Lab 5

Memory and Mobile Device Forensics

Mina Salehi

University of Maryland Baltimore County

CYBR 642 Introduction to Digital Forensics

Presented to: Gina Scaldaferri

03/05/2025

Introduction

In today's digital landscape, cyber intrusions and malware attacks present significant threats to organizations, necessitating forensic expertise to detect, analyze, and mitigate security breaches. This investigation aims to examine system memory for indicators of compromise (IOCs) that could suggest the presence of malware. Memory forensics involves analyzing a computer's volatile memory to uncover evidence of malicious activity that may not be present on the hard drive (Intezer, 2024). Additionally, mobile device forensics will be conducted to determine the involvement of mobile devices in the intrusion, as these devices can be vectors for unauthorized access and data breaches (SecurityScorecard, 2022). By conducting comprehensive memory and mobile device analyses, investigators aim to uncover critical digital evidence, trace the origin of the breach, and implement security measures to prevent future cyber threats.

Pre-Analysis

Memory forensics is a critical component of digital investigations, focusing on the analysis of volatile memory (RAM) to detect malware, unauthorized processes, and system intrusions. Since RAM stores real-time system activity, it can reveal artifacts such as running processes, network connections, encryption keys, and injected malicious code that may not be present on a hard drive (Ligh et al., 2014). Forensic tools like Volatility and Rekall enable investigators to extract and analyze memory dumps, helping to identify indicators of compromise (IOCs) and reconstruct attack timelines. By examining memory for suspicious activities and hidden threats, memory forensics plays a vital role in incident response, malware detection, and cybersecurity investigations (Nelson et al., 2020).

In this lab, the focus expands beyond basic disk imaging to include memory forensics and mobile device analysis. Upon arrival at the scene, the suspicious system was still powered on, and the intruder was actively connected, making a memory capture crucial for preserving volatile data that may not leave traces on the file system. Memory forensics allows investigators to analyze active processes, network connections, and potential malware execution that could otherwise go undetected (Casey, 2011). Additionally, a cellular phone was discovered at the scene, raising concerns about whether it belonged to the threat actor or was simply lost. This lab involves analyzing a forensic dump of the mobile device and reviewing network traffic captured from the device to determine its relevance to the ongoing investigation.

1. Memory Forensics with Volatility

In this lab, we will utilize Volatility, a powerful open-source memory forensics tool, to examine the xp-laptop-2005-06-25.img memory image. By copying this file from the E:\Memory Forensics folder into a working directory, we will execute various Volatility commands to extract forensic artifacts such as running processes, network connections, open files, and potential malware activity. Each command will be analyzed to understand its purpose, syntax, and forensic significance, helping investigators reconstruct system activity and detect security threats (Ligh et al., 2014).

- volatility -f "E:\Memory Forensics\xp-laptop-2005-06-25.img" imageinfo
    - Volatility.exe –f [path to file] imageinfo

- Volatility.exe: Calls the Volatility framework to execute memory forensics analysis.
- -f [path to file]: Specifies the file path of the memory dump (.img, .bin, .raw, .vmem).
- Imageinfo: Runs the imageinfo plugin, which provides key details about the memory dump.

- When executed, this command extracts and displays crucial metadata about the memory image, such as:
  - Identifies the best-matching Windows profile (e.g., Win7SP1x64, Win10x64_19041).
  - Shows when the memory was captured, useful for timeline reconstruction.
  - PAE (Physical Address Extension) (Volatility Foundation, n.d.)

```
C:\Users\student\Downloads\volatility_2.6_win64_standalone>volatility -f "E:\Memory Forensics\xp-laptop-2005-06-25.img"
imageinfo
Volatility Foundation Volatility Framework 2.6
INFO    : volatility.debug    : Determining profile based on KDBG search...
          Suggested Profile(s) : WinXPSP2x86, WinXPSP3x86 (Instantiated with WinXPSP2x86)
                     AS Layer1 : IA32PagedMemory (Kernel AS)
                     AS Layer2 : FileAddressSpace (E:\Memory Forensics\xp-laptop-2005-06-25.img)
                      PAE type : No PAE
                           DTB : 0x39000L
                          KDBG : 0x8054c060L
          Number of Processors : 1
     Image Type (Service Pack) : 2
                KPCR for CPU 0 : 0xffdff000L
             KUSER_SHARED_DATA : 0xffdf0000L
           Image date and time : 2005-06-25 16:58:47 UTC+0000
     Image local date and time : 2005-06-25 12:58:47 -0400
```

- volatility -f "E:\Memory Forensics\xp-laptop-2005-06-25.img" --profile WinXPSP2x86 psscan
  - Psscan: Runs the process scanner plugin, which searches for process structures that may not be linked to active lists.
  - Unlike pslist, which shows only active processes, psscan detects:

- Hidden processes and malware that can use rootkits
- Shows terminated processes that existed but were closed before the memory capture.
- Identifies processes that lack a parent-child relationship, a potential indicator of malware. (Volatility Foundation, n.d.)

```
C:\Users\student\Downloads\volatility_2.6_win64_standalone>volatility -f "E:\Memory Forensics\xp-l
aptop-2005-06-25.img" --profile WinXPSP2x86 psscan
Volatility Foundation Volatility Framework 2.6
Offset(P)          Name                PID   PPID PDB        Time created                   Time e
xited
------------------ ---------------- ------ ------ ---------- ------------------------------ ------
-----------------------
0x0000000001343790 mqtgsvc.exe        2536    580 0x17406000 2005-06-25 16:48:05 UTC+0000
0x00000000014b13b0 iexplore.exe       2392   1812 0x16f8f000 2005-06-25 16:51:02 UTC+0000
0x0000000001ed76b0 PluckTray.exe      2740    944 0x175fc000 2005-06-25 16:51:10 UTC+0000
0x0000000001ed84e8 dd.exe             4012   2624 0x0eee8000 2005-06-25 16:58:46 UTC+0000
0x0000000001f269e0 PluckUpdater.ex    3076   1812 0x1a6c5000 2005-06-25 16:51:15 UTC+0000   2005-0
6-25 16:51:30 UTC+0000
0x0000000001f48da0 tcpsvcs.exe        1400    580 0x14e54000 2005-06-25 16:47:58 UTC+0000
0x0000000001f5a3b8 csrss.exe           504    448 0x0dac6000 2005-06-25 16:47:30 UTC+0000
0x0000000001f5f020 ssonsvr.exe        1632   1580 0x12b3f000 2005-06-25 16:47:46 UTC+0000
0x0000000001f67500 TaskSwitch.exe     1952   1812 0x139d2000 2005-06-25 16:47:48 UTC+0000
0x0000000001f68518 Crypserv.exe        688    580 0x14a49000 2005-06-25 16:47:55 UTC+0000
0x0000000001f6ca90 Fast.exe           1960   1812 0x13aaf000 2005-06-25 16:47:48 UTC+0000
0x0000000001f6db28 msdtc.exe          1076    580 0x14b6f000 2005-06-25 16:47:55 UTC+0000
0x0000000001f6e7e8 svchost.exe        1024    580 0x1043e000 2005-06-25 16:47:35 UTC+0000
0x0000000001f8dda0 svchost.exe         984    580 0x10220000 2005-06-25 16:47:35 UTC+0000
0x0000000001f8eb10 winlogon.exe        528    448 0x0dcf3000 2005-06-25 16:47:31 UTC+0000
0x0000000001f9a670 spoolsv.exe        1224    580 0x1147b000 2005-06-25 16:47:39 UTC+0000
0x0000000001fa5aa0 svchost.exe         740    580 0x0e575000 2005-06-25 16:47:32 UTC+0000
0x0000000001fa8240 Smc.exe             876    580 0x0eb72000 2005-06-25 16:47:33 UTC+0000
0x0000000001fa8650 svchost.exe         800    580 0x0e8ea000 2005-06-25 16:47:33 UTC+0000
0x0000000001faba78 svchost.exe         840    580 0x0ea71000 2005-06-25 16:47:33 UTC+0000
0x0000000001faf280 jusched.exe         188   1812 0x1413d000 2005-06-25 16:47:49 UTC+0000
0x0000000001fdf020 smss.exe            448      4 0x0c55a000 2005-06-25 16:47:28 UTC+0000
0x0000000002000980 wmiprvse.exe       4080    740 0x10b87000 2005-06-25 16:57:53 UTC+0000
0x0000000002021a78 Rtvscan.exe        1304    580 0x14cc6000 2005-06-25 16:47:58 UTC+0000
0x00000000020238e0 snmp.exe           1424    580 0x14f3a000 2005-06-25 16:47:58 UTC+0000
0x0000000002025608 atiptaxx.exe       2040   1812 0x13d79000 2005-06-25 16:47:49 UTC+0000
0x000000000202bda0 explorer.exe       1812   1764 0x131eb000 2005-06-25 16:47:47 UTC+0000
0x0000000002059da0 DefWatch.exe        864    580 0x14aa7000 2005-06-25 16:47:55 UTC+0000
0x000000000205eda0 wuauclt.exe        2424    840 0x1d3da000 2005-06-25 16:49:21 UTC+0000
0x0000000002076558 ati2evxx.exe        432    580 0x14959000 2005-06-25 16:47:55 UTC+0000
0x0000000002079c18 cmd.exe            2624   1812 0x00868000 2005-06-25 16:57:36 UTC+0000
0x0000000002081da0 svchost.exe        1484    580 0x1515f000 2005-06-25 16:47:59 UTC+0000
0x00000000020dd588 VPTray.exe         1980   1812 0x13a61000 2005-06-25 16:47:49 UTC+0000
0x00000000020e0da0 services.exe        580    528 0x0df16000 2005-06-25 16:47:31 UTC+0000
0x00000000021125d0 EM_EXEC.EXE         224    112 0x14267000 2005-06-25 16:47:50 UTC+0000
0x0000000002113c48 Directcd.exe       1936   1812 0x1386c000 2005-06-25 16:47:48 UTC+0000
0x0000000002199668 lsass.exe           592    528 0x0dfc0000 2005-06-25 16:47:31 UTC+0000
```

