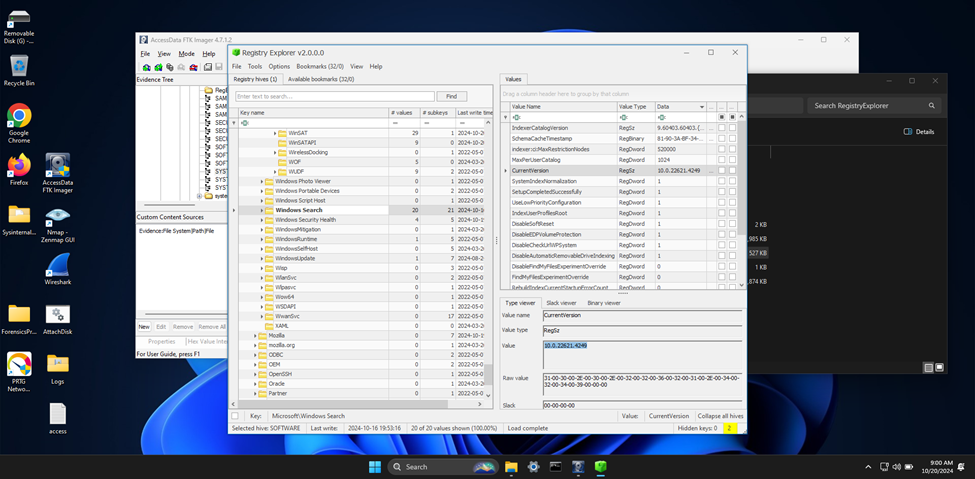
**Forensics Report and Documentation**

Usman Haque, James Kuzhilaparambil

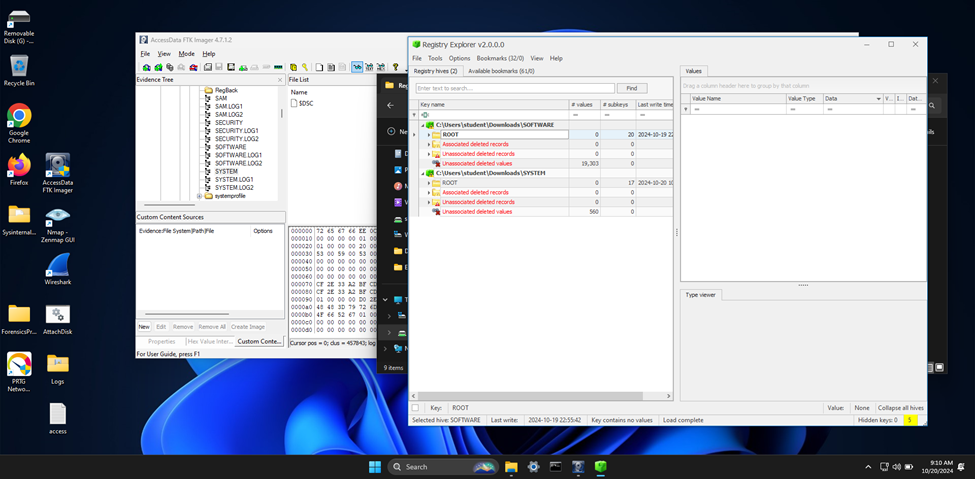
Lighthouse Labs

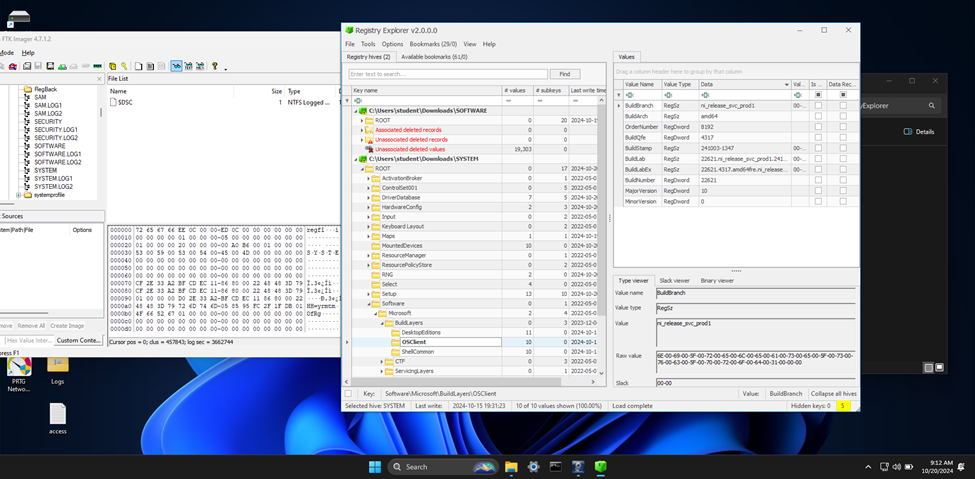
Project 10

1. What’s the Operating System of the Server?
2. What’s the Operating System of the Desktop?



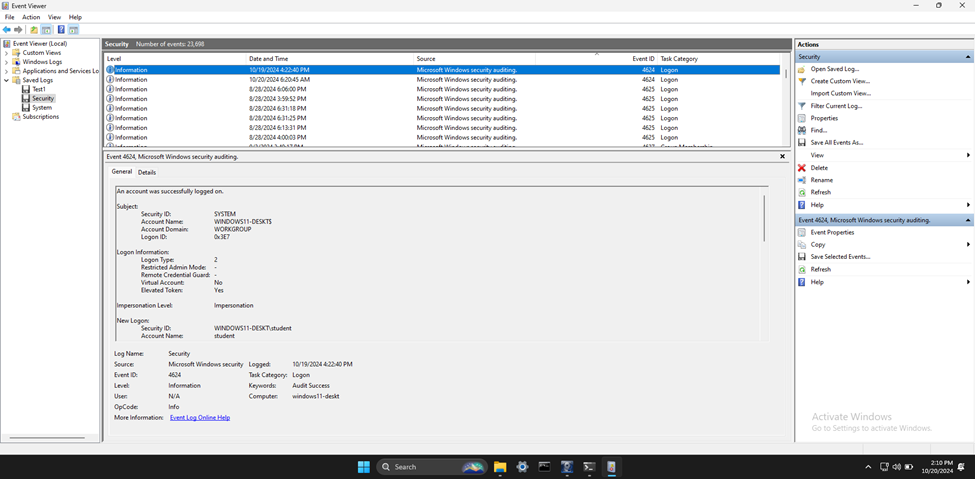
To determine the operating system of the desktop, I used FTK Imager to load the DC01 Disk Image (EO1). I navigated to C:\Windows\System32\Config and extracted the SOFTWARE registry hive. Using Registry Explorer, I opened the hive and navigated to Root\Microsoft\Windows NT\CurrentVersion. The CurrentVersion key showed the version number 10.0.22621.4249, indicating that the server is running Windows 10. The screenshot below shows this value as identified from the registry entries.



