ATTACK DEFENSE

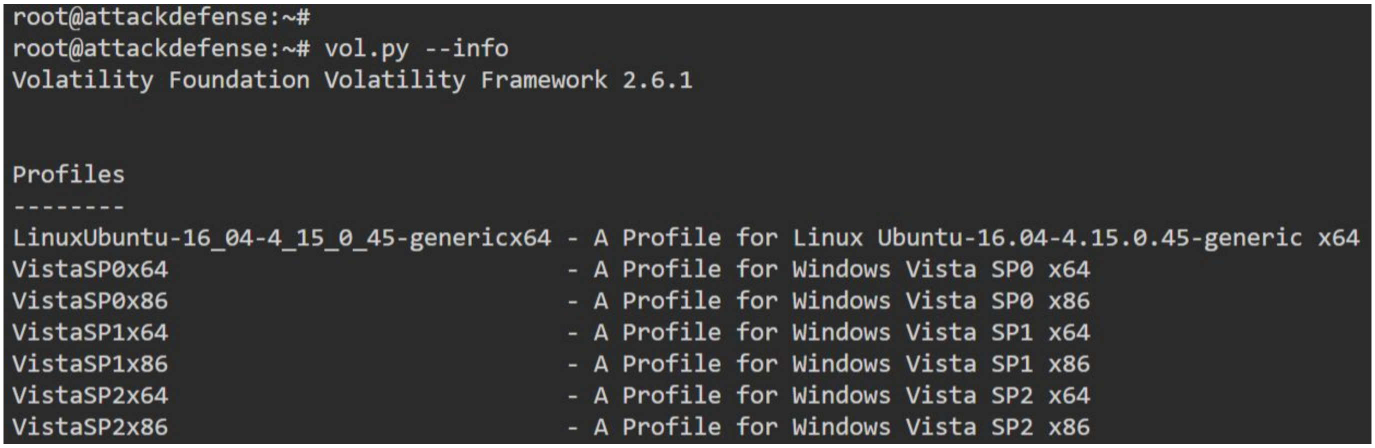
by PentesterAcademy

Name	Volatility: Basics
URL	https://attackdefense.com/challengedetails?cid=1099
Type	Forensics: Memory Forensics

Important Note: This document illustrates all the important steps required to complete this lab. This is by no means a comprehensive step-by-step solution for this exercise. This is only provided as a reference to various commands needed to complete this exercise and for your further research on this topic. Also, note that the IP addresses and domain names might be different in your lab.

Q1. Which command is used to list all profiles supported by Volatility?