- >volatility -f "E:\Memory Forensics\xp-laptop-2005-06-25.img" --profile WinXPSP2x86 pslist

- pslist : Runs the process list plugin, which retrieves and displays all processes that were active and linked to the process list at the time of capture.
  - Provides a snapshot of all active processes in memory at the time of capture.
  - Identifies malware & suspicious processes; e.g., `malware.exe` may indicate an intrusion.
  - Verifies process relationships to detect unauthorized process execution.
  - Establishes forensic timeline data to help in reconstructing attack timelines and user activity. (Volatility Foundation, n.d.)

```
------------------
0x823c87c0 System                    4      0     61     1140 ------      0
0x81fdf020 smss.exe                448      4      3       21 ------      0 2005-06-25 16:47:28 UTC+0000
0x81f5a3b8 csrss.exe               504    448     12      596      0      0 2005-06-25 16:47:30 UTC+0000
0x81f8eb10 winlogon.exe            528    448     21      508      0      0 2005-06-25 16:47:31 UTC+0000
0x820e0da0 services.exe            580    528     18      401      0      0 2005-06-25 16:47:31 UTC+0000
0x82199668 lsass.exe               592    528     21      374      0      0 2005-06-25 16:47:31 UTC+0000
0x81fa5aa0 svchost.exe             740    580     17      198      0      0 2005-06-25 16:47:32 UTC+0000
0x81fa8650 svchost.exe             800    580     10      302      0      0 2005-06-25 16:47:33 UTC+0000
0x81faba78 svchost.exe             840    580     83     1589      0      0 2005-06-25 16:47:33 UTC+0000
0x81fa8240 Smc.exe                 876    580     22      423      0      0 2005-06-25 16:47:33 UTC+0000
0x81f8dda0 svchost.exe             984    580      6       90      0      0 2005-06-25 16:47:35 UTC+0000
0x81f6e7e8 svchost.exe            1024    580     15      207      0      0 2005-06-25 16:47:35 UTC+0000
0x81f9a670 spoolsv.exe            1224    580     12      136      0      0 2005-06-25 16:47:39 UTC+0000
0x81f5f020 ssonsvr.exe            1632   1580      1       24      0      0 2005-06-25 16:47:46 UTC+0000
0x8202bda0 explorer.exe           1812   1764     22      553      0      0 2005-06-25 16:47:47 UTC+0000
0x82113c48 Directcd.exe           1936   1812      4       40      0      0 2005-06-25 16:47:48 UTC+0000
0x81f67500 TaskSwitch.exe         1952   1812      1       21      0      0 2005-06-25 16:47:48 UTC+0000
0x81f6ca90 Fast.exe               1960   1812      1       22      0      0 2005-06-25 16:47:48 UTC+0000
0x820dd588 VPTray.exe             1980   1812      2       89      0      0 2005-06-25 16:47:49 UTC+0000
0x82025608 atiptaxx.exe           2040   1812      1       51      0      0 2005-06-25 16:47:49 UTC+0000
0x81faf280 jusched.exe             188   1812      1       22      0      0 2005-06-25 16:47:49 UTC+0000
0x821125d0 EM_EXEC.EXE             224    112      2       74      0      0 2005-06-25 16:47:50 UTC+0000
0x82076558 ati2evxx.exe            432    580      4       38      0      0 2005-06-25 16:47:55 UTC+0000
0x81f68518 Crypserv.exe            688    580      3       34      0      0 2005-06-25 16:47:55 UTC+0000
0x82059da0 DefWatch.exe            864    580      3       27      0      0 2005-06-25 16:47:55 UTC+0000
0x81f6db28 msdtc.exe              1076    580     14      166      0      0 2005-06-25 16:47:55 UTC+0000
0x82021a78 Rtvscan.exe            1304    580     37      300      0      0 2005-06-25 16:47:58 UTC+0000
0x81f48da0 tcpsvcs.exe            1400    580      2       94      0      0 2005-06-25 16:47:58 UTC+0000
0x820238e0 snmp.exe               1424    580      5      192      0      0 2005-06-25 16:47:58 UTC+0000
0x82081da0 svchost.exe            1484    580      6      119      0      0 2005-06-25 16:47:59 UTC+0000
0x821ca3d0 wdfmgr.exe             1548    580      4       65      0      0 2005-06-25 16:47:59 UTC+0000
0x821ce4d8 Fast.exe               1700    580      2       32      0      0 2005-06-25 16:48:01 UTC+0000
0x821d4da0 mqsvc.exe              1948    580     23      205      0      0 2005-06-25 16:48:02 UTC+0000
0x81343790 mqtgsvc.exe            2536    580      9      119      0      0 2005-06-25 16:48:05 UTC+0000
0xffab8020 alg.exe                2868    580      6      108      0      0 2005-06-25 16:48:11 UTC+0000
0x8205eda0 wuauclt.exe            2424    840      4      160      0      0 2005-06-25 16:49:21 UTC+0000
0xffaa0c10 firefox.exe            2160   1812      6      182      0      0 2005-06-25 16:49:22 UTC+0000
0x82218020 PluckSvr.exe            944    740      9      227      0      0 2005-06-25 16:51:00 UTC+0000
0x814b13b0 iexplore.exe           2392   1812      9      365      0      0 2005-06-25 16:51:02 UTC+0000
0x81ed76b0 PluckTray.exe          2740    944      3      105      0      0 2005-06-25 16:51:10 UTC+0000
0x81f269e0 PluckUpdater.ex        3076   1812      0 --------      0      0 2005-06-25 16:51:15 UTC+0000
16:51:30 UTC+0000
0xffadc9d0 PluckUpdater.ex        1916    944      0 --------      0      0 2005-06-25 16:51:40 UTC+0000
16:53:49 UTC+0000
0x821fb3b8 PluckTray.exe          3256   1812      0 --------      0      0 2005-06-25 16:54:28 UTC+0000
16:54:28 UTC+0000
0x82079c18 cmd.exe                2624   1812      1       29      0      0 2005-06-25 16:57:36 UTC+0000
0x82000980 wmiprvse.exe           4080    740      7        0 ------      0 2005-06-25 16:57:53 UTC+0000
0x822148f0 PluckTray.exe          3100   1812      0 --------      0      0 2005-06-25 16:57:59 UTC+0000
16:57:59 UTC+0000
0x81ed84e8 dd.exe                 4012   2624      1       22      0      0 2005-06-25 16:58:46 UTC+0000
```

- volatility -f "E:\Memory Forensics\xp-laptop-2005-06-25.img" --profile WinXPSP2x86 psxview
    - Psxview: Runs the process cross-view scanner, which compares results from various Volatility plugins to identify inconsistencies.
        - Detects hidden malware and rootkits
        - Cross-validates process listings to ensure no process is being hidden by an attacker.

- If malware injected code into a legitimate process, psxview can highlight anomalies.

```
C:\Users\student\Downloads\volatility_2.6_win64_standalone>volatility -f "E:\Memory Forensics\
xview
Volatility Foundation Volatility Framework 2.6
Offset(P)  Name                    PID pslist psscan thrdproc pspcid csrss session deskthrd Ex
---------- -------------------- ------ ------ ------ -------- ------ ----- ------- -------- --
0x01f67500 TaskSwitch.exe         1952 True   True   True     True   True  True    True
0x01faf280 jusched.exe             188 True   True   True     True   True  True    True
0x021ca3d0 wdfmgr.exe             1548 True   True   True     True   True  True    True
0x02081da0 svchost.exe            1484 True   True   True     True   True  True    True
0x020dd588 VPTray.exe             1980 True   True   True     True   True  True    True
0x17fdb020 alg.exe                2868 True   True   True     True   True  True    True
0x01f8eb10 winlogon.exe            528 True   True   True     True   True  True    True
0x02079c18 cmd.exe                2624 True   True   True     True   True  True    True
0x01f68518 Crypserv.exe            688 True   True   True     True   True  True    True
0x01fa5aa0 svchost.exe             740 True   True   True     True   True  True    True
0x020e0da0 services.exe            580 True   True   True     True   True  True    True
0x014b13b0 iexplore.exe           2392 True   True   True     True   True  True    True
0x01343790 mqtgsvc.exe            2536 True   True   True     True   True  True    True
0x01f48da0 tcpsvcs.exe            1400 True   True   True     True   True  True    True
0x01f6db28 msdtc.exe              1076 True   True   True     True   True  True    False
0x01ed76b0 PluckTray.exe          2740 True   True   True     True   True  True    True
0x02025608 atiptaxx.exe           2040 True   True   True     True   True  True    True
0x0202bda0 explorer.exe           1812 True   True   True     True   True  True    True
0x01f8dda0 svchost.exe             984 True   True   True     True   True  True    False
0x01f6ca90 Fast.exe               1960 True   True   True     True   True  True    True
0x01fa8240 Smc.exe                 876 True   True   True     True   True  True    True
0x01f5f020 ssonsvr.exe            1632 True   True   True     True   True  True    True
0x186fec10 firefox.exe            2160 True   True   True     True   True  True    True
0x02218020 PluckSvr.exe            944 True   True   True     True   True  True    True
0x02113c48 Directcd.exe           1936 True   True   True     True   True  True    True
0x01fa8650 svchost.exe             800 True   True   True     True   True  True    False
0x02021a78 Rtvscan.exe            1304 True   True   True     True   True  True    True
0x021d4da0 mqsvc.exe              1948 True   True   True     True   True  True    True
0x02076558 ati2evxx.exe            432 True   True   True     True   True  True    True
0x01ed84e8 dd.exe                 4012 True   True   True     True   True  True    True
0x020238e0 snmp.exe               1424 True   True   True     True   True  True    True
0x021125d0 FM_EXEC.EXE             224 True   True   True     True   True  True    True
```

