CSEC 730 - Advanced Computer Forensics
Homework 2 – Memory Analysis Using Volatility

Please submit your answers (in pdf format) to the assignment submission folder on *myCourses* > *Assignments* by the due date.

Goal
The open-source toolkit, volatility framework, is one of the best memory forensic analysis tools to extract valuable information from a memory dump or a .vmem file. In this activity, you will practice volatility's basic plugins for extracting important volatile data from memory images.

Software

- Volatility2 is installed on SIFT Workstation. Volatility documentation, including a list of image types that Volatility can analyze
- Volatility3 was released in 2020. You have to install it on the SIFT VM by yourself. See Volatility3 documentation

Memory images to be analyzed

In this lab, you will use both volatility2 and volatility3 to analyze the Windows memory dump, *zeus.vmem*, from the Malware Analyst's Cookbook DVD. Zeus is a malware designed to steal credentials. You will also use volatility2 to analyze your SIFT memory.

- Download *zeus.vmem.zip* from myCourses Content > Project and Homework > Homework2
- Extract the zip file and save it to your desktop.
- The md5 hash for *zeus.vmem* is b6e4817d7c1aea69bbd8b19b42075681
- The sha1 hash for *zeus.vmem* is e67f018663089c05a2ad8dd8d5a2d7c53c35c4ca

Part 1. Windows Memory Analysis Using Volatility2 (40 points)

Instructions
1. Launch SIFT Workstation. Volatility2 is installed by default.
2. Run vol.py –h to see volatility2's options and plugins
3. Read the Windows volatility2 plugins at https://github.com/volatilityfoundation/volatility/wiki/Command-Reference
4. Practice volatility2 basic plugins to understand how you can use the result for your investigation.

imageinfo	shows basic system information such as type of OS
pslist	Lists the processes of a system
psscan	Finds processes that previously terminated (inactive) and processes that have been hidden or unlinked by a rootkit
pstree	Displays the process listing in tree form
connections	Shows the TCP connections that were active at the time of the memory acquisition
connscan	Extracts TCP connections that were active at the time of the memory acquisition and previous connections that have since been terminated.
hivelist	Locates the virtual addresses of registry hives in memory and the full paths to the corresponding hive on disk
hivescan	Displays the physical addresses of registry hives in memory

printkey	Displays the subkeys, values, data, and data types contained within a specified registry key
userassist	Prints userassist registry keys and information showing what programs were executed on the system

Using volatility2 to analyze *zeus.vmem* and answer all the questions for part 1.

1. **What** is the suggested type of OS of *zeus.vmem* and when was the sample collected? Provide screenshots as your supporting data.
   - The suggested OS of zeus.vmem is Windows and the sample was collected on 08/15/2010.

```
sansforensics@siftworkstation: ~/homework2
$ vol.py imageinfo -f ./zeus.vmem
Volatility Foundation Volatility Framework 2.6.1
INFO    : volatility.debug    : Determining profile based on KDBG sear
ch...
          Suggested Profile(s) : WinXPSP2x86, WinXPSP3x86 (Instantiate
d with WinXPSP2x86)
                     AS Layer1 : IA32PagedMemoryPae (Kernel AS)
                     AS Layer2 : FileAddressSpace (/home/sansforensics
/homework2/zeus.vmem)
                      PAE type : PAE
                           DTB : 0x319000L
                          KDBG : 0x80544ce0L
          Number of Processors : 1
     Image Type (Service Pack) : 2
                KPCR for CPU 0 : 0xffdff000L
             KUSER_SHARED_DATA : 0xffdf0000L
           Image date and time : 2010-08-15 19:17:56 UTC+0000
     Image local date and time : 2010-08-15 15:17:56 -0400
sansforensics@siftworkstation: ~/homework2
$
```

2. Comparing *pslist* with *psscan*, which plugin walks through the doubly-linked list of EPROCESS pointed by PsActiveProcessHead? **Which** one does not rely on the doubly-list of EPROCESS and can detect unlinked (hidden) processes?
   - The *pslist* plugin walks through the doubly-linked list of EPROCESS pointed by PsActiveProcessHead. The *psscan* does not rely on the doubly-list of EPROCESS and can detect unlinked (hidden) processes.

Provide a screenshot to show the processes that are potentially hidden.

```
sansforensics@siftworkstation: ~/homework2
$ vol.py pslist -f ./zeus.vmem
Volatility Foundation Volatility Framework 2.6.1
Offset(V)  Name                    PID   PPID   Thds     Hnds   Sess  Wow64 Start                          Exit
---------- -------------------- ------ ------ ------ -------- ------ ------ ------------------------------ ------------------------------
0x810b1660 System                    4      0     58      379 ------      0
0xff2ab020 smss.exe                544      4      3       21 ------      0 2010-08-11 06:06:21 UTC+0000
0xff1ecda0 csrss.exe               608    544     10      410      0      0 2010-08-11 06:06:23 UTC+0000
0xff1ec978 winlogon.exe            632    544     24      536      0      0 2010-08-11 06:06:23 UTC+0000
0xff247020 services.exe            676    632     16      288      0      0 2010-08-11 06:06:24 UTC+0000
0xff255020 lsass.exe               688    632     21      405      0      0 2010-08-11 06:06:24 UTC+0000
0xff218230 vmacthlp.exe            844    676      1       37      0      0 2010-08-11 06:06:24 UTC+0000
0x80ff88d8 svchost.exe             856    676     29      336      0      0 2010-08-11 06:06:24 UTC+0000
0xff217560 svchost.exe             936    676     11      288      0      0 2010-08-11 06:06:24 UTC+0000
0x80fbf910 svchost.exe            1028    676     88     1424      0      0 2010-08-11 06:06:24 UTC+0000
0xff22d558 svchost.exe            1088    676      7       93      0      0 2010-08-11 06:06:25 UTC+0000
0xff203b80 svchost.exe            1148    676     15      217      0      0 2010-08-11 06:06:26 UTC+0000
0xff1d7da0 spoolsv.exe            1432    676     14      145      0      0 2010-08-11 06:06:26 UTC+0000
0xff1b8b28 vmtoolsd.exe           1668    676      5      225      0      0 2010-08-11 06:06:35 UTC+0000
0xff1fdc88 VMUpgradeHelper        1788    676      5      112      0      0 2010-08-11 06:06:38 UTC+0000
0xff143b28 TPAutoConnSvc.e        1968    676      5      106      0      0 2010-08-11 06:06:39 UTC+0000
0xff25a7e0 alg.exe                 216    676      8      120      0      0 2010-08-11 06:06:39 UTC+0000
0xff364310 wscntfy.exe             888   1028      1       40      0      0 2010-08-11 06:06:49 UTC+0000
0xff38b5f8 TPAutoConnect.e        1084   1968      1       68      0      0 2010-08-11 06:06:52 UTC+0000
0x80f60da0 wuauclt.exe            1732   1028      7      189      0      0 2010-08-11 06:07:44 UTC+0000
0xff3865d0 explorer.exe           1724   1708     13      326      0      0 2010-08-11 06:09:29 UTC+0000
0xff3667e8 VMwareTray.exe          432   1724      1       60      0      0 2010-08-11 06:09:31 UTC+0000
0xff374980 VMwareUser.exe          452   1724      8      207      0      0 2010-08-11 06:09:32 UTC+0000
0x80f94588 wuauclt.exe             468   1028      4      142      0      0 2010-08-11 06:09:37 UTC+0000
0xff224020 cmd.exe                 124   1668      0 --------      0      0 2010-08-15 19:17:55 UTC+0000   2010-08-15 19:17:56 UTC+0000
sansforensics@siftworkstation: ~/homework2
```

```
sansforensics@siftworkstation: ~/homework2
$ vol.py psscan -f ./zeus.vmem
Volatility Foundation Volatility Framework 2.6.1
Offset(P)          Name                 PID   PPID PDB        Time created                   Time exited
------------------ ---------------- ------ ------ ---------- ------------------------------ ------------------------------
0x00000000010c3da0 wuauclt.exe        1732   1028 0x06cc02c0 2010-08-11 06:07:44 UTC+0000
0x00000000010f7588 wuauclt.exe         468   1028 0x06cc0180 2010-08-11 06:09:37 UTC+0000
0x0000000001122910 svchost.exe        1028    676 0x06cc0120 2010-08-11 06:06:24 UTC+0000
0x000000000115b8d8 svchost.exe         856    676 0x06cc00e0 2010-08-11 06:06:24 UTC+0000
0x0000000001214660 System                4      0 0x00319000
0x000000000211ab28 TPAutoConnSvc.e    1968    676 0x06cc0260 2010-08-11 06:06:39 UTC+0000
0x00000000049c15f8 TPAutoConnect.e    1084   1968 0x06cc0220 2010-08-11 06:06:52 UTC+0000
0x0000000004a065d0 explorer.exe       1724   1708 0x06cc0280 2010-08-11 06:09:29 UTC+0000
0x0000000004b5a980 VMwareUser.exe      452   1724 0x06cc0300 2010-08-11 06:09:32 UTC+0000
0x0000000004be97e8 VMwareTray.exe      432   1724 0x06cc02e0 2010-08-11 06:09:31 UTC+0000
0x0000000004c2b310 wscntfy.exe         888   1028 0x06cc0200 2010-08-11 06:06:49 UTC+0000
0x0000000005471020 smss.exe            544      4 0x06cc0020 2010-08-11 06:06:21 UTC+0000
0x0000000005f027e0 alg.exe             216    676 0x06cc0240 2010-08-11 06:06:39 UTC+0000
0x0000000005f47020 lsass.exe           688    632 0x06cc00a0 2010-08-11 06:06:24 UTC+0000
0x0000000006015020 services.exe        676    632 0x06cc0080 2010-08-11 06:06:24 UTC+0000
0x00000000061ef558 svchost.exe        1088    676 0x06cc0140 2010-08-11 06:06:25 UTC+0000
0x0000000006238020 cmd.exe             124   1668 0x06cc02a0 2010-08-15 19:17:55 UTC+0000   2010-08-15 19:17:56 UTC+0000
0x0000000006384230 vmacthlp.exe        844    676 0x06cc00c0 2010-08-11 06:06:24 UTC+0000
0x00000000063c5560 svchost.exe         936    676 0x06cc0100 2010-08-11 06:06:24 UTC+0000
0x0000000006499b80 svchost.exe        1148    676 0x06cc0160 2010-08-11 06:06:26 UTC+0000
0x000000000655fc88 VMUpgradeHelper    1788    676 0x06cc01e0 2010-08-11 06:06:38 UTC+0000
0x00000000066f0978 winlogon.exe        632    544 0x06cc0060 2010-08-11 06:06:23 UTC+0000
0x00000000066f0da0 csrss.exe           608    544 0x06cc0040 2010-08-11 06:06:23 UTC+0000
0x0000000006945da0 spoolsv.exe        1432    676 0x06cc01a0 2010-08-11 06:06:26 UTC+0000
0x00000000069a7328 VMip.exe           1944    124 0x06cc0320 2010-08-15 19:17:55 UTC+0000   2010-08-15 19:17:56 UTC+0000
0x00000000069d5b28 vmtoolsd.exe       1668    676 0x06cc01c0 2010-08-11 06:06:35 UTC+0000
sansforensics@siftworkstation: ~/homework2
```