Answer: vol.py --info



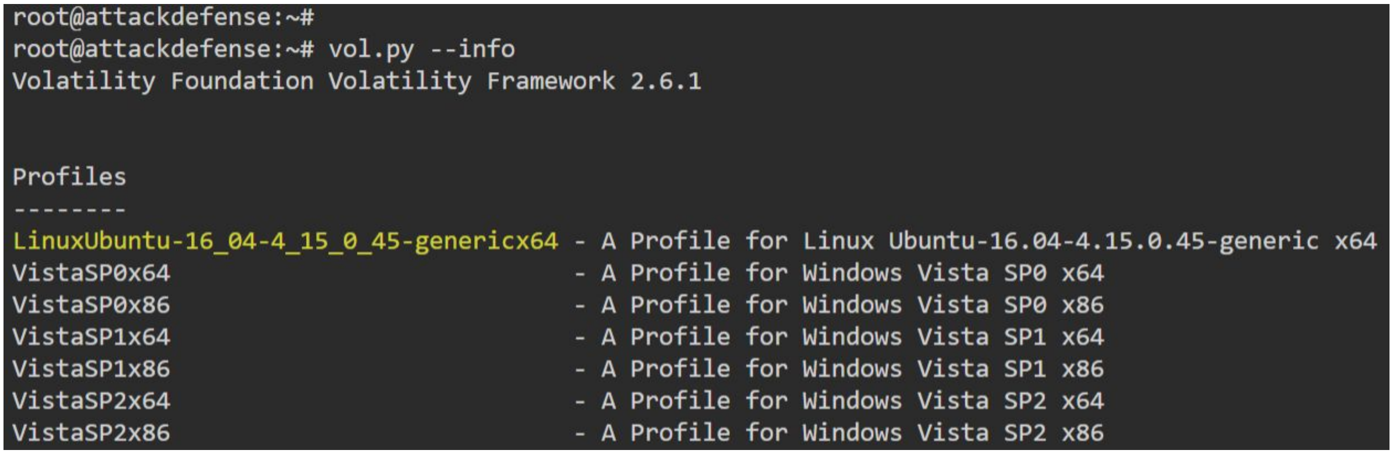
```
root@attackdefense:~#
root@attackdefense:~# vol.py --info
Volatility Foundation Volatility Framework 2.6.1


Profiles
--------
LinuxUbuntu-16_04-4_15_0_45-genericx64 - A Profile for Linux Ubuntu-16.04-4.15.0.45-generic x64
VistaSP0x64                            - A Profile for Windows Vista SP0 x64
VistaSP0x86                            - A Profile for Windows Vista SP0 x86
VistaSP1x64                            - A Profile for Windows Vista SP1 x64
VistaSP1x86                            - A Profile for Windows Vista SP1 x86
VistaSP2x64                            - A Profile for Windows Vista SP2 x64
VistaSP2x86                            - A Profile for Windows Vista SP2 x86
```

Q2. What is the name of the profile which is present for Ubuntu Linux?

Answer: LinuxUbuntu-16_04-4_15_0_45-genericx64

Command: vol.py --info

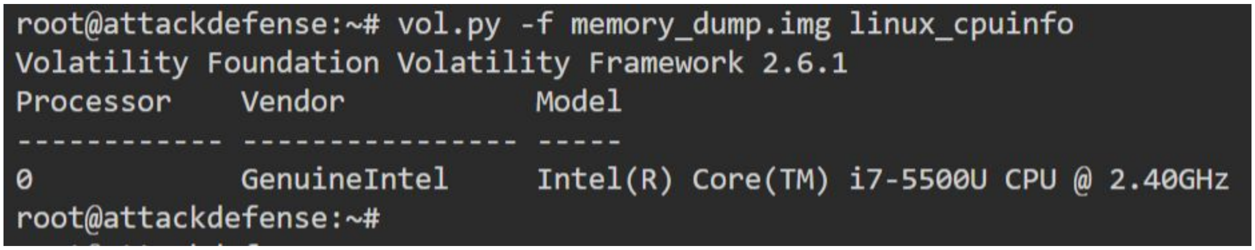


```
root@attackdefense:~#
root@attackdefense:~# vol.py --info
Volatility Foundation Volatility Framework 2.6.1


Profiles
--------
LinuxUbuntu-16_04-4_15_0_45-genericx64 - A Profile for Linux Ubuntu-16.04-4.15.0.45-generic x64
VistaSP0x64                            - A Profile for Windows Vista SP0 x64
VistaSP0x86                            - A Profile for Windows Vista SP0 x86
VistaSP1x64                            - A Profile for Windows Vista SP1 x64
VistaSP1x86                            - A Profile for Windows Vista SP1 x86
VistaSP2x64                            - A Profile for Windows Vista SP2 x64
VistaSP2x86                            - A Profile for Windows Vista SP2 x86
```

Q3. Which command can be used to extract the CPU details from the memory dump?

