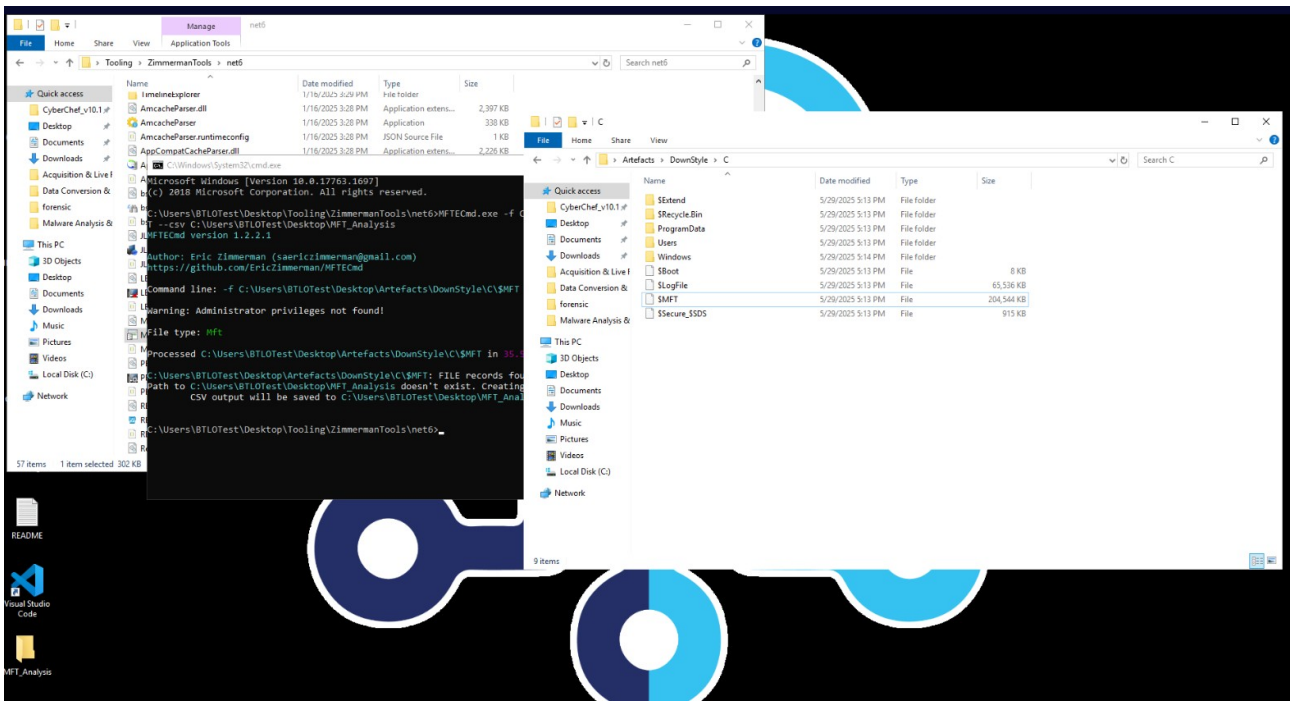
Incident Response Report

Simulated Unauthorised Intrusion via WSL Backdoor – "Black Screen" Alert
Date: 19–20 April 2025
Conducted as part of: Security Blue Team Labs
Prepared by: Kennedy Aikohi
Role: Cybersecurity Analyst / Incident Responder



Executive Summary

On 19 April 2025, a system administrator reported a critical anomaly: their workstation suddenly displayed a black screen, followed by a pop-up message stating:
"You got hacked."

This triggered an immediate response by the Incident Response (IR) team, who initiated a full-scale forensic investigation.

The analysis revealed a multi-stage attack involving:

- PowerShell script execution
- Abuse of Windows Subsystem for Linux (WSL) for persistence
- Credential theft
- Lateral movement within the network

Each phase of the intrusion was reconstructed using forensic tools and artefact analysis, including Event Logs, PowerShell logs, Prefect, MFT records, and autorun entries. This report outlines the timeline of the attack and provides supporting screenshots from the artefacts.

---

1. Initial Compromise

Finding:

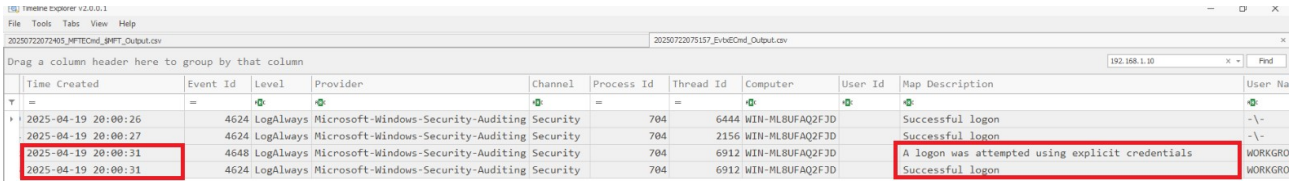
The exact date and time of the initial unauthorised access were identified by correlating a suspicious remote host IP address with successful logon events in the Windows Security logs. These logs recorded remote logon activity that included source network information, enabling a reliable reconstruction of the compromise timeline.

By parsing the Security.evtx file using EvtxECmd.exe, a successful logon (Event ID 4624) was found with a Logon Type of 10, indicating a RemoteInteractive session (e.g., RDP). The source IP 192[.]168[.]1[.]10 was not recognised as a legitimate administrative machine, confirming it as the attacker's system.

Details:

- Timestamp (UTC): 2025-04-19 20:00:31
- Method of Compromise:
  Remote Desktop Protocol (RDP) logon using stolen or brute-forced credentials:
  - Logon Type: 2 (RemoteInteractive)
  - Authentication Package: NTLM
  - Remote Host (IP:Port): 192[.]168[.]1[.]10:0
  - Target Account: Administrator
  - Event ID: 4624 (Successful Logon)

Supporting Evidence



Timeline Explorer v2.0.0.1

File Tools Tabs View Help

20250722072405_MFTECmd_$MFT_Output.csv | 20250722075157_EvtxECmd_Output.csv

Drag a column header here to group by that column

192.168.1.10 | Find

Time Created	Event Id	Level	Provider	Channel	Process Id	Thread Id	Computer	User Id	Map Description	User Na
2025-04-19 20:00:26	4624	LogAlways	Microsoft-Windows-Security-Auditing	Security	704	6444	WIN-ML8UFAQ2FJD		Successful logon	-\-
2025-04-19 20:00:27	4624	LogAlways	Microsoft-Windows-Security-Auditing	Security	704	2156	WIN-ML8UFAQ2FJD		Successful logon	-\-
2025-04-19 20:00:31	4648	LogAlways	Microsoft-Windows-Security-Auditing	Security	704	6912	WIN-ML8UFAQ2FJD		A logon was attempted using explicit credentials	WORKGRO
2025-04-19 20:00:31	4624	LogAlways	Microsoft-Windows-Security-Auditing	Security	704	6912	WIN-ML8UFAQ2FJD		Successful logon	WORKGRO

---

2. Attacker's Entry Point

Finding:

The attacker's entry point into the network was identified by analysing the source IP address associated with the initial unauthorised access. This IP address was mapped to the specific workstation used by the attacker during the incident.

By correlating the security event logs, the workstation name involved in the malicious logon attempts was determined, providing insight into the exact machine targeted and accessed.

Details:

- Workstation Name: WIN-HR-DEPARTME

- Method of Identification:
  The attacker's IP address from the initial logon events was traced to this workstation, confirming it as the origin of the compromise. This was done by examining the source workstation field in the security event logs related to the logon activity.