3. Run *vol.py* using both *connections* and *connscan*. (Note: both *connections* and *connscan* do not work for Windows Vista and later version memory image. You will use plugin *netscan* instead)

   Do you see any **active** TCP connections or **previous** connections? Provide screenshots as your supporting data.

   - Yes, there was a previous TCP connection made to the remote address of 193.104.41.75 at port 80.

```
sansforensics@siftworkstation: ~/homework2
$ vol.py connections -f ./zeus.vmem
Volatility Foundation Volatility Framework 2.6.1
Offset(V)  Local Address             Remote Address            Pid
---------- ------------------------- ------------------------- ---
sansforensics@siftworkstation: ~/homework2
$ vol.py connscan -f ./zeus.vmem
Volatility Foundation Volatility Framework 2.6.1
Offset(P)  Local Address             Remote Address            Pid
---------- ------------------------- ------------------------- ---
0x02214988 172.16.176.143:1054       193.104.41.75:80          856
0x06015ab0 0.0.0.0:1056              193.104.41.75:80          856
sansforensics@siftworkstation: ~/homework2
$
```

Which process made these TCP connections?

- The process that made the TCP connections was PID of 856.

Using "whois RemoteAddress" to find out where the IP is located. Provide screenshots as your supporting data.

- The IP address 193.104.41.75 is registered to an organization named ISP Alliance a.s., which is located in the Czech Republic. The network is named 'CZ-GRAPESC-20191115', and the abuse contact for this IP range is 'abuse@ecomp.eu'.

  The address and contact details associated with the organization are as follows:

  Organization Name: ISP Alliance a.s.
  Address: Kukanova 2262, 43003 Chomutov, CZECH REPUBLIC
  Phone: +420724612176
  Fax: +420415210805
  Email: noc@ispalliance.cz
  The technical contact appears to be a person named Ladislav Ruzicka, with the email addresses ruzicka@suds.cz and ruzicka@grapesc.cz, and a phone number of +420604469324.

  The routing for the IP range 193.104.41.0/24 is managed under the autonomous system number AS207886. The maintenance for this routing is handled by mnt-cz-ecomp-1 according to the RIPE database.

```
sansforensics@siftworkstation: ~/homework2
$ whois 193.104.41.75 -B
% This is the RIPE Database query service.
% The objects are in RPSL format.
%
% The RIPE Database is subject to Terms and Conditions.
% See https://apps.db.ripe.net/docs/HTML-Terms-And-Conditions

% Information related to '193.104.41.0 - 193.104.41.255'

% Abuse contact for '193.104.41.0 - 193.104.41.255' is 'abuse@ecomp.eu'

inetnum:        193.104.41.0 - 193.104.41.255
netname:        CZ-GRAPESC-20191115
country:        CZ
org:            ORG-GSa22-RIPE
admin-c:        LR2038-RIPE
tech-c:         GS5801-RIPE
abuse-c:        AC38633-RIPE
status:         ALLOCATED PA
mnt-by:         MNT-GRAPESC
mnt-by:         RIPE-NCC-HM-MNT
created:        2021-12-16T14:36:07Z
last-modified:  2023-10-10T09:02:23Z
source:         RIPE
```

```
organisation:   ORG-GSa22-RIPE
org-name:       ISP Alliance a.s.
country:        CZ
org-type:       LIR
address:        Kukanova 2262
address:        43003
address:        Chomutov
address:        CZECH REPUBLIC
phone:          +420724612176
fax-no:         +420415210805
e-mail:         noc@ispalliance.cz
admin-c:        LR2038-RIPE
abuse-c:        IA3261-RIPE
mnt-ref:        RIPE-NCC-HM-MNT
mnt-ref:        MNT-GRAPESC
mnt-by:         RIPE-NCC-HM-MNT
mnt-by:         MNT-GRAPESC
created:        2008-04-22T12:13:14Z
last-modified:  2023-12-18T13:24:38Z
source:         RIPE

role:           GRAPESC
address:        Karlovo nám?stí 559/28
address:        120 00 Praha 2
address:        Czech Republic
phone:          +420800911911
e-mail:         ruzicka@grapesc.cz
admin-c:        LR2038-RIPE
tech-c:         LR2038-RIPE
abuse-mailbox:  hotline@grapesc.cz
nic-hdl:        GS5801-RIPE
mnt-by:         MNT-GRAPESC
notify:         ruzicka@grapesc.cz
created:        2008-05-14T08:34:47Z
last-modified:  2023-07-18T16:39:57Z
source:         RIPE
```

```
person:         Ladislav Ruzicka
address:        Kostnicka 2341
address:        Chomutov
address:        Czech Republic
e-mail:         ruzicka@suds.cz
e-mail:         ruzicka@grapesc.cz
phone:          +420604469324
mnt-by:         MNT-GRAPESC
nic-hdl:        LR2038-RIPE
notify:         ruzicka@suds.cz
created:        2008-04-29T08:12:40Z
last-modified:  2018-07-27T07:45:42Z
source:         RIPE

% Information related to '193.104.41.0/24AS207886'

route:          193.104.41.0/24
origin:         AS207886
mnt-by:         mnt-cz-ecomp-1
created:        2019-11-26T08:28:59Z
last-modified:  2019-11-26T08:28:59Z
source:         RIPE

% This query was served by the RIPE Database Query Service version 1.109.1 (DEXTER)
```

4. Run *vol.py –f zeus.vmem hivelist*, *vol.py –f zeus.vmem hivescan*, and *vol.py -f zeus.vmem printkey -K "Microsoft\Windows NT\CurrentVersion\Winlogon"*.

