## Unit 2: MEMORY FORENSICS

## 1. MEMORY FORENSICS

Computer memory contains many important data that are relevant to forensic investigation. In a volatile memory, data are allocated and de-allocated dynamically and the data are not structured the way they are found in file-systems. Mostly it becomes impossible for the investigator to predict where exactly the data are stored in a volatile memory. Attackers may hide data in the memory by means of memory injection or by means of malicious codes like viruses, and worms that reside only in the memory and not on hard disk so as to avoid anti-virus software to detect the malicious code. Analyzing a hard disk by the traditional forensic method does not reveal malicious code if the attacker has placed it in the memory. Memory forensics is one method of investigation which can help forensic investigators to detect such files.

### 1.1. SEARCHING MEMORY FOR EVIDENCE

Attackers know about the vulnerability in the forensic examiners' approach to seizing computers – namely how data in the volatile memory RAM will be lost upon system shutdown. There are many hacker tools like DLL injections, hooks and other methods that execute code only in the memory without accessing the hard drive or other non-volatile storage media. Since forensic examiners will interact with system and system files accessed on the hard disk while carrying out forensic investigation which may alter the file time stamps, it is recommended not to interact with live system to preserve the accuracy of time stamps of system files. But in many cases, investigators might require to violate this rule for capturing the live data from the memory.

Shadow Walker is a rootkit that runs in memory and leaves no trace in the system. It hides its presence by creating fake views of system memory. It is a detection tool that pretends that it is accurately reading memory. This tool does not alter the flow of program execution and data structures in memory. All programs receive an inaccurate mapping of memory leading to memory cloaking. The aim of memory cloaking is to hide the rootkit's own code or some other related modules. Normal utilities will fail to identify the rootkit. Memory forensics is one method to identify the rootkit.

## 1.2. LIVE RESPONSE

Live response is the method used to interact with live systems to collect information on changes that has occurred possibly due to passage of time, running processes, data being saved and deleted, as network connections timed-out, etc. Running a program causes information to be loaded into physical memory that may also be used by other programs at the same time. Changes that occur to a system as the system itself apparently sits idle are referred to as evidence dynamics.

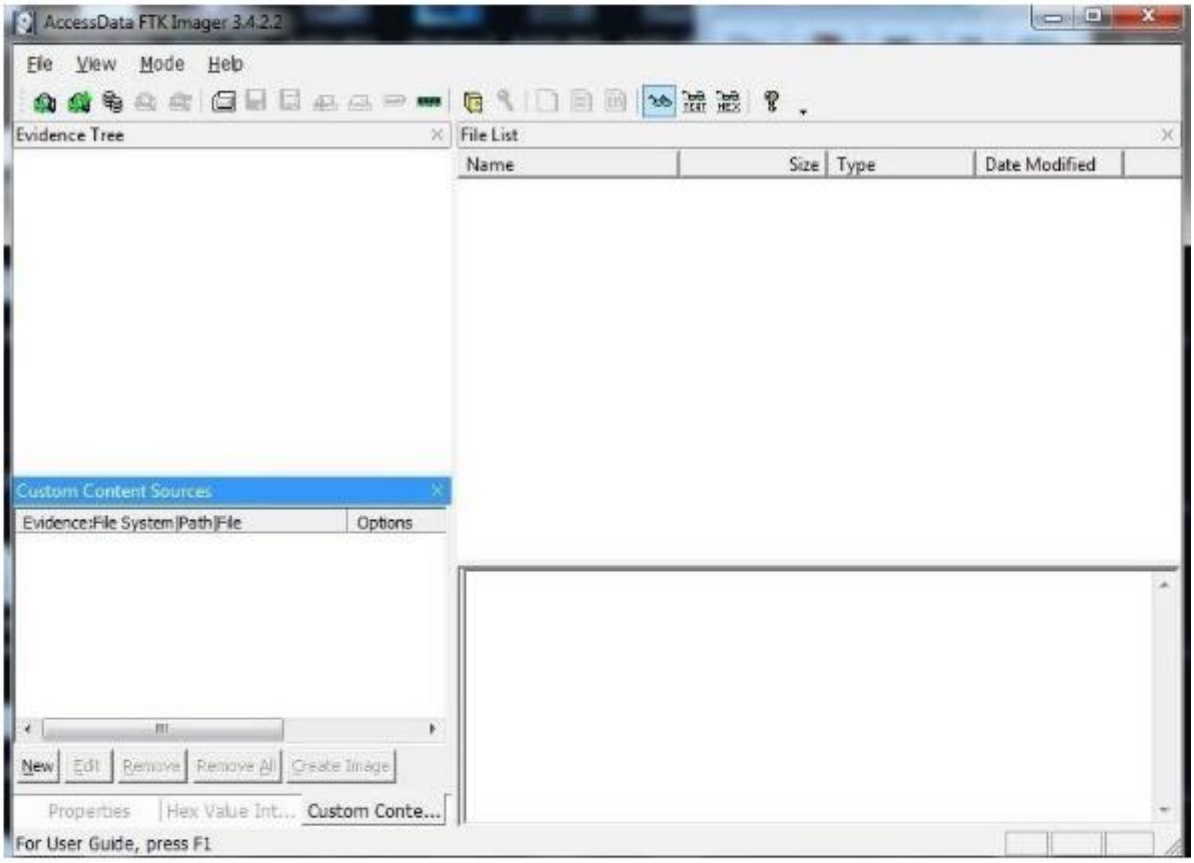
It is a response to the immediate threat and the active threat. Many times the details of the live threats are unknown, so the first step is to identify and quantify these threats. Using live response memory analysis techniques, investigators can retrieve process listing showing what processes are running and then identify the suspicious ones. Once we have identified the suspicious process, a sample of the code is pulled from running memory and then the analysis can be performed. Pulling codes from the volatile memory requires no special processing since it is unencrypted and un-packed.

