Memory Forensics

Volatility

Volatility is an open-source memory forensics toolkit written in Python. Volatility allows us to analyse memory dumps taken from Windows, Linux and Mac OS devices and is an extremely popular tool in memory forensics. For example, Volatility allows us to:

- List all processes that were running on the device at the time of the capture
- List active and closed network connections
- Use Yara rules to search for indicators of malware
- Retrieve hashed passwords, clipboard contents, and contents of the command prompt
- And much, much more!

And finally, now we need to decide what we want to analyse the image for. Volatility uses plugins to perform analysis, such as:

- Listing processes
- Listing network connections
- Listing contents of the clipboard, notepad, or command prompt
- And much more! If you're curious, you can read the documentation here

Using Volatility to Analyse an Image

Before proceeding with our analysis, we need to confirm the Operating System of the device that the memory has been captured from. We need to confirm this because it will determine what plugins we can use in our investigation.
First, let's use the plugin to analyse our memory dump file to determine the Operating System. To do this, we need to use the following command (remembering to include our memory dump by using the option:
Note: This can sometimes take a couple of minutes, depending on the size of the memory dump and the hardware of the system running the scan.

==========

These are some plugins for dealing with windows dumps.

Plugins: NOTE*	**Descript-ion**	**Objective**
windows.pslist	This plugin lists all of the processes that were running at the time of the capture.	To discover what processes were running on the system.
windows.psscan	This plugin allows us to analyse a specific process further.	To discover what a specific process was actually doing.
windows.dumpfiles	This plugin allows us to export the process, where we can perform further analysis (i.e. static or dynamic analysis).	To export a specific binary that allows us further to analyse it through static or dynamic analysis.

Plugins: NOTE*	Description	Objective
windows.netstat	This plugin lists all network connections at the time of the capture.	To understand what connections were being made. For example, was a process causing the computer to connect to a malicious server? We can use this IP address to implement defensive measures on other devices. For example, if we know an IP address is malicious, and another device is communicating with it, then we know that device is also infected.

**Please note that this is not all of the possible plugins. An extensive list of the Windows sub-set of plugins can be found here.*