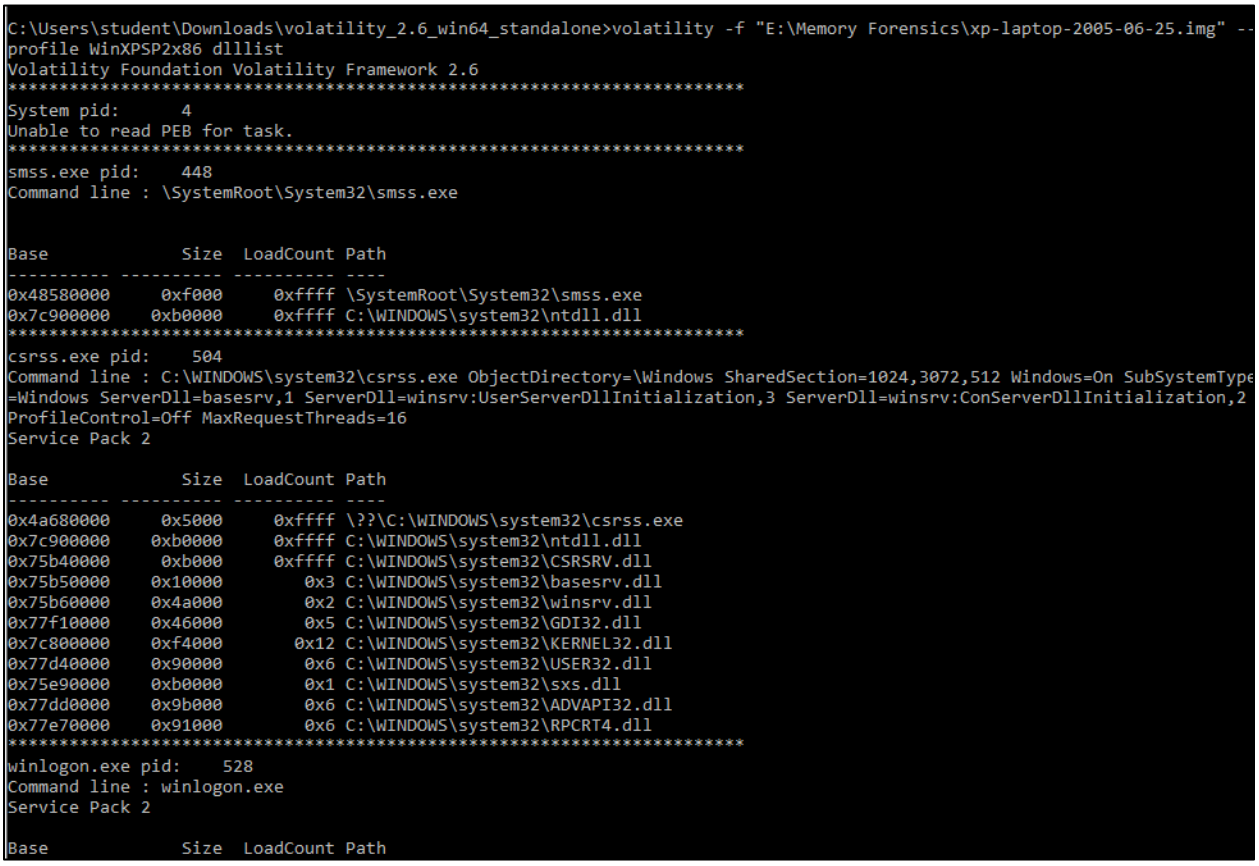
- volatility -f "E:\Memory Forensics\xp-laptop-2005-06-25.img" --profile WinXPSP2x86 connscan
    - Connscan: Uses pool scanning techniques to recover TCP connection structures that may have been terminated or hidden. (Volatility Foundation, n.d.)

```
C:\Users\student\Downloads\volatility_2.6_win64_standalone>volatility -f "E:\Memory Forensics\xp-laptop-2005-06-25.img" --
profile WinXPSP2x86 connscan
Volatility Foundation Volatility Framework 2.6
Offset(P)  Local Address             Remote Address            Pid
---------- ------------------------ ------------------------ ---
0x01370e70 192.168.2.7:1115          207.126.123.29:80         1916
0x01ed1a50 3.0.48.2:17985            66.179.81.245:20084       4287933200
0x01f0e358 192.168.2.7:1164          66.179.81.247:80          944
0x01f11e70 192.168.2.7:1082          205.161.7.134:80          2392
0x01f35cd0 192.168.2.7:1086          199.239.137.200:80        1916
0x01f88e70 192.168.2.7:1162          170.224.8.51:80           1916
0x020869b0 127.0.0.1:1055            127.0.0.1:1056            2160
0x021ca8b8 192.168.2.7:1116          66.161.12.81:80           1916
0x021d2e70 192.168.2.7:1161          66.135.211.87:443         1916
0x02201800 192.168.2.7:1091          209.73.26.183:80          1916
0x02207ab0 192.168.2.7:1151          66.150.96.111:80          1916
0x0220c008 192.168.2.7:1077          64.62.243.144:80          2392
0x0220d6b8 192.168.2.7:1066          199.239.137.200:80        2392
0x02210c48 192.168.2.7:1157          66.151.149.10:80          1916
0x02889800 192.168.2.7:1091          209.73.26.183:80          1916
0x108d2e70 192.168.2.7:1115          207.126.123.29:80         1916
0x187a8008 192.168.2.7:1155          66.35.250.150:80          1916
0x18fffaf0 127.0.0.1:1056            127.0.0.1:1055            2160
0x1d5bde70 192.168.2.7:1115          207.126.123.29:80         1916
0x1f4eb008 192.168.2.7:1155          66.35.250.150:80          1916
```

- volatility.exe -f "E:\Memory Forensics\xp-laptop-2005-06-25.img" --profile WinXPSP2X86 hivelist
    - Hivlist: Lists registry hives found in the memory image, along with their memory offsets.
        - The location of the registry hive in virtual memory.
        - The corresponding physical memory location of the hive.
    - The full file path of the registry hive on disk (Volatility Foundation, n.d.).

```
C:\Users\student\Downloads\volatility_2.6_win64_standalone>volatility -f "E:\Memory Forensics\xp-laptop-2005-06-25.img" --
profile WinXPSP2x86 hivelist
Volatility Foundation Volatility Framework 2.6
Virtual    Physical   Name
---------- ---------- ----
0xe1ecd008 0x11221008 \Device\HarddiskVolume1\Documents and Settings\Sarah\Local Settings\Application Data\Microsoft\Windo
ws\UsrClass.dat
0xe1eff758 0x1294a758 \Device\HarddiskVolume1\Documents and Settings\Sarah\NTUSER.DAT
0xe1bf9008 0x0e6d0008 \Device\HarddiskVolume1\Documents and Settings\LocalService\Local Settings\Application Data\Microsof
t\Windows\UsrClass.dat
0xe1c26850 0x0e882850 \Device\HarddiskVolume1\Documents and Settings\LocalService\NTUSER.DAT
0xe1bf1b60 0x0e213b60 \Device\HarddiskVolume1\Documents and Settings\NetworkService\Local Settings\Application Data\Micros
oft\Windows\UsrClass.dat
0xe1c2a758 0x0e88e758 \Device\HarddiskVolume1\Documents and Settings\NetworkService\NTUSER.DAT
0xe1982008 0x0c61d008 \Device\HarddiskVolume1\WINDOWS\system32\config\software
0xe197f758 0x0c622758 \Device\HarddiskVolume1\WINDOWS\system32\config\default
0xe1986008 0x0c632008 \Device\HarddiskVolume1\WINDOWS\system32\config\SAM
0xe197a758 0x0c60e758 \Device\HarddiskVolume1\WINDOWS\system32\config\SECURITY
0xe1558578 0x02d63578 [no name]
0xe1035b60 0x0283db60 \Device\HarddiskVolume1\WINDOWS\system32\config\system
0xe102e008 0x02837008 [no name]
```