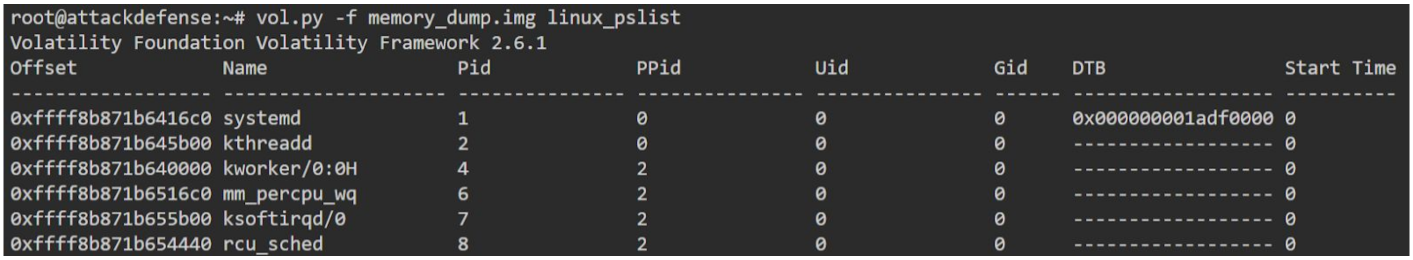
Command: vol.py -f memory_dump.img linux_cpuinfo



```
root@attackdefense:~# vol.py -f memory_dump.img linux_cpuinfo
Volatility Foundation Volatility Framework 2.6.1
Processor    Vendor           Model
------------ ---------------- -----
0            GenuineIntel     Intel(R) Core(TM) i7-5500U CPU @ 2.40GHz
root@attackdefense:~#
```

Q4. Which command can be used to retrieve the list running processes from memory dump?

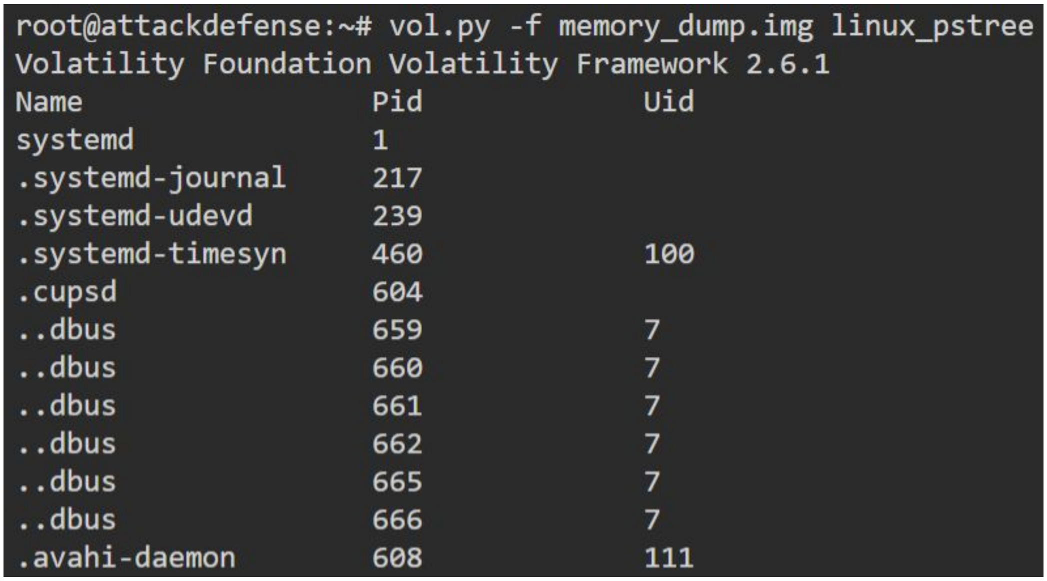
Command: vol.py -f memory_dump.img linux_pslist



```
root@attackdefense:~# vol.py -f memory_dump.img linux_pslist
Volatility Foundation Volatility Framework 2.6.1
Offset             Name                 Pid            PPid           Uid            Gid    DTB                Start Time
------------------ -------------------- -------------- -------------- -------------- ------ ------------------ ----------
0xffff8b871b6416c0 systemd              1              0              0              0      0x000000001adf0000 0
0xffff8b871b645b00 kthreadd             2              0              0              0      ------------------ 0
0xffff8b871b640000 kworker/0:0H         4              2              0              0      ------------------ 0
0xffff8b871b6516c0 mm_percpu_wq         6              2              0              0      ------------------ 0
0xffff8b871b655b00 ksoftirqd/0          7              2              0              0      ------------------ 0
0xffff8b871b654440 rcu_sched            8              2              0              0      ------------------ 0
```

Q5. Which command can be used to show the processes in the parent-child relationship format?

