Windows Memory Analysis

Thomas Benos

MSc UniWA, Cisco CyberOps
GR CSIRT Incident Responder
thomasbenos1291@gmail.com

Agenda

- Introduction to Memory Analysis
- Memory Acquisition
- Key Memory Components
- Volatility Framework
- Windows Memory Analysis

Introduction to Memory Analysis

What is Memory Forensics?

- **Definition:**
  - The process of analyzing a computer's RAM (Random Access Memory) to uncover digital evidence related to security incidents, cybercrimes, or forensic investigations.
  - Unlike disk forensics, which examines stored data on physical media, memory forensics focuses on volatile data that resides in the system's active memory.
  - RAM holds real-time and volatile information about running processes, open network connections, and active user sessions.
  - Memory forensics allows investigators to retrieve valuable insights into recent system activities, malware presence, and unauthorized access.

Memory forensics role in malware analysis

Memory Forensics Steps

- **Memory Acquisition**
  - Dumping memory from the target machine
    - DumpIt
    - FTK Imager
    - Winpmem
- **Memory Analysis**
  - Analyzing the memory file
    - Volatility Framework

Memory Acquisition

Pre-Acquisition Process

- DO NOT POWER OFF
- Acquisition tools should be executed from an external device
- The memory dump must be stored in an external drive

Memory Acquisition Process

- Determine the state of the machine
- Identify the operating system
- Insert acquisition media & perform the Volatile Memory Dump
- Hash and verify the acquired files
- Create Investigator copies
- Always work on a copy

Other Memory Locations in Windows Systems

- **Pagefile.sys:**
  - **Location:** %SystemDrive%\pagefile.sys
  - **Purpose:** Acts as virtual memory, extending the available RAM by using a portion of the hard drive.
  - **Size:** Configurable by the user or automatically managed by the system.
  - **Content:** Holds data that doesn't fit in physical RAM, transferring less frequently used memory pages.
  - **Impact on Forensics:** Contains artifacts reflecting the system's memory usage and may store sensitive information.

- **Hiberfil.sys:**
  - **Location:** Typically found in the root directory of the system drive (%SystemDrive%/hiberfil.sys).
  - **Purpose:** Used for hibernation, storing the system's state when it enters hibernation mode.
  - **Size:** Approximately equal to the amount of physical RAM on the system.
  - **Content:** Contains the contents of the system's memory when hibernation is initiated.
  - **Impact on Forensics:** Valuable for memory analysis as it preserves the system's state.

Hybrid sleep, Hibernation, Sleep

- **Hybrid Sleep:**
  - Combines benefits of sleep and hibernation.
  - Saves RAM content to a hibernation file during low-power transition.
  - Fast resume if connected to power; hibernation file preserves data during power loss.
- **Hibernation:**
  - Saves RAM content to a hibernation file.
  - Lowest low-power state; longer resume time than regular sleep.
  - Hibernation file includes on-the-fly encryption keys for crypto containers and encrypted VM apps.
- **Sleep:**
  - Minimal power consumption.
  - Fast startup upon waking.
  - Instantly resumes where you left off.
  - Automatic work-saving feature.
  - Protects against battery drain by shutting down when low.

