🐞MALWARE THREATS

Malware Threats

> Malware is malicious software that damages or disables computer systems and gives limited or full control of those systems to the malware creator for theft or fraud. Malware includes viruses, worms, Trojans, rootkits, backdoors, botnets, ransomware, spyware, adware, scareware, crapware, roughware, crypters, keyloggers, and other software.

Trojans

- In Ancient Greek mythology, the Greeks won the Trojan War with the aid of a giant wooden horse that the Greeks built to hide their soldiers. The Greeks left the horse in front of the gates of Troy. The Trojans, thinking that it was a gift from the Greeks that they had left before apparently withdrawing from the war, brought the horse into their city. At night, the hidden Greek soldiers emerged from the wooden horse and opened the city's gates for their soldiers, who eventually destroyed the city of Troy.
- Thus, taking its cue from this myth, a computer Trojan is a program in which malicious or harmful code is contained inside apparently harmless programming or data in such a way that it can gain control and cause damage such as ruining the file allocation table on your hard disk.

RAT Trojan

- **Attackers use Remote Access Trojans (RATs)** to infect the target machine to gain administrative access. RATs help an attacker to remotely access the complete GUI and control the victim's computer without his/her awareness. They can perform screening and camera capture, code execution, keylogging, file access, password sniffing, registry management, and other tasks. The virus infects victims via phishing attacks and drive-by downloads and propagates through infected USB keys or networked drives. It can download and execute additional malware, execute shell commands, read and write registry keys, capture screenshots, log keystrokes, and spy on webcams.
- **njRAT** is a RAT with powerful data-stealing capabilities. In addition to logging keystrokes, it is capable of accessing a victim's camera, stealing credentials stored in browsers, uploading and downloading files, performing process and file manipulations, and viewing the victim's desktop.
- This RAT can be used to control Botnets (networks of computers), allowing the attacker to update, uninstall, disconnect, restart, and close the RAT, and rename its campaign ID. The attacker can further create and configure the malware to spread through USB drives with the help of the Command and Control server software.

Crypter is a software that encrypts the original binary code of the .exe file to hide viruses, spyware, keyloggers, and RATs, among others, in any kind of file to make them undetectable by anti-viruses. SwayzCryptor is an encrypter (or "crypter") that allows users to encrypt their program's source code.

Theef RAT Trojan

- Theef is a Remote Access Trojan written in Delphi. It allows remote attackers access to the system via port 9871. Theef is a Windows-based application for both client and server. The Theef server is a virus that you install on a target computer, and the Theef client is what you then use to control the virus.

Viruses and Worms

- **Viruses** can attack a target host's system using a variety of methods. They can attach themselves to programs and transmit themselves to other programs by making use of specific events. Viruses need such events to take place, since they cannot self-start, infect hardware, or transmit themselves using non-executable files. "Trigger" and "direct attack" events can cause a virus to activate and infect the target system when the user triggers attachments received through email, Web sites, malicious advertisements, flashcards, pop-ups, or other methods. The virus can then attack a system's built-in programs, antivirus software, data files, and system startup settings, or perform other malicious activities.
- Like a virus, a worm does not require a host to replicate, but in some cases, the worm's host machine also infects. At first, Blackhat professionals treated worms as a mainframe problem. Later, with the introduction of the Internet, they concentrated and targeted Windows OSes using the same worms by sharing them by email, IRC, and other network functions.

The JPS Virus Maker tool is used to create its own customized virus. This tool has many options for building that can be used to create a virus. Some of the tool's features are auto-start, shutdown, disable security center, lock mouse and keyboard, destroy protected storage, and terminate windows. An ethical hacker and pen-tester can use the JPS Virus Maker Tool as a proof of concept to audit perimeter security controls in an organization.

Static Malware Analysis

- Static Malware Analysis, also known as code analysis, involves going through the executable binary code without executing it to gain a better understanding of the malware and its purpose. The process includes the use of different tools and techniques to determine the malicious part of the program or a file. It also gathers information about malware functionality and collects the technical pointers or simple signatures it generates. Such pointers include file name, MD5 checksums or hashes, file type, and file size. Analyzing the binary code provides information about the malware's functionality, network signatures, exploit packaging technique, dependencies involved, as well as other information.