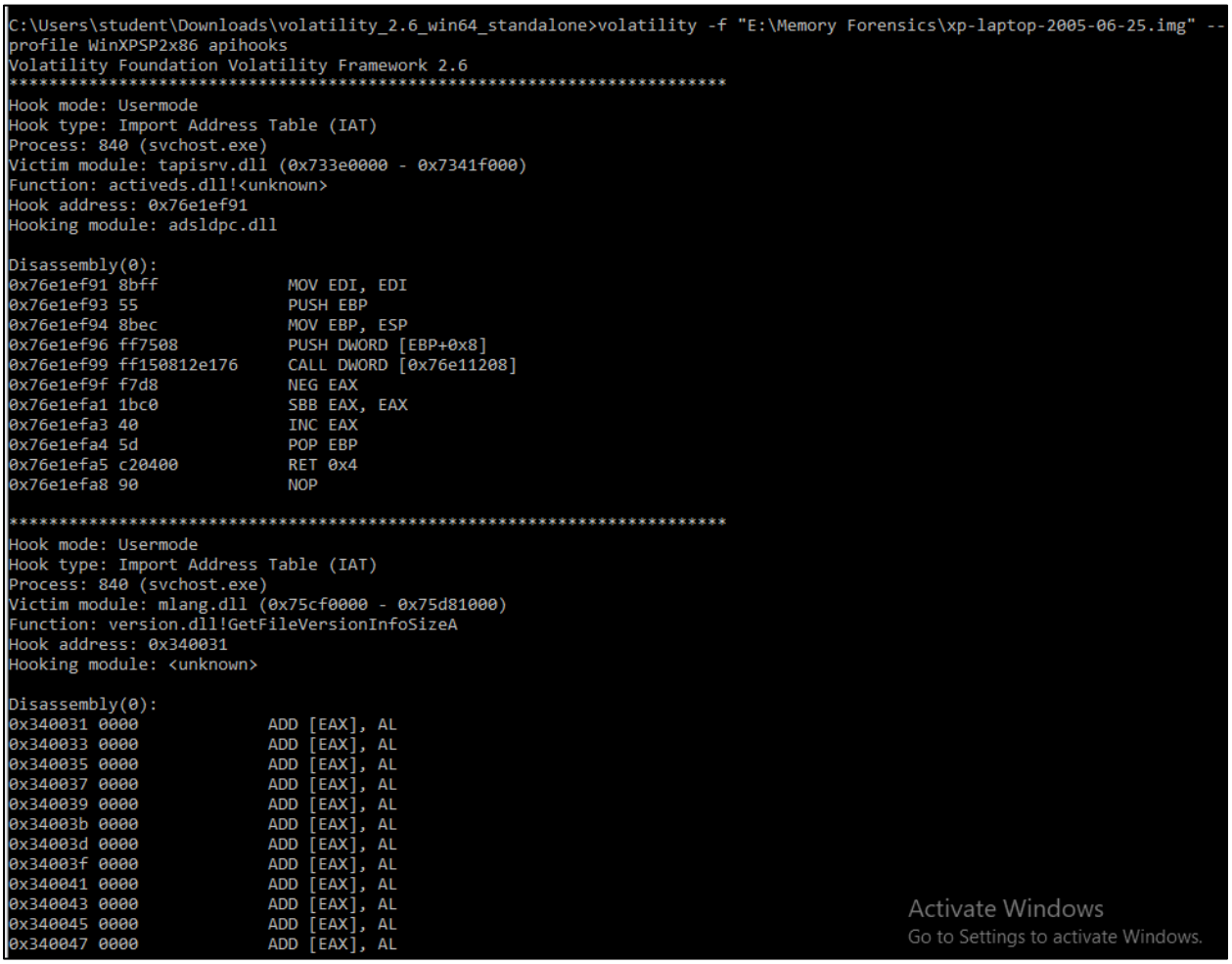
- volatility -f "E:\Memory Forensics\xp-laptop-2005-06-25.img" --profile WinXPSP2x86 dlllist
    - Dlllist: Lists all DLLs loaded by each active process in memory.
        - Identifies malicious DLLs injected into legitimate processes
    - Identifies system compromise (Volatility Foundation, n.d.).



```
C:\Users\student\Downloads\volatility_2.6_win64_standalone>volatility -f "E:\Memory Forensics\xp-laptop-2005-06-25.img" --
profile WinXPSP2x86 dlllist
Volatility Foundation Volatility Framework 2.6
************************************************************************
System pid:      4
Unable to read PEB for task.
************************************************************************
smss.exe pid:    448
Command line : \SystemRoot\System32\smss.exe


Base             Size  LoadCount Path
---------- ---------- ---------- ----
0x48580000     0xf000     0xffff \SystemRoot\System32\smss.exe
0x7c900000    0xb0000     0xffff C:\WINDOWS\system32\ntdll.dll
************************************************************************
csrss.exe pid:    504
Command line : C:\WINDOWS\system32\csrss.exe ObjectDirectory=\Windows SharedSection=1024,3072,512 Windows=On SubSystemType
=Windows ServerDll=basesrv,1 ServerDll=winsrv:UserServerDllInitialization,3 ServerDll=winsrv:ConServerDllInitialization,2
ProfileControl=Off MaxRequestThreads=16
Service Pack 2

Base             Size  LoadCount Path
---------- ---------- ---------- ----
0x4a680000     0x5000     0xffff \??\C:\WINDOWS\system32\csrss.exe
0x7c900000    0xb0000     0xffff C:\WINDOWS\system32\ntdll.dll
0x75b40000     0xb000     0xffff C:\WINDOWS\system32\CSRSRV.dll
0x75b50000    0x10000        0x3 C:\WINDOWS\system32\basesrv.dll
0x75b60000    0x4a000        0x2 C:\WINDOWS\system32\winsrv.dll
0x77f10000    0x46000        0x5 C:\WINDOWS\system32\GDI32.dll
0x7c800000    0xf4000       0x12 C:\WINDOWS\system32\KERNEL32.dll
0x77d40000    0x90000        0x6 C:\WINDOWS\system32\USER32.dll
0x75e90000    0xb0000        0x1 C:\WINDOWS\system32\sxs.dll
0x77dd0000    0x9b000        0x6 C:\WINDOWS\system32\ADVAPI32.dll
0x77e70000    0x91000        0x6 C:\WINDOWS\system32\RPCRT4.dll
************************************************************************
winlogon.exe pid:    528
Command line : winlogon.exe
Service Pack 2

Base             Size  LoadCount Path
```

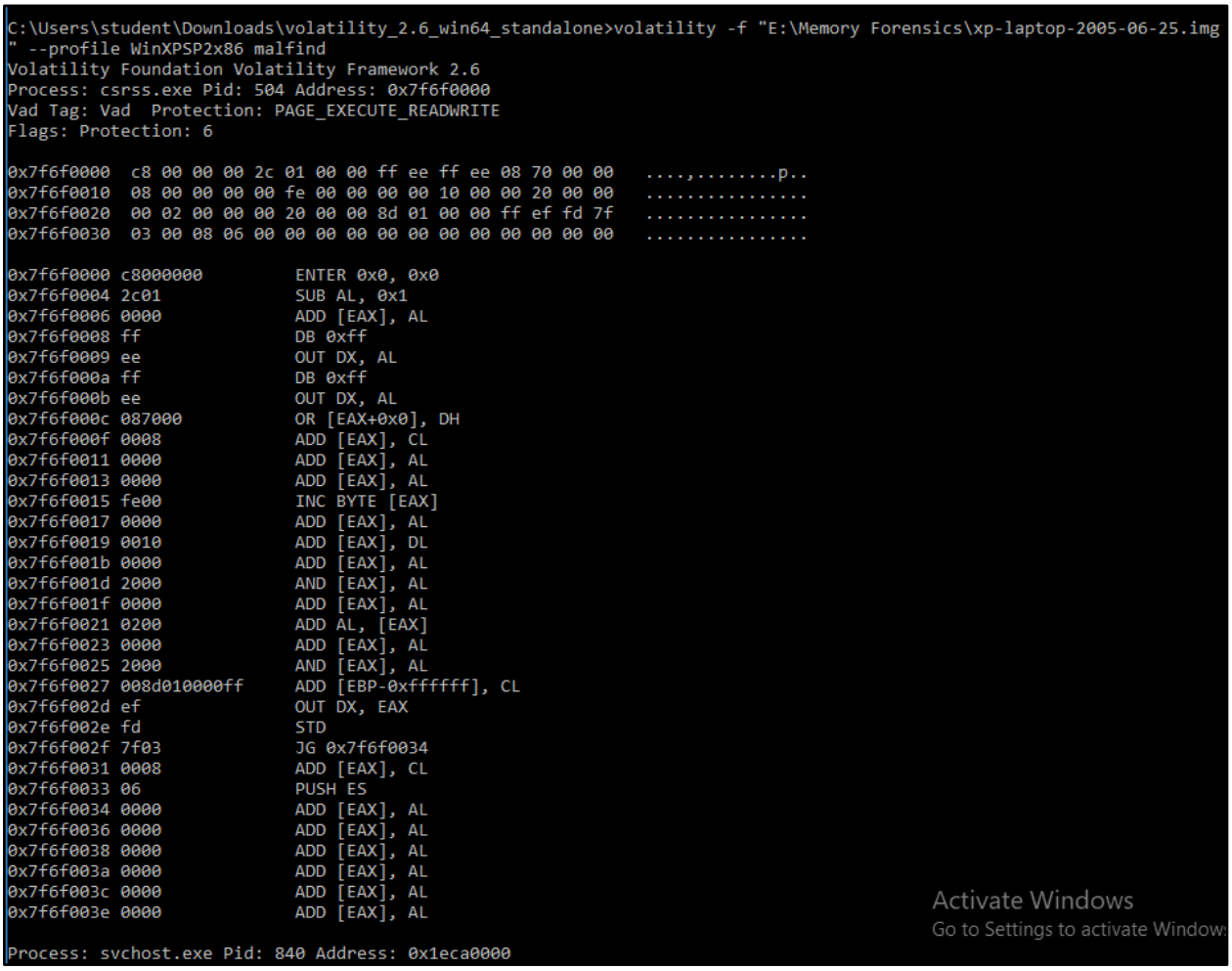
- volatility -f "E:\Memory Forensics\xp-laptop-2005-06-25.img" --profile WinXPSP2x86 apihooks
    - Apihooks: Scans for modifications to Windows API functions in both user-mode and kernel-mode memory.

