LAB 7: WINDOWS FORENSICS – PART II

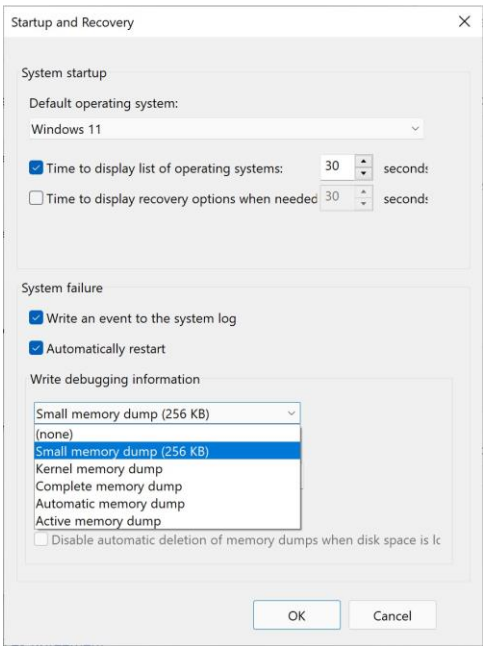
Lab Requirements

1. Microsoft Windows virtual machine
2. McAfee Bit Text
3. Belkasoft Live RAM Capturer

Content

Part I: Windows Crash Dump

STEP 1: In case of system failure, Windows 11 stores the memory backup. The memory backup, or crash dump, can later be used by the users or investigators to collect information about system state, memory locations, applications and program status, etc. Windows 11 can create any of the following memory dumps. Startup and Recovery window is accessible through the following: SYSTEM > ABOUT > ADVANCED SYSTEM SETTINGS > ADVANCED > STARTUP AND RECOVER > SETTINGS [> WRITE DEBUGGING INFORMATION]




```
# The memory crash dump is located in %SystemRoot%memory.dmp
C:\Windows\> dir *.dmp
    Volume in drive C has no label.
    Volume Serial Number is B879-6382

    Directory of c:\Windows

    02/16/2022  05:13 PM       420,822,909 MEMORY.DMP
                   1 File(s)    420,822,909 bytes
                   0 Dir(s)  68,369,600,512 bytes free
```

STEP 2: Use the dumpchk to analyze the crash dump files. The command dumpchk is from the Windows Debugging Tools (https://docs.microsoft.com/en-us/windows-hardware/drivers/debugger/)

```

# The bugcode is of interest for the debugging process, much more information
# is displayed.
C:\Program Files (x86)\Windows Kits\10\Debuggers\x64>dumpchk
C:\Windows\MEMORY.DMP | findstr "Bug*"
  BugCheckCode        000000ef
  BugCheckParameter1  ffffc101`45a31080
  BugCheckParameter2  00000000`00000000
  BugCheckParameter3            00000000`00000000      BugCheckParameter4
  00000000`00000000  fffff807`18f30000  fffff807`18f43000     CompositeBus
  063F7A78 (This is a reproducible build file hash, not a timestamp)
```

```
   fffff807`19510000 fffff807`1951d000   NdisVirtualBus  7C5FA602 (This is a
   reproducible build file hash, not a timestamp)
   *                        Bugcheck Analysis
   *
```

Part II: Collecting Process Information

STEP 3: Instead of analyzing the whole memory or all the running processes/services, investigators might be interested in analyzing a single or a group of services. To do that, one can dump a given service by performing the following steps.

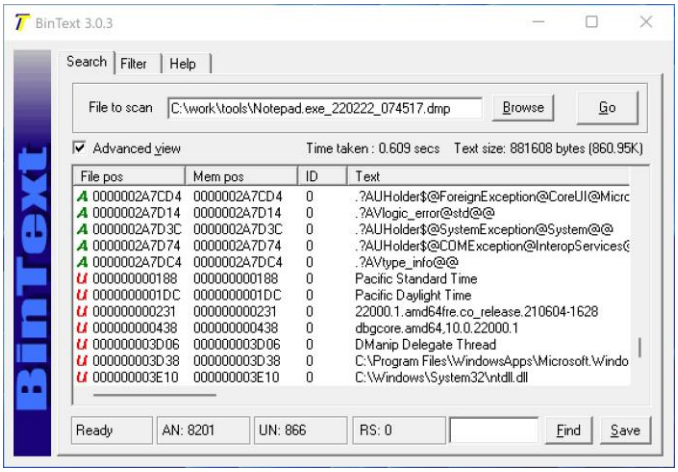
STEP 4: The tools pslist (from Sysinternals tools) or tasklist. Use pslist /? or tasklist /? for more information about these two tools.

```
# -a: all connections.
C:\work\tools> pslist -nobanner
   Process information for WINDEV2112EVAL:

   Name                Pid Pri Thd  Hnd   Priv        CPU Time    Elapsed Time
   Idle                  0   0   4    0     60    11:21:35.890    39:30:16.241
   System                4   8 134 3162     36     0:24:59.109    39:30:16.241
   Secure System        56   8   0    0    184     0:00:00.000    39:30:26.331
   Registry            108   8   4    0   8300     0:00:06.234    39:30:26.132
   ...
   Notepad            6272   8   7  828  24268     0:00:10.390     0:11:17.269
   pslist             9704  13   3  213   3372     0:00:00.296     0:00:00.274
   clip               6212   8   3   65   1000     0:00:00.000     0:00:00.260
```

STEP 5: To dump the memory of a particular process, use the command procdump (from Sysinternals tools).

```

# -mm: mini dump
# -ma full dump
# -mt triage dump
# -mk ‘kernel’ dump

# dump theNotepad process
C:\work\tools> procdump -nobanner -mm 6272
   [07:45:17] Dump 1 initiated: c:\work\tools\Notepad.exe_220222_074517.dmp
   [07:45:18] Dump 1 complete: 3 MB written in 0.3 seconds
   [07:45:18] Dump count reached.

