SANS Memory Forensics Cheat Sheet 2.0

Memory Acquisition

- winpmem
  - Syntax
    - -o Output file location
    - -p <path to pagefile.sys> Include page file
    - -e Extract raw image from AFF4 file
  - Examples
    - winpmem_<version>.exe -o F:\mem.aff4
    - winpmem_<version>.exe F:\mem.aff4 -e PhysicalMemory -o mem.raw
- DumpIt
  - Syntax
    - /f Output file location
    - .s <value> Hash function to use
    - <addr> Send to remote host (set up listener with /l)
  - Example
    - DumpIt.exe /f F:\mem.raw /s 1

Memory Artifact Timelining

- Timeliner Plugin
  - --output-file
    - Optional file to write output
  - --output=body
    - Bodyfile format (also txt, xlsx)
  - --type=Registry
    - Extract Registry key last write times
  - Syntax
    - vol.py -f mem.img timeliner --output-file out.body --output=body --profile=Win10x64

- Output is sorted by
    - Process creation time
    - Thread creation time
    - Driver compile time
    - DLL/EXE compile time
    - Network socket creation time
    - Memory resident registry key last write time
    - Memory resident event log entry creation time

Registry Analysis Plugins

- hivelist
  - Find and list available registry hives
  - vol.py hivelist
- hivedump
  - Print all keys and subkeys in a hive
  - -o Offset of registry hive to dump (virtual offset)
  - vol.py hivedump –o 0xe1a14b60
- printkey
  - Output a registry key, subkeys, and values
  - -K “Registry key path”
  - vol.py printkey –K “Microsoft\Windows\CurrentVersion\Run”
- dumpregistry
  - Extract all available registry hives
  - -o Extract using virtual offset of registry hive
  - --dump-dir Directory to save extracted files
  - vol.py dumpregistry --dump-dir ./output
- userassist
  - Find and parse UserAssist key values
  - vol.py userassist
- hashdump
  - Dump user NTLM and Lanman hashes

    - Dump user NTLM and Lanman hashes
    - vol.py hashdump
  - autoruns
    - Map ASEPs to running processes
    - -v Show everything
    - vol.py autoruns -v

Converting Hibernation Files and Crash Dumps

- imagecopy Plugin
  - Convert alternate memory sources to raw
  - Syntax
    - -f Name of source file
    - -O Output file name
    - --profile Source OS from imageinfo
  - Examples
    - vol.py imagecopy -f hiberfil.sys -O hiber.raw --profile=Win7SP1x64
    - vol.py imagecopy -f MEMORY.DMP -O crashdump.raw --profile=Win2016x64_14393

Alternate Memory Locations

- Hibernation File
  - Compressed RAM image, available in Volume Shadow Copies (VSCs)
  - %SystemDrive%\hiberfil.sys
- Page and Swap Files
  - %SystemDrive%\pagefile.sys
  - %SystemDrive%\swapfile.sys (Win8+/2012+)
- Memory Dump
  - %WINDIR%\MEMORY.DMP

Using Volatility

- Plugins
  - Identify Rogue Processes
    - pslist
      - High level view of running processes

    - vol.py pslist
  - psscan
    - Scan memory for EPROCESS blocks
    - vol.py psscan
  - pstree
    - Display parent-process relationships
    - vol.py pstree
- Analyze Process DLLs and Handles
  - dlllist
    - List of loaded DLLs by process
    - vol.py dlllist -p 1022,868
      - -p shows information for specific process IDs
  - getsids
    - Print process security identifiers (SIDs)
      - -p shows information for specific process IDs
    - vol.py getsids -p 868
  - handles
    - List of open handles for each process
    - vol.py handles -p 868 -t File,Key
      - -p shows information for specific process IDs
      - -t displays handles of a certain type
        - Processs
        - Thread
        - Key
        - Event
        - Mutant
        - Token
        - Port
- Review Network Artifacts
  - netscan

