Investigation Report – Suspicious Activity on `gab-intern-vm`

1. Executive Summary

Between **1–15 October 2025**, suspicious activity was observed on endpoint **`gab-intern-vm`**, an intern-operated workstation. A structured threat hunt was conducted using Log Analytics (KQL) to identify potential **initial execution**, **reconnaissance**, **staging**, **egress validation**, and **persistence mechanisms**, along with a **planted cover narrative**.

The investigation confirmed that:

- The host was used to **execute custom tooling from the Downloads directory**.
- The actor performed **host, user, and storage reconnaissance**, and tested **outbound connectivity**.
- Collected information was **staged into a single archive** (`ReconArtifacts.zip`) under `C:\Users\Public\`.
- **Simulated exfiltration / outbound transfer attempts** were identified, including a final unusual outbound connection to IP **`100.29.147.161`**.
- **Two forms of persistence** were implemented:
    - A **scheduled task** (`SupportToolUpdater`) configured to re-execute tooling at logon.
    - A **user-scope autorun registry value** (`RemoteAssistUpdater`).
- A **planted "support chat"–style artifact** (`SupportChat_log.lnk`) was created to provide a misleading narrative for the activity.

Overall, the activity is consistent with a **hands-on attacker or red-team simulation** focused on host recon, proof-of-access, and establishing resilient re-execution paths on `gab-intern-vm`.

---

2. Scope & Environment

- **Endpoint investigated:** `gab-intern-vm` (intern workstation)
- **Time window: 2025-10-01T00:00:00Z** to **2025-10-16T00:00:00Z**
- **Data sources:**
    - `DeviceProcessEvents`
    - `DeviceFileEvents`
    - `DeviceNetworkEvents`
    - `DeviceEvents` (task & registry–related telemetry)
- **Tools used:**
    - Azure Log Analytics (KQL)
    - Analyst-driven pivoting between processes, files, network, and scheduled tasks

---

3. Methodology

The hunt followed a **stage-based approach**, aligning with typical attack chains:

1. **Entry & Initial Execution** – Identify suspicious executions originating from user-controlled locations (Downloads), especially with unusual command-line parameters.
2. **Reconnaissance & Context Gathering** – Detect rapid, script-driven checks of clipboard, host info, user sessions, and storage surfaces.

3. **Defense / Tamper Simulation** – Identify artifacts related to security posture or tamper indicators.

4. **Staging & Egress** – Locate consolidation of artifacts (archives) and outbound connectivity checks or transfer attempts.

5. **Persistence** – Hunt for both **scheduled tasks** and **autorun registry keys**.

6. **Narrative / Cover Artifacts** – Identify user-facing files seemingly created to explain or legitimize the activity.

---

4. Key Findings by Phase

4.1 Initial Execution & Entry Point

- Activity originated from the **Downloads** directory on `gab-intern-vm`, matching the initial scenario description.

- The first suspicious program execution included a **CLI parameter name** indicating logging behavior:

    - **First CLI parameter name used: -ExecutionPolicy**

- This parameterized execution served as the logical **anchor point** for subsequent process trees and activity chains.

4.2 Defense Tampering Indicator

- A file named **`DefenderTamperArtifact.lnk`** was identified as a **planted tamper indicator**.

- The file appears designed to **simulate or imply Defender tamper activity** without necessarily reflecting a true configuration change.

- This aligns with the objective of **staging artifacts to suggest changes in security posture** while primarily signaling attacker intent.

4.3 Quick Data Probe – Clipboard

A short-lived PowerShell command was observed:

```
"powershell.exe" -NoProfile -Sta -Command "try { Get-Clipboard | Out-Null } catch { }"
```

- This represents a **low-effort, quick probe** for clipboard data.
- It matches the behavior of an actor looking for **easy, transient data wins** before investing in broader collection.

4.4 Host & User Context Reconnaissance

- Multiple **reconnaissance-oriented commands and scripts** were identified, including:
    - Host / environment context gathering.
    - Session / user enumeration.
- A key recon event was timestamped at:
    - **Last recon attempt:** `2025-10-09T12:51:44.3425653Z`
- These actions inform the attacker's understanding of **who is logged in, what's running, and what to target next**.

4.5 Storage Surface Mapping

- PowerShell activity was observed that **enumerated storage locations and/or available drives**, consistent with:
    - Assessing local and possibly networked storage surfaces.

- This supports the attacker's preparation to **identify where data of interest may reside** before staging or exfiltration.

4.6 Connectivity & Name Resolution

- Connectivity checks were observed via system processes including:
    - **Parent process for connectivity checks:** `RuntimeBroker.exe`
- These actions confirm **name resolution and outbound network reachability**, a prerequisite for future exfiltration or C2.

4.7 Process & Session Enumeration

- **Process inventory activity** was observed, including:
    - Execution of **`tasklist.exe`** to enumerate running processes.
- Interactive session discovery identified:
    - A unique initiating process identifier:
        - **Initiating Process Unique ID:** `2533274790397065`
- These actions help the attacker understand **which applications and sessions are active** on the host.

4.8 Privilege Surface Check

- Telemetry indicated **privilege and group membership checks**, consistent with mapping the current user's capabilities.
- The **very first privilege check attempt** was observed at:
    - **`2025-10-09T12:52:14.3135459Z`**

- This guides the attacker's decision to either proceed with available rights or to attempt privilege escalation.

4.9 Egress Validation & Proof of Access

- The actor validated outbound reachability and demonstrated effective access to host data:
    - Outbound connectivity test to:
        - `www.msftconnecttest.com`
    - Creation of consolidated recon artifacts:
        - `C:\Users\Public\ReconArtifacts.zip`
- Together, these actions show:
    - **"I can reach out"** (egress works), and
    - **"I can collect & package host data"** (archive ready for transfer).

4.10 Outbound Transfer Attempt

- Analysis of `DeviceNetworkEvents` highlighted **unusual outbound connections** associated with simulated upload/transfer attempts.
- The **last unusual outbound connection IP** was identified as:
    - `100.29.147.161`
- Even if the transfer was simulated or blocked, this reflects **explicit exfiltration intent** and reveals an egress path.

4.11 Persistence – Scheduled Re-Execution

- A **scheduled task** was created to ensure tooling could re-execute at logon:
    - **Task Name:** `SupportToolUpdater`
    - Evidence showed creation via:
        - `schtasks.exe /Create /SC ONLOGON /TN SupportToolUpdater ...`
- This provides **logon-triggered persistence**, allowing the actor's tooling to survive beyond a single session.

4.12 Persistence – Autorun Fallback (User Scope)

- A **user-scope autorun registry value** was identified as a fallback persistence mechanism:
    - **Registry Value Name:** `RemoteAssistUpdater`
- This likely points to an executable or script that will re-launch at user logon/startup.
- Even if the scheduled task is removed, this autorun provides **redundant persistence**, increasing resilience.

4.13 Planted Narrative / Cover Artifact

- A user-facing artifact was identified in the Downloads folder:
    - **File:** `SupportChat_log.lnk`
- This shortcut appears designed as a **"support chat log" narrative**, likely to:
    - Provide a **plausible cover story** for the observed activity.

    - Mislead casual reviewers into believing actions were part of legitimate support operations.

- The **timing and context** of its creation (shortly after suspicious activity) strongly suggest it is a **deliberate misdirection artifact**.

---

5. Assessment

The investigation confirms that `gab-intern-vm` was used in a **structured, multi-stage operation** resembling an attacker or red-team simulation:

- **Initial execution** from user-controlled locations (Downloads) with custom parameters.
- **Defense signaling**, via tamper-style artifacts.
- **Reconnaissance** against clipboard, host context, processes, sessions, privileges, and storage.
- **Staging** of collected data into `ReconArtifacts.zip`.
- **Egress validation and simulated exfiltration** via unusual outbound connections, including to `100.29.147.161`.
- **Resilient persistence**, achieved via both a scheduled task (`SupportToolUpdater`) and an autorun registry value (`RemoteAssistUpdater`).
- **Narrative management**, by leaving behind `SupportChat_log.lnk` to justify or obscure the activity.

The pattern is consistent with a **deliberate, persistent actor** validating access, survivability, and outbound paths on a lower-privilege intern workstation.

---

6. MITRE ATT&CK Mapping Summary

Initial Access / Execution

- **User Execution – Malicious or Untrusted File**
  - **T1204.002 – Malicious File**
  - Tooling is executed from the **Downloads** folder on an intern workstation (`gab-intern-vm`), using suspicious command-line parameters (e.g., **`-ExecutionPolicy`**). This matches user-driven execution of potentially malicious content.
- **Command & Script Interpreter – PowerShell**
  - **T1059.001 – PowerShell**
  - Multiple actions are driven via `powershell.exe` (clipboard probe, recon, storage mapping, artifact creation, staging, etc.).

Defense Evasion

- **Impair Defenses – Staged Tamper Artifacts**
  - **T1562.001 – Disable or Modify Tools** (simulated / signaled)
  - Creation of `DefenderTamperArtifact.lnk` is a *planted indicator* suggesting Defender tampering. Even if no real config change occurred, the artifact is clearly meant to imply interference with security tooling.