## 1.3. DATA ACQUISITION FROM MEMORY

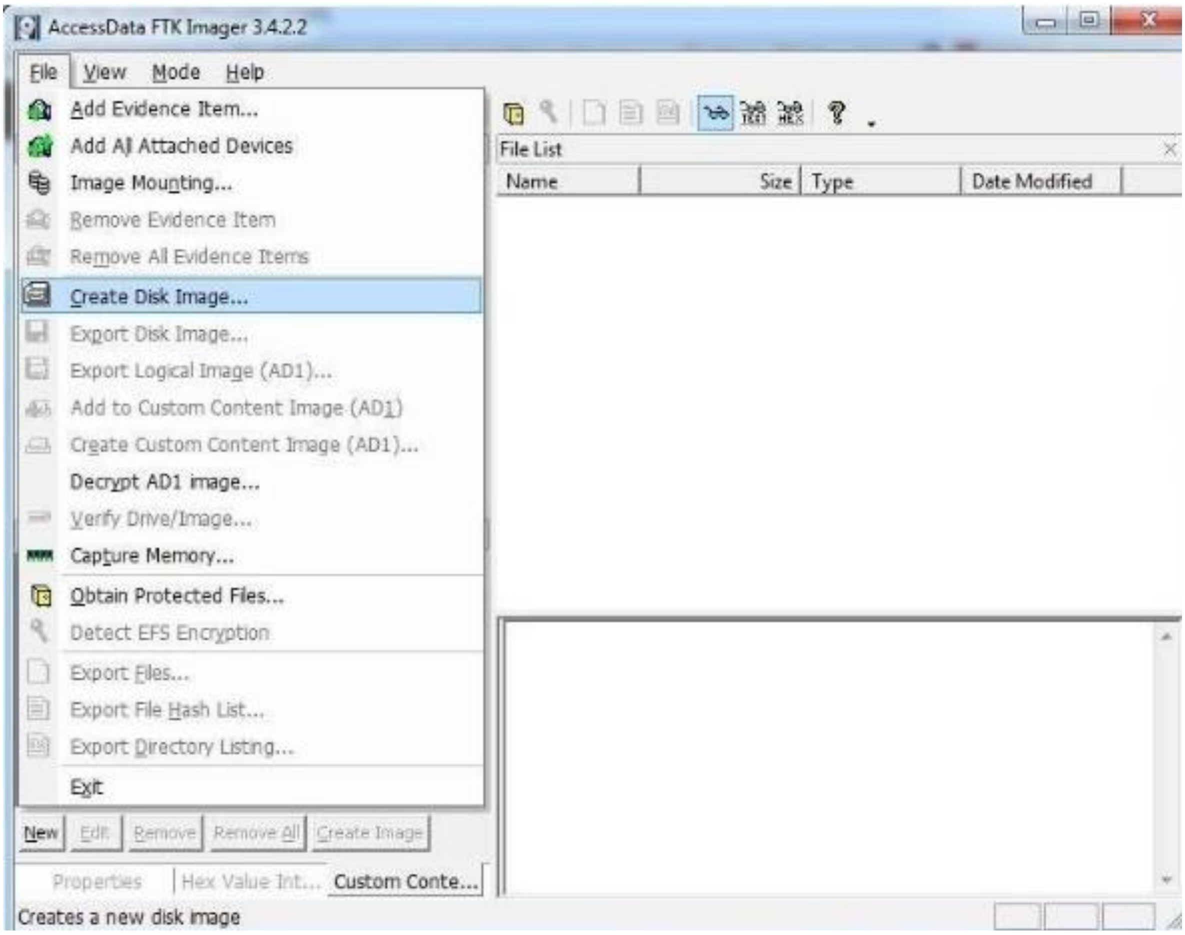
Forensic investigators should adhere to the practice to collect data from volatile memory RAM before effecting any change that may arise while performing an interaction with the system during investigation. Software tools that are used for investigation will cause changes in the volatile memory. When the software will be run, it will be loaded in the volatile memory therein causing changes. There are many tools for capturing data from memory. One of the commonly used tools is FTK (Forensic Tool Kit).

