MEMORY AND MOBILE DEVICE FORENSICS

Introduction

Mobile devices and humans have become inseparable in our world today. All ages are glued to these minicomputers as they depend on them to run their day-to-day operations. Their indispensable nature has led to the increase in use to aid almost every task we encounter. The rapid technological advancements have seen more sophisticated features introduced on these devices rather than their primary functionalities of making calls and messages. Now mobile devices have the capability to store and process large volumes of data, create multimedia files and explore any information in the world with the availability of internet (Sneha & Nilima, 2018).

Adversaries are taking advantage of this exponential dependency to satisfy their sinister desires. Crimes such as terror attacks, trafficking, pornographic content et cetera are carried out using these small devices. This complexity has led to the emergence of mobile forensics that aims at investigating the digital evidence acquired from mobile devices in form of text messages, calls, multimedia files, browsing history etc. (Sengul & Erhan, 2017).

Mobile devices continue to become more complex as privacy and security are a major concern which might be difficult to acquire valuable evidence. Different models are adapted to perform mobile device forensic we will be listing some of the general steps followed during this process:

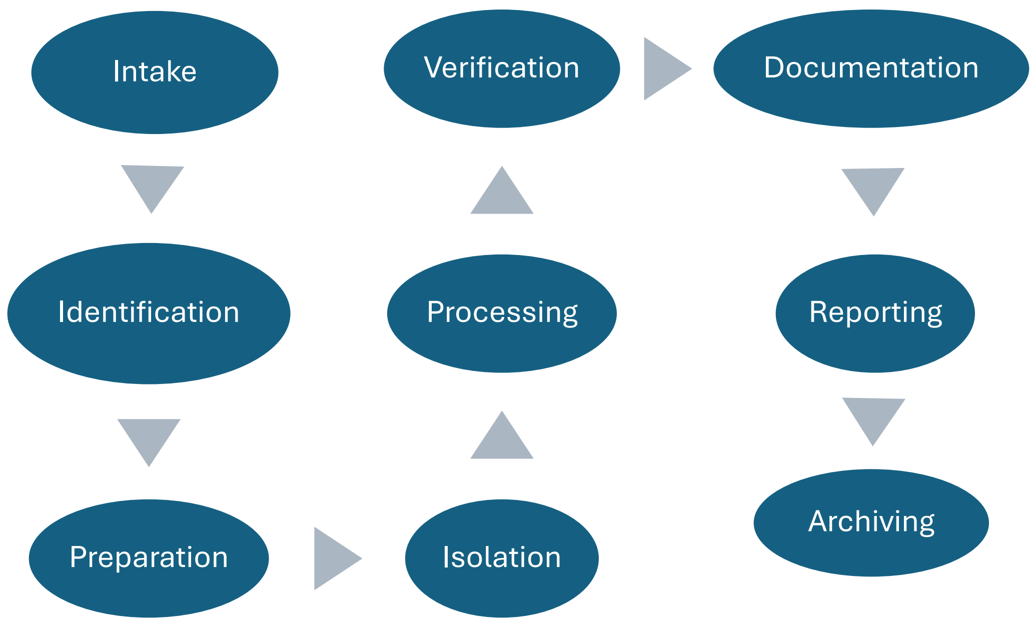


Figure 1/Forensic process (Rohit, Oleg, Heather, & Satish, 2020)

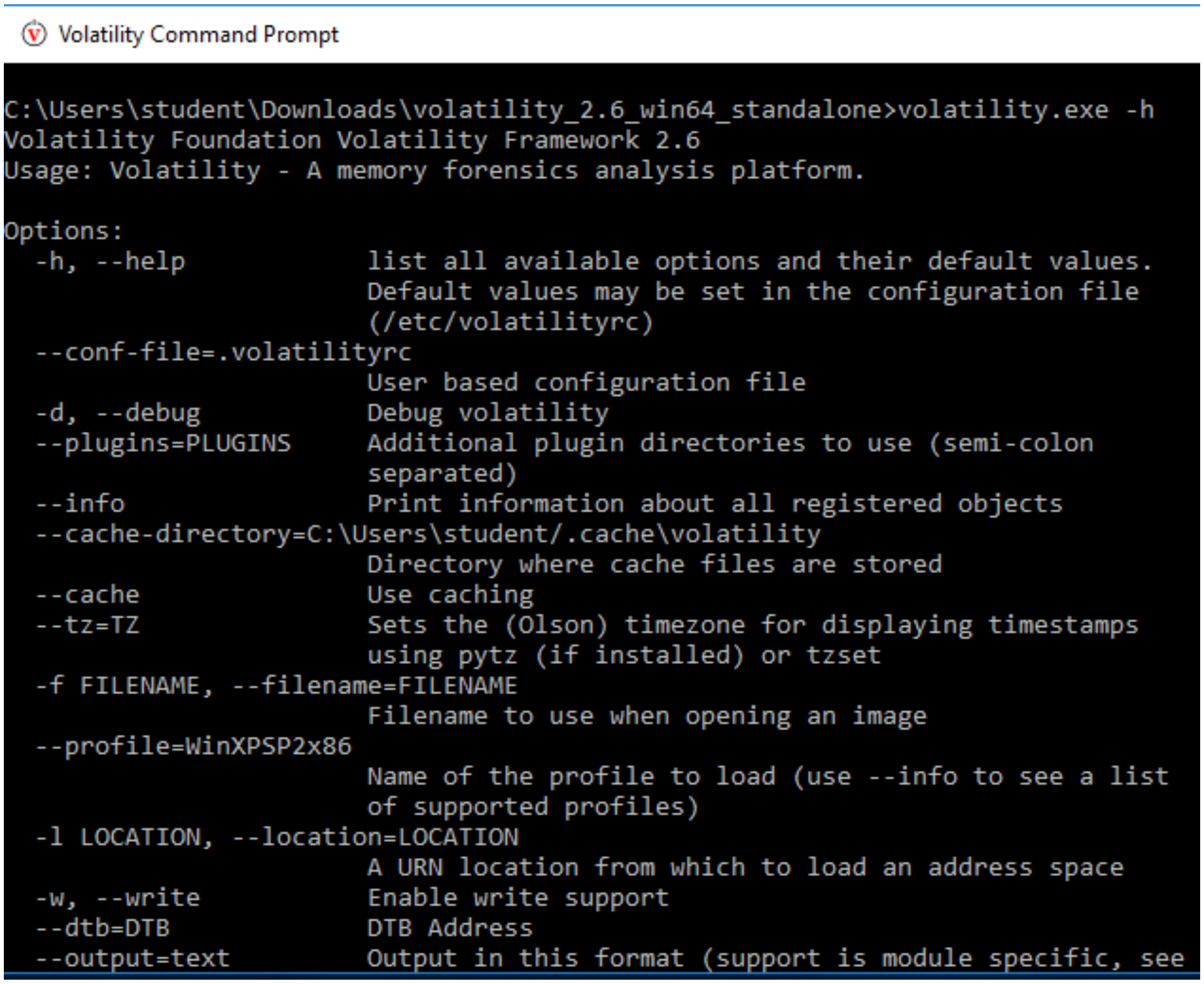
Background

In this lab we will be reviewing memory and mobile device file system as well as network traffic analysis from the mobile device.

Memory Forensics with Volatility

We will be copying the xp-laptop-2005-06-25.img file from the evidence drive into a working directory for the lab as we utilize the volatility tool. We will be detailing different options that we can run with this file in the following screenshots.

volatility.exe -h – This flag is used to prompt a list of options and their usability as shown in the figure below.