Discovery

The scenario is heavy on discovery / recon:

- **Clipboard Data Discovery**
    - **T1115 – Clipboard Data**
    - PowerShell command:
      `Get-Clipboard | Out-Null`
    - Represents probing of transient clipboard data for easy wins.
- **System Information & Host Context Discovery**
    - **T1082 – System Information Discovery**
    - Commands that gather host and environment context (OS, system details, etc.) to understand where the actor landed.
- **Process Discovery**
    - **T1057 – Process Discovery**
    - Use of `tasklist.exe` to enumerate running processes and understand what is active on the host.
- **Account / Session Discovery**
    - **T1033 – System Owner/User Discovery** and/or **T1087 – Account Discovery**
    - Activity to identify logged-in users and interactive sessions, including the initiating process unique ID `2533274790397065` used in session-related recon.
- **Logon Session / Interactive Session Discovery**

    - **T1078 / T1069-ish behaviors** (Account/Session enumeration)
    - Actor checks which sessions are active, helping decide whether to act immediately or wait.
- **File and Directory / Storage Discovery**
    - **T1083 – File and Directory Discovery**
    - **T1005 – Data from Local System** (as preparation)
    - Storage and filesystem probing, plus drive/space assessment (storage surface mapping) to locate likely data repositories.

Collection & Staging

- **Local Data Collection & Aggregation**
    - **T1074.001 – Local Data Staging**
    - Collection of multiple artifacts:
        - `Support_701.txt`
        - `Update_904.txt`
        - `Helpdesk_247.txt`
    - Followed by consolidation into:
        - `C:\Users\Public\ReconArtifacts.zip`
    - This is classic **local staging** prior to (attempted) exfiltration.

Command and Control / Egress Validation

- **Application Layer Protocol – Web**
    - **T1071.001 – Web Protocols**
    - Outbound connectivity checks using HTTP/HTTPS endpoints, including:
        - `www.msftconnecttest.com` (connectivity test)
        - Other web destinations and unusual IPs.
- **Traffic Signaling / Egress Test**
    - **T1016 / T1041 (as preparation)**
    - The activity validates that outbound connections (egress) are functional before any real data movement, aligning with **"Proof-of-Access & Egress Validation"** behavior.

Exfiltration

- **Exfiltration Over C2 or Web Service (Simulated Upload)**
    - **T1041 – Exfiltration Over C2 Channel** / **T1567.002 – Exfiltration to Cloud or Web Service** (conceptually)
    - Simulated outbound transfer attempts, ending with a **last unusual outbound connection** to:
        - **`100.29.147.161`**
    - Even if blocked or only staged, the pattern reflects **intent to move data off-host**.

Persistence

Two clear persistence mechanisms were identified:

- **Scheduled Task / Job**
    - **T1053.005 – Scheduled Task**
    - Creation of a logon-triggered scheduled task:
        - **Task Name:** `SupportToolUpdater`
    - Ensures tooling can re-run on user logon, providing **re-execution persistence**.
- **Registry Run Key / Autorun**
    - **T1060 / T1547.009 – Registry Run Keys / Startup Folder**
    - User-scope autorun registry value:
        - **Registry Value Name:** `RemoteAssistUpdater`
    - Acts as a **fallback persistence** mechanism if other methods are cleaned up.

Defense / Narrative Management

- **Masquerading / User-Facing Cover Artifacts**
    - **T1036 – Masquerading** (in a narrative sense)
    - Creation of **`SupportChat_log.lnk`** in the user's Downloads area:
        - Designed as a **"support chat log"** cover story.

- Provides a **misleading narrative** to justify or obscure the true nature of prior activity.

High-Level ATT&CK Chain (Narrative)

Summary:

1. **Execution from Downloads using PowerShell** → (*T1204.002, T1059.001*)
2. **Recon of clipboard, host, processes, sessions, storage, and privileges** → (*T1115, T1082, T1057, T1033/T1087, T1083*)
3. **Simulated Defender tamper artifact** → (*T1562.001*)
4. **Staging of collected data into `ReconArtifacts.zip`** → (*T1074.001*)
5. **Outbound connectivity checks & unusual egress (e.g., `100.29.147.161`)** → (*T1071.001, T1041/T1567.002*)
6. **Persistence via scheduled task `SupportToolUpdater` and autorun `RemoteAssistUpdater`** → (*T1053.005, T1547.009*)
7. **Cover story via `SupportChat_log.lnk`** → (*T1036 – Masquerading / narrative artifact*)

---

7. Who / What / Where / When / Why

Who

- **Endpoint:** `gab-intern-vm`
- **User context:** Intern-operated, likely lower-privileged but network-connected.
- **Actor assessment:**

    - Behavior strongly resembles a **controlled red-team / lab / CTF scenario**.
    - In a real environment, the same pattern would map to an **interactive attacker** conducting hands-on keyboard activity.

What

The actor executed a compact end-to-end attack chain on **gab-intern-vm**, including:

- **Initial execution** from the **Downloads** directory using suspicious parameters (e.g., **-ExecutionPolicy**).
- **Defense-tamper signaling** via `DefenderTamperArtifact.lnk`.
- **Reconnaissance**:
    - Clipboard probing via `Get-Clipboard`.
    - Host/user context, processes, sessions, privileges.
    - Storage surface / filesystem mapping.
- **Data staging**:
    - Multiple artifacts:
        - `Support_701.txt`
        - `Update_904.txt`
        - `Helpdesk_247.txt`
    - Consolidated into:
        - `C:\Users\Public\ReconArtifacts.zip`
- **Egress tests & simulated exfil**:

    - Connectivity check: `www.msftconnecttest.com`
    - Final unusual outbound IP: `100.29.147.161`
- **Persistence**:
    - Scheduled task: `SupportToolUpdater`
    - Autorun registry value: `RemoteAssistUpdater`
- **Narrative management / cover**:
    - `SupportChat_log.lnk` – a shortcut positioned as a “support chat log” to provide plausible cover.

Where

- **Primary system:** `gab-intern-vm`
- **Key locations:**
    - `C:\Users\g4bri3intern\Downloads\` – initial tooling & cover artifacts.
    - `C:\Users\g4bri3intern\Documents\HelpdeskLogs\` – log-like files (e.g., `Support_701.txt`).
    - `C:\Users\Public\ReconArtifacts.zip` – main data staging archive.
    - Registry and Task Scheduler – used for persistence configuration.

When

- **Investigation window:** `2025-10-01` to `2025-10-15` (UTC).

- **Main activity cluster:** ~`2025-10-09`, approximately **12:50–13:10Z**.

Example key timestamps:

- Recon activity: `2025-10-09T12:51:44.3425653Z`
- First privilege / “who am I” check: `2025-10-09T12:52:14.3135459Z`
- Staging, connectivity checks, exfil simulation, and persistence events occur shortly thereafter.

Why (Assessment)

The pattern indicates an actor aiming to:

- **Validate access and capabilities** on an intern endpoint.
- **Understand the environment:** who is logged in, what is running, what data exists, and what privileges are available.
- **Prove collection capability** by assembling data into `ReconArtifacts.zip`.
- **Verify outbound connectivity** and explore potential exfil paths.
- **Persist beyond a single session** (scheduled task + autorun registry value).
- **Control the narrative** with a planted “support chat log” (`SupportChat_log.lnk`) to justify suspicious actions.

In a real network, this represents **pre-exfiltration and pre-lateral-movement positioning** combined with **durable persistence**.

---

8. Methodology

Data Sources

- `DeviceProcessEvents`
- `DeviceFileEvents`
- `DeviceNetworkEvents`
- `DeviceEvents` (scheduled tasks, registry & autorun-related telemetry)

Approach

1. **Host scoping**
    - Identify the most suspicious machine based on **Downloads activity** and **file naming patterns** → `gab-intern-vm`.
2. **Timeline anchoring**
    - Use earliest suspicious **Downloads-based execution** to anchor the investigation.
3. **Process–file–network pivoting**
    - Initial process execution → file artifacts → network behavior.
4. **Persistence hunting**
    - Examine scheduled tasks and autorun registry values referencing “support/helpdesk/tool” naming patterns.
5. **Narrative artifact hunt**
    - Look for user-facing `.txt` / `.lnk` files (e.g., support or helpdesk logs) opened near the time of suspicious operations.

Key Techniques

- Filtering by `DeviceName == "gab-intern-vm"`.

- Narrowing time ranges around key timestamps.
- Parsing `AdditionalFields` JSON (HTTP method, task names, registry/autorun details).
- Using `InitiatingProcessId`, `InitiatingProcessFileName`, and `RemoteIP/RemoteUrl` to correlate stages.

