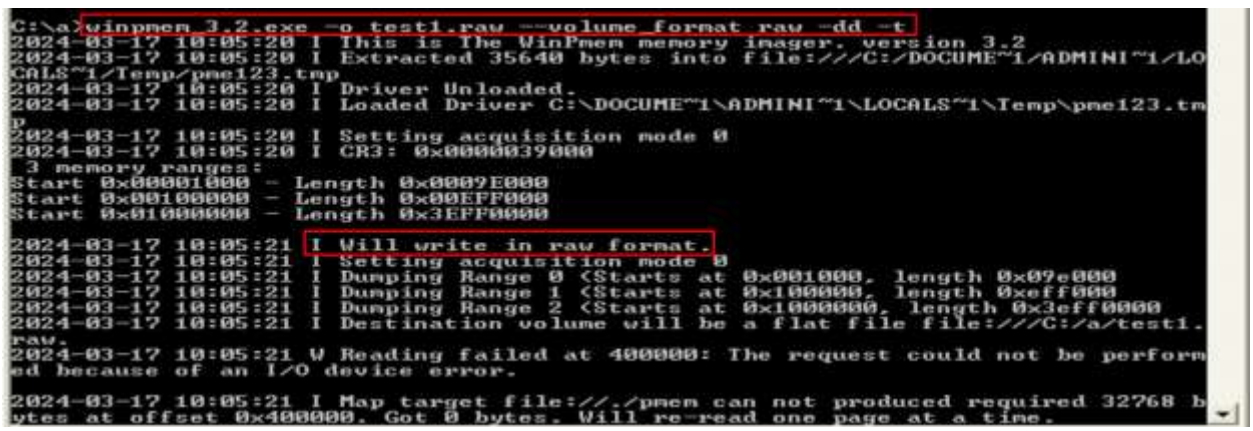
Contents

Executive Summary

This task emphasizes the practicality of volatility and its significance in memory forensics across various platforms, such as Windows, Linux, and macOS. The primary focus here lies in analyzing the kernel and registry of volatile memory images from both infected and clean operating systems. Key aspects addressed include utilizing volatility to identify loaded and unloaded kernel modules, system service descriptor tables, functions employed by various modules in memory, process and thread lists, orphan threads, explanation of module and function functionalities, and leveraging volshell—a Python-based interactive command prompt within the volatility framework—for scripting purposes.

Winpmem Image Acquisition

Before delving into the memory analysis of images from both operating systems, the initial step required us to acquire the memory image of the appropriate clean operating system. This was accomplished using Winpmem, a memory acquisition tool designed for capturing volatile memory across various architecture systems.



As you can see in the screenshot above, following command has been used to capture the memory of 32-bit Windows XP system:

'**winpmem.exe -o test.raw –volume format raw -dd -t**'

In the above command the Winpmem.exe initiates the execution of the memory acquisition process, -o tag is used to specify the output file which is test.raw, --volume format raw defines that the volume needs to be captured should be in raw format, -dd tag enables direct disk access mode in Winpmem allowing to bypass cache and capture the most up to data memory data and -t tag defines the process timeout which by default is 0.

And as you can also see in the screenshot below, we have successfully acquired our image of the clean Windows XP using volatility. The driver unloaded instruction you see in the screenshot below corresponds to the working of Winpmem in acquiring the bit-by-bit image of an operating

system. It loads its own driver into the system whose image needs to be acquired, creates a temporary image by an extension of pmem and when the specific format image is completed it deletes the temporary image and unloads its driver from the operating system.