Some of the static malware analysis techniques are:

- File fingerprinting
- Local and online malware scanning

- Performing strings search
- Identifying packing and obfuscation methods
- Finding portable executable (PE) information
- Identifying file dependencies
- Malware disassembly

Hybrid Analysis is a free service that analyzes suspicious files and URLs and facilitates the quick detection of unknown threats such as viruses, worms, Trojans, and other kinds of malware.

- https://www.hybrid-analysis.com/

Identify File Dependencies using Dependency Walker

- Any software program depends on the various inbuilt libraries of an OS that help in performing specified actions in a system. Programs need to work with internal system files to function correctly. Programs store their import and export functions in a kernel32.dll file. File dependencies contain information about the internal system files that the program needs to function properly; this includes the process of registration and location on the machine.
- Find the libraries and file dependencies, as they contain information about the run-time requirements of an application. Then, check to find and analyze these files to provide information about the malware in the file. File dependencies include linked libraries, functions, and function calls. Check the dynamically linked list in the malware executable file. Finding out all library functions may allow guessing about what the malware program can do. You should know the various DLLs used to load and run a program.
- The Dependency Walker tool lists all dependent modules of an executable file and builds hierarchical tree diagrams. It also records all functions that each module exports and calls. Further, it detects many common application problems such as missing and invalid modules, import and export mismatches, circular dependency errors, mismatched machine modules, and module initialization failures.

Perform Malware Disassembly using IDA and OllyDbg

- Static analysis also includes the dismantling of a given executable into binary format to study its functionalities and features. This process helps identify the language used for programming the malware, look for APIs that reveal its function, and retrieve other information. Based on the reconstructed assembly code, you can inspect the program logic and recognize its threat potential. This process uses debugging tools such as IDA Pro and OllyDbg.

Perform Malware Disassembly using Ghidra

- Ghidra is a software reverse engineering (SRE) framework that includes a suite of full-featured, high-end software analysis tools that enable users to analyze compiled code on a variety of platforms including Windows, MacOS, and Linux. It's capabilities include disassembly, assembly, decompilation, debugging, emulation, graphing, and scripting. Ghidra supports a wide variety of processor instruction sets and executable formats and

can be run in both user-interactive and automated modes. Analysts can also develop their own Ghidra plug-in components and/or scripts using the exposed API. In addition there are numerous ways to extend Ghidra such as new processors, loaders/exporters, automated analyzers, and new visualizations.

Dynamic Malware Analysis

- Dynamic Malware Analysis, also known as behavioral analysis, involves executing malware code to learn how it interacts with the host system and its impact after infecting the system.
- Dynamic analysis involves the execution of malware to examine its conduct and operations and identify technical signatures that confirm the malicious intent. It reveals information such as domain names, file path locations, created registry keys, IP addresses, additional files, installation files, and DLL and linked files located on the system or network.
- This type of analysis requires a safe environment such as machines and sandboxes to deter the spreading of malware. The environment design should include tools that can capture every movement of the malware in detail and give feedback. Typically, systems act as a base for conducting such experiments.
- Dynamic analysis is performed to gather valuable information about malware activity, including the files and folders created, ports and URLs accessed, called functions and libraries, applications and tools accessed, information transferred, settings modified processes, and services the malware started, and other items. You should design and set up the environment for performing the dynamic analysis in such a way that the malware cannot propagate to the production network, and ensure that the testing system can recover to an earlier set timeframe (prior to launching the malware) in case anything goes wrong during the test.

To achieve this, you need to perform the following:

- System Baselining Baselining refers to the process of capturing a system's state (taking snapshot of the system) at the time the malware analysis begins. This can be used to compare the system's state after executing the malware file, which will help understand the changes that the malware has made across the system. A system baseline involves recording details of the file system, registry, open ports, network activity, and other items.
- Host Integrity Monitoring Host integrity monitoring is the process of studying the changes that have taken place across a system or a machine after a series of actions or incidents. It involves using the same tools to take a snapshot of the system before and after the incident or actions and analyzing the changes to evaluate the malware's impact on the system and its properties. In malware analysis, host integrity monitoring helps to understand the runtime behavior of a malware file as well as its activities, propagation techniques, URLs accessed, downloads initiated, and other characteristics.
    - Host integrity monitoring includes:
    - Port monitoring
    - Process monitoring
    - Registry monitoring
    - Windows services monitoring

- Startup program monitoring
- Event logs monitoring and analysis
- Installation monitoring
- Files and folder monitoring
- Device driver monitoring
- Network traffic monitoring and analysis
- DNS monitoring and resolution
- API calls monitoring

