiene. Attackers hide droppers, scripts, exfil staging archives, or tooling in user-writable locations to blend into clutter and delay manual detection.

What the KQL detects:

- ❖ "attrib.exe" where command line has “+h”
- ❖ Where it is target to user-writable OR to script/binary extensions.
- ❖ Trigger if multiple hits (>=3) within 5 min

Writable path accepted: \appdata\, \temp\, \roaming\, \downloads\, \users\public\, \programdata\

Script/binary extension accepted: ps1, psm1, bat, cmd, vbs, js, jse, wsf, exe, dll, scr

IOCs:

- attrib +h (± +s/+r) against files in writable path
- Targets have scripts/binaries extenstion
- Multiple hide operations within minutes.

Remediation if malicious:

- Quarantine and enumerate recently hidden files (attrib -h -s in staging dirs).
- Collect and detonate samples, block hashes and clean drop locations.

FP exclusions: Excludes a specific benign initiator (draw.io.exe)

Likelihood of Malicious intent: Low 🟡

MITRE mapping: T1564 (Hide Artifacts)

8)Tampering with Microsoft Defender via PowerShell

Why / how attackers use it:

Set-MpPreference changes (exclusions, disabling real-time/script scanning) carve out safe execution zones where malicious binaries/scripts can run unscanned. This is frequently automated with encoded PowerShell and executed pre-encryption so the payload and its helpers are not blocked.

What the KQL detects: ("powershell.exe", "powershell_ise.exe") where command lines have any ("set-mppreference","add-mppreference","remove-mppreference","-exclusionpath","-exclusionprocess", "-exclusionextension","-disablerealtimemonitoring","-disableioavprotection","-disablescriptscanning")

IOCs

- PowerShell with Set-MpPreference / Add/Remove-MpPreference and flags like -DisableRealtimeMonitoring, -ExclusionPath, -ExclusionProcess, -ExclusionExtension, -DisableIOAVProtection, -DisableScriptScanning.
- Non-authorized admin contexts performing changes.

Remediation if malicious:

- Isolate and revert Defender settings (remove exclusions, re-enable protections; enforce tamper protection).
- Review who triggered changes; rotate creds if abused.
- Baseline Defender policies with Intune/GPO and block local overrides.
- Hunt for payloads executed while protections were disabled.

FP exclusions: Query includes a benign filter related to JetBrains Rider

Likelihood of Malicious intent: High 🔴

MITRE mapping: T1562.001 (Impair Defenses: Disable windows event logging) / T1562 (Impair Defenses).

9) Attempt to Disable Cortex XDR

Why / how attackers use it:

Targeting Cyvera/Traps/XDR services or processes directly aims to blind the endpoint and remove behavioral protection. Attackers try multiple levers (SC/NET stop, taskkill /F, PowerShell Stop-Service/Set-Service) and often re-issue the commands to defeat agent self-protection.

What the KQL detects:

- (“sc.exe”, "net.exe”) where encoded command lines have any ("cyveraservice","traps","palo alto cortex xdr","cortex xdr") AND has any (“stop”, “config”, “delete”)
- “taskkill.exe” where encoded command lines have any ("cyserver.exe","cyveraservice.exe","traps.exe") AND has “/f”
- “PowerShell” invocations targeting Cortex/XDR service/process names (including decoding of base64-encoded command args).

IOCs:

- sc stop/config/delete, net stop, taskkill /f against Cyvera/Traps/XDR services or processes.
- PowerShell Stop-Service / Set-Service targeting those services, sometimes base64-encoded.
- Unsigned or suspicious parents launching the commands.

Remediation if malicious:

- Quarantine, re-enable XDR agent and tamper protection and force re-registration.
- Collect commands and parents that blocked tooling.
- Hunt for EDR-disabled hosts and validate policy inheritance and prevent local tampering.

Likelihood of Malicious intent: High 🔴

MITRE mapping: T1562.001 (Impair Defenses) / T0894 (System binary proxy execution).

10) VPN Configuration Command

Why / how attackers use it:

Creating or altering VPN connections provides an on-demand egress path (exfiltration) or backdoor access that bypasses corporate proxies and DLP. Adversaries script Add-VpnConnection / rasdial / netsh interface to stand up tunnels to non-corporate endpoints, then move data or return later without noisy C2.

What the KQL detects: Where command lines has any ("Add-VpnConnection", "Set-VpnConnection", "rasdial", "netsh interface set", "netsh interface add") used to change VPN/network interfaces

IOCs:

- Manual VPN add/set or netsh interface add/set without the standard corporate VPN client/parent.
- New connections pointing to unknown servers or non-corporate DNS names.
- Follow-on outbound connections to unfamiliar IPs.

Remediation if malicious:

- Isolate device, export current VPN profiles and remove unauthorized connections and credentials.
- Block egress to suspicious endpoints and review firewall and DNS logs for matching traffic.

FP exclusions: Excludes Panorama/GlobalProtect helpers (PanGPA.exe, PanSupport.exe) and ignores show queries

Likelihood of Malicious intent: Medium 🟠

MITRE mapping: T1090 (Proxy/Egress — Application Layer Protocols) / T1133 (External Remote Services, when used to persist remote access).

11) AUDIT POLICY DISABLED (AUDITPOL)

Why / how attackers use it:

Turning off success/failure auditing and setting "no auditing" suppresses telemetry at the source, reducing SIEM visibility into logon events, object access, or policy changes. Attackers pair this with log clearing to erase the past and mute the future, buying quiet dwell time.

What the KQL detects: "auditpol.exe" with:

- Commands that disable success/failure auditing (has "/set" AND has any ("success:disable","failure:disable","no auditing"))
- OR command that set categories/subcategories to no auditing / disable.

IOCs

- auditpol /set with success:disable, failure:disable, or no auditing on categories/subcategories.
- Executed by non-authorized admins and followed by other evasion steps (log clear, service stop).

Remediation if malicious.

- Quarantine, restore audit baselines via GPO/Intune and verify they persist after reboot.
- Rotate credentials if needed.
- Ensure central log forwarding so local changes don't suppress telemetry.

FP exclusions: Excludes some management agents (ccmexec, intunemanagementextension, softwarecenter, gpscript) when the commands are read/query operations (/r, /get).

Likelihood of Malicious intent: High 🔴

MITRE mapping: T1562 (Impair Defenses — Disable or Modify Security Tools) / T1098 (Account Manipulation when used in concert with policy changes).

12) FIREWALL DISABLED (WINDOWS FIREWALL)

Why / how attackers use it:

Disabling local firewall profiles opens unfiltered inbound/outbound paths for C2, lateral movement, and exfil. It also neutralizes host-based containment policies. Adversaries use netsh advfirewall … state off or PowerShell (Set-NetFirewallProfile -Enabled False), often encoded and chained with service stops and VSS deletion during the impact phase.

What the KQL detects: Where ("powershell.exe","pwsh.exe","netsh.exe","net.exe","cmd.exe") and cmd has any (“Set-NetFirewallProfile -Enabled False”, “netsh advfirewall set allprofiles state off”, ”netsh advfirewall set <domain or private or public>profile state off”, “disable-firewall”, “netsh firewall set opmod”, "advfirewall set allprofiles state off")

IOCs:

- Set-NetFirewallProfile -Enabled False; advfirewall set allprofiles state off; netsh … domainprofile state off.
- Use of encoded PowerShell (-EncodedCommand) to mask intent.
- Command launched by an unsigned process or unauthorized account.
- Disabling followed quickly by other indicators (taskkill/sc config disable/vssadmin delete/hosts tampering/suspicious network connections).

Remediation si malveillant:

- Isolate the host
- Immediately reactivate firewall profiles (Set-NetFirewallProfile -Enabled True or via GPO/Intune) and restore the firewall baseline.
- Apply/enforce the central policy (GPO/Intune) and verify that it persists after reboot.
- Capture forensics (process tree, EDR logs, network flows), look for associated changes (hosts, permissive firewall rules).
- Block/contact the IP/C2 if identified; consider rotating credentials if a compromise is proven.

FP exclusions: where not cmd contains lecture flags ("show", "get-netfirewallrule", "get-netfirewallprofile")

Likelihood of Malicious intent: Medium 🟠 or High 🔴 (if unsigned parent or encoded commands or count of service disabled >3)

MITRE mapping: T1562.004 (Impair Defenses, Disable or Modify Security Tools, system/cloud firewall)

[Mercuria] - Ransomware Stage 5 - Internal Reconnaissance Activity

This analytics rule detects internal reconnaissance activity frequently performed during ransomware staging and post-exploitation campaigns.