---

9. Queries, Flags, and Detailed Findings:

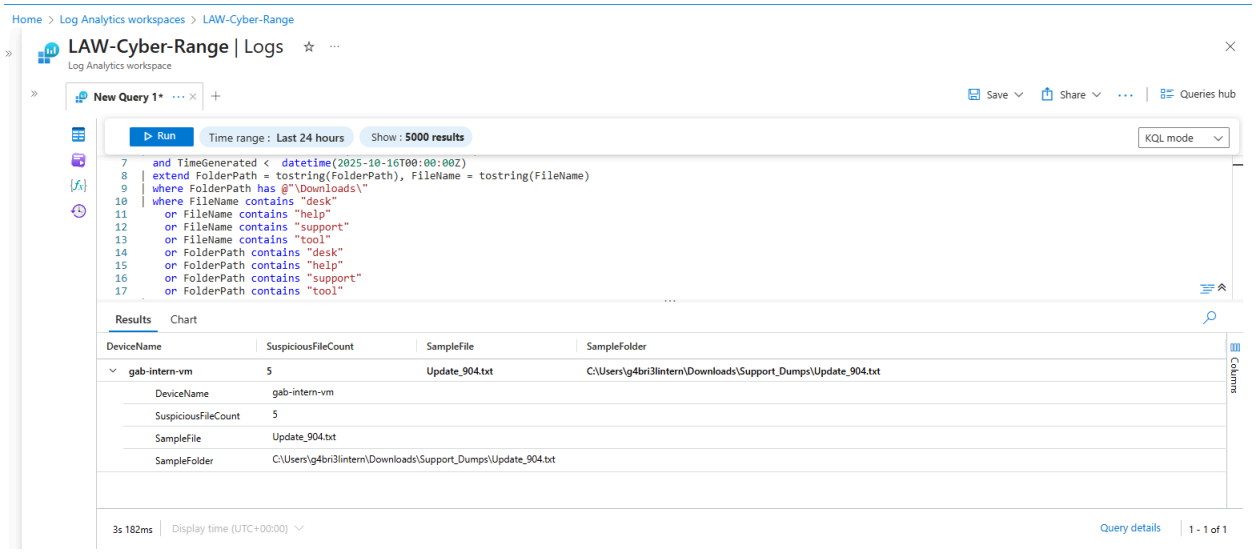
9.0 Starting Point – Host Identification

Objective:
Determine the most suspicious machine based on Downloads activity, naming patterns, and intern-related hosts.

Query Used:

```
DeviceFileEvents
| where TimeGenerated >= datetime(2025-10-01T00:00:00Z)
  and TimeGenerated <  datetime(2025-10-16T00:00:00Z)
| extend FolderPath = tostring(FolderPath), FileName = tostring(FileName)
| where FolderPath has @"\Downloads\"
| where FileName contains "desk"
    or FileName contains "help"
    or FileName contains "support"
    or FileName contains "tool"
    or FolderPath contains "desk"
    or FolderPath contains "help"
    or FolderPath contains "support"
    or FolderPath contains "tool"
| where DeviceName contains "intern"
| summarize SuspiciousFileCount = count(),
            SampleFile   = any(FileName),
            SampleFolder = any(FolderPath)
          by DeviceName
| order by SuspiciousFileCount desc
```

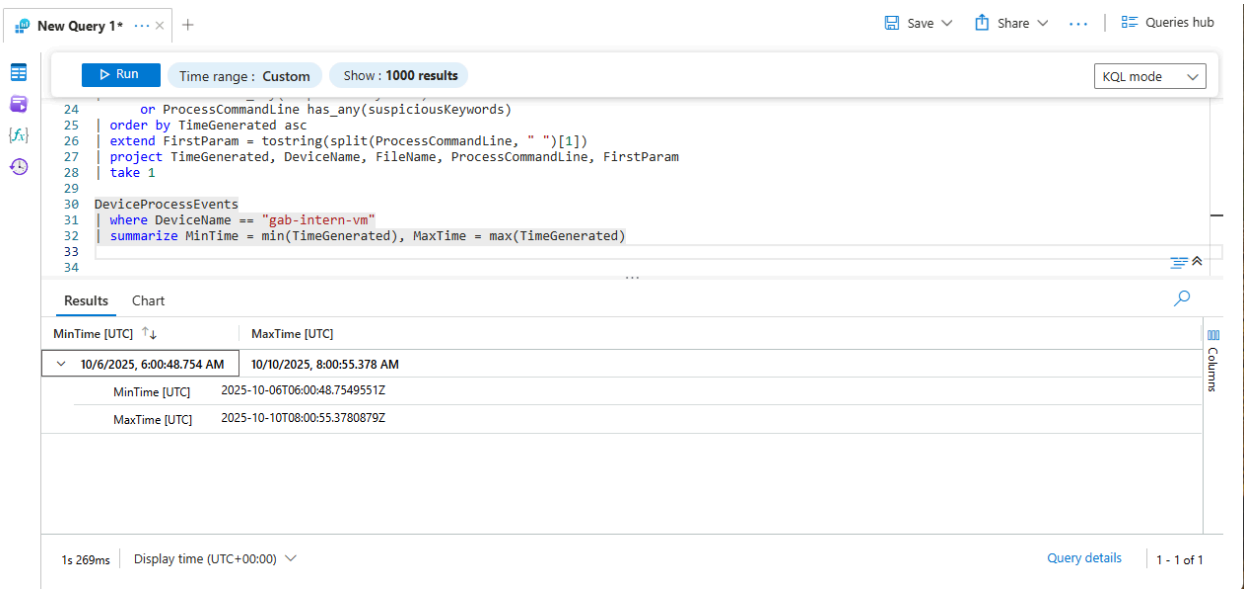
Evidence:



Verification of October data:

DeviceProcessEvents
| where DeviceName == "gab-intern-vm"
| summarize MinTime = min(TimeGenerated),
MaxTime = max(TimeGenerated)

Evidence:



Result:
The conditions point to `gab-intern-vm` as the **most suspicious host** and primary starting point.

9.1 Initial Execution from Downloads

Behavior:
Execution of tooling from the **Downloads** directory with a suspicious first CLI parameter `/Log`.

ATT&CK Mapping:

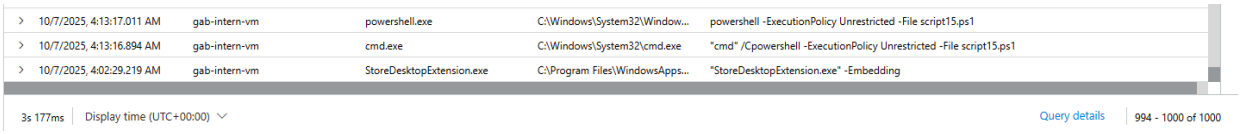
- **T1204.002 – Malicious File (User Execution)**
- **T1059.001 – Command and Scripting Interpreter: PowerShell**

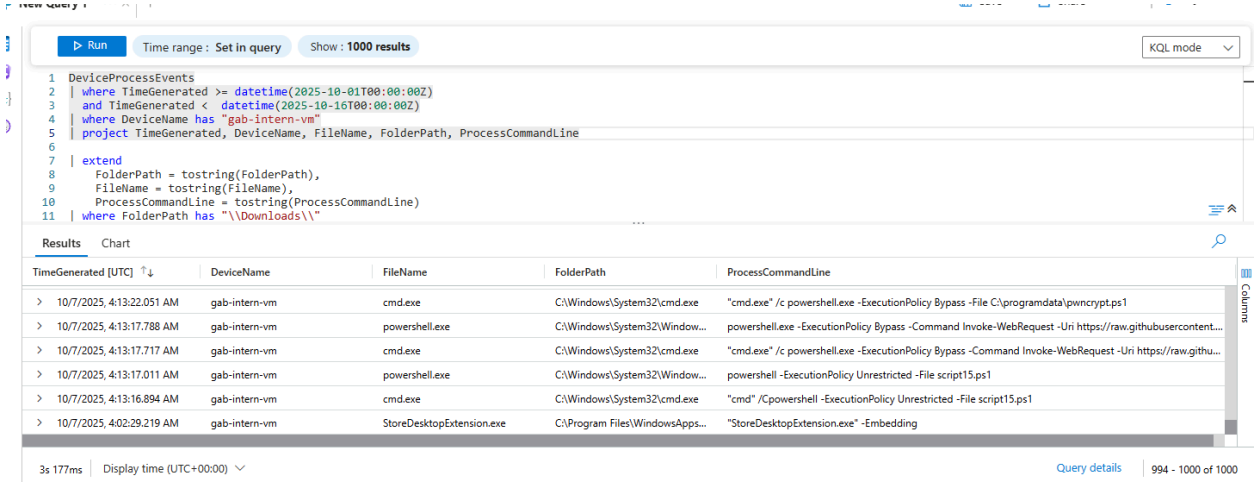
These events serve as the entry point and **timeline anchor** for later activity.

Query Used:

```
DeviceProcessEvents
| where TimeGenerated >= datetime(2025-10-01T00:00:00Z)
  and TimeGenerated <  datetime(2025-10-16T00:00:00Z)
| where DeviceName has "gab-intern-vm"
| project TimeGenerated, DeviceName, FileName, FolderPath, ProcessCommandLine
```

Evidence:





Question:
What was the first CLI parameter name used during the execution of the suspicious program?

FLAG #1: `-ExecutionPolicy`

9.2 Defense Tamper Indicator

Artifact: `DefenderTamperArtifact.lnk`

Role:
Planted tamper-related indicator implying interference with security controls, even if no actual tampering occurred.

