INTERNATIONAL CYBERSECURITY AND DIGITAL FORENSICS ACADEMY

Assignment Title: Memory Forensics

Course Code: CI901 Cybercrime Investigations Case Studies

Student Name: Ahtisham Tanveer

Student ID: 2025/ACI/9979

Programme: Advance Cybercrime Investigations

Instructor Name: Aminu Idris

Date of Submission: 01/10/2026

1. Environment Setup and Methodology

To conduct this investigation, a specialized forensic environment was established using the **Volatility 2 Framework**.

- **Environment Configuration:** The analysis was performed on a Linux-based system (typically Kali Linux or a similar forensic workstation) where Volatility 2 was installed. This framework was chosen for its robust ability to parse Windows memory artifacts using specific profiles.
- **Memory Source:** The primary evidence file, `memdumpWin7.mem`, was acquired from a target Windows machine. This file is a raw image of the system's RAM at a specific point in time, captured using a memory acquisition tool (such as FTK Imager or DumpIt) while the system was still powered on. This allows for the analysis of "volatile" data that would otherwise be lost if the computer were shut down.

2. Forensic Topics and Analysis Summary

Memory and Memory Dumps

A memory dump is the process of capturing all information currently held in RAM and writing it to a storage drive. These binary files often referred to as core dumps or the "Blue Screen of Death" (BSOD) in Windows are essential for forensic investigators to identify the runtime state of a machine.

Virtual Memory and Pagefile.sys

Windows utilizes virtual memory to allow processes to use more memory than is physically available. When physical RAM is exhausted, the system offloads data to `Pagefile.sys` on the disk. Analyzing this helps recover data from inactive applications.

Kernel Processor Control Region (KPCR)

The KPCR is a kernel-level data structure that stores information for each processor. By locating the KPCR within a memory dump, Volatility can traverse the system's architecture to find the list of active processes and threads.

The Volatility Framework

Volatility is an open-source framework for extracting digital artifacts from volatile memory. It supports Windows, Linux, and iOS, allowing investigators to reconstruct system activity, such as open network connections, running processes, and loaded DLLs.

Registry Analysis and SIDs

The Windows Registry stores configuration data for the system and users. Every user is assigned a unique Security Identifier (SID). Analyzing registry hives within memory helps link specific actions or files to a particular user account.

Network Forensics

By scanning memory for network objects, we can identify active connections and suspicious IP addresses. Using specialized plugins, we can determine which Process ID (PID) was responsible for specific network traffic, such as a browser downloading a malicious file.

Command History Investigation

Plugins like `cmdscan` and `consoles` are used to extract the history of commands entered into the command prompt. This provides a direct look at the suspect's actions, such as file deletions or unauthorized data transfers.

USB Forensics and Instance IDs

When a USB device is connected, Windows creates registry entries and assigns a unique Instance ID. This allows investigators to prove a specific external drive was plugged into the computer, including the exact time and assigned drive letter.

Shellbags and File Explorer History

Shellbags track folder viewing preferences in File Explorer. These are critical for showing that a user navigated through specific folders, even if those folders were later deleted or resided on a removable USB drive.

Timeline Analysis

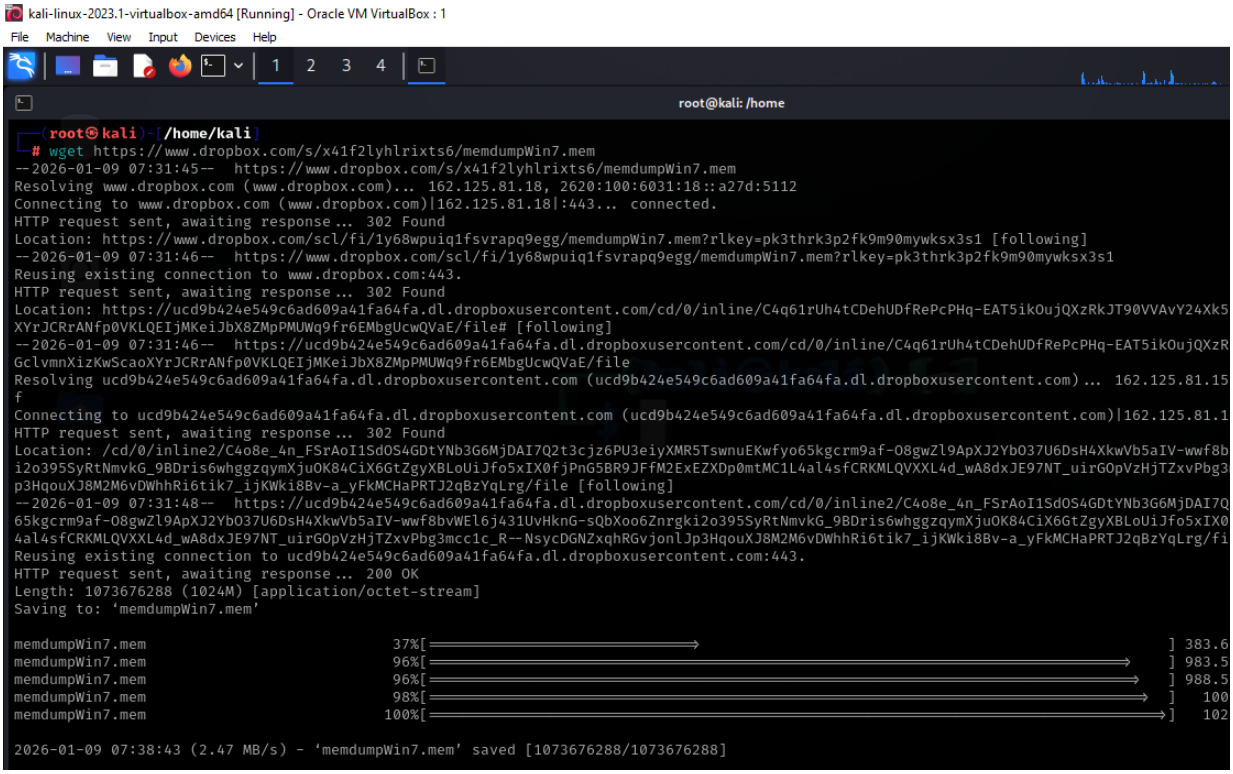
This is the process of merging various artifacts browser history, file access, and USB connections into a single chronological list. It allows for the reconstruction of a suspect’s "story," such as downloading a sensitive document and immediately moving it to a flash drive.

Password Cracking and Hash Dumping

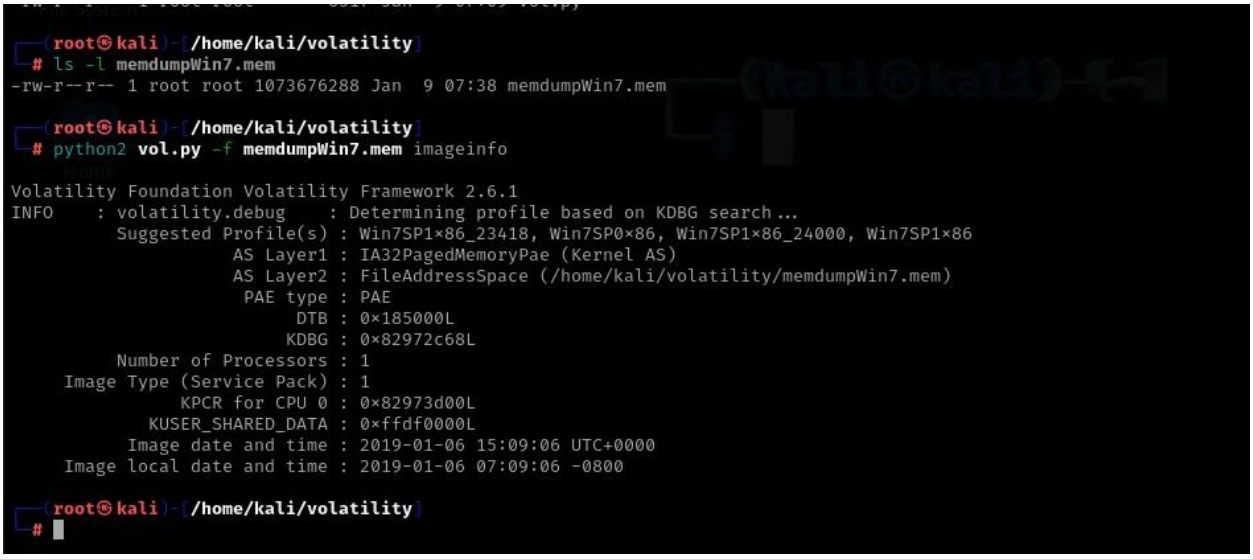
Memory forensics allows for the extraction of NTLM password hashes from the `LSASS` process in RAM. These hashes can then be processed through tools like John the Ripper to recover the actual plaintext passwords of the system users.