   Which plugin shows the virtual addresses of registry hives in memory along with the full paths to the corresponding hive on disk? Provide screenshots as your supporting data.
   - The hivelist shows in memory along with the full paths to the corresponding hive on disk.

```
sansforensics@siftworkstation: ~/homework2
$ vol.py hivelist -f ./zeus.vmem
Volatility Foundation Volatility Framework 2.6.1
Virtual    Physical   Name
---------- ---------- ----
0xe1c49008 0x036dc008 \Device\HarddiskVolume1\Documents and Settings\LocalService\Local Settings\Application Data\Microsoft\Windows\UsrClass.d
at
0xe1c41b60 0x04010b60 \Device\HarddiskVolume1\Documents and Settings\LocalService\NTUSER.DAT
0xe1a39638 0x021eb638 \Device\HarddiskVolume1\Documents and Settings\NetworkService\Local Settings\Application Data\Microsoft\Windows\UsrClass
.dat
0xe1a33008 0x01f98008 \Device\HarddiskVolume1\Documents and Settings\NetworkService\NTUSER.DAT
0xe153ab60 0x06b7db60 \Device\HarddiskVolume1\WINDOWS\system32\config\software
0xe1542008 0x06c48008 \Device\HarddiskVolume1\WINDOWS\system32\config\default
0xe1537b60 0x06ae4b60 \SystemRoot\System32\Config\SECURITY
0xe1544008 0x06c4b008 \Device\HarddiskVolume1\WINDOWS\system32\config\SAM
0xe13ae580 0x01bbd580 [no name]
0xe101b008 0x01867008 \Device\HarddiskVolume1\WINDOWS\system32\config\system
0xe1008978 0x01824978 [no name]
0xe1e158c0 0x009728c0 \Device\HarddiskVolume1\Documents and Settings\Administrator\Local Settings\Application Data\Microsoft\Windows\UsrClass.
dat
0xe1da4008 0x00f6e008 \Device\HarddiskVolume1\Documents and Settings\Administrator\NTUSER.DAT
sansforensics@siftworkstation: ~/homework2
```

The string '*Userinit*' specifies the executables that Winlogon runs after a user logs into Windows. The default executable is C:\windows\system32\userinit.exe which restores your profile, fonts, colors, etc. for your username. **It is possible to add additional executables to '*Userinit*' by separating the executables with a comma. It's a common place for trojans.** The *Userinit* entry is resided in *"Microsoft\Windows NT\CurrentVersion\Winlogon"*.

After you run *vol.py -f zeus.vmem printkey -K "Microsoft\Windows NT\CurrentVersion\Winlogon",* **Which** suspicious executable(s) do you see in *Userinit*? Provide screenshots as your supporting data.

- The suspicious executable that I see in Userinit is the sdra64.exe file.

```
REG_SZ        DefaultUserName : (S) Administrator
REG_SZ        LegalNoticeCaption : (S)
REG_SZ        LegalNoticeText : (S)
REG_SZ        PowerdownAfterShutdown : (S) 0
REG_SZ        ReportBootOk    : (S) 1
REG_SZ        Shell           : (S) Explorer.exe
REG_SZ        ShutdownWithoutLogon : (S) 0
REG_SZ        System          : (S)
REG_SZ        Userinit        : (S) C:\WINDOWS\system32\userinit.exe,C:\WINDOWS\system32\sdra64.exe,
REG_SZ        VmApplet        : (S) rundll32 shell32,Control_RunDLL "sysdm.cpl"
REG_DWORD     SfcQuota        : (S) 4294967295
REG_SZ        allocatecdroms  : (S) 0
REG_SZ        allocatedasd    : (S) 0
REG_SZ        allocatefloppies : (S) 0
REG_SZ        cachedlogonscount : (S) 10
REG_DWORD     forceunlocklogon : (S) 0
REG_DWORD     passwordexpirywarning : (S) 14
REG_SZ        scremoveoption  : (S) 0
REG_DWORD     AllowMultipleTSSessions : (S) 1
REG_EXPAND_SZ UIHost          : (S) logonui.exe
REG_DWORD     LogonType       : (S) 1
```

5. Run *vol.py* using the plugin, *pstree,* to view the process listing in tree form.

   Based on the results from Q3 and Q4 above, **what** can you conclude by analyzing Pid and PPid in the process tree list? (hint: which program launched the process that made the internet connection in Q3?). Provide screenshots as your supporting data.

   - According to the tree, it seems that svchost.exe is the process with the PID of 856. The PPID of 856 is 676, and the PPID of the process with PID of 676 is 636. The process with PID of 636 is winlogon.exe. If we look at the screenshot in Q4, we can see a suspicious executable and that is sdra64.exe. In conclusion, as said in Q4 that the string 'Userinit' specifies the executables that Winlogon runs after a user logs into Windows, after the winlogon.exe with the PID of 636 is ran, the malware that is a trojan ran hid itself in the svchost.exe and ran in their and probably trying to connect to the remote host of 193.104.41.75.

```
sansforensics@siftworkstation: ~/homework2
$ vol.py pstree -f ./zeus.vmem
Volatility Foundation Volatility Framework 2.6.1
Name                                                  Pid   PPid   Thds   Hnds Time
-------------------------------------------------- ------ ------ ------ ------ ----
 0x810b1660:System                                      4      0     58    379 1970-01-01 00:00:00 UTC+0000
. 0xff2ab020:smss.exe                                 544      4      3     21 2010-08-11 06:06:21 UTC+0000
.. 0xff1ec978:winlogon.exe                            632    544     24    536 2010-08-11 06:06:23 UTC+0000
... 0xff255020:lsass.exe                              688    632     21    405 2010-08-11 06:06:24 UTC+0000
... 0xff247020:services.exe                           676    632     16    288 2010-08-11 06:06:24 UTC+0000
.... 0xff1b8b28:vmtoolsd.exe                         1668    676      5    225 2010-08-11 06:06:35 UTC+0000
..... 0xff224020:cmd.exe                              124   1668      0 ------ 2010-08-15 19:17:55 UTC+0000
.... 0x80ff88d8:svchost.exe                           856    676     29    336 2010-08-11 06:06:24 UTC+0000
.... 0xff1d7da0:spoolsv.exe                          1432    676     14    145 2010-08-11 06:06:26 UTC+0000
.... 0x80fbf910:svchost.exe                          1028    676     88   1424 2010-08-11 06:06:24 UTC+0000
..... 0x80f60da0:wuauclt.exe                         1732   1028      7    189 2010-08-11 06:07:44 UTC+0000
..... 0x80f94588:wuauclt.exe                          468   1028      4    142 2010-08-11 06:09:37 UTC+0000
..... 0xff364310:wscntfy.exe                          888   1028      1     40 2010-08-11 06:06:49 UTC+0000
.... 0xff217560:svchost.exe                           936    676     11    288 2010-08-11 06:06:24 UTC+0000
.... 0xff143b28:TPAutoConnSvc.e                      1968    676      5    106 2010-08-11 06:06:39 UTC+0000
..... 0xff38b5f8:TPAutoConnect.e                     1084   1968      1     68 2010-08-11 06:06:52 UTC+0000
.... 0xff22d558:svchost.exe                          1088    676      7     93 2010-08-11 06:06:25 UTC+0000
.... 0xff218230:vmacthlp.exe                          844    676      1     37 2010-08-11 06:06:24 UTC+0000
.... 0xff25a7e0:alg.exe                               216    676      8    120 2010-08-11 06:06:39 UTC+0000
.... 0xff203b80:svchost.exe                          1148    676     15    217 2010-08-11 06:06:26 UTC+0000
.... 0xff1fdc88:VMUpgradeHelper                      1788    676      5    112 2010-08-11 06:06:38 UTC+0000
.. 0xff1ecda0:csrss.exe                               608    544     10    410 2010-08-11 06:06:23 UTC+0000
 0xff3865d0:explorer.exe                             1724   1708     13    326 2010-08-11 06:09:29 UTC+0000
. 0xff374980:VMwareUser.exe                           452   1724      8    207 2010-08-11 06:09:32 UTC+0000
. 0xff3667e8:VMwareTray.exe                           432   1724      1     60 2010-08-11 06:09:31 UTC+0000
sansforensics@siftworkstation: ~/homework2
$
```