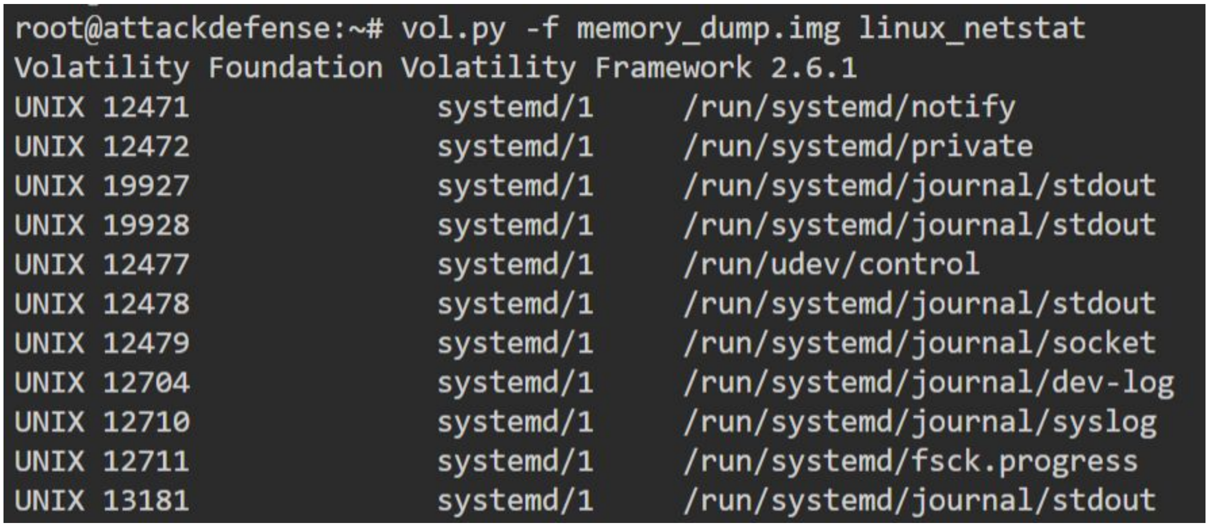
Command: vol.py -f memory_dump.img linux_pstree



```
root@attackdefense:~# vol.py -f memory_dump.img linux_pstree
Volatility Foundation Volatility Framework 2.6.1
Name                 Pid             Uid
systemd              1
.systemd-journal     217
.systemd-udevd       239
.systemd-timesyn     460             100
.cupsd               604
..dbus               659             7
..dbus               660             7
..dbus               661             7
..dbus               662             7
..dbus               665             7
..dbus               666             7
.avahi-daemon        608             111
```

Q6. Which command can be used to extract the list of open TCP connections from the memory dump?