Steps:

1. Download FTK and install FTK imager.

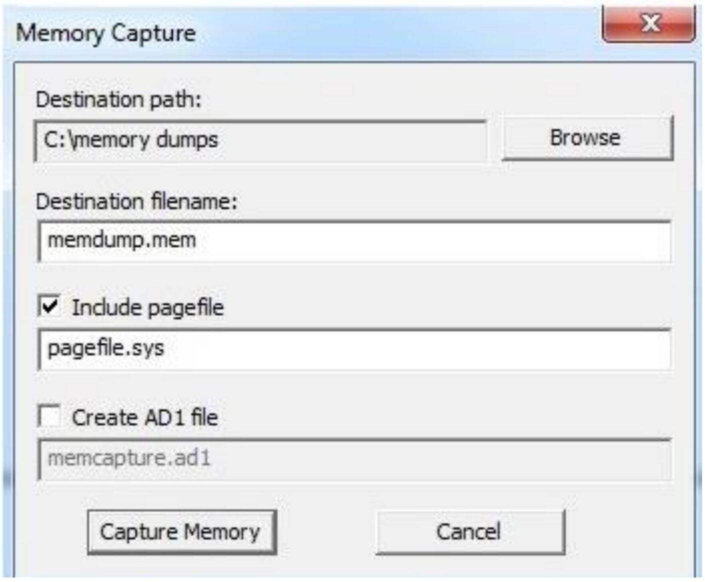


**2.** Click on **File** menu and then select **Capture Memory.**



Creates a new disk image

**3.** It will open the following window below.



Destination path: C:\memory dumps

Destination filename: memdump.mem

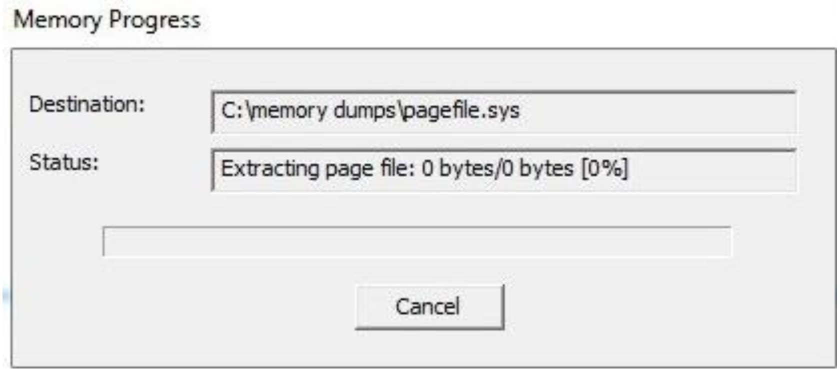
☑ Include pagefile

pagefile.sys

☐ Create AD1 file

memcapture.ad1

The following window shows the progress of your capture.



Destination: C:\memory dumps\pagefile.sys

Status: Extracting page file: 0 bytes/0 bytes [0%]

While performing memory forensics, there are several useful data that have to be collected to get an accurate result. These data include: system time, network information, network connections, network status, memory processes, logged-on users, opened files, command history, share history, mapped drives, clipboard content.

**Table 2.1: Useful Data for Memory Forensics**

| Data | Details |
|---|---|
| System Time | System time is the first data collected so as to acquire an accurate timeline of events that have occurred on system. It notes the actual time at which events occurred and these are recorded in the log files. It is important to combine time stamps from more than one source. |
| System Information | time zone, installed software, general system information, operating systems version, uptime, file system information |
| Logged-on users | **psloggedon.exe** is one of the best tools that shows the name of the user logged on locally as well as users who are logged on remotely. |
| Net Sessions | Net sessions command is native to Windows systems and is used to see username used to access the system through a remote login session, IP address and type of client from accessing system. |
| LogonSessions | **Logonsessions.exe** is a tool that lists all the active logon sessions on a system. It will not show users that are logged on via a backdoor. |

<table>
<tr>
<td>Open Files</td>
<td>

**psfile.exe** and **openfiles.exe** are net file commands that are used to see files open on a system through remote connection.

