Defense Evasion

In the defense evasion phase, **attackers use techniques to avoid detection, hide their presence, and maintain control over a compromised system**. These tactics allow them to bypass security mechanisms, avoid forensic analysis, and prolong their access within the target environment. Key methods of defense evasion include **disabling detection software, reverting VM/cloud instances, and techniques like process hollowing, injection, and bootkits**.

1. Disable Detection Software & Logging

- Definition: Attackers disable or modify security software and logging mechanisms to prevent the detection of their activities, making it harder for defenders to monitor and respond.
- Methods:
    - **Disabling Antivirus and EDR**: Attackers may stop or uninstall antivirus programs, endpoint detection and response (EDR) tools, or firewalls to eliminate alerts.
    - **Modifying or Deleting Logs**: Attackers can manipulate logs by turning off logging, deleting entries, or overwriting logs to cover their tracks.
- Security Implications: Disabling detection software reduces the visibility defenders have into the attacker's actions, making it easier for the attacker to proceed undetected. If logging is compromised, investigators lose critical evidence, complicating forensic analysis.

2. Revert VM/Cloud Instances

- Definition: In cloud and virtualized environments, **attackers may use snapshot and rollback features to revert a system to a previous state**, effectively erasing traces of their activities.
- Common Tactics:
    - **Reverting Virtual Machines (VMs)**: Attackers may revert VMs to snapshots taken before their malicious activity, removing any files, processes, or changes they made during the attack.
    - **Cloud Instance Rollbacks**: In cloud environments, attackers can use rollback or scaling mechanisms to revert instances to a clean state, effectively erasing signs of compromise.
- Security Implications: Reverting VMs or cloud instances makes it difficult for security teams to analyze and track the attacker's actions, as evidence can disappear with the rollback. It also disrupts the continuity of logs, leaving gaps in the timeline.

3. Process Hollowing & Injection

- Definition: **Process hollowing and injection are code injection techniques where attackers run malicious code within the memory space of a legitimate process, making it harder to detect**.
- Techniques:
    - **Process Hollowing**: Attackers **replace the code in a legitimate process with malicious code**, but the process retains its original name and appears legitimate in task managers and monitoring tools.
    - **DLL Injection**: Attackers inject malicious code into a running process **by loading a custom Dynamic Link Library (DLL) into the memory space of a target process**.
- Security Implications: These techniques allow attackers to disguise their malware as a legitimate system or application process, evading detection by antivirus and monitoring software. Security tools focusing on static process names may overlook these injected or hollowed processes.

4. Bootkits

- Definition: A bootkit is a **type of rootkit that modifies the boot process**, injecting malicious code into the system's bootloader or kernel, which then loads each time the system starts up.
- How Bootkits Work:
  - **Bootloader Manipulation**: Bootkits modify the bootloader to load malicious code early in the boot process, before security software is active.
  - **Kernel Modifications**: Bootkits inject code into the system kernel, which gives them high-level control over the operating system, allowing them to manipulate processes, files, and other system functions.
- Security Implications: Bootkits are **challenging to detect** and remove because they embed themselves in the earliest stages of the boot process, allowing them to evade traditional security solutions and persist even through system reboots.

Summary

- Disabling Detection Software & Logging prevents security alerts and forensic evidence collection, allowing attackers to operate with less risk of detection.
- Reverting VM/Cloud Instances enables attackers to erase evidence of their actions by reverting to previous system states, complicating forensic efforts.
- Process Hollowing & Injection hide malicious code within legitimate processes, evading security tools that rely on static process monitoring.
- Bootkits manipulate the boot process to embed malware in the kernel, making detection and removal extremely challenging.

By understanding these defense evasion techniques, security teams can implement stronger countermeasures, such as **monitoring for unusual process behavior, using integrity checks on boot processes, and enforcing strict permissions on logging and rollback capabilities in virtualized and cloud environments**.