Goal: Map the network and identify critical targets.

Aggregation: Events are grouped in 15 minutes bins per device

1) Combined Host and Network Reconnaissance

Why / how attackers use it:

Operators front-load quick situational awareness to choose lateral paths and choke points. They fingerprint the identity & privilege (whoami), host posture (systeminfo), and network reachability (ipconfig /all, arp) while probing name resolution & adjacency (nbtstat, nslookup, net view, tracert). Mixing multiple native tools within a short window makes the recon resilient to single-signal detections and reveals internal targets (RFC1918 ranges, corp DNS suffixes) for staging, share hunting, and credential operations.

What the KQL detects: Correlated execution of host-enum tools (whoami, hostname, systeminfo, ipconfig /all) plus multiple network tools (ping, arp, nbtstat, net view, tracert, nslookup, nltest) within ~30 minutes, with anti-noise thresholds (≥2 non-ping, ≥3 distinct tools) and internal/external target extraction.

IOCs:

- Short-window bursts of whoami, hostname, systeminfo, ipconfig /all plus network tools (ping, arp -a, nbtstat, net view \\, tracert, nslookup, nltest).
- Multiple distinct tools with multiple non-ping commands.
- Targets include RFC1918 IPs, internal hostnames, or corp suffixes.
- Unsigned or unusual parents launching the tools.

Remediation if malicious

- Isolate host and preserve EDR telemetry/command lines.
- Review scope: queried hosts, trusts, and shares; block outbound from compromised account.

FP exclusions: Exclude common ping targets (url ping, local host, google)

Likelihood of Malicious intent: Medium 🟠 or Low 🟡 (Medium if network reconnaissance is done with one of these tools "net","nbtstat","nltest")

MITRE mapping: T1082 (System Information Discovery), T1033 (Account Discovery), T1016 (System Network Configuration Discovery), T1018 (Remote System Discovery).

2) AD Enumeration via PowerShell (Get-AD*)

Why / how attackers use it:

PowerShell's AD cmdlets provide structured, high-fidelity directory maps without dropping external tools. Adversaries enumerate users, computers, groups, OUs, trusts, and attributes (SPNs, admin flags) to identify admin paths and weak links. Filters and custom selections (-Filter, -LDAPFilter, -SearchBase, -Properties) enable broad but stealthy sweeps that blend with admin activity and feed into privilege-escalation and lateral-movement planning.

What KQL detects:

- ("powershell.exe","pwsh.exe") where the command line has “Get-AD<*>”
- Where the filters/selects implies broad directory queries => where cmd has any “computer”, “user”, “group”, “domain”, “forest”, “organizationalunit”, “object”, “trust”) AND has any ("-filter", "-ldapfilter", "-searchbase", "-properties", " | select")

IOCs:

- PowerShell with Get-AD* (e.g., Get-ADUser, Get-ADGroupMember) using -Filter, -LDAPFilter, -SearchBase, -Properties, | Select.
- Non-management parent processes (not SCCM/Intune) and unsigned parents.
- Repeated broad queries in a short interval.

Remediation if malicious:

- Quarantine device and review AD audit logs for the querying account.
- Temporarily restrict the account and rotate credentials if necessary.

FP exclusions : Excludes ("intunemanagementextension.exe","ccmexec.exe","softwarecenter.exe","companyportal.exe")

Likelihood of Malicious intent: Medium 🟠

MITRE mapping: T1069.002 (Permission Group Discovery: Domain), T1087.002 (Account Discovery: Domain Account), T1482 (Domain Trust Discovery).

3) AD/DC Discovery via NLTEST

Why / how attackers use it:

nltest is a fast lane to the domain’s backbone, listing DCs, sites, and inter-domain trusts. Attackers use it to pick nearby DCs (low latency, higher success), understand trust boundaries for cross-domain pivots, and validate domain presence before Kerberos abuse (DCSync, AS-REP roasting, constrained delegation).

What the KQL detects: “nltest.exe” where command line has any ("/dclist", "/dclist:", "/domain_trusts", "/dsgetdc:", "/trusted_domains")

IOCs:

- nltest /dclist, /dsgetdc:, /domain_trusts, /trusted_domains.
- Unsigned parents or uncommon parents on workstations outside admin troubleshooting.
- Temporal proximity to credential dumping or Kerberos abuse.

Remediation if malicious:

- Isolate and review AD trust and DC authentication logs for the actor.
- Monitor for subsequent DC-targeted activity (DCSync, replication abuse).

Likelihood of Malicious intent: High 🔴

MITRE mapping: T1482 (Domain Trust Discovery), T1018 (Remote System Discovery).

4) SPN Enumeration via setspn -Q

Why / how attackers use it:

SPNs reveal service accounts that can be Kerberoasted or targeted for lateral movement (SQL, HTTP app pools, CIFS). Attackers query broad classes (e.g., */, HTTP/, MSSQL/) to surface high-value service principals, then request TGS tickets en masse for offline cracking and privilege escalation. It's a classic bridge from recon to initial privesc.

What the KQL detects:

- "setspn.exe" where the command line has "-q" and optionally a "-f"
- Where cmd has "*/" OR cmd has common service classes ("ldap", "http", "cifs", "host", "mssql", "wsman", "smtp")

IOCs:

- setspn -Q */ or protocol-specific queries (LDAP/, HTTP/, CIFS/, MSSQL/, etc.).
- Burst of SPN queries from a non-admin workstation.
- Follow-on signs: TGS requests spikes, hash-cracking artifacts.

Remediation if malicious:

- Isolate and review KDC/TGS request spikes from the account.
- Enforce complex, long passwords on service accounts; consider gMSA.
- Rotate high-value SPN credentials.

Likelihood of Malicious intent: High 🔴

MITRE mapping: T1558.003 (Steal or Forge Kerberos Tickets: Kerberoasting — reconnaissance precursor), T1087.002 (Account Discovery: Domain Account).

5) Privileged AD Group Enumeration

Why / how attackers use it:

Ransomware crews quickly identify who owns the crown jewels (Domain/Enterprise/Schema Admins). Enumerating these groups highlights credential targets for phishing, token theft, or session hijack, and helps plan Tier-0 takeover. Results also guide password-spray focus and Golden Ticket ambitions.

What the KQL detects:

- ("net.exe","net1.exe", "dsquery.exe","dsget.exe") where cmd "group" AND has any ("domain admins", "enterprise admins","schema admins")
- OR ("powershell.exe","pwsh.exe") where cmd "get-adgroupmember" AND has any ("domain admins", "enterprise admins","schema admins")

IOCs:

- net group "Domain Admins" /domain, dsquery/dsget group, or Get-ADGroupMember for privileged groups.
- Unsigned/suspicious parent and queries from non-admin endpoints.
- Clustered with other AD recon (SPN, NLTEST).

Remediation if malicious

- Isolate, audit access to privileged groups and check if membership changed.
- Monitor targeted admin accounts for logons/NTLM/Kerberos anomalies.
- Enforce tiered admin model and PAWs (Privileged Access Workstations).

Likelihood of Malicious intent: 🟠 Medium

MITRE mapping: T1069.002 (Permission Group Discovery: Domain).

6) BloodHound/PowerView Collection Activity

Why / how attackers use it:

Graph-based collection (SharpHound/PowerView) turns raw AD data into attack paths: local admin edges, ACL abuse, unconstrained delegation, shadow admins. Operators run stealthy or all-methods collection to build a roadmap to domain dominance and pinpoint the minimum-effort escalation route—often right before lateral sprawl and impact.

What the KQL detects: ("sharphound.exe","bloodhound.exe","powershell.exe","pwsh.exe") where cmd has any ("invoke-bloodhound", "sharphound.ps1", "collectionmethod", "get-netdomain", "get-netcomputer","get-netuser","get-netgroup", "invoke-sharefinder", "find-localadminaccess", "get-objectacl", "get-domaingroupmember")

IOCs:

- sharphound.exe, bloodhound.exe, or PowerShell Invoke-BloodHound, SharpHound.ps1, plus discovery verbs (Get-Net*, Find-LocalAdminAccess, Get-ObjectACL).
- Large, multi-minute enumeration and archives or JSON output in temp/user folders.
- Unsigned parents or execution from user-writable paths.

