Case Study 3 – Investigating a Cyber Attack

CYSE2002_2237: LawForensics & Investigation

Prof. Alain N'Dalla

Investigation Team

Himesh Girish Patel	himeshgirishpatel@loyalistcollege.com	500209946
Harshkumar Lilabhai Raval	harshkuamrlilabha@loyalistcollege.com	500216597
Ugochukwu Henry Nwosu	ugochukwunwosu@loyalistcollege.com	500194968
Amarjeet Kaur	amarjeetkaur@loyalistcollege.com	500211088
Manju Rani	manjurani@loyalistcollege.com	500203949
Mandeep Kaur	mandeepkaur61@loyalistcollege.com	500211149
Saravjeet Kaur	saravjeetkaur@loyalistcollege.com	500212002
Gagandeep Kaur	gagandeepkaur65@loyalistcollege.com	500202277
Avi Gandotra	avigandotra@loyalistcollege.com	500212277
Gurvinder Singh	gurvindersingh8@loyalistcollege.com	500204811

Executive Summary:

An analysis was conducted on a compromised system – WIN7-CFO. The system is running a Windows 7 Pro with Service Pack 1 installed on Thursday, October 4, 2018, 5:23:54 AM (UTC). The attack occurred on October 9,2018. The state of the memory during the time of compromise was captured using FTK Imager. A static analysis was performed on the captured memory using tools, Volatility, Registry Explorer, Event Log Explorer and VirusTotal.

Key Findings:

1. Malicious File Discovery: A malicious file, Enhancement Tablxslx.exe , was uncovered during the analysis. This file was downloaded by the user shauser by being a victim to a phishing email indicating that the user was unknown of the nature of the file.
2. Credential Compromise: User credentials on the system were compromised during the incident. The attacker exploited the usage of the outdated authentication system Wdigest, which stores user credentials in plain text, providing an avenue for unauthorized access and data extraction.
3. Remote Desktop Connections: The attacker employed remote desktop connections as a mechanism to transfer the data from the compromised system to their own system.

Vulnerability & Threat:

1. Wdigest: It is an outdated and unsecure authentication system being used on the system. It is a high risk to security as it stores user credentials in plain text. This allowed the attacker to gain credentials of the users on the compromised system.
2. Mimikatz: The analysis identified the presence of Mimikatz on the compromised system. It is a well-known post-exploitation tool specializing in the extraction of credentials, plaintext passwords, and hashes. It poses a severe threat to the system.
3. Single Factor Authentication: The system is only using a single factor authentication system which makes breaking into the system easy once the password is uncovered.

Limitations:

1. Static Memory Dump Constraints: The reliance on a static memory dump provided only a partial view of the incident, limiting the depth of information available for analysis.
2. Static Analysis Tool Constraints: The use of static analysis tools only uncovers parts of the incident which limits the investigation of the attack. This only helps partially solve the problem and only provide limited information for securing the system.

Findings

The investigation began with the analysis of the memory dump file extracted from the device under investigation to look at the state of the computer at that moment. FTK imager a forensic tool used for Registry hive extraction and memory capture was used.

Once the memory dump was captured the investigation began using Volatility2. The first step we took was to find out information about the system. The table below provides system information.

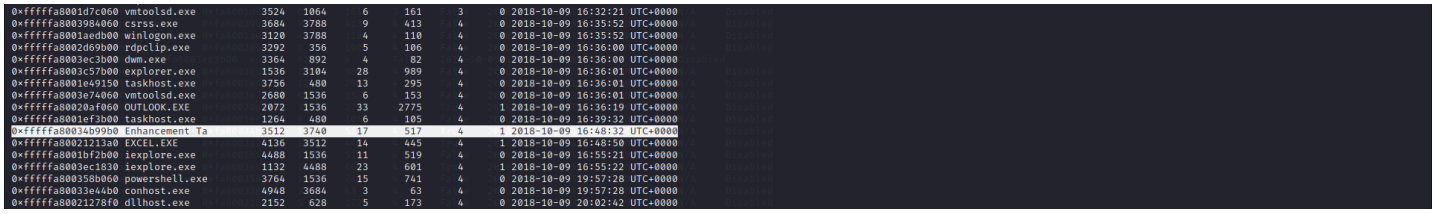
Information Found	
System OS	Windows 7 Pro
Build Number	7601
CSDVersion	Service Pack 1
Installed Date & Time	Thursday, October 4, 2018, 5:23:54 AM (UTC) Wednesday, October 3, 2018, 10:23:54 PM (PST)
Time Zone	Pacific Standard Time
Shutdown Time	Tuesday, October 9, 2018, 3:36:30 AM (UTC) Monday, October 8, 2018, 8:36:30 PM (PST)
Computer Name	WIN7-CFO

The next step we took was to obtain a list of all the running processes when the memory was captured. For this we used the “pslist” plugin of volatility. With the help of this plugin we were able to identify that one such process which is not part of the regular windows process list.

Suspicious Process: Enhancement Tablxslx.exe

PID: 3512

Evidence Image 1:

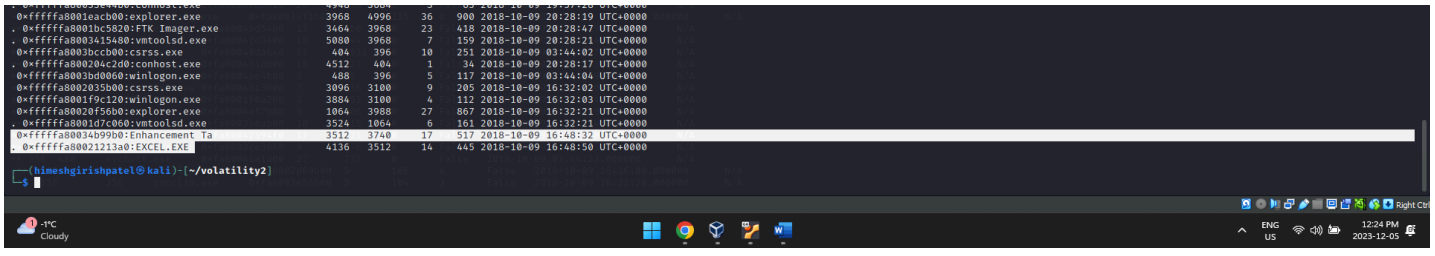


Now we used the “pstree” plugin in volatility to figure out which processes are running under the Enhancement Tablsxls.exe process.

After successfully running the pstree command we found that Microsoft Excel was started in correspondence with the **Enhancement Tablxslx.exe** process.

This helped determine that the **Enhancement Tablxslx.exe** was posing as an **Excel file**.

Evidence Image 2:



Now to further investigate what exactly the suspicious process might be doing we ran the “cmdline” plugin to see what command line arguments associated with the specified process. This information is crucial for understanding what the process was doing at the time of the memory capture.

While it didn't reveal much in terms of commands running or what it might be doing. It did give us information about the location of the file and the user using the file.