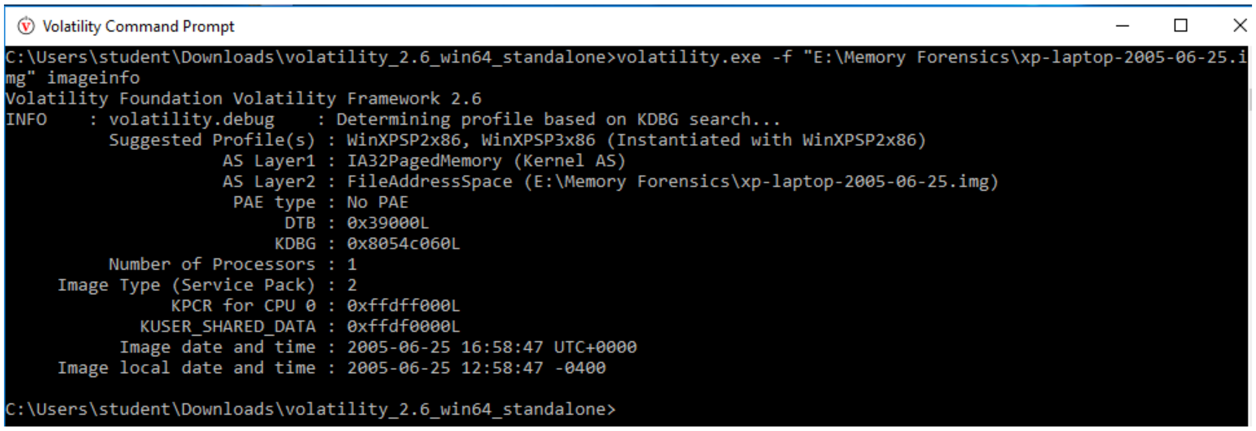
Volatility Command Prompt

```
C:\Users\student\Downloads\volatility_2.6_win64_standalone>volatility.exe -h
Volatility Foundation Volatility Framework 2.6
Usage: Volatility - A memory forensics analysis platform.

Options:
  -h, --help            list all available options and their default values.
                        Default values may be set in the configuration file
                        (/etc/volatilityrc)
  --conf-file=.volatilityrc
                        User based configuration file
  -d, --debug           Debug volatility
  --plugins=PLUGINS     Additional plugin directories to use (semi-colon
                        separated)
  --info                Print information about all registered objects
  --cache-directory=C:\Users\student/.cache\volatility
                        Directory where cache files are stored
  --cache               Use caching
  --tz=TZ               Sets the (Olson) timezone for displaying timestamps
                        using pytz (if installed) or tzset
  -f FILENAME, --filename=FILENAME
                        Filename to use when opening an image
  --profile=WinXPSP2x86
                        Name of the profile to load (use --info to see a list
                        of supported profiles)
  -l LOCATION, --location=LOCATION
                        A URN location from which to load an address space
  -w, --write           Enable write support
  --dtb=DTB             DTB Address
  --output=text         Output in this format (support is module specific, see
```

Figure 2/Help Command

volatility.exe -f c:\users\student\Desktop\Work\Labs\Evidence\lab5\xp-laptop-2005-06-25.img imageinfo

The imageinfo option is used to identify the information of the image file that we have loaded. This returns a high-level summary of the image loaded such as the suggested profile, service park, the date the image was acquired etcetera as shown in the figure below.

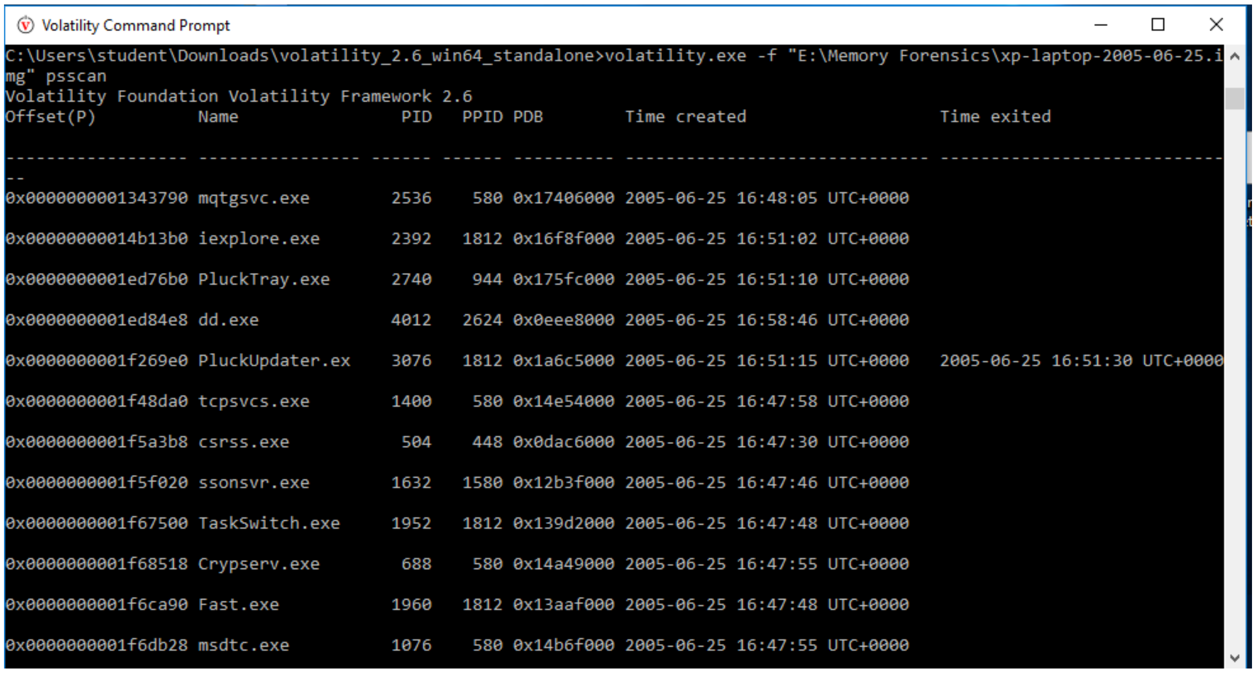


```
C:\Users\student\Downloads\volatility_2.6_win64_standalone>volatility.exe -f "E:\Memory Forensics\xp-laptop-2005-06-25.i
mg" imageinfo
Volatility Foundation Volatility Framework 2.6
INFO    : volatility.debug    : Determining profile based on KDBG search...
          Suggested Profile(s) : WinXPSP2x86, 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

C:\Users\student\Downloads\volatility_2.6_win64_standalone>
```

Figure 3/Imageinfo

volatility.exe -f c:\users\student\Desktop\Work\Labs\Evidence\lab5\xp-laptop-2005-06-25.img --profile WinXPSP2x86 psscan

psscan option is used to pool scanner for process object as described by the help option discussed above. It can find processes that are inactive and hidden.

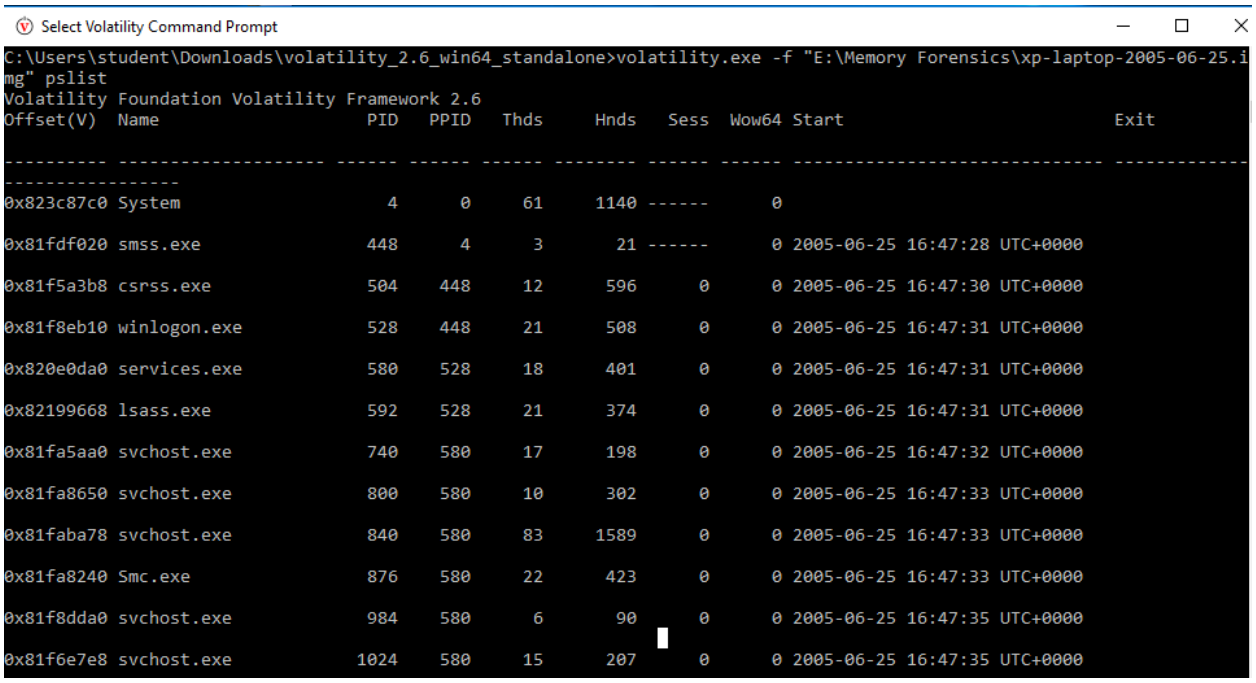


```
C:\Users\student\Downloads\volatility_2.6_win64_standalone>volatility.exe -f "E:\Memory Forensics\xp-laptop-2005-06-25.i
mg" psscan
Volatility Foundation Volatility Framework 2.6
Offset(P)          Name                PID   PPID PDB        Time created                   Time exited

------------------ ---------------- ------ ------ ---------- ------------------------------ ---------------------------
--
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   2005-06-25 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
```

Figure 4/psscan

volatility.exe -f c:\users\student\Desktop\Work\Labs\Evidence\lab5\xp-laptop-2005-06-25.img --profile WinXPSP2x86 pslist

This option prints all the running processes by following the EPROCESS lists. It shows the offset, process name, process ID, parent process ID, number of threads, number of handlers, sessions and the start and exit date.