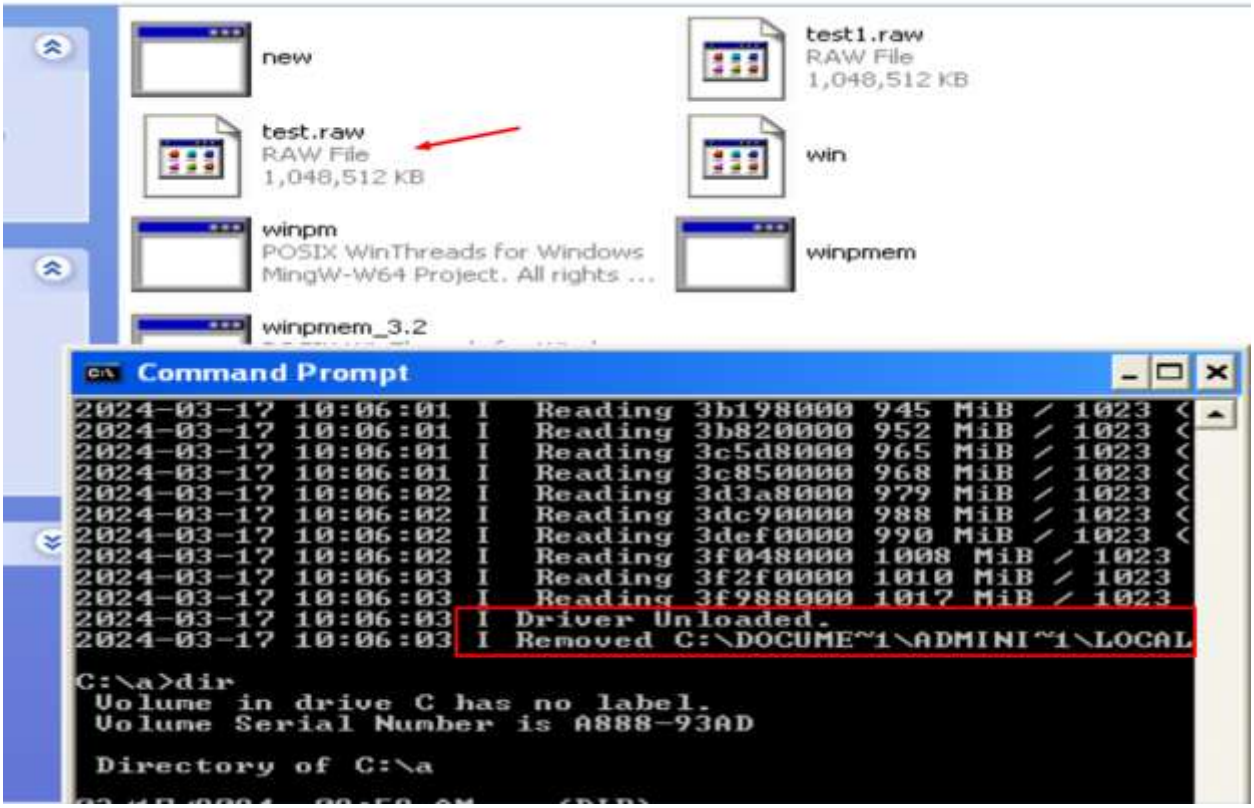
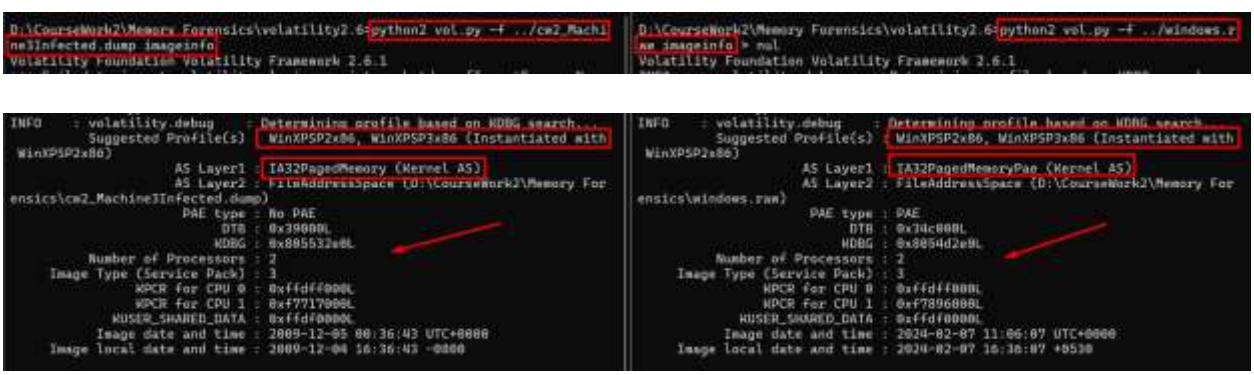


Image information

I used the following command to get the information on both infected and clean images:

'python2 vol.py -f <path to image file (infected & clean)> imageinfo '



Now that we have checked that both of our images belong to the same operating system 32-bit Windows XP, let us get started with the kernel and registry analysis as we were assigned to do in this section.

Q1- Describe 5 modules that are currently loaded in both dumps.

