Volatility

# 🔍 Volatility – Notes

## **Task 1: Introduction**

* Volatility is an **open-source memory forensics framework**.
* Used in DFIR to analyze RAM images and detect malicious activity.
* Helps uncover malware, hidden processes, network connections, and rootkits.

## **Task 2: Volatility Overview**

* Works on memory dumps from **Windows, Linux, MacOS, and Android**.
* Python-based, plugin-driven.
* Key advantage: **detects artifacts that disappear after reboot**.

## **Task 3: Installing Volatility**

* Install via pip install volatility3 or from GitHub repo.
* Requires Python 3.
* Dependencies: yara-python, pefile.
* Two versions:
  + **Volatility 2** (older, Python2).
  + **Volatility 3** (modern, Python3, better plugin support).

## **Task 4: Memory Extraction**

* RAM images are acquired using tools like:
  + **DumpIt**, **FTK Imager**, **LiME**, **Magnet RAM Capture**.
* File formats supported: **.raw, .dd, .mem, .vmem**.
* Always keep **hash values** for integrity.

## **Task 5: Plugins Overview**

* Volatility functionality is plugin-based.
* Examples:
  + pslist, pstree → processes.
  + connscan, netscan → network activity.
  + dlllist → loaded DLLs.
  + malfind → injected code/malware.
  + filescan, handles → file/registry artifacts.

## **Task 6: Identifying Image Info and Profiles**

* Command: vol.py -f memory.raw imageinfo
* Detects OS version, architecture, service pack.
* Profile examples: Win7SP1x64, Win10x64.
* In Volatility 3: profiles are **auto-detected**.

## **Task 7: Listing Processes and Connections**

* pslist, pstree → list processes.
* psscan → shows hidden/terminated processes.
* dlllist → modules/DLLs per process.
* connscan (Vol2) / netscan (Vol3) → active/closed connections.

## **Task 8: Volatility Hunting and Detection Capabilities**

* Detect rootkits: malfind, ldrmodules.
* Detect hidden processes: psscan.
* Memory injection detection: malfind, yarascan.
* Persistence detection: registry analysis (printkey).

## **Task 9: Advanced Memory Forensics**

* Malware detection with yarascan (YARA rules).
* Analyze dumped executables with procdump or dumpfiles.
* Registry hives, cached passwords, browser data can be extracted.
* Supports timeline creation (timeliner).

## **Task 10: Practical Investigations**

* Workflow example:
  1. Acquire RAM image.
  2. Identify profile/image info.
  3. List processes (pslist, pstree).
  4. Look for anomalies (hidden processes, injected code).
  5. Check network connections.
  6. Dump suspicious binaries for offline malware analysis.

## **Task 11: Conclusion**

* Volatility is essential for **memory forensics**.
* Detects malware and persistence mechanisms invisible to disk forensics.
* Supports both **threat hunting** and **incident response**.