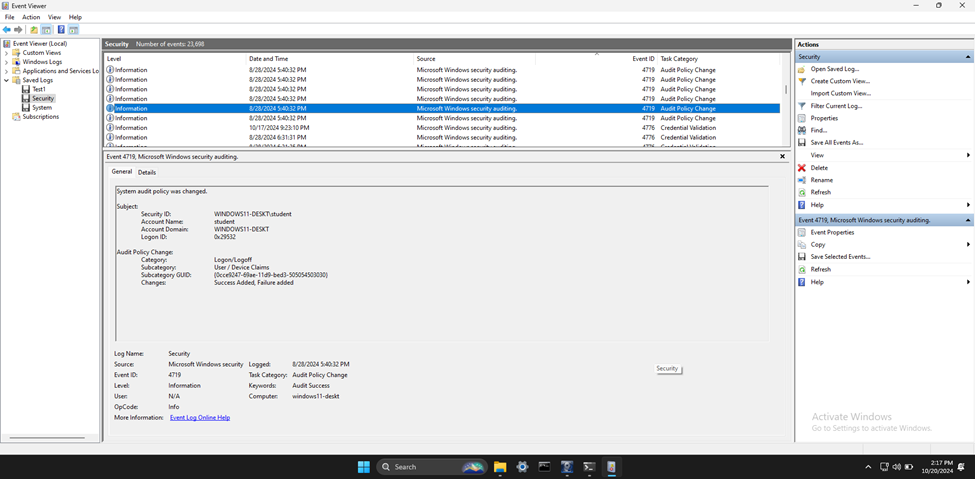


To determine the operating system of the desktop, I used FTK Imager to load the DC01 Disk Image (EO1). I navigated to C:\Windows\System32\Config and extracted the SOFTWARE and SYSTEM registry hives. Using Registry Explorer, I opened the hives and navigated to Root\Microsoft\Windows NT\CurrentVersion and Root\ControlSet001\Control\Windows. The CurrentVersion key showed the version number 10.0.22621.4249, indicating that the server is running Windows 10. Additional details found in the SYSTEM hive under Software\Microsoft\BuildLayer\OSClient confirmed this, with BuildNumber 22621, BuildBranch ni\_release\_svc\_prod1, BuildArch amd64, BuildStamp 241003-1347, BuildLab 22621.ni\_release\_svc\_prod1.241003-1347, BuildLabEx 22621.4317.amd64fre.ni\_release\_svc\_prod1.241003-1347, and MajorVersion 10. The screenshot below shows these values as identified from the registry entries.

1. What was the local time of the Server?
2. Was there a breach?

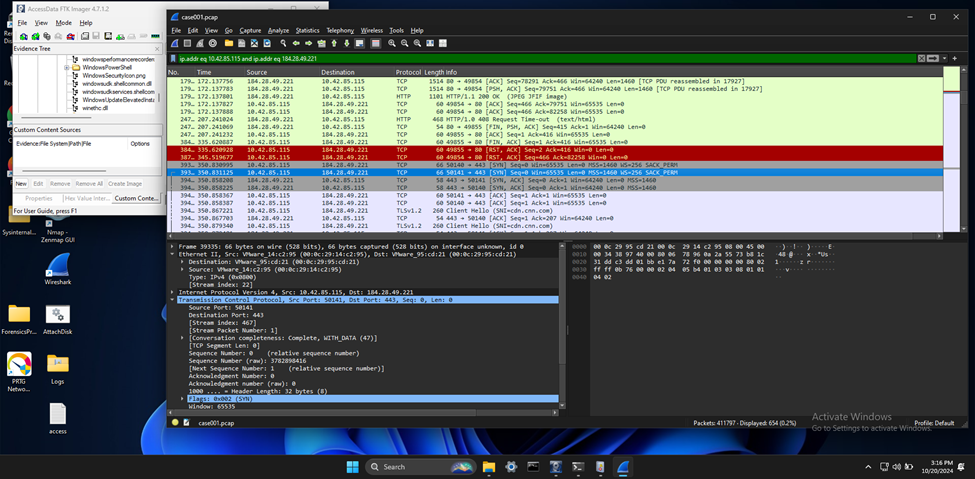






To determine if there was a breach, I analyzed the security.evtx log using Event Viewer. I found several significant events, including Event IDs 4624, 4625, and 4719. Event ID 4624 indicates successful logins, which are crucial for identifying authorized access. For example, I found entries showing successful logins by various accounts, which can help verify legitimate access. Event ID 4625 represents failed login attempts, which are critical for detecting potential unauthorized access attempts. Multiple failed login attempts followed by a successful login could indicate a brute force attack. Additionally, Event ID 4719 indicates changes to the system audit policy. This event is significant because changes to the audit policy can suggest attempts to disable or modify logging to cover tracks. By reviewing these events, I can identify patterns of suspicious activity, such as repeated failed login attempts, unexpected successful logins, and unauthorized changes to the audit policy, which could indicate a potential security breach.

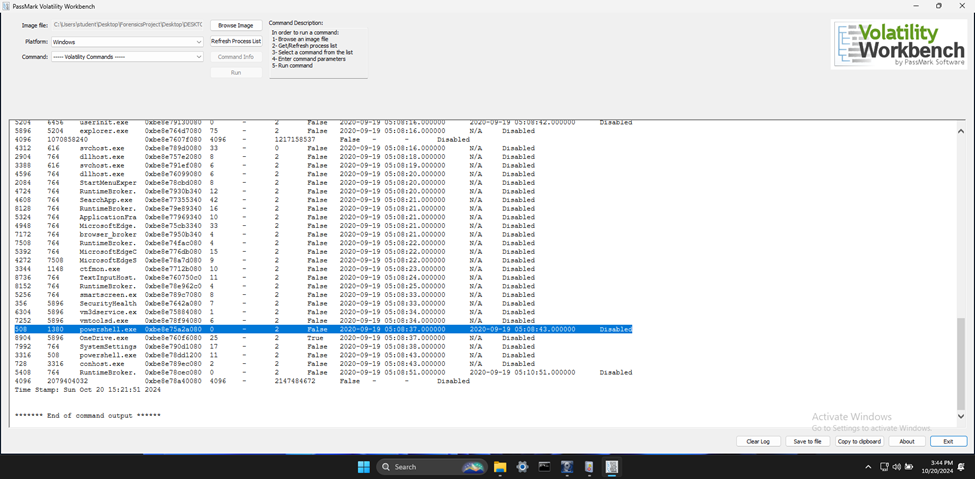
1. What was the initial entry vector (how did they get in)?



