Intro to Malware Analysis

## **Intro to Malware Analysis**

### **Task 1 – Introduction**

* Define malware and its role in cyberattacks.
* Explain the importance of malware analysis for defenders, SOC analysts, and threat hunters.
* Goals: detection, prevention, attribution, and incident response.

### **Task 2 – Malware Analysis**

* The process of **studying malicious software** to understand its behavior, functionality, and impact.
* Helps in developing IOCs, signatures, and defense strategies.
* Divided into **Static Analysis** and **Dynamic Analysis**.

### **Task 3 – Techniques of Malware Analysis**

1. **Static Analysis** – Examining the file without execution.
2. **Dynamic Analysis** – Running malware in a sandbox/VM to observe behavior.
3. **Hybrid Analysis** – Combining static + dynamic for deeper insight.
4. **Manual vs. Automated** approaches.

### **Task 4 – Basic Static Analysis**

* Uses **hashing (MD5/SHA256)**, **strings extraction**, and **file metadata** checks.
* **VirusTotal** lookups.
* Checking **imports, exports, and libraries**.
* Useful tools: PEiD, PEview, strings, Detect It Easy (DIE).

### **Task 5 – The PE File Header**

* Windows executables use **Portable Executable (PE) format**.
* Key structures:
  + **DOS Header** – basic info, "MZ" signature.
  + **PE Header** – architecture, sections, imports/exports.
  + **Sections** – .text, .data, .rdata, .rsrc.
* Important for identifying **packing, obfuscation, and malicious code injection**.

### **Task 6 – Basic Dynamic Analysis**

* Run the malware in **isolated VM/sandbox**.
* Observe behavior with tools like:
  + **Process Monitor (Procmon)** – file/registry activity.
  + **Process Explorer** – running processes.
  + **Wireshark** / **TCPView** – network activity.
  + **RegShot** – registry changes.
* Goal: detect persistence, C2 communications, privilege escalation attempts.

### **Task 7 – Anti-analysis Techniques**

* Malware often tries to **evade detection** by:
  + **Packing & Obfuscation**.
  + **Anti-VM checks** (detecting VirtualBox, VMware).
  + **Anti-debugging tricks** (e.g., API calls like IsDebuggerPresent).
  + **Delays / Junk code** to waste analyst time.
  + **Code injection** into legitimate processes.

### **Task 8 – Conclusion**

* Malware analysis provides critical threat intelligence.
* **Static analysis** is fast but limited; **dynamic analysis** reveals real behavior.
* Analysts must be aware of **anti-analysis tricks**.
* Outputs: IOCs, YARA rules, detection signatures, and defensive strategies.

 