Remediation if malicious

- Quarantine, seize collection files and block hashes.
- Hunt for the same user/tool across endpoints.

Likelihood of Malicious intent: High 🔴

MITRE mapping: T1069.002 (Permission Group Discovery: Domain), T1087.002 (Account Discovery: Domain), T1482 (Domain Trust Discovery).

7) SMB Share Enumeration (net view sweep)

Why / how attackers use it:

Share discovery finds where the data lives and which hosts accept remote file ops. Attackers sweep net view \\host across subnets to list shares, then test access for staging, tool hosting, and later mass encryption. It also reveals soft targets (weak ACLs, open ADMIN$/C$) for PsExec-style execution.

What the KQL detects: "net.exe" where cmd has “view” AND “\\<host>”

IOCs:

- Repeated net view \\<host> across many hosts; enumeration of shares in quick succession.
- Executed from non-IT devices
- Unsigned/suspicious parent process.
- Follow-on: file copy bursts, access denied events, or auth failures to many hosts.

Remediation if malicious:

- Isolate and review SMB logs from targeted hosts (success/failed access).
- Tighten share ACLs, disable unnecessary admin shares and enable SMB signing.
- Hunt for subsequent mass file operations or archive creation.

Likelihood of Malicious intent: Medium 🟠

MITRE mapping: T1135 (Network Share Discovery), T1018 (Remote System Discovery).

8) Forensic Artifact Parsers Executed (Amcache/AppCompat)

Why / how attackers use it:

Advanced crews sometimes read Amcache/AppCompat to learn what software and versions exist (EDR, backup, browsers) and to identify living-off-the-land options or vulnerable apps. Others use these tools for anti-forensics—copying or reviewing artifacts before wiping, to understand what evidence exists and adjust evasion accordingly.

What the KQL detects: ('AppCompatCacheParser.exe', 'AmcacheParser.exe', 'RECmd.exe', 'rip.exe', 'regripper.exe', ‘reg.exe’) where commands lines have any (“save”, “export”,” copy”) AND has “\Windows\AppCompat\Programs\Amcache.hve”

IOCs:

- Execution of AmcacheParser.exe, AppCompatCacheParser.exe, RECmd, rip.exe/regripper.
- reg save/export …\AppCompat\Programs\Amcache.hve.
- Run from temp/user paths, unusual on non-IR workstations.

Remediation if malicious

- Quarantine and secure copies of Amcache/AppCompat before they’re altered.
- Review why these tools ran and block non-IR endpoints from using them.
- Monitor for subsequent log tampering or cleanup activity.

Likelihood of Malicious intent: Medium 🟠

MITRE mapping: T1518 (Software Discovery), T1082 (System Information Discovery).

9) AD/DC Discovery via DNS SRV Records

Why / how attackers use it:

Querying _ldap._tcp, _kerberos._tcp, _gc._tcp, _kpasswd._tcp via SRV records yields authoritative DC endpoints without admin tools. This DNS-first approach blends with normal lookups, works from locked-down shells, and immediately informs Kerberos targeting, LDAP binds, and follow-on authentication against the right servers.

What the KQL detects:

- (“Nslookup”, "dig.exe") where cmd has any ("-type=srv","-q=srv","-type=SRV","-q=SRV") AND has any (“ldap”, “Kerberos”, “kpasswd”, “gc”)
- OR ("powershell.exe","pwsh.exe") where cmd has “Resolve-DnsName -Type SRV” AND has any (“ldap”, “Kerberos”, “kpasswd”, “gc”)

IOCs

- nslookup -type=SRV _ldap._tcp.<domain>, Resolve-DnsName -Type SRV, or dig <name> SRV.
- Follow-on LDAP/Kerberos traffic to discovered hosts
- Unusual parent process.

Remediation if malicious

- Isolate host and confirm whether DC targets received follow-up auth.
- Block egress DNS from endpoints to non-corporate resolvers

Likelihood of Malicious intent: Medium 🟠

MITRE mapping: T1016.001 (System Network Configuration Discovery: DNS), T1482 (Domain Trust Discovery).

[Mercuria] - Ransomware Stage 6 - Lateral Movement

This analytics rule surfaces lateral movement toolmarks commonly used in ransomware staging and post-exploitation.

Goal: Spread to other machines on the network.

Aggregation: Events are grouped in 5–10 minutes bins per device

1)Remote Execution via WMIC

Why / how attackers use it:

Operators leverage WMIC to launch processes on remote hosts without interactive logon by abusing WMI RPC. It blends with enterprise management traffic, works with just credentials + RPC reachability, and scales well for spraying payloads/scripts across many machines. Because execution is proxied through WMI providers, child processes often appear decoupled from an RDP/console session, reducing simple user-activity heuristics.

What is the KQL detecting: "wmic.exe" where cmd has "process call create" OR matches /node: patterns ("\b/node\s*:", "/node:\s*\\\\", "/node\s+\\\\").

IOCs

- wmic.exe process calls containing process call create or /node: /node: \\<host> patterns.
- Commands executed that spawn processes on remote systems.
- Initiating process not signed or not a known management tool
- Unusual source account.

Remediation if malicious:

- Isolate source host from network immediately.
- Collect WMIC command lines, process trees and remote target lists.
- Block or restrict WMIC usage via AppLocker/WDAC or GPOs for non-admin endpoints.

Likelihood of Malicious intent: High 🔴

MITRE mapping: T1047 (Windows Management Instrumentation).

2)Remote Execution via PsExec

Why / how attackers use it:

PsExec is the workhorse of SMB/SCM remote execution: copy a binary over ADMIN$ and create a service that runs it. It's fast, reliable, and trivial to automate for wide fan-out deployment (encrypters, loaders, tools). Even when detections target PsExec, actors pivot to PsExec-like flows (e.g., alternate binaries, renamed copies) because the service-based model is universally available on Windows fleets.

What the KQL detects:

- File name in (“psexec.exe”, psexesvc.exe”)
- OR “sc.exe” where cmd has “create” AND “PSEXESVC”

IOCs

- psexec.exe or psexesvc.exe activity, or sc create PSEXESVC commands.
- ProcessCommandLine containing psexec and \\<host> targets.
- New service creation on remote hosts or simultaneous similar commands across many machines.

Remediation if malicious:

- Quarantine the initiating endpoint and collect artifacts (PsExec logs, EDR telemetry).
- Disable or monitor the SMB admin shares and block lateral SMB from user segments.
- Remove stored credentials and investigate account usage; enforce credential rotation for compromised accounts.

Likelihood of Malicious intent: High 🔴

MITRE mapping: T1059.003 (Windows PsExec).

3)SUSPICIOUS PROCESSES SPAWNED BY WMIPRVSE.EXE

Why / how attackers use it:

Abusing WMI to spawn LOLBins and script interpreters under wmiprvse.exe gives “remote yet local” execution that inherits SYSTEM or high-integrity context and avoids an interactive logon trail. Attackers chain in encoded PowerShell, IEX/IWR, and UNC/HTTP fetch to stage and run payloads while making the activity look like normal instrumentation rather than hands-on keyboard.

What the KQL detects:
• Parent wmiprvse.exe AND child FileName in (cmd.exe, powershell.exe, pwsh.exe, wscript.exe, cscript.exe, regsvr32.exe, mshta.exe, rundll32.exe, net.exe, net1.exe) AND ProcessCommandLine matches abuse flags

Abuse flag list: -enc, -encodedcommand, -nop, -w hidden, /c , invoke-webrequest, iwr, invoke-expression, iex, downloadstring, start-bitstransfer, mshtml.dll,runhtmlapplication, javascript:, scrobj.dll, /i:, http://, https://, \\\\, \\admin$, \\c$

IOCs:

- wmiprvse.exe as parent with children like cmd.exe, powershell.exe, regsvr32.exe, mshta.exe, etc.
- Commands containing abusive flags.
- Execution under non-system accounts or in bursts with multiple children.

Remediation if malicious: Isolate affected host(s) and collect wmiprvse.exe process trees and WMI event logs.

FP exclusions: exclude system account AND excludes SCCM common cached Users*.cmd AND known management parents ("ccmexec.exe","intunemanagementextension.exe","softwarecenter.exe","sccmsetup.exe")

Likelihood of Malicious intent: High 🔴

MITRE mapping: T1543 (Create or modify system process).