Infected Memory Modules

Using 'modules' tag in volatility2 provides all the currently loaded kernel modules in a memory image file. Use the command shown in the screenshot below to get the currently loaded kernel modules for relative memory image file:

```
D:\CourseWork2\Memory Forensics\volatility2.6>python2 vol.py -f ../cw2_Machine3Infected.dump modules
Volatility Foundation Volatility Framework 2.6.1
```

As you can see in the screenshot below that we have quite a number of modules loaded into the memory image. Some famous modules among these are NTOSKernel and mount manager modules.

```
Offset(V)  Name                 Base             Size File
---------- -------------------- ---------- ---------- ----
0x89c3d390 ntoskrnl.exe         0x804d7000   0x228000 \WINDOWS\system32\ntoskrnl.exe
0x89c3d328 hal.dll              0x806ff000    0x20d00 \WINDOWS\system32\hal.dll
0x89c3d2c0 kdcom.dll            0xf7987000     0x2000 \WINDOWS\system32\KDCOM.DLL
0x89c3d250 BOOTVID.dll          0xf7897000     0x3000 \WINDOWS\system32\BOOTVID.dll
0x89c3d1e8 ACPI.sys             0xf75a8000    0x2e000 ACPI.sys
0x89c3d178 WMILIB.SYS           0xf7989000     0x2000 \WINDOWS\system32\DRIVERS\WMILIB.SYS
0x89c3d110 pci.sys              0xf7597000    0x11000 pci.sys
0x89c3d0a0 isapnp.sys           0xf75f7000     0xa000 isapnp.sys
0x89c3d030 PCIIde.sys           0xf7a4f000     0x1000 PCIIde.sys
0x89c20008 PCIIDEX.SYS          0xf7787000     0x7000 \WINDOWS\System32\Drivers\PCIIDEX.SYS
0x89c20f98 intelide.sys         0xf798b000     0x2000 intelide.sys
0x89c20f28 MountMgr.sys         0xf7607000     0xb000 MountMgr.sys
0x89c20eb8 ftdisk.sys           0xf74d8000    0x1f000 ftdisk.sys
0x89c20e48 dmload.sys           0xf798d000     0x2000 dmload.sys
0x89c20de0 dmio.sys             0xf74b2000    0x26000 dmio.sys
0x89c20d70 PartMgr.sys          0xf770f000     0x5000 PartMgr.sys
0x89c20d00 VolSnap.sys          0xf7617000     0xd000 VolSnap.sys
0x89c20c98 atapi.sys            0xf749a000    0x18000 atapi.sys
0x89c20c30 disk.sys             0xf7627000     0x9000 disk.sys
0x89c20bc0 CLASSPNP.SYS         0xf7637000     0xd000 \WINDOWS\system32\DRIVERS\CLASSPNP.SYS
0x89c20b50 fltMgr.sys           0xf747a000    0x20000 fltMgr.sys
0x89c20ae8 sr.sys               0xf7468000    0x12000 sr.sys
0x89c20a78 KSecDD.sys           0xf7451000    0x17000 KSecDD.sys
0x89c20a10 Ntfs.sys             0xf7b52000    0x8d000 Ntfs.sys
0x89c209a8 NDIS.sys             0xf7424000    0x2d000 NDIS.sys
0x89c20940 Mup.sys              0xf740a000    0x1a000 Mup.sys
0x89c208d0 avgrkx86.sys         0xf7871000    0x26000 avgrkx86.sys
0x89c20860 AVGIDSxx.sys         0xf7647000     0x9000 AVGIDSxx.sys
0x89b41278 intelppm.sys         0xf7557000     0x9000 \SystemRoot\system32\DRIVERS\intelppm.sys
0x89b0a008 ialmnt5.sys          0xba7fa000     0xa7000 \SystemRoot\system32\DRIVERS\ialmnt5.sys
0x89b082e0 VIDEOPRT.SYS         0xba7e6000    0x14000 \SystemRoot\system32\DRIVERS\VIDEOPRT.SYS
0x89b0ac70 usbuhci.sys          0xf776f000     0x6000 \SystemRoot\system32\DRIVERS\usbuhci.sys
0x89b34590 USBPORT.SYS          0xba7c2000    0x24000 \SystemRoot\system32\DRIVERS\USBPORT.SYS
0x89b32168 usbehci.sys          0xf7777000     0x8000 \SystemRoot\system32\DRIVERS\usbehci.sys
0x89b2e5f8 e1000325.sys         0xba7a4000    0x1e000 \SystemRoot\system32\DRIVERS\e1000325.sys
0x89b1ba48 fdc.sys              0xf777f000     0x7000 \SystemRoot\system32\DRIVERS\fdc.sys
0x89b0df00 serial.sys           0xf7547000    0x10000 \SystemRoot\system32\DRIVERS\serial.sys
0x89b00b48 serenum.sys          0xf793b000     0x4000 \SystemRoot\system32\DRIVERS\serenum.sys
```

```
0x89a3b768 netbios.sys          0xf7677000     0x9000 \SystemRoot\system32\DRIVERS\netbios.sys
0x89b11d90 rdbss.sys            0xb230c000    0x2b000 \SystemRoot\system32\DRIVERS\rdbss.sys
0x898881d0 mrxsmb.sys           0xb229c000    0x70000 \SystemRoot\system32\DRIVERS\mrxsmb.sys
0x89b114d8 Fips.SYS             0xf7687000     0xb000 \SystemRoot\System32\Drivers\Fips.SYS
0x898ac650 avgmfx86.sys         0xf77e7000     0x6000 \SystemRoot\System32\Drivers\avgmfx86.sys
0x8994dc98 avgldx86.sys         0xb1388000    0x50000 \SystemRoot\System32\Drivers\avgldx86.sys
0x89b2b300 Cdfs.SYS             0xf76e7000    0x10000 \SystemRoot\System32\Drivers\Cdfs.SYS
0x894ae8b8 dump_atapi.sys       0xb1348000    0x18000 \SystemRoot\System32\Drivers\dump_atapi.sys
0x894c18b8 dump_WMILIB.SYS      0xf7a05000     0x2000 \SystemRoot\System32\Drivers\dump_WMILIB.SYS
0x89b02008 win32k.sys           0xbf800000   0x1c4000 \SystemRoot\System32\win32k.sys
0x89b289a0 Dxapi.sys            0xf791f000     0x3000 \SystemRoot\System32\drivers\Dxapi.sys
0x89a4bb18 watchdog.sys         0xb24d2000     0x5000 \SystemRoot\System32\watchdog.sys
```