Perform Port Monitoring using TCPView and CurrPorts

- We know that the Internet uses a software protocol named TCP/IP to format and transfer data. Malware programs corrupt the system and open system input and output ports to establish connections with remote systems, networks, or servers to accomplish various malicious tasks. These open ports can also act as backdoors or communication channels for other types of harmful malware and programs. They open unused ports on the victim's machine to connect back to the malware handlers.
- You can identify the malware trying to access a particular port by installing port monitoring tools such as TCPView and CurrPorts.
- **TCPView** TCPView is a Windows program that shows the detailed listings of all the TCP and UDP endpoints on the system, including the local and remote addresses, and the state of the TCP connections. It provides a subset of the Netstat program that ships with Windows. The TCPView download includes Tcpvcon, a command-line version with the same functionality. When TCPView runs, it enumerates all active TCP and UDP endpoints, resolving all IP addresses to their domain name versions.
- **CurrPorts** CurrPorts is a piece of network monitoring software that displays a list of all the currently open TCP/IP and UDP ports on a local computer. For each port in the list, information about the process that opened the port is also displayed, including the process name, full path of the process, version information of the process (product name, file description, etc.), the time that the process was created, and the user that created it.
- In addition, CurrPorts allows you to close unwanted TCP connections, kill the process that opened the ports, and save the TCP/UDP port information to an HTML file, XML file, or to tab-delimited text file.

Perform Process Monitoring using Process Monitor

- Process monitoring will help in understanding the processes that malware initiates and takes over after execution. You should also observe the child processes, associated handles, loaded libraries, functions, and execution flow of boot time processes to define the entire nature of a file or program, gather information about processes running before the execution of the malware, and compare them with the processes running after execution. This method will reduce the time taken to analyze the processes and help in easy identification of all processes that malware starts.

- Process Monitor is a monitoring tool for Windows that shows the real-time file system, Registry, and process and thread activity. It combines the features of two legacy Sysinternals utilities, Filemon and Regmon. It adds an extensive list of enhancements including rich and non-destructive filtering, comprehensive event properties such session IDs and user names, reliable process information, full thread stacks with integrated symbol support for each operation, and simultaneous logging to a file. Unique features of Process Monitor make it a core utility in system troubleshooting and vital to the malware hunting toolkit.

Perform Registry Monitoring using Reg Organizer

- The Windows Registry stores OS and program configuration details such as settings and options. If the malware is a program, the registry stores its functionality. When an attacker installs a type of malware on the victim's machine, it generates a registry entry. One must have fair knowledge of the Windows Registry, its contents, and inner workings to analyze the presence of malware. Scanning for suspicious registries will help to detect malware. While most computer users generally do not do this, monitoring the registry entries is a great way to track any modifications made to your system.
- Registry monitoring tools such as Reg Organizer provide a simple way to track registry modifications, which is useful in troubleshooting and monitoring background changes.
- **Reg Organizer**
    - Reg Organizer is designed to edit keys and parameters, as well as to delete the content of.reg files. It allows users to perform various operations with the system registry such as export, import and copy key values. It can also perform a deep searches to find even those keys associated with the application that cannot be found by other similar programs.

Perform Startup Program Monitoring using Autoruns for Windows and WinPatrol

- Startup programs are applications or processes that start when your system boots up. Attackers make many malicious programs such as Trojans and worms in such a way that they are executed during startup, and the user is unaware of the malicious program running in the background.
- An ethical hacker or penetration tester must identify the applications or processes that start when a system boots up and remove any unwanted or malicious programs that can breach privacy or affect a system's health. Therefore, scanning for suspicious startup programs manually or using startup program monitoring tools like Autoruns for Windows and WinPatrol is essential for detecting malware.
- **Autoruns for Windows** This utility can auto-start the location of any startup monitor, display which programs are configured to run during system bootup or login, and show the entries in the order Windows processes them. As soon as this program is included in the startup folder, Run, RunOnce, and other Registry keys, users can configure Autoruns to show other locations, including Explorer shell extensions, toolbars, browser helper objects, Winlogon notifications, and auto-start services. Autoruns' Hide Signed Microsoft Entries option helps the user zoom in on third-party auto-starting images that add to the users'

system, and it has support for looking at the auto-starting images configured for other accounts configured on the system.