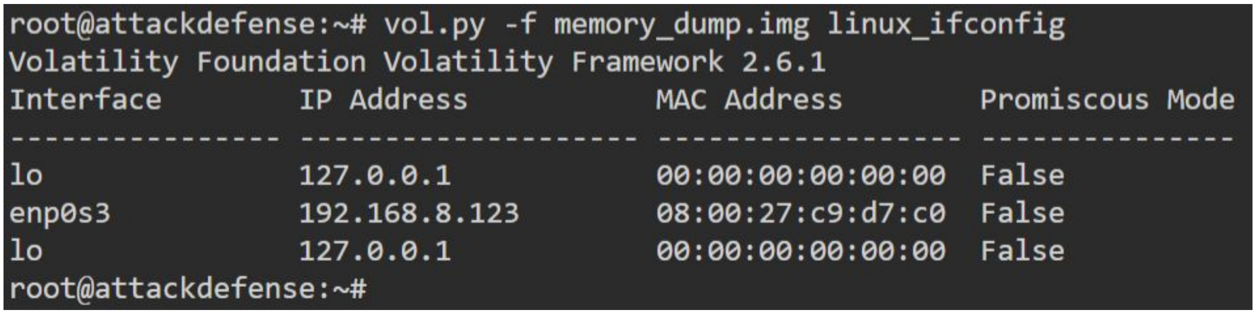
Command: vol.py -f memory_dump.img linux_netstat



```
root@attackdefense:~# vol.py -f memory_dump.img linux_netstat
Volatility Foundation Volatility Framework 2.6.1
UNIX 12471              systemd/1     /run/systemd/notify
UNIX 12472              systemd/1     /run/systemd/private
UNIX 19927              systemd/1     /run/systemd/journal/stdout
UNIX 19928              systemd/1     /run/systemd/journal/stdout
UNIX 12477              systemd/1     /run/udev/control
UNIX 12478              systemd/1     /run/systemd/journal/stdout
UNIX 12479              systemd/1     /run/systemd/journal/socket
UNIX 12704              systemd/1     /run/systemd/journal/dev-log
UNIX 12710              systemd/1     /run/systemd/journal/syslog
UNIX 12711              systemd/1     /run/systemd/fsck.progress
UNIX 13181              systemd/1     /run/systemd/journal/stdout
```

Q7. **What was the IP address of the machine on which the memory dump was taken?**

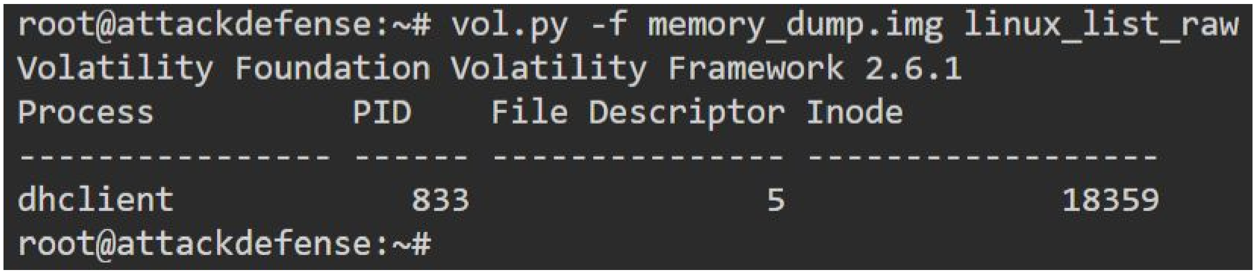
Command: vol.py -f memory_dump.img linux_ifconfig



```
root@attackdefense:~# vol.py -f memory_dump.img linux_ifconfig
Volatility Foundation Volatility Framework 2.6.1
Interface        IP Address           MAC Address        Promiscous Mode
---------------- -------------------- ------------------ ---------------
lo               127.0.0.1            00:00:00:00:00:00  False
enp0s3           192.168.8.123        08:00:27:c9:d7:c0  False
lo               127.0.0.1            00:00:00:00:00:00  False
root@attackdefense:~#
```

Q8. **Which command can identify the applications (which are using a promiscuous socket) from the memory dump?**