6. Try other plugins from the Windows volatility2 plugins at https://github.com/volatilityfoundation/volatility/wiki/Command-Reference or https://github.com/volatilityfoundation/volatility/wiki/Command-Reference-Mal. show me at least two other plugins that provide you interesting results.
   - The other two plugins that I used are malfine,and then impscan on the file. The output shown in the file, Part1_Step6.txt, indicates that malfind has detected potential malicious or injected code segments associated with the process with a Process ID (PID) of 856. The memory segments starting at 0x00b70000 and 0x00cb0000 have been identified because they are executable (PAGE_EXECUTE_READWRITE) and marked as private (not shared between processes). Such segments could contain executable code that is not part of the process's legitimate memory image. In particular, the presence of the MZ header (4D 5A at the start of the memory segment) suggests that this is an executable segment of memory, and since these headers are associated with the beginning of an executable file (PE format on Windows), it can indicate that an executable has been injected into the process' address space. This is commonly seen in cases of process injection, where an attacker injects malicious code into a running process to execute it stealthily.
   - Then I used impscan to the starting segments at 0x00b70000 and 0x00cb0000. There was only output for the 0x00b70000. The output of the impscan command indicates that a variety of Windows API functions are being called from different libraries like ADVAPI32.dll, KERNEL32.dll, USER32.dll, GDI32.dll, and others. These calls are part of the Import Address Table (IAT) that svchost.exe (PID: 856) would use to interact with the operating system and other processes.
     - Functions related to Network and File Operations: The list includes InternetReadFile, InternetOpenUrlA, HttpSendRequestA, WSASend, bind,

socket, which could be used for communication over the network, potentially with a command and control server.
    - Low-level API Calls: Calls to ZwQueryInformationProcess and RtlCreateUserThread from ntdll.dll are lower-level system calls that can be used for process injection or to query sensitive process information.
    - There are other functions that are mentioned in the file I have provided

7. (ADVANCED, NOT REQUIRED) Try plugins (for example, apihooks and malfind) from https://code.google.com/archive/p/volatility/wikis/CommandReferenceMal22.wiki to identify malicious code injection or hunt rootkits, what additional information do you find?
    - As stated, this question is not required. However, I used malfind and impscan in the previous question to show the use of two other plugins.

Part 2. Linux Memory Analysis Using Volatility2 (30 points)

In homework 1, you dumped out your SIFT memory to the file, *yourusername_memory_dump.bin* (assume the file is saved on the SIFT desktop). In this exercise, you will use volatility2 to extract useful information from your SIFT memory.

Linux profiles are available at https://github.com/volatilityfoundation/profiles . However, **volatility2 is very picky about Linux profile. In most cases, you have to build your own profile from the machine where the memory was exacted. In our case, it is the Ubuntux64 SIFT machine.**

The steps to build your own Ubuntux64 profile from your SIFT machine:

1. Install dwarfdump package and kernel headers
$sudo apt-get update
$sudo apt-get install dwarfdump pcregrep libpcre++-dev yara -y
$sudo -H pip install pycrypto distorm3 openpyxl ujson pillow

2. Download volatility repo
$cd ~/Downloads/
$git clone https://github.com/volatilityfoundation/volatility.git
$cd volatility/tools/linux

3. Generate the module.dwarf file using make
$make

4. Create the profile zip for SIFT (LinuxUbuntux64.zip) and place it in the volatility overlays/linux folder where volatility looks for all profiles, given the System.map file's for the kernel version.
$sudo zip /usr/local/lib/python2.7/dist-packages/volatility/plugins/overlays/linux/Ubuntu.zip module.dwarf /boot/System.map-$(uname -r)

```
sansforensics@siftworkstation: /usr/local/lib/python2.7/dist-packages/volatility/plugins/overlays/linux
$ cd ~/Downloads/
sansforensics@siftworkstation: ~/Downloads
$ ls
volatility
sansforensics@siftworkstation: ~/Downloads
$ cd volatility/tools/linux/
sansforensics@siftworkstation: ~/Downloads/volatility/tools/linux
$ sudo zip /usr/local/lib/python2.7/dist-packages/volatility/plugins/overlays/linux/Ubuntu.zip module.dwarf /boot/System.map-$(uname -r)
  adding: module.dwarf (deflated 91%)
  adding: boot/System.map-4.18.0-15-generic (deflated 79%)
sansforensics@siftworkstation: ~/Downloads/volatility/tools/linux
$ cd /usr/local/lib/python2.7/dist-packages/volatility/plugins/overlays/linux/
sansforensics@siftworkstation: /usr/local/lib/python2.7/dist-packages/volatility/plugins/overlays/linux
$ ls
elf.py  elf.pyc  __init__.py  __init__.pyc  linux.py  linux.pyc  Ubuntu.zip
sansforensics@siftworkstation: /usr/local/lib/python2.7/dist-packages/volatility/plugins/overlays/linux
$
```

5. Run *vol.py --info | grep Profile* to make sure the profile "LinuxUbuntux64" is in the profiles list.

```
Terminal
File  Edit  View  Search  Terminal  Help

Profiles
--------
LinuxUbuntux64          - A Profile for Linux Ubuntu 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
Win10x64                - A Profile for Windows 10 x64
Win10x64_10240_17770    - A Profile for Windows 10 x64 (10.0.10240.17770 / 2018-02-10)
```

6. Run *vol.py -f '/home/sansforensics/Desktop/yourusername_memory_dump.bin' --profile=LinuxUbuntux64 VolatilityLinuxCommand* (Note: replace *VolatilityLinuxCommand* with the volatility Linux commands from https://github.com/volatilityfoundation/volatility/wiki/Linux-Command-Reference)

```
$ vol.py -f '/home/sansforensics/Desktop/yin_memory_dump.bin' --profile=LinuxUbuntux64 linux_pslist
Volatility Foundation Volatility Framework 2.6.1
Offset             Name                 Pid             PPid            Uid             Gid    DTB
          Start Time
------------------ -------------------- --------------- --------------- --------------- ------ --------
---------- ----------
0xffff9a37faea4500 systemd              1               0               0               0      0x000000
0079728000 2021-02-09 12:51:51 UTC+0000
0xffff9a37faea1700 kthreadd             2               0               0               0      --------
---------- 2021-02-09 12:51:51 UTC+0000
```

Task for part 2.

Show me 3-5 volatility Linux commands along with the plugins and the data you recovered from your SIFT memory.

- Four volatility Linux commands along with the plugins and the data I recovered from my SIFT memory.
    - Plugin 1:vol.py -f '/home/sansforensics/Desktop/miftahul_memory_dump.bin' --profile=LinuxUbuntux64 linux_pslist
        - using the plugin linux_pslist I was able to recovered the process and some of the information related to each of them

```
sansforensics@siftworkstation: ~
$ vol.py -f '/home/sansforensics/Desktop/miftahul_memory_dump.bin' --profile=LinuxUbuntux64 linux_pslist
Volatility Foundation Volatility Framework 2.6.1
Offset             Name                 Pid             PPid            Uid             Gid    DTB                 Start Time
------------------ -------------------- --------------- --------------- --------------- ------ ------------------ ----------
0xffff9dd87b1f1740 systemd              1               0               0               0      0x0000000139b24000 2024-03-13 10:10:30 UTC+0000
0xffff9dd87b1f5d00 kthreadd             2               0               0               0      ------------------ 2024-03-13 10:10:30 UTC+0000
0xffff9dd87b1f0000 rcu_gp               3               2               0               0      ------------------ 2024-03-13 10:10:30 UTC+0000
0xffff9dd87b1f2e80 rcu_par_gp           4               2               0               0      ------------------ 2024-03-13 10:10:30 UTC+0000
0xffff9dd87b208000 kworker/0:0H         6               2               0               0      ------------------ 2024-03-13 10:10:30 UTC+0000
0xffff9dd87b209740 mm_percpu_wq         9               2               0               0      ------------------ 2024-03-13 10:10:30 UTC+0000
0xffff9dd87b20dd00 ksoftirqd/0          10              2               0               0      ------------------ 2024-03-13 10:10:30 UTC+0000
0xffff9dd87b2145c0 rcu_sched            11              2               0               0      ------------------ 2024-03-13 10:10:30 UTC+0000
0xffff9dd87b211740 migration/0          12              2               0               0      ------------------ 2024-03-13 10:10:30 UTC+0000
0xffff9dd87b215d00 idle_inject/0        13              2               0               0      ------------------ 2024-03-13 10:10:30 UTC+0000
```