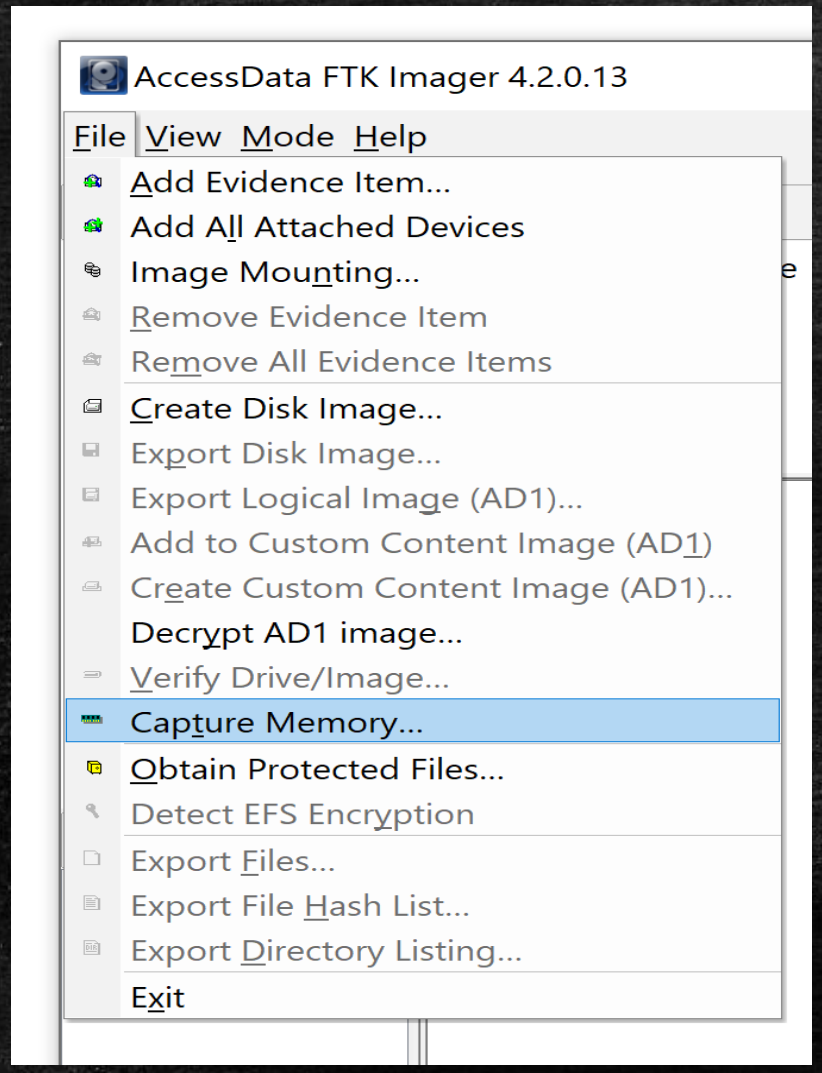
Shutdown vs Restart

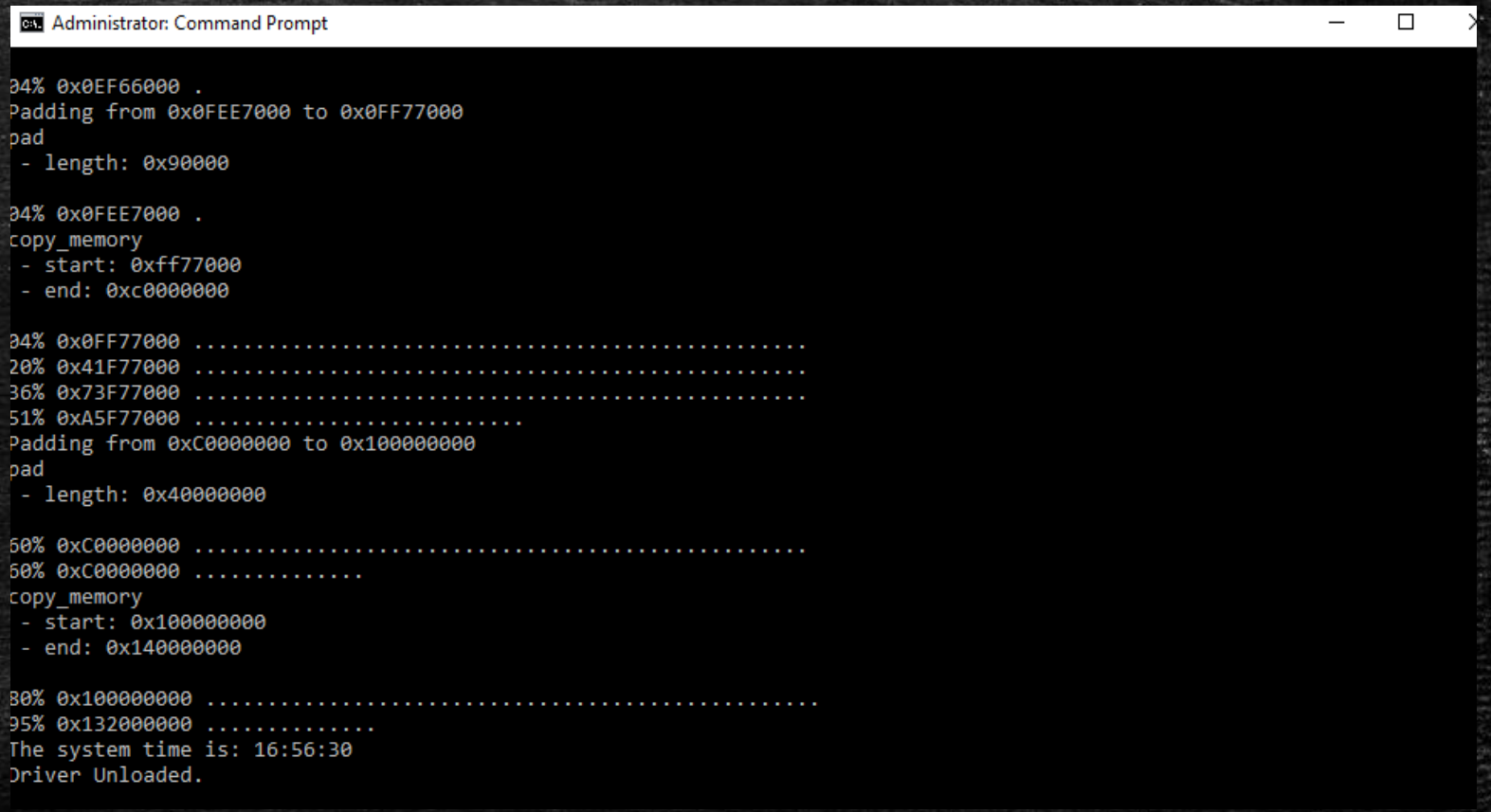
- **Shutdown:**
  - NOT really a shutdown
  - State of RAM is written to hard disk and then read back into ram when system starts
  - Hibernation-like
- **Restart:**
  - **Clears RAM**
  - Does not read state of windows when system restarts
  - Windows 8+ take longer to restart
- **Fast startup:**
  - Windows 10+
  - Reduced time to boot after gracious shut down
  - Saves a small hibernation file onto the disk – kernel memory

Memory Dump Acquisition Tools

- **Winpmem:**
  - **Description**: Open-source memory acquisition tool for Windows systems.
- **Belkasoft Live RAM Capturer:**
  - **Description**: A tool for capturing the content of the computer's volatile memory (RAM) and saving it to a file.
- **DumpIt:**
  - **Description**: A compact utility to acquire physical memory (RAM) on a Windows system.
- **FTK Imager:**
  - **Description**: A powerful and easy-to-use tool for acquiring physical memory or creating disk images.
- **Magnet RAM Capture:**
  - **Description**: A free utility to capture the physical memory of a suspect's computer into a memory dump file.

Memory Dump Acquisition Tools





Memory Acquisition from Virtual Machine

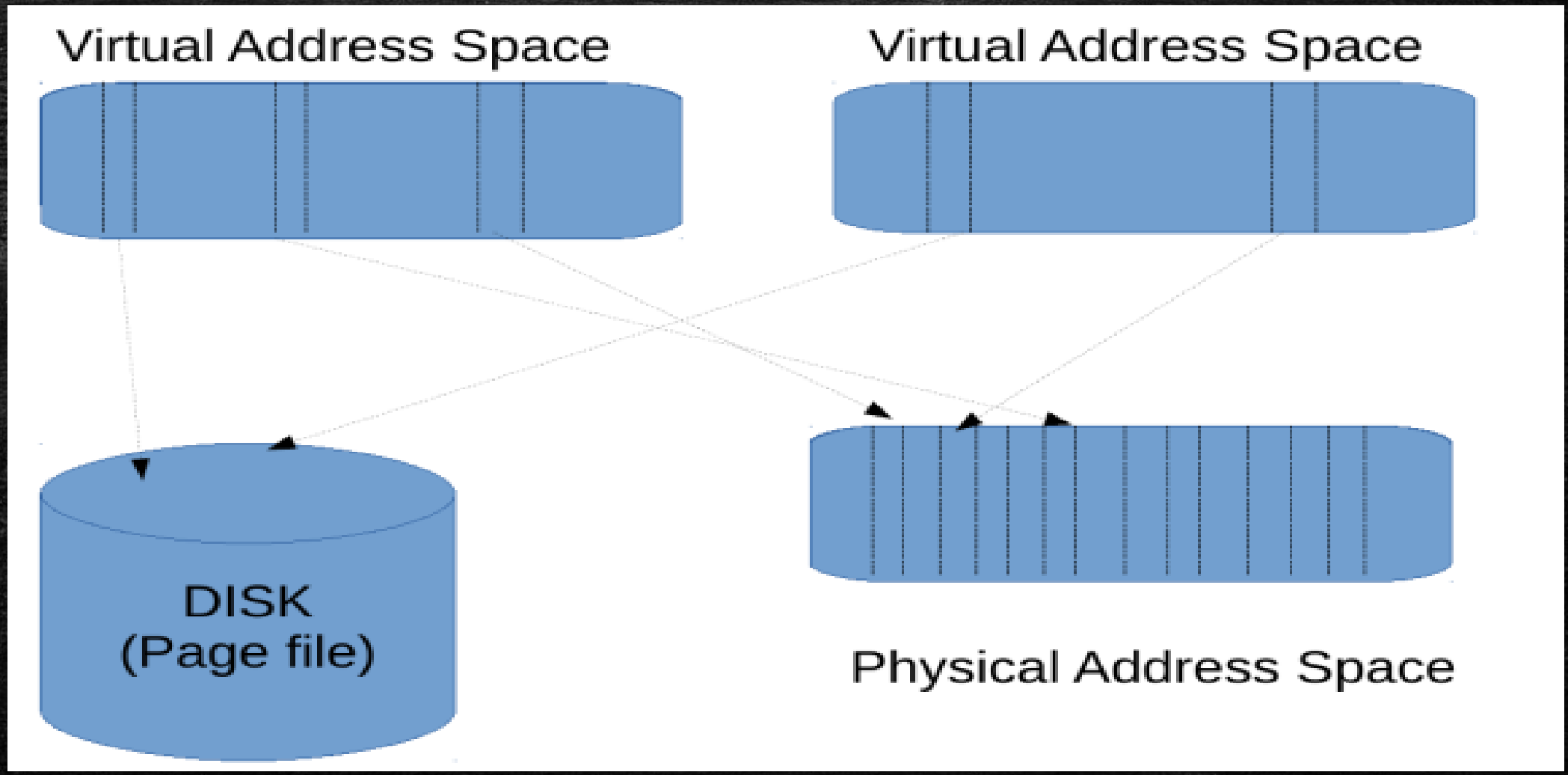
- **Suspend** the VM and copy the following files:
  - VMware (.vmdk)
    - .vmem file = raw memory image
    - .vmss = vmware saved state (suspend VM)
    - .vmsn = vmware snapshot file
  - Microsoft Hyper-V (.vhdx)
    - .bin file = raw memory image
- VirtualBox – **Don't suspend**
  - .sav file = machine state from snapshot
    - Ability to acquire guest memory using built-in VBoxManage.exe
      - vboxmanage debugvm <machinename> dumpvmcore → filename guest.dump