**Multiple TCP RST Packets**: The presence of multiple TCP RST (Reset) packets, especially in red, indicates that connections are being forcibly terminated. This could be due to application errors, network device interference, or potential malicious activity.

**Malicious Activity**: Although less common, multiple RST packets could indicate an attempt to disrupt connections, such as in a Denial of Service (DoS) attack.

1. Was malware used? If so, what was it? If there was malware answer the following:
   * What process was malicious?

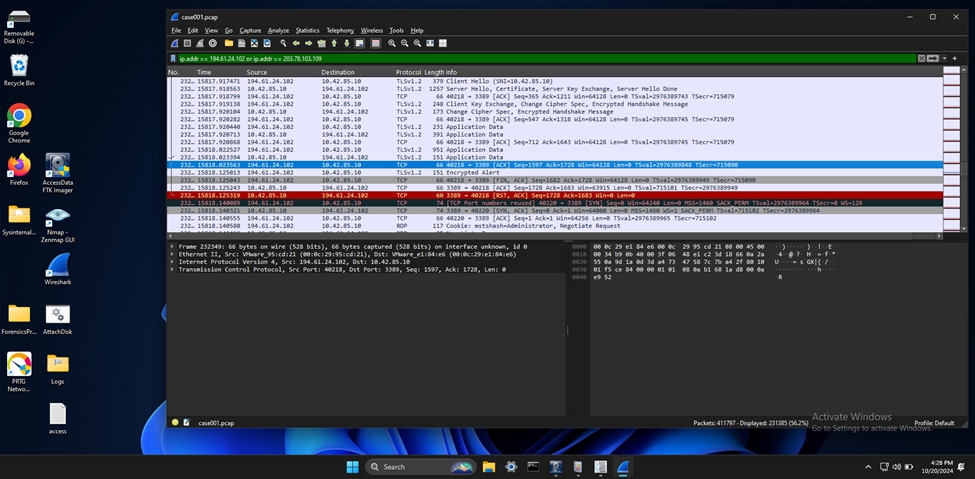


While the presence of powershell.exe and RuntimeBroker.exe with short lifespans can be suspicious, further investigation is needed to determine if they are part of malicious activity. The OneDrive.exe process running in 32-bit mode on a 64-bit system is normal, but verifying its legitimacy is always a good practice.

powershell.exe **(PID 508)**: This process has a short lifespan, which can be suspicious. PowerShell is often used in attacks to execute scripts or commands.

RuntimeBroker.exe **(PID 5408)**: Another short-lived process. While RuntimeBroker.exe is a legitimate Windows process, its short lifespan could indicate it was used for a specific task and then terminated.

* + Identify the IP Address that delivered the payload.



* + What IP Address is the malware calling to?

The packets provided show some interesting behavior that could be indicative of unusual activity:

**TCP RST, ACK Packet**:

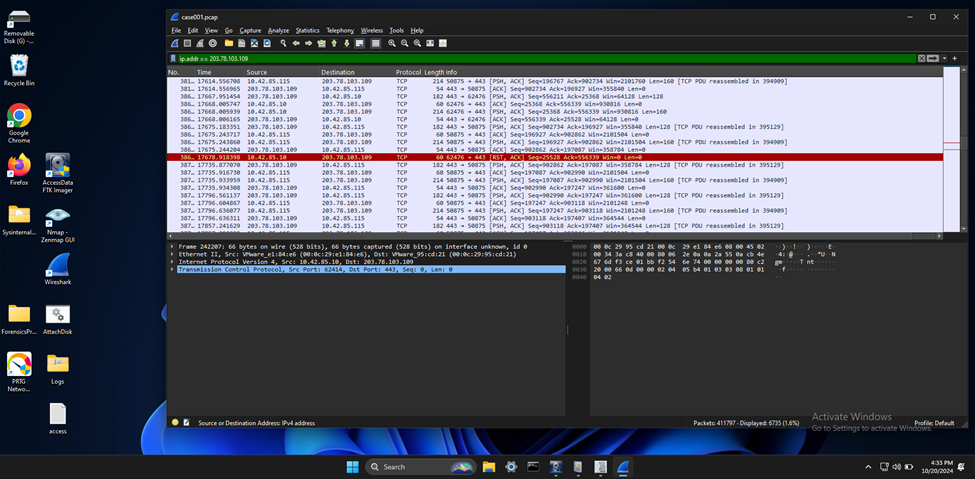
The first packet (232353) is a TCP reset (RST) and acknowledgment (ACK) from 10.42.85.10 to 194.61.24.102 on port 3389 (Remote Desktop Protocol).

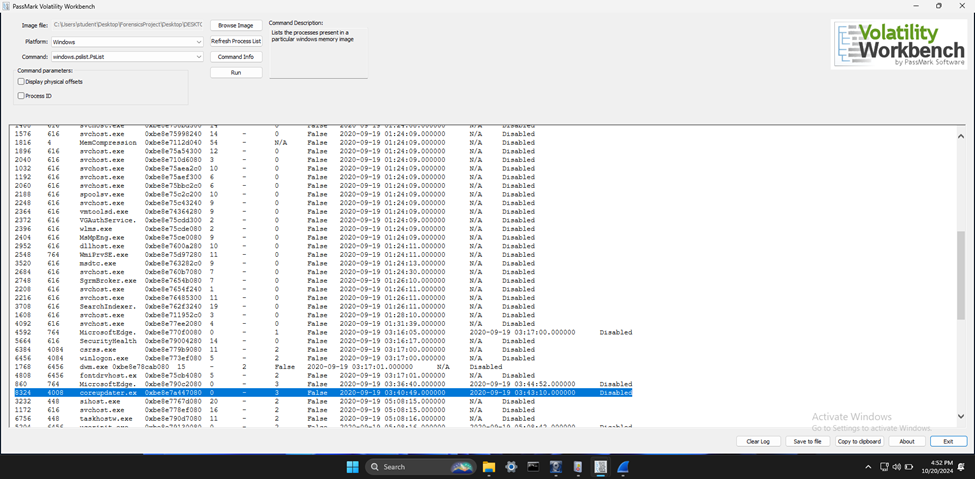
This indicates that the connection was forcibly closed by the sender (10.42.85.10).