ATT&CK Mapping:

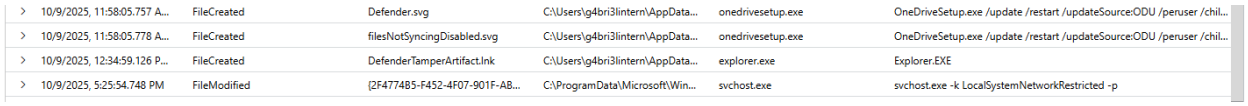
- **T1562.001 – Impair Defenses: Disable or Modify Tools** (simulated / staged evidence)

Query Used:

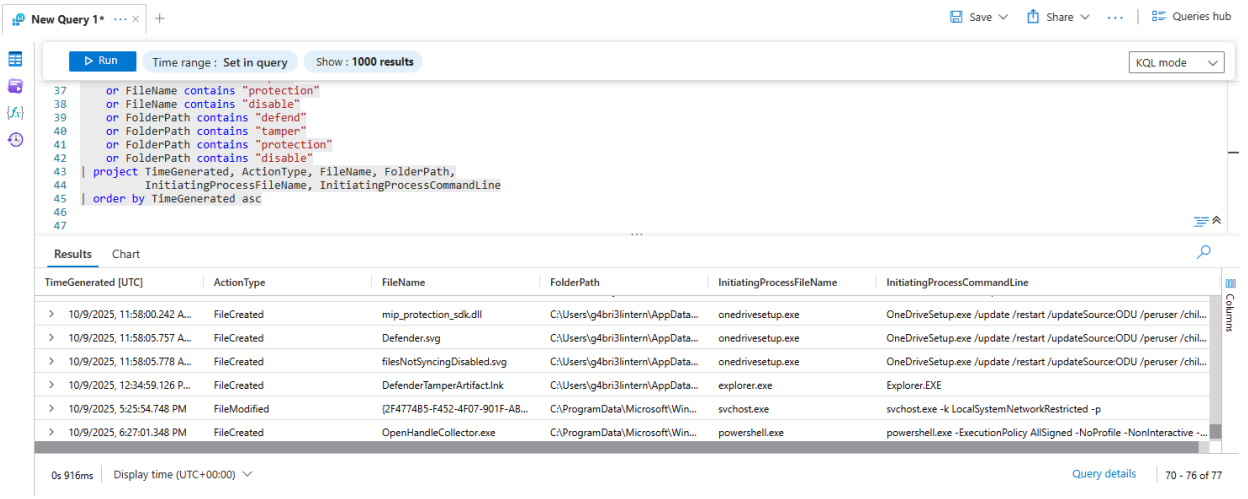
```
DeviceFileEvents
| where DeviceName == "gab-intern-vm"
| where TimeGenerated >= datetime(2025-10-01T00:00:00Z)
  and TimeGenerated <  datetime(2025-10-16T00:00:00Z)
| extend FolderPath = tostring(FolderPath),
       FileName   = tostring(FileName)
```

```
| where FileName contains "defend"
    or FileName contains "defender"
    or FileName contains "tamper"
    or FileName contains "protection"
    or FileName contains "disable"
    or FolderPath contains "defend"
    or FolderPath contains "tamper"
    or FolderPath contains "protection"
    or FolderPath contains "disable"
| project TimeGenerated, ActionType, FileName, FolderPath,
          InitiatingProcessFileName, InitiatingProcessCommandLine
| order by TimeGenerated asc
```

Evidence:



10/9/2025, 11:58:05.757 A...	FileCreated	Defender.svg	C:\Users\g4bri3lintern\AppData...	onedrivesetup.exe	OneDriveSetup.exe /update /restart /updateSource:ODU /peruser /chil...
10/9/2025, 11:58:05.778 A...	FileCreated	filesNotSyncingDisabled.svg	C:\Users\g4bri3lintern\AppData...	onedrivesetup.exe	OneDriveSetup.exe /update /restart /updateSource:ODU /peruser /chil...
10/9/2025, 12:34:59.126 P...	FileCreated	DefenderTamperArtifact.lnk	C:\Users\g4bri3lintern\AppData...	explorer.exe	Explorer.EXE
10/9/2025, 5:25:54.748 PM	FileModified	{2F4774B5-F452-4F07-901F-AB...	C:\ProgramData\Microsoft\Win...	svchost.exe	svchost.exe -k LocalSystemNetworkRestricted -p



Question:
What was the name of the file related to this exploit?

FLAG #2: `DefenderTamperArtifact.lnk`

9.3 Quick Data Probe – Clipboard

Command:

"powershell.exe" -NoProfile -Sta -Command "try { Get-Clipboard | Out-Null } catch { }"

Objective:
Low-friction probe for clipboard contents (e.g., passwords, tokens, copied data).

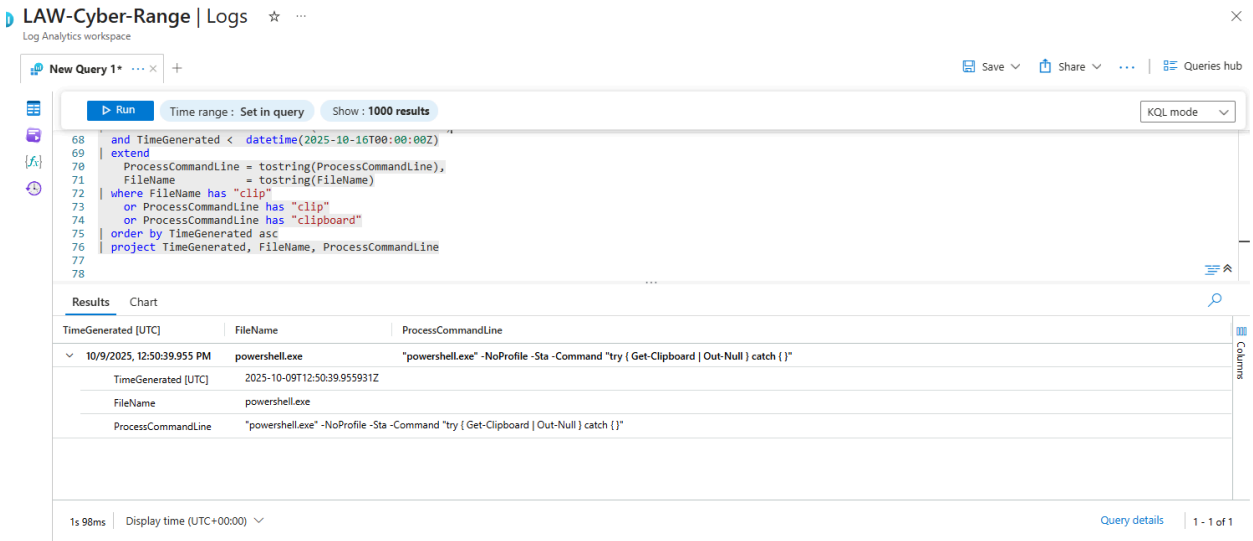
ATT&CK Mapping:

- **T1115 – Clipboard Data**
- **T1059.001 – Command and Scripting Interpreter: PowerShell**

Query Used:

```
DeviceProcessEvents
| where DeviceName == "gab-intern-vm"
| where TimeGenerated >= datetime(2025-10-01T00:00:00Z)
  and TimeGenerated <  datetime(2025-10-16T00:00:00Z)
| extend ProcessCommandLine = tostring(ProcessCommandLine),
         FileName           = tostring(FileName)
| where FileName has "clip"
    or ProcessCommandLine has "clip"
    or ProcessCommandLine has "clipboard"
| order by TimeGenerated asc
| project TimeGenerated, FileName, ProcessCommandLine
```

Evidence:



Question:
Provide the command value tied to this particular exploit.

FLAG #3:
`"powershell.exe" -NoProfile -Sta -Command "try { Get-Clipboard | Out-Null } catch { }"`

9.4 Host & Session Reconnaissance

Observed behaviors:

- Host / system information queries
- Process listing via `tasklist.exe`
- Logon/session enumeration (unique session-related process ID `2533274790397065`)
- Privilege and group membership checks

ATT&CK Mapping:

- **T1082 – System Information Discovery**
- **T1057 – Process Discovery**
- **T1033 / T1087 – Account / User Discovery**
- **T1005 / T1083 – Data from Local System / File and Directory Discovery**

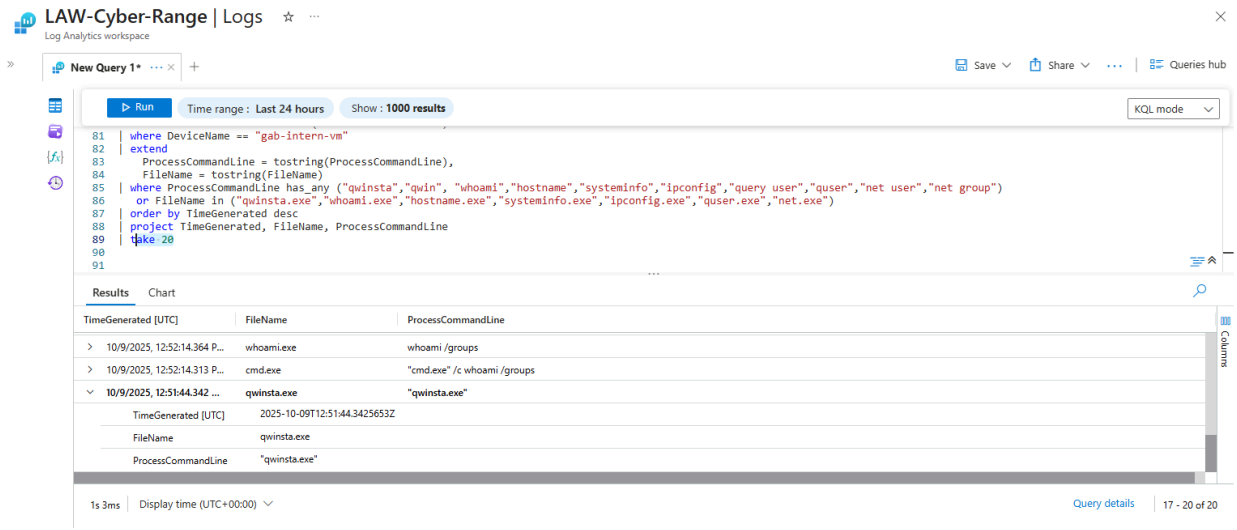
These steps give the actor a clear picture of **who**, **what**, and **where** on the host.