PRACTICAL

Download the memory image for investigations:



Identify the image profile:



Show all registry keys:

```
┌──(root㉿kali)-[/home/kali/volatility]
└─# python2 vol.py -f memdumpWin7.mem --profile=Win7SP1x86 pslist

Volatility Foundation Volatility Framework 2.6.1
Offset(V)  Name                    PID   PPID   Thds     Hnds   Sess  Wow64 Start                          Exit
---------- -------------------- ------ ------ ------ -------- ------ ------ ------------------------------ ------------------------------
0x8413a908 System                    4      0     73      518 ------      0 2019-01-06 18:02:44 UTC+0000
0x851ec488 smss.exe                236      4      2       29 ------      0 2019-01-06 18:02:44 UTC+0000
0x852b11e0 csrss.exe               300    292      8      434      0      0 2019-01-06 18:02:44 UTC+0000
0x841c32b8 wininit.exe             340    292      3       75      0      0 2019-01-06 18:02:44 UTC+0000
0x852b21d0 csrss.exe               348    328      9      316      1      0 2019-01-06 18:02:44 UTC+0000
0x858a4d28 winlogon.exe            388    328      3      112      1      0 2019-01-06 18:02:45 UTC+0000
0x858b0310 services.exe            432    340     10      228      0      0 2019-01-06 18:02:45 UTC+0000
0x858bbd28 lsass.exe               440    340      7      581      0      0 2019-01-06 18:02:45 UTC+0000
0x858be2c0 lsm.exe                 448    340     10      147      0      0 2019-01-06 18:02:45 UTC+0000
0x858f88d8 svchost.exe             552    432     10      359      0      0 2019-01-06 18:02:45 UTC+0000
0x8590cb00 VBoxService.ex          616    432     13      123      0      0 2019-01-06 18:02:45 UTC+0000
0x85913030 svchost.exe             672    432      8      260      0      0 2019-01-06 15:02:47 UTC+0000
0x85761030 svchost.exe             724    432     18      420      0      0 2019-01-06 15:02:47 UTC+0000
0x858d9208 svchost.exe             828    432     21      484      0      0 2019-01-06 15:02:47 UTC+0000
0x859d6518 svchost.exe             880    432     31      983      0      0 2019-01-06 15:02:47 UTC+0000
0x84f992c0 svchost.exe             992    432     13      275      0      0 2019-01-06 15:02:48 UTC+0000
0x85a23ce8 svchost.exe            1064    432     14      367      0      0 2019-01-06 15:02:48 UTC+0000
0x85a4fd28 spoolsv.exe            1184    432     12      272      0      0 2019-01-06 15:02:48 UTC+0000
0x85a91030 svchost.exe            1228    432     17      315      0      0 2019-01-06 15:02:48 UTC+0000
0x85ade488 vmicsvc.exe            1364    432      4       94      0      0 2019-01-06 15:02:48 UTC+0000
0x85ae3030 vmicsvc.exe            1388    432      5      105      0      0 2019-01-06 15:02:48 UTC+0000
0x85ae9030 taskhost.exe           1412    432      8      145      1      0 2019-01-06 15:02:48 UTC+0000
0x85af6d28 vmicsvc.exe            1440    432      3       66      0      0 2019-01-06 15:02:48 UTC+0000
0x85b06d28 vmicsvc.exe            1492    432      4       80      0      0 2019-01-06 15:02:48 UTC+0000
0x850553f0 vmicsvc.exe            1548    432      4       81      0      0 2019-01-06 15:02:48 UTC+0000
0x85b2dd28 svchost.exe            1588    432     10      147      0      0 2019-01-06 15:02:48 UTC+0000
0x859dc030 cygrunsrv.exe          1680    432      6      100      0      0 2019-01-06 15:02:48 UTC+0000
0x859a0d28 wlms.exe               1752    432      4       45      0      0 2019-01-06 15:02:48 UTC+0000
0x859bab00 taskeng.exe            1776    880      6       85      0      0 2019-01-06 15:02:48 UTC+0000
0x8500bd00 cygrunsrv.exe          1984   1680      0 ------      0      0 2019-01-06 15:02:49 UTC+0000   2019-01-06 15:02:49 UTC+0000
```

```
┌──(root㉿kali)-[/home/kali/volatility]
└─# python2 vol.py -f memdumpWin7.mem --profile=Win7SP1x86 printkey
Volatility Foundation Volatility Framework 2.6.1
Legend: (S) = Stable   (V) = Volatile

----------------------------
Registry: \REGISTRY\MACHINE\BCD00000000
Key name: NewStoreRoot (S)
Last updated: 2019-01-06 18:02:44 UTC+0000

Subkeys:
  (S) Description
  (S) Objects

Values:
----------------------------
Registry: \??\C:\Users\IEUser\ntuser.dat
Key name: CMI-CreateHive{6A1C4018-979D-4291-A7DC-7AED1C75B67C} (S)
Last updated: 2019-01-06 15:02:48 UTC+0000

Subkeys:
  (S) AppEvents
  (S) Console
  (S) Control Panel
  (S) Environment
  (S) EUDC
  (S) Keyboard Layout
  (S) Network
  (S) Printers
  (S) Software
  (S) System
  (V) Volatile Environment

Values:
----------------------------
```

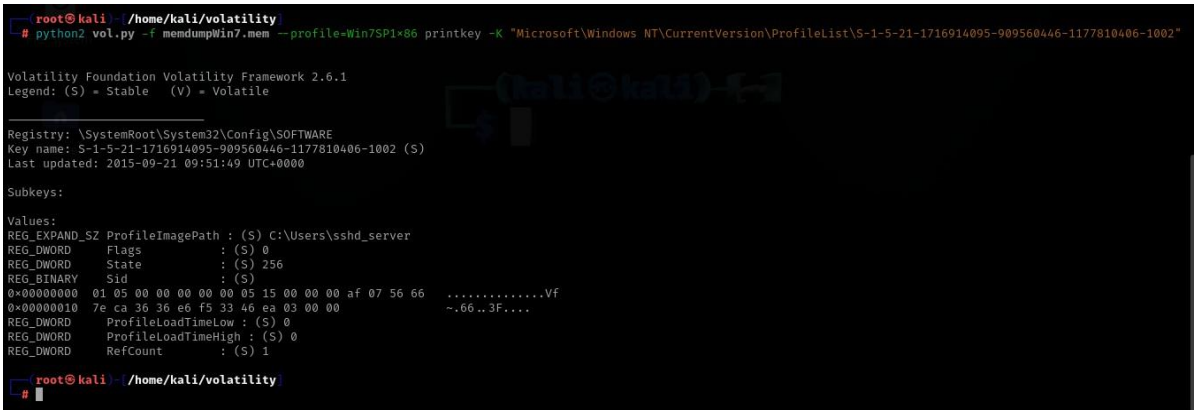
Who was using the device?

```
┌──(root㉿kali)-[/home/kali/volatility]
└─# python2 vol.py -f memdumpWin7.mem --profile=Win7SP1x86 printkey -K "Volatile Environment"
Volatility Foundation Volatility Framework 2.6.1
Legend: (S) = Stable   (V) = Volatile

----------------------------
Registry: \??\C:\Users\IEUser\ntuser.dat
Key name: Volatile Environment (V)
Last updated: 2019-01-06 15:02:48 UTC+0000

Subkeys:
  (V) 1

Values:
REG_SZ        LOGONSERVER     : (V) \\IE8WIN7
REG_SZ        USERDOMAIN      : (V) IE8WIN7
REG_SZ        USERNAME        : (V) IEUser
REG_SZ        USERPROFILE     : (V) C:\Users\IEUser
REG_SZ        HOMEPATH        : (V) \Users\IEUser
REG_SZ        HOMEDRIVE       : (V) C:
REG_SZ        APPDATA         : (V) C:\Users\IEUser\AppData\Roaming
REG_SZ        LOCALAPPDATA    : (V) C:\Users\IEUser\AppData\Local

┌──(root㉿kali)-[/home/kali/volatility]
└─#
```