    - Scan for TCP connections and sockets
    - vol.py netscan
  - XP Systems
    - Use connscan and sockscan instead of netscan
- Look for Evidence of Code Injection
  - malfind
    - Find injected code and dump sections
    - -p Show information only for specific PIDs
  - -o Provide physical offset of single process to scan
  - --dump-dir
    - Directory to save suspicious memory sections
    - vol.py malfind --dump-dir ./output_dir
  - ldrmodules
    - Detect unlinked DLLs
    - -p Show information only for specific PIDs
    - -v Verbose: show full paths from three DLL lists
    - vol.py ldrmodules –p 868 -v
  - hollowfind
    - Detect process hollowing techniques
    - -p Show information only for specific PIDs
    - -D Directory to save suspicious memory sections
    - vol.py hollowfind -D ./output_dir
- Check for Signs of a Rootkit
  - psxview
    - Find hidden processes using cross-view
    - vol.py psxview
  - modscan
    - Scan memory for loaded, unloaded, and unlinked drivers
    - vol.py modscan
  - apihooks

- apihooks
    - Find API/DLL function hooks
    - -p Operate only on specific PIDs
    - -Q Only scan critical processes and DLLS
    - vol.py apihooks
- ssdt
    - Hooks in System Service Descriptor Table
    - vol.py ssdt | egrep –v ‘(ntoskrnl|win32k)’
- driverirp
    - Identify I/O Request Packet (IRP) hooks
    - -r Analyze drivers matching REGEX name pattern
    - vol.py driverirp –r tcpip
- idt
    - Display Interrupt Descriptor Table
    - vol.py idt

- Extract Processes, Drivers, and Objects
    - dlldump
        - Extract DLLs from specific processes
        - -p Dump DLLs only for specific PIDs
        - -b Dump DLL using base offset
        - -r Dump DLLs matching REGEX name
        - --dump-dir Directory to save extracted files
        - vol.py dlldump --dump-dir ./output –r metsrv
    - moddump
        - Extract kernel drivers
        - -b Dump driver using offset address (from modscan)
        - -r Dump drivers matching REGEX name
        - --dump-dir Directory to save extracted files
        - vol.py moddump --dump-dir ./output –r gaopdx
    - procdump
        - Dump process to executable sample

    - Dump process to executable sample
    - -p Dump only specific PIDs
    - -o Specify process by physical memory offset
    - -n Use REGEX to specify process
    - --dump-dir Directory to save extracted files
    - vol.py procdump --dump-dir ./output –p 868
- memdump
    - Extract every memory section into one file
    - -p Dump memory sections from these PIDs
    - -n Use REGEX to specify process
    - --dump-dir Directory to save extracted files
    - vol.py memdump –-dump-dir ./output –p 868
- filescan
    - Scan memory for FILE_OBJECT handles
    - vol.py filescan
- dumpfiles
    - Extract FILE_OBJECTs from memory
    - -Q Dump using physical offset of FILE_OBJECT
    - -r Extract using a REGEX (add -i for case insensitive)
    - -n Add original file name to output name
    - --dump-dir Directory to save extracted files
    - vol.py dumpfiles -n -i -r \\.exe --dump-dir=./
- svcscan
    - Scan for Windows Service record structures
    - -v Show service DLL for svchost instances
    - vol.py svcscan -v
- cmdscan
    - Scan for COMMAND_HISTORY buffers
    - vol.py cmdscan
- consoles
    - Scan for CONSOLE_INFORMATION output

- Scan for CONSOLE_INFORMATION output
- vol.py consoles

Getting Started with Volatility

- Getting Help
  - vol.py –h (show options and supported plugins)
  - vol.py plugin –h (show plugin usage)
  - vol.py plugin --info (show available OS profiles)
- Sample Command Line
  - vol.py -f image --profile=profile plugin
- Identify System Profile
  - imageinfo
    - Display memory image metadata
    - vol.py –f mem.img imageinfo
- Using Environment Variables
  - Set name of memory image (takes place of -f)
    - export VOLATILITY_LOCATION=file:///images/mem.img
  - Set profile type (takes place of --profile=)
    - export VOLATILITY_PROFILE=Win10x64_14393