Query Used:

```
DeviceProcessEvents
| where TimeGenerated >= datetime(2025-10-01T00:00:00Z)
  and TimeGenerated <  datetime(2025-10-16T00:00:00Z)
| where DeviceName == "gab-intern-vm"
```

```
| extend ProcessCommandLine = tostring(ProcessCommandLine),
         FileName           = tostring(FileName)
| where ProcessCommandLine has_any
("qwinsta","qwin","whoami","hostname","systeminfo",
                              "ipconfig","query user","quser","net user","net group")
   or FileName in ("qwinsta.exe","whoami.exe","hostname.exe","systeminfo.exe",
                "ipconfig.exe","quser.exe","net.exe")
| order by TimeGenerated desc
| project TimeGenerated, FileName, ProcessCommandLine
| take 20
```

Evidence:



Question:
Point out when the last recon attempt was.

FLAG #4: `2025-10-09T12:51:44.3425653Z`

9.5 Storage Surface Mapping

Behavior:
Enumeration of drives and filesystem locations likely to contain logs or user documents.

Objective:
Identify **where data resides** and **how much capacity is available** for staging.

ATT&CK Mapping:

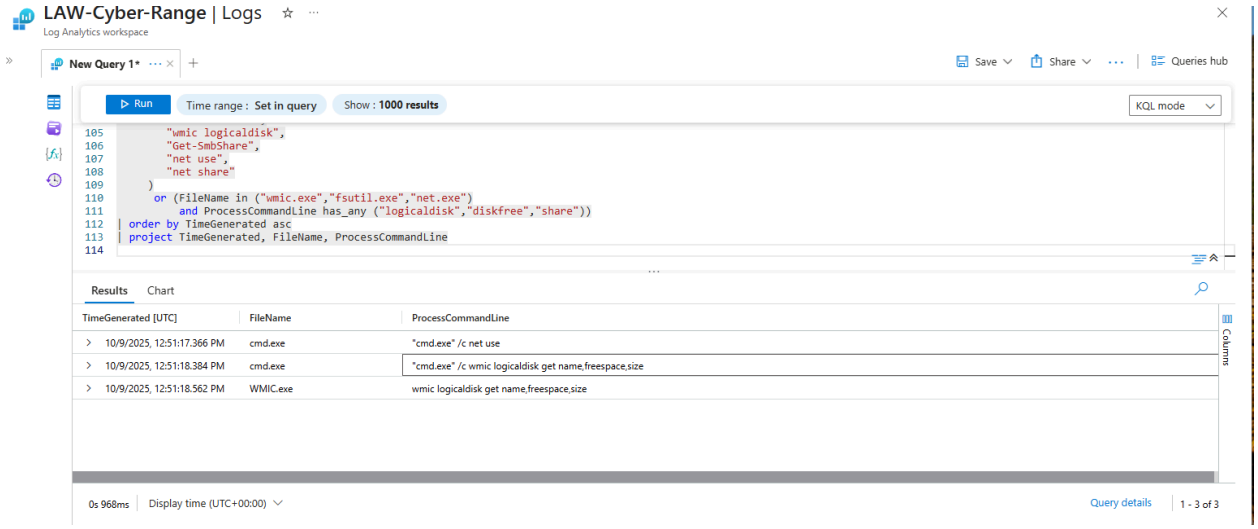
- **T1083 – File and Directory Discovery**

- **T1005 – Data from Local System** (as preparation for staging)

Query Used:

```
DeviceProcessEvents
| where TimeGenerated >= datetime(2025-10-01T00:00:00Z)
  and TimeGenerated <  datetime(2025-10-16T00:00:00Z)
| where DeviceName == "gab-intern-vm"
| extend FileName           = tostring(FileName),
         ProcessCommandLine = tostring(ProcessCommandLine)
| where ProcessCommandLine has_any (
        "Get-PSDrive",
        "Win32_LogicalDisk",
        "logicaldisk",
        "Get-Volume",
        "Get-Disk",
        "fsutil volume",
        "wmic logicaldisk",
        "Get-SmbShare",
        "net use",
        "net share"
    )
   or (FileName in ("wmic.exe","fsutil.exe","net.exe")
       and ProcessCommandLine has_any ("logicaldisk","diskfree","share"))
| order by TimeGenerated asc
| project TimeGenerated, FileName, ProcessCommandLine
```

Evidence:



Question:
Provide the 2nd command tied to this activity.

FLAG #5: `"cmd.exe" /c wmic logicaldisk get name,freespace,size`

9.6 Connectivity & Name Resolution Checks

Behavior:
Name resolution and connectivity checks, including system processes such as `RuntimeBroker.exe`.

Objective:
Validate outbound connectivity and potential egress paths before exfiltration.

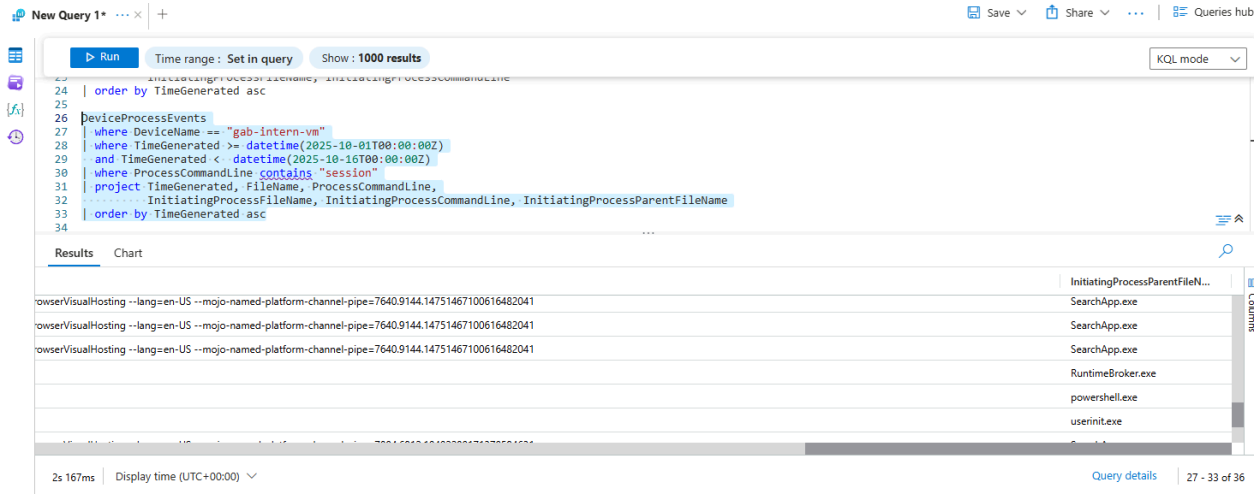
ATT&CK Mapping:

- **T1071.001 – Application Layer Protocol: Web**
- (Conceptually aligned with **T1016 – System Network Configuration Discovery** and exfil preparation.)

Query Used:

```
DeviceProcessEvents
| where DeviceName == "gab-intern-vm"
| where TimeGenerated >= datetime(2025-10-01T00:00:00Z)
  and TimeGenerated <  datetime(2025-10-16T00:00:00Z)
| where ProcessCommandLine contains "session"
| project TimeGenerated, FileName, ProcessCommandLine,
          InitiatingProcessFileName,
          InitiatingProcessCommandLine,
          InitiatingProcessParentFileName
| order by TimeGenerated asc
```

Evidence:



Question:
Provide the File Name of the initiating parent process.

FLAG #6: `RuntimeBroker.exe`

9.7 Data Collection & Staging

Files:

- `Support_701.txt`
- `Update_904.txt`

- `Helpdesk_247.txt`

Consolidated Archive:

- `C:\Users\Public\ReconArtifacts.zip`

Objective:
Aggregate collected artifacts into a single package suitable for **transfer or exfil**.

ATT&CK Mapping:

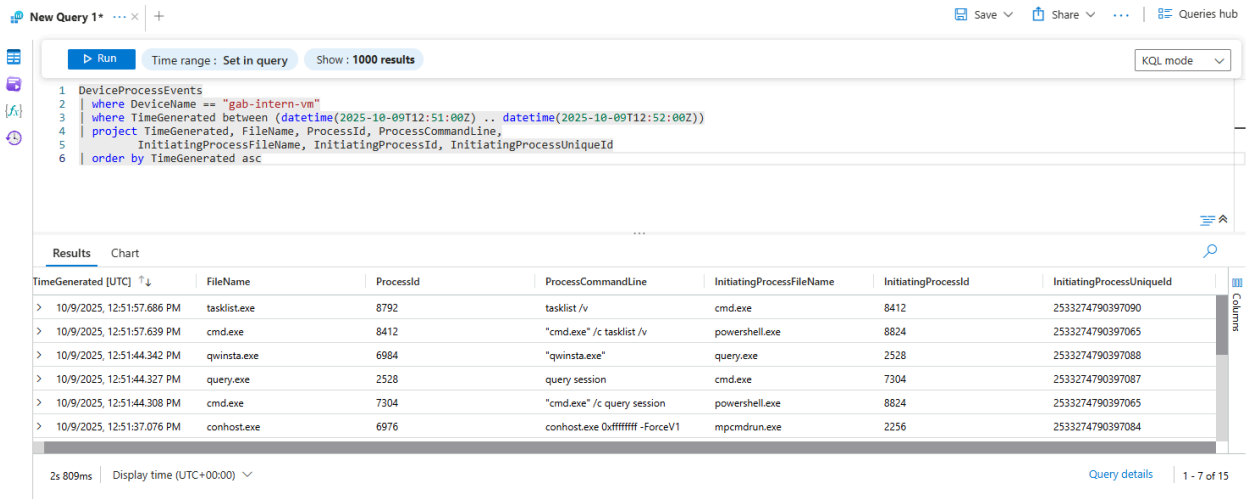
- **T1074.001 – Data Staged: Local Data Staging**

This is the key choke point where **recon output becomes exfil-ready content**.