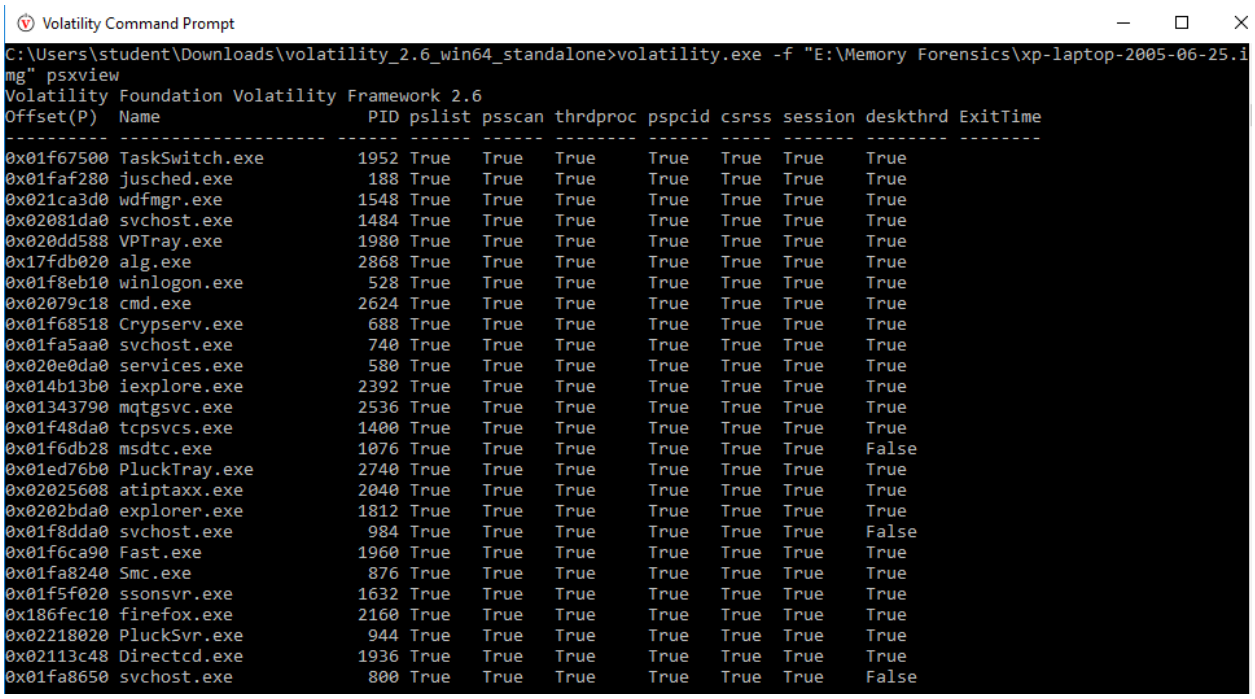


```
C:\Users\student\Downloads\volatility_2.6_win64_standalone>volatility.exe -f "E:\Memory Forensics\xp-laptop-2005-06-25.i
mg" pslist
Volatility Foundation Volatility Framework 2.6
Offset(V)  Name                    PID   PPID   Thds     Hnds   Sess  Wow64 Start                          Exit

---------- -------------------- ------ ------ ------ -------- ------ ------ ------------------------------ ------------
-----------------
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
```

Figure 5/pslist

volatility.exe -f c:\users\student\Desktop\Work\Labs\Evidence\lab5\xp-laptop-2005-06-25.img --profile WinXPSP2x86 psxview

This option finds hidden processes with various process listings. It compares the active processes indicated within psActiveProcessHead with any other possible sources within the memory image.

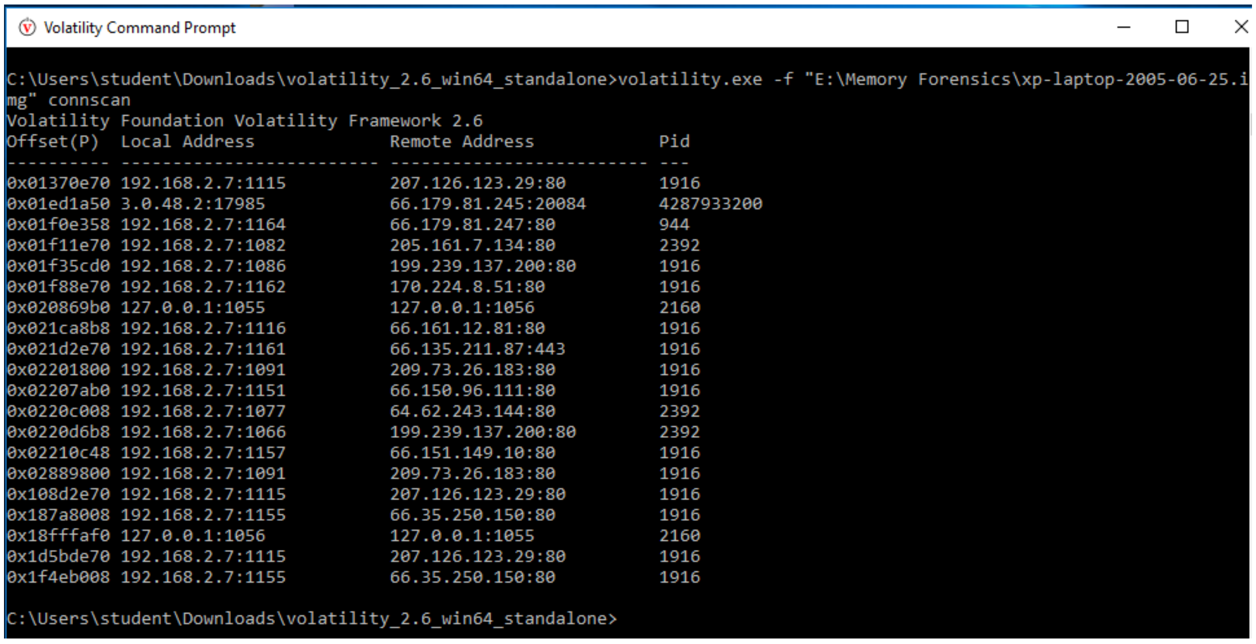


```
C:\Users\student\Downloads\volatility_2.6_win64_standalone>volatility.exe -f "E:\Memory Forensics\xp-laptop-2005-06-25.i
mg" psxview
Volatility Foundation Volatility Framework 2.6
Offset(P)  Name                    PID pslist psscan thrdproc pspcid csrss session deskthrd ExitTime
---------- -------------------- ------ ------ ------ -------- ------ ----- ------- -------- --------
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
```

Figure 6/psxview

volatility.exe -f c:\users\student\Desktop\Work\Labs\Evidence\lab5\xp-laptop-2005-06-25.img --profile WinXPSP2x86 connscan

From the help option connscan is used to Pool scanner for TCP (Transmission control Protocol) connections. This command can find both active and inactive connections.