Digital Evidence in memory

- Process Artifacts
- Running services
- Clipboard Information
- Browsing Sessions
- Passwords
- Network Artifacts
- Accessed Files and other media
- Registry Artifacts
- User Activity
- Chats/Running Application stored data

Key Memory Components

Virtual & Physical Address Spaces



Volatility Framework

Volatility Framework

- **Purpose:**
  - Extracting – analyzing information from memory dumps
  - Uncovers insights related to processes, network connections, registry hives, and more.
- **Supported Platforms:**
  - Windows (XP to Windows 10/11), Linux, macOS.
- **Key Features:**
  - **Plugin Architecture**
  - **Cross-Platform Support**
  - **Profile-Based Analysis**
  - **Wide Range of Artifacts**

Volatility Framework Plugins (1/3)

- **imageinfo**
  - Identifies the appropriate profile for the image given
  - python3 vol.py –f <image_file> imageinfo

- Command example:
  - python3 vol.py –f <image_file> --profile=<profile> <plugin>
  - *python3 vol.py –f image.bin --profile=WinXPSP2x86 psxview*

- Volatility 2.6 - Either vol.py or standalone (.exe)
  - python3 vol.py
  - .\volatility_2.6_win64_standalone.exe

Volatility Framework Plugins (2/3)

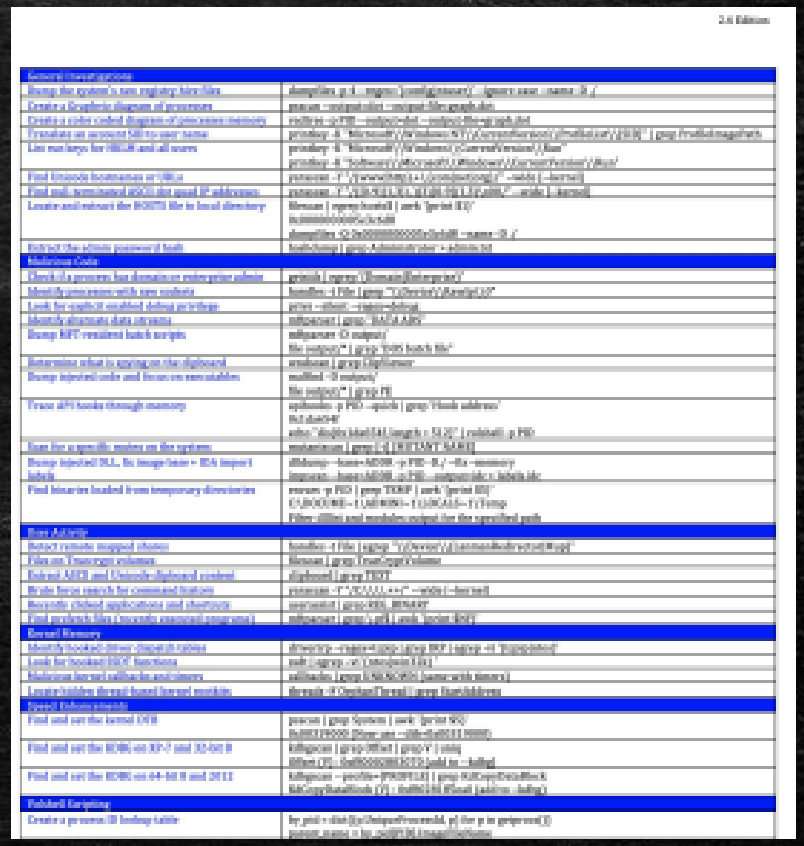
- **pslist - Process List:**
  - Displays a list of all running processes, including their Process ID (PID), parent PID, and other relevant information.
  - Command: "volatility -f <memory_dump> --profile=<profile> pslist"
- **pstree - Process Tree:**
  - Shows the hierarchical relationship between processes, including parent and child processes.
  - Command: "volatility -f <memory_dump> --profile=<profile> pstree"
- **psscan - Process Scan:**
  - Scans the memory for terminated or hidden processes that might not be visible in traditional process lists.
  - Command: "volatility -f <memory_dump> --profile=<profile> psscan"
- **dlllist - DLL List:**
  - Lists all loaded dynamic link libraries (DLLs) for each process.
  - Command: "volatility -f <memory_dump> --profile=<profile> dlllist"
- **handles - Handles:**
  - Enumerates open handles for each process, including file handles, registry keys, and other resources.
  - Command: "volatility -f <memory_dump> --profile=<profile> handles"

Volatility Framework Plugins (3/3)