- **WinPatrol** provides the user with 14 different tabs to help in monitoring the system and its files. This security utility gives the user a chance to look for programs that are running in the background of a system so that the user can take a closer look and control the execution of legitimate and malicious programs.

Perform Installation Monitoring using Mirekusoft Install Monitor

- When the system or users install or uninstall any software application, there is a chance that it will leave traces of the application data on the system. Installation monitoring help to detect hidden and background installations that malware performs.
- **Mirekusoft Install Monitor** automatically monitors what gets placed on your system and allows you to uninstall it completely. Install Monitor works by monitoring what resources such as file and registry, are created when a program is installed. It provides detailed information about the software installed, including how much disk space, CPU, and memory your programs are using. It also provides information about how often you use different programs. A program tree is a useful tool that can show you which programs were installed together.

Perform Device Driver Monitoring using DriverView and Driver Reviver

- When the user downloads infected drivers from untrusted sources, the system installs malware along with the device drivers; malware uses these drivers as a shield to avoid detection. One can scan for suspicious device drivers using tools such as DriverView and Driver Reviver that verify if they are genuine and downloaded from the publisher's original site.
- DriverView The DriverView utility displays a list of all device drivers currently loaded on the system. For each driver in the list, additional information is displayed such as the load address of the driver, description, version, product name, and developer.
- Driver Reviver Without proper drivers, computers start to misbehave. Sometimes updating the drivers using conventional methods can be a daunting task. Outdated drivers are more vulnerable to hacking and can lead to a breach in the system. Driver Reviver provides an effective way of scanning your PC to identify out of date drivers. Driver Reviver can quickly and easily update these drivers to restore optimum performance to your PC and its hardware and extend its life.

Perform DNS Monitoring using DNSQuerySniffer

- DNSQuerySniffer is a network sniffer utility that shows the DNS queries sent on your system. For every DNS query, the following information is displayed: Host Name, Port Number, Query ID, Request Type (A, AAAA, NS, MX, and other types), Request Time, Response Time, Duration, Response Code, Number of records, and the content of the returned DNS records. You can easily export the DNS query information to a CSV, tab-delimited, XML, or HTML file, or copy the DNS queries to the clipboard and then paste them into Excel or another spreadsheet application.

🧠 Lesson 2.0 – Real-World Case Studies (Expanded)

> "The best way to understand malware is to learn from the chaos it caused."

🧨 Case Study 1: WannaCry Ransomware (2017)

📌 Overview:

WannaCry was a **global ransomware attack** that infected over **300,000 systems** in **150+ countries** in less than 3 days. It encrypted files and demanded payment in Bitcoin.

🧬 Key Technical Details:

- **Propagation Method**: Exploited a Windows SMBv1 vulnerability (CVE-2017-0144) known as **EternalBlue**, leaked by the Shadow Brokers.
- **Payload**: Dropped an encryption module that used RSA + AES encryption.
- **Ransom Demand**: ~$300–$600 in Bitcoin.
- **Killswitch Domain**: Hardcoded domain which, when registered by a researcher (Marcus Hutchins), stopped the spread temporarily.

🔍 Behavioral Indicators:

- Created files like `@Please_Read_Me@.txt`
- Modified file extensions to `.WNCRY`
- Made outbound connections over port **445**
- Changed the wallpaper with ransom instructions

📣 Lessons Learned:

- Regular patching is critical (Microsoft had released a patch 2 months before)
- Disable outdated protocols (SMBv1)
- Isolated backups are essential

🧨 Case Study 2: Emotet Malware

📌 Overview:

Emotet started as a **banking Trojan** in 2014 and evolved into a **modular malware-as-a-service (MaaS)** that delivered payloads like TrickBot and Ryuk ransomware.

🧬 Key Technical Details:

- **Delivery Method**: Malicious email attachments (Word docs with macros)

- **Persistence Mechanism**: Registry autoruns + scheduled tasks
- **Modules Included**:
  - Email stealer
  - Credential dumper
  - Lateral movement engine

🔍 Behavioral Indicators:

- Created random `.exe` files in `AppData\Local\Temp`
- Used PowerShell to download payloads
- Abused Windows Management Instrumentation (WMI) for persistence
- Connected to C2 servers using HTTP/HTTPS (often encrypted)

📣 Lessons Learned:

- Educate users on phishing attacks
- Disable macros by default
- Monitor for unusual PowerShell or WMI activity

🧨 Case Study 3: Agent Tesla

📌 Overview:

Agent Tesla is a .NET-based **Remote Access Trojan (RAT)** and **keylogger** active since 2014. It's popular in cybercrime markets for stealing credentials and screenshots.