Description of 05 modules from infected memory image

1. **NTOSKernel**: **ntoskrnl.exe** (short for windows native operating system kernel executable), also known as the **kernel image**, contains the kernel and executive layers of the Microsoft Windows NT Kernel, and is responsible for hardware abstraction, process handling, and memory management.
2. **PCI**: The PCI module in Windows facilitates communication between the operating system and PCI devices, managing device recognition, initialization, and control.
3. **fltMgr**: The fltMgr.sys file in Windows is a kernel-mode filter manager responsible for managing file system filters, enabling features like encryption, compression, and antivirus scanning on files.
4. **mountMgr**: The mountmgr.sys file in Windows is a kernel-mode component responsible for managing and assigning drive letters, mounting and dismounting volumes, and handling requests related to storage devices and file system volumes.
5. **NetBIOS**: NetBIOS (Network Basic Input/Output System) is a legacy networking protocol used primarily for LAN communication and network resource sharing in Windows environments. It provides services for naming, session establishment, and datagram delivery between computers on a local network.

Clean Memory Modules

Repeat the same process using the same command but this time for the clean image file as below:

```
D:\CourseWork2\Memory Forensics\volatility2.6>python2 vol.py -f ../windows.raw modules
Volatility Foundation Volatility Framework 2.6.1
```