- **getsids - GetSIDs:**
  - Retrieves the Security Identifiers (SIDs) associated with each process.
  - Command: "volatility -f <memory_dump> --profile=<profile> getsids"
- **filescan - File Scan:**
  - Scans the memory for file-related objects and displays information about open files.
  - Command: "volatility -f <memory_dump> --profile=<profile> filescan"
- **malfind - Malicious Find:**
  - Identifies potentially malicious code or injected DLLs within the memory.
  - Command: "volatility -f <memory_dump> --profile=<profile> malfind"
- **cmdline - Command-Line:**
  - Retrieves the command-line arguments used to launch each process.
  - Command: "volatility -f <memory_dump> --profile=<profile> cmdline"
- **netscan - Network Connections:**
  - Displays information about active network connections and listening ports.
  - Command: "volatility -f <memory_dump> --profile=<profile> netscan"

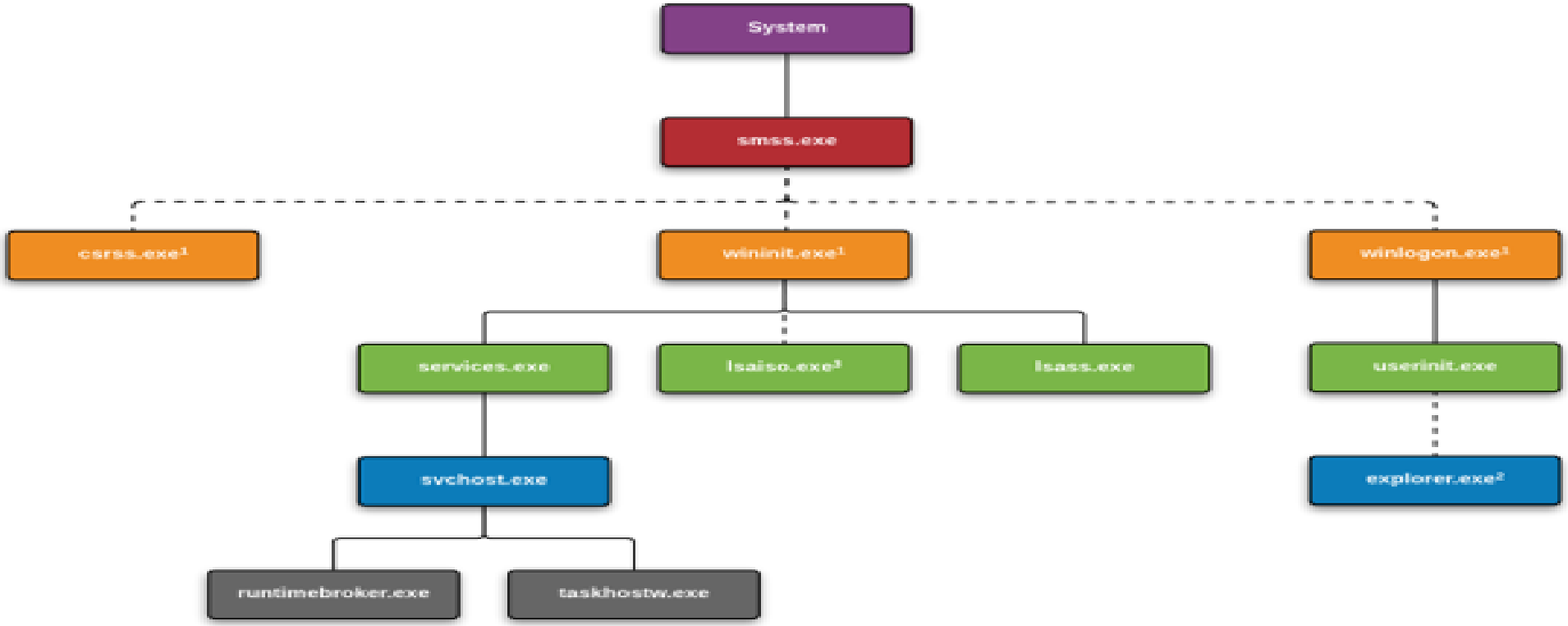
Volatility Framework Cheat Sheet

[CheatSheet_v2.4 (volatilityfoundation.org)]



Windows Memory Analysis

Windows Process Genealogy



[1] Created by an instance of **smss.exe** that exits, so analysis tools usually do not provide the parent process name.

[2] Created by an instance of **userinit.exe** that exits, so analysis tools usually do not provide the parent process name.

[3] Present only when **Credential Guard** is enabled. Functionality of lsass.exe is split between itself and this process.

Source: 13cubed.com

Process Analyzing – Identify rogue processes

Indicators of Rogue Processes:

- **Unusual Process Names:**
  - suspicious or misspelled names that mimic legitimate processes.
- **Mismatched Parent-Child Relationships:**
  - unusual parent-child relationships
- **Unusual Network Activity:**
  - processes with suspicious outbound or inbound traffic
- **Account Activity:**
  - processes owned by unusual accounts
- **Unauthorized DLLs:**
  - processes loading DLLs from uncommon or unauthorized locations.

Process Analyzing – Identify rogue processes – Volatility Plugins

- **pslist:**
  - Shows the offset, process name, process ID, the parent process ID, number of threads, number of handles, and date/time when the process started and exited.
  - It does not detect hidden or unlinked processes

```
remnux@remnux:~                vol.py pslist
Volatility Foundation Volatility Framework 2.6.1
Offset(V)          Name                    PID   PPID   Thds     Hnds   Sess  Wow64 Start                          Exit
------------------ -------------------- ------ ------ ------ -------- ------ ------ ------------------------------
------------------------------
0xffffaf8f430946c0 System                    4      0    119        0 ------      0 2023-02-28 15:09:37 UTC+0000
0xffffaf8f440a8800 smss.exe                308      4      2        0 ------      0 2023-02-28 15:09:37 UTC+0000
0xffffaf8f442b7080 csrss.exe               408    400     10        0      0      0 2023-02-28 15:09:39 UTC+0000
0xffffaf8f44dcc800 smss.exe                468    308      0 --------      1      0 2023-02-28 15:09:39 UTC+0000   2023-02-28 15:09:40 UTC+0000
0xffffaf8f44dc2080 csrss.exe               476    468     12        0      1      0 2023-02-28 15:09:39 UTC+0000
0xffffaf8f44dfb080 wininit.exe             496    400      1        0      0      0 2023-02-28 15:09:39 UTC+0000
...
```