***PsFile*** command shows a list of files on a system that are opened remotely, and it also allows you to close the opened files either by name or by a file identifier.

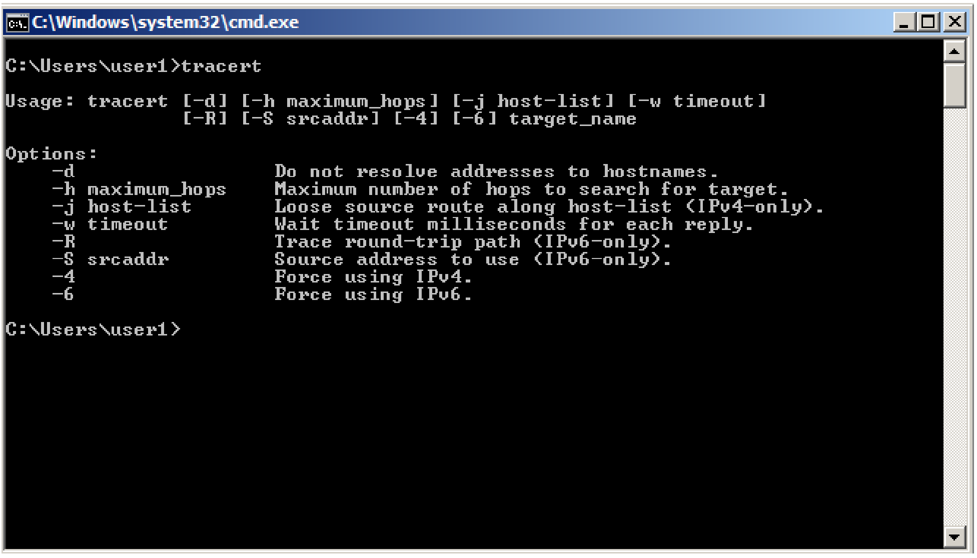
</td>
</tr>
<tr>
<td>Network Connection</td>
<td>

***Traceroute*** – *tracert* command is a Command prompt command that shows details about the path that a data packet takes from the computer or device you're on to whatever destination you specify.