Query Used:

```
DeviceProcessEvents
| where DeviceName == "gab-intern-vm"
| where TimeGenerated between (datetime(2025-10-09T12:51:00Z) .. datetime(2025-10-09T12:52:00Z))
| project TimeGenerated, FileName, ProcessId, ProcessCommandLine,
        InitiatingProcessFileName, InitiatingProcessId, InitiatingProcessUniqueId
| order by TimeGenerated asc
```

Evidence:



TimeGenerated [UTC]	FileName	ProcessId	ProcessCommandLine	InitiatingProcessFileName	InitiatingProcessId	InitiatingProcessUniqueId
10/9/2025, 12:51:57.686 PM	tasklist.exe	8792	tasklist /v	cmd.exe	8412	2533274790397090
10/9/2025, 12:51:57.639 PM	cmd.exe	8412	"cmd.exe" /c tasklist /v	powershell.exe	8824	2533274790397065
10/9/2025, 12:51:44.342 PM	qwinsta.exe	6984	"qwinsta.exe"	query.exe	2528	2533274790397088
10/9/2025, 12:51:44.327 PM	query.exe	2528	query session	cmd.exe	7304	2533274790397087
10/9/2025, 12:51:44.308 PM	cmd.exe	7304	"cmd.exe" /c query session	powershell.exe	8824	2533274790397065
10/9/2025, 12:51:37.076 PM	conhost.exe	6976	conhost.exe 0xffffffff -ForceV1	mpcmdrun.exe	2256	2533274790397084

Question:
What is the unique ID of the initiating process?

FLAG #7: `2533274790397065`

Runtime process enumeration (supporting recon chain):

Events that capture broad process/process-list snapshots or queries of running services.

A process inventory shows what's present and what to avoid or target for collection.

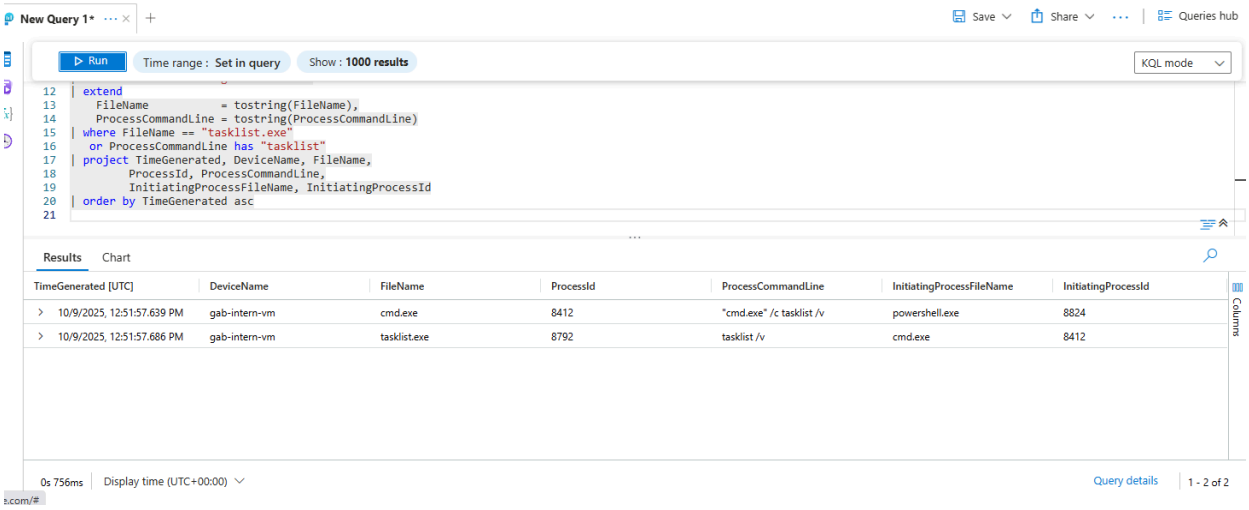
ATT&CK Mapping:

- **Reconnaissance (TA0043)**

Query Used:

```
DeviceProcessEvents
| where TimeGenerated >= datetime(2025-10-01T00:00:00Z)
  and TimeGenerated <  datetime(2025-10-16T00:00:00Z)
| where DeviceName == "gab-intern-vm"
| extend FileName           = tostring(FileName),
         ProcessCommandLine = tostring(ProcessCommandLine)
| where FileName == "tasklist.exe"
   or ProcessCommandLine has "tasklist"
| project TimeGenerated, DeviceName, FileName,
         ProcessId, ProcessCommandLine,
         InitiatingProcessFileName, InitiatingProcessId
| order by TimeGenerated asc
```

Evidence:



Question:
Provide the file name of the process that best demonstrates a runtime process enumeration event on the target host.

FLAG #8: `tasklist.exe`

Privilege mapping (who am I / token awareness):

Telemetry that reflects queries of group membership, token properties, or privilege listings.

Privilege mapping informs whether the actor proceeds as a user or seeks elevation.

ATT&CK Tie-In for Privilege Mapping:

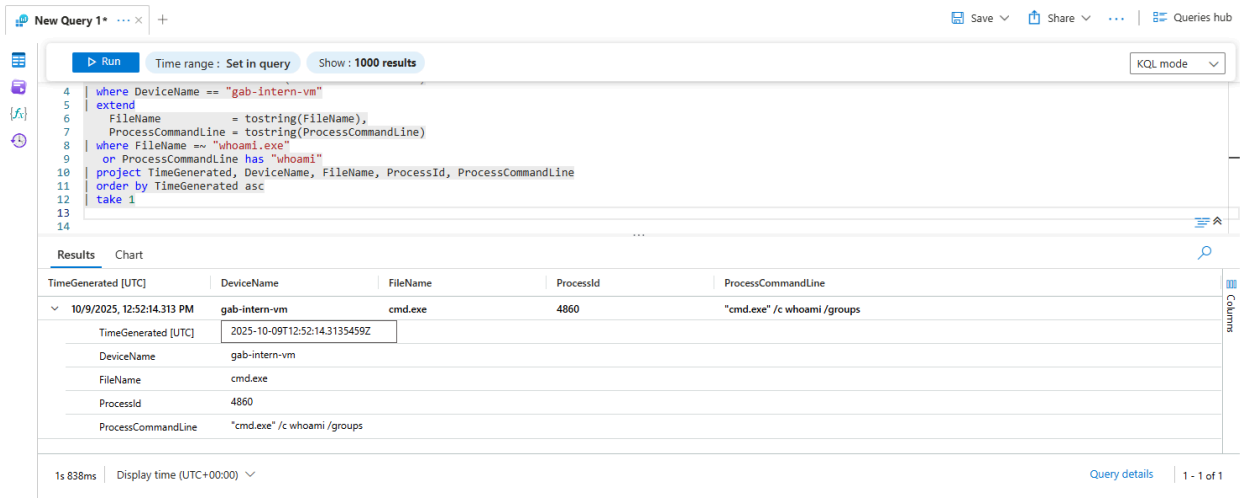
- **T1033 – Account Discovery**
- **T1069 – Permission Groups Discovery**

Query Used:

```
DeviceProcessEvents
| where TimeGenerated >= datetime(2025-10-01T00:00:00Z)
  and TimeGenerated <  datetime(2025-10-16T00:00:00Z)
| where DeviceName == "gab-intern-vm"
```

```
| extend FileName          = tostring(FileName),
         ProcessCommandLine = tostring(ProcessCommandLine)
| where FileName =~ "whoami.exe"
   or ProcessCommandLine has "whoami"
| project TimeGenerated, DeviceName, FileName, ProcessId, ProcessCommandLine
| order by TimeGenerated asc
| take 1
```

Evidence:



Question:
Identify the timestamp of the very first attempt.

