Memory Analysis with Volatility3 Framework

AIM: Use volatility3 framework and analyse memory dump file to find the operating system, hash of passwords , network info

Introduction

The Volatility3 framework is a powerful and flexible tool used for memory forensics and analysis. It allows digital forensic investigators and security professionals to extract valuable information from memory dumps of operating systems, including details about running processes, network connections, open files, and even artifacts like passwords and encryption keys. Volatility3 supports various operating systems and provides a range of plugins and capabilities to analyze memory images efficiently, making it a go-to tool in the field of digital forensics for uncovering crucial evidence during investigations.

After downloading Volatility from GitHub, the initial step is to execute the command **python3 vol.py -h**. This command provides a comprehensive list of all available commands that can be executed within the Volatility framework.

```
PS C:\Users\chitr\OneDrive\Desktop\BTECH SEM VI\Cybersecurity\lastActivity\volatility3> python3 vol.py -h
Volatility 3 Framework 2.7.0
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] [--filters FILTERS]
                  [--single-location SINGLE_LOCATION] [--stackers [STACKERS ...]]
                  [--single-swap-locations [SINGLE_SWAP_LOCATIONS ...]]
                  plugin ...

An open-source memory forensics framework

options:
  -h, --help            Show this help message and exit, for specific plugin options use
                        'volatility <pluginname> --help'
  -c CONFIG, --config CONFIG
                        Load the configuration from a json file
  --parallelism [{processes,threads,off}]
                        Enables parallelism (defaults to off if no argument given)
  -e EXTEND, --extend EXTEND
                        Extend the configuration with a new (or changed) setting
  -p PLUGIN_DIRS, --plugin-dirs PLUGIN_DIRS
                        Semi-colon separated list of paths to find plugins
  -s SYMBOL_DIRS, --symbol-dirs SYMBOL_DIRS
                        Semi-colon separated list of paths to find symbols
  -v, --verbosity       Increase output verbosity
  -l LOG, --log LOG     Log output to a file as well as the console
  -o OUTPUT_DIR, --output-dir OUTPUT_DIR
                        Directory in which to output any generated files
```

Analyse of memory dump file

Command to analyse : *python3 vol.py -f "C:\Users\chitr\OneDrive\Desktop\BTECH SEM VI\Cybersecurity\lastActivity\memdump.mem" windows.info.Info*

Result

```
PS C:\Users\chitr\OneDrive\Desktop\BTECH SEM VI\Cybersecurity\lastActivity\volatility3> python3 vol.py -f "C:\Users\chitr\OneDrive\Desktop\BTECH SEM VI\Cybersecurity
\lastActivity\memdump.mem" windows.info.Info
Volatility 3 Framework 2.7.0
Progress:  100.00               PDB scanning finished
Variable        Value

Kernel Base     0x8183a000
DTB     0x122000
Symbols file:///C:/Users/chitr/OneDrive/Desktop/BTECH%20SEM%20VI/Cybersecurity/lastActivity/volatility3/volatility3/symbols/windows/ntkrpamp.pdb/37D328E3BAE5460F8E662756ED80951D-2.json.xz
Is64Bit False
IsPAE   True
layer_name      0 WindowsIntelPAE
memory_layer    1 FileLayer
KdDebuggerDataBlock     0x81931c90
NTBuildLab      6001.18000.x86fre.longhorn_rtm.0
CSDVersion      1
KdVersionBlock  0x81931c68
Major/Minor     15.6001
MachineType     332
KeNumberProcessors      3405774849
SystemTime      2014-01-08 17:54:20
NtSystemRoot    C:\Windows
NtProductType   NtProductServer
NtMajorVersion  6
NtMinorVersion  0
PE MajorOperatingSystemVersion  6
PE MinorOperatingSystemVersion  0
PE Machine      332
PE TimeDateStamp        Sat Jan 19 05:30:58 2008
```

After analyzing the memory dump file, we can interpret certain observations, which can be summarized as follows:

- Kernel Base: This is the base address of the kernel in memory. In this case, it is 0x81832000.
- DTB: This is the Dynamic Trace Buffer, which is a kernel component that can be used to trace system activity. In this case, it is located at 0x122000.
- Symbols: This section shows the path to the symbol file that was used to help Volatility interpret the memory dump. The symbol file is located at "C:\Users\chitr\OneDrive\Desktop\BTECH SEM VI\Cybersecurity\lastActivity\volatility3\volatility3\symbols\windows\ntkrpamp.pdb".
- 64-bit support: The analysis shows that the memory dump is from a 32-bit system (x86).
- OS: The system is Windows Server 2008 RTM (release to manufacturing).
- Build info: The build lab of the system is 6001.18000.x86fre.longhorn_rtm.0
- Date/Time: The system time captured in the dump is 2014-01-08 17:54:20.

Network Information

Command :

python vol.py -f "C:\Users\chitr\OneDrive\Desktop\memdump.mem" windows.netscan.NetScan

```
Volatility 3 Framework 2.7.0
Progress:  100.00               PDB scanning finished
Offset  Proto   LocalAddr       LocalPort       ForeignAddr     ForeignPort     State   PID     Owner   Created

0x1cde008       UDPv4   0.0.0.0 56153   *       0               1088    svchost.exe     2014-01-08 02:17:50.000000
0x1cde008       UDPv6   ::      56153   *       0               1088    svchost.exe     2014-01-08 02:17:50.000000
0x1cde490       UDPv4   0.0.0.0 56152   *       0               1088    svchost.exe     2014-01-08 02:17:50.000000
0x3c45008       UDPv4   0.0.0.0 1701    *       0               4       System  2014-01-08 02:19:53.000000
0x3c45008       UDPv6   ::      1701    *       0               4       System  2014-01-08 02:19:53.000000
0x4ff9de0       TCPv4   0.0.0.0 1031    0.0.0.0 0       LISTENING       604     services.exe    N/A
0x4ff9de0       TCPv6   ::      1031    ::      0       LISTENING       604     services.exe    N/A
0x7f68de0       TCPv4   0.0.0.0 1031    0.0.0.0 0       LISTENING       604     services.exe    N/A
0x7f68de0       TCPv6   ::      1031    ::      0       LISTENING       604     services.exe    N/A
0x8231008       UDPv4   0.0.0.0 1701    *       0               4       System  2014-01-08 02:19:53.000000
0x8231008       UDPv6   ::      1701    *       0               4       System  2014-01-08 02:19:53.000000
0x886bde0       TCPv4   0.0.0.0 1031    0.0.0.0 0       LISTENING       604     services.exe    N/A
0x886bde0       TCPv6   ::      1031    ::      0       LISTENING       604     services.exe    N/A
0x1e0ea430      TCPv4   0.0.0.0 1723    0.0.0.0 0       LISTENING       4       System  N/A
0x1e0ec008      UDPv4   0.0.0.0 1701    *       0               4       System  2014-01-08 02:19:53.000000
0x1e0ec008      UDPv6   ::      1701    *       0               4       System  2014-01-08 02:19:53.000000
0x1e0ef008      UDPv6   ::1     65012   *       0               1000    svchost.exe     2014-01-08 02:19:54.000000
0x1e0f3a08      UDPv6   ::1     65011   *       0               1000    svchost.exe     2014-01-08 02:19:54.000000
0x1e10a878      UDPv4   192.168.119.191 137     *       0               4       System  2014-01-08 17:29:33.000000
0x1e13f6f8      TCPv4   0.0.0.0 445     0.0.0.0 0       LISTENING       4       System  N/A
0x1e13f6f8      TCPv6   ::      445     ::      0       LISTENING       4       System  N/A
0x1e143e10      TCPv4   -       1399    54.213.58.70    80      CLOSED  1888    iexplore.exe    -
0x1e159008      TCPv4   192.168.119.191 139     0.0.0.0 0       LISTENING       4       System  N/A
0x1e168008      TCPv4   -       1392    54.230.117.162  80      CLOSED  1888    iexplore.exe    -
0x1e172c30      TCPv4   -       1424    93.184.216.139  80      CLOSED  1888    iexplore.exe    -
0x1e18de10      TCPv4   -       1407    23.214.3.146    80      CLOSED  1888    iexplore.exe    -
0x1e20c050      UDPv4   0.0.0.0 0       *       0               1620    snmp.exe        2014-01-08 02:17:49.000000
0x1e20c050      UDPv6   ::      0       *       0               1620    snmp.exe        2014-01-08 02:17:49.000000
0x1e20c280      UDPv4   0.0.0.0 52325   *       0               1620    snmp.exe        2014-01-08 02:17:49.000000
0x1e20d9b8      UDPv4   0.0.0.0 161     *       0               1620    snmp.exe        2014-01-08 02:17:49.000000
0x1e20d9b8      UDPv6   ::      161     *       0               1620    snmp.exe        2014-01-08 02:17:49.000000
```