4)Remote Registry READ/WRITE via reg.exe

Why / how attackers use it:

Remote registry lets adversaries change persistence and security posture at a distance (Run/IFEO keys, service parameters, RDP/firewall settings) and read sensitive configuration without logging into the desktop. It's quiet (single command, no GUI), credential-only, and synergizes with service/scheduled-task creation to finish a full remote foothold in one shot.

What the KQL detects:" reg.exe" where cmd has "\breg" AND has any "add", "delete", "copy", "restore", "import", "load", "unload", "save", "export", "query")

IOCs:

- reg.exe commands referencing remote hives \\<host>\HKLM\... or \\<host>\HKU\....
- Write operations: reg add, reg delete, reg import, reg load against remote hives.
- Remote host names in the command string and elevated privileges used.

Remediation if malicious:

- Isolate source host; take registry exports and timeline.
- Identify and revert unauthorized registry writes on target hosts (restore from backup if needed).

FP exclusions: Exclude if the remote host is in ("127.0.0.1", "localhost")

Likelihood of Malicious intent: Medium 🟠 or High 🔴 (high if write et medium if read)

MITRE mapping: T0886, T1021 (Remote Services & Lateral Movement).

5)PowerShell Remoting / WinRM (client-side)

Why / how attackers use it:

PSRemoting/WinRM provides native, encrypted remote shells with object transfer and file copy (Copy-Item -ToSession). It's scriptable and highly parallel, ideal for coordinated lateral jobs (deploy, start, clean). Because it's legitimate admin tech, activity can blend with ops, especially when sourced from IT-like hosts or accounts with delegated rights.

What the KQL detects:

- (powershell.exe or pwsh.exe) where cmd matches has any ("invoke-command", "enter-pssession", "new-pssession") AND includes ("-computername", "-cn", "-hostname", "-session") OR "copy-item" with "ToSession";
- OR "winrs.exe" where cmd has "bwinrs" AND has any ("-r:", "/r:", "\s")

IOCs:

- PowerShell commands with Invoke-Command, New-PSSession, Enter-PSSession, or Copy-Item -ToSession.
- winrs.exe with -r: remote targets.
- Remote hostnames not equal to localhost and sessions created from non-admin workstations.

Remediation if malicious

- Network-isolate the initiating host and capture PowerShell logs and session targets.
- Disable or strictly restrict WinRM on endpoints not requiring it and enforce IPSec/NTLM blocking where possible.

FP exclusions: Exclude if the remote host is in ("127.0.0.1", "localhost")

Likelihood of Malicious intent: Medium 🟠

MITRE mapping: T1021.006 (Remote Services: Windows Remote Management).

6)SUSPICIOUS PROCESSES SPAWNED BY WSMPROVHOST.EXE

Why / how attackers use it:

wsmprovhost.exe is the on-box broker for PSRemoting. When it spawns PowerShell/cmd/mshta/rundll32 with encoded or remote arguments, it's a tell that the host is receiving remote instructions. Attackers prefer this to keep credential use on the controller host, execute inline payloads, and avoid noisy login sessions while fanning out across servers.

What the KQL detects:
• Parent "wsmprovhost.exe" AND child FileName in ("powershell.exe", "pwsh.exe", "cmd.exe", "wscript.exe", "cscript.exe", "rundll32.exe", "regsvr32.exe", "mshta.exe") AND ProcessCommandLine contains abuse tokens ("-enc". "-encodedcommand", "-nop", "-w hidden", "/c", "/http", "/UNC").

IOCs:

- wsmprovhost.exe as parent spawning powershell.exe, cmd.exe, mshta.exe, etc.
- Child command lines containing encoded payloads, remote URLs, or UNC paths.
- Multiple such spawns in a short timeframe.

Remediation if malicious:

- Quarantine targeted host and collect WinRM/wsmprovhost logs plus spawned process details.
- Review WinRM listener configuration, enabled authentication methods and disable unneeded listeners.

Likelihood of Malicious intent: High 🔴

MITRE mapping: T1021.006 (WinRM) / T1047.

7)Remote Scheduled Task (schtasks /S)

Why / how attackers use it:

Remote schtasks gives time-based or immediate execution on target machines without installing services. Adversaries use it to plant persistence, run once to launch an encrypter, or schedule rebuilds (re-establishing tooling after reboots). Pointing /tr at LOLBins, scripts, or writable paths/UNC reduces need for dropping binaries first.

What the KQL detects:

- schtasks.exe/schtasks.com where cmd has "/S <remote>" AND cmd has any of ("/create", "/run", "/change", "/delete")
- AND cmd has "/tr" action and references script/LOLBins or user-writable/remote paths
    - ("ps1", "bat", "cmd", "vbs", "js", "powershell", "wscript", "cscript", "mshta", "rundll32", "regsvr32", "certutil", "bitsadmin", "curl", "ftp", "\\appdata\\","\\programdata\\","\\users\\public\\","\\temp\\","http://","https://","\\\\")

IOCs:

- schtasks.exe with /S <remote> and /create, /run, /change, /delete.
- /tr triggers pointing to scripts in user-writable paths (\AppData\, \Temp\, HTTP/UNC) or to LOLBins.
- Remote host list in command and same pattern across many targets.

Remediation if malicious:

- Isolate initiating host, gather created task definitions and removal timestamps.
- Revoke or rotate credentials used for remote task creation and review scheduled tasks on targeted hosts and remove unauthorized tasks.
- Alert and scan target machines for payloads referenced by /tr.

FP exclusions: Exclude if the remote host is in ("127.0.0.1", "localhost")

Likelihood of Malicious intent: 🔴 High or 🟠 Medium (high if scheduled task created)

MITRE mapping: T1053.005 (Scheduled Task) / T1021.002 (SMB/SCM scheduling).

8)Remote Exec via PowerShell WMI/CIM (Win32_Process.Create)

Why / how attackers use it:

Invoke-WmiMethod/Invoke-CimMethod calling Win32_Process.Create is "PsExec without PsExec", purely API-driven remote process creation over WMI. It keeps the workflow inside PowerShell, enabling inline obfuscation/encoding, credential reuse, and fewer on-disk artifacts, while still achieving code execution at scale across reachable hosts.

What the KQL detects: ("powershell.exe", "pwsh.exe") where cmd matches has any (" invoke-wmimethod", "invoke-cimmethod") AND referencing "win32_process" AND has "create" AND has any ("classname", "class", "name", "methodname")

IOCs:

- PowerShell commands containing Invoke-WmiMethod/Invoke-CimMethod targeting Win32_Process with Create and -ComputerName/-CN arguments.
- Remote host list and use of encoded/obfuscated commands.

Remediation if malicious:

- Isolate and collect PS command logs and remote target list.
- Monitor for Win32_Process.Create calls and block suspicious encoded command patterns.

FP exclusions: Exclude if the remote host is in ("127.0.0.1", "localhost")

Likelihood of Malicious intent: High 🔴

MITRE mapping: T1047 (WMI) / T1021.006 (windows remote management).

9)Remote Execution via PaExec

Why / how attackers use it:

PaExec mirrors PsExec semantics but is less commonly blocked/alerted, so operators swap it in to evade PsExec-specific rules. It keeps the SMB + service creation playbook (quick copy, run, clean), integrates with batch tooling, and offers a drop-in replacement when defenders clamp down on PsExec strings or services.

What the KQL detects: Where the file name is "paexec.exe" OR command line contains "paexec"

IOCs:

- paexec.exe invocation or paexec \\<host> targets in command lines.
- Remote service creation or file drops on remote admin shares following PaExec usage.

Remediation if malicious:

- Block or remove PaExec binaries from endpoints; add detections for paexec usage.
- Rotate credentials used for remote admin access and check for credential theft (Kerberos, NTLM/Hash reuse).
- Audit remote hosts for newly created services or scheduled tasks after PaExec events.

Likelihood of Malicious intent: High 🔴

MITRE mapping: T1059.003 / T1021.001 (SMB/SCM remote execution).

10)Remote Service WRITE/Control via sc.exe

Why / how attackers use it:

sc \\host create/config/start/stop/delete is a native, surgical knob for services on remote systems. Attackers use it to install trojan services pointing to staged binaries (often under ADMIN$/C$), repoint binpaths, or kill defenses/backups. Because it's signed, ubiquitous, and admin-familiar, it routinely slips past coarse allowlists.