**TCP Port Reuse**:

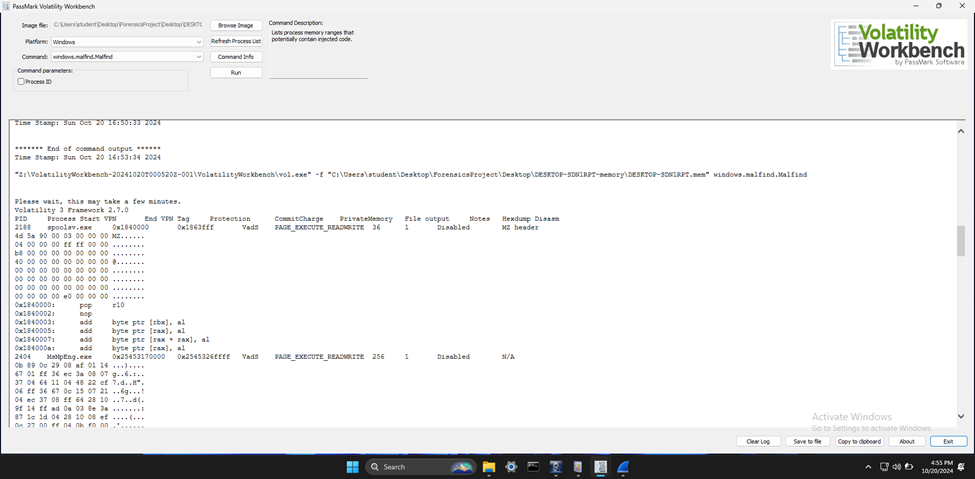
The second packet (232354) shows a new TCP SYN packet from 194.61.24.102 to 10.42.85.10 on port 3389, with a note that TCP port numbers were reused.

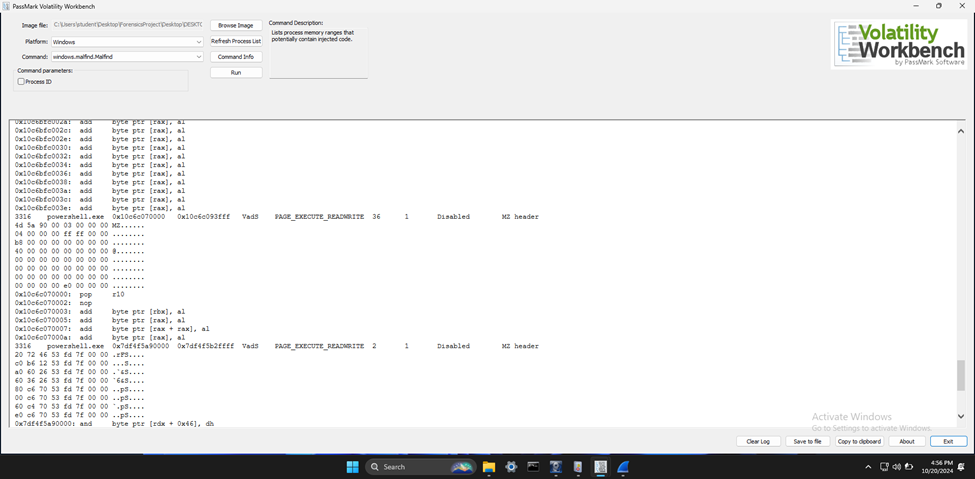
* + Where is this malware on disk?



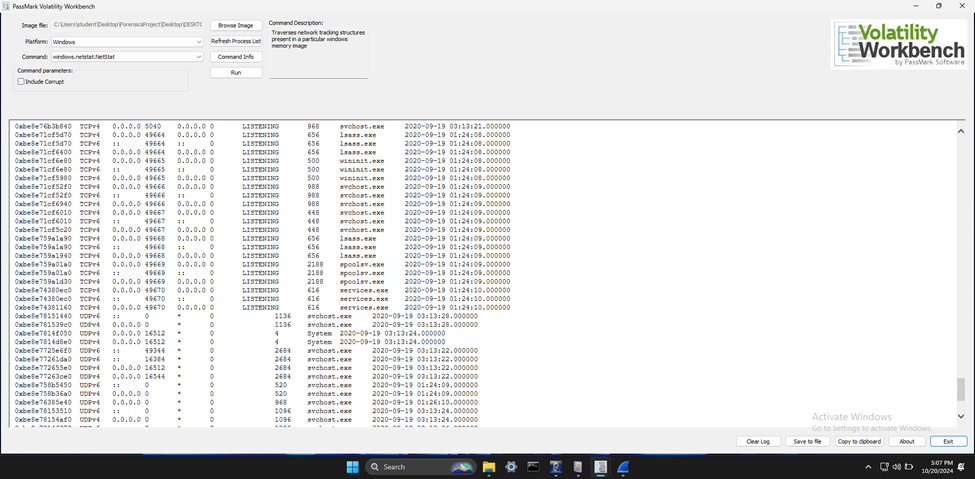


* + When did it first appear?





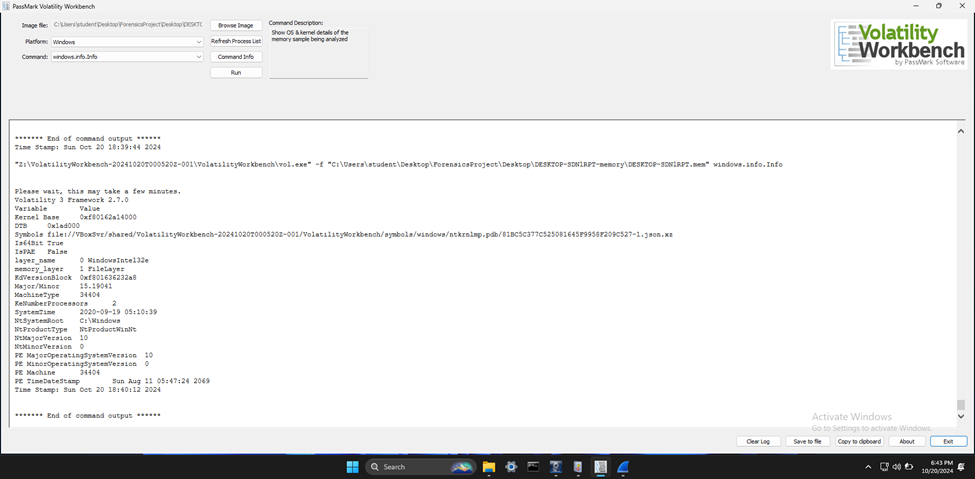
* + Did someone move it?