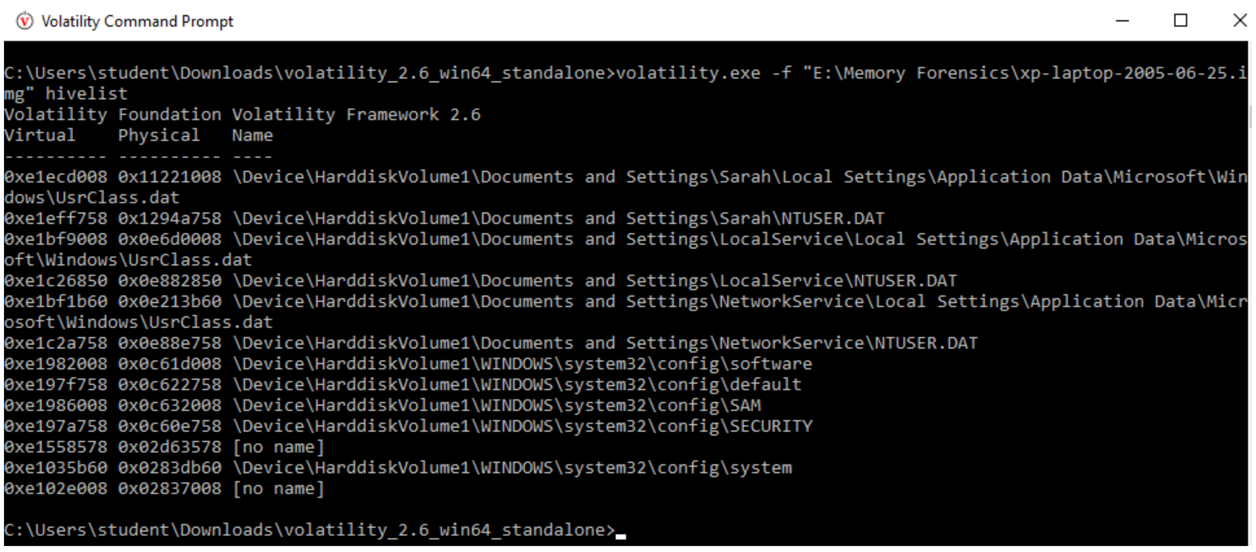


```
C:\Users\student\Downloads\volatility_2.6_win64_standalone>volatility.exe -f "E:\Memory Forensics\xp-laptop-2005-06-25.i
mg" connscan
Volatility Foundation Volatility Framework 2.6
Offset(P)  Local Address             Remote Address            Pid
---------- ------------------------- ------------------------- ---
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

C:\Users\student\Downloads\volatility_2.6_win64_standalone>
```

Figure 7/connscan

volatility.exe -f c:\users\student\Desktop\Work\Labs\Evidence\lab5\xp-laptop-2005-06-25.img --profile WinXPSP2x86 hivelist

Referring to our help option this command hivelist is used to Print list of registry hives. Used to locate the virtual addresses of registry hives in memory and the full paths to the corresponding hive on disk.



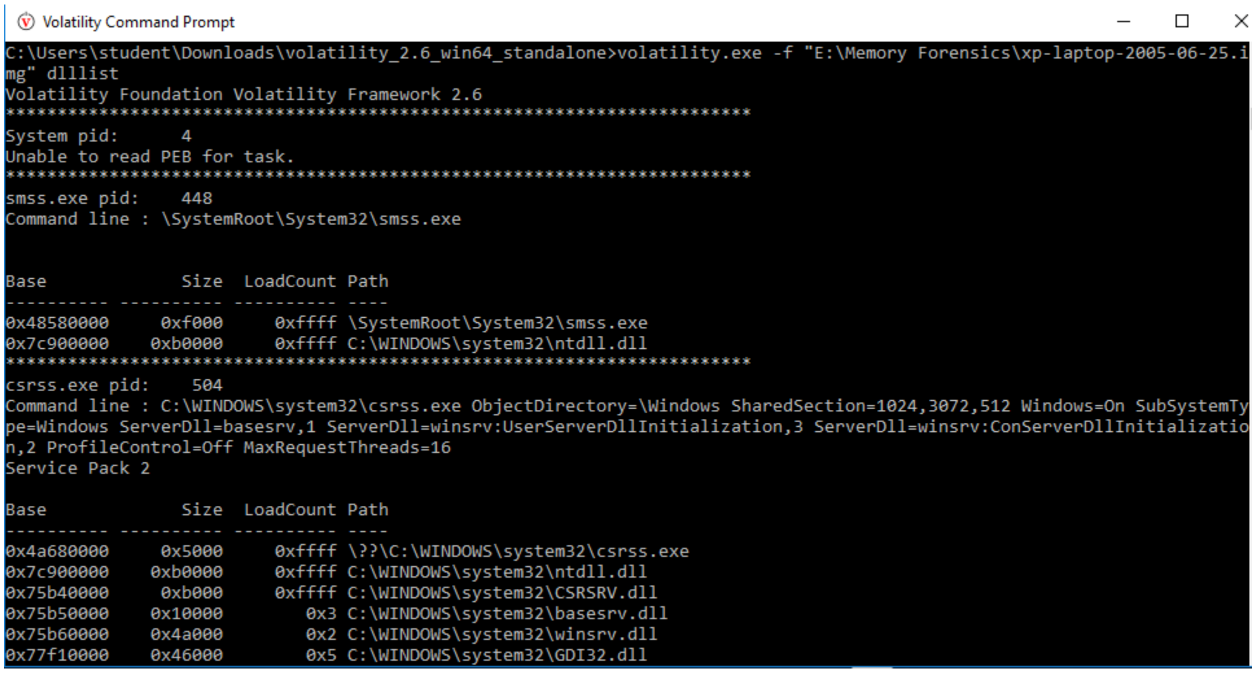
Volatility Command Prompt

```
C:\Users\student\Downloads\volatility_2.6_win64_standalone>volatility.exe -f "E:\Memory Forensics\xp-laptop-2005-06-25.i
mg" hivelist
Volatility Foundation Volatility Framework 2.6
Virtual    Physical   Name
---------- ---------- ----
0xe1ecd008 0x11221008 \Device\HarddiskVolume1\Documents and Settings\Sarah\Local Settings\Application Data\Microsoft\Win
dows\UsrClass.dat
0xe1eff758 0x1294a758 \Device\HarddiskVolume1\Documents and Settings\Sarah\NTUSER.DAT
0xe1bf9008 0x0e6d0008 \Device\HarddiskVolume1\Documents and Settings\LocalService\Local Settings\Application Data\Micros
oft\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\Micr
osoft\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]

C:\Users\student\Downloads\volatility_2.6_win64_standalone>
```

Figure 8/Hivelist

volatility.exe -f c:\users\student\Desktop\Work\Labs\Evidence\lab5\xp-laptop-2005-06-25.img --profile WinXPSP2x86 dlllist

This command is used to Print list of loaded dlls for each process. This command walks the doubly linked list of the _LDR_DATA_TABLE_ENTRY the structure of dlls is loaded into a process. The load count column tells you if a DLL was statically loaded (Andrea, 2017).

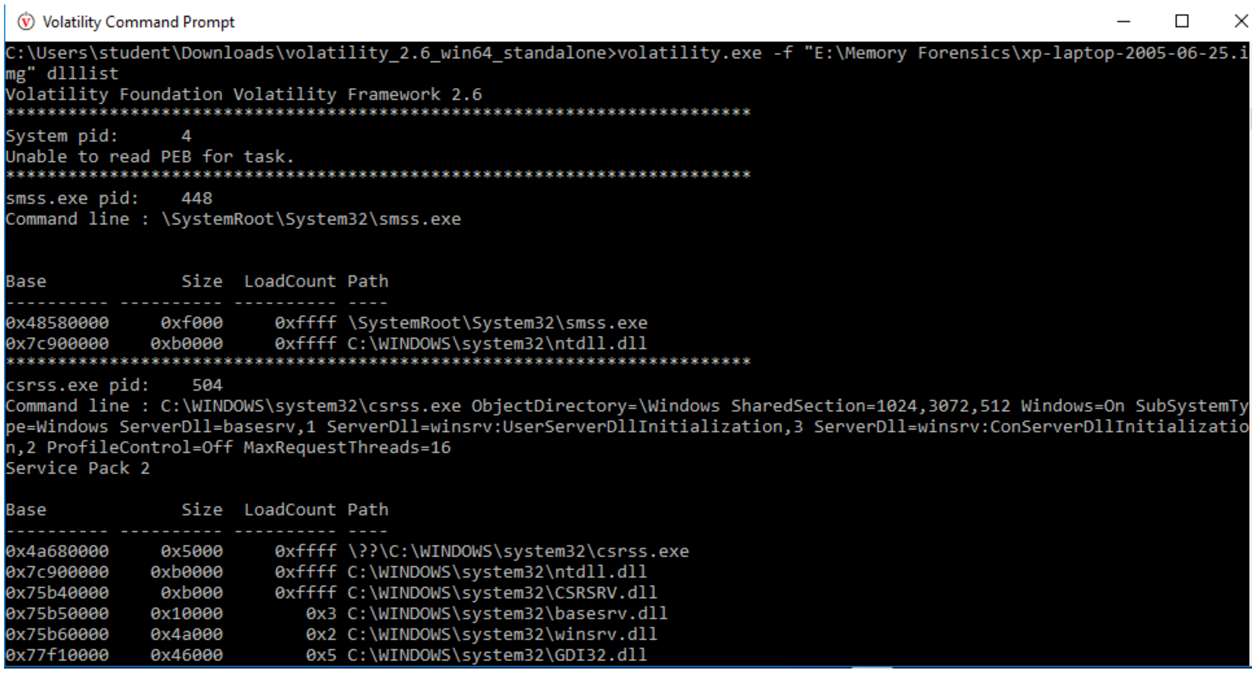


```
C:\Users\student\Downloads\volatility_2.6_win64_standalone>volatility.exe -f "E:\Memory Forensics\xp-laptop-2005-06-25.i
mg" dlllist
Volatility Foundation Volatility Framework 2.6
************************************************************************
System pid:      4
Unable to read PEB for task.
************************************************************************
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 SubSystemTy
pe=Windows ServerDll=basesrv,1 ServerDll=winsrv:UserServerDllInitialization,3 ServerDll=winsrv:ConServerDllInitializatio
n,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
```