🧬 Key Technical Details:

- **Delivery Method**: Usually via phishing emails (with attachments or links)
- **Capabilities**:
  - Keylogging
  - Clipboard capture
  - Screen capture
  - Credential theft from browsers, email clients, VPN apps

🔍 Behavioral Indicators:

- Injects into trusted processes like `RegAsm.exe`
- Sends data over SMTP or FTP to attacker-controlled servers
- Avoids detection using .NET obfuscation
- Frequently updates its code to bypass AV engines

📣 Lessons Learned:

- Application whitelisting blocks unknown programs
- Monitor outbound SMTP/FTP traffic
- Use network sandboxing for suspicious files

🧪 Module 3 Practicals – Hands-On Malware Analysis

> 🔐 All practicals must be performed in a virtual lab, fully isolated from the host machine.

🧰📦 Lab Setup Instructions (Once for All Practicals)

✅ Tools Required:

- VirtualBox or VMware
- Windows 7/10 VM
- **FlareVM** (Windows malware analysis suite)
- **REMnux** (Linux distro for malware reverse engineering)
- Tools inside lab:
  - **PEStudio**, **Wireshark**, **Process Monitor**, **RegShot**, **x64dbg**, **Cuckoo Sandbox**

🧪🧩 Practical 1 – Static Analysis of a Malware Sample

📂 Objective:

Analyze a malware file without executing it. Identify file structure, embedded strings, and possible intent.

⚙️ Steps:

1. **Download sample** from Malware Traffic Analysis or use a known test file like EICAR test file.
2. Open it in **PEStudio** and analyze:
   - Imports/Exports
   - Suspicious functions (e.g., `CreateRemoteThread`, `WinExec`)
   - Signature warnings
3. Run **Detect It Easy (DIE)**:
   - Check for packing or encryption

4. Extract strings using **Strings** or `strings` CLI:
    - Look for IPs, URLs, hidden commands

📌 Output:

- Screenshot of PEStudio findings
- List of interesting strings and why they matter

🧪🧩 Practical 2 – Registry Tracking with RegShot

📂 Objective:

Track what changes the malware makes to the Windows registry.

⚙️ Steps:

1. Take a snapshot of the registry using **RegShot** before running the malware.
2. Execute the malware in the VM.
3. Take a second snapshot using RegShot.
4. Compare both snapshots.

📌 Output:

- Log of registry keys added/modified
- Explanation of persistence mechanisms (e.g., startup entries)

🧪🧩 Practical 3 – Process Behavior Analysis

📂 Objective:

Observe what processes the malware spawns and interacts with.

⚙️ Steps:

1. Launch **Process Monitor (ProcMon)**.
2. Filter by the malware's `.exe` filename.
3. Execute the malware inside the VM.
4. Watch for:
    - File creation/modification
    - Registry editing
    - Process injection or new process spawning

📌 Output:

- Summary of malware behavior

- Screenshot of key actions

🧪🧩 Practical 4 – Network Traffic Analysis

📂 Objective:

Check if the malware connects to any IPs/domains (beaconing or C2).

⚙️ Steps:

1. Run **Wireshark** in the background.
2. Execute malware in the VM.
3. Filter traffic using:

   ```
   yaml
   CopyEdit
   ip.addr != your_local_ip
   ```

4. Look for:
   - DNS queries
   - HTTP POST/GET
   - Suspicious ports (e.g., 6667 for IRC)

📌 Output:

- List of contacted IPs
- Protocols used
- Any unencrypted data transfers

🧪🧩 Practical 5 – Behavior Analysis with Cuckoo Sandbox

> 🔥 Best suited for slightly advanced students

📂 Objective:

Automate dynamic analysis of malware behavior in a sandboxed environment.

⚙️ Steps:

1. Install **Cuckoo Sandbox** on a separate Ubuntu VM.
2. Configure guest agent on the Windows VM.
3. Upload a sample to Cuckoo.
4. Observe:

- File system changes
- Registry changes
- Process tree
- Network behavior

📌 Output:

- Cuckoo report (PDF or HTML summary)
- Screenshots of file drop locations or payload behavior

🧠 Bonus Challenge – Reverse Engineering

📂 Objective:

Open a sample malware binary in **Ghidra** or **IDA Free** and trace its logic.