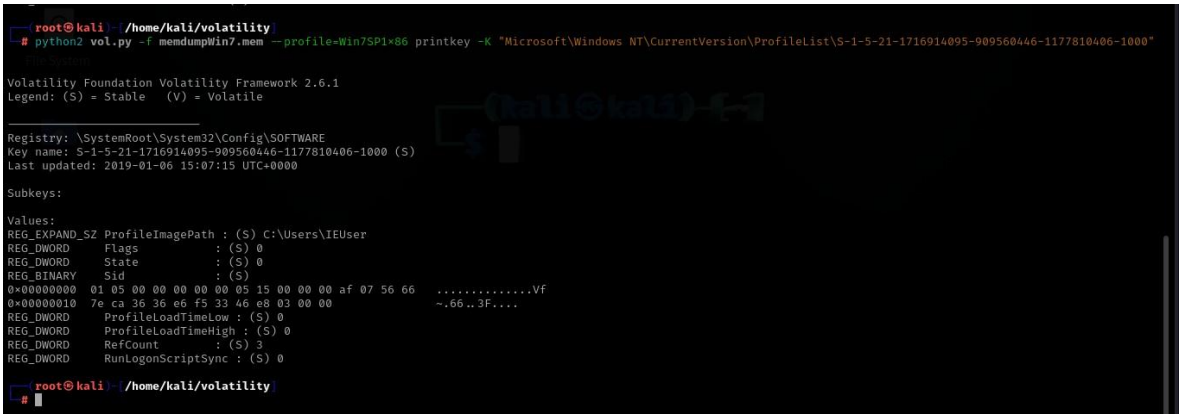
Who are associated with the device?

```
┌──(root㉿kali)-[/home/kali/volatility]
└─# python2 vol.py -f memdumpWin7.mem --profile=Win7SP1x86 printkey -K "Microsoft\Windows NT\CurrentVersion\ProfileList"
Volatility Foundation Volatility Framework 2.6.1
Legend: (S) = Stable   (V) = Volatile

----------------------------
Registry: \SystemRoot\System32\Config\SOFTWARE
Key name: ProfileList (S)
Last updated: 2015-09-21 09:50:52 UTC+0000

Subkeys:
  (S) S-1-5-18
  (S) S-1-5-19
  (S) S-1-5-20
  (S) S-1-5-21-1716914095-909560446-1177810406-1000
  (S) S-1-5-21-1716914095-909560446-1177810406-1002

Values:
REG_EXPAND_SZ ProfilesDirectory : (S) %SystemDrive%\Users
REG_EXPAND_SZ Default           : (S) %SystemDrive%\Users\Default
REG_EXPAND_SZ Public            : (S) %SystemDrive%\Users\Public
REG_EXPAND_SZ ProgramData       : (S) %SystemDrive%\ProgramData

┌──(root㉿kali)-[/home/kali/volatility]
└─#
```

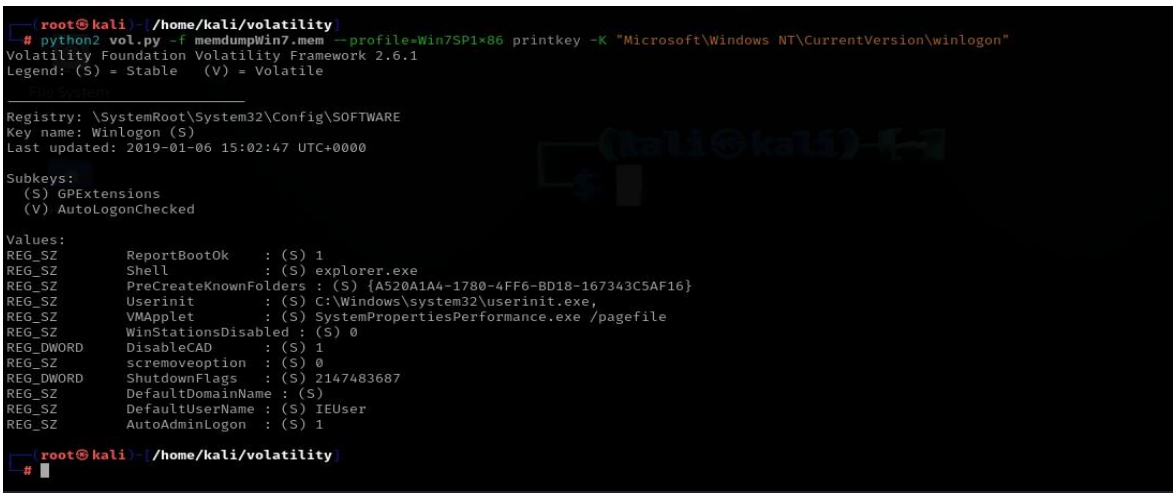
Who has SID= S-1-5-21-1716914095-909560446-1177810406-1002?



Who has SID= S-1-5-21-1716914095-909560446-1177810406-1000?



Who is the default logon user?



Complete process tree:

```
┌──(root㉿kali)-[/home/kali/volatility]
└─# python2 vol.py -f memdumpWin7.mem --profile=Win7SP1x86 pstree

Volatility Foundation Volatility Framework 2.6.1
Name                                                  Pid   PPid   Thds   Hnds Time
-------------------------------------------------- ------ ------ ------ ------ ----
 0x85bba860:explorer.exe                             2524   2500     32    952 2019-01-06 15:02:54 UTC+0000
. 0x8431fb08:cmd.exe                                 3996   2524      1     22 2019-01-06 15:06:11 UTC+0000
. 0x85c7f030:VBoxTray.exe                            2632   2524     13    140 2019-01-06 15:02:55 UTC+0000
. 0x8583a030:chrome.exe                              2388   2524     33    819 2019-01-06 15:04:30 UTC+0000
.. 0x851e8520:chrome.exe                             1280   2388      7     82 2019-01-06 15:04:30 UTC+0000
.. 0x842eb6b0:chrome.exe                              976   2388     14    198 2019-01-06 15:06:23 UTC+0000
.. 0x850316b8:chrome.exe                             2380   2388      2     57 2019-01-06 15:04:30 UTC+0000
.. 0x842b4d28:chrome.exe                             2912   2388      9    159 2019-01-06 15:04:32 UTC+0000
. 0x84312030:FTK Imager.exe                          2596   2524     14    358 2019-01-06 15:07:15 UTC+0000
 0x852b11e0:csrss.exe                                 300    292      8    434 2019-01-06 18:02:44 UTC+0000
. 0x84202030:conhost.exe                             2000    300      2     33 2019-01-06 15:02:49 UTC+0000
 0x841c32b8:wininit.exe                               340    292      3     75 2019-01-06 18:02:44 UTC+0000
. 0x858b0310:services.exe                             432    340     10    228 2019-01-06 18:02:45 UTC+0000
.. 0x85ae9030:taskhost.exe                           1412    432      8    145 2019-01-06 15:02:48 UTC+0000
.. 0x85ae3030:vmicsvc.exe                            1388    432      5    105 2019-01-06 15:02:48 UTC+0000
.. 0x85b06d28:vmicsvc.exe                            1492    432      4     80 2019-01-06 15:02:48 UTC+0000
.. 0x85b11398:sppsvc.exe                             1036    432      4    143 2019-01-06 15:02:50 UTC+0000
.. 0x859dc030:cygrunsrv.exe                          1680    432      6    100 2019-01-06 15:02:48 UTC+0000
... 0x8500bd00:cygrunsrv.exe                         1984   1680      0 ------ 2019-01-06 15:02:49 UTC+0000
.... 0x84203998:sshd.exe                             2016   1984      4    100 2019-01-06 15:02:49 UTC+0000
.. 0x85a07030:svchost.exe                            2072    432      5     91 2019-01-06 15:02:50 UTC+0000
.. 0x85913030:svchost.exe                             672    432      8    260 2019-01-06 15:02:47 UTC+0000
.. 0x84fa2ad0:msiexec.exe                            3748    432      4    174 2019-01-06 15:03:34 UTC+0000
.. 0x858f88d8:svchost.exe                             552    432     10    359 2019-01-06 18:02:45 UTC+0000
... 0x850d5a38:dllhost.exe                           2228    552      6 17 ... 8 2019-01-06 15:09:07 UTC+0000
... 0x841eb658:WmiPrvSE.exe                          4032    552      5    114 2019-01-06 15:03:54 UTC+0000
... 0x857ce6d0:rundll32.exe                          2320    552      4     85 2019-01-06 15:02:52 UTC+0000
.. 0x858be710:svchost.exe                            3784    432     14    370 2019-01-06 15:04:51 UTC+0000
.. 0x85b2dd28:svchost.exe                            1588    432     10    147 2019-01-06 15:02:48 UTC+0000
```