Figure 9/dlllist

volatility.exe -f c:\users\student\Desktop\Work\Labs\Evidence\lab5\xp-laptop-2005-06-25.img --profile WinXPSP2x86 apihooks

This command detects API hooks in process and kernel memory. API hooking is a technique of which we can instrument and modify how API calls work (SecRat, 2014).



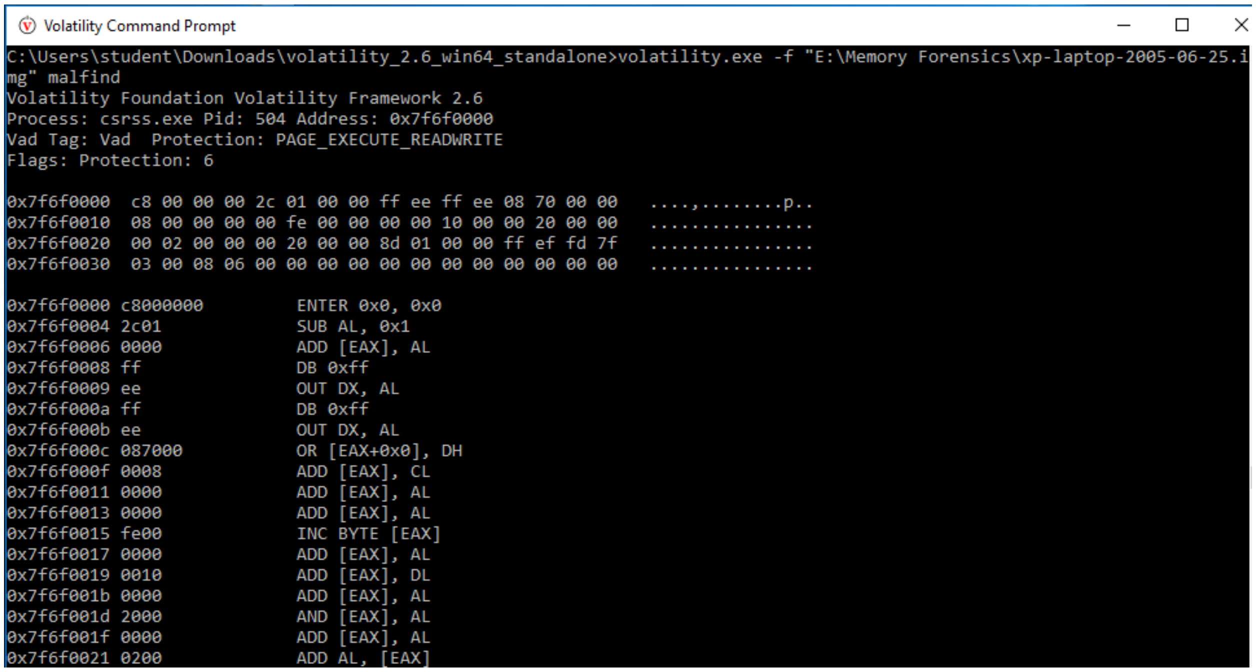
Volatility Command Prompt

```
C:\Users\student\Downloads\volatility_2.6_win64_standalone>volatility.exe -f "E:\Memory Forensics\xp-laptop-2005-06-25.i
mg" 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 SubSystemTy
pe=Windows ServerDll=basesrv,1 ServerDll=winsrv:UserServerDllInitialization,3 ServerDll=winsrv:ConServerDllInitializatio
n,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
```

Figure 10/apihooks

volatility.exe -f c:\users\student\Desktop\Work\Labs\Evidence\lab5\xp-laptop-2005-06-25.img --profile WinXPSP2x86 malfind

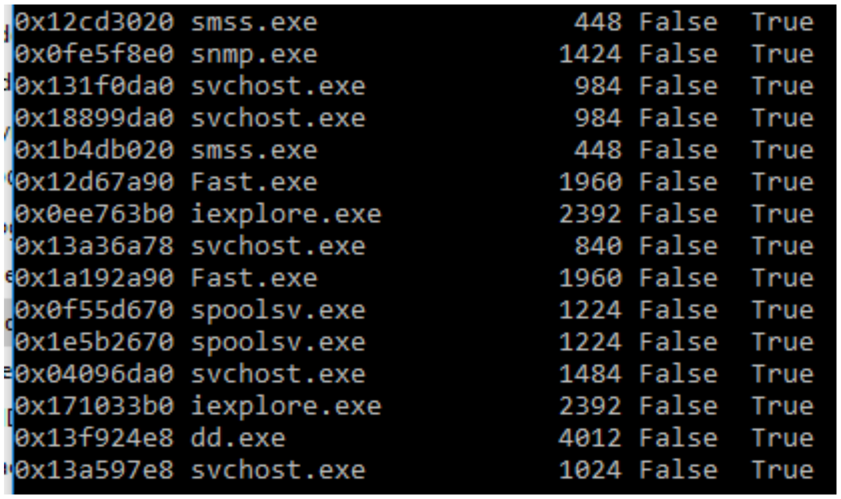
This option finds hidden and injected codes.



```
C:\Users\student\Downloads\volatility_2.6_win64_standalone>volatility.exe -f "E:\Memory Forensics\xp-laptop-2005-06-25.i
mg" malfind
Volatility Foundation Volatility Framework 2.6
Process: csrss.exe Pid: 504 Address: 0x7f6f0000
Vad Tag: Vad  Protection: PAGE_EXECUTE_READWRITE
Flags: Protection: 6

0x7f6f0000  c8 00 00 00 2c 01 00 00 ff ee ff ee 08 70 00 00   ....,........p..
0x7f6f0010  08 00 00 00 00 fe 00 00 00 00 10 00 00 20 00 00   ................
0x7f6f0020  00 02 00 00 00 20 00 00 8d 01 00 00 ff ef fd 7f   ................
0x7f6f0030  03 00 08 06 00 00 00 00 00 00 00 00 00 00 00 00   ................

0x7f6f0000 c8000000         ENTER 0x0, 0x0
0x7f6f0004 2c01             SUB AL, 0x1
0x7f6f0006 0000             ADD [EAX], AL
0x7f6f0008 ff               DB 0xff
0x7f6f0009 ee               OUT DX, AL
0x7f6f000a ff               DB 0xff
0x7f6f000b ee               OUT DX, AL
0x7f6f000c 087000           OR [EAX+0x0], DH
0x7f6f000f 0008             ADD [EAX], CL
0x7f6f0011 0000             ADD [EAX], AL
0x7f6f0013 0000             ADD [EAX], AL
0x7f6f0015 fe00             INC BYTE [EAX]
0x7f6f0017 0000             ADD [EAX], AL
0x7f6f0019 0010             ADD [EAX], DL
0x7f6f001b 0000             ADD [EAX], AL
0x7f6f001d 2000             AND [EAX], AL
0x7f6f001f 0000             ADD [EAX], AL
0x7f6f0021 0200             ADD AL, [EAX]
```

