Volatility is one of the best open source software programs for analyzing RAM in 32 bit/64 bit systems. It supports analysis for Linux, Windows, Mac, and Android systems. It is based on Python and can be run on Windows, Linux, and Mac systems. It can analyze raw dumps, crash dumps, VMware dumps, virtual box dumps, and many others. Volatility, memory forensics framework, is capable to perform monitoring runtime processes and state of any system using the data found in RAM. Therefore, it can perform reconnaissance on process lists, ports, network connections, registry files, DLL’s, crash dumps and cached sectors. This framework also provides a unique platform that enables better efficiency of Forensic research (Law Enforcement, defense forces, commercial investigators, etc. Volatility supports memory dumps from all Windows versions and service packs including XP, 2003 Server, Vista, Server 2008, Server 2008 R2, and Seven. It also supports Linux memory dumps and distributions such as Debian, Ubuntu, Fedora, CentOS, and Mandrake. The memory forensics framework supports Mac OSX memory dumps, Mountain Lion, Android phones with ARM processors.

**Plugins:**

Imageinfo – When the user is trying to perform an evaluation test when using Volatility, we need to set a profile so that Volatility can select which OS the memory dump is using. When you have the memory dump and you do not know which profile to select, you can use imageinfo. Imageinfo gives you different profiles of the memory dump, different layers, number of processors, image time and date, and other factors for the memory dump. **EX: volatility imageinfo -f /root/Desktop/MemChallenge.dmp**

Pslsit – If you need to show a high view of all the running processes from a memory dump in Volatility, you use pslist. Pslist will show the name of the process, the process ID number, the parent process ID number, the number of threads and handles, and the offset. Pslist will show you so much but it cannot detect hidden or unlinked processes. **EX:** **volatility pslist -f /root/Desktop/Challenge2.dmp --profile=WinXPSP2x86**

Psscan – Psscan is used for finding processes that were terminated or inactive and processes that are hidden or unlinked by using rootkit. Since pslist cannot find the hidden/unlinked processes, psscan is capable of this task. Rootkits can still hide by being overwritten by pool tags. Psscan includes offset, name of the exe. file, process ID, parental process ID, time created, and time exited. **EX:** **volatility psscan -f /root/Desktop/Challenge2.dmp --profile=WinXPSP2x86**

Consoles – This is a plugin that shows what the hacker has typed into the command prompt of the computer and it shows all the information corresponding to those actions. It collects the input and outputs of these actions and also prints the command that was written. With each command that is being printed, it shows the address that matches with it. **EX: volatility consoles -f /root/Desktop/MemChallenge.dmp --profile=Win7SP0x86**

Connections – in order to view all of the TCP connections from the memory dump that are active, you have to use the connections plugin. The concepts associated with this plugin are offsets, local address, remote address, and the process ID. **EX:** **volatility connections -f /root/Desktop/MemChallenge.dmp --profile=Win7SP0x86**

Netscan – In order to scan for network findings, you have to use netscan. This plugin will find TCP endpoints and listeners and also UDP endpoints and listeners. It shows the offset, local address, foreign address, the state of the offset, and others. Netscan command will use pool tag scanning. **EX:** **volatility netscan -f /root/Desktop/MemChallenge.dmp --profile=Win7SP0x86**

Hivescan – In order to find the physical address of the registry hives in the memory, you can use hivescan. It is supposed to be used along with another plugin such as hivelist and not used by itself. **EX: volatility hivescan -f /root/Desktop/MemChallenge.dmp --profile=WinXPSP2x86**

Hivelist – When using the hivelist plugin, it allows you to print a list of all the current registry hives that were located in the memory dump. When you run this plugin on Volatility, it gives you all the virtual addresses, physical addresses, and the path from where it is stored in the computer of the memory dump. You can also be specific to a certain hive and run the test for that. **EX: volatility hivelist -f /root/Desktop/MemChallenge.dmp --profile=Win7SP0x86**

Hashdump – This plugin is used for obtaining and decrypting the passwords of the user’s computer that is stored in the registry. For a more specific search, you can put the allotted address to directly search in there for possible passwords. **EX: volatility hashdump -f /root/Desktop/MemChallenge.dmp --profile=Win7SP0x86**

Privs – This plugin shows you the privileges that are present in the process's token, which have been enabled, and which were enabled by default. If any privileges are enabled by not enabled by default, you know they were explicitly set. If any privileges are enabled but not present, that's a strong indicator of DKOM. **EX: volatility privs -f /root/Desktop/Challenge2.dmp --profile=WinXPSP2x86**

Printkey – In order for you to see the values, data, subkeys, and data types within a specific registry key, you should use printkey. The plugin will search all hives and print the information, if it was found, for that specific key. For example if a key is found in more than one location, the information for that key will be printed for as many times it is found. **volatility printkey -f /root/Desktop/Challenge2.dmp --profile=WinXPSP2x86**

Handles – If you want to see the open handles in a process, you use this plugin. It will show you the offset, Pid, handle, access, and the type. This plugin is associated with registry keys, events, desktops, files, threads, and mutexes. **EX:** **volatility handles -f /root/Desktop/Challenge2.dmp --profile=WinXPSP2x86**

Iehistory – This plugin is used for showing the history and cache files and it finds links and entries. **EX:** **volatility iehistory -f /root/Desktop/Challenge2.dmp --profile=WinXPSP2x86**

Modules – In order to view the list of drivers in the kernel that are loaded onto the system, you have to use this plugin. It shows the base, the size, the file path, and the name of the file. **EX:** **volatility modules -f /root/Desktop/Challenge2.dmp --profile=WinXPSP2x86**

Pstree – In order to view processes in a tree form listing, use this plugin. This tallies up processes using similar techniques as pslist so it will also not show hidden or unlinked processes. EX: **volatility pstree -f /root/Desktop/Challenge2.dmp --profile=WinXPSP2x86**

Kdbgscan – This plugin is designed to positively identify the correct profile and the correct KDBG address. This plugin scans for the KDBGHeader signatures linked to Volatility profiles and applies sanity checks to reduce false positives. EX: **volatility kdbgscan -f /root/Desktop/Challenge2.dmp --profile=WinXPSP2x86**

Psxview – This plugin helps you detect hidden processes by comparing what PsActiveProcessHead contains with what is reported by various other sources of process listings. EX: **volatility psxview -f /root/Desktop/Challenge2.dmp --profile=WinXPSP2x86**

Procdump – In order to dump a processes executable file, use this plugin. If you use the “-u” in the command, you can bypass certain checks and restrictions. **EX: volatility procdump -f /root/Desktop/Challenge2.dmp --profile=WinXPSP2x86**

Memdump – In order to extract all of the memory into a single file, use this plugin. **EX: volatility memdump -f /root/Desktop/Challenge2.dmp --profile=WinXPSP2x86**