> Look for:

- Entry point
- Obfuscated functions
- Hardcoded IPs/domains

📋 Deliverables Template for Students

Practical No	Tool Used	Malware Sample Name	Key Observations	Screenshot (Y/N)	Reflection
1	PEStudio	sample.exe	Found suspicious WinExec call	✅	Very informative
2	RegShot	sample.exe	Registry entry added to Run key	✅	Learned about persistence

🧠 Module 4 – Post-Practical Quiz (Malware Analysis)

10 Multiple-Choice Questions with Answers + Explanations

🔍 Practical 1: Static Analysis

Q1. Which tool is used to inspect PE headers and imports without executing the file?

A. ProcMon

B. PEStudio

C. Wireshark

D. x64dbg

✅ **Answer:** B – PEStudio

🧠 *Explanation:* PEStudio allows you to analyze Windows executable files statically without running them.

Q2. What does the presence of suspicious API calls like `CreateRemoteThread` indicate?

A. Network activity

B. Keylogging

C. Code injection behavior

D. File creation

✅ **Answer:** C – Code injection behavior

🧠 *Explanation:* Functions like `CreateRemoteThread` are often used by malware to inject code into other processes.

🧬 Practical 2: Registry Tracking

Q3. Which of the following tools helps you detect registry changes caused by malware?

A. RegShot

B. Wireshark

C. Cuckoo

D. Ghidra

✅ **Answer:** A – RegShot

🧠 *Explanation:* RegShot takes a before-and-after snapshot of the Windows Registry to detect changes.

Q4. A new registry key is found under `HKCU\Software\Microsoft\Windows\CurrentVersion\Run`. What does this suggest?

A. Malware will auto-update

B. Malware will open ports

C. Malware will persist on reboot

D. Malware is scanning the file system

✅ **Answer:** C – Malware will persist on reboot

🧠 *Explanation:* This registry key is commonly used to auto-launch programs at Windows startup.

🌐 Practical 3 & 4: Behavioral and Network Analysis

Q5. What tool shows live file, registry, and process events from running malware?

A. PEStudio

B. ProcMon

C. RegShot

D. x64dbg

✅ **Answer:** B – ProcMon

🧠 *Explanation:* Process Monitor gives a real-time log of system calls and resource activity.

Q6. In Wireshark, which protocol is most commonly seen during malware command and control (C2) communication?

A. SSH

B. HTTP/HTTPS

C. FTP

D. DNSSEC

✅ **Answer:** B – HTTP/HTTPS

🧠 *Explanation:* Malware often uses HTTP/HTTPS to blend with regular traffic and avoid detection.

Q7. You see repeated DNS queries to strange domain names like `xyu78djk.com`. This may indicate:

A. Legitimate traffic

B. C2 beaconing

C. Email phishing

D. OS patch download

✅ **Answer:** B – C2 beaconing

🧠 *Explanation:* Malware often uses DNS to check in with C2 servers at regular intervals.

🧪 Practical 5: Automated Sandbox Analysis

Q8. What is the main purpose of Cuckoo Sandbox in malware analysis?

A. Static binary reverse engineering

B. Obfuscation detection

C. Safe dynamic execution

D. Firewall inspection

✅ **Answer:** C – Safe dynamic execution

🧠 *Explanation:* Cuckoo Sandbox safely runs malware and captures its system/network behavior.

Q9. Which of these is NOT typically found in a Cuckoo report?

A. Process tree

B. Network activity

C. Decompiled source code

D. File system changes

✅ **Answer:** C – Decompiled source code

🧠 *Explanation:* Cuckoo gives behavioral data, not full reverse-engineered code.

🧠 Conceptual Understanding

Q10. Why is it important to analyze both static and dynamic aspects of malware?

A. It saves time

B. To get a full behavioral picture

C. Antivirus tools are always wrong

D. Malware doesn't need to be run

✅ **Answer:** B – To get a full behavioral picture

🧠 *Explanation:* Static tells what it *might* do, dynamic confirms what it *does*.

🧾 Scoring Guide:

- **0–4**: Revisit lab instructions & try again.
- **5–7**: Good job, but re-analyze key findings.
- **8–10**: Excellent! You're ready for advanced malware behavior analysis. 🧠🔥