How to find CPU of the Suspect's PC?

```
┌──(root㉿kali)-[/home/kali/volatility]
└─# python2 vol.py -f memdumpWin7.mem --profile=Win7SP1x86 printkey -K "DESCRIPTION\System\CentralProcessor\0"
Volatility Foundation Volatility Framework 2.6.1
Legend: (S) = Stable   (V) = Volatile

----------------------------
Registry: \REGISTRY\MACHINE\HARDWARE
Key name: 0 (S)
Last updated: 2019-01-06 18:02:40 UTC+0000

Subkeys:

Values:
REG_BINARY    Component Information : (S)
0x00000000  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
REG_SZ        Identifier      : (S) x86 Family 6 Model 158 Stepping 9
REG_FULL_RESOURCE_DESCRIPTOR Configuration Data : (S) ••••••••
REG_SZ        ProcessorNameString : (S) Intel(R) Core(TM) i7-7700 CPU @ 3.60GHz
REG_SZ        VendorIdentifier : (S) GenuineIntel
REG_DWORD     FeatureSet      : (S) 2697805823
REG_DWORD     ~MHz            : (S) 3600
REG_DWORD     Update Status   : (S) 2
REG_BINARY    Update Signature : (S)
0x00000000  00 00 00 00 00 00 00 00                         ........
REG_BINARY    Previous Update Signature : (S)
0x00000000  00 00 00 00 00 00 00 00                         ........
REG_DWORD     Platform ID     : (S) 2

┌──(root㉿kali)-[/home/kali/volatility]
└─#
```

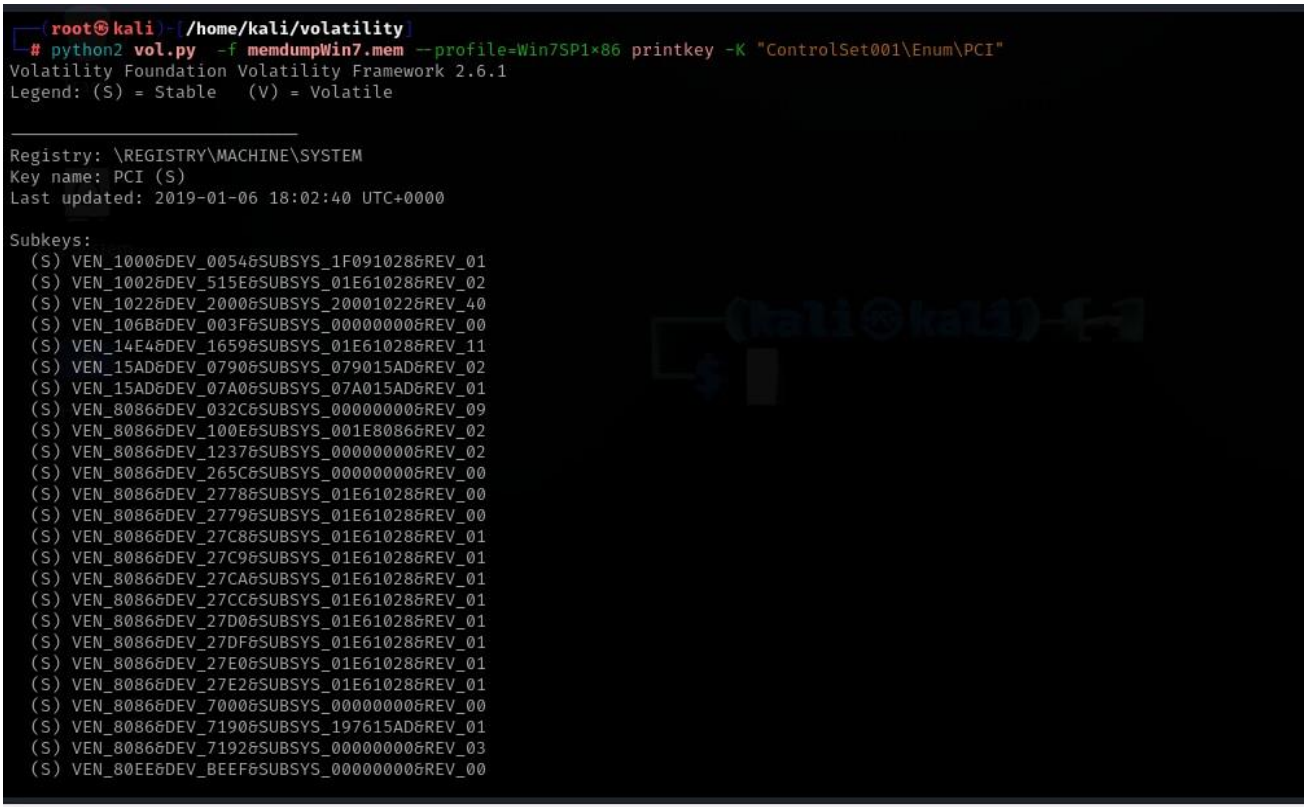
How to find other PC system information?

```
┌──(root㉿kali)-[/home/kali/volatility]
└─# python2 vol.py -f memdumpWin7.mem --profile=Win7SP1x86 printkey -K "DESCRIPTION\System"
Volatility Foundation Volatility Framework 2.6.1
Legend: (S) = Stable   (V) = Volatile

----------------------------
Registry: \REGISTRY\MACHINE\HARDWARE
Key name: System (S)
Last updated: 2019-01-06 18:02:44 UTC+0000

Subkeys:
  (S) CentralProcessor
  (S) FloatingPointProcessor
  (S) MultifunctionAdapter

Values:
REG_BINARY    Component Information : (S)
0x00000000  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
REG_SZ        Identifier      : (S) AT/AT COMPATIBLE
REG_FULL_RESOURCE_DESCRIPTOR Configuration Data : (S) ********
                                                      ••?•

REG_SZ        SystemBiosDate  : (S) 06/23/99
REG_MULTI_SZ  SystemBiosVersion : (S) ['VBOX   - 1', '', '']
REG_DWORD     BootArchitecture : (S) 3
REG_DWORD     PreferredProfile : (S) 0
REG_DWORD     Capabilities    : (S) 1345
REG_MULTI_SZ  VideoBiosVersion : (S) ['Oracle VM VirtualBox Version 6.0.0 VGA BIOS', 'Oracle VM VirtualBox Version 6.0.0 VGA BIOS', 'Oracle VM VirtualBox Version 6.0.
0', 'Oracle VM VirtualBox Version 6.0.0', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '']

┌──(root㉿kali)-[/home/kali/volatility]
└─#
```

Where the enumerated devices Info saved?