What the KQL detects:

- "sc.exe" where cmd has "\bsc" AND has ("\\\\"<remote>) AND has an operation in ("create", "start", "stop", "config", "delete") AND service name present.
- where "Write Operation" OR "High-Risk Service "OR "Bin path" or (not (Allowed Services) and Distinct remotes >= 3)

IOCs:

- sc \\<host> create|config|delete or sc \\<host> start|stop patterns.
- BinPath pointing to user-writable paths, UNC URLs, or unsigned executables and high-risk services (EDR/backup) targeted.

Remediation if malicious:

- Isolate the initiating host and inspect targeted hosts for new/modified services.
- Revert malicious service creations and restore legitimate service binaries from trusted sources.

FP exclusions: Exclude if the remote host is in ("127.0.0.1", "localhost")

Likelihood of Malicious intent: Medium 🟠 or High 🔴 (if operation in ("create", "config", "delete") or suspicious path)

MITRE mapping: T1021.001 (SMB/SCM) / T1543.003 (Create or Modify System Process: Windows Service).

11)SMB ADMIN$ PAYLOAD STAGING (COPY/XCOPY/ROBOCOPY/POWERSHELL)

Why / how attackers use it:

Before remote service/task creation, adversaries must place code on the target. Copying to \\HOST\ADMIN$ or C$ is the simplest universal method (no agents needed). They push EXEs/DLLs/PS1s from user-writable or temp locations (or fetch via HTTP/UNC), then trigger them via SC/Schtasks/WMIC/WinRM, enabling assembly-line lateral deployment.

What the KQL detects:

- Where file name in ("cmd.exe", "powershell.exe", "pwsh.exe", "xcopy.exe","robocopy.exe", "copy.exe","move.exe")
- OR ProcessCommandLine contains ("robocopy"," xcopy", "copy", "move", "copy-item", "new-psdrive" AND cmd contains (UNC \\\\<host>\\(admin$|[cdefgh]$)\\)

IOCs:

- robocopy, xcopy, copy, move, Copy-Item with UNC targets \\<host>\ADMIN$ or \\<host>\C$.
- Commands referencing .exe/.dll/.ps1 or suspicious paths like \AppData\, \temp\, or remote URLs.
- Staging to many hosts or to hosts other than the short device name.

Remediation if malicious:

- Identify and remove staged payloads from Admin$/C$ and examine timestamps and originating accounts.

- Rotate credentials used on affected targets if compromised.
- **Likelihood of Malicious intent:** Medium 🟠 or High 🔴 (High if there are suspicious flags (executable extension, writable path, presence of URL or UNC) or if there are more than 3 different remote hosts)

FP exclusions: Exclude if the remote host is in ("127.0.0.1", "localhost") and if the remote host is equal to the short device name

12) Pass-the-Certificate tooling (PFX/PKINIT)

Why / how attackers use it:

Certificate-based auth (PKINIT) lets actors authenticate as a principal without knowing the password, by presenting a private key/PFX. This enables stealthy TGT acquisition, bypasses password rotation, and often dodges password-centric detections. With Certipy/Rubeus/Kekeo/Impacket, they can mint tickets and traverse domains, then pivot laterally while blending with legitimate certificate flows.

What the KQL detects:

- ("python.exe", "py.exe") where cmd has "\passthecert.py"
- OR Certipy PKINIT auth with PFX: ("certipy.exe", "certipy2.exe") where cmd has all (" auth", "-pfx")
- OR Rubeus PKINIT with certificate: "rubeus.exe" where cmd has "\basktgt\b" AND (cmd has "/certificate:" OR "/pfx:")
- OR Kekeo PKINIT: "kekeo.exe" where cmd has "tgt::ask" AND (has "/pfx" or "/tgtcert")
- OR Impacket/PKINITtools variants: ("python.exe") where cmd has any ("gettgtpkinit.py", "getnthash.py") AND cmd has any ("-cert", "-key")

IOCs:

- Execution of certipy, rubeus with asktgt and /certificate:/ /pfx: flags, or Python scripts wrapping passthecert.py.
- Commands requesting auth with -pfx, -certificate, or using cert-based TGT acquisition.
- Parent processes not signed or run from non-standard paths.

Remediation if malicious:

- Immediately revoke or suspend certificates suspected of compromise and rotate relevant service account certificates.
- Audit AD CS activity and issuance logs for abnormal requests; validate templates and permissions on certificate templates.
- Hunt for lateral logons using certificate-based authentication and correlate with subsequent privileged activity.

FP exclusions: Exclude where parent is unsigned or not in ("cmd.exe","powershell.exe","pwsh.exe")

Likelihood of Malicious intent: High 🔴

MITRE mapping: T1550 – Use Alternate Authentication Material (Pass-the-ticket); T1649 – Steal or Forge Authentication Certificates (as upstream dependency).

Mercuria] - Ransomware Stage 7 - Data Exfiltration

This analytics rule highlights exfiltration toolmarks frequently used just before the destructive stage of ransomware.

Goal: Steal data before encryption.

Aggregation: Events are grouped in 15 minutes bins per device

1) Archive creation with password/encryption (correlated)

Why / how attackers use it:

Attackers stage large data sets into a few encrypted archives to:

- shrink transfer time
- evade inline DLP/content inspection (headers only)
- chunk data for resumable uploads, and
- keep stolen data readable only after ransom

Tools like 7-Zip/WinRAR are ubiquitous and scriptable, making it easy to sweep many folders, respect file filters, and encrypt both payload and filenames before exfil.

What the KQL detects:

- (“7z.exe”, ”7za.exe”, ”7zr.exe”, ”7zg.exe”") where cmd has any (“a”, “u”) AND has any of (“-p*”, “-hp*”, “-mhe”)
- OR (“rar.exe”, “winrar.exe”) where cmd has any of (“a”, “u”) AND has any (“-p*”, “-hp*”)
- AND, within ±10 min, file writes to (“7z”, “zip”, “rar”)
- AND (CountFiles ≥ 3 OR TotalMB ≥ 100)

IOCs:

- 7z/7za/7zr/7zg or rar/winrar with a/u (add/update) plus -p, -hp, or -mhe.
- Multiple archive files created/modified within minutes; large cumulative size (e.g., ≥100 MB) and ≥3 files.
- Parent process unsigned or non-standard (scripts, LOLBins).

Remediation if malicious

- Isolate the source host and preserve volatile data (archive files, paths, timestamps, command lines).
- Kill archiving processes, quarantine produced archives and block egress for the host/account.

- Search enterprise for similar archive patterns and the same parent process across devices.
- Apply DLP rules to block encrypted archive uploads and restrict archiver binaries via WDAC/AppLocker.

Likelihood of Malicious intent: High 🔴

MITRE mapping: T1560.001 (Archive Collected Data), T1074.001 (Local Data Staging).

2) Cloud CLI likely exfil (rclone/azcopy/aws/gsutil/curl/mc/…)

Why / how attackers use it:

Cloud CLIs offer API-grade throughput, retries, and parallelism with simple commands (sync, copy, cp). They can bypass browser-focused controls, run headless under scheduled tasks or PS scripts, and tuck credentials in per-user config files (e.g., rclone.conf). Pointing at public buckets/drives gives global CDN egress and pre-signed URLs for stealthy, authenticated uploads.

What the KQL detects:

- File name in Cloud CLI OR cmd mentions those tools
- AND tool-specific upload semantics from local drive/UNC → remote (e.g., aws s3 cp/sync, azcopy copy, rclone with anay (“copy”, “sync”, “move”), gsutil cp/rsync, megacmd put/sync, mc cp/mirror)
- OR curl/wget with POST/PUT and body/file flags to public/file-sharing/cloud hosts (domain extraction matches known exfil domains list)

Cloud CLI list: rclone.exe, megacmd.exe, azcopy.exe, aws.exe, aws.cmd, gsutil.exe, dbxcli.exe, gdrive.exe, b2.exe, mc.exe, wget.exe, curl.exe, aws_completer.exe)

Exfilration domain list: mega.nz, api.mega.co.nz, wetransfer.com, transfer.sh, anonfiles.com, gofile.io, file.io, dropbox.com, dropboxapi.com, drive.google.com, storage.googleapis.com, box.com, boxcloud.com, onedrive.live.com, 1drv.ms, sharefile.com, files.com, files.com, mediafire.com, mediate.com, sendspace.com, pcloud.com, blob.core.windows.net,storage.cloud.com

IOCs:

- File in cloud CLI list
- Local-to-remote copy semantics in commands; parent process is script/PowerShell or unsigned binary.
- New credentials/config files for these tools under user profiles.