Supporting Evidence

Remote Host	Payload Data1
WIN-HR-DEPARTME (192.168.1.10)	Target: WIN-ML8UFAQ2FJD\Administrator
WIN-HR-DEPARTME (192.168.1.10)	Target: WIN-ML8UFAQ2FJD\Administrator
192.168.1.10:0	Target: WIN-ML8UFAQ2FJD\Administrator
WIN-ML8UFAQ2FJD (192.168.1.10)	Target: WIN-ML8UFAQ2FJD\Administrator

---

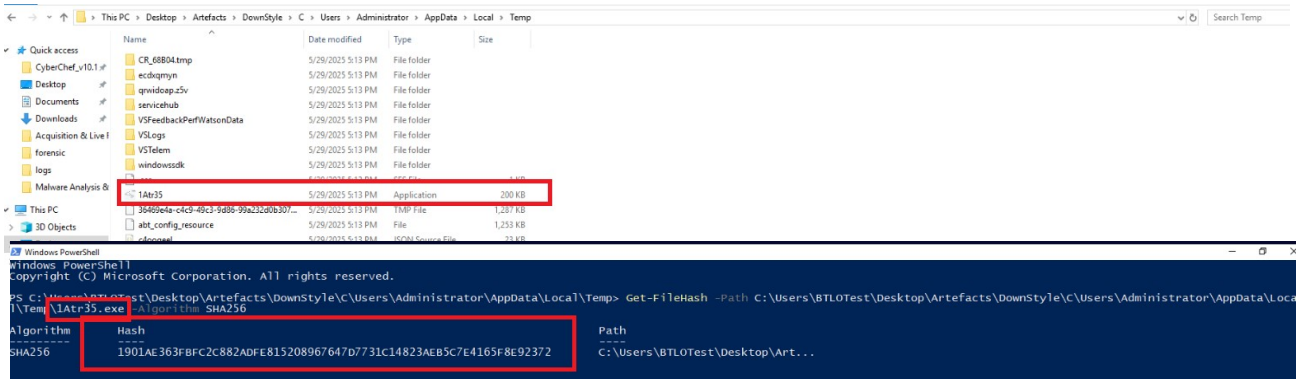
3. Persistence Mechanism
Finding:
Analysis of the compromised system revealed a persistence mechanism designed to maintain attacker access across reboots and logons. During examination of the Administrator user's profile directory, a suspicious executable named 1ATR35.EXE was discovered. File analysis and threat intelligence confirmed it to be a malicious dropper.

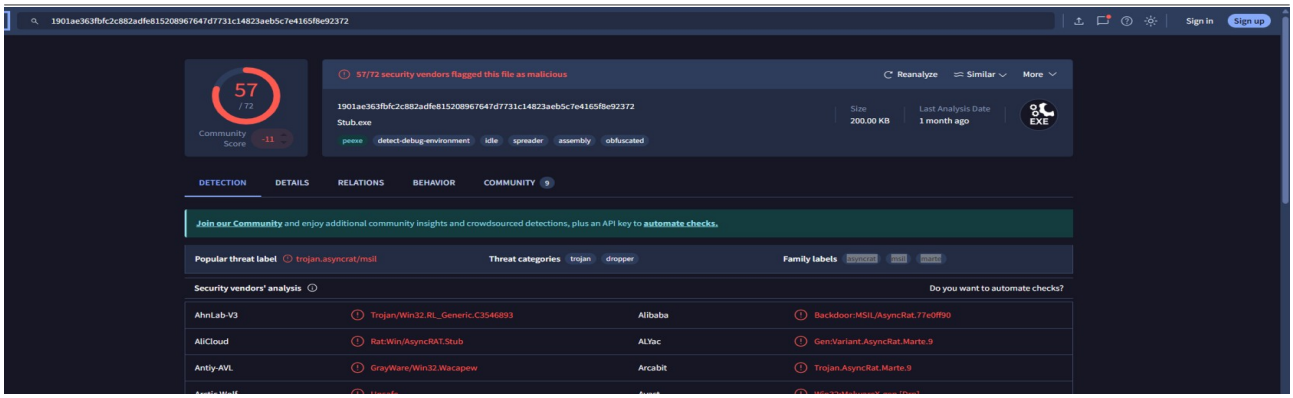
Details:

- File Name: 1ATR35.EXE
- File Hash (SHA-256): 1901ae363fbfc2c882adfe815208967647d7731c14823aeb5c7e4165f8e92372
- File Location: C:\Users\Administrator\AppData\Local\Temp
- Malware Family Labels: AsyncRAT, MSIL, Marte

Persistence Method:
While no direct registry or startup folder entry was initially identified, the placement of the executable in the Temp directory and timing of its access correlating with user logons suggest it was configured to auto-launch—likely through a scheduled task, logon script, or other stealthy startup mechanism.

Supporting Evidence:





---

4. Enumeration Activity

Finding:

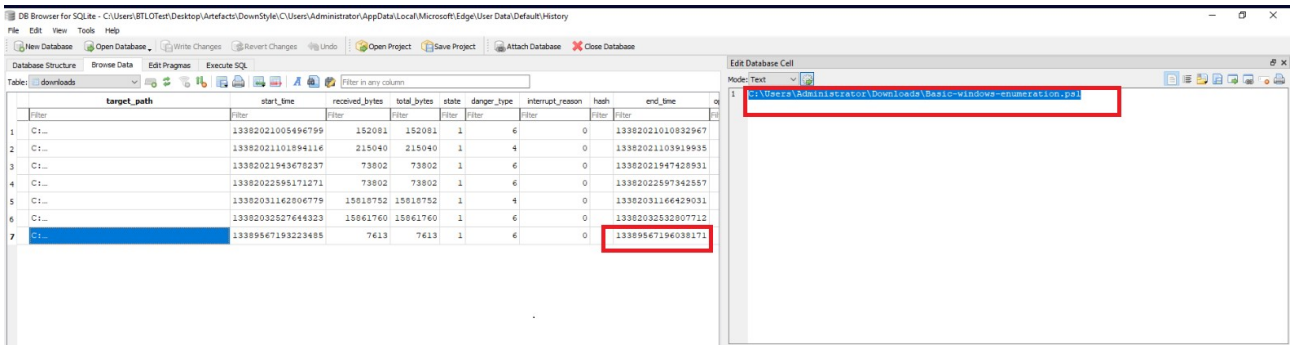
The threat actor performed system enumeration by downloading and executing a reconnaissance script. The PowerShell script, retrieved from a public GitHub repository, was designed to collect host, user, and network information.

The script's presence and execution were later confirmed by examining Microsoft Edge browser history using SQLite analysis, revealing the URL access and script download.

Details:

- Script Name: Basic-windows-enumeration.ps1
- Defanged URL: hxxps://github[.]com/Hacker22o2/Basic-windows-enumeration/blob/master/Basic-windows-enumeration.ps1
- Purpose:
  A PowerShell enumeration tool intended to gather system metadata including active users, IP configuration, running processes, and installed software.