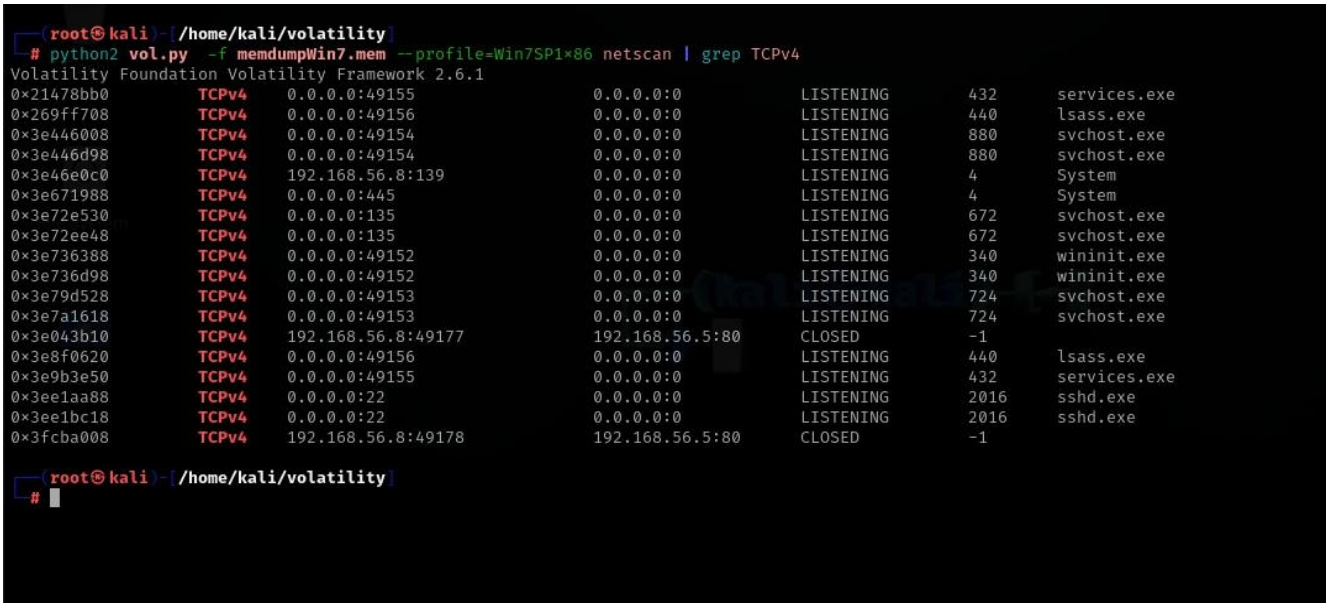
```
┌──(root㉿kali)-[/home/kali/volatility]
└─# python2 vol.py  -f memdumpWin7.mem --profile=Win7SP1x86 hivelist
Volatility Foundation Volatility Framework 2.6.1
Virtual    Physical   Name
---------- ---------- ----
0x87c10370 0x27fd5370 [no name]
0x87c1c008 0x27fa3008 \REGISTRY\MACHINE\SYSTEM
0x87c459c8 0x27d8e9c8 \REGISTRY\MACHINE\HARDWARE
0x889c0430 0x1285a430 \??\C:\Windows\ServiceProfiles\NetworkService\NTUSER.DAT
0x8b46d008 0x16d8c008 \??\C:\Windows\ServiceProfiles\LocalService\NTUSER.DAT
0x8b5d39b0 0x120049b0 \??\C:\Users\IEUser\ntuser.dat
0x97104008 0x1c45f008 \SystemRoot\System32\Config\SECURITY
0x9711d9c8 0x1e6a09c8 \SystemRoot\System32\Config\SOFTWARE
0x981f9008 0x1b446008 \SystemRoot\System32\Config\DEFAULT
0x98209518 0x1f3a7518 \SystemRoot\System32\Config\SAM
0x982d55c0 0x1e17c5c0 \REGISTRY\MACHINE\BCD00000000
0xa620b008 0x1997b008 \??\C:\Users\IEUser\AppData\Local\Microsoft\Windows\UsrClass.dat
0xa65269c8 0x17ad09c8 \??\C:\Users\sshd_server\ntuser.dat
0xa6539260 0x0f690260 \??\C:\Users\sshd_server\AppData\Local\Microsoft\Windows\UsrClass.dat
0xa7d2a9c8 0x01c469c8 \??\C:\System Volume Information\Syscache.hve

┌──(root㉿kali)-[/home/kali/volatility]
└─#
```

How to find devices connected to PCI?



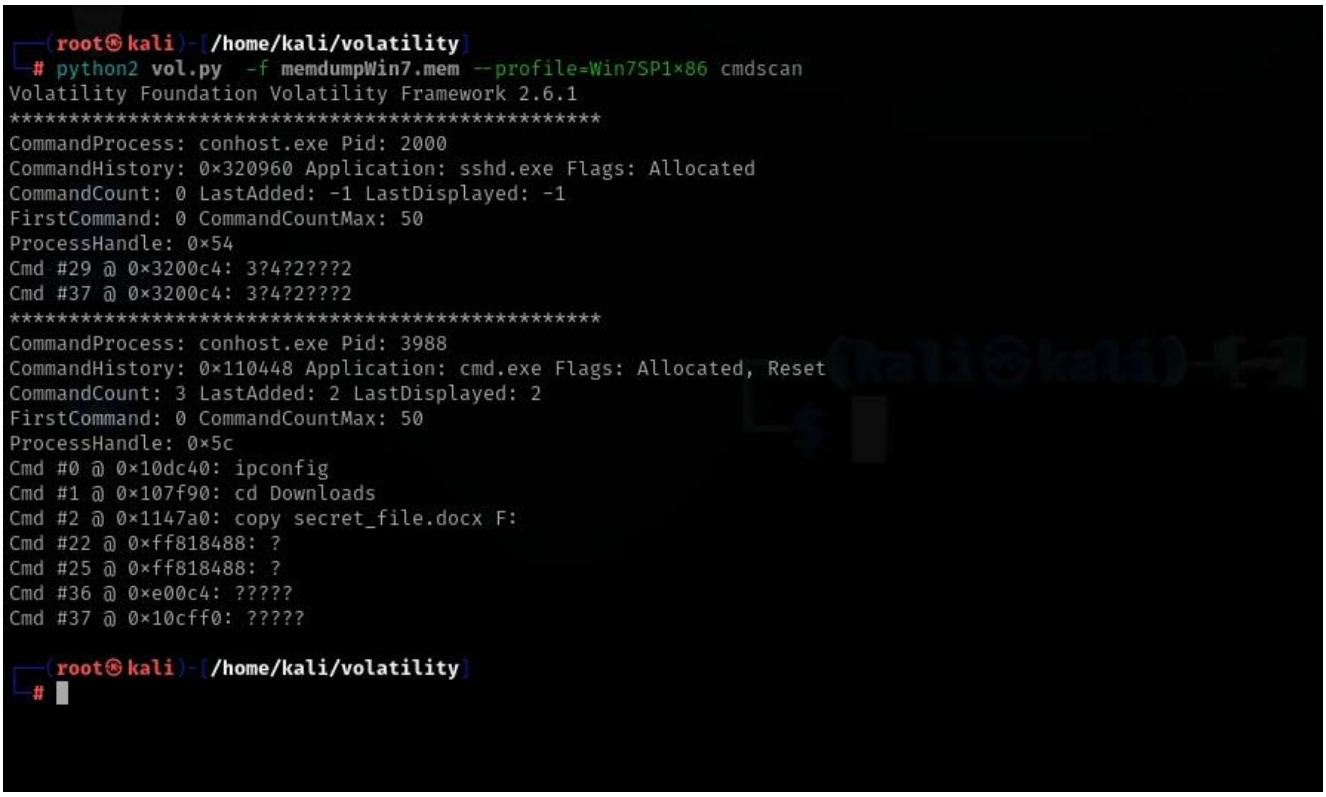
What is the name of suspect's device?

Are there suspicious IPs (processes) connected to the PC?



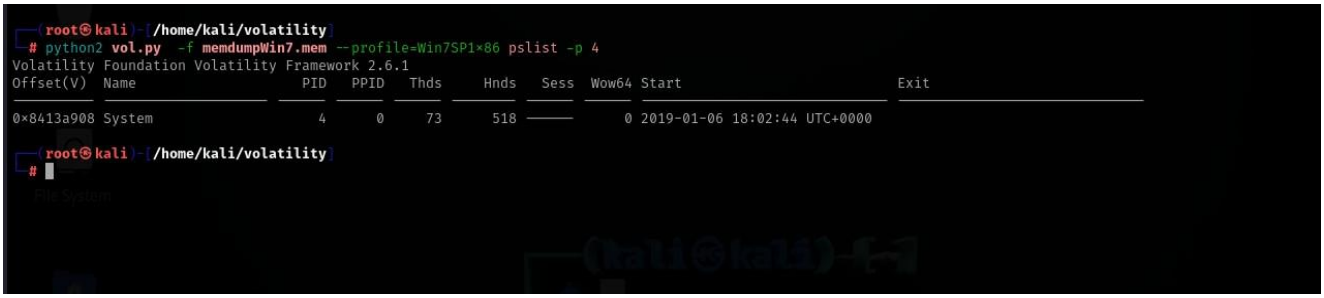
```
┌──(root㉿kali)-[/home/kali/volatility]
└─# python2 vol.py  -f memdumpWin7.mem --profile=Win7SP1x86 netscan | grep TCPv4
Volatility Foundation Volatility Framework 2.6.1
0x21478bb0         TCPv4    0.0.0.0:49155                  0.0.0.0:0              LISTENING        432      services.exe
0x269ff708         TCPv4    0.0.0.0:49156                  0.0.0.0:0              LISTENING        440      lsass.exe
0x3e446008         TCPv4    0.0.0.0:49154                  0.0.0.0:0              LISTENING        880      svchost.exe
0x3e446d98         TCPv4    0.0.0.0:49154                  0.0.0.0:0              LISTENING        880      svchost.exe
0x3e46e0c0         TCPv4    192.168.56.8:139               0.0.0.0:0              LISTENING        4        System
0x3e671988         TCPv4    0.0.0.0:445                    0.0.0.0:0              LISTENING        4        System
0x3e72e530         TCPv4    0.0.0.0:135                    0.0.0.0:0              LISTENING        672      svchost.exe
0x3e72ee48         TCPv4    0.0.0.0:135                    0.0.0.0:0              LISTENING        672      svchost.exe
0x3e736388         TCPv4    0.0.0.0:49152                  0.0.0.0:0              LISTENING        340      wininit.exe
0x3e736d98         TCPv4    0.0.0.0:49152                  0.0.0.0:0              LISTENING        340      wininit.exe
0x3e79d528         TCPv4    0.0.0.0:49153                  0.0.0.0:0              LISTENING        724      svchost.exe
0x3e7a1618         TCPv4    0.0.0.0:49153                  0.0.0.0:0              LISTENING        724      svchost.exe
0x3e043b10         TCPv4    192.168.56.8:49177             192.168.56.5:80        CLOSED           -1
0x3e8f0620         TCPv4    0.0.0.0:49156                  0.0.0.0:0              LISTENING        440      lsass.exe
0x3e9b3e50         TCPv4    0.0.0.0:49155                  0.0.0.0:0              LISTENING        432      services.exe
0x3ee1aa88         TCPv4    0.0.0.0:22                     0.0.0.0:0              LISTENING        2016     sshd.exe
0x3ee1bc18         TCPv4    0.0.0.0:22                     0.0.0.0:0              LISTENING        2016     sshd.exe
0x3fcba008         TCPv4    192.168.56.8:49178             192.168.56.5:80        CLOSED           -1

┌──(root㉿kali)-[/home/kali/volatility]
└─#
```

