Volatile Evidence Collection

Objectives:

At the end of this episode, I will be able to:

Understand what the Order of Volatility of evidence is.

Explain what the volatile data collection methodology is.

Identify how IH&R team members should collect volatile system information from both Windows and Linux machines.

Define how IH&R team members should collect volatile network information.

External Resources:

Volatile Evidence Collection

Why is Volatile Data Important? -

Volatile Information refers to the data stored in the registries, cache, and RAM of digital devices.

This information is lost or erased whenever the system is turned off or rebooted.

What is the Order of Volatility?

▪ Registers and cache - exists for nanoseconds, and are always changing, so it is the most volatile data.

▪ Routing table, process table, kernel statistics, and memory - are all found in the ordinary memory of the computer, and will be resident for slightly longer than register and cache data.

▪ Temporary file systems - tend to be present for a longer time on the computer compared to routing tables, ARP cache, and so on. These systems are eventually over written or changed, sometimes within seconds or minutes.

▪ Disk or other storage media - Data stored on a disk can be available for a while, but it may be erased or overwritten.

▪ Remote logging and monitoring data related to the target system

▪ Physical configuration and network topology

▪ Archival media

What is the volatile data collection methodology? -

Step 1: Incident Response Preparation
Step 2: Incident Documentation
Step 3: Policy Verification
Step 4: Volatile Data Collection Strategy
Step 5: Volatile Data Collection Setup
Step 6: Volatile Data Collection Process

How do we collect Volatile Information: System Information -

System information can act as evidence in a criminal or security incident.

This information includes the current configuration and running state of the suspicious computer

The System Profile describes the baseline configuration of the suspicious computer and provides a physical snapshot of the system that is often requested by a forensic examiner.

A System Profile includes the following details about the configuration of the suspicious computer:

```
▪ OS type and version
▪ System installation date
▪ Registered owner
▪ System directory
▪ Total amount of physical memory
▪ Pagefile location
▪ Installed physical hardware and configurations
▪ Installed software applications
```

Tools and commands to collect the information:

```
▪ Systeminfo.exe (Windows)
▪ PsInfo (Windows)
▪ Cat (Linux)
▪ Uname (Linux)
```

How do we collect Volatile Information: Current System Date and Time/Command History -

System time refers to the exact date and time of the day when the incident happened, as per the coordinated universal time (UTC).

The system provides the system time so that the applications launched have access to the accurate time and date.

The knowledge of system time will give a great deal of context to the information collected in the subsequent steps. It will also assist in developing an accurate timeline of events that have occurred on the system.

NOTE: Incident Responders should also record the real time, or wall time, when recording the system time. Comparison of both the timings allows the incident responder to further determine whether the system clock was accurate or inaccurate.

The responders can extract system time and date with the help of the date /t & time /t commands, or use the net statistics server command.

NOTE: An alternative way for obtaining the system time details is by using the GetSystemTime function. This function copies the time details to a SYSTEMTIME structure that contains information of individual logged in members and the exact information of month, day, year, weekday, hour, minute, second, and milliseconds. Hence, this function provides better accuracy to the system time details.

The Command History shows the recent user activities and serves as an audit trail of investigative activity. Recent user activities include a list of recently executed commands performed by a remote or local user within an established command shell or terminal.

The incident responder should use the doskey /history command, which shows the history of the commands typed into that prompt.

How do we collect Volatile Information: Current System Uptime -

Current system Uptime indicates how long the system has been running since the last reboot.

Tools to collect uptime information include:

```
▪ PsUptime (Windows)
▪ Net Statistics (Windows)
▪ Uptime and W (Linux)
```

How do we collect Volatile Information: Running Processes -

The responders should gather information about all the processes running on the system, but no single utility systematically assesses ALL running processes. Therefore, use a combination of the following commands and utilities:

▪ Windows Operating System:

```
o Use netstat –ab output to determine all the executable files for running
```

processes.
o Use ListDLLs to determine DLLs loaded into processes. It is a utility that reports the DLLs loaded into processes. You can use it to list all DLLs loaded into all the processes, into a specific process, or to list the processes that have a particular DLL loaded. ListDLLs can also display full version information for DLLs, including their digital signature, and can scan processes for unsigned DLLs.

```
     https://docs.microsoft.com/en-us/sysinternals/downloads/listdlls

o Use Pslist.exe to display basic information about the already running
```

processes on a system, including the amount of time each process has been running (in both kernel and user modes). For example, Pslist-x switch shows processes, memory information, and threads.

```
o Create a process memory dump using the pmdump.exe utility and then perform
```

string searches on the file to know about suspected rogue process.

▪ Linux Operating System:

```
o Use top command to display system summary information as well as a list of
```

processes or threads Linux kernel is currently managing

```
o Use w command to display the current processes for each shell of each user

o Use ps command to display information about the root’s currently running
```

processes

```
o Use pstree command to display the processes on a system in the form of a tree
```

How do we collect Volatile Information: Open Files, Clipboard Data,
Service/Driver Information -

▪ Open Files -

```
▪ net file command
▪ PsFile utility
▪ Openfiles command
```

▪ Service/Driver Information -

```
▪ tasklist command line tool
▪ Windows Management Instrumentation Command (wmic)
```

How do we collect Volatile Information: Logged-On Users -

```
▪ PsLoggedOn Tool - displays both the locally logged on users and users
```

logged on via resources for either the local computer, or a remote one

Syntax: psloggedon [-] [-l] [-x] [\computername | username]

-l = Displays only local logons

-x = Does not display logon times

\computername = System name for which logon information should be shown

username = Searches the network for those systems to which that user is logged on

```
▪ net sessions Command - displays information about all logged in sessions of
```

the local computer

Syntax: net session [\ComputerName] [/delete]

\ComputerName = Identifies the computer for which you want to list or disconnect
sessions.