Supporting Evidence



---

5. Enumeration Execution Method

Finding:

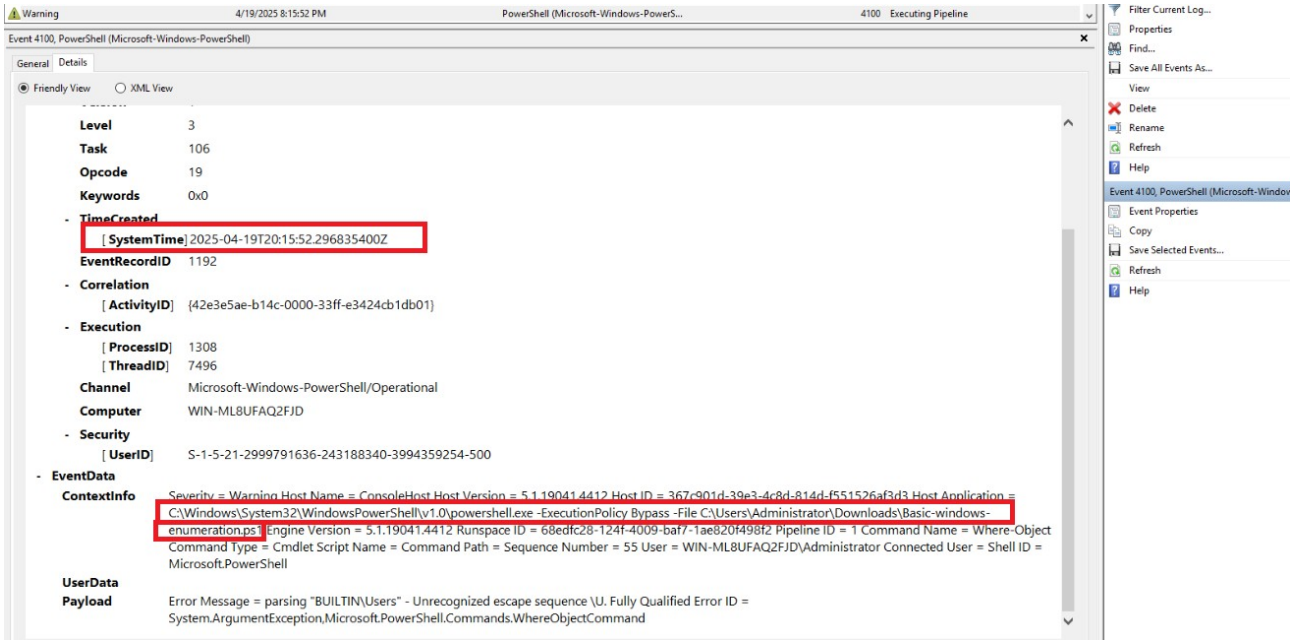
The attacker executed the enumeration script using PowerShell with an explicit command that bypassed script execution policies. Since the file was a .ps1 script, identifying the method of execution required analysing PowerShell-related logs, specifically Event ID 4100 (PowerShell engine start).

By correlating log entries with the known script name (Basic-windows-enumeration.ps1), the exact command used for execution was identified in the PowerShell operational logs.

Details:

- Command Used:
  powershell.exe -ExecutionPolicy Bypass -File "C:\Users\Administrator\Downloads\Basic-windows-enumeration.ps1"
- Log Source:
  PowerShell Event Logs → Event ID 4100 (Engine Start)
  This event indicated the launch of PowerShell with the above parameters, confirming execution of the enumeration script.

Supporting Evidence



6. Discovered Sensitive File

Finding:

During the post-compromise phase of the intrusion, a sensitive file named common-Users-Passwords.txt was identified on the Administrator's Desktop. While the original file was not directly recovered, a .lnk (Windows shortcut) file referencing it was discovered, strongly suggesting the attacker interacted with it after gaining access.

File Details:

- File Name: common-Users-Passwords.txt
- Location: C:\Users\Administrator\Desktop\common-Users-Passwords.txt
- Shortcut Evidence: common-Users-Passwords.txt.lnk

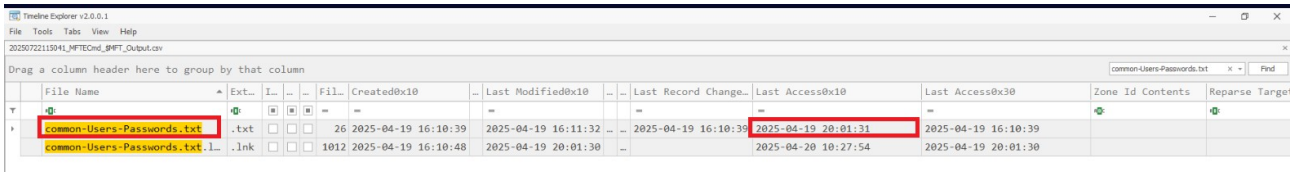
Evidence & Timeline:

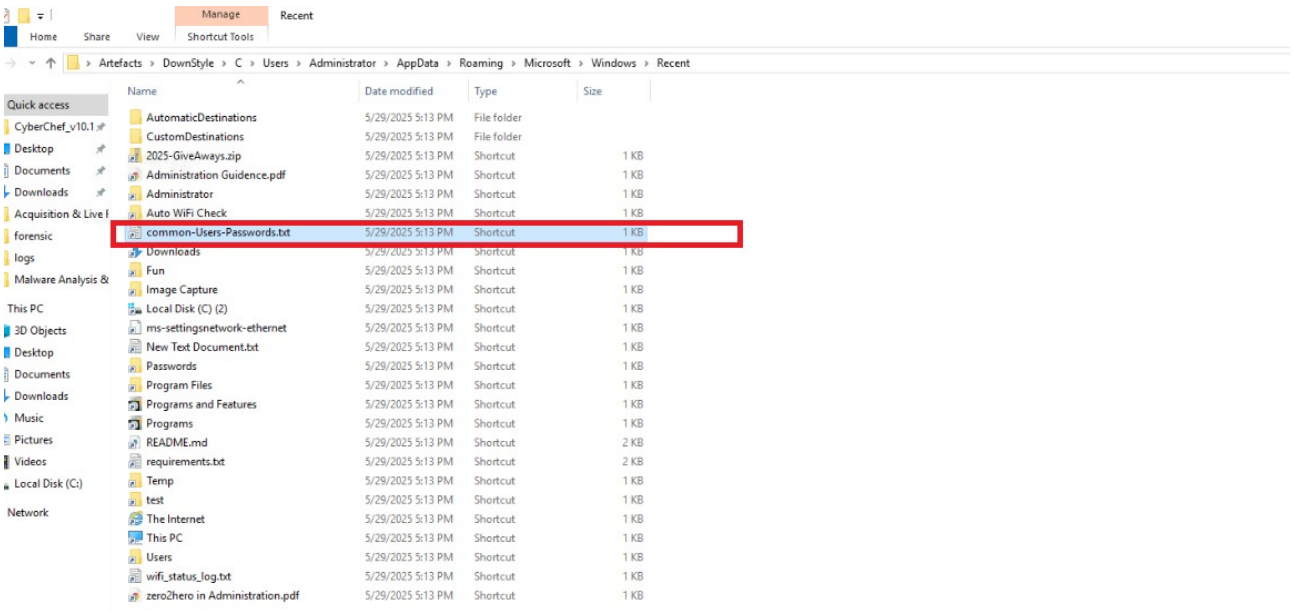
The .lnk file, parsed using forensic tooling, revealed that the original file was accessed on April 19, 2025, at 20:01:30, which occurred after the initial compromise. This delayed access implies the attacker conducted further internal reconnaissance before locating and opening the credentials file.

MFT records corroborate this timeline, showing:

- .txt file last accessed: 2025-04-19 20:01:31
- .lnk file last modified: 2025-04-19 20:01:30
- File creation and modification events align with typical file exploration behaviour

Screenshot Evidence:





Conclusion:
Although the file was not accessed at initial entry, the attacker located and opened it later, likely after surveying the system environment. This action indicates credential harvesting and elevates the severity of the compromise due to the exposure of plaintext credentials.

---

7. Lateral Movement Target
Finding:
Following the initial compromise and internal reconnaissance, evidence shows the attacker attempted lateral movement within the internal network by targeting a specific internal IP address.

- Targeted IP Address: 192[.]168[.]1[.]3
- Verification Method: Corroborated through security event logs showing connection attempts and remote logon activities directed at the target IP.