FLAG #9: `2025-10-09T12:52:14.3135459Z`

9.8 Outbound Reachability & Exfil Simulation

Connectivity Testing:

- Early outbound destination: `www.msftconnecttest.com`

Simulated Exfil Attempt:

- Last unusual outbound IP: `100.29.147.161`

ATT&CK Mapping:

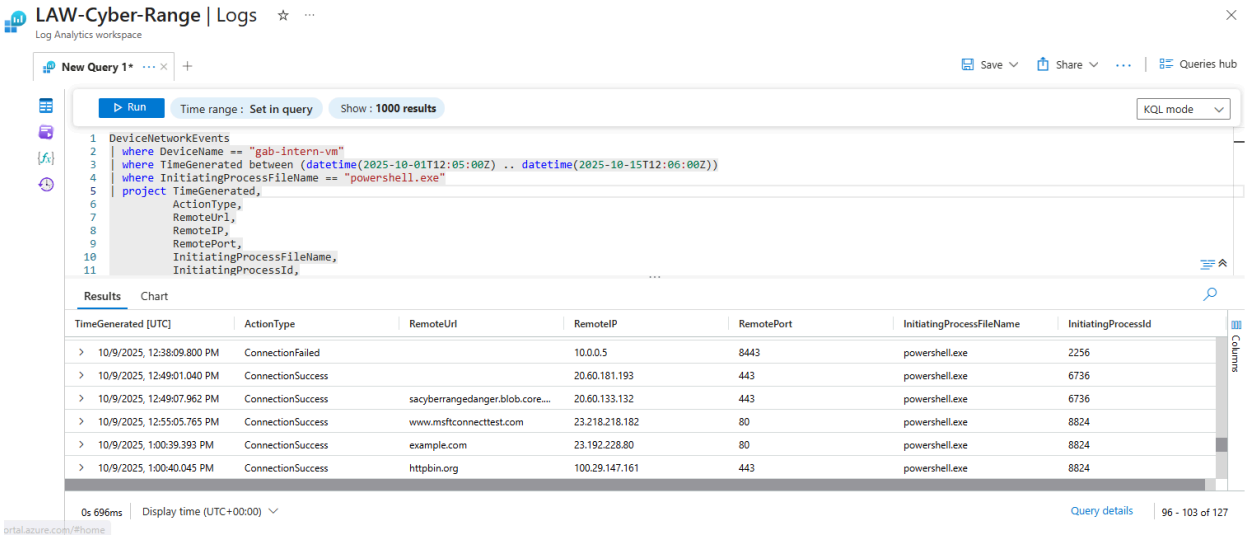
- **T1071.001 – Application Layer Protocol: Web**
- **T1041 – Exfiltration Over C2 Channel** (conceptually)
- **T1567.002 – Exfiltration to Cloud Storage** (mapped as generic web-service exfil in this scenario)

Network Query – PowerShell-initiated outbound connections:

Query Used:

```
DeviceNetworkEvents
| where DeviceName == "gab-intern-vm"
| where TimeGenerated between (datetime(2025-10-01T12:05:00Z) ..
datetime(2025-10-15T12:06:00Z))
| where InitiatingProcessFileName == "powershell.exe"
| project TimeGenerated,
        ActionType,
        RemoteUrl,
        RemoteIP,
        RemotePort,
        InitiatingProcessFileName,
        InitiatingProcessId,
        AdditionalFields
| order by TimeGenerated asc
```

Evidence:



Question :
Which outbound destination was contacted first?

FLAG #10: www.msftconnecttest.com

Staging Evidence – File Consolidation:

File system events or operations that show grouping, consolidation, or packaging of gathered items.

Staging is the practical step that simplifies exfiltration and should be correlated back to prior recon.

ATT&CK Mapping:

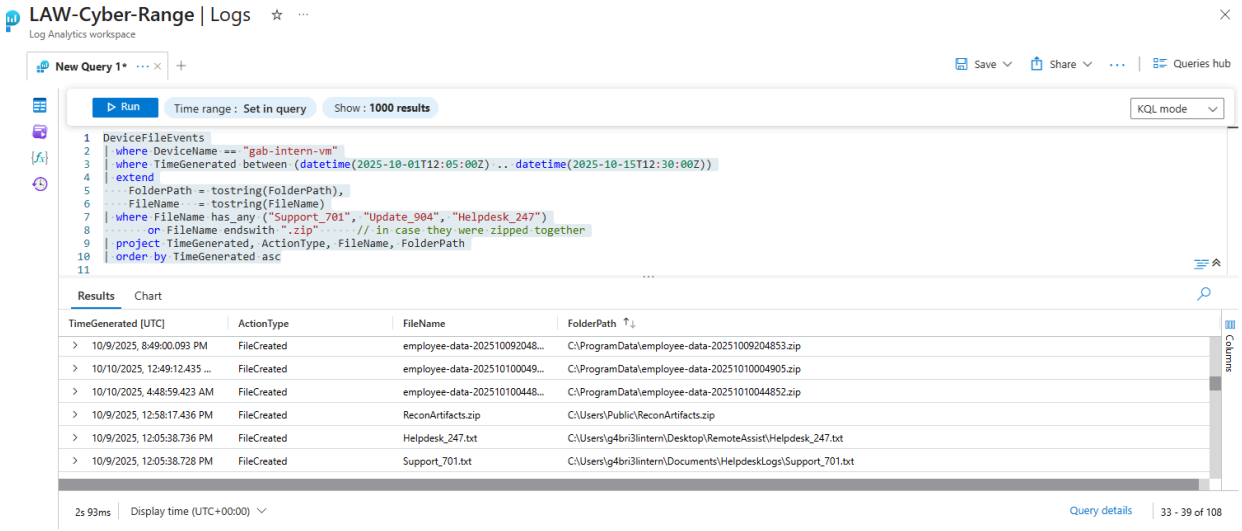
- **Reconnaissance (TA0043)**

Query Used:

```
DeviceFileEvents
| where DeviceName == "gab-intern-vm"
| where TimeGenerated between (datetime(2025-10-01T12:05:00Z) ..
datetime(2025-10-15T12:30:00Z))
| extend FolderPath = tostring(FolderPath),
        FileName   = tostring(FileName)
```

```
| where FileName has_any ("Support_701", "Update_904", "Helpdesk_247")
      or FileName endswith ".zip"
| project TimeGenerated, ActionType, FileName, FolderPath
| order by TimeGenerated asc
```

Evidence:



Question:
Provide the full folder path value where the artifact was first dropped.

FLAG #11: `C:\Users\Public\ReconArtifacts.zip`

Outbound Transfer / Egress Attempts:

Network events or process activity indicating outbound transfers or upload attempts, even if they fail.

Succeeded or not, attempt is still proof of intent — and it reveals egress paths or block points.

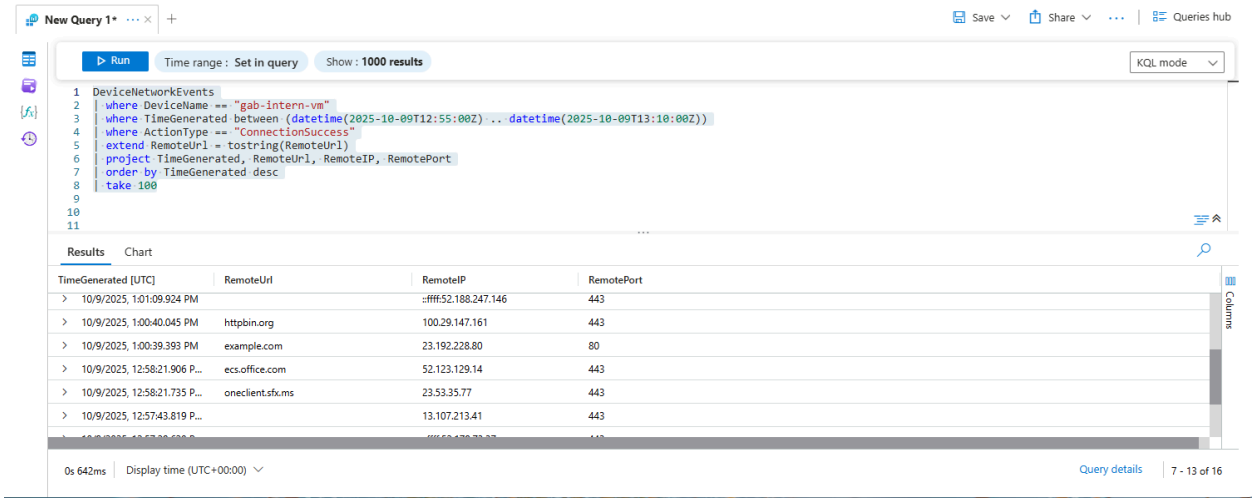
ATT&CK Mapping:

- **T1071.001 – Application Layer Protocol: Web**
- **T1041 – Exfiltration Over C2 Channel** (conceptually)
- **T1567.002 – Exfiltration to Cloud Storage** (mapped as generic web-service exfil in this scenario)

Query Used:

```
DeviceNetworkEvents
| where DeviceName == "gab-intern-vm"
| where TimeGenerated between (datetime(2025-10-09T12:55:00Z) ..
datetime(2025-10-09T13:10:00Z))
| where ActionType == "ConnectionSuccess"
| extend RemoteUrl = tostring(RemoteUrl)
| project TimeGenerated, RemoteUrl, RemoteIP, RemotePort
| order by TimeGenerated desc
| take 100
```

Evidence:



Question:
Provide the IP of the last unusual outbound connection.

FLAG #12: `100.29.147.161`

9.9 Persistence – Scheduled Task

Mechanism:

schtasks.exe /Create /SC ONLOGON /TN SupportToolUpdater ...

Task Name:

- `SupportToolUpdater`

Behavior:
Ensures recurring or logon-triggered execution of the actor's tooling.