- Detects malware and rootkits
- Analyzes process injection



- volatility -f "E:\Memory Forensics\xp-laptop-2005-06-25.img" --profile WinXPSP2x86 malfind
    - Malfind: Scans process memory for suspicious code injections and displays malicious memory regions.
        - Malware often injects into explorer.exe or svchost.exe to avoid detection.

- Detects malware techniques like DLL injection and reflective DLL loading (Volatility Foundation, n.d.).



1) Were there any processes running on this computer that were hidden?

- Used psxview
- The psxview command in Volatility is used to detect hidden or stealthy processes in memory by cross-referencing multiple process enumeration techniques. Malware and rootkits often attempt to hide their presence by unlinking from the standard process list, making them invisible to standard tools like Task Manager or pslist.

However, psxview scans multiple process tracking structures to uncover these hidden processes (Volatility Foundation, n.d.).

- If a process appears in psscan but not pslist, it suggests the process is hidden

Process Name	PID	pslist	psscan	Suspicious/hidden
smss.exe	448	False	True	Yes
snmp.exe	1424	False	True	Yes
svchost.exe	984	False	True	Yes
svchost.exe	1024	False	True	Yes
svchost.exe	1484	False	True	Yes
svchost.exe	840	False	True	Yes
Fast.exe	1960	False	True	Yes
iexplore.exe	2392	False	True	Yes
spoolsv.exe	1224	False	True	Yes
dd.exe	4012	False	True	Yes

2) What is the username of the primary user on this computer?

- Sarah
- Used hivelist command
- Hivelist Scans memory and identifies registry hives, displaying their memory offsets and file paths.
- Common Registry Hives and Their Forensic Importance
    - SAM (Security Accounts Manager): Stores user account credentials and login history
    - SECURITY: Contains security policies, user rights, and authentication settings.
    - SOFTWARE: Lists installed applications, registry keys, and system settings.
    - SYSTEM: Stores system startup configurations, drivers, and running services.

- NTUSER.DAT: Tracks user activity, preferences, and browser history.

```
C:\Users\student\Downloads\volatility_2.6_win64_standalone>volatility -f "E:\Memory Forensics\xp-laptop-2005-06-25.img
" --profile WinXPSP2x86 hivelist
Volatility Foundation Volatility Framework 2.6
Virtual    Physical   Name
---------- ---------- ----
0xe1ecd008 0x11221008 \Device\HarddiskVolume1\Documents and Settings\Sarah\Local Settings\Application Data\Microsoft\W
indows\UsrClass.dat
0xe1eff758 0x1294a758 \Device\HarddiskVolume1\Documents and Settings\Sarah\NTUSER.DAT
0xe1bf9008 0x0e6d0008 \Device\HarddiskVolume1\Documents and Settings\LocalService\Local Settings\Application Data\Micr
osoft\Windows\UsrClass.dat
0xe1c26850 0x0e882850 \Device\HarddiskVolume1\Documents and Settings\LocalService\NTUSER.DAT
0xe1bf1b60 0x0e213b60 \Device\HarddiskVolume1\Documents and Settings\NetworkService\Local Settings\Application Data\Mi
crosoft\Windows\UsrClass.dat
0xe1c2a758 0x0e88e758 \Device\HarddiskVolume1\Documents and Settings\NetworkService\NTUSER.DAT
0xe1982008 0x0c61d008 \Device\HarddiskVolume1\WINDOWS\system32\config\software
0xe197f758 0x0c622758 \Device\HarddiskVolume1\WINDOWS\system32\config\default
0xe1986008 0x0c632008 \Device\HarddiskVolume1\WINDOWS\system32\config\SAM
0xe197a758 0x0c60e758 \Device\HarddiskVolume1\WINDOWS\system32\config\SECURITY
0xe1558578 0x02d63578 [no name]
0xe1035b60 0x0283db60 \Device\HarddiskVolume1\WINDOWS\system32\config\system
0xe102e008 0x02837008 [no name]
```

Other account holders:

```
C:\Users\student\Downloads\volatility_2.6_win64_standalone>volatility -f "E:\Memory Forensics\xp-laptop-2005-0
6-25.img" --profile WinXPSP2x86 printkey -K "SAM\Domains\Account\Users\Names"
Volatility Foundation Volatility Framework 2.6
Legend: (S) = Stable   (V) = Volatile

----------------------------
Registry: \Device\HarddiskVolume1\WINDOWS\system32\config\SAM
Key name: Names (S)
Last updated: 2005-06-05 21:06:11 UTC+0000

Subkeys:
  (S) Administrator
  (S) ASPNET
  (S) Guest
  (S) HelpAssistant
  (S) phoenix
  (S) Sarah
  (S) SUPPORT_388945a0
```

3) What is the system time?

- volatility -f "E:\Memory Forensics\xp-laptop-2005-06-25.img" --profile WinXPSP2x86 printkey -K "ControlSet001\Control\TimeZoneInformation"

How to interpret the output:

Registry Key	Description	

ActiveTimeBias	Shows the difference (in minutes) from UTC.	Eastern Standard Time
Bias	Indicates the base time zone offset from UTC.	300
DaylightBias	Displays the Daylight-Saving Time offset.	4294967236
DaylightName	Provides the name of the time zone during daylight saving time	Eastern Daylight Time
StandardBias	Adjusts for standard time offset when daylight saving time is not active.	0
StandardName	Shows the official name of the time zone	Eastern Standard Time

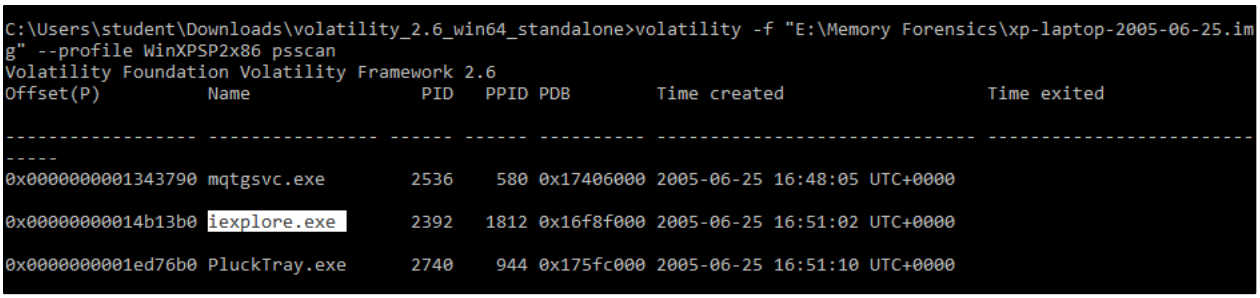
```
C:\Users\student\Downloads\volatility_2.6_win64_standalone>volatility -f "E:\Memory Forensics\xp-laptop-2005-06-25.im
g" --profile WinXPSP2x86 printkey -K "ControlSet001\Control\TimeZoneInformation"
Volatility Foundation Volatility Framework 2.6
Legend: (S) = Stable   (V) = Volatile

----------------------------
Registry: \Device\HarddiskVolume1\WINDOWS\system32\config\system
Key name: TimeZoneInformation (S)
Last updated: 2005-04-03 21:13:24 UTC+0000

Subkeys:

Values:
REG_DWORD     Bias            : (S) 300
REG_SZ        StandardName    : (S) Eastern Standard Time
REG_DWORD     StandardBias    : (S) 0
REG_BINARY    StandardStart   : (S)
0x00000000  00 00 0a 00 05 00 02 00 00 00 00 00 00 00 00 00   ................
REG_SZ        DaylightName    : (S) Eastern Daylight Time
REG_DWORD     DaylightBias    : (S) 4294967236
REG_BINARY    DaylightStart   : (S)
0x00000000  00 00 04 00 01 00 02 00 00 00 00 00 00 00 00 00   ................
REG_DWORD     ActiveTimeBias  : (S) 240
```

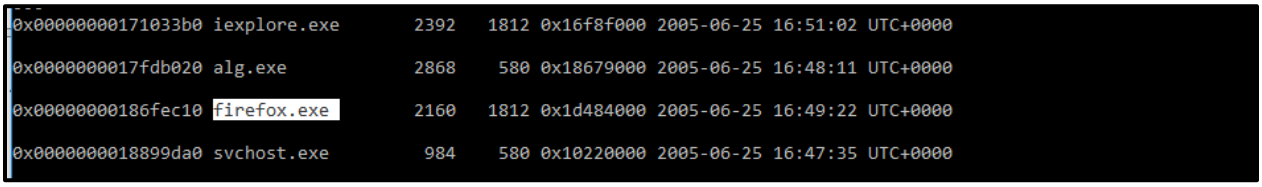
4)What browser(s) were running?