Remediation if malicious:

- Immediately block the host’s outbound to cloud endpoints and suspend the account’s OAuth tokens/API keys.
- Collect tool configs (e.g., rclone.conf), access keys, command histories and rotate/revoke keys.
- Hunt for parallel transfers from other endpoints and notify data owners for potential breach impact.

FP exclusions: Exclude mc profiles that match internal aliases ("dev","uat","prd","prdr","prdw","devr","devw","uatrw") AND exclude ls-only operations

Likelihood of Malicious intent: High 🔴

MITRE mapping: T1567.002 (Exfiltration to Cloud Storage), (secondary) T1041 (Exfiltration Over C2 Channel).

3) HTTP UPLOAD VIA POWERSHELL/CURL

Why / how attackers use it:

Plain HTTP(S) POST/PUT blends into normal web traffic and works from locked-down networks where only 80/443 are open. PowerShell makes file uploads one-liners (Invoke-WebRequest -InFile, UploadFile()), while CURL supports multipart, auth headers, and proxies. Operators favor ephemeral pre-signed links and API tokens to avoid interactive sessions and to script retries.

What the KQL detects:

- (“powershell.exe, “pwsh.exe”) where cmd has any (“Invoke-WebRequest”, “Invoke-RestMethod”) AND has (“-Method POST”, “-Method PUT”) OR has “-InFile” OR has “UploadFile()”
- OR “curl.exe” where cmd has any (“--upload-file”, “-T”, “-f”, “—form”)

IOCs:

- PowerShell Invoke-WebRequest / Invoke-RestMethod with upload parameters; curl with SMTP/HTTP upload flags or multipart forms.
- Destinations are external/public and commands may include tokens, API keys, or pre-signed URLs.

Remediation if malicious:

- Block egress to the destination host(s) and capture full command lines and outbound PCAP/HTTP logs.
- Inspect proxy logs for additional uploads using same URL/API key and invalidate tokens/URLs.

Likelihood of Malicious intent: High 🔴

MITRE mapping: T1041 (Exfiltration Over C2 Channel), T1567.001 (Exfiltration to code repository).

4) SFTP/SCP/FTP TO PUBLIC IP

Why / how attackers use it:

SSH/SFTP/SCP and FTP provide direct pipe-to-internet transfers that sidestep SaaS visibility. Commodity clients (WinSCP, scp, psftp) are portable, automatable, and accept embedded creds/keys, letting actors push bulk archives to rented VPS or bulletproof hosts. Because these protocols often share egress with admin workflows, they can blend with legitimate ops.

What the KQL detects:

- Action “Conection success” on outbound remote port ∈ {22,21,990} AND remote IP is pubic
- AND, within ±5 min, where tool in ("winscp.exe","winscp.com","filezilla.exe","pscp.exe","psftp.exe","scp.exe","sftp.exe","ssh.exe","plink.exe

") AND cmd contains upload/sync verbs (“put”, “mput”, “synchronize remote”, “scp” <src> <user@host:dest>”, “-put”, “-mput”)

- AND parent is script host ("cmd.exe","powershell.exe","pwsh.exe","wscript.exe","cscript.exe","python.exe") or unsigned

IOCs:

- Process executions of transfer tools with upload verbs (put, mput, synchronize remote), correlated with outbound connections to public IPs on 22/21/990 within ±5 minutes.
- Parent is scripting host (cmd, PowerShell, python) or unsigned.

Remediation if malicious:

- Block outbound to the destination Ips, pull client session logs and saved site credentials.
- Reset credentials used and scan the host for staging directories and remove staged data.

Likelihood of Malicious intent: Medium 🟠

MITRE mapping: T1048.003 (Exfiltration Over Unencrypted/Encrypted Non-C2 Protocol).

5) BITS UPLOAD JOB CREATED

Why / how attackers use it:

BITS gives throttled, resilient, background transfers that survive reboots, resume on network changes, and honor proxy/NTLM automatically. Attackers create upload jobs to trickle data off-host with minimal user impact or spikes, often scheduled to quiet hours. Because BITS is trusted OS plumbing, it can evade naive “high bandwidth” alerts.

What the KQL detects:

- bitsadmin.exe where cmd has all (“/transfer, “/upload”)
- OR (powershell.exe|pwsh.exe) where cmd has all (“Start-BitsTransfer”, “TransferType:”, “upload”

IOCs:

- bitsadmin /transfer … /upload or PowerShell Start-BitsTransfer with TransferType: Upload.
- Jobs executing under user context outside software update scenarios.

Remediation if malicious:

- Enumerate and cancel malicious BITS jobs, purge BITS job store and block destination domains.
- Collect BITS job metadata (remote URL, local paths), inspect for staged data and remove.

Likelihood of Malicious intent: High 🔴

MITRE mapping: T1197 (BITS Jobs), T1041 (Exfiltration Over C2 Channel).

6) Browser activity to file-sharing domains (burst)

Why / how attackers use it:

When CLIs are blocked, actors fall back to web UIs (MEGA, WeTransfer, Dropbox, Box, GDrive). Manual or automated browser sessions (Selenium, extensions) can drag-and-drop large folders, leverage chunked uploads, and inherit user SSO. Bursts indicate batch uploads from a staging directory just before encryption or egress blocking.

What the KQL detects:

- Where the initiating process in ("chrome.exe","msedge.exe","firefox.exe","iexplore.exe")
- AND RemoteUrl.Host has any of known file-sharing domains list
- AND Hits ≥ 5 within 10 minutes

Known file-sharing domains list: mega.nz, meganz.cmd, api.mega.co.nz, wetransfer.com, transfer.sh, anonfiles.com, gofile.io, file.io, dropbox.com, dropboxapi.com, drive.google.com, storage.googleapis.com, box.com, boxcloud.com, onedrive.live.com, 1drv.ms, sharefile.com, files.com, mediafire.com, sendspace.com, pcloud.com

IOCs:

- Many connections (burst) from browsers (chrome.exe, msedge.exe, etc.) to domains like MEGA, WeTransfer, Dropbox, Box, Google Drive, OneDrive personal, etc.
- Activity outside user's normal pattern/hours and large upstream bandwidth.

Remediation if malicious:

- Review secure web gateway logs and block or throttle consumer file-sharing domains for non-sanctioned use.
- Revoke access.

Likelihood of Malicious intent: Medium 🟠

MITRE mapping: T1567.001 (Exfiltration to Web Service).

7) CLI-based SMTP email exfil

Why / how attackers use it:

Scripted SMTP sends archives as attachments straight to attacker mailboxes or relays, bypassing corporate mail/DLP routes. Tools like blat, sendemail, smtp-cli, or raw .NET SmtpClient allow credential embedding, TLS, and quiet batching from headless jobs. Good for smaller but sensitive sets (keys, finance docs) separate from bulk exfil.

What the KQL detects:

- ("powershell.exe", "pwsh.exe") where cmd has "Send-MailMessage" with "-Attachments" OR cmd has any ("new-object\s+net\.mail\.smtpclient", "system\.net\.mail\.smtpclient") with attachment indicators

- OR ("curl.exe") where cmd targets “smtp://|smtps://” AND has “--mail-from” AND “--mail-rcpt” AND has any (“--upload-file”, “-T”)
- OR ("blat.exe","blat64.exe", “sendemail.exe") where cmd has “-to” AND has any ("-att", "-attach", “-a”)
- OR ("mailsend.exe") where cmd has “-to” AND has any ("-file", "-attach", “-stdin”)
- OR ("smtp-cli.exe ") where cmd has all ("--server”, “--to) AND has any (“--data”, "--file", “-stdin”)
- OR (""swaks.exe") where cmd has all ("--server”, “--to) AND has any ("--attach", “--data", “--body”)

IOCs:

- Send-MailMessage with -Attachments; raw .NET SmtpClient usage with .Attachments.Add().
- CLI tools (blat, sendemail, smtp-cli, swaks, curl smtp://…) with --attach/-T/--upload-file.
- External SMTP servers and credentials embedded in scripts.