Process Analyzing – Identify rogue processes – Volatility Plugins

- **pstree:**
  - This enumerates processes using the same technique as “pslist” plugin
  - It does not detect hidden or unlinked processes
  - Identifies the parent/children from pslist and shows them as a tree.

```
Volatility Foundation Volatility Framework 2.6.1
Name                                                 Pid   PPid   Thds   Hnds Time
---------------------------------------------------- ------ ------ ------ ------ ----
 0xffffaf8f442b7080:csrss.exe                        408    400     10       0 2023-02-28 15:09:39 UTC+0000
 0xffffaf8f44dfb080:wininit.exe                      496    400      1       0 2023-02-28 15:09:39 UTC+0000
. 0xffffaf8f44664800:services.exe                    608    496      5       0 2023-02-28 15:09:40 UTC+0000
.. 0xffffaf8f446fc580:svchost.exe                    896    608     52       0 2023-02-28 15:09:41 UTC+0000
...
 0xffffaf8f430946c0:System                            4      0    119       0 2023-02-28 15:09:37 UTC+0000
. 0xffffaf8f440a8800:smss.exe                        308    4       2       0 2023-02-28 15:09:37 UTC+0000
.. 0xffffaf8f44dcc800:smss.exe                       468    308     0 ------ 2023-02-28 15:09:39 UTC+0000
... 0xffffaf8f442b6080:winlogon.exe                  556    468     5       0 2023-02-28 15:09:40 UTC+0000
.... 0xffffaf8f45604800:fontdrvhost.ex               968    556     5       0 2023-02-28 15:47:15 UTC+0000
..... 0xffffaf8f444d4800:PanGPA.exe                 3392   3216    11       0 2023-02-28 15:47:15 UTC+0000
..... 0xffffaf8f45b87800:vm3dservice.ex              536    3216    4       0 2023-02-28 15:47:12 UTC+0000
..... 0xffffaf8f45a88800:WINWORD.EXE                3676   3216    30      0 2023-02-28 15:49:25 UTC+0000
..... 0xffffaf8f43e90080:ai.exe                      844    3676    8       0 2023-02-28 15:49:26 UTC+0000
..... 0xffffaf8f44488080:vmtoolsd.exe                2936   3216    8       0 2023-02-28 15:47:13 UTC+0000
... 0xffffaf8f44dc2080:csrss.exe                     476    468     12      0 2023-02-28 15:09:39 UTC+0000
. 0xffffaf8f44cb6780:MemCompression                  1980   4       20      0 2023-02-28 15:09:42 UTC+0000
 0xffffaf8f45a77080:whoami.exe                       3380   4188    0 ------ 2023-02-28 15:50:24 UTC+0000
```

Process Analyzing – Identify rogue processes – Volatility Plugins

- **psscan:**
  - This can find processes that previously terminated (inactive) and processes that have been hidden or unlinked by a rootkit.
  - If a process has previously terminated, the Time exited field will show the exit time.
  - Identifies the parent/children from pslist and shows them as a tree.

pslist

```
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
```

psscan

```
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
0x0000000006943da0 spoolsv.exe        1432    676 0x06cc01a0 2010-08-11 06:06:26 UTC+0000
0x000000000069a7328 VMip.exe          1944    124 0x06cc0320 2010-08-15 19:17:55 UTC+0000   2010-08-15 19:17:56 UTC+0000
0x00000000069e1d28 vmtoolsd.exe       1668    676 0x06cc01c0 2010-08-11 06:06:35 UTC+0000
```

Process Analyzing – Identify rogue processes – Volatility Plugins

- **psxview:**
  - Uses different mechanisms to search for processes and shows a comparison between them.
  - Good for finding discrepancies (i.e. processes hidden by the attacker).
  - A "False" in any column indicates that the respective process is missing.

```
Volatility Foundation Volatility Framework 2.6.1
Offset(P)  Name                    PID pslist psscan thrdproc pspcid csrss session deskthrd ExitTime
---------- -------------------- ------ ------ ------ -------- ------ ----- ------- -------- --------
0x06015020 services.exe            676 True   True   True     True   True  True    True
0x063c5560 svchost.exe             936 True   True   True     True   True  True    True
0x06499b80 svchost.exe            1148 True   True   True     True   True  True    True
0x04c2b310 wscntfy.exe             888 True   True   True     True   True  True    True
0x049c15f8 TPAutoConnect.e        1084 True   True   True     True   True  True    True
0x05f027e0 alg.exe                 216 True   True   True     True   True  True    True
0x05f47020 lsass.exe               688 True   True   True     True   True  True    True
0x010f7588 wuauclt.exe             468 True   True   True     True   True  True    True
0x01122910 svchost.exe            1028 True   True   True     True   True  True    True
0x069d5b28 vmtoolsd.exe           1668 True   True   True     True   True  True    True
0x06384230 vmacthlp.exe            844 True   True   True     True   True  True    True
0x0115b8d8 svchost.exe             856 True   True   True     True   True  True    True
0x04b5a980 VMwareUser.exe          452 True   True   True     True   True  True    True
0x010c3da0 wuauclt.exe            1732 True   True   True     True   True  True    True
0x04a065d0 explorer.exe           1724 True   True   True     True   True  True    True
0x04be97e8 VMwareTray.exe          432 True   True   True     True   True  True    True
0x0211ab28 TPAutoConnSvc.e        1968 True   True   True     True   True  True    True
0x06945da0 spoolsv.exe            1432 True   True   True     True   True  True    True
0x066f0978 winlogon.exe            632 True   True   True     True   True  True    True
0x0655fc88 VMUpgradeHelper        1788 True   True   True     True   True  True    True
0x061ef558 [illegible]            1088 True   True   True     True   True  True    True
0x06238020 cmd.exe                 124 True   True   False    True   False False   False    2010-08-15 19:17:56 UTC+0000
0x[illegible] csrss.exe            608 True   True   True     True   False True    True
0x05471020 smss.exe                [illegible]                                      [illegible]
0x[illegible] System                 4 True   True   True     True   False False   False
0x069a7328 VMip.exe               1944 False  True   False    False  False False   False    2010-08-15 19:17:56 UTC+0000
```

Process Analyzing – Process Object Analysis

- **cmdline:**
  - Displays the process command-line arguments.
  - This plugin can be used to detect whether the process is launched using a malicious command or not.

```
Volatility Foundation Volatility Framework 2.6.1
**************************************************
System pid:      4
**************************************************
smss.exe pid:    544
Command line : \SystemRoot\System32\smss.exe
**************************************************
csrss.exe pid:    608
Command line : C:\WINDOWS\system32\csrss.exe ObjectDirectory=\Windows SharedSection=1024,3072,512 Windows=On SubSystemType=Windows ServerDll=basesrv,1 ServerDll=winsrv:UserServerDllInitialization,3 ServerDll=win
srv:ConServerDllInitialization,2 ProfileControl=Off MaxRequestThreads=16
**************************************************
winlogon.exe pid:    632
Command line : winlogon.exe
**************************************************
services.exe pid:    676
Command line : C:\WINDOWS\system32\services.exe
**************************************************
lsass.exe pid:    688
Command line : C:\WINDOWS\system32\lsass.exe
**************************************************
vmacthlp.exe pid:    844
Command line : "C:\Program Files\VMware\VMware Tools\vmacthlp.exe"
**************************************************
```

