Persistence

In the persistence phase of an attack, **attackers establish methods to maintain access to a compromised system, even after reboots, log-offs, or security measures**. Persistence techniques ensure that attackers can resume control without needing to re-exploit vulnerabilities, helping them stay in the system for extended periods. Here are some common persistence methods, including creating additional accounts, modifying start-up mechanisms, and using scheduled tasks.

1. Additional Accounts/Credentials

- Definition: Attackers may create new user accounts or add credentials to existing ones, ensuring they have authorized (but hidden) access to the system.
- How It Works:
  - **New Accounts**: Attackers with administrative access can create additional user accounts with high privileges, making it easier to regain access even if the primary entry point is discovered.
  - **Credential Dumping and Reuse**: Attackers may dump existing credentials from memory (e.g., through tools like Mimikatz) and store these for later use, or they may add new SSH keys or password hashes to retain access on Unix systems.
- Security Implications: New accounts and added credentials often blend in with legitimate users, making them challenging to detect. **Privileged accounts pose a particularly high risk**, as they grant attackers greater control over the system.

2. Start-Up/Log-On/Boot Scripts, Modify Launch Agents, DLL Side-Loading, Webshells

- **Start-Up/Log-On/Boot Scripts**:
  - Definition: Attackers modify scripts or create new ones that execute automatically during system start-up, user log-on, or boot processes.
  - Purpose: These scripts allow attackers to reinstate malware, execute commands, or establish a connection back to their servers whenever the system reboots or a user logs in.
  - Security Implications: Start-up and log-on scripts are easy to overlook, especially on complex systems with multiple start-up programs, and allow attackers to restart malicious processes after every reboot.
- **Modify Launch Agents (macOS)**:
  - Definition: On macOS, attackers can add or modify Launch Agents and Launch Daemons, which are files that control processes that start automatically.
  - Purpose: Modifying these agents allows attackers to launch malware invisibly in the background upon start-up.
  - Security Implications: Launch Agents are difficult to monitor without specialized tools, making them an effective way to establish persistence on macOS systems.
- **DLL Side-Loading (Windows)**:
  - Definition: DLL side-loading is a technique where attackers place a malicious DLL with the same name as a legitimate one in the application's directory, tricking the application into loading the malicious DLL instead.
  - Purpose: This technique enables malware to load automatically as part of a trusted application.
  - Security Implications: Since the malicious DLL is loaded as part of a legitimate process, it can evade detection by security tools that focus on standalone malware.

- **Webshells**:
    - Definition: A webshell is a script file (often PHP, ASP, or JSP) that allows attackers to control a compromised web server remotely via HTTP requests.
    - Purpose: **Webshells provide a persistent backdoor** into a server, allowing attackers to execute commands and upload/download files.
    - Security Implications: Webshells are often difficult to detect because they blend into the server's existing codebase, and attackers can use them to access the server remotely without needing interactive shells.

3. Scheduled Tasks

- Definition: Similar to the execution phase, attackers can set up or modify scheduled tasks to ensure their malicious code runs at specific intervals, during system events, or upon login.
- Purpose: Scheduled tasks give attackers a way to automate their scripts or malware to maintain presence, often running scripts regularly to reestablish backdoors or check for updates.
- Security Implications: Scheduled tasks are harder to detect once established, as they blend in with legitimate administrative tasks. If attackers schedule tasks under system or administrator privileges, it also helps them retain higher access levels.

Summary

- Additional Accounts/Credentials allow attackers to create backdoor accounts or manipulate existing ones, giving them easy re-entry points with legitimate credentials.
- Start-Up/Log-On Scripts, Launch Agents, DLL Side-Loading, and Webshells are powerful tools for executing code automatically during start-up, often allowing attackers to maintain access even after reboots.
- Scheduled Tasks enable attackers to automate malware execution, reinforcing persistence and allowing for regular or event-driven execution of malicious processes.

Together, these persistence techniques ensure attackers can continue to access compromised systems, even when direct access methods are removed. Monitoring for unusual accounts, scheduled tasks, and changes in start-up configurations can help detect and prevent unauthorized persistence methods.