***ARP*** command lists computers that are/have recently connected to the system.

***Ipconfig*** command displays information about network state and the MAC address of the computer.

***tracert***:

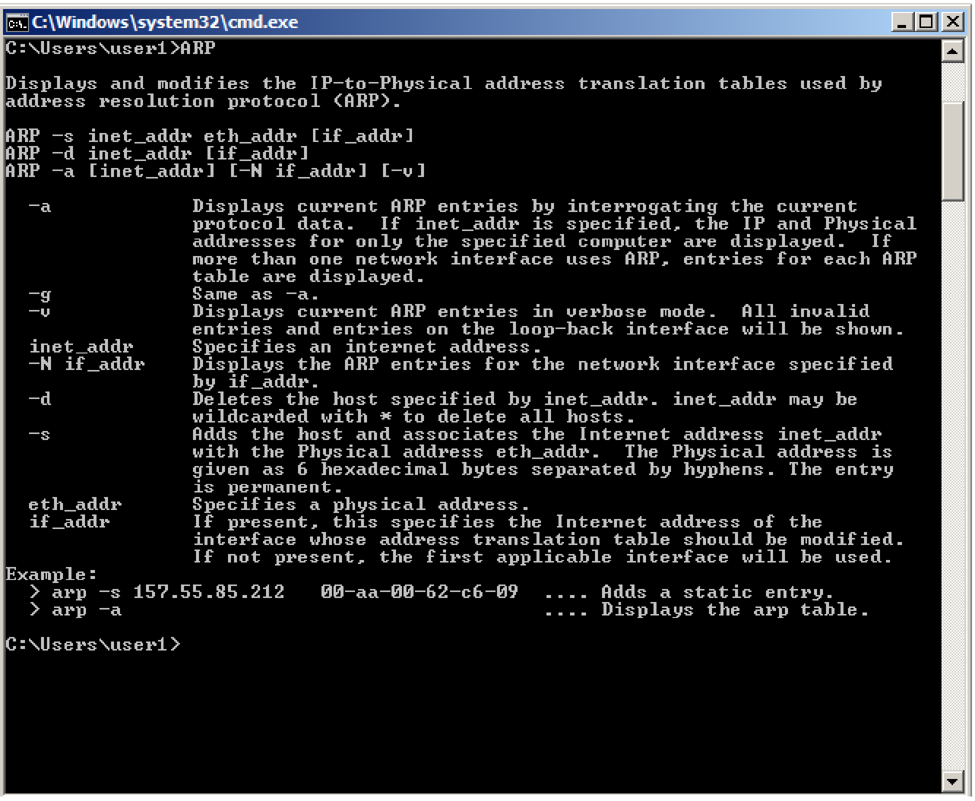


C:\Windows\system32\cmd.exe

```
C:\Users\user1>tracert

Usage: tracert [-d] [-h maximum_hops] [-j host-list] [-w timeout]
               [-R] [-S srcaddr] [-4] [-6] target_name

Options:
    -d                 Do not resolve addresses to hostnames.
    -h maximum_hops    Maximum number of hops to search for target.
    -j host-list       Loose source route along host-list (IPv4-only).
    -w timeout         Wait timeout milliseconds for each reply.
    -R                 Trace round-trip path (IPv6-only).
    -S srcaddr         Source address to use (IPv6-only).
    -4                 Force using IPv4.
    -6                 Force using IPv6.

C:\Users\user1>
```

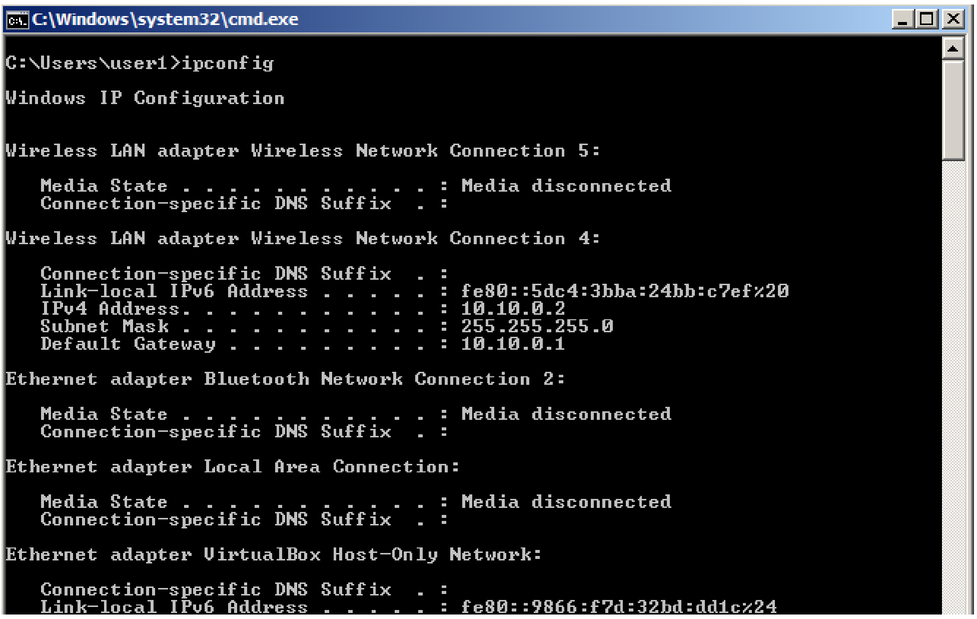
***ARP***:

</td>
</tr>
</table>



```
C:\Users\user1>ARP

Displays and modifies the IP-to-Physical address translation tables used by
address resolution protocol (ARP).

ARP -s inet_addr eth_addr [if_addr]
ARP -d inet_addr [if_addr]
ARP -a [inet_addr] [-N if_addr] [-v]

  -a            Displays current ARP entries by interrogating the current
                protocol data.  If inet_addr is specified, the IP and Physical
                addresses for only the specified computer are displayed.  If
                more than one network interface uses ARP, entries for each ARP
                table are displayed.
  -g            Same as -a.
  -v            Displays current ARP entries in verbose mode.  All invalid
                entries and entries on the loop-back interface will be shown.
  inet_addr     Specifies an internet address.
  -N if_addr    Displays the ARP entries for the network interface specified
                by if_addr.
  -d            Deletes the host specified by inet_addr. inet_addr may be
                wildcarded with * to delete all hosts.
  -s            Adds the host and associates the Internet address inet_addr
                with the Physical address eth_addr.  The Physical address is
                given as 6 hexadecimal bytes separated by hyphens. The entry
                is permanent.
  eth_addr      Specifies a physical address.
  if_addr       If present, this specifies the Internet address of the
                interface whose address translation table should be modified.
                If not present, the first applicable interface will be used.
Example:
  > arp -s 157.55.85.212   00-aa-00-62-c6-09  .... Adds a static entry.
  > arp -a                                    .... Displays the arp table.

C:\Users\user1>
```