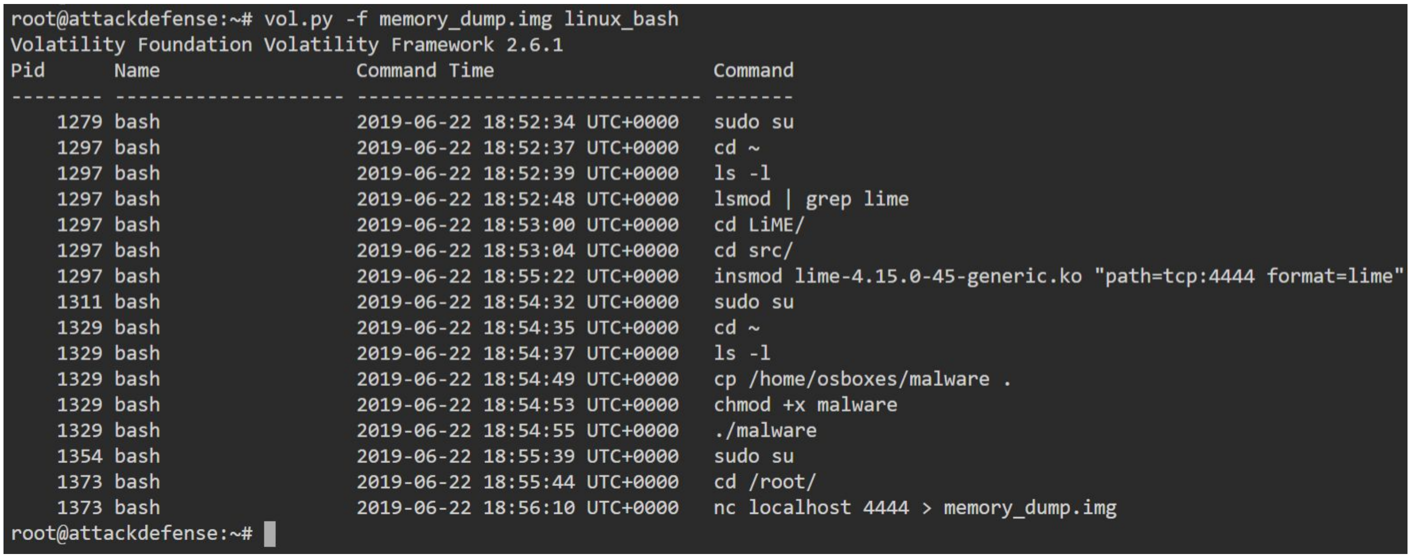
Command: vol.py -f memory_dump.img linux_list_raw



```
root@attackdefense:~# vol.py -f memory_dump.img linux_list_raw
Volatility Foundation Volatility Framework 2.6.1
Process           PID    File Descriptor Inode
----------------- ------ --------------- ------------------
dhclient             833               5              18359
root@attackdefense:~#
```

Q9. Which command can be used to recover the bash command history from the memory dump?

Command: vol.py -f memory_dump.img linux_bash



```
root@attackdefense:~# vol.py -f memory_dump.img linux_bash
Volatility Foundation Volatility Framework 2.6.1
Pid      Name                 Command Time                   Command
-------- -------------------- ------------------------------ -------
    1279 bash                 2019-06-22 18:52:34 UTC+0000   sudo su
    1297 bash                 2019-06-22 18:52:37 UTC+0000   cd ~
    1297 bash                 2019-06-22 18:52:39 UTC+0000   ls -l
    1297 bash                 2019-06-22 18:52:48 UTC+0000   lsmod | grep lime
    1297 bash                 2019-06-22 18:53:00 UTC+0000   cd LiME/
    1297 bash                 2019-06-22 18:53:04 UTC+0000   cd src/
    1297 bash                 2019-06-22 18:55:22 UTC+0000   insmod lime-4.15.0-45-generic.ko "path=tcp:4444 format=lime"
    1311 bash                 2019-06-22 18:54:32 UTC+0000   sudo su
    1329 bash                 2019-06-22 18:54:35 UTC+0000   cd ~
    1329 bash                 2019-06-22 18:54:37 UTC+0000   ls -l
    1329 bash                 2019-06-22 18:54:49 UTC+0000   cp /home/osboxes/malware .
    1329 bash                 2019-06-22 18:54:53 UTC+0000   chmod +x malware
    1329 bash                 2019-06-22 18:54:55 UTC+0000   ./malware
    1354 bash                 2019-06-22 18:55:39 UTC+0000   sudo su
    1373 bash                 2019-06-22 18:55:44 UTC+0000   cd /root/
    1373 bash                 2019-06-22 18:56:10 UTC+0000   nc localhost 4444 > memory_dump.img
root@attackdefense:~#
```

References:

1. Volatility (https://github.com/volatilityfoundation/volatility)