**The Ultimate Guide to Linux Hardening: Boost Your System Security [HINDI]**

* Making your linux system more secure by turning off things you don’t need, limiting access, and adding extra layers of protection to keep it safe from hackers
* Say we have a linux server where we have deployed a website
* Now via internet everyone can access this website
* Now its possible that a hacker could try to hack the system
* Hackers usually hack the system that are less secure and try to access the site
* This is also called as vulnerability or weakness
* If a server has a weak point, then hackers use this point to access data
* For hardening we have to follow the below steps
  + Principle of least privilege(PoLP)
    - Grants only the minimum level of access necessary for users, processs, or programs to perform the tasks
    - Avoid running applications as root unless absolutely necessary
    - Utilize sudo to provide limited administrative privilages
  + Minimize attack surface
  + Keep the system updated
    - Regular patching – keep the OS, kernel, and installed software updated to protect against known vulnerabilities
    - Use tools like apt, yum or other package managers to update and automate patches as much as possible
    - Here “patches” are the **software-update files that fix bugs or security flaws in installed packages**.
  + Strong authentication and password policies
    - Enforce strong passwords by setting up password policies(/etc/security/pwquality.conf)
    - Use multi-factor authentication(MFA) for critical accounts to improve security
    - Lock user accounts after repeated failed login attempts(/etc/pam.d can be configured for this)
    - Use PAM(Pluggable Authentication Modules)
    - To manage authentication across different services and enforce security policies like strong password requirements and account lockout
  + Secure remote access
    - Disable root login via SSH – Update /etc/ssh/sshd\_config to set PermitRootLogin as No
    - Use key-based authentication - Instead of passwords, use SSH key pairs for remote access
    - Restrict SSH access to specific users and use a non-standard port to avoid common scanning attacks
  + Logging and monitoring
    - Enable logging – use tools like rsyslog, syslog-ng, or jourald to record system events
    - Log analysis – use intrusion detection tools like fail2ban or AIDE(Advanced intrusion detection environment) to monitor logs and detect suspicious activities, these tools will keeps checking if there are unauthorized access if yes it will ban the user)
    - Regular check system logs and automate alerts for unusual activities(eg /var/log/auth.log)
  + Firewall configuration
    - Use iptables/nftables/ufw : configure firewall rules to restrict inbound and outbound traffic to trusted sources only
    - Limit open ports to services that are essential and block unused ports
    - Create logging rules for unusual traffic attempts
  + File system security
    - Partitioning and mount options – use separate partitions for different parts of the file system like /var, /tmp and /home to limit damage if compromised
    - Use secure mount options such ad noexec, nosuid, and nodev, to prevent unauthorized execution of files especially in /tmp
    - Encrypt sensitive data – encrypt sensitive partitions or files using LUKS, eCryptfs, or similar tools to protect date from the rest
  + System auditing
    - Auditing with tools – use tools like auditd to monitor and log system activities, changes in key files, and user activities
    - Conduct periodic audits using tools like Lynis, openSCAP, or Tiger to ensure security compliance and identify weak points
  + Disable unused network services and ports
    - Run netstat, ss or similar tools to view listening ports and disable services that aren’t required
    - Close open ports using firewall setting s to limit network exposure
  + File and directory permissions
    - Correct ownership – ensure files and directories have proper ownership and permissions(chown, chmod) to prevent unauthorized access
    - Set umask values appropriately to prevent creating files with insecure default permissions
    - Use tools like find to search for and rectify permissions for sensitive files(eg /etc/passwd and /etc/shadow)
  + Kernel hardening
    - Sysctl settings – harden the kernel by modifying /etc/sysctl.conf
    - Important configurations include –
      * Disable IP packet forwarding (net.ipv4.ip\_forward = 0\_.
      * Prevent IP source routing(net.ipv4.conf.all.accept\_source\_rote = 0).
      * Enable SYN flood protection and disable ICMP redirection
    - Use security modules – use linux security modules like SELinux or AppArmor for enforcing mandatory access control policies
  + Use security tools and enhanced security applications
    - AppArmor/ SELinux - use these to enforce security policies that limit the actions that programs and services can perform
    - Antivirus/Anti-malware – use tools like ClamAV to scan for malware
    - Rootkit Detection – use chkrootkit or rkhunter to detect rootkits on the system
  + Data backup and recovery
    - Regular backup critical data using secure backup solutions
    - Ensure backups are encrypted and stored in a separate, secure location
    - Periodically test backup and recovery processes to ensure reliability
  + Physical security
    - Restrict physical access to the server using physical security measures like locked cabinets and biometrics
  + Secure boot process
    - Protect the GRUB boot loader with a password to prevent unauthorized modification of boot parameters
    - Restrict kernel boot parameters that could weaken system security
  + Network security configurations
    - TCP Wrappers – use /etc/hosts.allow and /etc/hosts.deny to allow or deny specific IP address
    - Segment the network and place sensitive systems in protected zones
    - Use VPN or other secure tunneling protocols for external access to the network
  + Intrusion detection and prevention
    - Implement IDS/IPS (eg Snort or Suricata) to monitor network traffic for malicious activity
    - Integrate host-based IDS like OSSEC to provide activity monitoring and alerting for critical files and directories
  + Authentication management
  + Reduce possible entry points
    - Disable unnecessary services – stop or disable services that are not needed to reduce the number of potential entry points
    - Remove unused software – remove applications, packages and libraries that are not in use, especially since unnecessary software can introduce vulnerabilities