- Explorer and Firefox
- Utilized psscan command
- This Volatility command is used to scan for and recover hidden, unlinked, or previously terminated processes in a memory dump. It is particularly useful in malware analysis, intrusion detection, and forensic investigations where attackers

attempt to hide processes from standard process listings (Volatility Foundation, n.d.).

- chrome.exe, firefox.exe, iexplore.exe indicate browser processes were running.





5) What command was typed/running in a command prompt?

- The cmdscan plugin searches the memory of csrss.exe on XP/2003/Vista/2008 and conhost.exe on Windows 7 for commands that attackers entered through a console shell (cmd.exe). This is one of the most powerful commands you can use to gain visibility into an attacker's actions on a victim system, whether they opened cmd.exe through an RDP session or proxied input/output to a command shell from a networked backdoor. (Volatility Foundation, n.d.).

```
Cmd #0 @ 0x4e2d28: d:
Cmd #1 @ 0x4e1f78: cd  dd
Cmd #2 @ 0x4e2cc8: dir
Cmd #3 @ 0x4e2e00: cd UnicodeRelease
Cmd #4 @ 0x4e2cb8: dir
Cmd #5 @ 0x4e1f90: dd
Cmd #6 @ 0x4e1ff8: dd if=\\.\PhysicalMemory of=c:\xp-laptop-2005-06-25.img conv=noerror
Cmd #7 @ 0x4e2df0: c
Cmd #8 @ 0x4e2e00: cd UnicodeRelease
Cmd #10 @ 0x4e2e40: N?N??
dd.exe
Cmd #11 @ 0x4e2e50: d.exe
Cmd #13 @ 0x4e2ee8: md.exe
**************************************************
CommandProcess: csrss.exe Pid: 504
CommandHistory: 0x11253b0 Application: dd.exe Flags: Allocated, Reset
CommandCount: 1 LastAdded: 0 LastDisplayed: 0
FirstCommand: 0 CommandCountMax: 50
ProcessHandle: 0x2a4
```

```
C:\Users\student\Downloads\volatility_2.6_win64_standalone>volatility -f "E:\Memory Forensics\xp-laptop-2005-06-25.im
g" --profile WinXPSP2x86 cmdscan
Volatility Foundation Volatility Framework 2.6
**************************************************
CommandProcess: csrss.exe Pid: 504
CommandHistory: 0x4e4d88 Application: cmd.exe Flags: Allocated, Reset
CommandCount: 7 LastAdded: 6 LastDisplayed: 6
FirstCommand: 0 CommandCountMax: 50
ProcessHandle: 0x4c8
Cmd #0 @ 0x4e2d28: d:
Cmd #1 @ 0x4e1f78: cd  dd
Cmd #2 @ 0x4e2cc8: dir
Cmd #3 @ 0x4e2e00: cd UnicodeRelease
Cmd #4 @ 0x4e2cb8: dir
Cmd #5 @ 0x4e1f90: dd
Cmd #6 @ 0x4e1ff8: dd if=\\.\PhysicalMemory of=c:\xp-laptop-2005-06-25.img conv=noerror
Cmd #7 @ 0x4e2df0: c
Cmd #8 @ 0x4e2e00: cd UnicodeRelease
Cmd #10 @ 0x4e2e40: N?N??
dd.exe
Cmd #11 @ 0x4e2e50: d.exe
Cmd #13 @ 0x4e2ee8: md.exe
**************************************************
CommandProcess: csrss.exe Pid: 504
CommandHistory: 0x11253b0 Application: dd.exe Flags: Allocated, Reset
CommandCount: 1 LastAdded: 0 LastDisplayed: 0
FirstCommand: 0 CommandCountMax: 50
ProcessHandle: 0x2a4
Cmd #0 @ 0x4e2df0: c
```


6) What processes potentially were running malware?

In memory forensics, identifying malware requires a combination of process analysis, hidden process detection, and injected code identification. Volatility provides powerful plugins such as malfind, pstree, psscan, pslist, and psxview to uncover malicious activity, process injection, and stealth techniques used by attackers.

- The pstree command in Volatility provides a hierarchical view of active and historical processes in a Windows memory dump. It displays the parent-child relationships between processes, allowing forensic analysts to identify malware, suspicious process chains, and anomalies in system behavior (Volatility Foundation, n.d.).
- volatility -f "E:\Memory Forensics\xp-laptop-2005-06-25.img" --profile WinXPSP2x86 malfind
    - csrss.exe Pid: 504 Address: 0x7f6f0000
    - svchost.exe Pid: 840 Address: 0x1eca0000
    - svchost.exe Pid: 840 Address: 0x25860000
    - svchost.exe Pid: 840 Address: 0x45430000
    - svchost.exe Pid: 840 Address: 0x51c70000
    - svchost.exe Pid: 840 Address: 0x63bb0000
    - explorer.exe Pid: 1812 Address: 0x46e0000

Command	Purpose
pslist	Lists active processes in memory.
psscan	Finds terminated or unlinked processes.
malfind	Detects code injection in processes.
psxview	Cross-checks multiple techniques to find hidden processes.
pstree	Displays parent-child process relationship

Forensic Analysis Report

1. Process Analysis

- svchost.exe (PID: 840, Address: 0x63bb0000)
    - This system process spawned wuauclt.exe, which is the Windows Update AutoUpdate Client.
    - No immediate anomalies detected, but further monitoring is recommended.
- explorer.exe (PID: 1812, Address: 0x46e0000)
    - Spawned a suspicious process: dd.exe, which is running under cmd.exe (PID: 4012).
    - dd.exe does not have a Parent Process ID (PPID), indicating possible orphaning or process manipulation.

2. Hidden Processes Detected (psxview Output)

- dd.exe
    - Identified as hidden in psxview, suggesting malware attempting to evade detection.
- iexplore.exe
    - Also detected as hidden with no PPID, which is highly suspicious.
    - Internet Explorer should not run hidden, as legitimate instances are typically visible in process listings.

3. Suspicious File: PluckUpdater.ex

- File Name Irregularity
    - Windows executable files typically use the .exe extension.
    - The missing or altered extension (.ex instead of .exe) suggests an attempt to bypass security mechanisms or evade detection.
- Possible Malware Behavior
    - This could be a fake updater designed to download and execute malicious payloads.
- Process Termination (psscan Output)
    - The file PluckUpdater.ex was found terminated in psscan, indicating it may have executed and then self-terminated to avoid detection.
    - PluckUpdater.ex does not have a Parent Process ID (PPID), indicating possible orphaning or process manipulation.

Hidden processes are a common technique used by malware, rootkits, and advanced threats to evade detection. The combination of pstree, psscan, psxview, and malfind effectively uncovered anomalies that could indicate process injection, unauthorized execution, and malware persistence mechanisms.

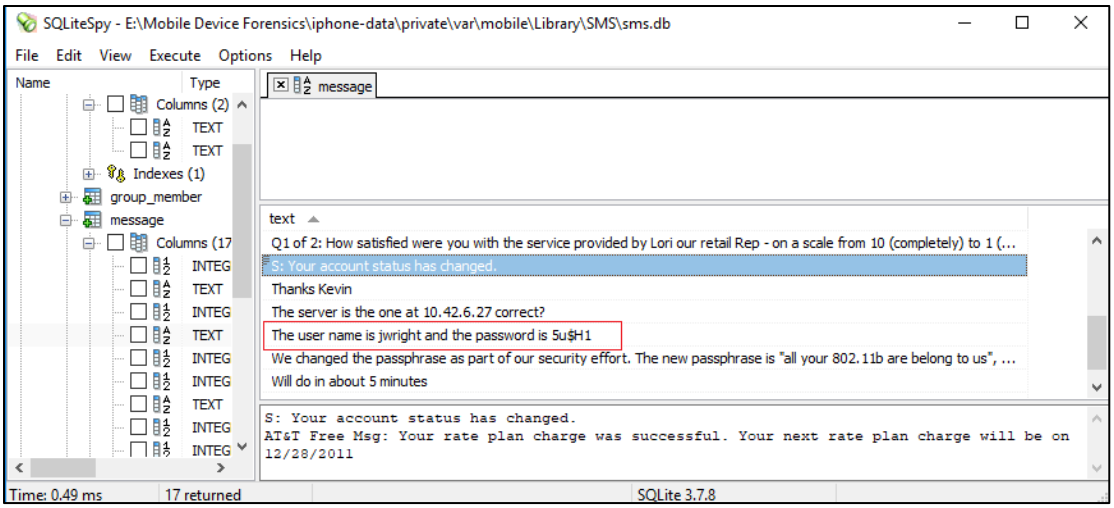
To mitigate potential threats, immediate action should be taken, including isolating the system from the network, performing a full disk and registry analysis, and checking for any persistence mechanisms that could allow malware to reinfect the system upon reboot. Given the severity of the findings, forensic analysts must remain vigilant for further signs of

intrusion, ensure log integrity for timeline reconstruction, and consider advanced malware reverse engineering if necessary. By combining these forensic techniques, analysts can successfully trace the malware's origin, containment, and impact, strengthening cybersecurity defenses against future incidents.