Process Analyzing – Process Object Analysis

- **dlllist:**
  - Displays a process's loaded DLLs
  - The load count column tells you if a DLL was statically loaded (i.e. as a result of being in the exe or another DLL's import table) or dynamically loaded.

```
svchost.exe pid:    856
Command line : C:\WINDOWS\system32\svchost -k DcomLaunch
Service Pack 2

Base             Size  LoadCount LoadTime                       Path
---------- ---------- ---------- ------------------------------ ----
0x01000000     0x6000     0xffff                                C:\WINDOWS\system32\svchost.exe
0x7c900000    0xb0000     0xffff                                C:\WINDOWS\system32\ntdll.dll
0x7c800000    0xf4000     0xffff                                C:\WINDOWS\system32\kernel32.dll
0x77dd0000    0x9b000     0xffff                                C:\WINDOWS\system32\ADVAPI32.dll
0x77e70000    0x91000     0xffff                                C:\WINDOWS\system32\RPCRT4.dll
0x5cb70000    0x26000        0x1                                C:\WINDOWS\system32\ShimEng.dll
0x6f880000   0x1ca000        0x1                                C:\WINDOWS\AppPatch\AcGenral.DLL
0x77d40000    0x90000       0x85                                C:\WINDOWS\system32\USER32.dll
0x77f10000    0x46000       0x61                                C:\WINDOWS\system32\GDI32.dll
0x76b40000    0x2d000        0x4                                C:\WINDOWS\system32\WINMM.dll
0x774e0000   0x13c000       0x14                                C:\WINDOWS\system32\ole32.dll
0x77c10000    0x58000       0x97                                C:\WINDOWS\system32\msvcrt.dll
0x77120000    0x8c000        0xb                                C:\WINDOWS\system32\OLEAUT32.dll
0x77be0000    0x15000        0x1                                C:\WINDOWS\system32\MSACM32.dll
0x77c00000     0x8000        0x8                                C:\WINDOWS\system32\VERSION.dll
0x7c9c0000   0x814000        0x4                                C:\WINDOWS\system32\SHELL32.dll
0x77f60000    0x76000       0x10                                C:\WINDOWS\system32\SHLWAPI.dll
0x769c0000    0xb3000        0x4                                C:\WINDOWS\system32\USERENV.dll
0x5ad70000    0x38000        0x1                                C:\WINDOWS\system32\UxTheme.dll
0x773d0000   0x102000        0x4                                C:\WINDOWS\WinSxS\x86_Microsoft.Windows.Common-Controls_6595b64144ccf1df_6.0.2600.2180_x-ww_a84f1ff9\comctl32.dll
0x5d090000    0x97000        0x2                                C:\WINDOWS\system32\comctl32.dll
0x77690000    0x21000        0x1                                C:\WINDOWS\system32\NTMARTA.DLL
0x76f60000    0x2c000        0x3                                C:\WINDOWS\system32\WLDAP32.dll
```

Process Analyzing – Process Object Analysis

- **handles:**
  - Enumerates open handles for each process.
  - Applies to files, registry keys, mutexes, named pipes, events, window stations, desktops, threads, and all other types of objects

```
Volatility Foundation Volatility Framework 2.6.1
Offset(V)     Pid     Handle     Access Type            Details
---------- ------ ---------- ---------- --------------- -------
0x810b1660      4        0x4   0x1f0fff Process         System(4)
0x810b0020      4        0x8        0x0 Thread          TID 12 PID 4
0xe10192c0      4        0xc    0xf003f Key             MACHINE\SYSTEM\CONTROLSET001\CONTROL\SESSION MANAGER\MEMORY MANAGEMENT\PREFETCHPARAMETERS
0xe1019880      4       0x10        0x0 Key
0xe13b4578      4       0x14    0x2001f Key             MACHINE\SYSTEM\SETUP
0xe101d140      4       0x18    0x20019 Key             MACHINE\HARDWARE\DESCRIPTION\SYSTEM\MULTIFUNCTIONADAPTER
0xe13b46e0      4       0x1c    0x20019 Key             MACHINE\SYSTEM\WPA\MEDIACENTER
0xe13b4748      4       0x20    0x20019 Key             MACHINE\SYSTEM\WPA\KEY-CJ27J3P2XV9J9JCPB4DVT
0xe13b4678      4       0x24    0x20019 Key             MACHINE\SYSTEM\WPA\PNP
0xe13b45e0      4       0x28    0x20019 Key             MACHINE\SYSTEM\WPA\SIGNINGHASH-6KCM6KFTX6MD62
0xe13be4a8      4       0x2c    0x2001f Key             MACHINE\SYSTEM\CONTROLSET001\CONTROL\PRODUCTOPTIONS
0xe13be308      4       0x30    0x20019 Key             MACHINE\SYSTEM\CONTROLSET001\SERVICES\EVENTLOG
```

Network Artifacts & Memory

Indicators of Abnormal Network Activity:

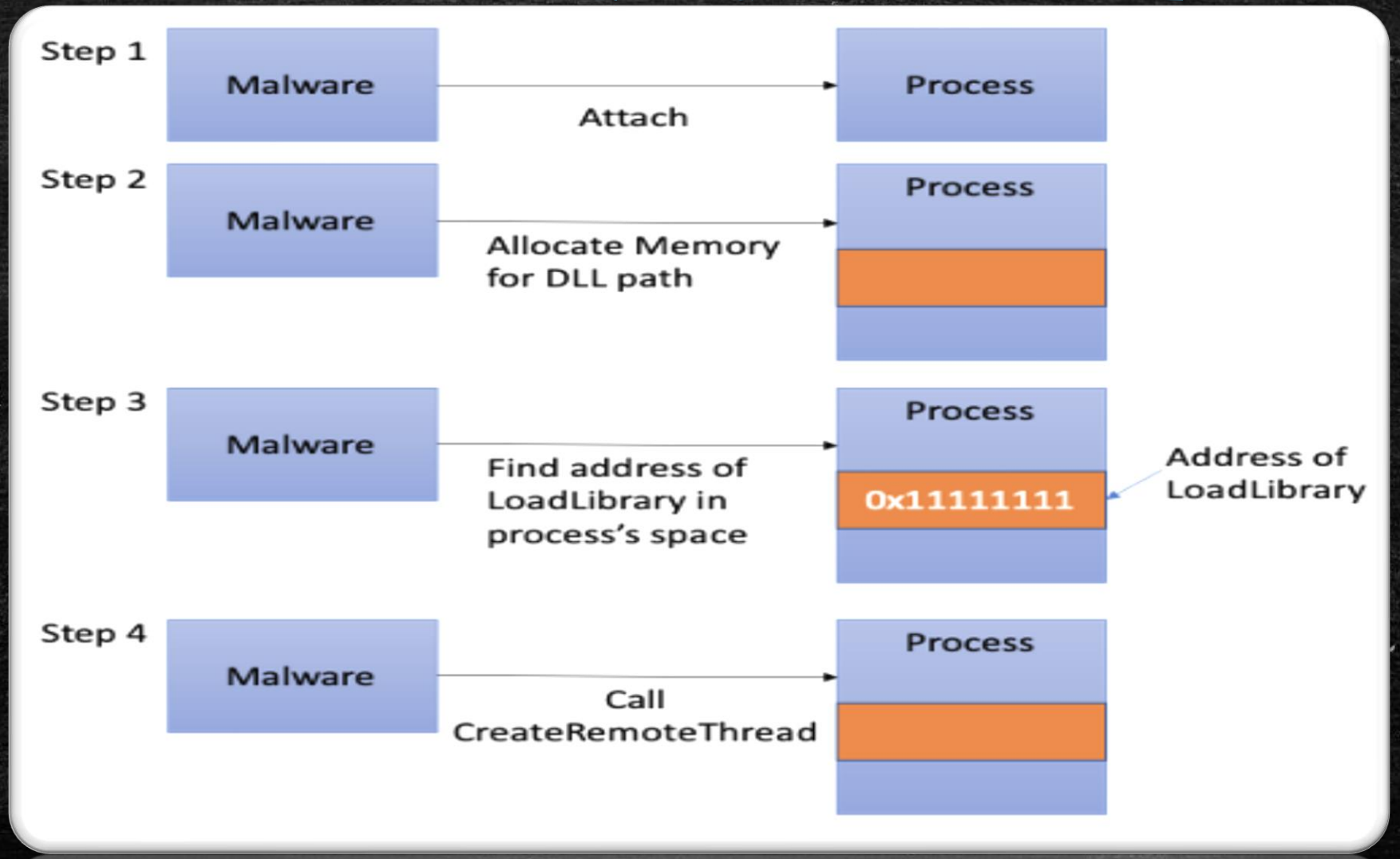
- Processes that run over ports 80,443 or 8080 and are NOT browsers
- Browsers that do not run over ports 80,443 or 8080
- Unexplained communications with internal or external IP addresses
- Web requests directly to IP addresses (not domain names)
- Unauthorized RDP connections (port 3389) or other remote access protocol
- Communications with domain names that have malicious reputation

Network Artifacts & Memory

- **netscan:**
  - Scans for network artifacts.
  - It finds TCP endpoints, TCP listeners, UDP endpoints, and UDP listeners.
  - It distinguishes between IPv4 and IPv6, prints the local and remote IP (if applicable), the local and remote port (if applicable), the time when the socket was bound or when the connection was established, and the current state (for TCP connections only).

```
Volatility Foundation Volatility Framework 2.6.1
Offset(P)          Proto    Local Address                Foreign Address      State          Pid      Owner          Created
0x8680000dd010     UDPv4    0.0.0.0:3702                 *:*                                 2900     svchost.exe    2023-02-28 15:09:47 UTC+0000
0x8680000f6ab0     UDPv4    0.0.0.0:0                    *:*                                 896      svchost.exe    2023-02-28 15:09:42 UTC+0000
0x8680000f6ab0     UDPv6    :::0                         *:*                                 896      svchost.exe    2023-02-28 15:09:42 UTC+0000
0x8680001272d0     UDPv4    0.0.0.0:500                  *:*                                 896      svchost.exe    2023-02-28 15:09:42 UTC+0000
...
...
...
00xaf8f43b13540    TCPv4    172.16.35.129:49969          20.0.219.79:443      ESTABLISHED    4752     netwifi.exe    2023-02-28 15:49:31 UTC+0000
```

Code Injection - Remote DLL Injection



Detecting DLL Injection & Malicious Code

- **malfind:**
  - Finds hidden and injected code.
  - Only processes with the **MZ** parameter in the header and **PAGE_EXECUTE_READWRITE** in the vad tags are important.
  - Normal processes executed with **PAGE_EXECUTE_WRITE** and **PAGE_EXECUTE_READWRITE** are malicious.
  - The -W flag refines the output only showing regions with an MZ header or that start with well known opcode combinations

```
Volatility Foundation Volatility Framework 2.6.1
Process: svchost.exe Pid: 856 Address: 0xb70000
Vad Tag: VadS Protection: PAGE_EXECUTE_READWRITE
Flags: CommitCharge: 38, MemCommit: 1, PrivateMemory: 1, Protection: 6

0x000000000b70000  4d 5a 90 00 03 00 00 00 04 00 00 00 ff ff 00 00  MZ..............
0x000000000b70010  b8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00  ........@.......
0x000000000b70020  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x000000000b70030  00 00 00 00 00 00 00 00 00 00 00 00 d0 00 00 00  ................
```

Suspicious process and files dumping (1/3)

- **procdump:**
  - Dumps a process's executable

```
Volatility Foundation Volatility Framework 2.4
************************************************************************
Dumping csrss.exe, pid:    296 output: executable.296.exe
```

- **memdump:**
  - Extracts all memory resident pages in a process into an individual file.

```
Volatility Foundation Volatility Framework 2.4
************************************************************************
Writing System [     4] to 4.dmp
```

Suspicious process and files dumping (2/3)

- **dlldump:**
  - Extracts a DLL from a process's memory space in order to proceed to further analysis.
  - If the extraction fails, it probably means that some of the memory pages in that DLL were not memory resident (due to paging).
  - To dump a PE file that doesn't exist in the DLLs list (for example, due to code injection or malicious unlinking), just specify the base address of the PE in process memory.

```
python vol.py -f ~/Desktop/win7_trial_64bit.raw --profile=Win7SP0x64 dlldump -D dlls/

Process(V)       Name             Module Base      Module Name       Result
---------------- ---------------- ---------------- ----------------- ------
0xfffffa8000ce97f0 smss.exe         0x0000000047a90000 smss.exe         OK:
module.208.176e97f0.47a90000.dll
0xfffffa8000ce97f0 smss.exe         0x0000000076d40000                Error: DllBase is paged
0xfffffa8000c006c0 csrss.exe        0x0000000049700000 csrss.exe        OK:
module.296.176006c0.49700000.dll
```

Suspicious process and files dumping (3/3)

- **Dumpregistry:**
  - Dumps registry hives for further analysis
  - By default the plugin will dump all registry files
  - Must specify the virtual offset for a specific hive in order to only dump one registry at a time.