Remediation if malicious:

- Identify and revoke credentials used; inspect mail relay logs for matching recipients/subjects.
- Restrict PowerShell mail cmdlets via execution policy/WDAC.

Likelihood of Malicious intent: Medium 🟠

MITRE mapping: T1048.003 (Exfiltration Over Non-C2 Protocol).

8) Mass Download via SharePoint/OneDrive

Why / how attackers use it:

Before leaving or encrypting, actors mirror team sites and personal drives using non-sync user agents or scripted REST calls. This preserves document structure/metadata and yields clean archives for leak sites. Doing it en masse reduces time on target and front-loads extortion leverage (proof of theft) even if encryption later fails.

What the KQL detects:

- CloudAppEvents where application in ("Office 365 SharePoint Online","Microsoft OneDrive for Business") AND Action is “FileDownloaded”
- AND extension in ("zip", "7z", "rar", "pst", "bak", "sqlite", "mdb", "csv", "xlsx", "xls", "doc", "docx", "ppt", "pptx","pdf")
- AND (DownloadCount ≥ 100) AND (DistinctFiles ≥ 100 OR DistinctUAs ≥ 1) (within 10 min)

IOCs:

- High counts of FileDownloaded in SPO/OneDrive from non-client user agents (not the official OneDrive/Office clients).
- Spikes across sensitive extensions (archives, PST, office docs) and unusual IPs/agents.

Remediation if malicious

- Temporarily suspend the user account or require step-up auth and block risky sessions in the IdP/CASB.

- Apply conditional access, egress controls, and DLP for mass download patterns and notify data owners.

FP exclusions: Exclude user agent that are not typical Office/OneDrive clients, not in ("OneDrive", "SkyDrive", "Microsoft Office", "MSOffice", "Excel", "Word", "PowerPoint", "Upload Center", "Office16")

Likelihood of Malicious intent: Low 🟡

MITRE mapping: T1530 (Data from Cloud Storage), (secondary) T1074 (Data Staging).

9) USB Exfiltration Suspected (post-mount writes)

Why / how attackers use it:

When egress is tight or monitored, removable media is a simple, offline path. Attackers (or coerced insiders) mount a device, bulk-copy office docs, databases, and keys, and walk the data out. Large write bursts right after mount signal staging → physical export, often coordinated with EDR/AV tampering to avoid prompts.

What the KQL detects:

- Action is "UsbDriveMounted" (captures device details)
- AND, within +2h, action is "FileCreated" AND FileSize ≥ 1 MB AND extension in a sensitive list
- AND threshold (FilesWritten ≥ 10 OR TotalBytes ≥ 200 MB)

Sensitive list : docx, xlsx, pptx, pdf, csv, txt, rtf, zip, 7z, rar, pst, ost, bak, db, sqlite, mdb, accdb, sql, pem, ppk, pfx, key, crt, conf, json, yml, yaml)

IOCs:

- UsbDriveMounted followed by ≥10 file writes or ≥200 MB (often much more) to the mounted drive within ~2 hours.
- Sensitive file extensions (docs, spreadsheets, archives, databases, keys/certs).
- Device details (vendor/product) captured at mount.

Remediation if malicious:

- If policy prohibits, disable the USB storage device class via GPO/MDM and collect mount metadata and copy lists.
- Confiscate or image the removable media (if available) and inventory other endpoints for the same device IDs.

Likelihood of Malicious intent: Medium 🟠 or High 🔴 (high if ≥1 GB or ≥200 files)

MITRE mapping: T1052.001 (Exfiltration Over Physical Medium: Removable Media).

[Mercuria] - Ransomware Stage 8 - Destructive Behavior

This analytics rule surfaces destructive behavior patterns typically seen during the execution phase of ransomware and wipers.

Goal: Encrypt, destroy or disrupt systems.

Aggregation: Events are grouped in 15 minutes bins per device

1) HIGH-VOLUME FILE ACCESS (I/O BURST)

Why / how attackers use it:

Encryptors and wipers walk directory trees fast, opening/reading and then modifying/renaming files in tight loops. They prioritize user shares and working sets (Documents, OneDrive, %TEMP%, ProgramData, UNC shares) to maximize business impact and leverage multi-threading + async I/O to saturate local disks and SMB. Renames (temp → encrypted) and rapid metadata changes are a by-product of chunked encryption and progress checkpoints; wipers will often mass-modify/rename before deletion to destroy recovery context.

What the KQL detects:

- Action in ("FileModified","FileRenamed") AND file extension in sensitive list
- AND (account ≠ SYSTEM OR path in suspicious file indicator for system list)
- AND initiator NOT in common OS/installer allowlist (e.g., ccmexec, tiworker, msiexec, trusted svchost)
- Aggregate in 5 min bins where Events ≥ 400

File extension sensitive list: doc, docx, xls, xlsx, ppt, pptx, pdf, txt, csv, rtf, zip, 7z, rar, tar, gz, tgz, 7zip, pst, ost, bak, db, sqlite, mdb, accdb, sql, json, xml, yml, yaml, cfg, conf, ini, pem, ppk, pfx, key, crt, cert

Suspicious file indicator for system list: \users\, \documents and settings\, \users\public\, \programdata\, \inetpub\wwwroot\, \onedrive\, \shares\, \downloads\, \appdata\local\temp\, \appdata\local\microsoft\windows\inetcache\, \users\default\, \temp\, \$recycle.bin\, c:\\windows\\ccm\\cidownloader\\staging\\

IOCs:

- Hundreds of FileModified / FileRenamed within ~5 minutes, targeting docs, archives, DBs, keys.
- Activity under user-writable or shared paths (\Users\, \ProgramData\, \Shares\, %TEMP%, OneDrive sync folders).
- Non-system parents (or system with suspicious file indicator).

Remediation if malicious:

- Immediately isolate the endpoint(s) and pause SMB/DFS access for affected user accounts.
- Kill offending processes, capture memory and EDR timeline, preserve $MFT/usn journal where possible.
- Block the account's Kerberos/NTLM tokens and rotate creds used by the process owner and any mapped drives.
- Snapshot impacted file servers, disable client-side caching (if applicable) and start restore-point creation.
- Roll out application control (WDAC/AppLocker/CrowdStrike/Defender) to block unknown encryptors.

FP exclusions: Already excludes several OS/installer processes and clean svchost patterns

Likelihood of Malicious intent: Low 🟡

MITRE mapping: T1486 (Data Encrypted for Impact), T1491.001 (internal Defacement), T1561 (Disk Wipe, contextually).

2) Mass File Encryption Detected

Why / how attackers use it:

Modern families append branded extensions/IDs (e.g., *.id[xxx].enc) to simplify ransom ops (victim tracking, decryptor matching) and deter casual recovery. Double-extensions (docx.enc) and folder-wide surges let operators measure coverage and throttle per host. Burst-style encryption across many directories reduces the chance blue teams can isolate in time and signals automated spread from a single process or coordinated wave.

What the KQL detects:

- Where action in ("FileCreated","FileModified","FileRenamed") AND filename matches:
  - where extension in ransom extensions list
  - OR file name has ".id[...]" marker
  - OR double-extension pattern like docx.enc, xlsx.locked, pptx.encrypted, txt.crypt,
- Where account ≠ SYSTEM OR path in suspicious file indicator for system list
- Where encrypted file ≥ 40 OR (≥ 20 AND distinct path ≥ 10) within 10 min

Suspicious file indicator for system list: \users\, \documents and settings\, \users\public\, \programdata\, \inetpub\wwwroot\, \onedrive\, \shares\, \downloads\, \appdata\local\temp\, \appdata\local\microsoft\windows\inetcache\, \users\default\, \temp\, \$recycle.bin\, c:\\windows\\ccm\\cidownloader\\staging\\

Extension in ransom extensions list: locked, encrypted, crypt, crypted, cryp1, cry, enc, krab, locky, egregor, conti, ryuk, ransom, mallox, leak, pay, phobos, hive, blackcat, akira, noescape)

IOCs:

- New extensions like enc, encrypted, locky, cry, krab, or *.id[xxxxx].
- 20–40+ files across 10+ directories within minutes; unsigned or unusual parent processes.

Remediation if malicious:

- Isolate hosts, block user/service accounts and disable interactive logons.
- Pull known bad filenames/paths and hash lists and push global block rules via EDR.
- Stop scheduled tasks/services created shortly before the burst and remove persistence if needed.
- Restore from clean backups and verify integrity before reconnecting to domain/shares.

