Armaan Shamsaasef

Operating System Vulnerabilities

Operating System Vulnerabilities - SY0-701 - 2.3

Definition: Flaws or weaknesses within the core operating system (OS) software that can be exploited to compromise the security of a device.

1. Default Configurations

- Description: Operating systems often ship with pre-configured settings for ease of use, but these defaults can be insecure.
- The Problem: Default settings often include:
    - Default Usernames and Passwords (e.g., "admin/admin").
    - Unnecessary Enabled Services running in the background.
    - Overly Permissive Security Policies.
- Example: A new network router or an IoT device has a well-documented default username and password that the user never changes. An attacker can look up these defaults and easily gain access.

2. Weak or Misconfigured Security Settings

- Description: Security controls that are present but are not set to a strong enough level, often due to human error or a desire for convenience.
- The Problem: This creates security gaps that attackers can easily bypass.
- Examples:
    - Weak Passwords: Using simple, common, or short passwords.
    - Misconfigured Permissions: Setting file or share permissions to "Everyone - Full Control" instead of applying the principle of least privilege.
    - Disabled Security Features: Turning off the firewall or User Account Control (UAC) for convenience.

3. Unsecured Root Accounts

- Description: Failing to properly secure the built-in, all-powerful administrative account.
- The Problem: The root (Linux/macOS) or Administrator (Windows) account has unlimited system access. If compromised, the entire system is compromised.
- Best Practices:
    - Use a named user account for daily tasks and avoid using the root account directly.
    - For the root account, use an extremely strong, unique password.
    - Disable remote root login for SSH on servers.

4. Open Permissions

- Description: Granting users or system services more access rights than they need to perform their job functions.
- The Problem: Violates the Principle of Least Privilege. If a user account or service is compromised, the attacker inherits those excessive permissions.
- Example: A user's account is added to the "Administrators" group on a Windows PC just to install one piece of software. This now means any malware they accidentally run will also have full administrative privileges.

5. Unsecure APIs

- Description: Application Programming Interfaces (APIs) that allow applications to communicate with the OS or other services, but which lack proper security controls.
- The Problem: An attacker can exploit a vulnerable or poorly designed API to gain unauthorized access or execute commands.
- Example: A mobile app uses an OS API to access the device's contact list. If the API is unsecure, a malicious app could abuse it to steal that data without proper user consent.

6. Unencrypted Data

- Description: Storing sensitive data on the device in a plaintext, readable format.
- The Problem: If the device is lost, stolen, or compromised, the attacker can immediately read all the unencrypted data.

- Examples:
    - Storing passwords in a clear text file on the desktop.
    - A database on a server that stores credit card numbers without encryption.
    - Not using full-disk encryption (e.g., BitLocker, FileVault) on a laptop.

Summary

- OS vulnerabilities are often not "bugs" but misconfigurations and poor security practices.
- Key issues include weak defaults, excessive permissions, unsecured admin accounts, and unencrypted data.
- Mitigation requires hardening the OS: changing defaults, applying the principle of least privilege, using strong authentication, and enabling encryption.