Did the suspect use commands to copy files?

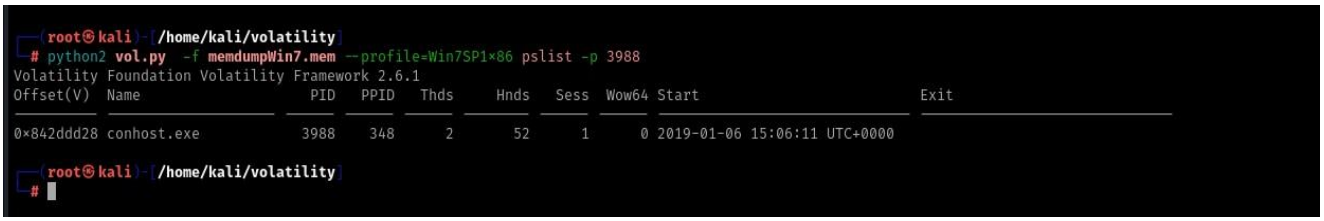


```
┌──(root㉿kali)-[/home/kali/volatility]
└─# python2 vol.py  -f memdumpWin7.mem --profile=Win7SP1x86 cmdscan
Volatility Foundation Volatility Framework 2.6.1
**************************************************
CommandProcess: conhost.exe Pid: 2000
CommandHistory: 0x320960 Application: sshd.exe Flags: Allocated
CommandCount: 0 LastAdded: -1 LastDisplayed: -1
FirstCommand: 0 CommandCountMax: 50
ProcessHandle: 0x54
Cmd #29 @ 0x3200c4: 3?4?2???2
Cmd #37 @ 0x3200c4: 3?4?2???2
**************************************************
CommandProcess: conhost.exe Pid: 3988
CommandHistory: 0x110448 Application: cmd.exe Flags: Allocated, Reset
CommandCount: 3 LastAdded: 2 LastDisplayed: 2
FirstCommand: 0 CommandCountMax: 50
ProcessHandle: 0x5c
Cmd #0 @ 0x10dc40: ipconfig
Cmd #1 @ 0x107f90: cd Downloads
Cmd #2 @ 0x1147a0: copy secret_file.docx F:
Cmd #22 @ 0xff818488: ?
Cmd #25 @ 0xff818488: ?
Cmd #36 @ 0xe00c4: ?????
Cmd #37 @ 0x10cff0: ?????

┌──(root㉿kali)-[/home/kali/volatility]
└─#
```

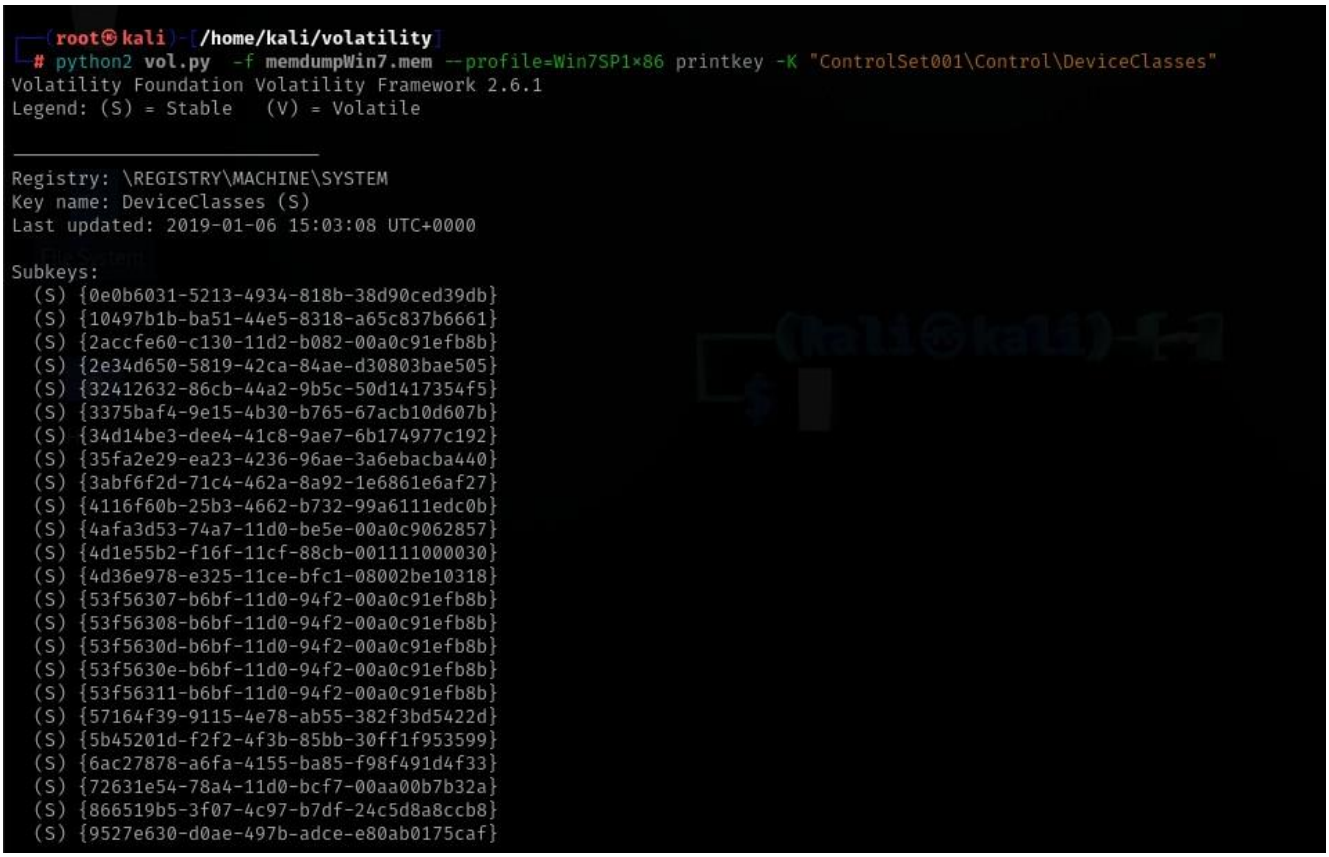
Which program that was connected to this suspicious IP address?



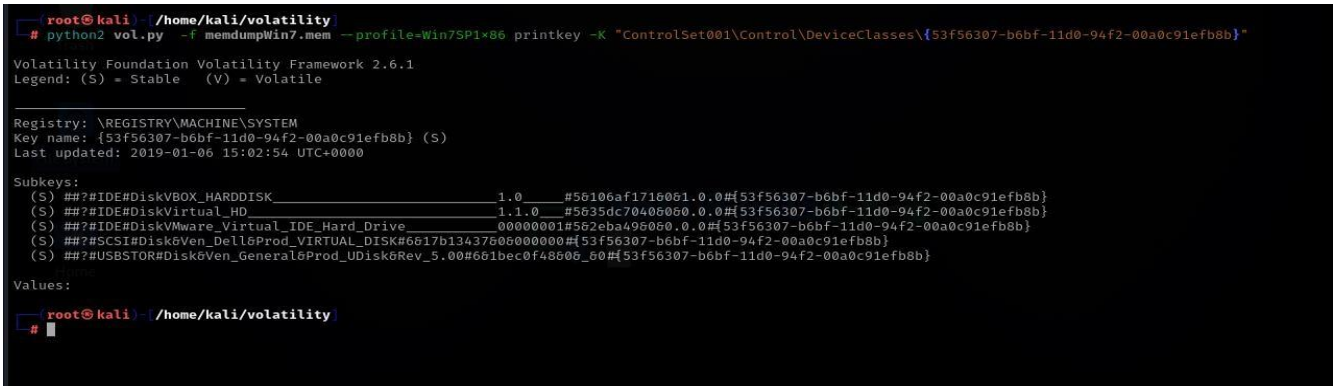
When are these commands executed?