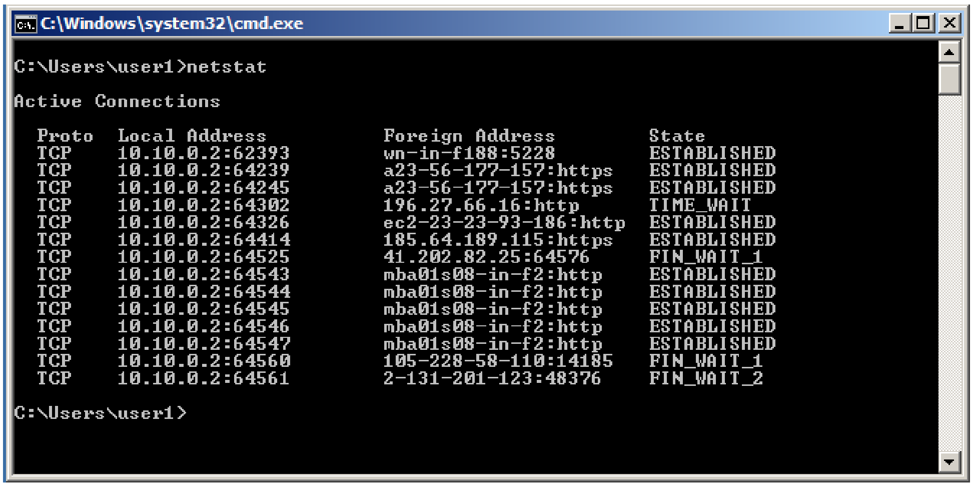
***ipconfig***:



```
C:\Users\user1>ipconfig

Windows IP Configuration


Wireless LAN adapter Wireless Network Connection 5:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Wireless LAN adapter Wireless Network Connection 4:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::5dc4:3bba:24bb:c7ef%20
   IPv4 Address. . . . . . . . . . . : 10.10.0.2
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 10.10.0.1

Ethernet adapter Bluetooth Network Connection 2:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Ethernet adapter Local Area Connection:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Ethernet adapter VirtualBox Host-Only Network:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::9866:f7d:32bd:dd1c%24
```

| | |
|---|---|
| netstat | ***Netstat*** displays the active computer connections. Investigators can obtain the list of protocols running and open ports. |

<table>
<tr>
<td></td>
<td>

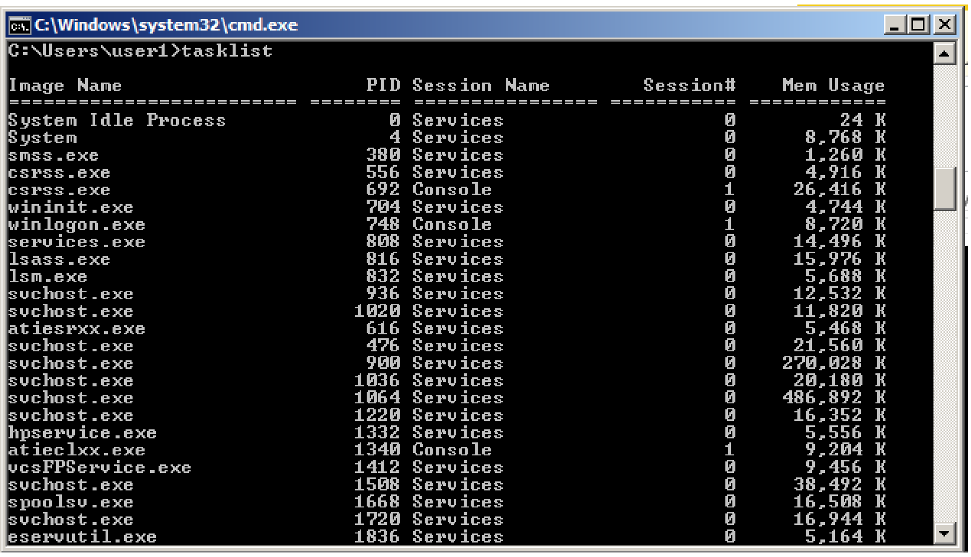


```
C:\Users\user1>netstat

Active Connections

  Proto  Local Address          Foreign Address        State
  TCP    10.10.0.2:62393        wn-in-f188:5228        ESTABLISHED
  TCP    10.10.0.2:64239        a23-56-177-157:https   ESTABLISHED
  TCP    10.10.0.2:64245        a23-56-177-157:https   ESTABLISHED
  TCP    10.10.0.2:64302        196.27.66.16:http      TIME_WAIT
  TCP    10.10.0.2:64326        ec2-23-23-93-186:http  ESTABLISHED
  TCP    10.10.0.2:64414        185.64.189.115:https   ESTABLISHED
  TCP    10.10.0.2:64525        41.202.82.25:64576     FIN_WAIT_1
  TCP    10.10.0.2:64543        mba01s08-in-f2:http    ESTABLISHED
  TCP    10.10.0.2:64544        mba01s08-in-f2:http    ESTABLISHED
  TCP    10.10.0.2:64545        mba01s08-in-f2:http    ESTABLISHED
  TCP    10.10.0.2:64546        mba01s08-in-f2:http    ESTABLISHED
  TCP    10.10.0.2:64547        mba01s08-in-f2:http    ESTABLISHED
  TCP    10.10.0.2:64560        105-228-58-110:14185   FIN_WAIT_1
  TCP    10.10.0.2:64561        2-131-201-123:48376    FIN_WAIT_2

C:\Users\user1>
```