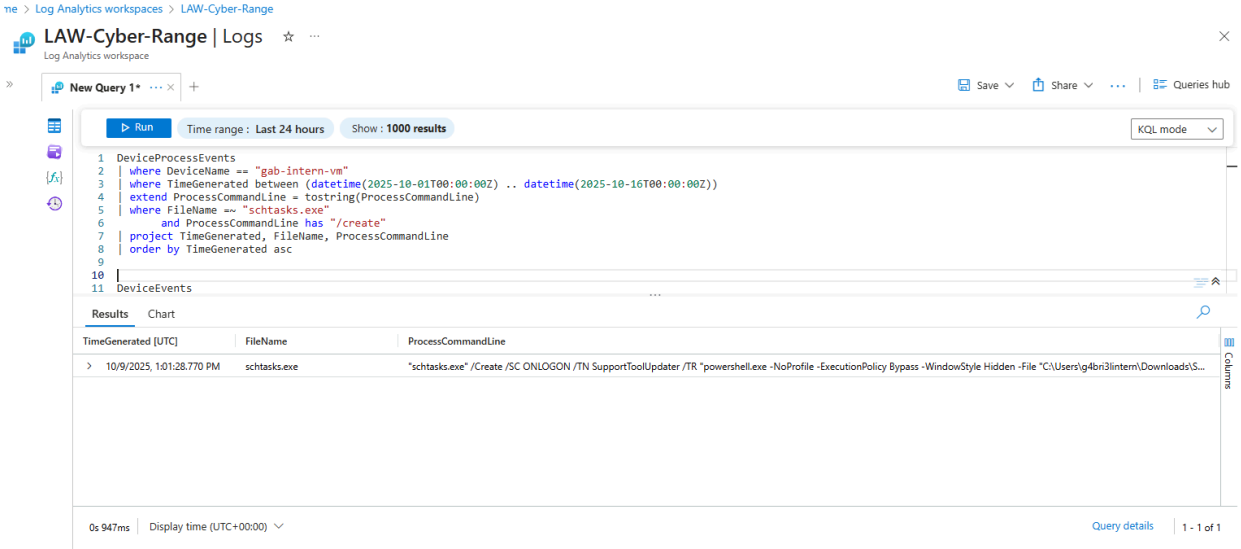
ATT&CK Mapping:

- **T1053.005 – Scheduled Task/Job: Scheduled Task**

Query Used:

```
DeviceProcessEvents
| where DeviceName == "gab-intern-vm"
| where TimeGenerated between (datetime(2025-10-01T00:00:00Z) ..
datetime(2025-10-16T00:00:00Z))
| extend ProcessCommandLine = tostring(ProcessCommandLine)
| where FileName =~ "schtasks.exe"
      and ProcessCommandLine has "/create"
| project TimeGenerated, FileName, ProcessCommandLine
| order by TimeGenerated asc
```

Evidence:



Question:
Provide the value of the task name.

FLAG #13: `SupportToolUpdater`

9.10 Persistence – Autorun Fallback

Mechanism:
User-scope autorun persistence via registry **Run/RunOnce** keys.

Registry Value Name:

- `RemoteAssistUpdater`

Role:
Secondary/user-scope persistence if the scheduled task is removed.

ATT&CK Mapping:

- **T1547.009 – Boot or Logon Autostart Execution: Registry Run Keys / Startup Folder**

Queries Attempted (No direct return in dataset, value provided by scenario):

```
DeviceRegistryEvents
| where DeviceName == "gab-intern-vm"
| where TimeGenerated between (datetime(2025-10-01T00:00:00Z) ..
datetime(2025-10-16T00:00:00Z))
| where ActionType in ("RegistryValueSet", "RegistryValueAdded")
| where RegistryKey has @"\Software\Microsoft\Windows\CurrentVersion\Run"
   or RegistryKey has @"\Software\Microsoft\Windows\CurrentVersion\RunOnce"
| project TimeGenerated,
        DeviceName,
        ActionType,
        RegistryKey,
        RegistryValueName,
        RegistryValueData,
        InitiatingProcessFileName,
        InitiatingProcessCommandLine
```

```
| order by TimeGenerated asc
```

-and–

```
DeviceRegistryEvents
| where DeviceName == "gab-intern-vm"
| where TimeGenerated between (datetime(2025-10-01T00:00:00Z) ..
datetime(2025-10-16T00:00:00Z))
| where ActionType in ("RegistryValueSet", "RegistryValueAdded")
| where RegistryValueName =~ "RemoteAssistUpdater"
| project TimeGenerated,
        RegistryKey,
        RegistryValueName,
        RegistryValueData,
        InitiatingProcessFileName
| order by TimeGenerated asc
```

Question:
What was the name of the registry value?

FLAG #14: `RemoteAssistUpdater`

9.11 Planted Narrative / Cover Artifact

Artifact:

- `SupportChat_log.lnk`

Behavior:
Acts as a **cover** or planted explanation, giving the impression of a legitimate “support chat log” to justify unusual activity.

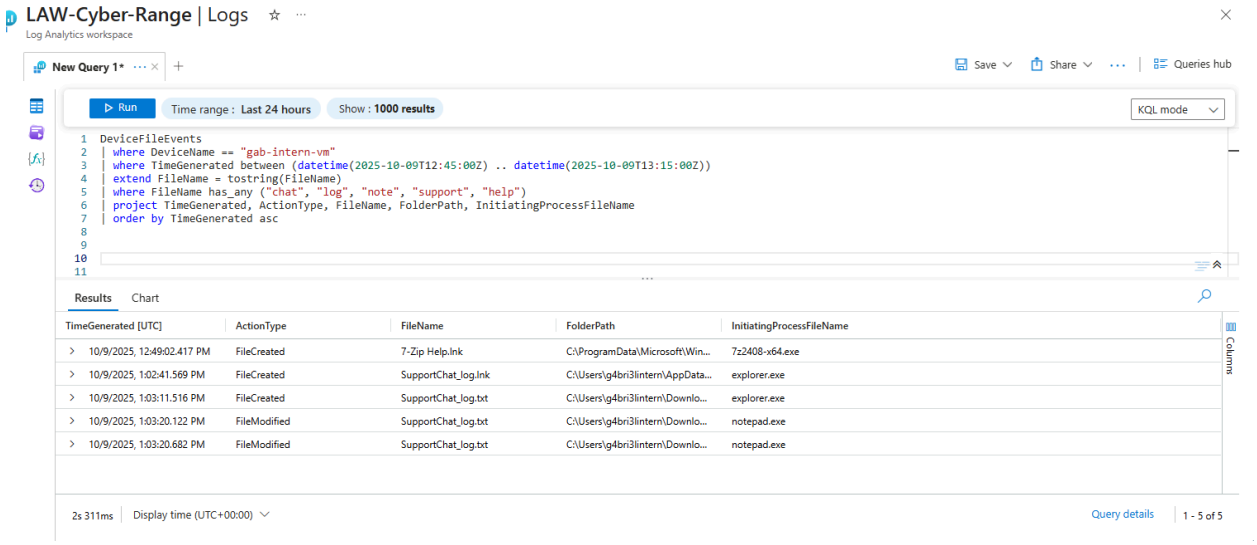
ATT&CK Mapping:

- **T1036 – Masquerading** (using naming/placement to disguise malicious activity as normal support operations)

Query Used:

```
DeviceFileEvents
| where DeviceName == "gab-intern-vm"
| where TimeGenerated between (datetime(2025-10-09T12:45:00Z) ..
datetime(2025-10-09T13:15:00Z))
| extend FileName = tostring(FileName)
| where FileName has_any ("chat","log","note","support","help")
| project TimeGenerated, ActionType, FileName, FolderPath, InitiatingProcessFileName
| order by TimeGenerated asc
```

Evidence:



Question:
Identify the file name of the artifact left behind.

FLAG #15: `SupportChat_log.lnk`

---

10. Lessons Learned

- **Low-privilege / intern systems still matter.**
    - They are often used for testing and can become quiet staging areas. Apply full telemetry and monitoring.

- **Execution from user-writable paths is high-value signal.**
    - Detect process starts from `Downloads`, `Desktop`, `Temp`, especially involving scripting hosts or unusual parameters.
- **Recon chains > isolated events.**
    - Clipboard → host info → processes → sessions → privileges → storage enumeration in quick succession is a strong indicator of interactive recon.
- **Staging in shared/public locations is a key exfil precursor.**
    - Monitor `.zip` / `.7z` creation in `C:\Users\Public\` with names suggesting recon, logs, or support data.
- **Persistence is layered.**
    - Here, `SupportToolUpdater` (scheduled task) + `RemoteAssistUpdater` (autorun) provide redundant re-entry paths.
    - Detections should jointly cover **scheduled tasks AND autorun keys**.
- **Narrative artifacts are detection opportunities.**
    - Files like `SupportChat_log.lnk` created or opened near suspicious behavior are red flags for **planted explanations**.
- **KQL pivoting is essential for storytelling.**
    - Combining `DeviceProcessEvents`, `DeviceFileEvents`, `DeviceNetworkEvents`, and `DeviceEvents` via timestamps, PIDs, and filenames allows you to reconstruct the **full attack chain**, not just isolated events.

---

11. Recommended Next Steps

Cleanup

Remove:

- `SupportToolUpdater` scheduled task.
- `RemoteAssistUpdater` autorun registry value.
- `ReconArtifacts.zip` and all related staged artifacts.
- `SupportChat_log.lnk` and any similar cover files.

Then:

- Validate that no additional persistence (services, WMI, other autoruns) exists on `gab-intern-vm`.

Detection Engineering

Create or refine analytic rules for:

- Scheduled task creation containing `support`, `helpdesk`, `tool`, `remote` patterns.
- Autorun registry changes with similar naming.
- Archive creation under `C:\Users\Public\` with recon/log-style names.
- Short-window chains of:
    - Clipboard access + recon + storage/drive enumeration.

Policy & Hardening

- Restrict script interpreter use (PowerShell, CMD) on intern/low-trust machines.

- Use AppLocker / WDAC (or equivalents) to limit execution from **Downloads** and other user-writable locations.

Training & Exercises

Use this scenario as a **walkthrough case**:

- Map each stage to **MITRE ATT&CK techniques** (as documented above).
- Emphasize how technical telemetry ties to **attacker objectives**: recon, staging, persistence, and exfil.

Bruce Thornton
11/09/2025