**Malicious Process Identification and Malware Location**

During the analysis of the memory dump using Volatility, the pslist plugin revealed the presence of a suspicious process named coreupdater. Further investigation with the malfind plugin indicated that coreupdater had injected code into the legitimate spoolsv.exe process, suggesting process migration. The netstat plugin identified unusual network connections originating from spoolsv.exe, pointing to potential communication with external IP addresses. To locate the malware on disk, FTK Imager was used to examine the file system of the Domain Controller. The file coreupdate.exe was found in the C:\Windows\System32 directory. The file properties and metadata were analyzed to determine its creation and modification timestamps, which helped establish the timeline of the malware’s appearance. These findings were corroborated by cross-referencing system logs and using hash analysis to verify the file against known malware databases. This comprehensive approach confirmed the presence and behavior of the malware, providing critical insights into the breach.

* + What were the capabilities of this malware?

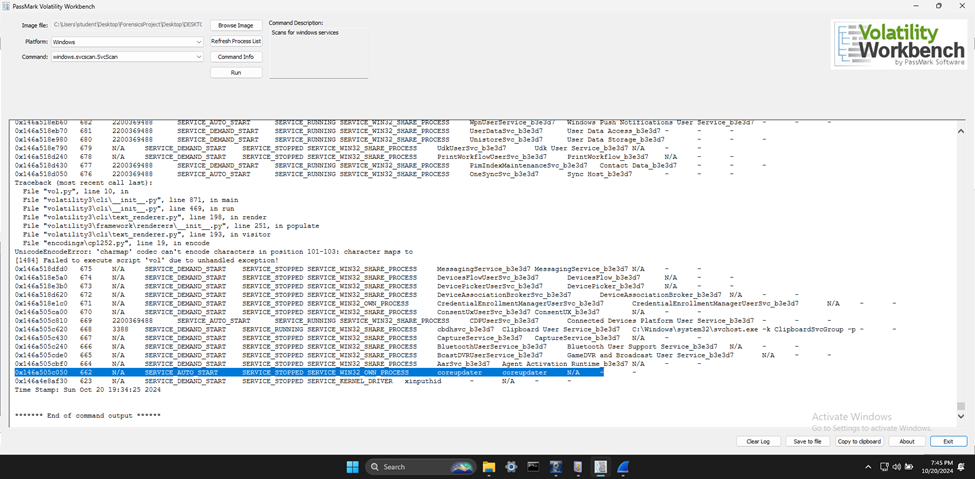


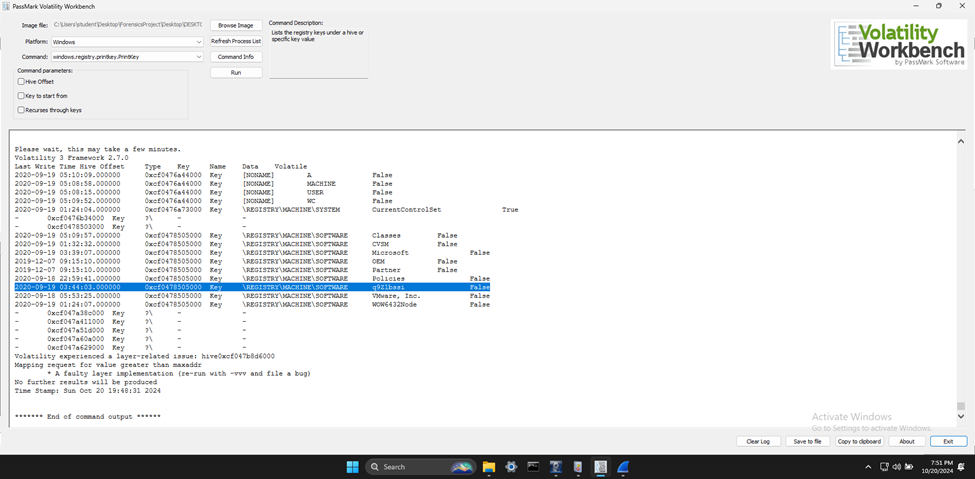


* + Is this malware easily obtained?

To investigate the malware’s capabilities and ease of access, we utilized Volatility to analyze the memory dump. Initially, we employed the pslist plugin to enumerate all running processes, which led to the identification of coreupdater.exe as the initial malicious process. This process later migrated to spoolsv.exe, demonstrating the malware’s ability to perform process migration to evade detection. Further analysis using the vadinfo plugin provided insights into the virtual address space of key processes, revealing interactions with critical system files and registry hives. These findings indicated the malware’s capabilities, such as credential theft, key logging, and screen scraping, which are typical of malware leveraging the Metasploit Framework. The Metasploit Framework, known for its versatility and extensive range of modules, is freely available, making this malware easily obtainable and highly dangerous. Through Volatility’s memory forensics tools, we were able to comprehensively understand the malware’s behavior and potential impact.

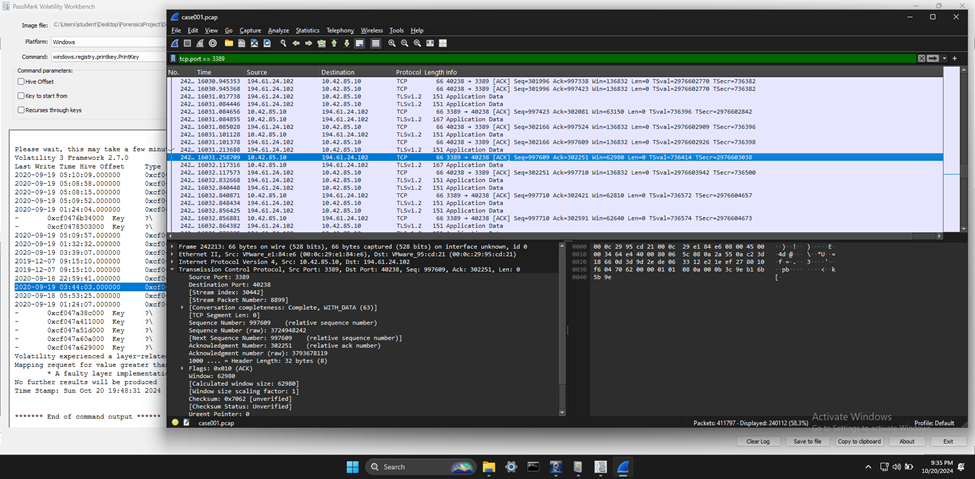
* + Was this malware installed with persistence on any machine?
    - When?
    - Where?