</td>
</tr>
<tr>
<td>Process Information</td>
<td>

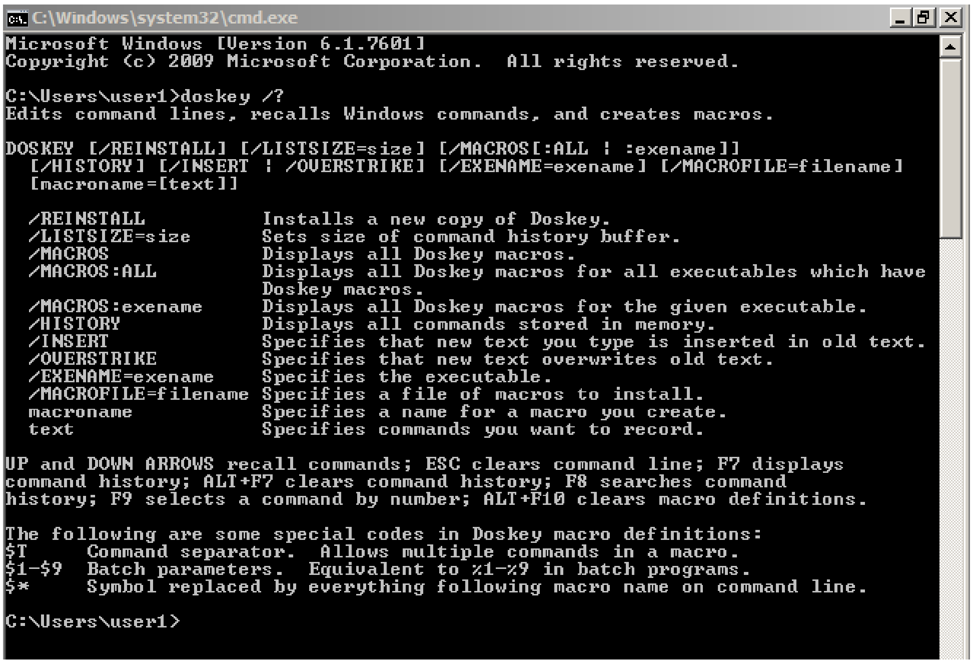
There are several different types of processes running in the volatile memory of a computer. All currently running processes may be recovered from data structures that house them. Terminated processes may still reside in memory if the computer has not been rebooted since and the memory space is not  yet reallocated.

***Tlist.exe*** or ***Tasklist.exe*** command displays information about running processes.



```
C:\Users\user1>tasklist

Image Name                     PID Session Name        Session#    Mem Usage
========================= ======== ================ =========== ============
System Idle Process              0 Services                   0         24 K
System                           4 Services                   0      8,768 K
smss.exe                       380 Services                   0      1,260 K
csrss.exe                      556 Services                   0      4,916 K
csrss.exe                      692 Console                    1     26,416 K
wininit.exe                    704 Services                   0      4,744 K
winlogon.exe                   748 Console                    1      8,720 K
services.exe                   808 Services                   0     14,496 K
lsass.exe                      816 Services                   0     15,976 K
lsm.exe                        832 Services                   0      5,688 K
svchost.exe                    936 Services                   0     12,532 K
svchost.exe                   1020 Services                   0     11,820 K
atiesrxx.exe                   616 Services                   0      5,468 K
svchost.exe                    476 Services                   0     21,560 K
svchost.exe                    900 Services                   0    270,028 K
svchost.exe                   1036 Services                   0     20,180 K
svchost.exe                   1064 Services                   0    486,892 K
svchost.exe                   1220 Services                   0     16,352 K
hpservice.exe                 1332 Services                   0      5,556 K
atieclxx.exe                  1340 Console                    1      9,204 K
vcsFPService.exe              1412 Services                   0      9,456 K
svchost.exe                   1508 Services                   0     38,492 K
spoolsv.exe                   1668 Services                   0     16,508 K
svchost.exe                   1720 Services                   0     16,944 K
eservutil.exe                 1836 Services                   0      5,164 K
```