Location: C:\Users\shauser\Downloads\ Enhancement Tablxslx.exe

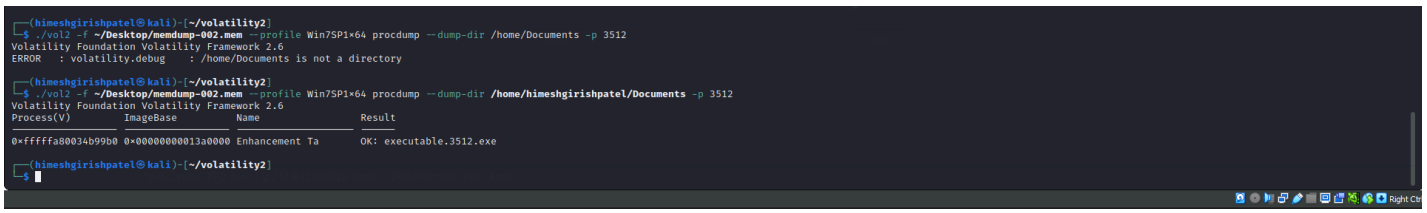
User: shauser

This information that we uncovered will later help us determine the timeline of how this all started when we used registry hives and event logs.

Evidence Image 3:

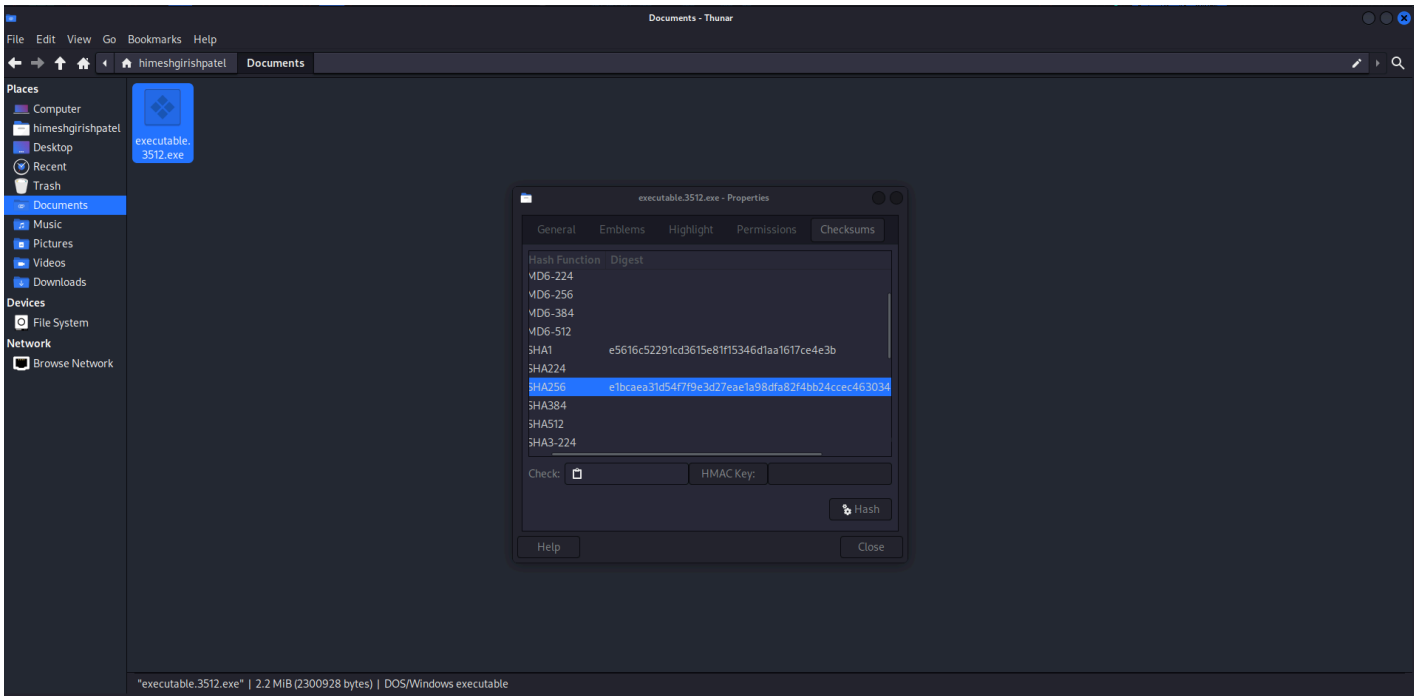
Since we couldn't figure out what the suspicious excel file was doing. We decided to extract it.

Evidence Image 4:



After successfully extracting the file. We took its SHA 256 value and searched it against the VirusTotal database to see if it is a malware file or not.

Evidence Image 5:

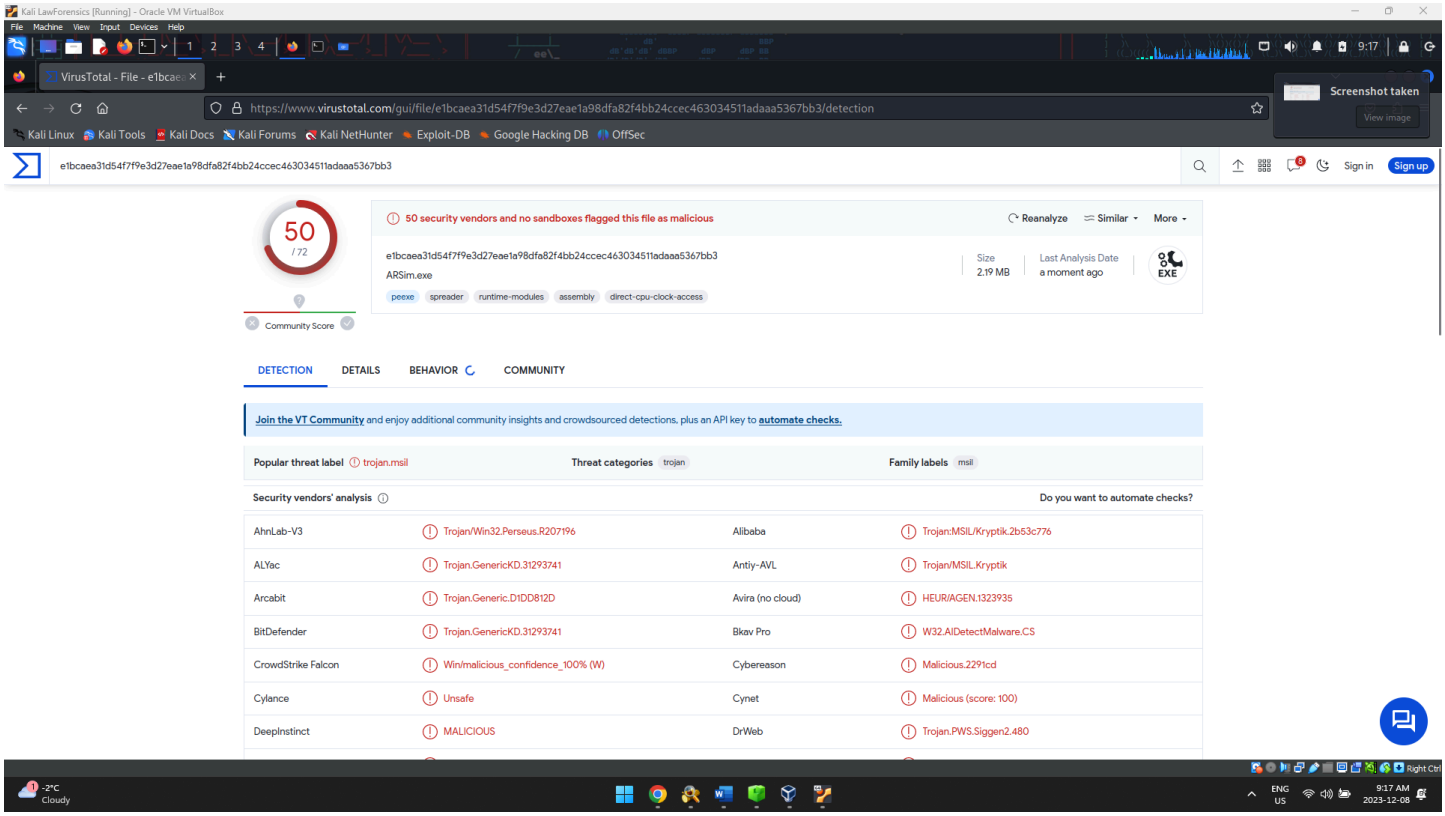


We found that 50 security vendors have flagged this file to be a malicious file and a thorough analysis on the website showed that it runs background commands in powershell to install Mimikatz.

Note: What is Mimikatz ? - Mimikatz is a potent post-exploitation tool that extracts sensitive information, including passwords and Kerberos tickets, from Windows systems' memory. Used for security testing, it

exposes vulnerabilities but also poses a threat if exploited by malicious actors, emphasizing the need for robust cybersecurity practices and defenses against credential theft.

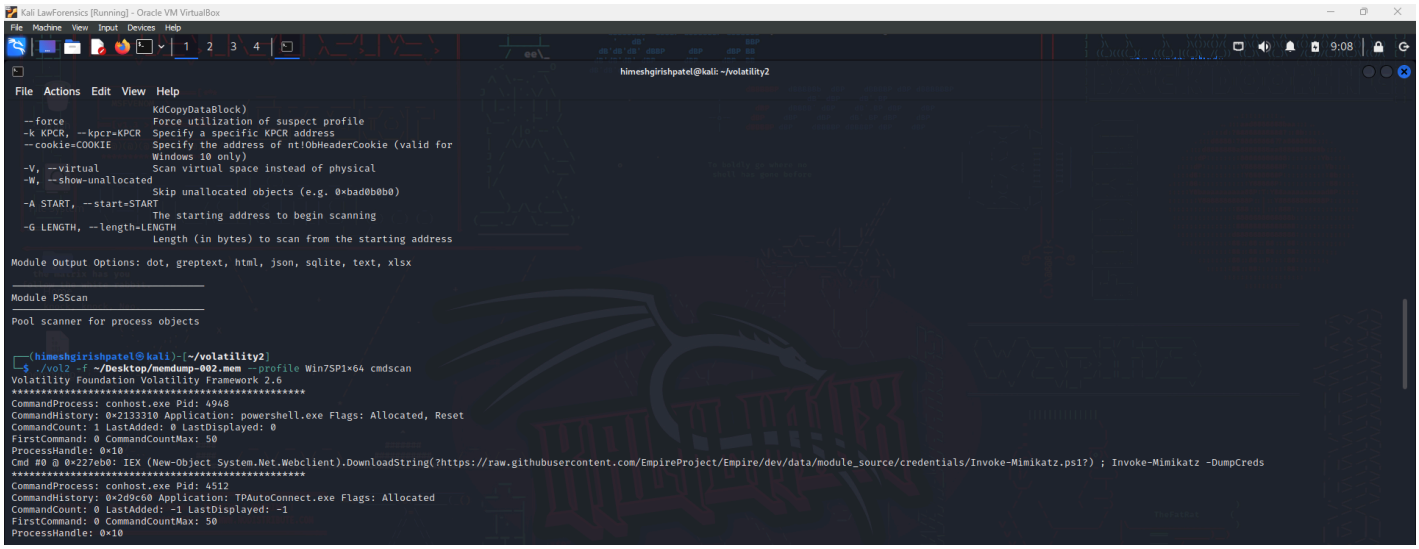
Evidence Image 6:



Link to the VirusTotal analysis of the file under investigation:
https://www.virustotal.com/gui/file/e1bcaea31d54f7f9e3d27eae1a98dfa82f4bb24ccec463034511adaaa5367bb3/details

To find out if the file was behaving the same way in the system under investigation we went back to Volatility and ran the "cmdscan" plugin to see what commands were running. And as suspected the command to install and run Mimikatz was executed.

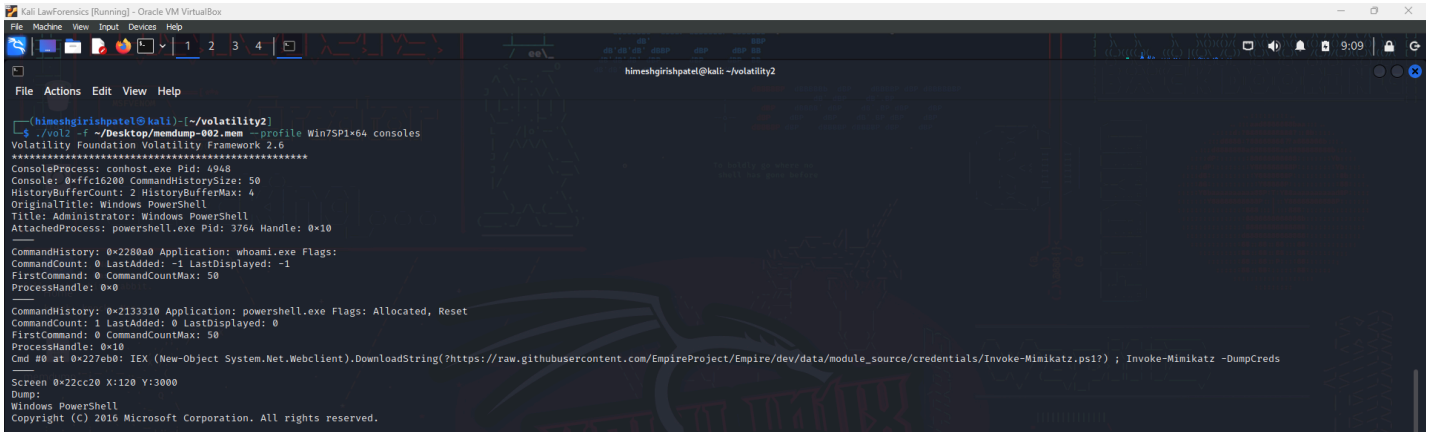
Evidence Image 7:



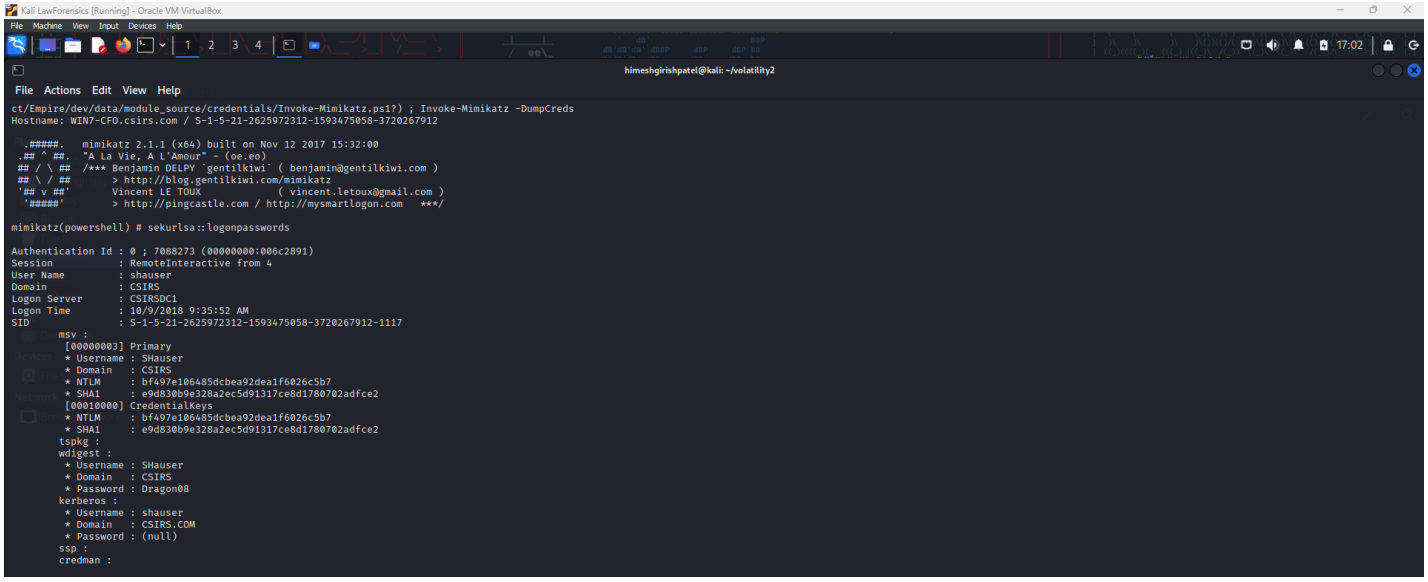
Now to figure out what information might have been leaked using Mimikatz. We ran the "consoles" plugin and found out that the passwords of multiple users were leaked. This uncovered a vulnerability in the system. The

system is using Wdigest for HTTP Digest authentication. This is a security risk as it stores passwords in plaintext. This made it easier for the threat actor to gain the passwords of all the users on the system.

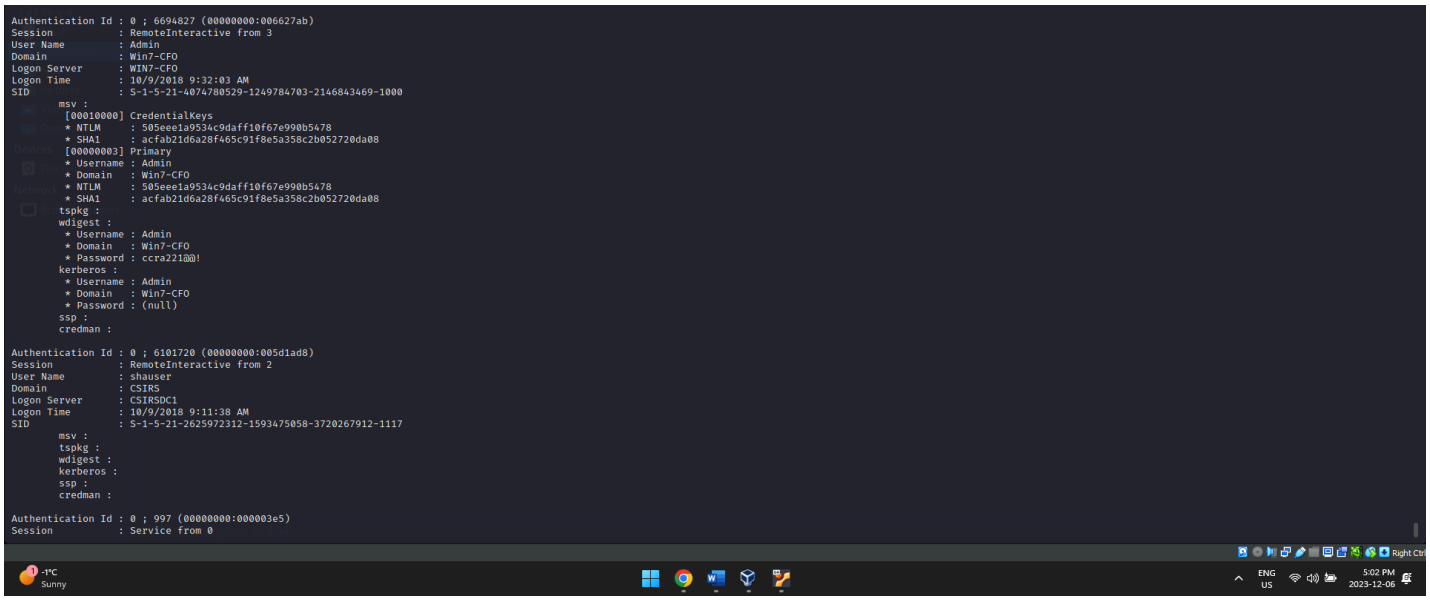
Evidence Image 8:



Evidence Image 9:

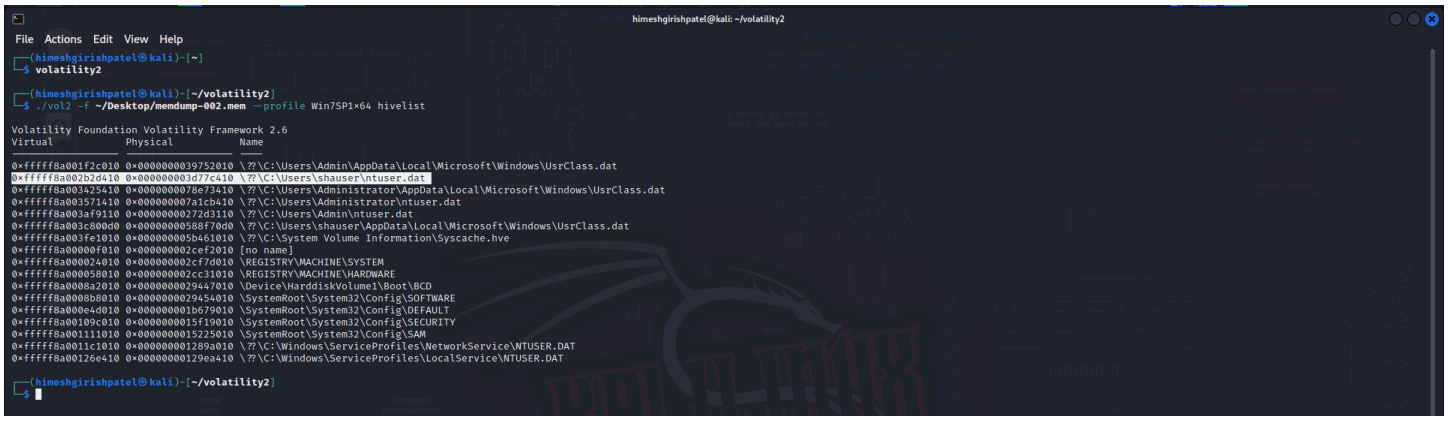


Evidence Image 10:

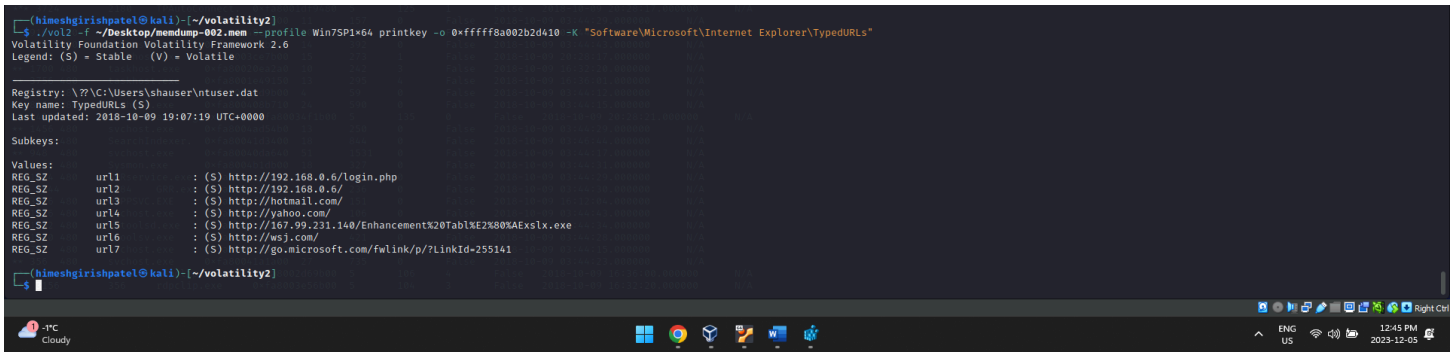


After figuring out the nature of the Suspicious file. We decided to find out where the file came from.

For this we decided to check the online activity of the user “shauser”. For this we used the “hivelist” plugin to find our registry hives available in the memory dump and found the NTUSER.dat file corresponding to shauser.
Evidence Image 11:

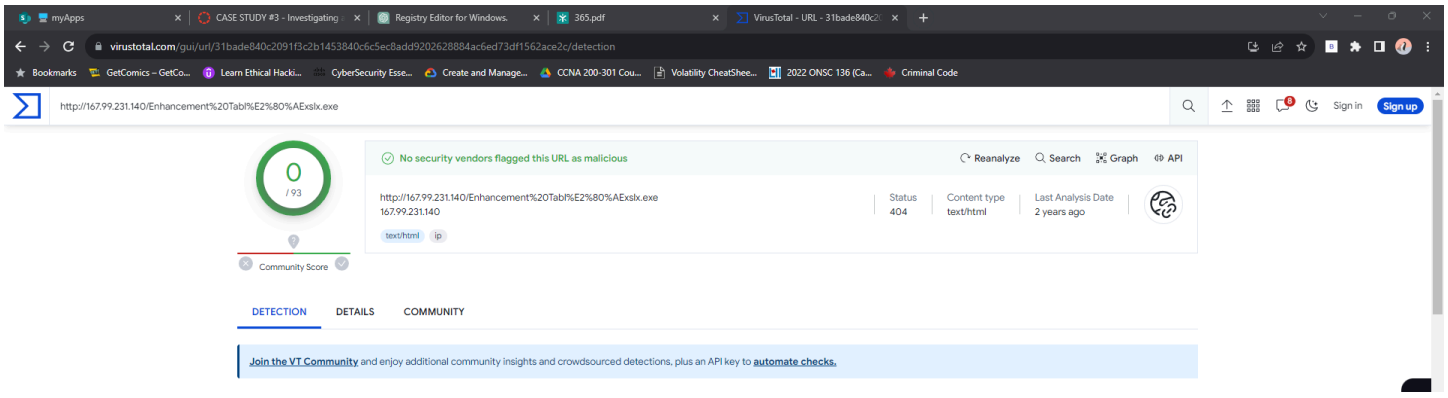


Now to find the online activity of the user. We used the “printkey” and provided the path to find out the URLs accessed by the user.
From this we found that the user accessed the Hotmail site and it might be possible that the suspicious file was downloaded from the link provided in one of the mails the user was accessing. The link in this part is believed to have been from an IP address-based domain with no domain name possibly known.
Evidence Image 12:

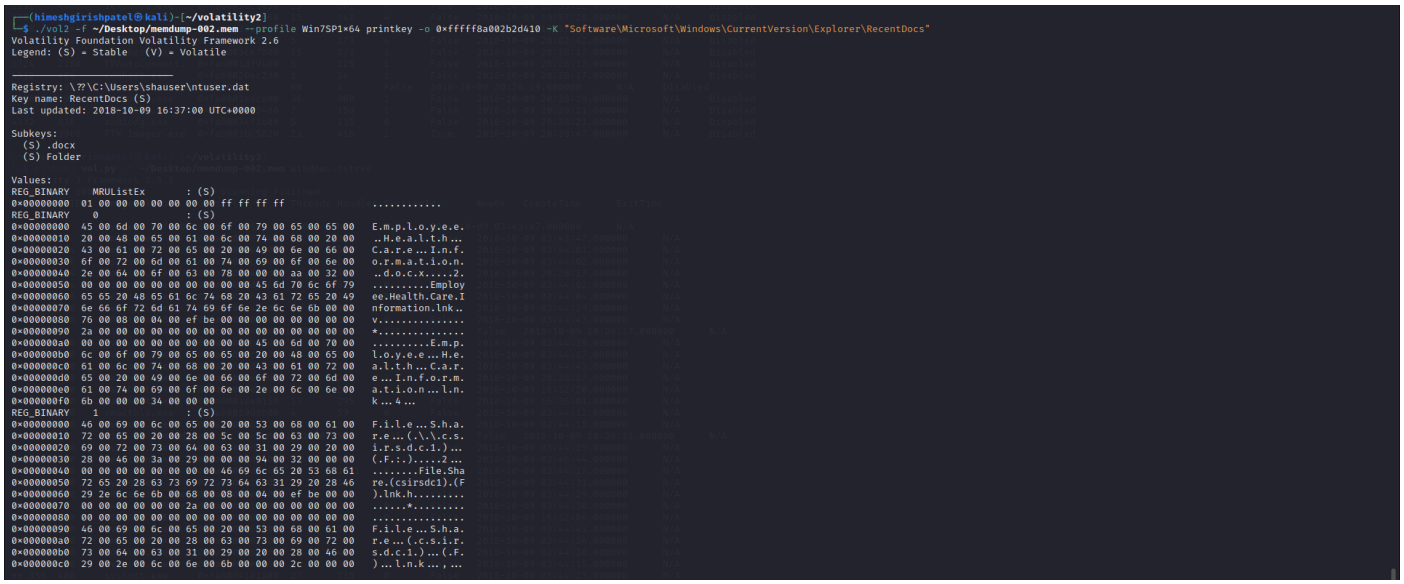


The web address which was accessed to download the file did not come up as a flagged on VirusTotal.
Even after scanning it against 93 security vendors.

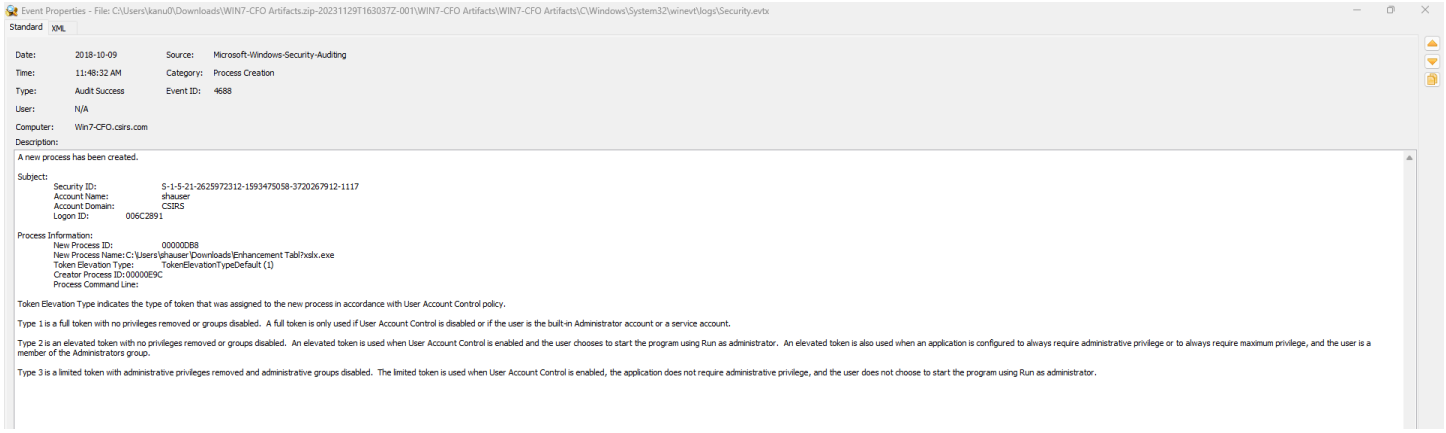
Evidence Image 13:



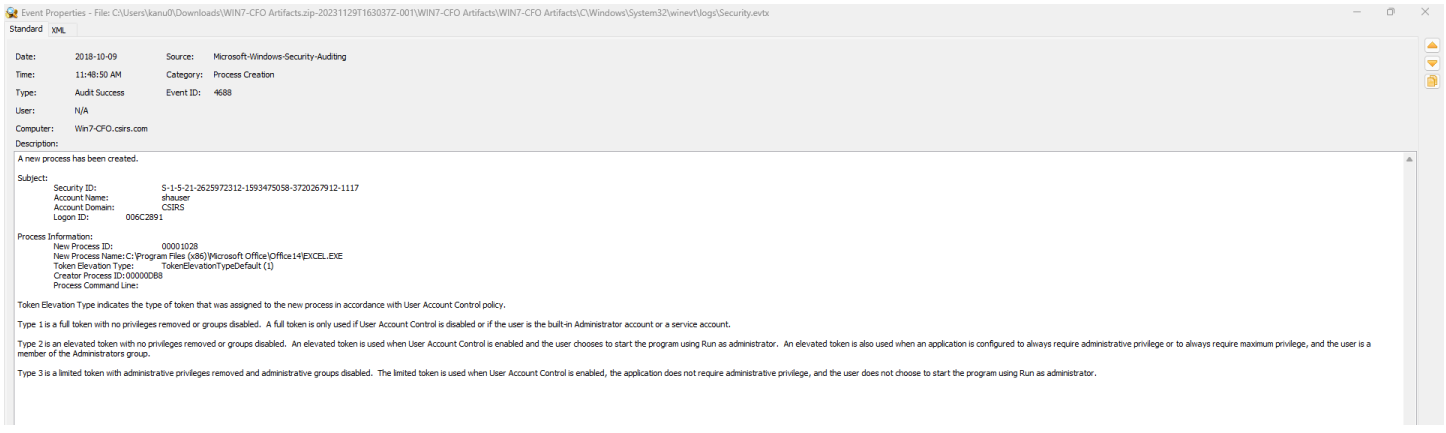
We also check the recent files that might have been accessed by the user. It didn't show much, however; two files were accessed: Employee Health Care Information document and a Shared Folder.
Evidence Image 14:



After this we decided to take a look at the event logs to figure out a more detailed version of what might have happened.
We found that the file was downloaded on 9th October 2018 at 11:48AM
Evidence Image 15:

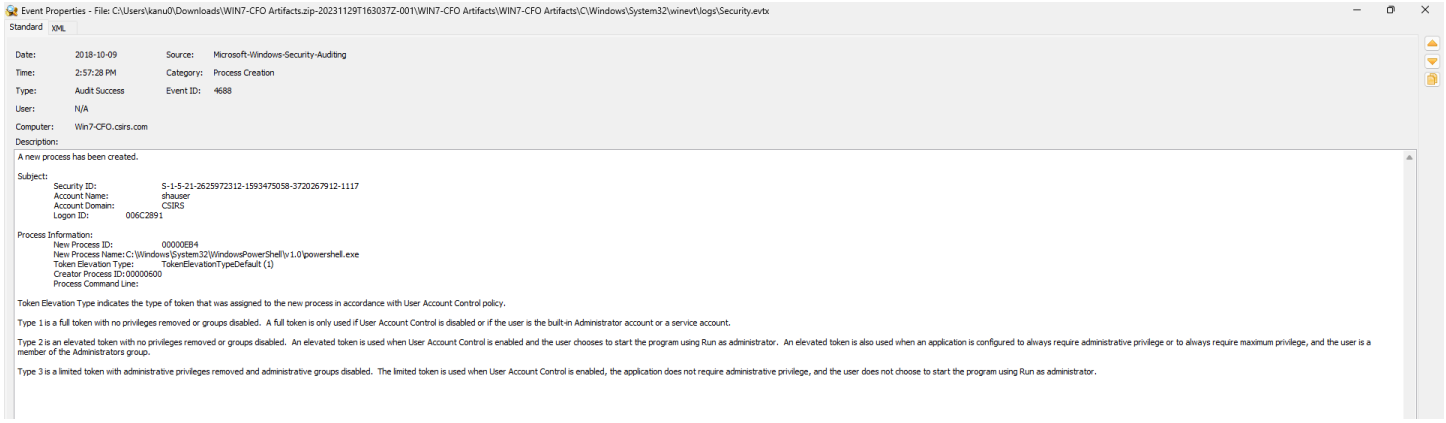


At the same time Excel was also initiated which means that after the installation the file was opened.
Evidence Image 16:

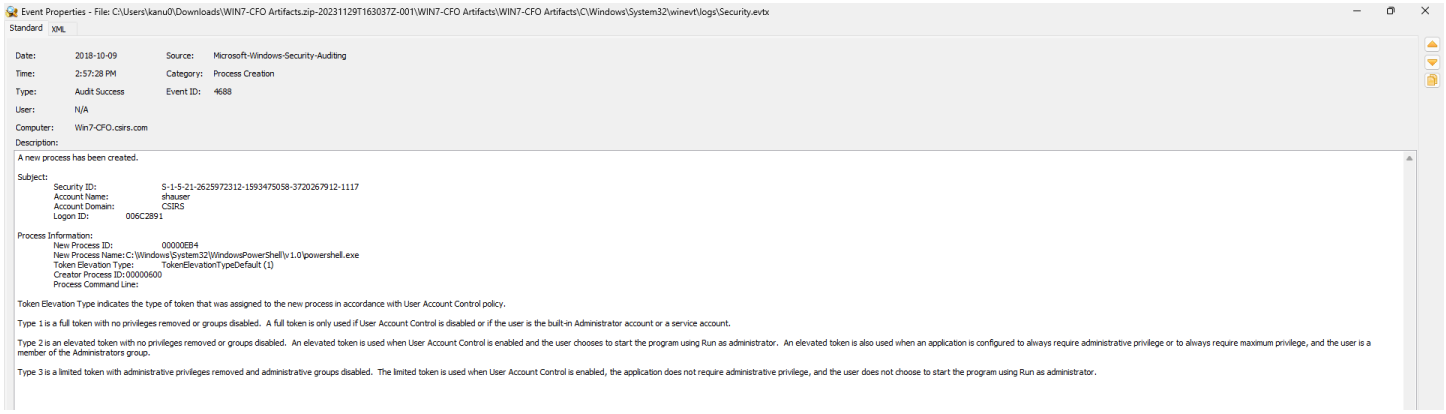


After this the user logged off the system and later around 2:56 PM logged back on remotely.
Evidence Image 17:

From what we discovered using volatility it was clear that powershell was initiated to install Mimikatz.
The image provided gives information about the time when it happened.
Evidence Image 18:

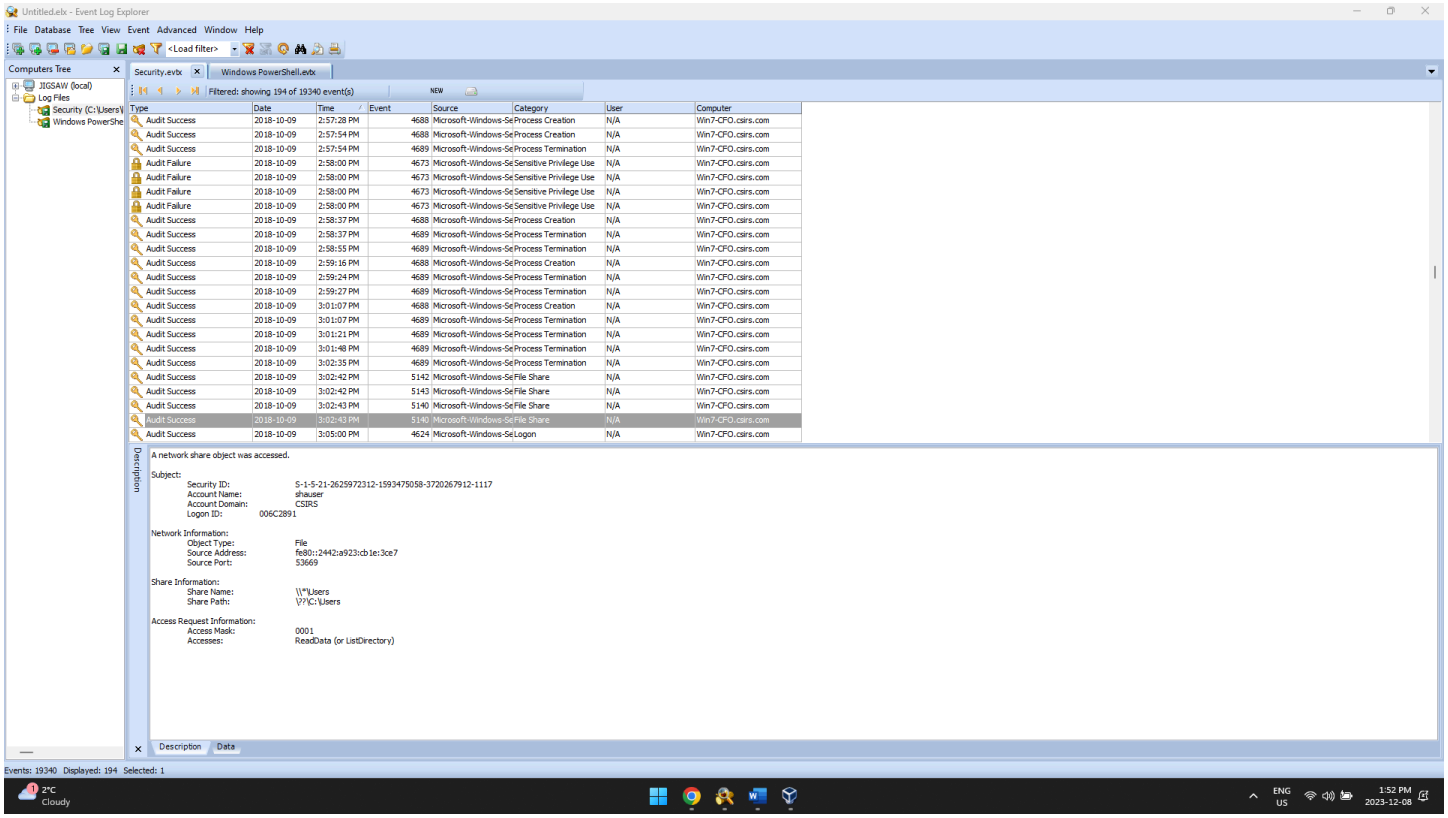


The one detail that was missing in memory was discovered in event logs.
Notepad was accessed at 2:59 PM possible to store the credentials found using Mimikatz.
Evidence Image 19:



After this the user accessed a folder/directory containing some files which leds us to believe had the text file with all the credentials on it.