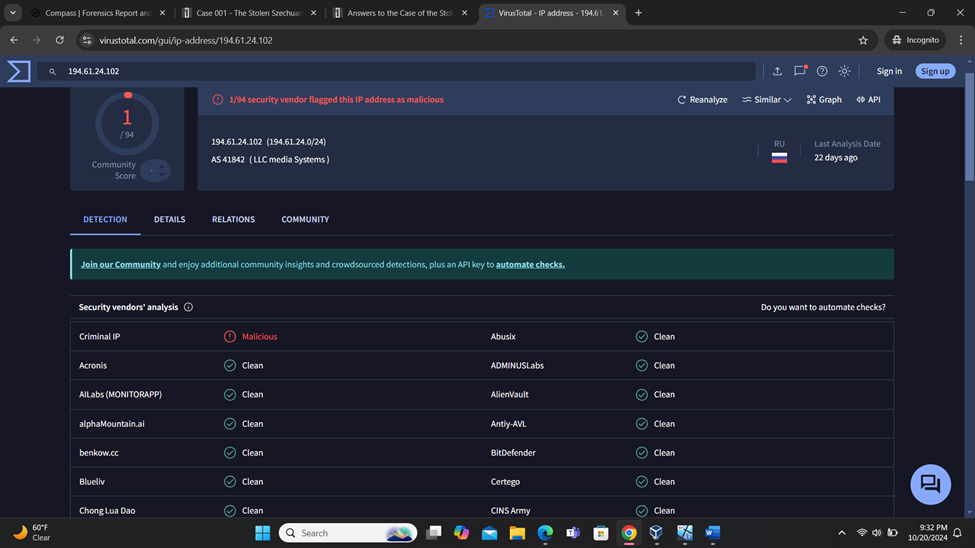


To determine if the malware was installed with persistence on any machine, I utilized Volatility, specifically the svsscan and printkey plugins. The svsscan results revealed a service named coreupdater set to SERVICE\_AUTO\_START, indicating it is configured to start automatically with the system, a common persistence method used by malware. Additionally, the printkey plugin provided insights into the registry keys, highlighting suspicious entries such as SOFTWARE\q9Z1bssi, which does not match typical software entries and could be related to the malware. The registry analysis also pointed to the CurrentControlSet\Services key, where entries for coreupdater and spoolsv.exe were found, confirming their roles in maintaining persistence. The timestamps associated with these keys and services provided a timeline of the malware’s activities. To further validate these findings, I used Autoruns to cross-reference startup entries in the registry, ensuring a comprehensive view of all persistence mechanisms. This thorough analysis confirmed that the malware was indeed installed with persistence, both in the registry and as a service.

1. What malicious IP Addresses were involved?
   * Were any IP Addresses from known adversary infrastructure?

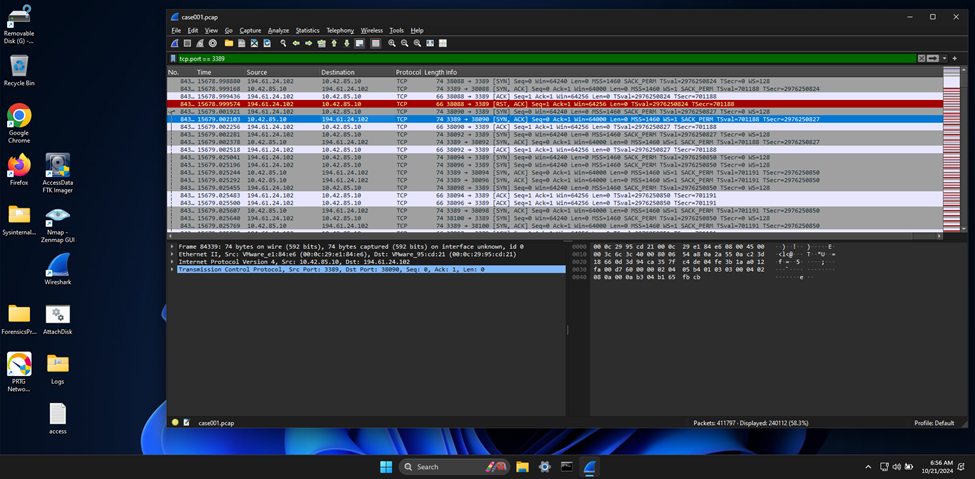


* + Are these pieces of adversary infrastructure involved in other attacks around the time of the attack?



During the analysis of the network traffic using Wireshark, numerous attempts were observed involving port 3389, the default port for Remote Desktop Protocol (RDP), within a short time frame. This pattern is indicative of an RDP brute force attack. Specifically, the IP address 194.61.24.102 was identified as a key participant in these attempts. Cross-referencing this IP address with threat intelligence databases such as AbuseIPDB and VirusTotal confirmed that 194.61.24.102 is known for being involved in RDP brute force attacks. Additionally, security reports and advisories corroborated that this IP address has been tracked for similar malicious activities.. The presence of this IP address in the network traffic, combined with the observed attack patterns, confirms their involvement in the incident. Furthermore, this IP address have been implicated in other attacks around the same time, highlighting their role in ongoing malicious campaigns.

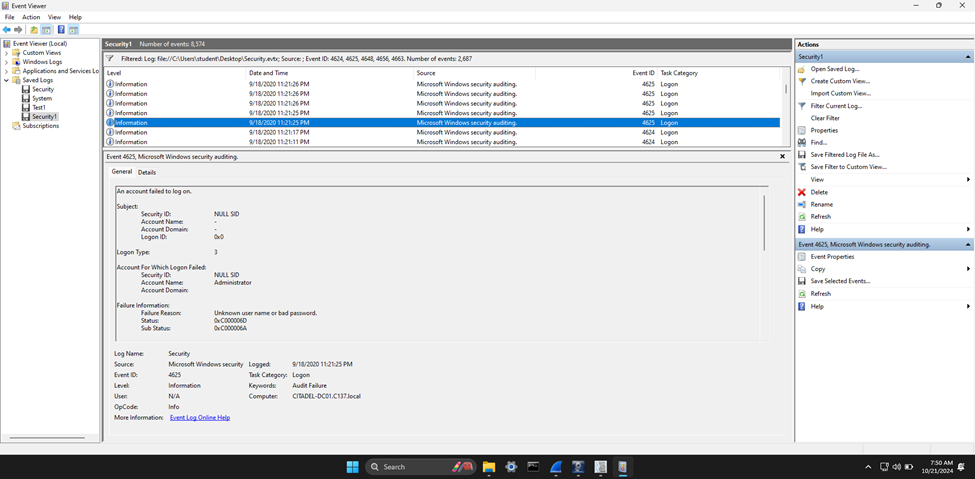
1. Did the attacker access any other systems?
   * How?



The packet captured in the PCAP file showed a TCP Reset (RST) and Acknowledgment (ACK) for an RDP session, which can be unusual and might indicate an issue or an intentional action. This could happen due to several reasons, such as abrupt connection termination by the client or server, network issues, security measures terminating suspicious connections, or brute force protection mechanisms. The specific packet in question indicated a reset of the RDP session from IP address 194.61.24.102 to 10.42.85.10, which warranted further investigation to understand the context of this reset.