- Plugin 2: vol.py -f '/home/sansforensics/Desktop/miftahul_memory_dump.bin' --profile=LinuxUbuntux64 linux_netstat -U
  - using the linux_netstat plugin I was able to recover the network data similar to what would've shown if I ran the netstat command in a live linux system

```
sansforensics@siftworkstation: ~
$ vol.py -f '/home/sansforensics/Desktop/miftahul_memory_dump.bin' --profile=LinuxUbuntux64 linux_netstat -U
Volatility Foundation Volatility Framework 2.6.1
UDP      127.0.0.53      :   53 0.0.0.0         :    0                    systemd-resolve/451
TCP      127.0.0.53      :   53 0.0.0.0         :    0 LISTEN             systemd-resolve/451
UDP      0.0.0.0         : 5353 0.0.0.0         :    0                       avahi-daemon/591
UDP      ::              : 5353 ::              :    0                       avahi-daemon/591
UDP      0.0.0.0         :60883 0.0.0.0         :    0                       avahi-daemon/591
UDP      ::              :38613 ::              :    0                       avahi-daemon/591
UDP      192.168.194.99  :   68 192.168.207.254 :   67                    NetworkManager/594
UDP      0.0.0.0         : 2055 0.0.0.0         :    0                            nfcapd/694
TCP      0.0.0.0         :   22 0.0.0.0         :    0 LISTEN                       sshd/712
TCP      ::              :   22 ::              :    0 LISTEN                       sshd/712
UDP      0.0.0.0         :  137 0.0.0.0         :    0                              nmbd/783
UDP      0.0.0.0         :  138 0.0.0.0         :    0                              nmbd/783
UDP      192.168.194.99  :  137 0.0.0.0         :    0                              nmbd/783
UDP      192.168.207.255 :  137 0.0.0.0         :    0                              nmbd/783
UDP      192.168.194.99  :  138 0.0.0.0         :    0                              nmbd/783
```

- Plugin 3: vol.py -f '/home/sansforensics/Desktop/miftahul_memory_dump.bin' --profile=LinuxUbuntux64 linux_iomem
  - using the plugin linux_iomem I was able to recover the physical addresses currently reserved for IO devices.

```
sansforensics@siftworkstation: ~
$ vol.py -f '/home/sansforensics/Desktop/miftahul_memory_dump.bin' --profile=LinuxUbuntux64 linux_iomem
Volatility Foundation Volatility Framework 2.6.1
Reserved                                0x0                 0xFFF
System RAM                              0x1000              0x9F7FF
Reserved                                0x9F800             0x9FFFF
PCI Bus 0000:00                         0xA0000             0xBFFFF
Video ROM                               0xC0000             0xC7FFF
Adapter ROM                             0xCA000             0xCAFFF
PCI Bus 0000:00                         0xCC000             0xCFFFF
PCI Bus 0000:00                         0xD0000             0xD3FFF
PCI Bus 0000:00                         0xD4000             0xD7FFF
PCI Bus 0000:00                         0xD8000             0xDBFFF
Reserved                                0xDC000             0xFFFFF
  System ROM                            0xF0000             0xFFFFF
System RAM                              0x100000            0xBFEDFFFF
  Kernel code                           0x80400000          0x81200EB0
  Kernel data                           0x81200EB1          0x81C580BF
  Kernel bss                            0x81F27000          0x823FFFFF
ACPI Tables                             0xBFEE0000          0xBFEFEFFF
ACPI Non-volatile Storage               0xBFEFF000          0xBFEFFFFF
System RAM                              0xBFF00000          0xBFFFFFFF
PCI Bus 0000:00                         0xC0000000          0xFFBFFFFF
```

- Plugin 4: vol.py -f '/home/sansforensics/Desktop/miftahul_memory_dump.bin' --profile=LinuxUbuntux64 linux_volshell
  - I thought the plugin linux_volshell was really cool and with this plugin I was able to have an interactive shell in the linux memory image. Once you get a shell, you can use it to simply list processes, linux data structure, you can change into a specific process's context and then access the task_struct object, and etc.

```
$ vol.py -f '/home/sansforensics/Desktop/miftahul_memory_dump.bin' --profile=LinuxUbuntux64 linux_volshell
Volatility Foundation Volatility Framework 2.6.1
Current context: process systemd, pid=1 DTB=0x139b24000
Welcome to volshell! Current memory image is:
file:///home/sansforensics/Desktop/miftahul_memory_dump.bin
To get help, type 'hh()'
>>> ps()
Name             PID    Offset
systemd          1      0xffff9dd87b1f1740
kthreadd         2      0xffff9dd87b1f5d00
rcu_gp           3      0xffff9dd87b1f0000
rcu_par_gp       4      0xffff9dd87b1f2e80
kworker/0:0H     6      0xffff9dd87b208000
mm_percpu_wq     9      0xffff9dd87b209740
ksoftirqd/0      10     0xffff9dd87b20dd00
rcu_sched        11     0xffff9dd87b2145c0
```

Note: If your SIFT memory dump from Homework 1 does not work with Volatility. You should try to use LiME again to dump out your SIFT memory.

References for part 2:

[1] https://www.andreafortuna.org/2019/08/22/how-to-generate-a-volatility-profile-for-a-linux-system/

[2] Linux memory acquisition and analysis, https://opensource.com/article/21/4/linux-memory-forensics

Part 3. Windows Memory Analysis Using Volatility3 (30 points)

Volatility3 is released in 2021. The goal of this part is to familiarize yourself with this new version.

Volatility3 is a complete rewrite of Volatility2 to address many of the technical and performance challenges. Volatility3 does not require an OS profile to start your analysis.

Steps to build Volatility3:

1. Follow the instructions in https://github.com/volatilityfoundation/volatility3 to install volatility3.

- Check your Python version (*python3 --version*) to make sure you have Python 3.6 or later
- *Clone the latest version of Volatility from GitHub (I cloned volatility3 on the SIFT Desktop):* git clone https://github.com/volatilityfoundation/volatility3.git
- *cd volatility3* and install Pefile 2017.8.1 or later: *sudo pip3 install -r requirements-minimal.txt (***Note:** you may have to install pip3 first by *sudo apt update; sudo apt install python3-pip)*
- Run vol.py in **your Volatility3 directory**, and check python3 is install correctly: *python3 vol.py -h*

```
sansforensics@siftworkstation: ~/Desktop/volatility3
$ python3 vol.py -h
Volatility 3 Framework 2.3.0
usage: volatility [-h] [-c CONFIG] [--parallelism [{processes,threads,off}]] [-e EXTEND]
                  [-p PLUGIN_DIRS] [-s SYMBOL_DIRS] [-v] [-l LOG] [-o OUTPUT_DIR] [-q]
                  [-r RENDERER] [-f FILE] [--write-config] [--save-config SAVE_CONFIG]
                  [--clear-cache] [--cache-path CACHE_PATH] [--offline]
                  [--single-location SINGLE_LOCATION]
                  [--stackers [STACKERS [STACKERS ...]]]
                  [--single-swap-locations [SINGLE_SWAP_LOCATIONS [SINGLE_SWAP_LOCATIONS ...
```

2. *Symbol table* are used for various operating systems. You have to add the windows symbols for volatility3 to work.
   - Download the windows symbols

     https://downloads.volatilityfoundation.org/volatility3/symbols/windows.zip
   - Extract windows.zip and replace the windows dir in *volatility3/volatility3/symbols/* with the new windows dir extracted from windows.zip

```
sansforensics@siftworkstation: ~/Desktop/volatility3/volatility3/symbols
$ ls
__init__.py  __pycache__  windows
```

