Linux Security Fundamentals — A Must for Every Security Professional

1. Check for Elevated Permissions (SUID/SGID bits):
    - Files or directories with **SUID(**Set User ID) or **SGID**(Set Group ID)bits set (the 's' bit replacing 'x') run with elevated privileges.
    - Disable all unwanted SUID/SGID binaries to prevent privilege escalation attempts
2. Ensure sticky bits are set for shared directories:
    - Sticky bits are like lock on files within shared spaces. It adds security when set on directories ensuring only certain permitted users can modify or delete those files, even if others have access to the directory.
    - Useful for directories like /tmp or collaborative folders.
      Example permissions:
      drwxrwxrw**t** → Sticky bit set (t) means x is set
      drwxrwxrw**T** → Sticky bit set (T): all other users do not have execute (x) permissions- cannot see the contents nor run any programs in this directory
3. Network Access Control (NAC): ensures that only authorized and compliant devices are granted access to the network. NAC models include:

    - **Discretionary Access Control (DAC):** The file owner decides who can access it.
    - **Mandatory Access Control (MAC):** The system enforces access rules and not the owner of the file— more secure, less flexible.
    - **Role-Based Access Control (RBAC):** Permissions assigned to roles, not individuals — simplifies privilege management.

4. Use Linux Hardening Mechanism

    - **SELinux (Security Enhanced Linux):** Enforces mandatory access controls at the kernel level.
    - **AppArmor:** Restricts program capabilities based on per-application profiles.
    - **TCP Wrappers**: Controls access to network services based on client Ips. When a network request is made, TCP wrappers intercept it, checking the request against a list of allowed or denied IP addresses (though deprecated in newer distributions, still relevant in legacy systems. They are limited by the fact that they can only control access to services and not to ports.)

5. Enforce Least Privilege Principle:

    - Users and processes should only have the permissions they absolutely need. For example, if a user needs to run a command as root, then that command should be specified in the sudoers configuration instead of giving them full sudo rights.
6. Enable Network Time Protocol (NTP):
   Accurate timestamps are critical for log correlation, monitoring, and incident investigation. Ensure NTP or **chronyd** is enabled and synchronized
7. Disable insecure protocols:

- Turn off older or unused protocols such as: **X11 / XDMCP** (TCP 6001–6009, UDP 177 respectively) as these expose unnecessary network surfaces due to its unencrypted communication.

8. Secure SSH configuration:

   If SSH is enabled: Disable direct root login → PermitRootLogin no

   Disable password-based login → PasswordAuthentication no

   Use key-based authentication

   File: /etc/ssh/sshd_config

9. Use linux firewall (UFW) and/or iptables to restrict the traffics into/out of the host
10. Use Fail2Ban to Block Brute-Force Attacks
    - Fail2Ban monitors failed login attempts and bans IPs that exceed thresholds.
11. Enable logging and monitoring mechanism:
    - Kernel logs (/var/log/kern.log): Reveals the presence of vulnerable and outdated drivers, system crashes, resource limitations and other events that could lead to DOS.
    - System logs (/var/log/syslog):Contains system level events: service start and stop, login attempts, system reboots.
    - Authentication logs (/var/log/auth.log): Contains information about user authentication attempts (failed and successful attempts)
    - Application logs : contains information about activities of specific applications in the system. These logs are stored in their own files like /var/log/apache2/error.log for the Apache web server or /var/log/mysql/error.log for the MySQL database server.
12. Most importantly, keep your OS and installed packages up-to-date