```

STEP 6: To display the content of the dump, you might use the McAfee's BinText tool. The tool is not available anymore on MacAfee website, but it could be downloaded at github.com/mfput/McAfeeTools, among other useful tools. Once BinTxt is installed, run it and open the memory dump created in Step 5.



In the above BinText view, scroll down and check the different types collected information.

STEP 7: A process has a unique identifier (known as process ID or PID). When a process is created, a set of handles are created, which can be used by its internal functions to access resources. Such handlers have a similar concept to pointers in certain programming languages, like C.

```
# List all processes and their handlers – a long list
C:\work\tools> handle
  # a long list of services and handles is displayed


# display the handles associated with a particular process
C:\work\tools> handle -p 10116


   48: File  (RW-)   C:\Windows\System32
  1CC: File  (RW-)   C:\Windows\Temp\vmware-vmsvc-SYSTEM.log
  234: Section       \BaseNamedObjects\windows_shell_global_counters
  2B0: File  (R-D)   C:\Windows\System32\en-US\KernelBase.dll.mui
  370: Section       \BaseNamedObjects\__ComCatalogCache__
  380: Section       \BaseNamedObjects\__ComCatalogCache__
  384: File  (R--)   C:\Windows\Registration\R000000000006.clb
  464: File  (R-D)   C:\Windows\System32\en-US\mpr.dll.mui
  498: Section       \BaseNamedObjects\HGFSMEMORY
  4A8: Section       \BaseNamedObjects\windows_shell_global_counters
```

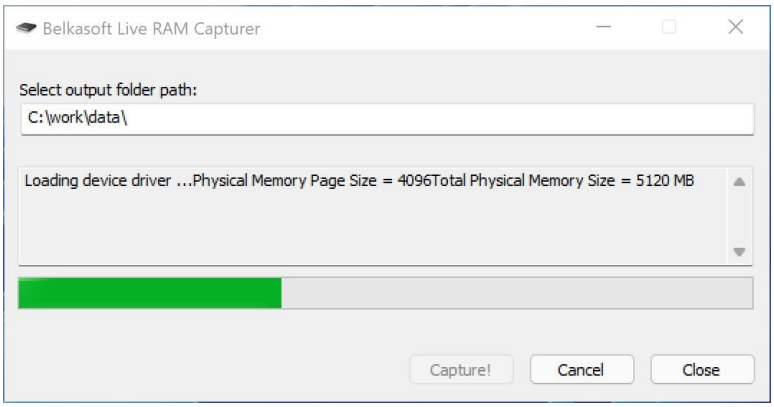
STEP 8: To list the executable and list dynamic link library (DLL files) loaded into processes, use the command listdlls.

```
# List all processes and loaded dll files – a long list
C:\work\tools> listdlls
  # a long list of services and handles is displayed
# display the dlls associated with a particular process
C:\work\tools> listdlls notepad.exe
    ----------------------------------------------------------------------------
    Notepad.exe pid: 8832
    Command line: "C:\Program
    Files\WindowsApps\Microsoft.WindowsNotepad_11.2112.32.0_x64__8wekyb3d8bbwe\
    Notepad\Notepad.exe"

    Base                Size      Path
    0x00000000ac0b0000  0x76000   C:\Program
    Files\WindowsApps\Microsoft.WindowsNotepad_11.2112.32.0_x64__8wekyb3d8bbwe\
    Notepad\Notepad.exe
    0x0000000006ca0000  0x209000  C:\Windows\SYSTEM32\ntdll.dll
    0x0000000005890000  0xbd000   C:\Windows\System32\KERNEL32.DLL
    0x00000000046a0000  0x374000  C:\Windows\System32\KERNELBASE.dll
    0x0000000006c00000  0x5d000   C:\Windows\System32\SHLWAPI.dll
    0x0000000006460000  0xa3000   C:\Windows\System32\msvcrt.dll
    0x00000000055b0000  0x1ac000  C:\Windows\System32\USER32.dll
    0x0000000004a20000  0x26000   C:\Windows\System32\win32u.dll
    0x0000000006980000  0x29000   C:\Windows\System32\GDI32.dll
    0x00000000042b0000  0x112000  C:\Windows\System32\gdi32full.dll
    0x0000000004600000  0x9d000   C:\Windows\System32\msvcp_win.dll
    0x0000000004190000  0x111000  C:\Windows\System32\ucrtbase.dll

```

Part III: RAM Acquisition

STEP 9: RAM can be acquired during live acquisition. The free Belkasoft RAM Capturer or AccessData FTK Imager can be used for this purpose.

STEP 10: The following is a snapshot of the main screen of the Belkasoft RAM Capturer. The memory dump is located in the specified folder and is of .mem extension.


Belkasoft Live RAM Capturer
Select output folder path:
C:\work\data\
Loading device driver ...Physical Memory Page Size = 4096Total Physical Memory Size = 5120 MB
Capture!
Cancel
Close