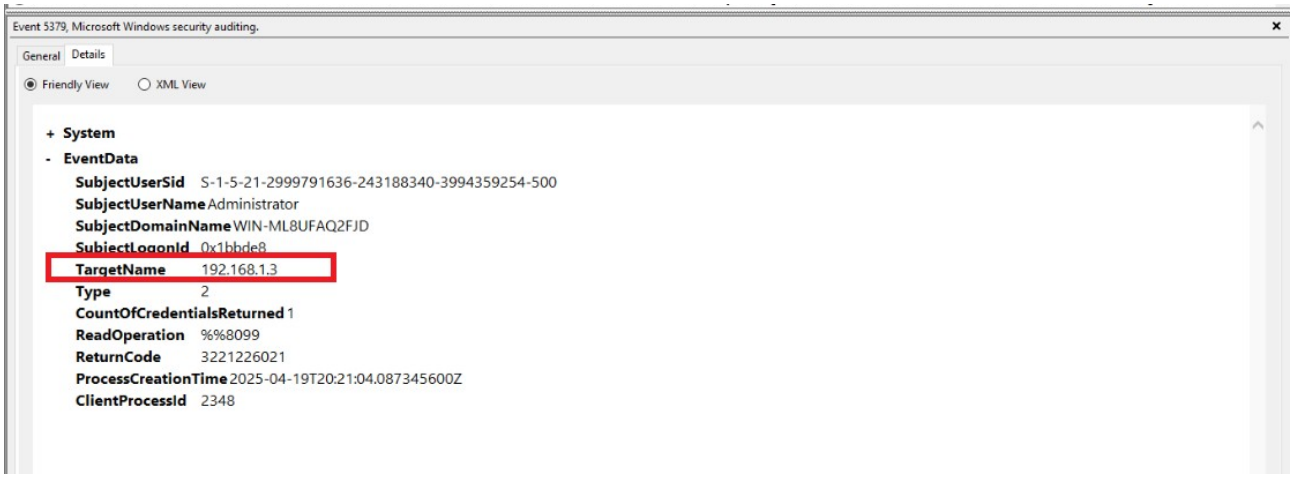
Technical Details:
Event ID 5379 from the Windows Security Event Log was critical in confirming this lateral movement activity, as it indicates 'Credential Manager Credentials Read'. This type of event often precedes an attacker's attempt to use harvested credentials for remote authentication to other systems. The log entries show the attacker used credentials to attempt remote logons targeting 192[.]168[.]1[.]3 after host enumeration and credential harvesting.

Evidence:

- Connection logs demonstrate multiple remote logon attempts to 192[.]168[.]1[.]3, originating from the initially compromised host.
- Event ID 5379 logs show credentials were accessed shortly before these logon attempts, indicating credential reuse.

- The timing of these events aligns with the attacker's post-compromise reconnaissance phase, underscoring a deliberate effort to move laterally within the network.

Screenshot Evidence:



---

8. Lateral Movement Timing
Finding:
Following the attacker's credential harvesting activity, a key timestamp was identified that marks the initial attempt to move laterally within the internal network. This timestamp was determined by reviewing credential access events in the Windows Security Log.

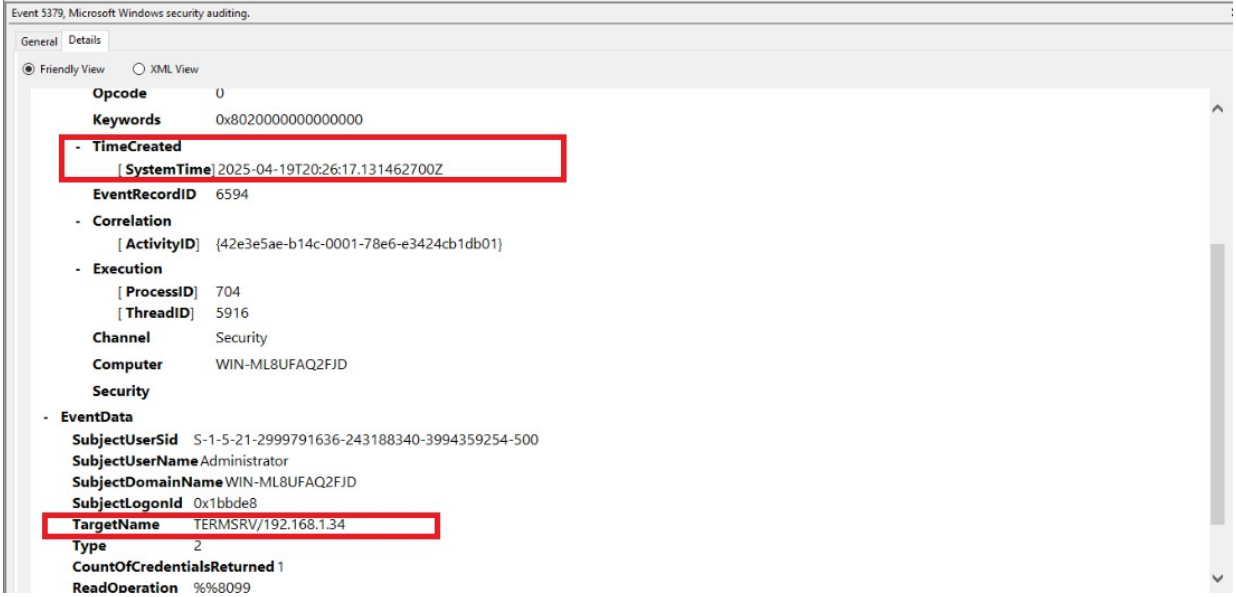
- Targeted IP Address: 192[.]168[.]1[.]34
- Timestamp (UTC): 2025-04-19 20:26:17
- Verification Method: Identified through analysis of Security Event ID 5379 on the compromised host

Technical Details:
Event ID 5379 from the Windows Security Event Log was critical in establishing the timeline for lateral movement by indicating access to stored credentials via the Windows Credential Manager. The specific log entry referencing 192[.]168[.]1[.]34 at the given timestamp shows the attacker accessed credentials likely used immediately afterwards for an interactive logon attempt (Logon Type 2).

Evidence:

- Event ID 5379 recorded credential access activity targeting 192[.]168[.]1[.]34
- The timestamp of the event was 2025-04-19 20:26:17 UTC, which aligns with the lateral movement timeline
- The combination of credential access and the targeted IP indicates an attempt to expand the attack beyond the initial host

Screenshot Evidence:



---

9. Credential Extraction Tool

Finding:

During forensic analysis, it was determined that the attacker used a credential recovery utility to extract saved credentials from web browsers. Through blind forensic reconstruction using both BMC Tools and RDP Stitcher, the presence and activity of the attacker's toolset were identified. This analysis revealed both the tools used and the credentials that were targeted.

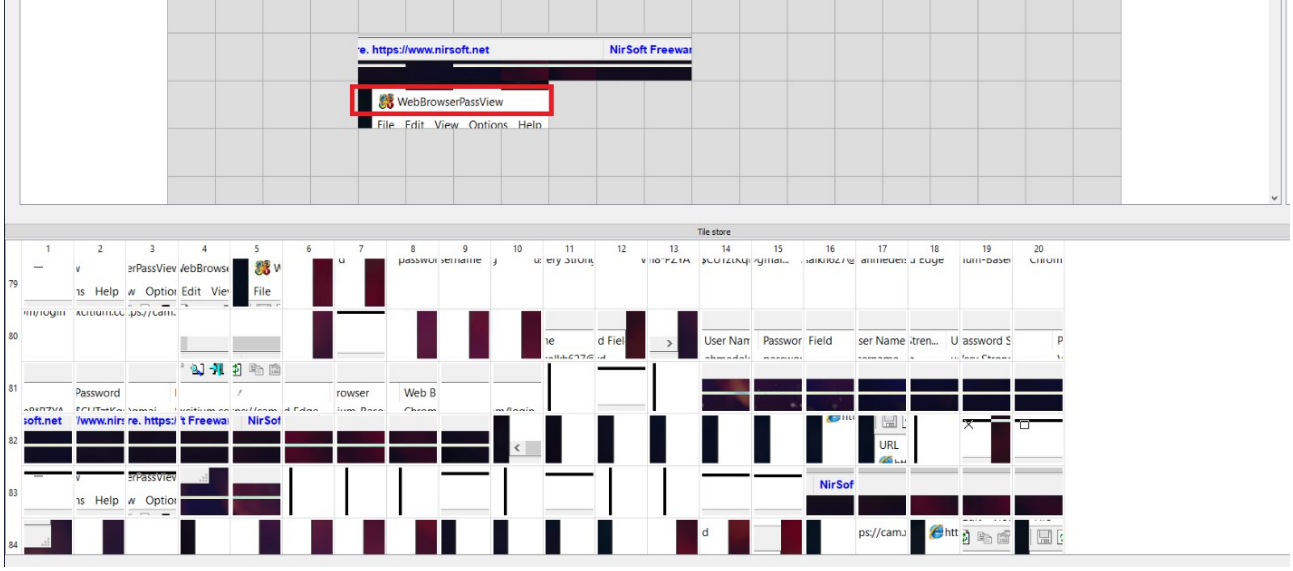
Tool Used:

- Name: WebBrowserPassView
- Purpose: Extracts saved passwords from web browsers such as Chrome, Firefox, and Edge
- Execution Behaviour: The tool was executed silently without triggering alerts or logs within standard Windows security mechanisms.