List all device interfaces:

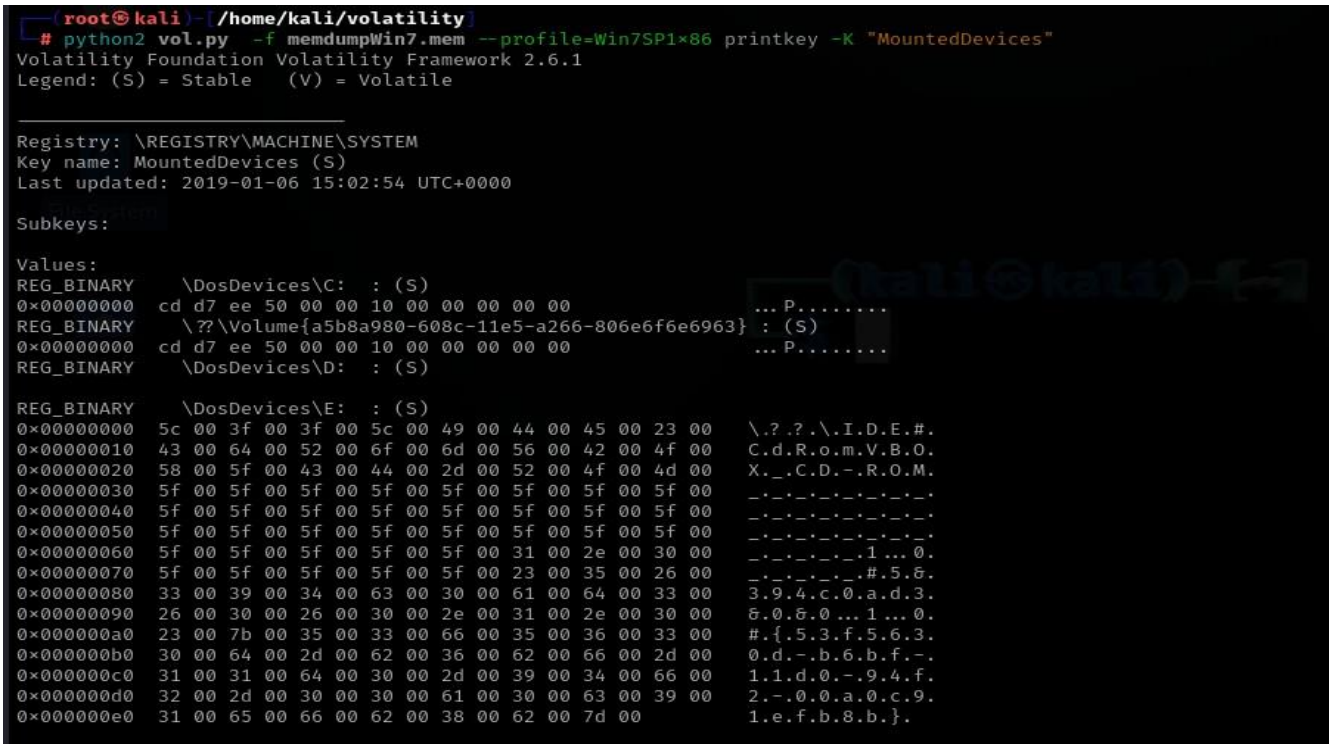


List Disk Interfaces

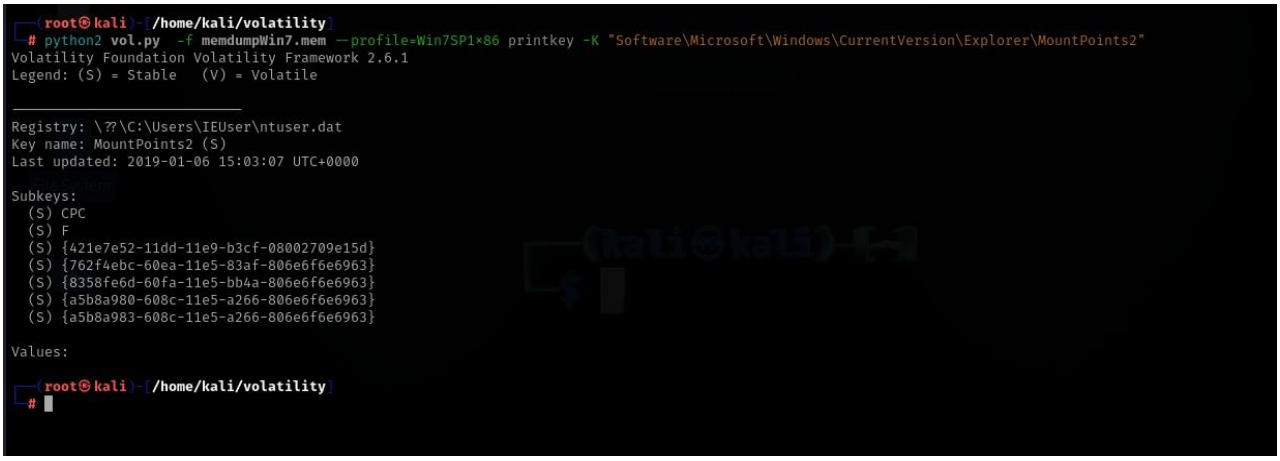


Access DeviceClass\Disk Interface\USB Interface

Which drive letter does the USB was assigned to?

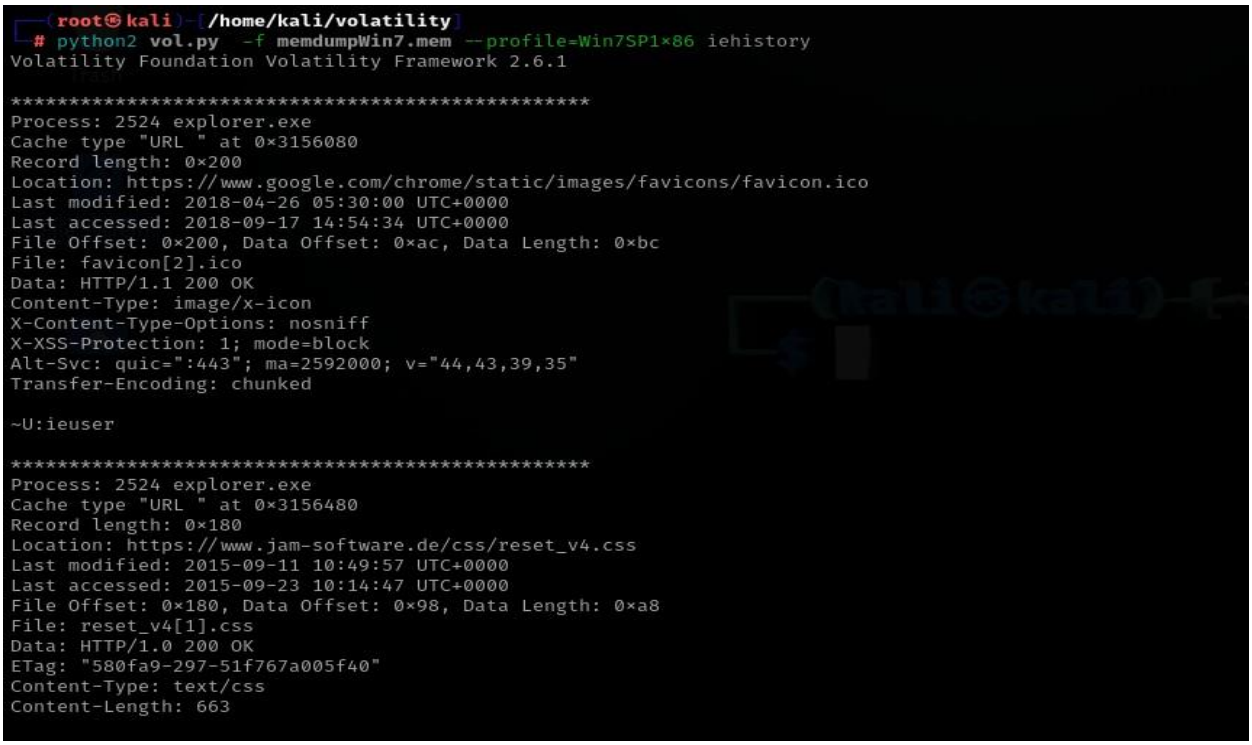


Who did mount F volume to PC?