3. Volatility3 should work now
   - List all the plugins in Volatility3, you run *pythoy3 vol.py -h* (in **your Volatility3 directory)**

```
sansforensics@siftworkstation: ~/Desktop/volatility3
$ python3 vol.py -h
Volatility 3 Framework 2.3.0
usage: volatility [-h] [-c CONFIG] [--parallelism [{processes,threads,off}]] [-e EXTEND]
                  [-p PLUGIN_DIRS] [-s SYMBOL_DIRS] [-v] [-l LOG] [-o OUTPUT_DIR] [-q]
                  [-r RENDERER] [-f FILE] [--write-config] [--save-config SAVE_CONFIG]
                  [--clear-cache] [--cache-path CACHE_PATH] [--offline]
                  [--single-location SINGLE_LOCATION]
                  [--stackers [STACKERS [STACKERS ...]]]
                  [--single-swap-locations [SINGLE_SWAP_LOCATIONS [SINGLE_SWAP_LOCATIONS ...
```

Tasks for part 3:

We will use Volatility3 to analyze *zeus.vmem*. Try to run the following volatility3 Windows plugins. For **each plugin**, show a top portion of the screenshot including the command you run (see an example below) to prove that Volatility3 is working. Feel free to explore other Volatility3 plugins.

Note: When you run volatility 3, PDB initial scanning takes a while. Please be patient.

**** volatility3 Windows plugins ****

windows.pslist.PsList,

windows.psscan.PsScan,

windows.registry.userassist.UserAssist,

windows.registry.hivelist.HiveList,

windows.registry.hivescan.HiveScan,
windows.registry.printkey.PrintKey,
windows.cmdline.CmdLine,
windows.getsids.GetSIDs,
windows.svcscan.SvcScan.

```
sansforensics@siftworkstation: ~/Desktop/volatility3
$ python3 vol.py -f '/home/sansforensics/Desktop/Images/zeus.vmem' windows.pslist.PsList
Volatility 3 Framework 2.3.0
Progress:  100.00               PDB scanning finished
PID     PPID    ImageFileName   Offset(V)       Threads Handles SessionId       Wow64   CreateTime  ExitTime
        File output

4       0       System  0x810b1660      58      379     N/A     False   N/A     N/A     Disabled
544     4       smss.exe        0xff2ab020      3       21      N/A     False   2010-08-11 06:06:21.000000  N/A
     Disabled
608     544     csrss.exe       0xff1ecda0      10      410     0       False   2010-08-11 06:06:23.000000  N/A
     Disabled
632     544     winlogon.exe    0xff1ec978      24      536     0       False   2010-08-11 06:06:23.000000  N/A
     Disabled
676     632     services.exe    0xff247020      16      288     0       False   2010-08-11 06:06:24.000000  N/A
     Disabled
688     632     lsass.exe       0xff255020      21      405     0       False   2010-08-11 06:06:24.000000  N/A
     Disabled
844     676     vmacthlp.exe    0xff218230      1       37      0       False   2010-08-11 06:06:24.000000  N/A
     Disabled
856     676     svchost.exe     0x80ff88d8      29      336     0       False   2010-08-11 06:06:24.000000  N/A
     Disabled
```

Screenshots of each of the volatility windows plugins their output:

- windows.pslist.PsList

```
sansforensics@siftworkstation: ~/homework2/volatility3
$ python3 vol.py -f '/home/sansforensics/homework2/zeus.vmem' windows.pslist.PsList
Volatility 3 Framework 2.7.0
WARNING  volatility3.framework.layers.vmware: No metadata file found alongside VMEM file. A VMSS or VMSN file may be required to correctly pro
cess a VMEM file. These should be placed in the same directory with the same file name, e.g. zeus.vmem and zeus.vmss.
Progress:  100.00               PDB scanning finished
PID     PPID    ImageFileName   Offset(V)       Threads Handles SessionId       Wow64   CreateTime      ExitTime        File output

4       0       System  0x810b1660      58      379     N/A     False   N/A     N/A     Disabled
544     4       smss.exe        0xff2ab020      3       21      N/A     False   2010-08-11 06:06:21.000000      N/A     Disabled
608     544     csrss.exe       0xff1ecda0      10      410     0       False   2010-08-11 06:06:23.000000      N/A     Disabled
632     544     winlogon.exe    0xff1ec978      24      536     0       False   2010-08-11 06:06:23.000000      N/A     Disabled
676     632     services.exe    0xff247020      16      288     0       False   2010-08-11 06:06:24.000000      N/A     Disabled
688     632     lsass.exe       0xff255020      21      405     0       False   2010-08-11 06:06:24.000000      N/A     Disabled
844     676     vmacthlp.exe    0xff218230      1       37      0       False   2010-08-11 06:06:24.000000      N/A     Disabled
856     676     svchost.exe     0x80ff88d8      29      336     0       False   2010-08-11 06:06:24.000000      N/A     Disabled
936     676     svchost.exe     0xff217560      11      288     0       False   2010-08-11 06:06:24.000000      N/A     Disabled
1028    676     svchost.exe     0x80fbf910      88      1424    0       False   2010-08-11 06:06:24.000000      N/A     Disabled
1088    676     svchost.exe     0xff22d558      7       93      0       False   2010-08-11 06:06:25.000000      N/A     Disabled
1148    676     svchost.exe     0xff203b80      15      217     0       False   2010-08-11 06:06:26.000000      N/A     Disabled
1432    676     spoolsv.exe     0xff1d7da0      14      145     0       False   2010-08-11 06:06:26.000000      N/A     Disabled
1668    676     vmtoolsd.exe    0xff1b8b28      5       225     0       False   2010-08-11 06:06:35.000000      N/A     Disabled
1788    676     VMUpgradeHelper 0xff1fdc88      5       112     0       False   2010-08-11 06:06:38.000000      N/A     Disabled
```

- windows.psscan.PsScan

```
sansforensics@siftworkstation: ~/homework2/volatility3
$ python3 vol.py -f '/home/sansforensics/homework2/zeus.vmem' windows.psscan.PsScan
Volatility 3 Framework 2.7.0
WARNING  volatility3.framework.layers.vmware: No metadata file found alongside VMEM file. A VMSS or VMSN file may be required to correctly pro
cess a VMEM file. These should be placed in the same directory with the same file name, e.g. zeus.vmem and zeus.vmss.
Progress:  100.00               PDB scanning finished
PID     PPID    ImageFileName   Offset(V)       Threads Handles SessionId       Wow64   CreateTime      ExitTime        File output

1732    1028    wuauclt.exe     0x10c3da0       7       189     0       False   2010-08-11 06:07:44.000000      N/A     Disabled
468     1028    wuauclt.exe     0x10f7588       4       142     0       False   2010-08-11 06:09:37.000000      N/A     Disabled
1028    676     svchost.exe     0x1122910       88      1424    0       False   2010-08-11 06:06:24.000000      N/A     Disabled
856     676     svchost.exe     0x115b8d8       29      336     0       False   2010-08-11 06:06:24.000000      N/A     Disabled
4       0       System  0x1214660       58      379     N/A     False   N/A     N/A     Disabled
1968    676     TPAutoConnSvc.e 0x211ab28       5       106     0       False   2010-08-11 06:06:39.000000      N/A     Disabled
1084    1968    TPAutoConnect.e 0x49c15f8       1       68      0       False   2010-08-11 06:06:52.000000      N/A     Disabled
1724    1708    explorer.exe    0x4a065d0       13      326     0       False   2010-08-11 06:09:29.000000      N/A     Disabled
452     1724    VMwareUser.exe  0x4b5a980       8       207     0       False   2010-08-11 06:09:32.000000      N/A     Disabled
432     1724    VMwareTray.exe  0x4be97e8       1       60      0       False   2010-08-11 06:09:31.000000      N/A     Disabled
888     1028    wscntfy.exe     0x4c2b310       1       40      0       False   2010-08-11 06:06:49.000000      N/A     Disabled
544     4       smss.exe        0x5471020       3       21      N/A     False   2010-08-11 06:06:21.000000      N/A     Disabled
1699873240      1182038117      tCharacterPlacem        0x55b47fc       1953394502      -       -       False   2022-11-24 22:52:14.000000   -
Disabled
216     676     alg.exe 0x5f027e0       8       120     0       False   2010-08-11 06:06:39.000000      N/A     Disabled
688     632     lsass.exe       0x5f47020       21      405     0       False   2010-08-11 06:06:24.000000      N/A     Disabled
676     632     services.exe    0x6015020       16      288     0       False   2010-08-11 06:06:24.000000      N/A     Disabled
```