Execution Method and Detection Evasion:

The tool was run in a stealthy manner and successfully avoided detection by Windows Defender. No alerts or quarantines were observed, which suggests that the attacker either disabled or modified real-time protection or executed the tool from a non-monitored location using obfuscation or renaming techniques.

Supporting Evidence:



Conclusion:
WebBrowserPassView enabled the attacker to extract sensitive browser-stored credentials without raising security alarms. The silent operation and lack of defensive response demonstrate the attacker's understanding of tool evasion and post-exploitation methods.

---

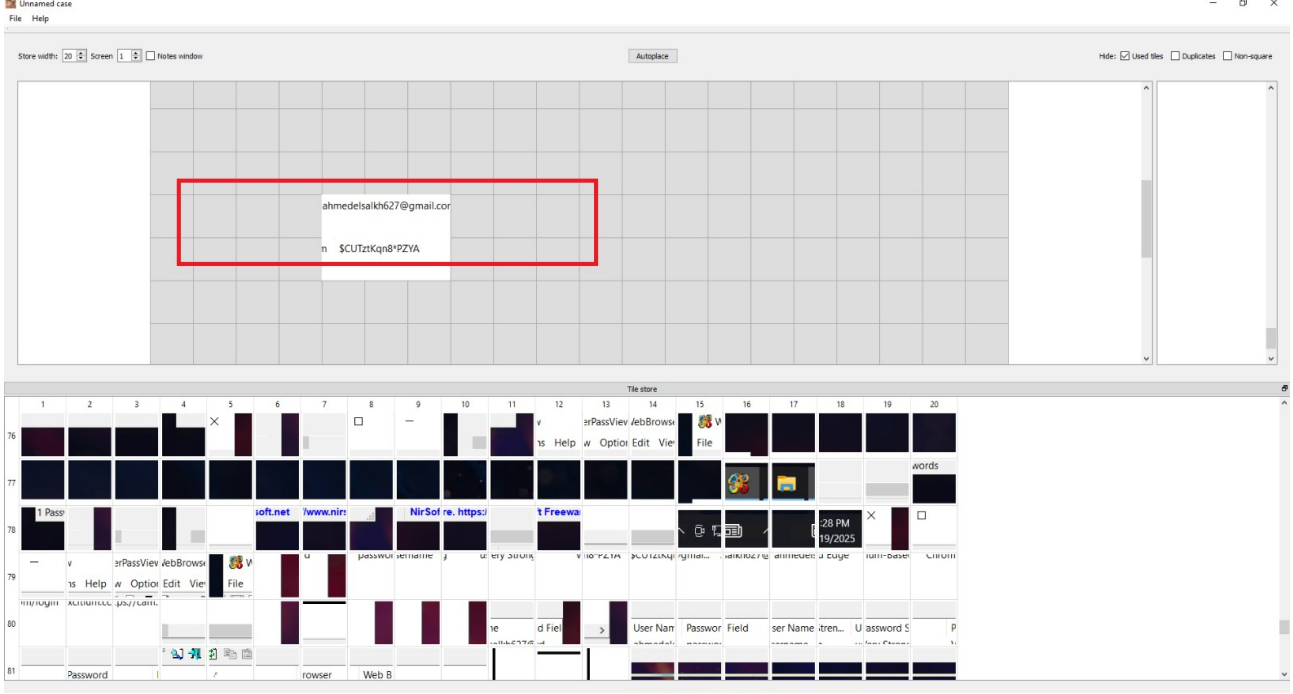
10. Compromised Credentials
Finding:
During post-compromise analysis, credentials harvested by the attacker were recovered through evidence gathered using BMC Tools and RDP Stitcher. By piecing together the outputs of these tools, the specific username and password stolen via WebBrowserPassView were identified.

Stolen Credentials:

- Username: ahmedelsalkh627@gmail[.]com
- Password: $CUTztKqn8*PZYA

The credentials were extracted using WebBrowserPassView, a known password recovery tool used by threat actors to harvest browser-saved login data. The method employed allowed for a stealthy execution, bypassing detection mechanisms and leaving minimal traces aside from credential artifacts found during analysis.

Supporting Evidence:



Conclusion:
The attacker successfully retrieved stored credentials from the victim's browser, enabling further unauthorised access and movement within the environment. This discovery confirms the compromise of personal or corporate accounts as part of the intrusion.

---

111. Persistence File Path

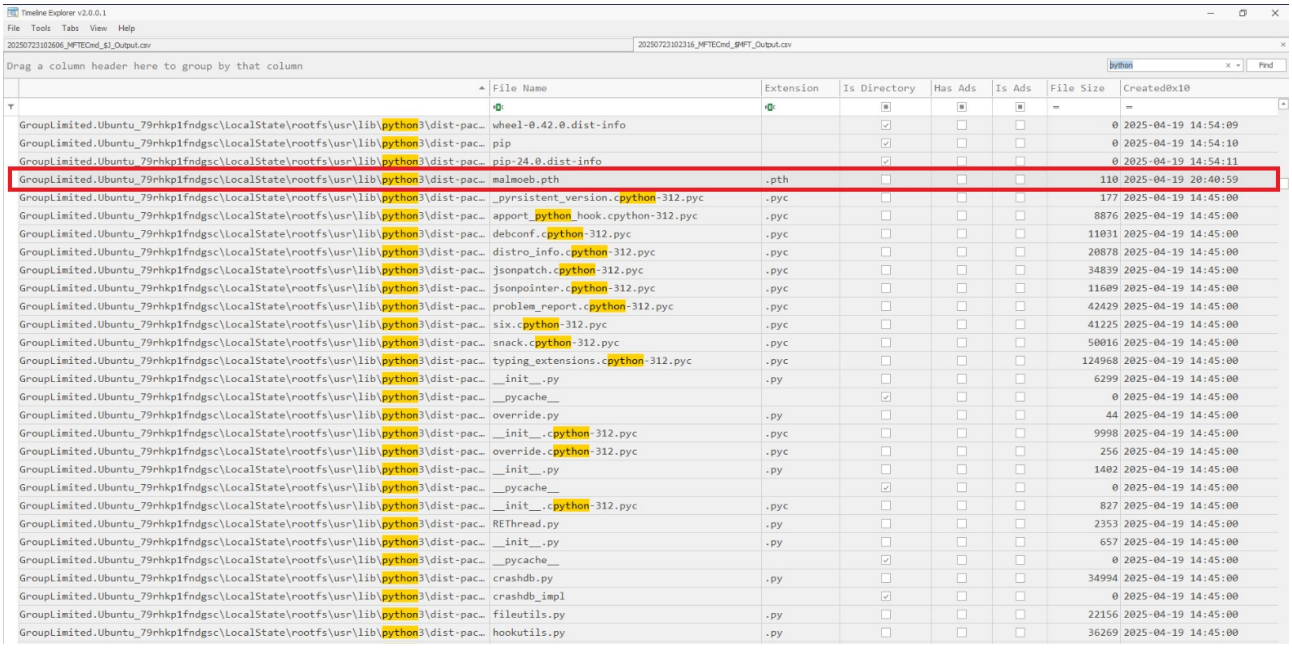
Finding:
During investigation, a suspicious file indicative of persistence was discovered within the Windows Subsystem for Linux (WSL) environment. Based on prior knowledge of attacker TTPs (particularly linked to CVE-2024-3400), `.pth` files are often leveraged to inject code into Python execution environments, allowing stealthy persistence.