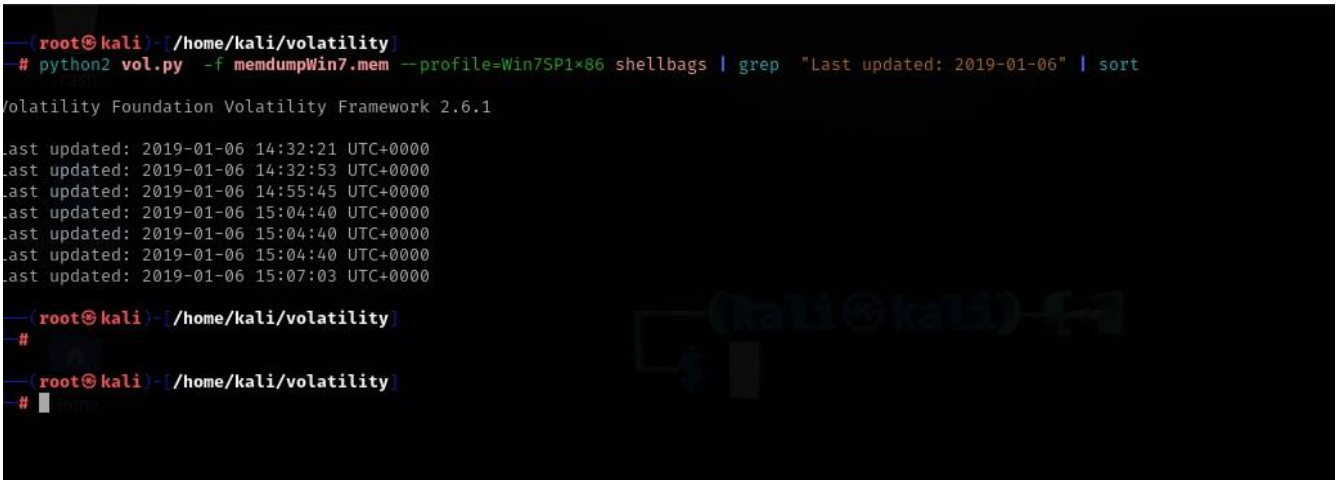


When the USB Last Attached to PC?

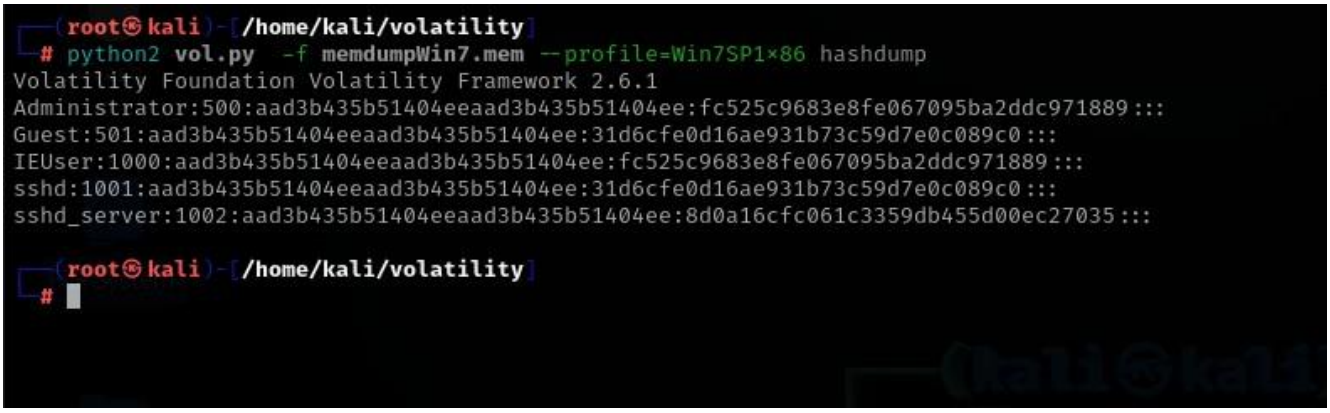
Did the suspect use Internet Explore?



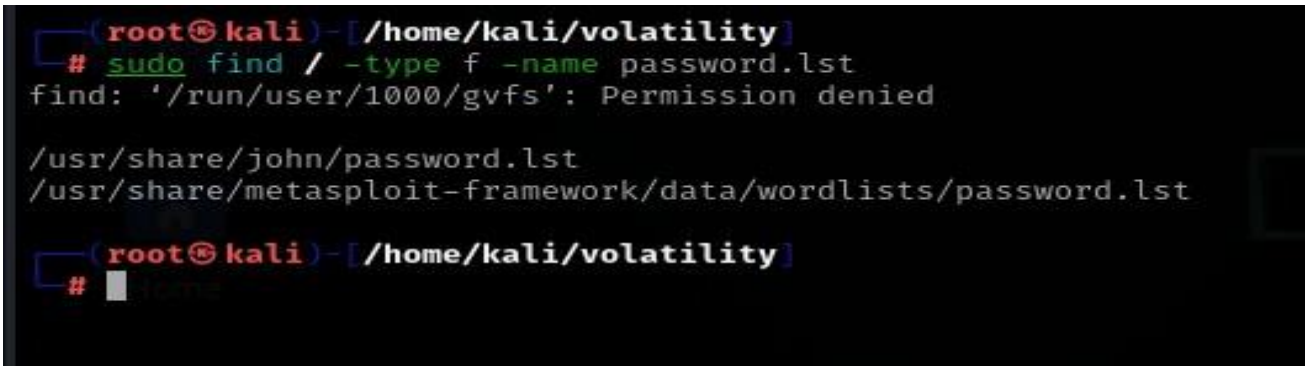
Are there any folder access activities on 2019-01-06?



Dump password hashes from memory:

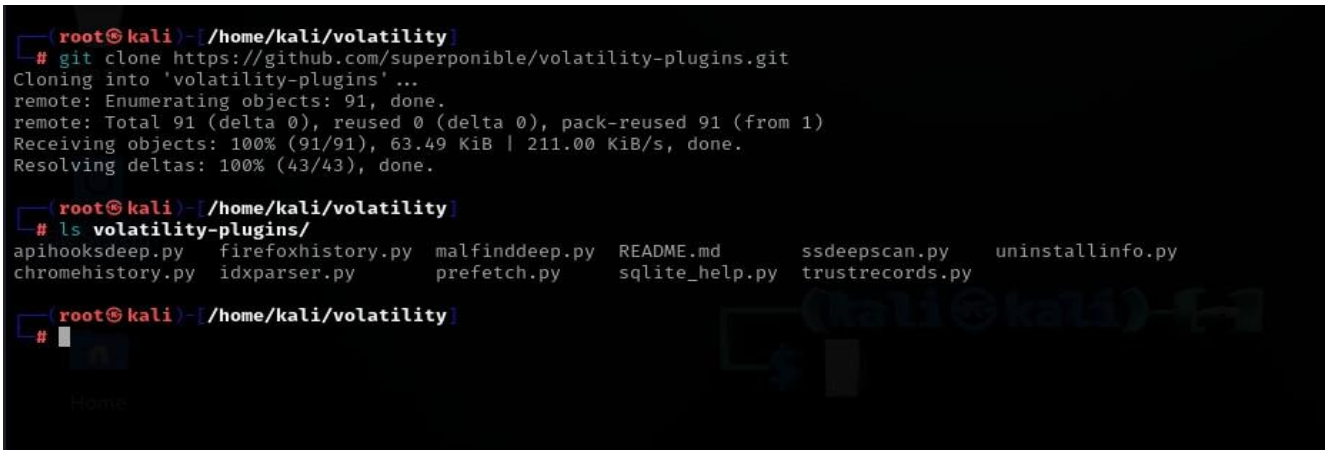


Find the location of John's default password.lst:



Search passwords contain the string “pass” ignore case

How to download plugins



How to find chrome history?