FP exclusions: exclude .lnk extension

Likelihood of Malicious intent: High 🔴

MITRE mapping: T1486 (Data Encrypted for Impact).

3) Drive Wipe via cipher.exe

Why / how attackers use it:

Post-exfil/encryption, actors run cipher /w to overwrite free space, erasing previous file bodies and carved remnants so responders can't recover staging files or earlier data versions. It's a signed Windows binary (LoLBin), blends into admin scripts, and doesn't need extra tooling, ideal for anti-forensics on endpoints and file servers right before/after impact.

What the KQL detects: Where file name is "cipher.exe" AND command line has "/w"

IOCs:

- cipher.exe /w:<path> runs, often from script shells (PowerShell/cmd)
- Unsigned or suspicious parents.

Remediation if malicious:

- Isolate host, terminate cipher.exe and capture volatile artifacts and remaining shadow copies immediately.
- Block cipher.exe execution for users via WDAC/AppLocker.
- Hunt for preceding exfil/encryption on the same host and operator activity (RDP/WinRM/WMI).

Likelihood of Malicious intent: High 🔴

MITRE mapping: T1070.004 (Indicator Removal: File Deletion), T1561.001 (Disk Wipe: Disk Content Wipe).

4) Backup Deletion via wbadmin.exe

Why / how attackers use it:

Killing on-host backups raises leverage and extends downtime. Deleting catalogs/versions ensures even if shadow copies or EDR rollback survive, official recovery points vanish. Attackers like wbadmin because it's built-in, privileged, and scriptable, so it fits well in pre-impact runbooks alongside VSS deletions and service stops (e.g., SQLWriter, VSS, wbengine).

What the KQL detects:

- Where file name is “wbadmin.exe” AND cmd has any (“backup”, “catalog”, “systemstatebackup”, “version(s)”)
- Aggregate per 10 min where DeleteCount > 0

IOCs:

- wbadmin delete backup|catalog|systemstatebackup|version with administrative parents
- Bursts before/after shadow copy deletion.

Remediation if malicious:

- Revoke admin tokens, isolate host and suspend any scheduled backup jobs to prevent poisoning.
- Lock backup infrastructure credentials and verify offsite/immutable copies (object lock/WORM) are intact.

Likelihood of Malicious intent: High 🔴

MITRE mapping: T1490 (Inhibit System Recovery).

5) Boot Configuration Tampering

Why / how attackers use it:

Manipulating BCD/boot settings (e.g., recoveryenabled no, ignoreallfailures) breaks safe recovery paths, suppresses repair prompts, and can create boot loops or prevent BitLocker interlocks from intervening. Combined with MBR/EFI changes or recovery partition damage, it complicates incident response and stretches RTOs, pressuring payment.

What the KQL detects:

- Where file name in ("bootrec.exe","bcdedit.exe","bootcfg.exe")
- Where cmd has any of: (“fixboot”, “fixmbr”, “rebuildbcd”, “/rebuildbcd”, “recoveryenabled no”, “bootstatuspolicy ignoreallfailures”, “safeboot minimal” OR cmd has bcdedit AND has any (“/delete”, “/export”, “/import”, “/set”) OR cmd has bootcfg and has any (“delete”, “default”)

IOCs:

- bcdedit with recoveryenabled no, bootstatuspolicy ignoreallfailures, /delete|/export|/import|/set.
- bootrec fixmbr/fixboot/rebuildbcd, bootcfg /delete|/default.

Remediation if malicious:

- Isolate, suspend BitLocker auto-unlock where applicable and secure recovery keys.
- Block bcdedit/bootrec/bootcfg for users via WDAC/AppLocker, if needed.
- Verify Secure Boot, TPM policies, MBAM/BitLocker escrow and prepare offline recovery media.

Likelihood of Malicious intent: High 🔴

MITRE mapping: T1490 (Inhibit System Recovery), T1491.001 (Service Stop/boot impact, contextual).

6) Destructive Wipe Utility Executed (sdelete/shred/wipe/eraser)

Why / how attackers use it:

Third-party wipers provide explicit pass counts, recursive modes, and quiet flags to systematically shred evidence (staging dirs, logs, tooling) and sometimes sabotage data at scale. Because these tools are common in IT/IR contexts, actors piggyback their operational cover while benefiting from predictable erase semantics across local/remote volumes.

What the KQL detects:

- Where file name has any ("sdelete.exe", "shred.exe", "wipe.exe", "eraser.exe", "securewipe.exe", "nwipe.exe")
- AND cmd shows wipe/recursion/force indicators (" -p ", " -n ", " -s ", " -r ", " /p ", " /r ", " --recursive", "--zero", "/s ", "/p:", "/p=") or cmd has any (" /force", " /all", " /quiet", " -f")

IOCs: Execution of sdelete.exe, shred.exe, wipe.exe, etc., with recursive / pass flags (/p, /r, --recursive, /all, /quiet).

Remediation if malicious

- Kill processes, collect command lines and target paths and image disk ASAP (before more overwrites).
- Hunt for precursors (exfil scripts, encryptor dropper) and for follow-on wiping on servers/NAS.

Likelihood of Malicious intent: High 🔴

MITRE mapping: T1561.001 (Disk Wipe: Disk Content Wipe), T1070 (Indicator Removal).

7) BitLocker Protection Suspended/Protector Removed

Why / how attackers use it:

Suspending BitLocker or removing protectors downgrades disk-at-rest security, allowing offline reads/writes, easier bootloader tampering, and unprotected re-imaging of system volumes. It also weakens post-incident containment (e.g., hot-plugged disks) and may bypass recovery prompts, making later wipes or driver-level payloads more reliable.

What the KQL detects:

- "manage-bde.exe" where cmd has any ("-protectors -disable", "-protectors -remove", " -protectors -removeall", " -protectors -adbackup")
- OR ("powershell.exe", "pwsh.exe") where cmd has any ("suspend-bitlocker", "remove-bitlockerkeyprotector") OR cmd has all ("get-bitlockervolume", " -protectors ")
- OR cmd has any ("manage-bde -protectors -disable", "manage-bde -protectors -remove")

IOCs: manage-bde -protectors -disable/-remove or PowerShell Suspend-BitLocker / Remove-BitLockerKeyProtector.

Remediation:

- Immediately re-enable protectors if legitimate activity not confirmed and rotate recovery keys.
- Restrict BitLocker cmdlets to device-management pipelines.
- Verify no offline boot/USB boot occurred and check for boot policy changes and credential theft on the host.

Likelihood of Malicious intent: High 🔴

MITRE mapping: T1490 (Inhibit System Recovery).

8) Remote share I/O burst (UNC)

Why / how attackers use it:

Once credentials land, actors encrypt/wipe file servers over SMB from a compromised client/jump box. Working over UNC paths avoids interactive logons, leverages existing access tokens, and lets a single host fan out across many shares quickly. Massive modify/rename/delete spikes reflect parallelized worker threads pushing impact to centralized data stores where pain is largest.

What the KQL detects:

- Where folder path starts with UNC paths (\\...) AND action is in ("FileCreated","FileModified","FileRenamed","FileDeleted")
- AND initiator in suspicious parent list
- Where IO_Count ≥ 1000 OR DelCount ≥ 300 within 5 min

Suspicious parent list: cmd.exe, powershell.exe, pwsh.exe, wscript.exe, cscript.exe, mshta.exe, rundll32.exe, 7z.exe, winrar.exe, explorer.exe

IOCs:

- 5-minute bursts with 1,000+ I/O ops or 300+ deletes on \\HOST\SHARE\....
- Parents are in the suspicious parent list.

Remediation if malicious:

- Quarantine the client and the targeted share server, block the user account and disable share access temporarily.
- Snapshot volumes immediately, stop VSS tampering and preserve logs (SMB, DFS, NAS audit).
- Roll back affected folders; enforce least-privilege on shares (deny write for broad groups) and enable ransomware-aware snapshots on NAS/SAN.

FP exclusions: Exclude when UNC path not in ("SYSVOL", "/NETLOGON", "/PRINT", "IPC") AND when folder path has any ("\Library\", "_Python\", "_Anaconda_\", "\site-packages\")

Likelihood of Malicious intent: Medium 🟠

MITRE mapping: T1486 (Data Encrypted for Impact), T1565.001 (Stored Data Manipulation), T1490 (Inhibit System Recovery).