- windows.registry.userassist.UserAssist

```
sansforensics@siftworkstation: ~/homework2/volatility3
$ python3 vol.py -f '/home/sansforensics/homework2/zeus.vmem' windows.registry.userassist.UserAssist
Volatility 3 Framework 2.7.0
WARNING  volatility3.framework.layers.vmware: No metadata file found alongside VMEM file. A VMSS or VMSN file may be required to correctly pro
cess a VMEM file. These should be placed in the same directory with the same file name, e.g. zeus.vmem and zeus.vmss.
Progress:  100.00               PDB scanning finished
Hive Offset     Hive Name       Path    Last Write Time Type    Name    ID      Count   Focus Count     Time Focused    Last Updated    Raw Da
ta

0xe1da4008      \Device\HarddiskVolume1\Documents and Settings\Administrator\NTUSER.DAT NTUSER.DAT\Software\Microsoft\Windows\CurrentVersion\E
xplorer\UserAssist\{5E6AB780-7743-11CF-A12B-00AA004AE837}\Count 2010-06-10 16:11:44.000000      Key     N/A     N/A     N/A     N/A     N/A  N
/A      N/A
* 0xe1da4008    \Device\HarddiskVolume1\Documents and Settings\Administrator\NTUSER.DAT NTUSER.DAT\Software\Microsoft\Windows\CurrentVersion\E
xplorer\UserAssist\{5E6AB780-7743-11CF-A12B-00AA004AE837}\Count 2010-06-10 16:11:44.000000      Value   UEME_CTLSESSION -       -       -    -
-
00 00 00 00 00 00 00 00 ........
0xe1da4008      \Device\HarddiskVolume1\Documents and Settings\Administrator\NTUSER.DAT NTUSER.DAT\Software\Microsoft\Windows\CurrentVersion\E
xplorer\UserAssist\{75048700-EF1F-11D0-9888-006097DEACF9}\Count 2010-08-15 19:17:23.000000      Key     N/A     N/A     N/A     N/A     N/A  N
/A      N/A
* 0xe1da4008    \Device\HarddiskVolume1\Documents and Settings\Administrator\NTUSER.DAT NTUSER.DAT\Software\Microsoft\Windows\CurrentVersion\E
xplorer\UserAssist\{75048700-EF1F-11D0-9888-006097DEACF9}\Count 2010-08-15 19:17:23.000000      Value   UEME_CTLSESSION -       -       -    -
-
d7 c8 59 0e 02 00 00 00 ..Y.....
* 0xe1da4008    \Device\HarddiskVolume1\Documents and Settings\Administrator\NTUSER.DAT NTUSER.DAT\Software\Microsoft\Windows\CurrentVersion\E
xplorer\UserAssist\{75048700-EF1F-11D0-9888-006097DEACF9}\Count 2010-08-15 19:17:23.000000      Value   UEME_RUNPIDL:%csidl2%\MSN.lnk   1    1
4       N/A     N/A     2010-06-10 16:10:27.000000
01 00 00 00 13 00 00 00
```

- windows.registry.hivelist.HiveList

```
sansforensics@siftworkstation: ~/homework2/volatility3
$ python3 vol.py -f '/home/sansforensics/homework2/zeus.vmem' windows.registry.hivelist.HiveList
Volatility 3 Framework 2.7.0
WARNING  volatility3.framework.layers.vmware: No metadata file found alongside VMEM file. A VMSS or VMSN file may be required to correctly pro
cess a VMEM file. These should be placed in the same directory with the same file name, e.g. zeus.vmem and zeus.vmss.
Progress:  100.00               PDB scanning finished
Offset  FileFullPath    File output

0xe1e158c0      \Device\HarddiskVolume1\Documents and Settings\Administrator\Local Settings\Application Data\Microsoft\Windows\UsrClass.dat  D
isabled
0xe1da4008      \Device\HarddiskVolume1\Documents and Settings\Administrator\NTUSER.DAT Disabled
0xe1c49008      \Device\HarddiskVolume1\Documents and Settings\LocalService\Local Settings\Application Data\Microsoft\Windows\UsrClass.dat   D
isabled
0xe1c41b60      \Device\HarddiskVolume1\Documents and Settings\LocalService\NTUSER.DAT  Disabled
0xe1a39638      \Device\HarddiskVolume1\Documents and Settings\NetworkService\Local Settings\Application Data\Microsoft\Windows\UsrClass.dat D
isabled
0xe1a33008      \Device\HarddiskVolume1\Documents and Settings\NetworkService\NTUSER.DAT        Disabled
0xe153ab60      \Device\HarddiskVolume1\WINDOWS\system32\config\software        Disabled
0xe1542008      \Device\HarddiskVolume1\WINDOWS\system32\config\default Disabled
0xe1537b60      \SystemRoot\System32\Config\SECURITY    Disabled
0xe1544008      \Device\HarddiskVolume1\WINDOWS\system32\config\SAM     Disabled
0xe13ae580              Disabled
0xe101b008      \Device\HarddiskVolume1\WINDOWS\system32\config\system  Disabled
0xe1008978              Disabled
sansforensics@siftworkstation: ~/homework2/volatility3
```

- windows.registry.hivescan.HiveScan

```
sansforensics@siftworkstation: ~/homework2/volatility3
$ python3 vol.py -f '/home/sansforensics/homework2/zeus.vmem' windows.registry.hivescan.HiveScan
Volatility 3 Framework 2.7.0
WARNING  volatility3.framework.layers.vmware: No metadata file found alongside VMEM file. A VMSS or VMSN file may be required to correctly pro
cess a VMEM file. These should be placed in the same directory with the same file name, e.g. zeus.vmem and zeus.vmss.
Progress:  100.00               PDB scanning finished
Offset

0x9728c0
0xf6e008
0x1824978
0x1867008
0x1bbd580
0x1f98008
0x21eb638
0x36dc008
0x4010b60
0x6ae4b60
0x6b7db60
0x6c48008
0x6c4b008
sansforensics@siftworkstation: ~/homework2/volatility3
```

- windows.registry.printkey.PrintKey

```
sansforensics@siftworkstation: ~/homework2/volatility3
$ python3 vol.py -f '/home/sansforensics/homework2/zeus.vmem' windows.registry.printkey.PrintKey
Volatility 3 Framework 2.7.0
WARNING  volatility3.framework.layers.vmware: No metadata file found alongside VMEM file. A VMSS or VMSN file may be required to correctly pro
cess a VMEM file. These should be placed in the same directory with the same file name, e.g. zeus.vmem and zeus.vmss.
Progress:  100.00               PDB scanning finished
Last Write Time Hive Offset     Type    Key     Name    Data    Volatile

2010-06-10 16:12:08.000000      0xe1e158c0      Key     \Device\HarddiskVolume1\Documents and Settings\Administrator\Local Settings\Applicatio
n Data\Microsoft\Windows\UsrClass.dat   Software                False
2010-06-10 16:11:42.000000      0xe1da4008      Key     \Device\HarddiskVolume1\Documents and Settings\Administrator\NTUSER.DAT AppEvents    F
alse
2010-06-10 16:11:42.000000      0xe1da4008      Key     \Device\HarddiskVolume1\Documents and Settings\Administrator\NTUSER.DAT Console      F
alse
2010-06-10 16:13:03.000000      0xe1da4008      Key     \Device\HarddiskVolume1\Documents and Settings\Administrator\NTUSER.DAT Control PanelF
alse
2010-06-10 16:11:42.000000      0xe1da4008      Key     \Device\HarddiskVolume1\Documents and Settings\Administrator\NTUSER.DAT Environment  F
alse
2010-06-10 16:12:06.000000      0xe1da4008      Key     \Device\HarddiskVolume1\Documents and Settings\Administrator\NTUSER.DAT Identities   F
alse
2010-06-10 16:11:42.000000      0xe1da4008      Key     \Device\HarddiskVolume1\Documents and Settings\Administrator\NTUSER.DAT Keyboard Layou
t               False
2010-06-10 16:17:08.000000      0xe1da4008      Key     \Device\HarddiskVolume1\Documents and Settings\Administrator\NTUSER.DAT Printers     F
alse
2010-08-11 06:06:48.000000      0xe1da4008      Key     \Device\HarddiskVolume1\Documents and Settings\Administrator\NTUSER.DAT Software     F
alse
```