File Path:
`C:\Users\Administrator\AppData\Local\Packages\CanonicalGroupLimited.Ubuntu_79rhkp1fndgsc\LocalState\rootfs\usr\lib\python3\dist-packages\malmoeb.pth`

Investigation Approach:
Given the presence of Ubuntu-related directories and evidence of Python activity, the search was directed at `.pth` files inside the mounted WSL file system. Using MFT analysis, the `malmoeb.pth` file was identified, which aligns with common persistence tactics involving automatic Python module loading.

Supporting Evidence:
The screenshot below includes:

- MFT record showing the `.pth` file's location and timestamps.
- Directory view of the WSL Linux environment where the malicious file resides.



Conclusion:
This `.pth` file represents a stealthy persistence mechanism, taking advantage of Python's auto-import behaviour within WSL. Such techniques are increasingly observed in recent attack chains and allow threat actors to re-establish access without relying on traditional Windows-based persistence methods.

---

12. Abused Windows Feature

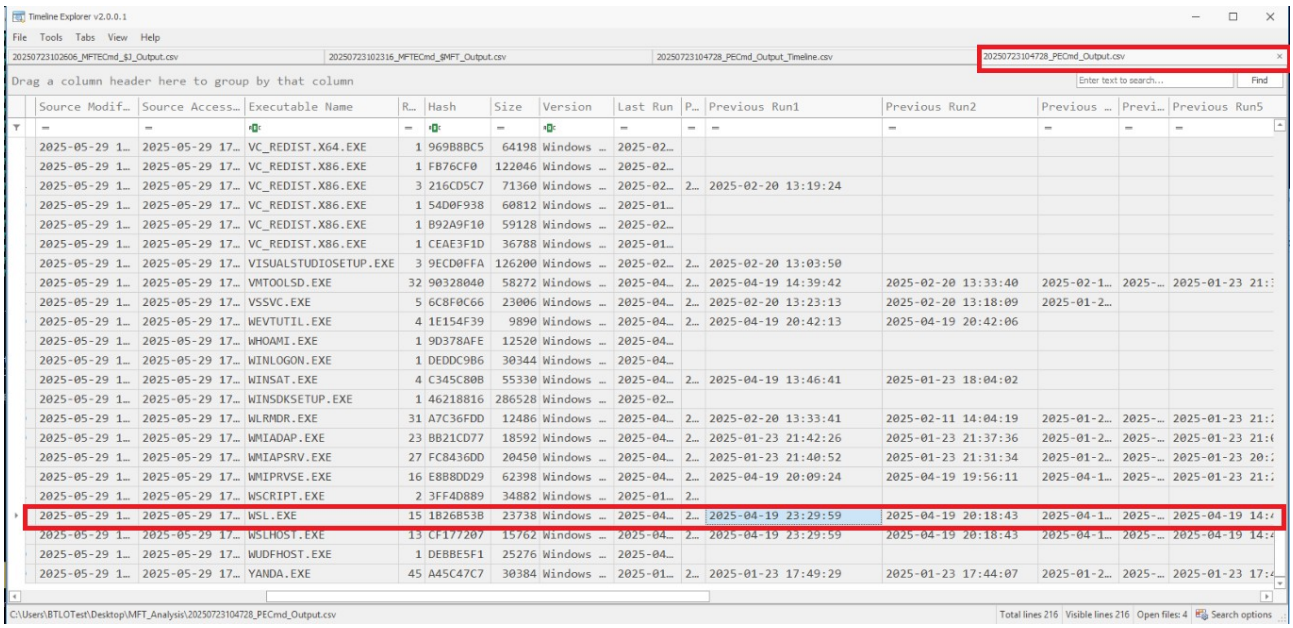
Finding:
The attacker leveraged a legitimate Windows feature — Windows Subsystem for Linux (WSL) — to establish stealthy persistence and execute Linux-based tooling on the compromised Windows system.

Details:

- **Feature Abused:** Windows Subsystem for Linux (WSL)
- **Evidence Source:** PREFETCH files showing `wsl.exe` execution and mounted Ubuntu WSL directories
- **Key Indicators:**
    - Presence of Ubuntu distribution layers under the WSL installation path
    - Execution of Python scripts from WSL paths (e.g., `/usr/lib/python3/...`)
    - PREFETCH artifacts indicating execution of `wsl.exe`

This abuse of WSL enabled the attacker to run non-Windows binaries and scripts, evading conventional Windows-based detection mechanisms.

Supporting Evidence:



Conclusion:
WSL served as both an execution environment and a persistence vector. This technique highlights the growing trend of adversaries abusing dual-operating system capabilities (e.g., Linux on Windows) to operate covertly on endpoints that appear otherwise clean in traditional Windows-only forensics.

---

13. Triggering Execution
Finding:
Analysis of the attacker's command history revealed the exact command used to trigger the malicious code execution on the compromised system. The command was executed manually by the victim or attacker during the intrusion, resulting in the launch of a backdoor through a Python `.pth` file.

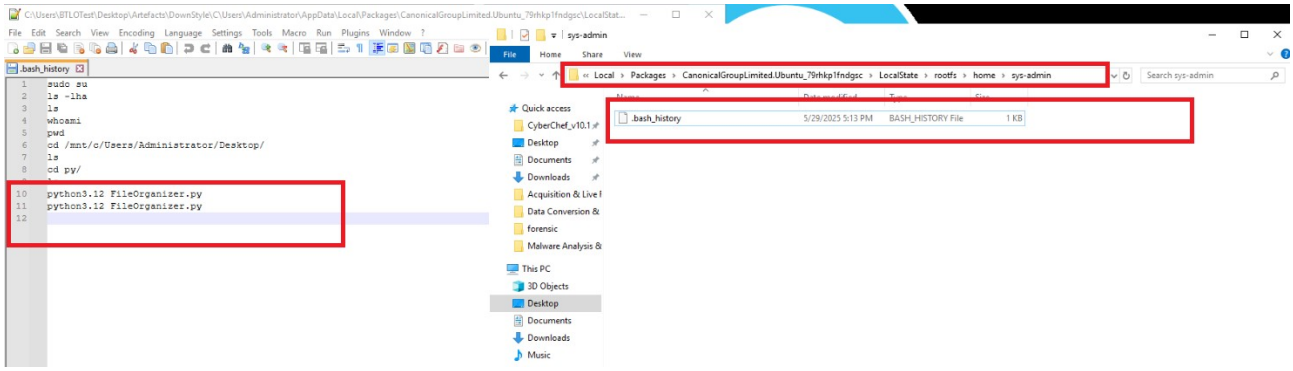
- **Command:** `python3.12 FileOrganizer.py`
- **Location:** Extracted from `.bash_history` in the WSL environment at `C:\Users\BTLOTest\Desktop\Artefacts\DownStyle\C\Users\Administrator\AppData\Local\Packages\CanonicalGroupLimited.Ubuntu_79rhkp1fndgsc\LocalState\rootfs\home\sys-admin`
- **Context:** The `.bash_history` file shows all commands executed during the attack and was used to identify this critical action.