These are some of the modules as a result if the command above:

```
0x8659ee30 Msfs.SYS            0xf7926000     0x5000 \SystemRoot\System32\Drivers\Msfs.SYS
0x86561110 Npfs.SYS            0xf792e000     0x8000 \SystemRoot\System32\Drivers\Npfs.SYS
0x863babe8 rasacd.sys          0xf7aea000     0x3000 \SystemRoot\system32\DRIVERS\rasacd.sys
0x8651e150 ipsec.sys           0xf6c0b000    0x13000 \SystemRoot\system32\DRIVERS\ipsec.sys
0x8654ccb8 tcpip.sys           0xf6bb2000    0x59000 \SystemRoot\system32\DRIVERS\tcpip.sys
0x865921b8 netbt.sys           0xf6b8a000    0x28000 \SystemRoot\system32\DRIVERS\netbt.sys
0x8654c1c0 ipnat.sys           0xf6b64000    0x26000 \SystemRoot\system32\DRIVERS\ipnat.sys
0x865c6a80 afd.sys             0xf6b42000    0x22000 \SystemRoot\System32\drivers\afd.sys
0x864c9748 wanarp.sys          0xf775e000     0x9000 \SystemRoot\system32\DRIVERS\wanarp.sys
0x86433e40 netbios.sys         0xf776e000     0x9000 \SystemRoot\system32\DRIVERS\netbios.sys
0x8645b730 VBoxSF.sys          0xf6ae7000     0x5b000 \SystemRoot\System32\drivers\VBoxSF.sys
0x864dcd98 rdbss.sys           0xf6abc000     0x2b000 \SystemRoot\system32\DRIVERS\rdbss.sys
0x865abd58 mrxsmb.sys          0xf6a24000    0x70000 \SystemRoot\system32\DRIVERS\mrxsmb.sys
0x865c9f28 Fips.SYS            0xf77ae000     0xb000 \SystemRoot\System32\Drivers\Fips.SYS
0x864c62b8 Cdfs.SYS            0xf77ce000    0x10000 \SystemRoot\System32\Drivers\Cdfs.SYS
0x8647c008 dump_atapi.sys      0xf6a0c000    0x18000 \SystemRoot\System32\Drivers\dump_atapi.sys
0x8647dcc8 dump_WMILIB.SYS     0xf7b04000     0x2000 \SystemRoot\System32\Drivers\dump_WMILIB.SYS
0x86450898 win32k.sys          0xbf800000   0x1cb000 \SystemRoot\System32\win32k.sys
0x864cef18 Dxapi.sys           0xf6cec000     0x3000 \SystemRoot\System32\drivers\Dxapi.sys
0x864835f8 watchdog.sys        0xf793e000     0x5000 \SystemRoot\System32\watchdog.sys
0x86454008 dxg.sys             0xbf9cb000    0x12000 \SystemRoot\System32\drivers\dxg.sys
0x863cf398 dxgthk.sys          0xf7cce000     0x1000 \SystemRoot\System32\drivers\dxgthk.sys
0x865c5850 hidusb.sys          0xf6ce8000     0x3000 \SystemRoot\system32\DRIVERS\hidusb.sys
0x86462ca0 HIDCLASS.SYS        0xf77fe000     0x9000 \SystemRoot\system32\DRIVERS\HIDCLASS.SYS
0x86421098 HIDPARSE.SYS        0xf7946000     0x7000 \SystemRoot\system32\DRIVERS\HIDPARSE.SYS
0x863a3a58 mouhid.sys          0xf6ce4000     0x3000 \SystemRoot\system32\DRIVERS\mouhid.sys
0x8658d698 VBoxDisp.dll        0xbf9dd000    0x16000 \SystemRoot\System32\VBoxDisp.dll
0x85ff42a0 ndisuio.sys         0xf30e4000     0x4000 \SystemRoot\system32\DRIVERS\ndisuio.sys
0x85ff13d0 rspndr.sys          0xf77de000    0x10000 \SystemRoot\system32\DRIVERS\rspndr.sys
0x863ff6c8 mrxdav.sys          0xf2d94000     0x2c000 \SystemRoot\system32\DRIVERS\mrxdav.sys
0x8647e138 srv.sys             0xf2cec000    0x58000 \SystemRoot\system32\DRIVERS\srv.sys
0x86386ef8 intelppm.sys        0xf2c5c000     0x9000 \SystemRoot\system32\DRIVERS\intelppm.sys
0x8646a160 TDTCP.SYS           0xf795e000     0x6000 \SystemRoot\System32\Drivers\TDTCP.SYS
0x8646a3d8 RDPWD.SYS           0xf2ac1000    0x23000 \SystemRoot\System32\Drivers\RDPWD.SYS
0x863314b8 wdmaud.sys          0xf2a0c000    0x15000 \SystemRoot\system32\drivers\wdmaud.sys
0x85f68730 sysaudio.sys        0xf697c000     0xf000 \SystemRoot\system32\drivers\sysaudio.sys
0x8645b2c8 HTTP.sys            0xf27bd000    0x41000 \SystemRoot\System32\Drivers\HTTP.sys
0x85f686c0 kmixer.sys          0xf267a000    0x2b000 \SystemRoot\system32\drivers\kmixer.sys
0x863a3948 pme37.tmp           0xf69dc000     0xc000 \??\C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\pme37.tmp
```

Description of 05 modules from Clean memory image

1. **tcpip.sys (Microsoft TCP/IP Driver)**: This is a critical system file responsible for implementing the TCP/IP protocol suite in Windows operating systems.
2. **win32k.sys (Windows 32-bit Kernel Driver)**: A core part of the Windows graphics subsystem, it handles input and video memory management.
3. **ndisuio.sys (Network Data Link Interface User-Mode Driver)**: This is a user-mode driver for NDIS, which facilitates communication between NDIS and the user-mode components.
4. **Http.sys (HTTP Driver)**: This is a driver responsible for handling HTTP requests and responses. It is an integral part of the Windows HTTP stack and is used for various server-side functionalities.
5. **mrxsmb.sys (Microsoft Server Message Block Driver)**: This is a core driver that handles SMB (Server Message Block) protocol for file and printer sharing. It facilitates communication between Windows systems and other SMB-enabled devices.