Evidence Image 20:

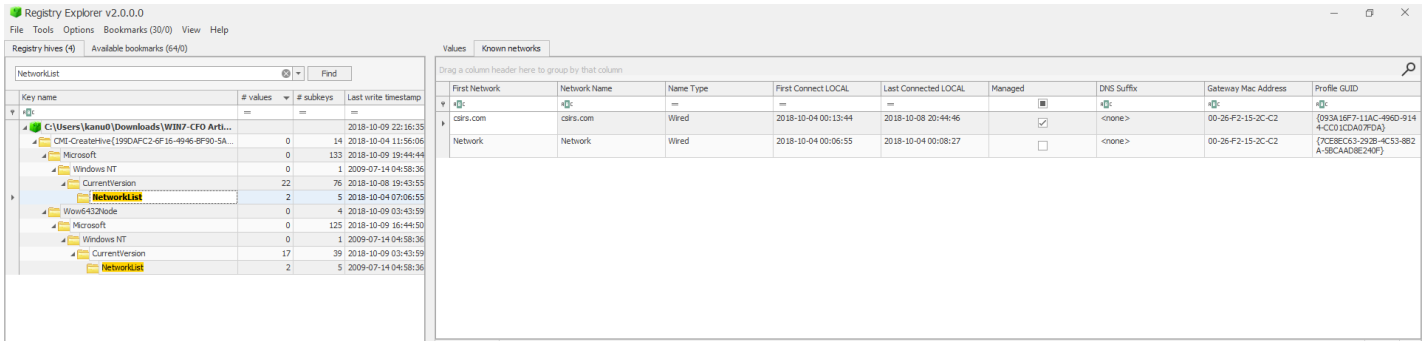


We took a look at the Registry hives to find out more information. However it didn't provide us with anything significant. We did find out about Networks and Users available on the system. This information wasn't available in the memory dump that was captured.

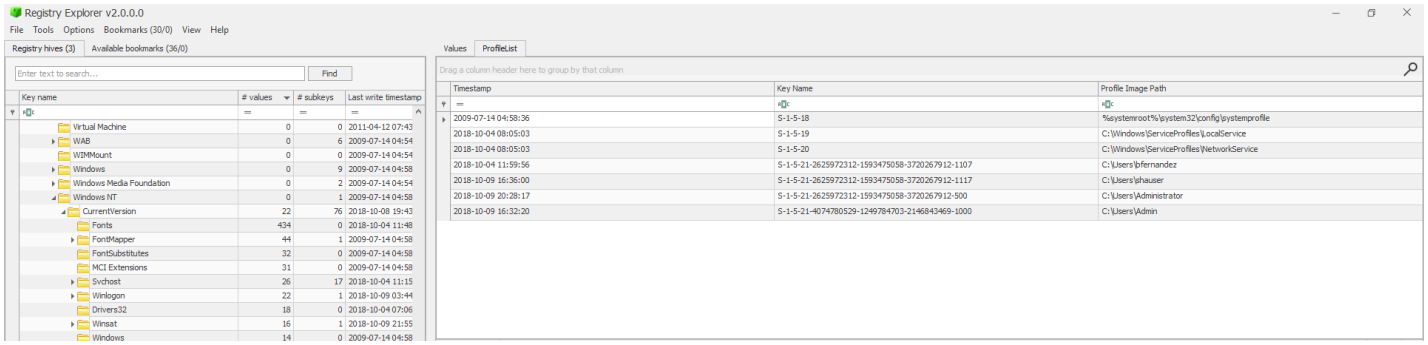
Image 21 has information about the Networks.

Image 22 has information about the Users.

Evidence Image 21:



Evidence Image 22:



The investigation revealed that **shauser** was the target victim on who a possible phishing attack was performed by sending him a malicious email from which the user downloaded a malicious file **Enhancement Tablxslx.exe** which allowed the attacker to steal credentials present on the system by exploiting the lack of security features of **wdigest**.

Incident Timeline:

To figure out the timeline of the incident we used the event logs and analyzed them using Event Log Explorer. And since we already know the user account through which all this occurred. We filtered the Event Logs to help better find out the timeline.

The table below provides a timed description of the activities that lead to the exploit on the system.

Event	**Time/Date**	**Description**
User Logon	6:53 AM, October 4, 2018	Logs on to the system
User Logoff	6:55 AM, October 4, 2018	Logs off the system
User Logon	11:11 AM, October 9, 2018	Logs on to the system
Outlook Open	11:11 AM, October 9, 2018	opens outlook
User Logs on Remotely	11:24 AM, October 9, 2018	User logged on remotely using SSH connection through 192.168.0.4
Outlook Closed	11:35 AM, October 9, 2018	closes outlook
File System	11:36 AM, October 9, 2018	User accessed some files.
Outlook Open	11:36 AM, October 9, 2018	opens outlook
Internet Explorer	11:47 AM, October 9, 2018	User access internet explorer browser.
Downloads Malware File	11:48 AM, October 9, 2018	User downloaded the Enhancement Tablxslx.exe file
Excel	11:48 AM, October 9, 2018	User opened the downloaded file which initialized excel.
PowerShell	2:57 PM, October 9, 2018	PowerShell was opened possible for MimiKatz installation.
whoami	2:57 PM, October 9, 2018	whoami was initiated
PowerShell	2:58 PM, October 9, 2018	Sensitive Privilege Used in PowerShell.
Notepad	2:59 PM, October 9, 2018	Notepad was opened.
User Logoff	4:02 PM, October 9, 2018	Used logged off.

Recommendations:

1. Train Employees to understand security practices and better identify suspicious emails and applications.
2. Adopt using a more secure method of authentication and credential storage and removal of Wdigest from all the devices and network.
3. Update system regularly and stay updated with all the system patches.
4. Use a more robust and commercially approved anti-virus or anti-malware system.
5. Create a strong password policy and ensure changing of passwords frequently.
6. Adopt an email spam and malware identifier system to reduce the chances of a phishing attack.
7. Use of a multi-factor authentication system for login purposes will prevent the attacker from logging into the system even if the password is discovered.