Figure 11/malfind

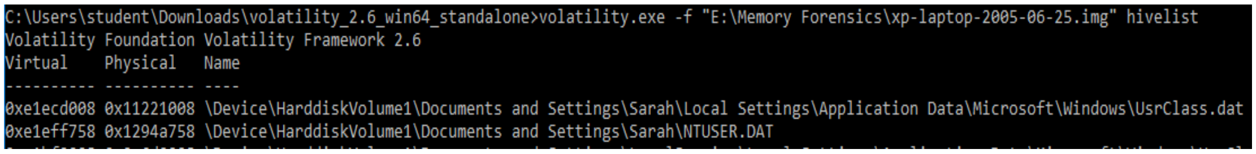
This computer had hidden processes as shown by the psscan command output since it returned duplicate processes but with the different offset. There are more details into this as the duplicate entries didn’t have exited time to show that the process was inactive and couldn’t be captured by the pslist. This shows anomalies in the two commands output. This can be extracted by running the psxview command which compares and returns the output in the figure below.



```
0x12cd3020 smss.exe              448 False  True
0x0fe5f8e0 snmp.exe             1424 False  True
0x131f0da0 svchost.exe           984 False  True
0x18899da0 svchost.exe           984 False  True
0x1b4db020 smss.exe              448 False  True
0x12d67a90 Fast.exe             1960 False  True
0x0ee763b0 iexplore.exe         2392 False  True
0x13a36a78 svchost.exe           840 False  True
0x1a192a90 Fast.exe             1960 False  True
0x0f55d670 spoolsv.exe          1224 False  True
0x1e5b2670 spoolsv.exe          1224 False  True
0x04096da0 svchost.exe          1484 False  True
0x171033b0 iexplore.exe         2392 False  True
0x13f924e8 dd.exe               4012 False  True
0x13a597e8 svchost.exe          1024 False  True
```

Figure 12/Hidden

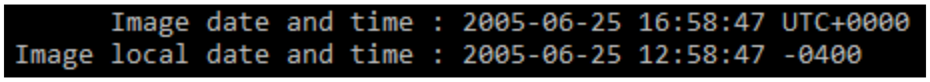
The primary user of this computer is Sarah as we can fetch this from the hivelist command and the getsids shows a consistent output of Sarah as the primary user. This is shown in the figure below.



```
C:\Users\student\Downloads\volatility_2.6_win64_standalone>volatility.exe -f "E:\Memory Forensics\xp-laptop-2005-06-25.img" hivelist
Volatility Foundation Volatility Framework 2.6
Virtual    Physical   Name
---------- ---------- ----
0xe1ecd008 0x11221008 \Device\HarddiskVolume1\Documents and Settings\Sarah\Local Settings\Application Data\Microsoft\Windows\UsrClass.dat
0xe1eff758 0x1294a758 \Device\HarddiskVolume1\Documents and Settings\Sarah\NTUSER.DAT
```

Figure 13/user

The system time is shown in the figure below after running imageinfo which returns the local time and the time the image was captured.

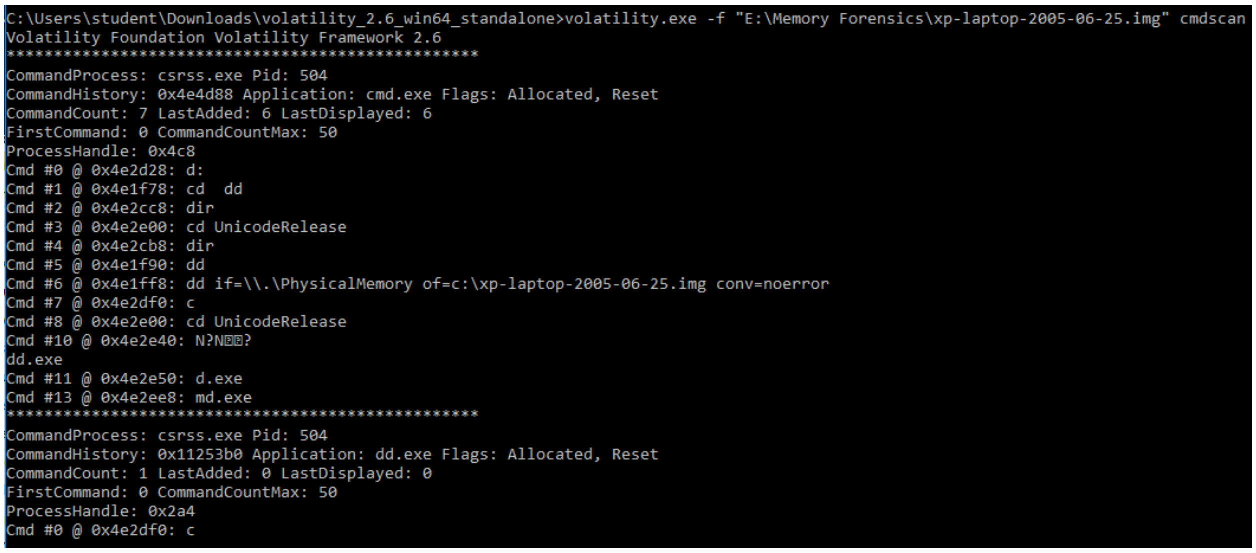


```
     Image date and time : 2005-06-25 16:58:47 UTC+0000
Image local date and time : 2005-06-25 12:58:47 -0400
```

Figure 14/System Time

The browser running in this system were firefox.exe and iexplore.exe. These details are shown in the psscan and the pslist commands.

The command typed/running in the command prompt are shown in the figure below after running the command cmdscan or consoles on the image. The command is shown in the 6th entry below.



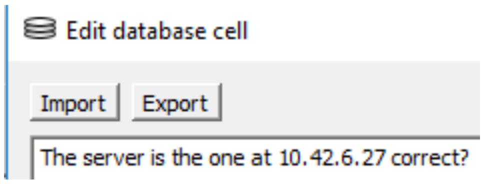
```
C:\Users\student\Downloads\volatility_2.6_win64_standalone>volatility.exe -f "E:\Memory Forensics\xp-laptop-2005-06-25.img" cmdscan
Volatility Foundation Volatility Framework 2.6
**************************************************
CommandProcess: csrss.exe Pid: 504
CommandHistory: 0x4e4d88 Application: 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
```

Figure 15/Command Prompt

Using the malfind command it returned the processes that were potentially injected with malware which were: Process: csrss.exe Pid: 504 Address: 0x7f6f0000, Process: 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.

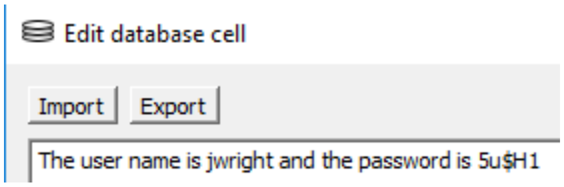
Mobile Device Filesystem Forensics

We will be doing mobile forensics as we populate the valuable questions associated with the mobile phone. We did access the SMS database and look for login credentials and wireless network credentials that were on this device using the SQLite Database Browser. We got the following credentials showing the username, password and the passphrase as shown in the figures below.



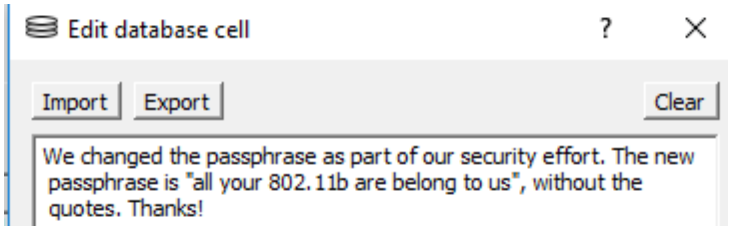
The server is the one at 10.42.6.27 correct?

Figure 16/Server



The user name is jwright and the password is 5u$H1

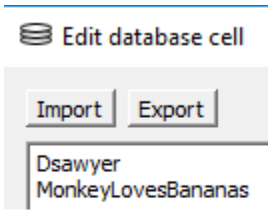
Figure 17/username



We changed the passphrase as part of our security effort. The new passphrase is "all your 802.11b are belong to us", without the quotes. Thanks!