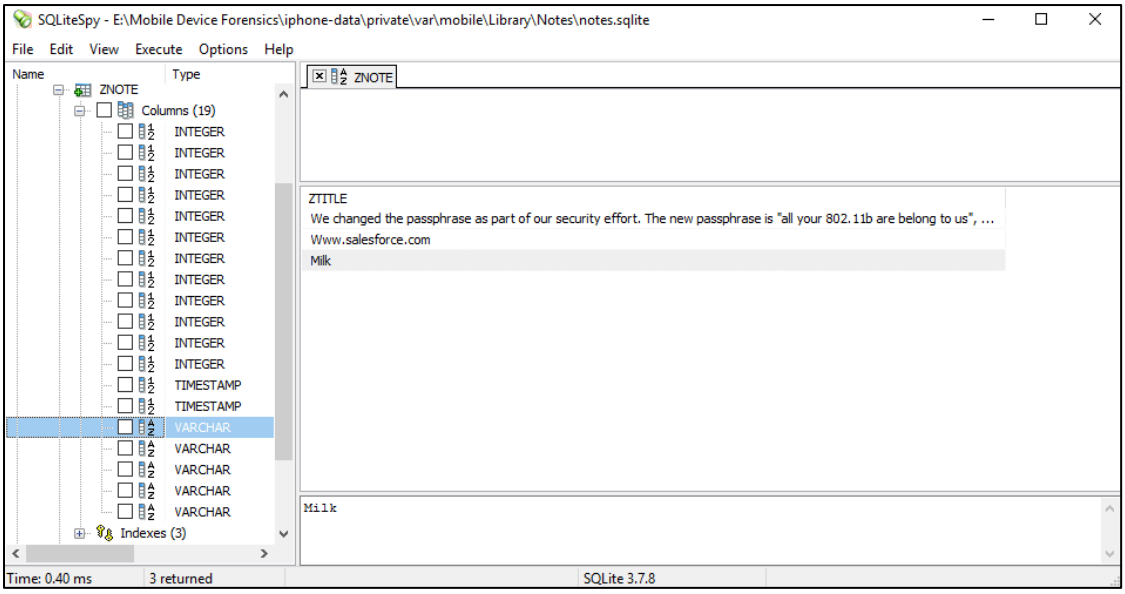
2. Mobile Device Filesystem Forensics

Mobile device filesystem forensics involves analyzing the file structures, databases, and system artifacts stored within a mobile device's storage. In this lab, investigators will navigate through the E:\Mobile Device Forensics\iphone-data folder to locate and examine key forensic artifacts such as database (.db, .sqlite) and plist (.plist) files. These files store critical user data, application records, and system logs, which can provide insights into user activity, communication history, and potential evidence relevant to an investigation. Using SQLiteSpy for database analysis and Plist Editor for .plist file review, forensic analysts will extract and interpret valuable forensic data. Additionally, manual directory exploration is essential for uncovering screenshots, images, and other digital traces that may not be stored in structured databases (Zdziarski, 2008).

1) Access the SMS database and look for login credentials and wireless network credentials that were texted on the device.
    - The username is jwright and the password is 5u$H1

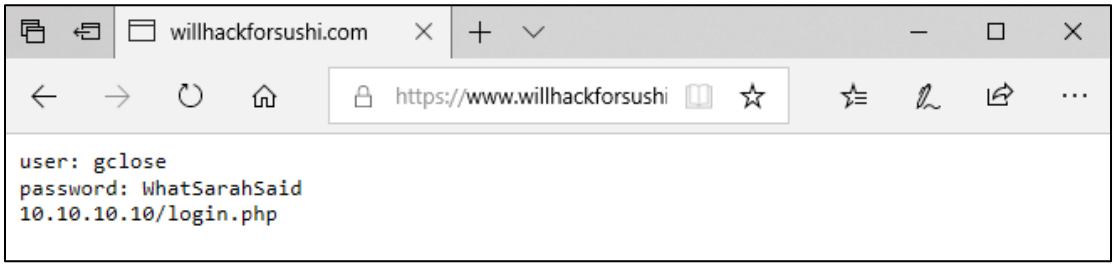


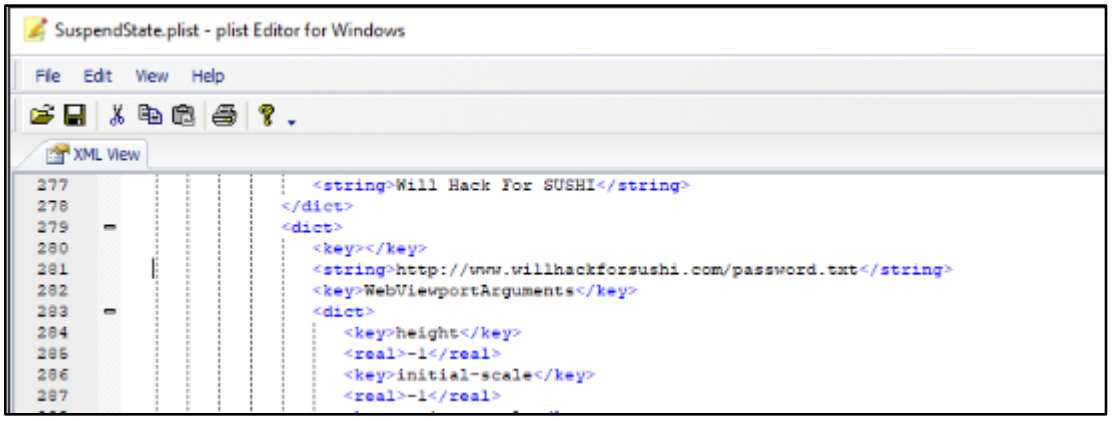
2) Access the notes database to look for information related to salesforce.com credentials.



3) Access the Safari history plist file and review it for a visit to a website that has a password document

- http://www.willhackforsushi.com/password.txt





4) Access the Safari History snapshot to view the image of the last screen seen in the browser.

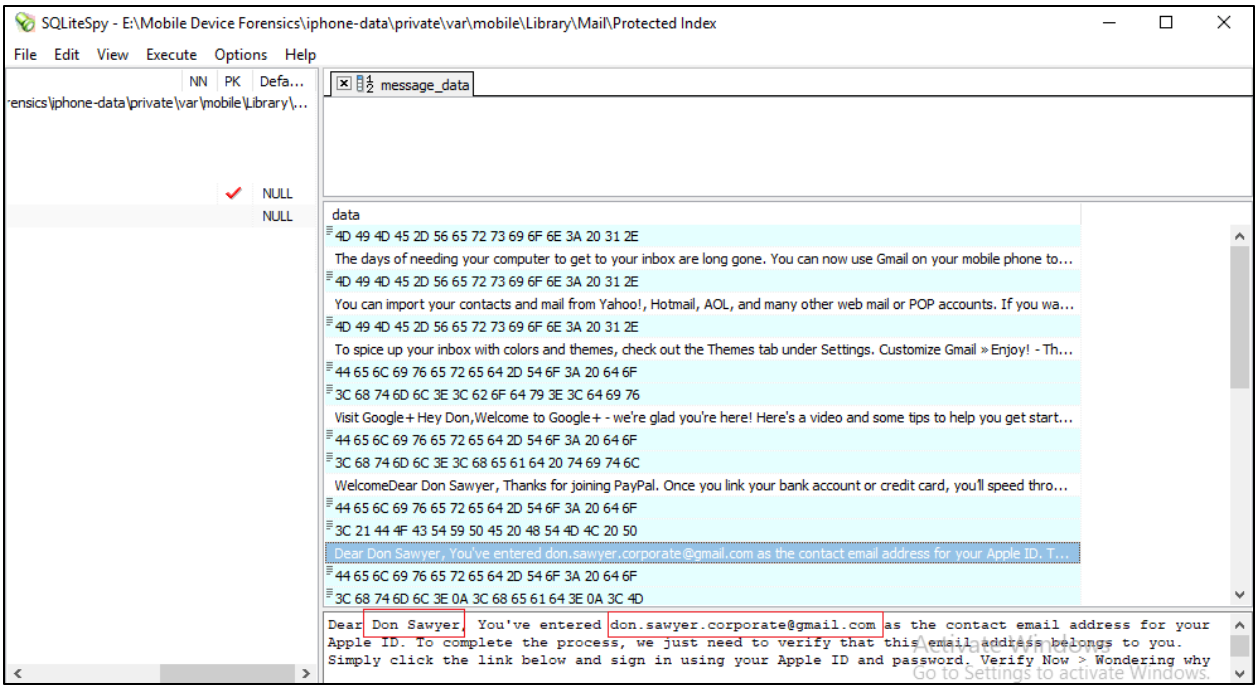
- I opened History.plist in safari folder suing plist Editor for Windows
- <key>lastVisitedDate</key> Timestamp in Plist Files:
  - The value 344923122.9 represents a timestamp stored in a plist file, commonly found in Apple's macOS and iOS systems.
  - Apple uses a CFAbsoluteTime timestamp format
  - To convert I used EpochConverter: https://www.epochconverter.com/
- Last screen seen in the browser:

  http://www.willhackforsushi.com/password.txt

Website	LastVisitDate	GMT
http://www.willhackforsushi.com/password.txt	345433097.1	Friday, December 12, 1980 1:38:17.100 AM
http://www.willhackforsushi.com/	345432894.3	Friday, December 12, 1980 1:34:54.300 AM
http://www.google.com/	344923122.9	Saturday, December 6, 1980 3:58:42.900 AM

5) Access the Email snapshot to determine a possible email and owner of the phone.