The windows.netscan.NetScan plugin is a tool designed to scan and present network-related information extracted from memory dumps. It provides detailed insights into network connections, including protocols, local and foreign addresses, ports, connection statuses, Process IDs (PIDs), and process owners. This data is valuable for identifying active network connections, potential malicious activities, and the services operating on the system. The plugin covers a wide range of connections, both TCP and UDP, displaying their states, associated processes, and timestamps of creation. Noteworthy connections include those related to commonly used services like DNS (port 53), FTP (port 21), RDP (port 3389), and SMB (port 445), offering a comprehensive view of network operations and potential security threats.

Hash of passwords

Command:

python vol.py -f "C:\Users\chitr\OneDrive\Desktop\memdump.mem" windows.hashdump

```
Volatility 3 Framework 2.7.0
Progress:  100.00               PDB scanning finished
User    rid     lmhash  nthash

Administrator   500     aad3b435b51404eeaad3b435b51404ee        e19ccf75ee54e06b06a5907af13cef42
Guest   501     aad3b435b51404eeaad3b435b51404ee        31d6cfe0d16ae931b73c59d7e0c089c0
student 1000    aad3b435b51404eeaad3b435b51404ee        e19ccf75ee54e06b06a5907af13cef42
probe   1002    aad3b435b51404eeaad3b435b51404ee        e19ccf75ee54e06b06a5907af13cef42
waldo   1004    aad3b435b51404eeaad3b435b51404ee        cfeac129dc5e61b2eb9b2e7131fc7e2b
YOUR-NAME       1005    aad3b435b51404eeaad3b435b51404ee        958c8526e4252b277d8d70adbd2ea2ce
```

The windows.hashdump.Hashdump plugin retrieves password hashes from a memory dump, including the username, Relative Identifier (RID), LM hash, and NT hash for each user account detected in memory. The LM hash and NT hash represent the user's password in a cryptographic format, which can be decrypted offline to reveal the actual password.

Here's an analysis of the extracted data:

- Administrator (RID: 500) - LM hash: aad3b435b51404eeaad3b435b51404ee, NT hash: e19ccf75ee54e06b06a5907af13cef42
- Guest (RID: 501) - LM hash: aad3b435b51404eeaad3b435b51404ee, NT hash: 31d6cfe0d16ae931b73c59d7e0c089c0
- student (RID: 1000) - LM hash: aad3b435b51404eeaad3b435b51404ee, NT hash: e19ccf75ee54e06b06a5907af13cef42
- probe (RID: 1002) - LM hash: aad3b435b51404eeaad3b435b51404ee, NT hash: e19ccf75ee54e06b06a5907af13cef42
- waldo (RID: 1004) - LM hash: aad3b435b51404eeaad3b435b51404ee, NT hash: cfeac129dc5e61b2eb9b2e7131fc7e2b
- YOUR-NAME (RID: 1005) - LM hash: aad3b435b51404eeaad3b435b51404ee, NT hash: 958c8526e4252b277d8d70adbd2ea2ce

Learning

- Get to know about the Volatility framework that is used for memory forensics and analysis, extracting valuable information from memory dumps for cybersecurity investigations.
- Common commands in Volatility include imageinfo, pslist, netscan, hashdump, malfind, and connections, each serving specific analysis purposes.
- Learning Volatility involves understanding memory structures, interpreting output, identifying malicious activity, and correlating findings for comprehensive forensic analysis.