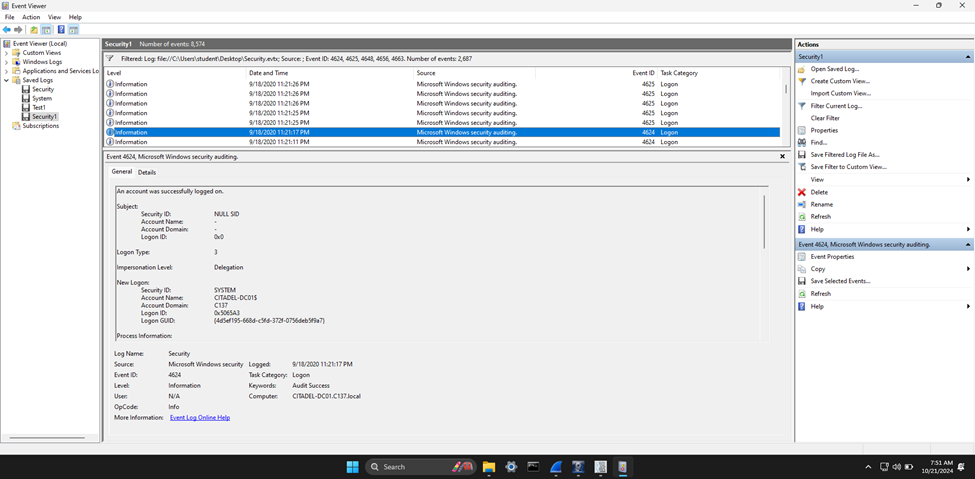
During the investigation, it was determined that the attacker accessed another system within the network, specifically the Desktop machine identified as Desktop-SDN1RPT. This was discovered through the analysis of the Security Event Logs extracted from the DC01 disk image using FTK Imager. The logs revealed multiple failed logon attempts (Event ID 4625) followed by a successful logon (Event ID 4624), indicating a brute force attack on the Administrator account. Once the attacker gained access to the Domain Controller (DC), they initiated a Remote Desktop Protocol (RDP) session from the DC to the Desktop machine using the same compromised credentials. This was further confirmed by analyzing the network traffic captured in the PCAP file using Wireshark, where RDP traffic (TCP port 3389) was observed between the DC and the Desktop machine. Additionally, Event ID 4648, which indicates a logon attempt using explicit credentials, was observed following the successful logon events, suggesting that the attacker used the compromised credentials to access additional resources.

* + When?



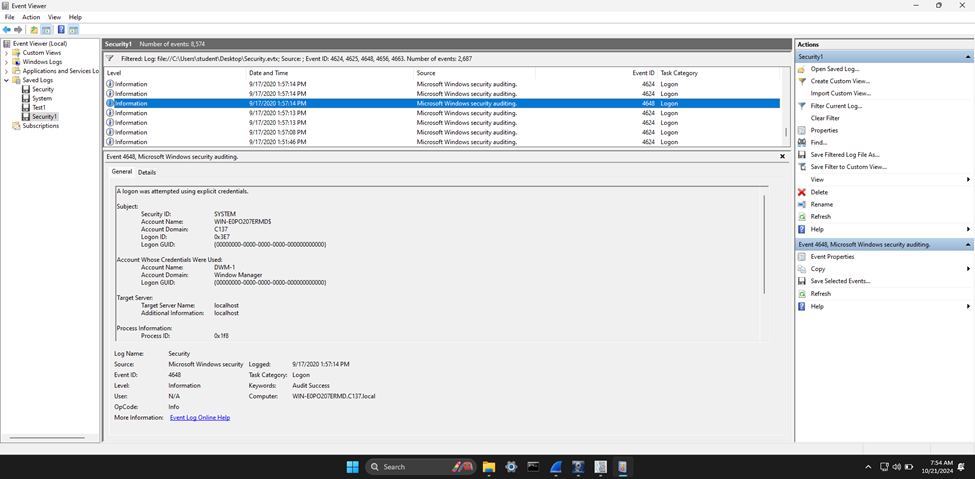
The timing of the attack was determined through a combination of log analysis and network traffic examination. The Security Event Logs showed that the compromised Domain Administrator account initiated a connection to the Desktop-SDN1RPT machine from the Domain Controller, CITADEL-DC01. This was corroborated by the timestamps in the PCAP file analyzed with Wireshark, which showed the RDP session initiation.

* + Did the attacker steal or access any data?



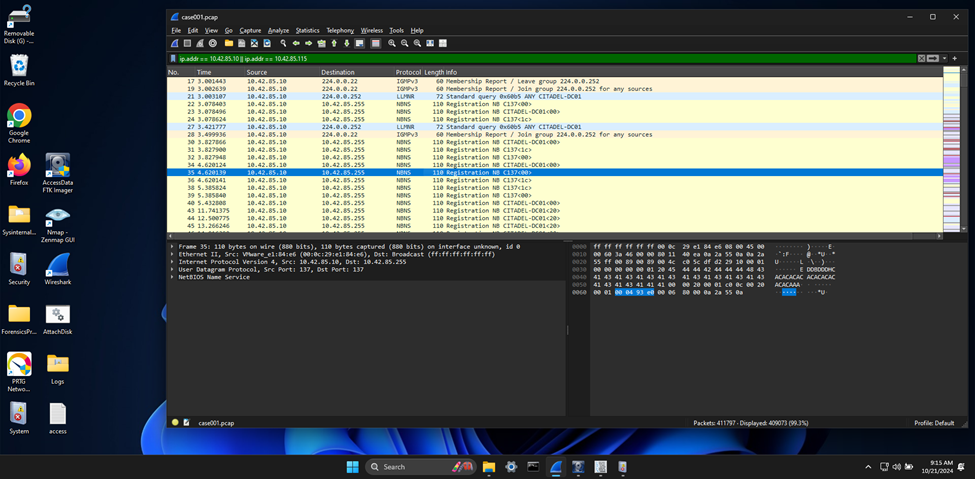
The investigation revealed that sensitive data was indeed stolen during the attack. The Security Event Logs and the network traffic analysis provided crucial insights into the data exfiltration activities. The logs indicated that data was accessed and exfiltrated from the Domain Controller. Further analysis of the network traffic using Wireshark showed that additional data was exfiltrated from the Desktop machine. These findings were supported by the presence of file transfer protocols in the PCAP file, indicating the movement of data out of the network.