- windows.cmdline.CmdLine

```
sansforensics@siftworkstation: ~/homework2/volatility3
$ python3 vol.py -f '/home/sansforensics/homework2/zeus.vmem' windows.cmdline.CmdLine
Volatility 3 Framework 2.7.0
WARNING  volatility3.framework.layers.vmware: No metadata file found alongside VMEM file. A VMSS or VMSN file may be required to correctly pro
cess a VMEM file. These should be placed in the same directory with the same file name, e.g. zeus.vmem and zeus.vmss.
Progress:  100.00               PDB scanning finished
PID     Process Args

4       System  Required memory at 0x10 is not valid (process exited?)
544     smss.exe        \SystemRoot\System32\smss.exe
608     csrss.exe       C:\WINDOWS\system32\csrss.exe ObjectDirectory=\Windows SharedSection=1024,3072,512 Windows=On SubSystemType=Windows Se
rverDll=basesrv,1 ServerDll=winsrv:UserServerDllInitialization,3 ServerDll=winsrv:ConServerDllInitialization,2 ProfileControl=Off MaxRequestTh
reads=16
632     winlogon.exe    winlogon.exe
676     services.exe    C:\WINDOWS\system32\services.exe
688     lsass.exe       C:\WINDOWS\system32\lsass.exe
844     vmacthlp.exe    "C:\Program Files\VMware\VMware Tools\vmacthlp.exe"
856     svchost.exe     C:\WINDOWS\system32\svchost -k DcomLaunch
936     svchost.exe     C:\WINDOWS\system32\svchost -k rpcss
1028    svchost.exe     C:\WINDOWS\System32\svchost.exe -k netsvcs
1088    svchost.exe     C:\WINDOWS\system32\svchost.exe -k NetworkService
1148    svchost.exe     C:\WINDOWS\system32\svchost.exe -k LocalService
1432    spoolsv.exe     C:\WINDOWS\system32\spoolsv.exe
1668    vmtoolsd.exe    "C:\Program Files\VMware\VMware Tools\vmtoolsd.exe"
1788    VMUpgradeHelper "C:\Program Files\VMware\VMware Tools\VMUpgradeHelper.exe" /service
```

- windows.getsids.GetSIDs

```
sansforensics@siftworkstation: ~/homework2/volatility3
$ python3 vol.py -f '/home/sansforensics/homework2/zeus.vmem' windows.getsids.GetSIDs
Volatility 3 Framework 2.7.0
WARNING  volatility3.framework.layers.vmware: No metadata file found alongside VMEM file. A VMSS or VMSN file may be required to correctly pro
cess a VMEM file. These should be placed in the same directory with the same file name, e.g. zeus.vmem and zeus.vmss.
Progress:  100.00               PDB scanning finished
PID     Process SID     Name

4       System  S-1-5-18        Local System
4       System  S-1-5-32-544    Administrators
4       System  S-1-1-0 Everyone
4       System  S-1-5-11        Authenticated Users
544     smss.exe        S-1-5-18        Local System
544     smss.exe        S-1-5-32-544    Administrators
544     smss.exe        S-1-1-0 Everyone
544     smss.exe        S-1-5-11        Authenticated Users
608     csrss.exe       S-1-5-18        Local System
608     csrss.exe       S-1-5-32-544    Administrators
608     csrss.exe       S-1-1-0 Everyone
608     csrss.exe       S-1-5-11        Authenticated Users
632     winlogon.exe    S-1-5-18        Local System
632     winlogon.exe    S-1-5-32-544    Administrators
632     winlogon.exe    S-1-1-0 Everyone
632     winlogon.exe    S-1-5-11        Authenticated Users
676     services.exe    S-1-5-18        Local System
676     services.exe    S-1-5-32-544    Administrators
```

- windows.svcscan.SvcScan

```
sansforensics@siftworkstation: ~/homework2/volatility3
$ python3 vol.py -f '/home/sansforensics/homework2/zeus.vmem' windows.svcscan.SvcScan
Volatility 3 Framework 2.7.0
WARNING  volatility3.framework.layers.vmware: No metadata file found alongside VMEM file. A VMSS or VMSN file may be required to correctly pro
cess a VMEM file. These should be placed in the same directory with the same file name, e.g. zeus.vmem and zeus.vmss.
Progress:  100.00               PDB scanning finished
Offset  Order   PID     Start   State   Type    Name    Display Binary  Binary (Registry)       Dll

0x6e1e90        1       N/A     SERVICE_DISABLED        SERVICE_STOPPED SERVICE_KERNEL_DRIVER   Abiosdsk        Abiosdsk        N/A     -    -
0x6e1f20        2       N/A     SERVICE_DISABLED        SERVICE_STOPPED SERVICE_KERNEL_DRIVER   abp480n5        abp480n5        N/A     -    -
0x6e1fb0        3       N/A     SERVICE_BOOT_START      SERVICE_RUNNING SERVICE_KERNEL_DRIVER   ACPI    Microsoft ACPI Driver   \Driver\ACPI -
-
0x6e2038        4       N/A     SERVICE_DISABLED        SERVICE_STOPPED SERVICE_KERNEL_DRIVER   ACPIEC  ACPIEC  N/A     -       -
0x6e20c8        5       N/A     SERVICE_DISABLED        SERVICE_STOPPED SERVICE_KERNEL_DRIVER   adpu160m        adpu160m        N/A     -    -
0x6e2158        6       N/A     SERVICE_DEMAND_START    SERVICE_STOPPED SERVICE_KERNEL_DRIVER   aec     Microsoft Kernel Acoustic Echo Cancell
er      N/A     -       -
0x6e21e0        7       N/A     SERVICE_SYSTEM_START    SERVICE_RUNNING SERVICE_KERNEL_DRIVER   AFD     AFD     \Driver\AFD     -       -
0x6e2268        8       N/A     SERVICE_BOOT_START      SERVICE_RUNNING SERVICE_KERNEL_DRIVER   agp440  Intel AGP Bus Filter    \Driver\agp440
-       -
0x6e22f8        9       N/A     SERVICE_DISABLED        SERVICE_STOPPED SERVICE_KERNEL_DRIVER   Aha154x Aha154x N/A     -       -
0x6e2388        10      N/A     SERVICE_DISABLED        SERVICE_STOPPED SERVICE_KERNEL_DRIVER   aic78u2 aic78u2 N/A     -       -
0x6e2418        11      N/A     SERVICE_DISABLED        SERVICE_STOPPED SERVICE_KERNEL_DRIVER   aic78xx aic78xx N/A     -       -
0x6e24a8        12      N/A     SERVICE_DISABLED        SERVICE_STOPPED SERVICE_WIN32_SHARE_PROCESS     Alerter Alerter N/A     -       -
0x6e2538        13      216     SERVICE_DEMAND_START    SERVICE_RUNNING SERVICE_WIN32_OWN_PROCESS       ALG     Application Layer Gateway Serv
ice     C:\WINDOWS\System32\alg.exe     -       -
0x6e25c0        14      N/A     SERVICE_DISABLED        SERVICE_STOPPED SERVICE_KERNEL_DRIVER   AliIde  AliIde  N/A     -       -
0x6e2650        15      N/A     SERVICE_DISABLED        SERVICE_STOPPED SERVICE_KERNEL_DRIVER   amsint  amsint  N/A     -       -
```

Bonus Part 4. Linux-Memory Analysis Using Volatility3 (30 bonus points)

Part 4 task:

As we know, Volatility3 does not require an OS profile to start your analysis. However, building the appropriate symbol table for a Linux image is still very challenging and there is not much public information in this topic at this point.

My students from the 2211 Advanced Forensics Course successfully built the Linux symbol for our SIFT VM. If you are interested in Linux memory analysis using Volatility 3, please read the "*Volatility 3 for Linux Memory Analysis.pdf*" posted on myCourses > Project and Homework > Homework 2.