Figure 18/Passphrase

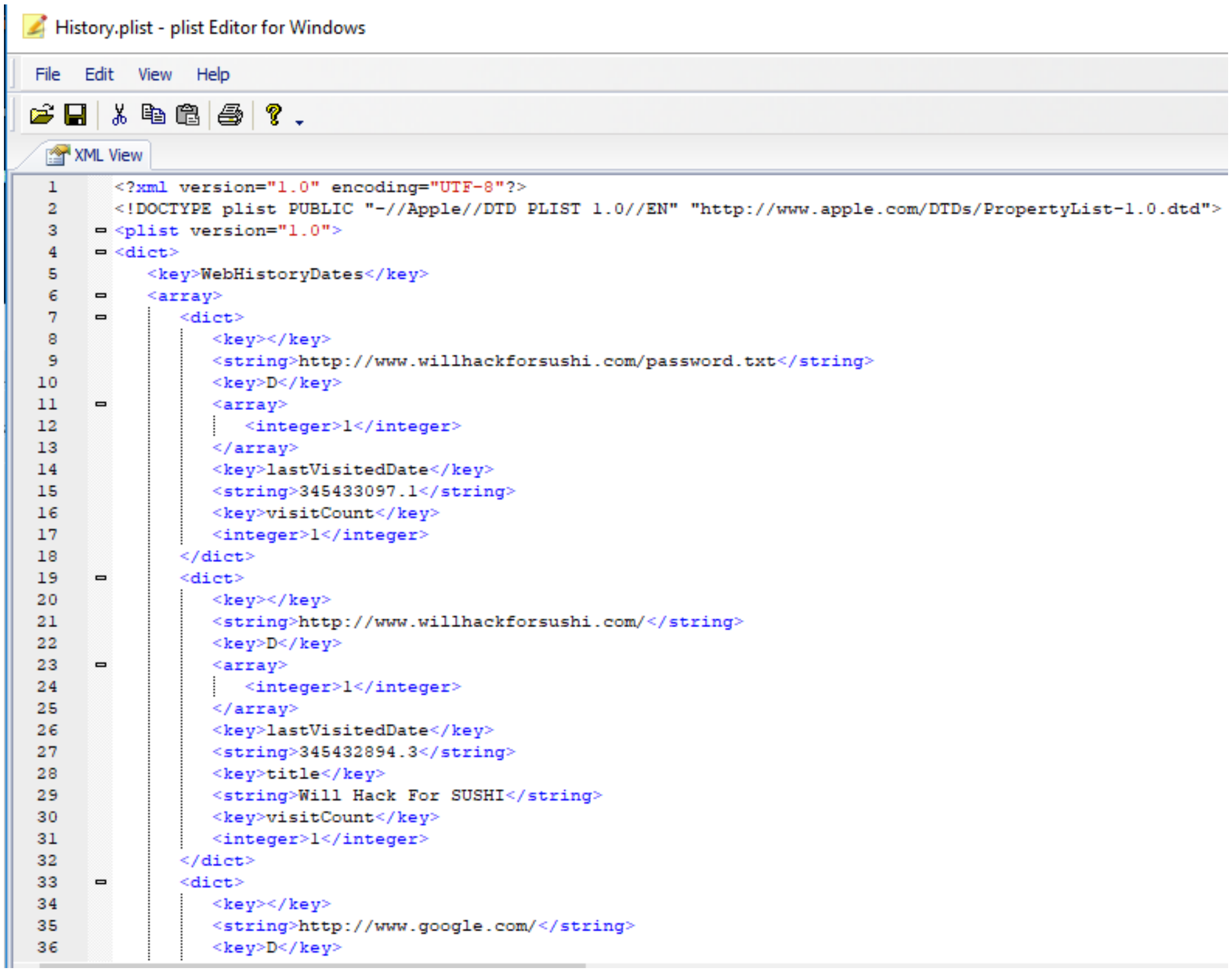
The salesforce.com credentials found in the notes database contained a username and a password as shown in the figure below:



Dsawyer
MonkeyLovesBananas

Figure 19/Salesforce

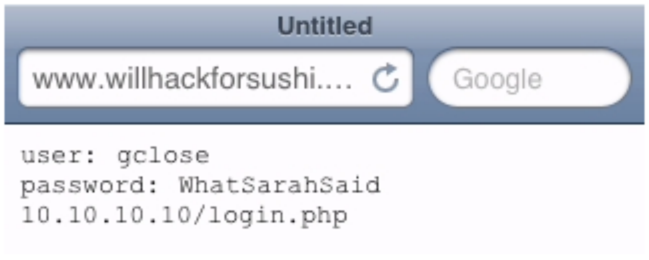
The safari History plist file had a website that had a password document which is password.txt as shown in the figure below.



```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
   <key>WebHistoryDates</key>
   <array>
      <dict>
         <key></key>
         <string>http://www.willhackforsushi.com/password.txt</string>
         <key>D</key>
         <array>
            <integer>1</integer>
         </array>
         <key>lastVisitedDate</key>
         <string>345433097.1</string>
         <key>visitCount</key>
         <integer>1</integer>
      </dict>
      <dict>
         <key></key>
         <string>http://www.willhackforsushi.com/</string>
         <key>D</key>
         <array>
            <integer>1</integer>
         </array>
         <key>lastVisitedDate</key>
         <string>345432894.3</string>
         <key>title</key>
         <string>Will Hack For SUSHI</string>
         <key>visitCount</key>
         <integer>1</integer>
      </dict>
      <dict>
         <key></key>
         <string>http://www.google.com/</string>
         <key>D</key>
```

Figure 20/Website

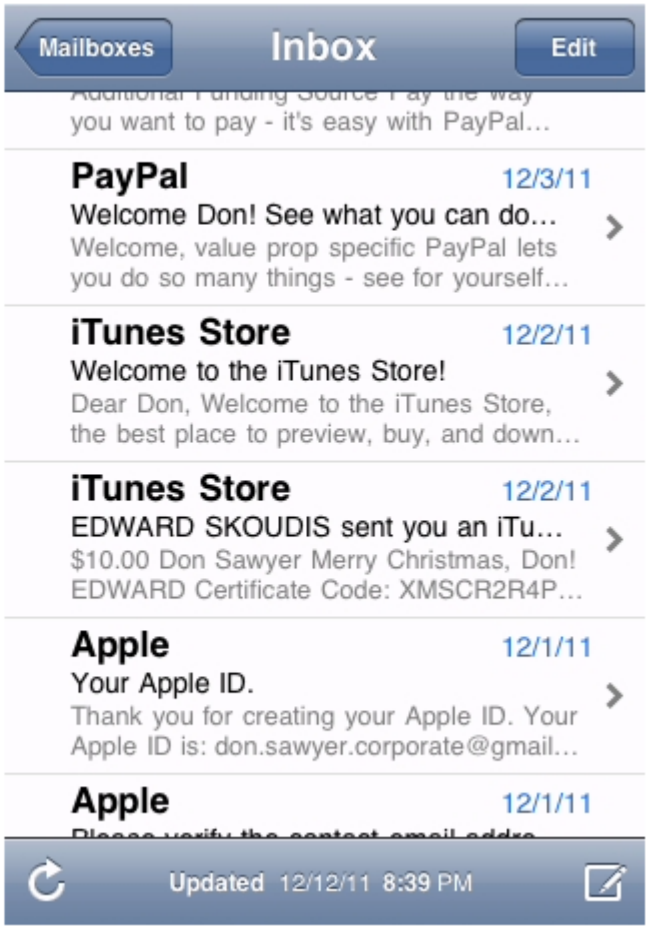
The last screen seen in the browser shows credentials found in the safari history as shown in figure below.



```
user: gclose
password: WhatSarahSaid
10.10.10.10/login.php
```

Figure 21/Snapshot

The email snapshot shows the email of don sawyer as he could be the owner of the mobile phone as shown below.



you want to pay - it's easy with PayPal...

PayPal 12/3/11
Welcome Don! See what you can do...
Welcome, value prop specific PayPal lets you do so many things - see for yourself...

iTunes Store 12/2/11
Welcome to the iTunes Store!
Dear Don, Welcome to the iTunes Store, the best place to preview, buy, and down...

iTunes Store 12/2/11
EDWARD SKOUDIS sent you an iTu...
$10.00 Don Sawyer Merry Christmas, Don! EDWARD Certificate Code: XMSCR2R4P...

Apple 12/1/11
Your Apple ID.
Thank you for creating your Apple ID. Your Apple ID is: don.sawyer.corporate@gmail...