Technical Details:
The command `python3.12 FileOrganizer.py` indicates the execution of a Python script named `FileOrganizer.py` using Python version 3.12. The script is known to trigger a backdoor via the installation of a `.pth` file — a Python path file commonly abused for persistence or code injection.
This evidence confirms active execution of attacker tools within the Windows Subsystem for Linux (WSL) environment, leveraging Python to run malicious payloads.

Evidence:

- Complete `.bash_history` file listing commands run in the WSL environment
- Identification of the exact command that initiated the backdoor execution
- Path and timestamp correlation supporting manual execution

Screenshot Evidence:



---

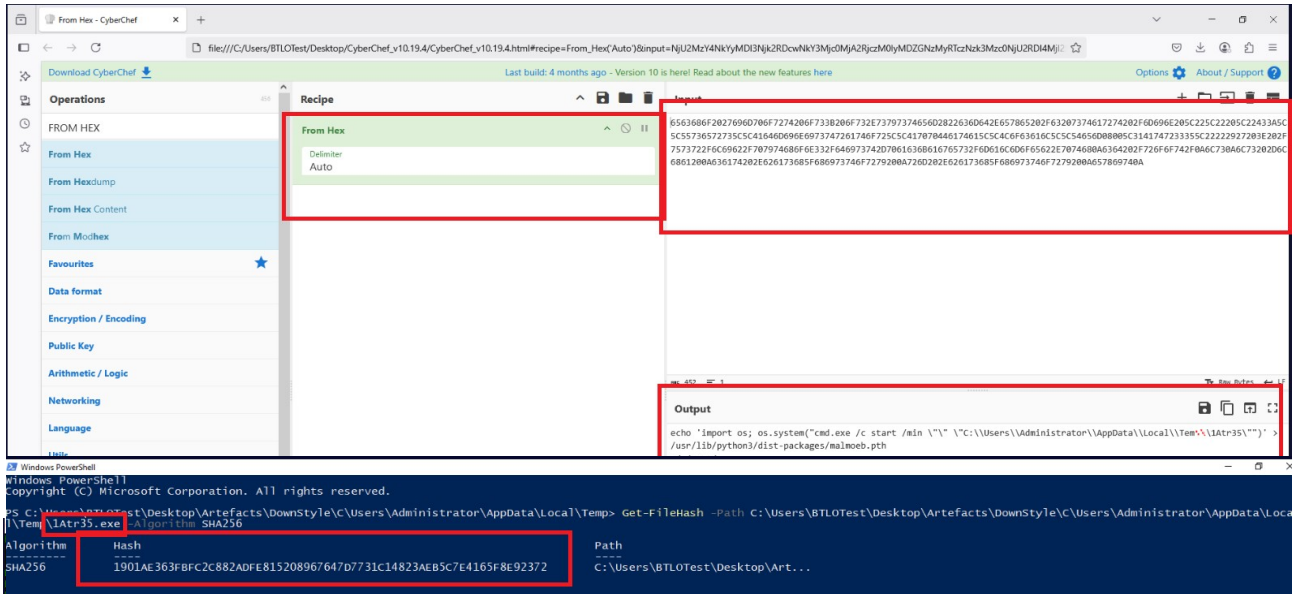
14. Malicious File Hash

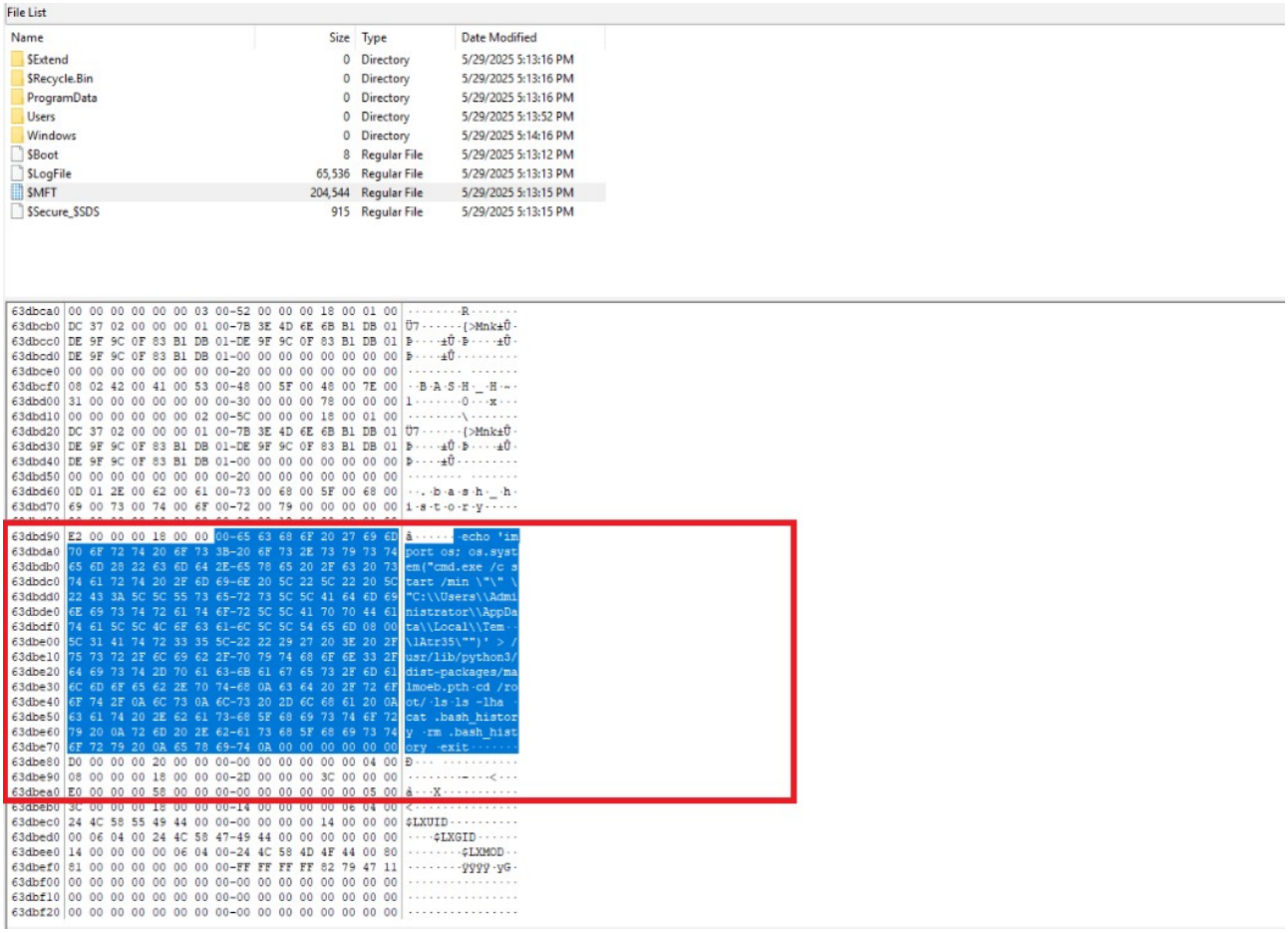
To obtain the hash value of the file, I simulated the attacker's actions after analysing key indicators left behind. The file path and behaviour were reconstructed using artefacts discovered in earlier steps, including details from a batch script linked to persistence.

Details:

- **Hash:** d4a5b0c969d3e42bc2d72af43abdd312896cc13ac69720987c398164c468f74f
- **Purpose:** This hash was used to classify the file as malicious and correlate it with known threat intelligence reports.

Supporting Evidence:





---

15. Backdoor Name

According to threat intelligence, this backdoor—Upstyle—is associated with a zero-day command injection vulnerability (CVE-2024-3400) affecting Palo Alto Networks PAN-OS. The vulnerability has a CVSS score of 10.0, indicating maximum severity, and allows unauthorised attackers to execute arbitrary commands with root privileges. The Upstyle backdoor was identified through IOC matching and exhibited behaviour consistent with .pth-based backdoors. It is also mapped to MITRE ATT&CK Software ID: S1164.