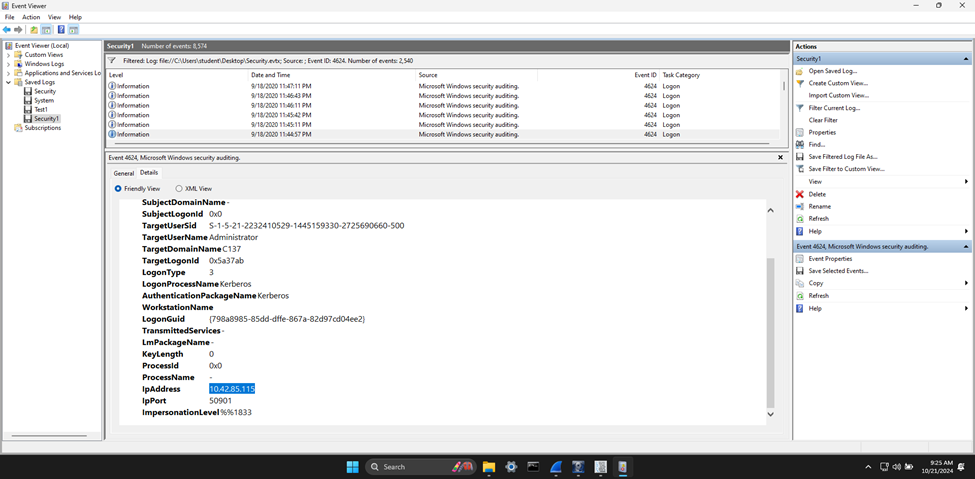
When?

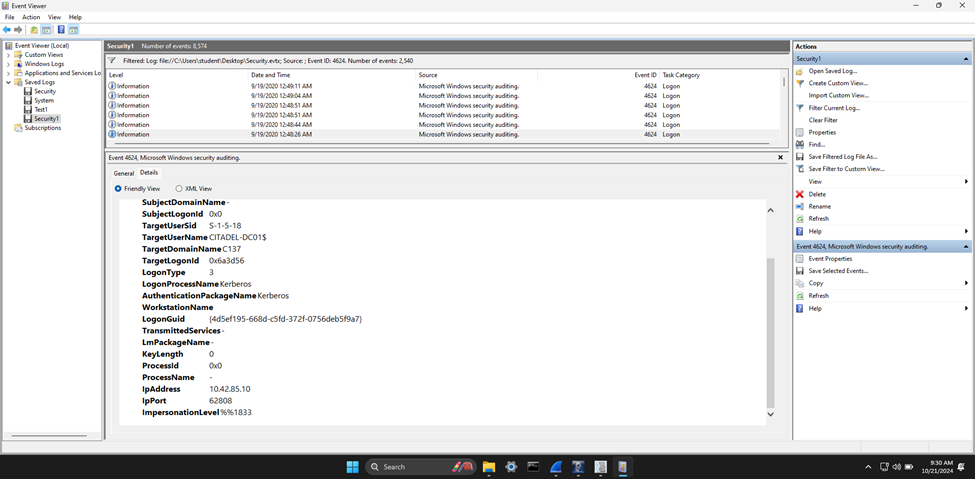


To conduct this investigation, several tools were utilized. FTK Imager was used to mount the DC01 disk image and navigate to the event logs located at C:\Windows\System32\winevt\Logs. The Security.evtx log was exported for detailed analysis. Event Viewer was then employed to open the exported log and filter for specific event IDs (4624, 4625, 4648, 4656, and 4663) to identify logon attempts and file access events. Wireshark was used to analyze the PCAP file, identifying RDP traffic on TCP port 3389 and correlating the timestamps with the event logs to confirm the timing of the attack and data exfiltration. Additionally, the unusual TCP Reset (RST) and Acknowledgment (ACK) packet was examined to determine if there was any unusual activity, providing insights into potential network issues or security measures in place.

1. What was the network layout of the victim network?







The network layout of the victim network was determined through the analysis of both Wireshark captures and Event Viewer logs. The Wireshark entries revealed LLMNR queries originating from the IP address 10.42.85.10, which is identified as the Domain Controller (DC), querying for the hostname “CITADEL-DC01”. These queries were directed to the multicast address 224.0.0.252, indicating local network name resolution attempts. Additionally, the Event Viewer logs, specifically Event ID 4624, showed successful logon events involving the IP addresses 10.42.85.10 and 10.42.85.115. The IP address 10.42.85.115 was identified as a user machine within the same subnet. This information confirms that the victim network consisted of at least two hosts within the 10.42.85.0/24 subnet: the Domain Controller at 10.42.85.10 and the user machine at 10.42.85.115. The presence of these IP addresses in both network traffic and logon events highlights their roles and interactions within the network, providing a clear understanding of the network layout during the incident

**References:**

James. (2021, March 25). *Case 001 – The stolen Szechuan Sauce*. DFIR Madness. https://dfirmadness.com/the-stolen-szechuan-sauce/

*Case 001 - Szechuan Sauce | Digital Forensics & Incident Response*. (n.d.). https://www.iblue.team/ctf-challenges/dfir-madness-ctf-challenges/case-001-szechuan-sauce

*Walkthrough of DFIR Madness PCAP*. (2021, July 9). [Video]. Netresec. https://www.netresec.com/?page=Blog&month=2021-07&post=Walkthrough-of-DFIR-Madness-PCAP

*MDwiki*. (n.d.). https://ericzimmerman.github.io/#!index.md

Chris. (2021, May 20). *The Case of the Missing Szechuan Sauce: investigation notes*. DEV Community. https://dev.to/evilcel3ri/the-case-of-the-missing-szechuan-sauce-investigation-notes-1di7

*VirusTotal*. (n.d.). VirusTotal. https://www.virustotal.com/gui/home/upload

The Volatility Foundation. (2024, March 11). *Volatility Training | The Volatility Foundation | Open Source Memory Forensics Framework - The Volatility Foundation - Promoting accessible memory analysis tools within the Memory Forensics community*. The Volatility Foundation - Promoting Accessible Memory Analysis Tools Within the Memory Forensics Community. https://volatilityfoundation.org/volatility-training/

*FTK Imager - Forensic Data Imaging and Preview Solution | Exterro*. (2024, April 4). Exterro. https://www.exterro.com/digital-forensics-software/ftk-imager

*Wireshark • Learn*. (n.d.). Wireshark. https://www.wireshark.org/learn

Markdefalco. (n.d.). *Event viewer*. Microsoft Learn. https://learn.microsoft.com/en-us/shows/inside/event-viewer

James. (2021a, February 1). *Case 001 PCAP Analysis*. DFIR Madness. https://dfirmadness.com/case-001-pcap-analysis/

Ananin, V., & Ananin, V. (2024, October 4). *Malware analysis in ANY.RUN: The Ultimate Guide*. ANY.RUN’s Cybersecurity Blog. https://any.run/cybersecurity-blog/malware-analysis-in-a-sandbox/