</td>
</tr>
</table>

| | |
|---|---|
| Capturing processes | ***Pslist.exe*** command provides basic information about running processes on a computer system, including the amount of time each process has been running.<br>***Listdlls.exe*** command shows the modules or DLLs, a process is using. It will show the full path to the image of the loaded module as well as whether the version of the DLL loaded in memory is different from that of the on-disk image. |
| Clipboard content | Data (username, password, text contents, etc) copied or cut stays in clipboard until it is replaced by a different content.<br>***Pclip.exe*** command retrieves the contents of the Clipboard. |
| Command history | ***doskey /history*** command shows the history of the previously typed commands at the command prompt.<br> |

```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\user1>doskey /?
Edits command lines, recalls Windows commands, and creates macros.

DOSKEY [/REINSTALL] [/LISTSIZE=size] [/MACROS[:ALL | :exename]]
  [/HISTORY] [/INSERT | /OVERSTRIKE] [/EXENAME=exename] [/MACROFILE=filename]
  [macroname=[text]]

  /REINSTALL          Installs a new copy of Doskey.
  /LISTSIZE=size      Sets size of command history buffer.
  /MACROS             Displays all Doskey macros.
  /MACROS:ALL         Displays all Doskey macros for all executables which have
                      Doskey macros.
  /MACROS:exename     Displays all Doskey macros for the given executable.
  /HISTORY            Displays all commands stored in memory.
  /INSERT             Specifies that new text you type is inserted in old text.
  /OVERSTRIKE         Specifies that new text overwrites old text.
  /EXENAME=exename    Specifies the executable.
  /MACROFILE=filename Specifies a file of macros to install.
  macroname           Specifies a name for a macro you create.
  text                Specifies commands you want to record.

UP and DOWN ARROWS recall commands; ESC clears command line; F7 displays
command history; ALT+F7 clears command history; F8 searches command
history; F9 selects a command by number; ALT+F10 clears macro definitions.

The following are some special codes in Doskey macro definitions:
$T     Command separator.  Allows multiple commands in a macro.
$1-$9  Batch parameters.  Equivalent to %1-%9 in batch programs.
$*     Symbol replaced by everything following macro name on command line.

C:\Users\user1>
```

### 3.1.1. Data Acquisition Methods

Passwords and Cryptographic Keys can recover user passwords and keys that are used to decrypt files access and user accounts. These passwords and keys are normally stored on disk with higher protection. Unencrypted content may recover encrypted files without password or cryptographic key when the files are opened as an opened file is unencrypted and loaded into the volatile memory.

There are 2 methods of data acquisition: Hardware- based acquisition and Software- based acquisition.
In a forensics perspective, it is better to use hardware- based acquisition since it is more reliable and difficult for an attacker to corrupt. Hardware-based acquisition suspends the processor and uses direct memory access to obtain a copy of memory. It is more reliable as even if the operating system and software are compromised or corrupted by an attacker, we still get an accurate image of the volatile memory. Hardware based acquisition are however costly. Whereas Software-based acquisition is more popular since it is more cost-effective and easily available.

#### 3.1.1.1.Hardware-Based Acquisition

For a hardware-based acquisition, the PCI card is plugged in the computer system and a copy of the memory is transferred to an external storage device. The card is installed into a PCI bus slot before an incident occur and is disabled until a physical switch on the back of the system is pressed. The card cannot easily be detected by an attacker and the acquisition procedure does not rely on untrusted resources. A card normally does not respond to bus queries from host system. When the device is enabled, it becomes a visible connection to the PCI bus. It is typically installed into a client’s critical servers. After an attack is suspected on the system, the administrator will press the physical switch therein the system state is saved by retrieving and storing the current memory image and processor registers. The card is then ejected from the system and sent to the forensic lab for detailed analysis.

#### 3.1.1.2.Software- Based Acquisition

Software-based acquisition is often carried out using a trusted toolkit by forensic examiners. Special forensics CDs/pen drives are used to run the software commands. This method can also collect volatile memory content using tools built in the operating system. Drawbacks of software- based acquisition are that it may alter contents of memory, and potentially overwrite data relevant to an investigation.