/delete = Ends the computer's session with ComputerName and closes all open
files on the computer for the session.

net help command: Displays help for the specified net command.

```
▪ LogonSessions Tool - lists the currently active logged-on sessions
```

Syntax: logonsessions [-c[t]] [-p]

-c = Prints output as CSV

-ct = Prints output as tab-delimited values

-p = Lists processes running in logged-on sessions

▪ Who (Linux: Local Users) It displays the user that is currently logged on locally

▪ Who Am I, Who –uH (Linux: Local Users) It determines the currently logged on
user, whereas Who –uH displays the idle times for logged on users

▪ Who –all/–a (Linux: Local and Remote Users) It displays all currently logged
on users, local and remote

▪ Last (Linux: Local and Remote Users) It displays a history of logged on users,
local and remote

▪ Lastlog (Linux: Local and Remote Users) It displays the last login times for
system accounts

▪ W (Linux: Local and Remote Users) It displays summaries of system usage,
currently logged on users, and logged on user activities

▪ Passwd (Linux: Local and Remote Users) It contains user account information,
including one-way encrypted passwords

How do we collect Volatile Information: DLLs or Shared Libraries -

Shared libraries are object files that installed programs and executables use to load different modules. These libraries share resources that are common among different applications. Collecting information about shared libraries helps to determine possible rogue or modified DLLs and shared libraries.

▪ ListDLLs - reports the DLLs loaded into processes.

You can use it to list all DLLs loaded into all the processes, into a specific process, or to list the processes that have a DLL loaded.

ListDLLs can also display full version information for DLLs, including their digital signature, and can also scan processes for unsigned DLLs.

Syntax:

listdlls [-r] [-v | -u] [processname|pid]

listdlls [-r] [-v] [-d dllname]

Parameters:

```
o Processname: Dump DLLs loaded by process (partial name accepted)
o Pid: Dump DLLs associated with the specified process id
o Dllname: Shows only processes that have loaded the specified DLL
o -r: Flags DLLs that relocated because they are not loaded at their base address
o -u: Lists unsigned DLLs
o -v: Shows DLL version information
```

NOTE: The tool displays the full path of the loaded module as well as the version of the loaded DLL.

How do we collect Volatile Information: Network Information -

The NetBIOS name table cache maintains a list of connections made to other systems using NetBIOS Networking. It contains the remote system's name and IP address. You can use the Windows built-in command line utility Nbtstat to view the NetBIOS name table cache.

▪ Nbtstat - The syntax of the Nbtstat command is:

Nbtstat [[-a RemoteName] [-A IP address] [-c] [-n][-r] [-R] [-RR] [-s] [-S] [interval]]

o nbtstat -c: This option shows the contents of the NetBIOS name cache, which contains NetBIOS name-to-IP address mappings.

o nbtstat -n: This displays the names that have been registered locally on the system by NetBIOS applications such as the server and redirector.

o nbtstat -r: This command displays the count of all NetBIOS names resolved by broadcast and by querying a WINS server.

o nbtstat -S: This option is used to list the current NetBIOS sessions and their statuses.

How do we collect Volatile Information: Network Connections -

▪ Netstat - collects information about network connections operative in a Windows system.

This tool provides a simple view of TCP and UDP connections, their state and network traffic statistics.

NOTE: Netstat.exe is a built-in tool with the Windows operating system.

The most common way to run Netstat is with the -ano switches. These switches tell the program to display the TCP and UDP network connections, listening ports, and the identifiers of the processes (PIDs).

Using Netstat with the -r switch will display the routing table and show, if any persistent routes are enabled in the system.

Syntax -

netstat [-a] [-e] [-n] [-o] [-p Protocol] [-r] [-s] [Interval]

Parameters:

o -a: Displays all active TCP connections as well as the TCP and UDP ports on which the computer is listening.

o -e: Displays Ethernet statistics, such as the number of bytes and packets sent and received. This parameter can be combined with -s.

o -n: Displays active TCP connections However, the addresses and port numbers are expressed numerically with no specified names.

o -o: Displays active TCP connections and includes the process ID (PID) for each connection. You can find the application based on the PID on the Processes tab in Windows Task Manager. This parameter can be combined with -a, -n, and -p.

o -p Protocol: Shows connections for the protocol specified. In this case, the Protocol can be TCP, UDP, ICMP, IP, ICMPv6, IPv6 TCPv6, or UDPv6. Using this parameter with –s will display protocol based statistics. -s: Displays statistics by protocol. By default, this will show the statistics for the TCP, UDP, ICMP, and IP protocols. In case of installed IPv6 protocol, the tool displays statistics for the TCP over IPv6, UDP over IPv6, ICMPv6, and IPv6 protocols. The use of -p parameter can specify a set of protocols.

o -r: Displays the contents of the IP routing table. This is equivalent to the route print command.

o Interval: Redisplays the selected information after the interval of defined number of seconds. Press CTRL+C to stop the redisplay. Omitting this parameter, will enable Netstat to print the selected information.