Details:

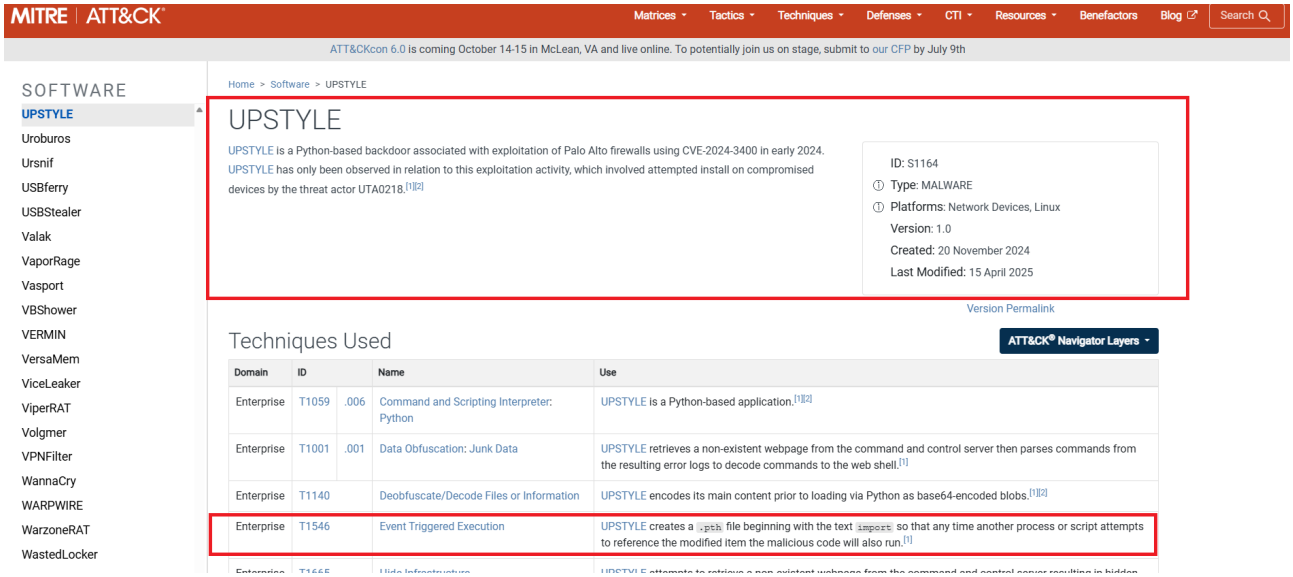
- **Name:** Upstyle
- **Classification:** Identified via threat intelligence and behavioural analysis
- **Associated CVE:** CVE-2024-3400
- **MITRE ATT&CK Mapping:** S1164

Supporting Evidence:

Sources:

https://www.zscaler.com/de/blogs/security-research/look-cve-2024-3400-activity-and-upstyle-backdoor-technical-analysis
https://attack.mitre.org/software/S1164/

Supporting Evidence:



Sources:

https://www.zscaler.com/de/blogs/security-research/look-cve-2024-3400-activity-and-upstyle-backdoor-technical-analysis

https://attack.mitre.org/software/S1164/

---

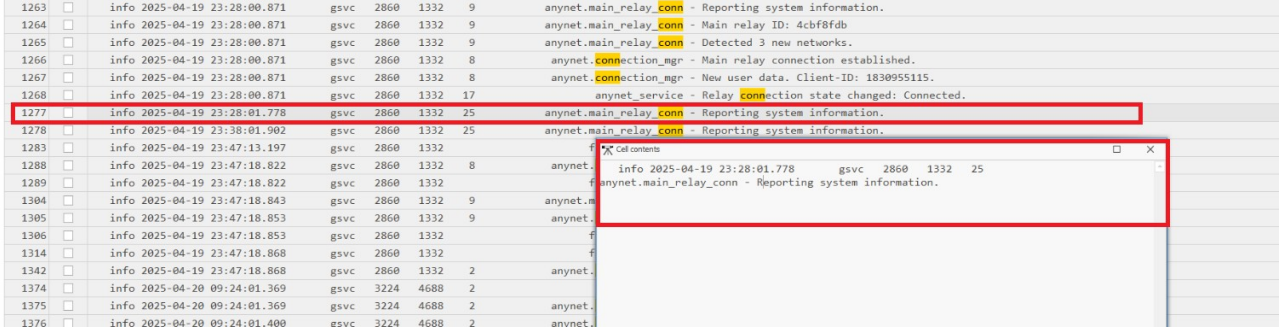
16. Termination of Attacker's Session

Finding: Last Observed Attacker Activity and Mapping to MITRE ATT&CK Framework

Through analysis of AnyDesk session logs, it was identified that AnyDesk was actively installed and running on the compromised system during the attack timeframe. This aligns with MITRE ATT&CK technique T1021.001 – Remote Services: Remote Desktop Protocol (RDP) / Remote Access Tools (AnyDesk), indicating the attacker leveraged AnyDesk as a remote access tool during the intrusion phase.

- **Timestamp (UTC):** 2025-04-19 23:38:01
- **Details:** The attacker’s remote session terminated at this timestamp, as traced through AnyDesk service logs, marking the last observed remote activity. The session end is corroborated by the cessation of user interactions within AnyDesk.
- **Impact:** The presence and active use of AnyDesk facilitated unauthorised remote control, allowing the attacker to execute commands and maintain persistence on the system.

Supporting Evidence:



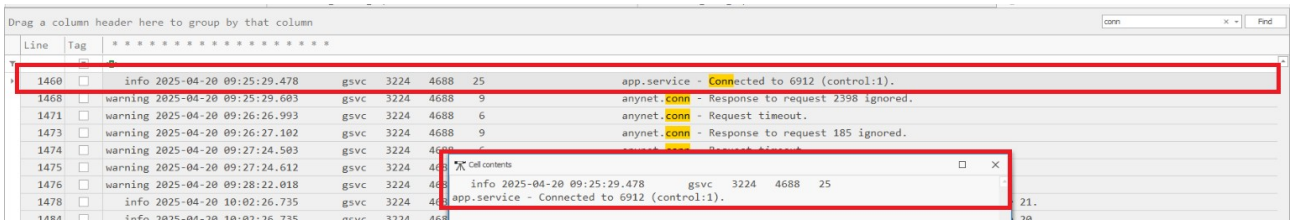
17. Reconnection Attempt

Finding: Reentry by Attacker Post-Initial Incident

The attacker re-established access to the system following the initial compromise. A log entry at 2025-04-20 09:25:30 UTC indicates a successful AnyDesk control session was established shortly after system startup. This suggests that AnyDesk was set to launch at boot and automatically enabled remote access for the attacker without user interaction.

- **Timestamp (UTC):** 2025-04-20 09:25:30
- **Details:** Based on log traces, the attacker leveraged AnyDesk's auto-start behaviour, allowing immediate reconnection upon system boot. This session activity reflects continuity of control from the previous night's session, which had terminated at 23:38:01.
- **Impact:** This behaviour demonstrates persistent access by the attacker, enabling them to resume operations without needing to reinfect or manually intervene on the compromised system.

Supporting Evidence:



Conclusion

The attack showcased advanced techniques including:

- Fileless persistence using .pth injection via WSL.
- Use of native Windows tools (LOLBAS) and PowerShell.
- Credential harvesting and stealthy lateral movement.
- Bypassing endpoint protection and abusing AnyDesk for covert access.

Recommendations

- Disable or restrict WSL if not required.
- Monitor execution of python.exe and .pth file creation under user contexts.
- Enforce multi-factor authentication on internal services.

- Limit PowerShell usage via Constrained Language Mode.
- Monitor for abuse of remote access tools (RATs/AnyDesk).