Exploring registry - Finding persistency (2/2)

- **printkey:**
  - Displays the subkeys, values, data, and data types contained within a specified registry key

```
Volatility Foundation Volatility Framework 2.6.1
Legend: (S) = Stable   (V) = Volatile

---------------------------
Registry: \Device\HarddiskVolume1\WINDOWS\system32\config\software
Key name: Run (S)
Last updated: 2010-06-10 16:13:03 UTC+0000

Subkeys:

Values:
REG_SZ        VMware Tools    : (S) "C:\Program Files\VMware\VMware Tools\VMwareTray.exe"
REG_SZ        VMware User Process : (S) "C:\Program Files\VMware\VMware Tools\VMwareUser.exe"
```

Exploring registry – Other Useful plugins (1/3)

- **userassist:**
  - Prints Userassist registry keys and information, which include all the programs that the user ran.

```
Volatility Foundation Volatility Framework 2.6.1
----------------------------
Registry: \Device\HarddiskVolume1\Documents and Settings\Administrator\NTUSER.DAT
Path: Software\Microsoft\Windows\CurrentVersion\Explorer\UserAssist\{5E6AB780-7743-11CF-A12B-00AA004AE837}\Count
Last updated: 2010-06-10 16:11:44 UTC+0000

Subkeys:

Values:

REG_BINARY    UEME_CTLSESSION : Raw Data:
0x00000000  00 00 00 00 00 00 00 00                         ........
----------------------------
Registry: \Device\HarddiskVolume1\Documents and Settings\Administrator\NTUSER.DAT
Path: Software\Microsoft\Windows\CurrentVersion\Explorer\UserAssist\{75048700-EF1F-11D0-9888-006097DEACF9}\Count
Last updated: 2010-08-15 19:17:23 UTC+0000

Subkeys:

Values:

REG_BINARY    UEME_CTLSESSION : Raw Data:
0x00000000  d7 c8 59 0e 02 00 00 00                         ..Y.....

REG_BINARY    UEME_RUNPIDL:%csidl2%\MSN.lnk :
ID:             1
Count:          14
Last updated:   2010-06-10 16:10:27 UTC+0000
Raw Data:
0x00000000  01 00 00 00 13 00 00 00 56 24 cf 70 b7 08 cb 01   ........V$.p....

REG_BINARY    UEME_RUNPIDL:%csidl2%\Windows Media Player.lnk :
ID:             1
Count:          13
Last updated:   2010-06-10 16:10:27 UTC+0000
Raw Data:
0x00000000  01 00 00 00 12 00 00 00 56 24 cf 70 b7 08 cb 01   ........V$.p....

REG_BINARY    UEME_RUNPIDL:%csidl2%\Windows Messenger.lnk :
ID:             1
Count:          12
Last updated:   2010-06-10 16:10:27 UTC+0000
Raw Data:
0x00000000  01 00 00 00 11 00 00 00 56 24 cf 70 b7 08 cb 01   ........V$.p....
```

Exploring registry – Other Useful plugins (2/3)

- **shellbags:**
  - Prints Shellbags info, which are a set of registry keys that allow the Windows operating system to track user window viewing preferences specific to Windows Explorer.
  - Some artifacts that can be found here:
    - Icon and folder view settings
    - Windows sizes and preferences
    - Metadata such as MAC timestamps
    - Most Recently Used (MRU) files and file type (zip, directory, installer)
    - Files, folders, zip files, and installers that existed at one point on the system (even if deleted)
    - Network shares and folders within the shares

Exploring registry – Other Useful plugins (3/3)

- **hashdump:**
  - Extracts cached domain credentials stored in the registry.
  - Hashes can be cracked using John the Ripper, rainbow tables, etc.

```
Volatility Foundation Volatility Framework 2.6.1
Administrator:500:e52cac67419a9a224a3b108f3fa6cb6d:8846f7eaee8fb117ad06bdd830b7586c:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
HelpAssistant:1000:4e857c004024e53cd538de64dedac36b:842b4013c45a3b8fec76ca54e5910581:::
SUPPORT_388945a0:1002:aad3b435b51404eeaad3b435b51404ee:8f57385a61425fc7874c3268aa249ea1:::
```

Searching for privilege escalation

- **privs:**
  - Shows you which process privileges are present, enabled, and/or enabled by default.

```
Volatility Foundation Volatility Framework 2.6.1
Pid      Process          Value  Privilege                         Attributes               Description
-------- --------------- ------ ---------------------------------- ------------------------ -----------
       4 System                7 SeTcbPrivilege                    Present,Enabled,Default  Act as part of the operating system
       4 System                2 SeCreateTokenPrivilege            Present                  Create a token object
       4 System                9 SeTakeOwnershipPrivilege          Present                  Take ownership of files/objects
       4 System               15 SeCreatePagefilePrivilege         Present,Enabled,Default  Create a pagefile
       4 System                4 SeLockMemoryPrivilege             Present,Enabled,Default  Lock pages in memory
       4 System                3 SeAssignPrimaryTokenPrivilege     Present                  Replace a process-level token
       4 System                5 SeIncreaseQuotaPrivilege          Present                  Increase quotas
       4 System               14 SeIncreaseBasePriorityPrivilege   Present,Enabled,Default  Increase scheduling priority
       4 System               16 SeCreatePermanentPrivilege        Present,Enabled,Default  Create permanent shared objects
       4 System               20 SeDebugPrivilege                  Present,Enabled,Default  Debug programs
       4 System               21 SeAuditPrivilege                  Present,Enabled,Default  Generate security audits
       4 System                8 SeSecurityPrivilege               Present                  Manage auditing and security log
       4 System               22 SeSystemEnvironmentPrivilege      Present                  Edit firmware environment values
       4 System               23 SeChangeNotifyPrivilege           Present,Enabled,Default  Receive notifications of changes to files or directories
       4 System               17 SeBackupPrivilege                 Present                  Backup files and directories
       4 System               18 SeRestorePrivilege                Present                  Restore files and directories
       4 System               19 SeShutdownPrivilege               Present                  Shut down the system
       4 System               10 SeLoadDriverPrivilege             Present                  Load and unload device drivers
       4 System               13 SeProfileSingleProcessPrivilege   Present,Enabled,Default  Profile a single process
       4 System               12 SeSystemtimePrivilege             Present                  Change the system time
       4 System               25 SeUndockPrivilege                 Present                  Remove computer from docking station
       4 System               28 SeManageVolumePrivilege           Present                  Manage the files on a volume
       4 System               29 SeImpersonatePrivilege            Present,Enabled,Default  Impersonate a client after authentication
       4 System               30 SeCreateGlobalPrivilege           Present,Enabled,Default  Create global objects
     544 smss.exe              7 SeTcbPrivilege                    Present,Enabled,Default  Act as part of the operating system
```

Thank you for your patience!