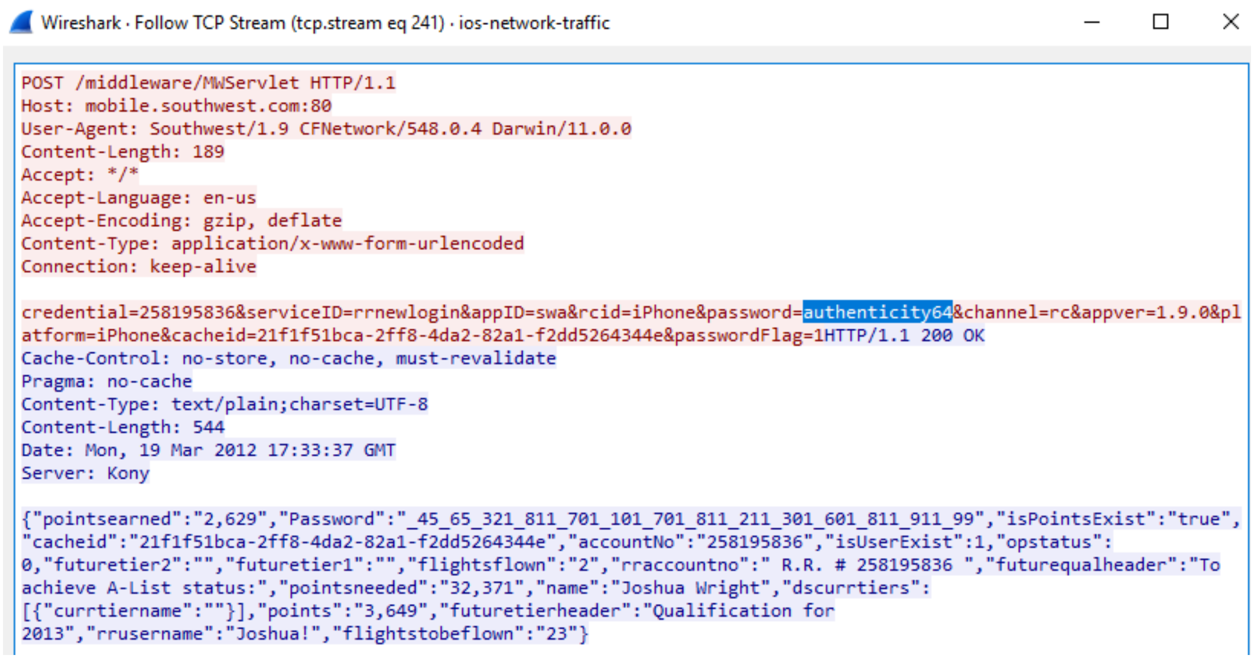
Apple 12/1/11

Updated 12/12/11 8:39 PM

Figure 22/email

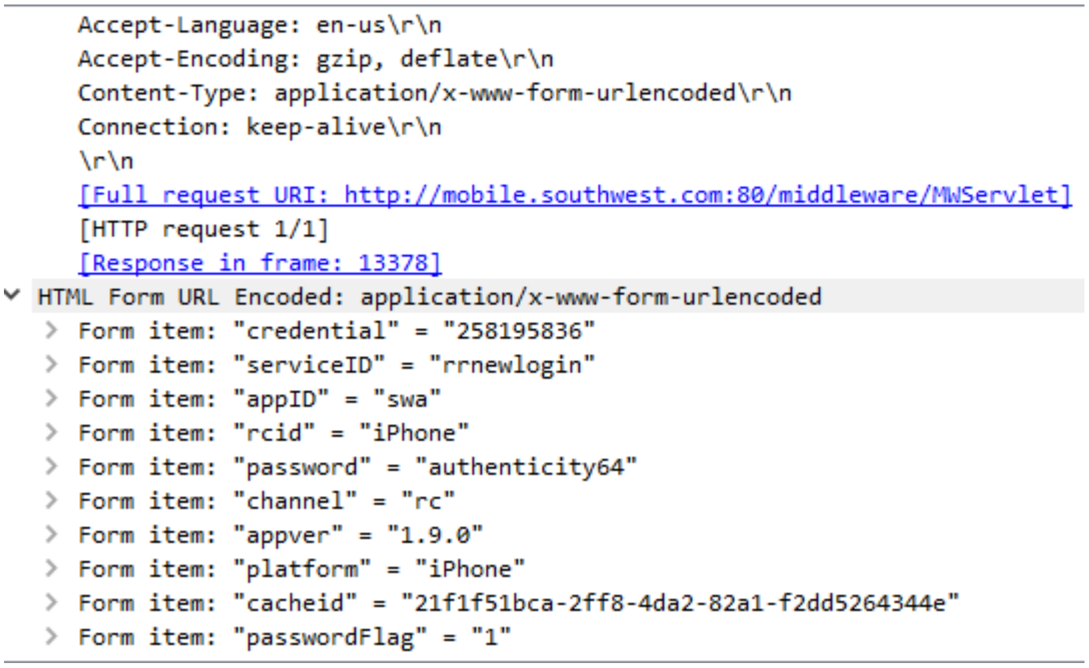
Mobile Device Network Forensics

We ran Wireshark an application used to capture network traffic to view the password which was authenticity64 used to log in in a mobile app of the southwest website. The screenshots below show the verified details.


Wireshark · Follow TCP Stream (tcp.stream eq 241) · ios-network-traffic

```
POST /middleware/MWServlet HTTP/1.1
Host: mobile.southwest.com:80
User-Agent: Southwest/1.9 CFNetwork/548.0.4 Darwin/11.0.0
Content-Length: 189
Accept: */*
Accept-Language: en-us
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Connection: keep-alive

credential=258195836&serviceID=rrnewlogin&appID=swa&rcid=iPhone&password=authenticity64&channel=rc&appver=1.9.0&pl
atform=iPhone&cacheid=21f1f51bca-2ff8-4da2-82a1-f2dd5264344e&passwordFlag=1HTTP/1.1 200 OK
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Content-Type: text/plain;charset=UTF-8
Content-Length: 544
Date: Mon, 19 Mar 2012 17:33:37 GMT
Server: Kony

{"pointsearned":"2,629","Password":"_45_65_321_811_701_101_701_811_211_301_601_811_911_99","isPointsExist":"true",
"cacheid":"21f1f51bca-2ff8-4da2-82a1-f2dd5264344e","accountNo":"258195836","isUserExist":1,"opstatus":
0,"futuretier2":"","futuretier1":"","flightsflown":"2","rraccountno":" R.R. # 258195836 ","futurequalheader":"To
achieve A-List status:","pointsneeded":"32,371","name":"Joshua Wright","dscurrtiers":
[{"currtiername":""}],"points":"3,649","futuretierheader":"Qualification for
2013","rrusername":"Joshua!","flightstobeflown":"23"}
```

Figure 23/Password



```
    Accept-Language: en-us\r\n
    Accept-Encoding: gzip, deflate\r\n
    Content-Type: application/x-www-form-urlencoded\r\n
    Connection: keep-alive\r\n
    \r\n
    [Full request URI: http://mobile.southwest.com:80/middleware/MWServlet]
    [HTTP request 1/1]
    [Response in frame: 13378]
v HTML Form URL Encoded: application/x-www-form-urlencoded
  > Form item: "credential" = "258195836"
  > Form item: "serviceID" = "rrnewlogin"
  > Form item: "appID" = "swa"
  > Form item: "rcid" = "iPhone"
  > Form item: "password" = "authenticity64"
  > Form item: "channel" = "rc"
  > Form item: "appver" = "1.9.0"
  > Form item: "platform" = "iPhone"
  > Form item: "cacheid" = "21f1f51bca-2ff8-4da2-82a1-f2dd5264344e"
  > Form item: "passwordFlag" = "1"
```

Figure 24/Wireshark

Glossary

API-Stands for Application Programming Interface are a set of rules or protocols that are used in software applications to communicate, features and functionality (Michael, 2024).

TCP -Stands for Transmission Control Protocol is a standard protocol on the internet that ensures the reliable communication on a network (Kinza, 2024).

References

Andrea, F. (2017, July 3). *Volatility, my own cheatsheet (Part 2): Processes and DLLs*. Retrieved from andreafortuna.org: https://andreafortuna.org/2017/07/03/volatility-my-own-cheatsheet-part-2-processes-and-dlls/#:~:text=Learn%20how%20to%20use%20Volatility%20plugins

Kinza, Y. (2024, June 13). *Transmission Control Protocol (TCP)*. Retrieved from TechTarget: https://www.techtarget.com/searchnetworking/definition/TCP#:~:text=Transmission%20Control%20Protocol%20(TCP)%20is%20a

Michael, G. (2024, April 9). *What is an API (application programming interface)?* Retrieved from IBM: https://www.ibm.com/topics/api#:~:text=Learn%20what%20an%20API%20is,%20how

Rohit, T., Oleg, S., Heather, M., & Satish, B. (2020). *Practical Mobile Forensics.* Birmingham: Packt Publishing Ltd.

SecRat. (2014, April 22). *API hooking*. Retrieved from Infosec: https://www.infosecinstitute.com/resources/hacking/api-hooking/

Sengul, D., & Erhan, A. (2017). Analysis of mobile phones in digital forensics. *2017 40th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)* (pp. 1241-1244). Opatija, Croatia: IEEE.

Sneha, C. S., & Nilima, M. D. (2018). Data acquisition techniques in mobile forensics. *2018 2nd International Conference on Inventive Systems and Control (ICISC)* (pp. 280-286). Coimbatore, India: IEEE.