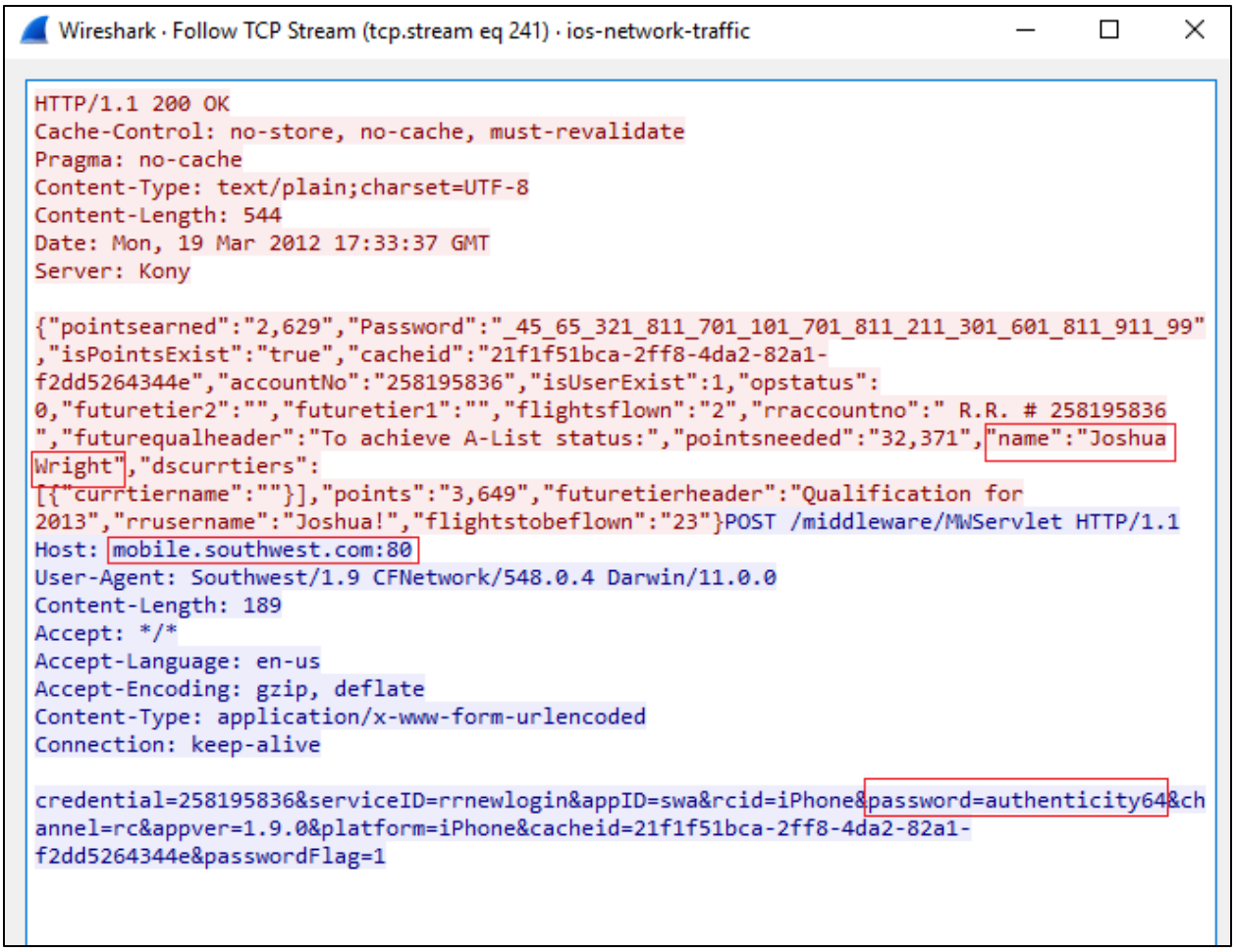
- Owner of the phone: Don Sawyer
- Email: don.sawyer.corporate@gmail.com



3. Mobile Device Network Forensics

In this section, navigate to the Evidence Drive → Mobile Device Forensics and open ios-network-traffic.pcap in Wireshark for analysis.

- In the filter bar, type tcp.stream eq 241
    - In Wireshark, the display filter tcp.stream eq 241 isolates and displays all packets associated with the TCP stream identified by the index number 241. Each TCP connection in a capture file is assigned a unique stream index, and this filter allows analysts to focus on the specific traffic of that connection (Wireshark User's Guide. n.d.).
- App used: mobile.southwest.com
- Password: Authenticity64



Conclusion

In Lab 5, we expanded our forensic analysis to include memory forensics and mobile device examinations, moving beyond traditional disk imaging techniques. Upon arriving at the scene, the system in question was still operational, with the intruder actively connected. This scenario underscored the necessity of capturing volatile memory (RAM) to preserve transient data that might not be recorded on the file system. Memory forensics enables investigators to scrutinize active processes, network connections, and potential malware executions that could otherwise remain undetected. Tools like Volatility facilitate the extraction of such information, offering insights into running processes, open files, and system configurations. intezer.com

Additionally, the discovery of a cellular phone at the scene prompted an analysis to ascertain its relevance to the investigation. Mobile device forensics involves accessing, recovering, and analyzing digital evidence from mobile devices using court-accepted methodologies. This process is vital for determining whether the device belonged to the threat actor or was unrelated to the incident. By examining the device's storage, investigators can uncover critical user data, application records, and system logs that provide insights into user activity and communication history. Utilizing tools like SQLiteSpy for database analysis and Plist Editor for .plist file review, forensic analysts can extract and interpret valuable data. Moreover, manual directory exploration is essential for uncovering screenshots, images, and other digital traces that may not be stored in structured databases

In summary, integrating memory forensics and mobile device analysis into our investigative approach enhances our ability to detect and analyze sophisticated threats. By capturing volatile memory and thoroughly examining mobile devices, we can uncover evidence that traditional disk forensics might lack, thereby strengthening our overall cybersecurity posture.

Glossary

Volatile Memory: Temporary memory (RAM) that stores active system processes and disappears upon shutdown.

Indicators of Compromise (IOCs): Digital traces that signal potential security breaches, such as malware execution or unauthorized access.

Memory Dump: A complete snapshot of a system's RAM, used for forensic analysis.

Pcap (Packet Capture): Pcap refers to a file format used to capture and store network traffic data. It is commonly used in network forensics, intrusion detection, and cybersecurity analysis to examine raw packet data transmitted over a network. Tools such as Wireshark and tcpdump utilize Pcap files to analyze network activity, detect anomalies, and investigate security incidents (Orebaugh et al., 2011).

Registry Hives: Registry hives are major sections of the Windows Registry, which store configuration settings and system information.

API (Application Programming Interface): An Application Programming Interface (API) is a set of protocols, routines, and tools that enable software applications to communicate with each other. APIs facilitate interactions between an operating system and applications, allowing developers to build software with predefined functions and services.

References

Casey, E. (2011). *Digital evidence and computer crime: Forensic science, computers, and the internet* (3rd ed.). Academic Press.

Intezer. (2024). Memory analysis 101: Memory threats and forensic tools. Retrieved from https://intezer.com/blog/incident-response/memory-analysis-forensic-tools/

Ligh, M. H., Case, A., Levy, J., & Walters, A. (2014). The art of memory forensics: Detecting malware and threats in Windows, Linux, and Mac memory. Wiley.

Nelson, B., Phillips, A., & Steuart, C. (2020). Guide to computer forensics and investigations (6th ed.). Cengage Learning.

Orebaugh, A., Ramirez, G., Burke, J., & Morris, L. (2007). *Wireshark & Ethereal network protocol analyzer toolkit*. Syngress.

Robinson, R., & Fishbein, N. (2024, April 23). Memory analysis 101: Understanding memory threats and forensic tools. Intezer. https://intezer.com/blog/incident-response/memory-analysis-forensic-tools/

SecurityScorecard. (2022). Mobile device forensics: Challenges, threats, & solutions. Retrieved from https://securityscorecard.com/blog/mobile-device-forensics/

Volatility Foundation. (n.d.). Command Reference. GitHub. https://github.com/volatilityfoundation/volatility/wiki/command-reference

Wireshark User's Guide. (n.d.). Following Protocol Streams. Retrieved from https://www.wireshark.org/docs/wsug_html_chunked/ChAdvFollowStreamSection.html

Zdziarski, J. (2008). *iPhone Forensics: Recovering Evidence, Personal Data, and Corporate Assets*. O'Reilly Media.