# Linus Security Commands

**User and Group Management**

Managing users and groups is a critical aspect of Linux system administration. These commands allow you to create, modify, and delete user accounts, set passwords, and manage group memberships. Proper user and group management ensures secure access control and resource allocation within the system.

| Command | Description | Example |
|---|---|---|
| **passwd** | Change or set user password. | passwd user1<br>Changes the password for user named "user1". |
| **chpasswd** | Change passwords in bulk using a text file. | chpasswd < user_passwords.txt<br>Reads passwords from a file and sets them for respective users. |
| **chage** | Set aging properties for user passwords (expiration, warning period). | chage -M 90 user1<br>Sets maximum password age to 90 days for user "user1". |
| **useradd** | Create a new user account. | useradd -m user2<br>Creates a new user "user2" and creates a home directory for them. |
| **usermod** | Modify existing user account settings (e.g., change username, home directory, group membership). | usermod -l newuser olduser<br>Renames user "olduser" to "newuser". |
| **userdel** | Delete a user account. | userdel user2<br>Deletes the user account "user2". |

| Command | Description | Example |
| --- | --- | --- |
| **groupadd** | Create a new group. | groupadd group1<br>Creates a new group named "group1". |
| **groupmod** | Modify existing group settings (e.g., change group name or GID). | groupmod -n newgroup oldgroup<br>Renames group "oldgroup" to "newgroup". |
| **groupdel** | Delete a group. | groupdel group1<br>Deletes the group named "group1". |

**Privilege Management**

Privilege management commands enable administrators to temporarily elevate or switch user privileges, granting access to perform administrative tasks or run commands with elevated permissions. Tools like su and sudo facilitate controlled access to restricted operations, providing a secure and auditable way to manage system resources.

| Command | Description | Example |
| --- | --- | --- |
| **su** | Switch user (temporarily become another user). | su - username<br>Switches to the user named "username". |
| **sudo** | Execute commands with elevated privileges. | sudo command<br>Executes "command" with superuser privileges. |
| **visudo** | Edit the sudoers file safely. | sudo visudo<br>Opens the sudoers file for editing, ensuring syntax safety. |

**File and Directory Management**

These commands are essential for managing files and directories on a Linux system. They allow you to change file permissions, ownership, and default access rights, ensuring proper access control and security. Additionally, commands like ls provide a way to list and inspect file system contents and metadata.

| Command | Description | Example |
|---|---|---|
| **chmod** | Change file permissions. | chmod 644 file.txt<br>Sets read/write permissions for owner, read-only for group and others on "file.txt". |
| **chown** | Change file ownership. | chown user1 file.txt<br>Changes the owner of "file.txt" to "user1". |
| **chgrp** | Change file group ownership. | chgrp group1 file.txt<br>Changes the group ownership of "file.txt" to "group1". |
| **umask** | Set default file permissions for newly created files. | umask 077<br>Sets default permissions to deny access to group and others for newly created files. |
| **ls** | List directory contents. | ls -l<br>Lists files and directories in long format, including permissions, owner, group, and size. |

**Process Management**

Monitoring and managing processes is crucial for system administration. These commands provide insights into running processes, their resource usage, and network connections. Tools like ps, top, and lsof help identify and troubleshoot issues related to system performance, resource consumption, and potential security threats.

| Command | Description | Example |
| --- | --- | --- |
| **ps** | Display information about running processes. | ps aux<br>Lists all running processes on the system. |
| **top** | Display dynamic real-time information about running processes. | top<br>Displays a dynamic, real-time view of system processes and resource usage. |
| **netstat** | Display network connections, routing tables, interface statistics, masquerade connections, and multicast memberships. | netstat -tuln<br>Lists all listening TCP and UDP sockets along with their associated processes.ss |
| **ss** | A tool to investigate sockets. | ss -tuln<br>Similar to netstat, lists active sockets and related information. |
| **lsof** | List open files and the processes that opened them. | lsof /path/to/file<br>Displays processes that have opened the specified file. |

**Firewall and Security**

Linux offers various tools for configuring and managing firewalls, intrusion detection and prevention systems, auditing, and security policies. Commands like firewalld, fail2ban, and auditd help secure the system by controlling network traffic, detecting and mitigating malicious activities, and monitoring system events for security purposes.

| Command | Description | Example |
| --- | --- | --- |
| **firewalld** | Manage firewall rules (modern). | firewall-cmd --zone=public --add-port=80/tcp --permanent<br>Opens port 80 for TCP traffic permanently. |
| **fail2ban** | Intrusion prevention system that scans log files and bans IPs that show malicious signs. | fail2ban-client status<br>Displays the current status of Fail2Ban. |
| **auditd** | Linux audit daemon to monitor system calls and file system events. | auditctl -l<br>Lists the current audit rules. |
| **semanage** | SELinux policy management tool. | semanage fcontext -a -t httpd_sys_content_t '/web(/.*)?'<br>Adds a new SELinux file context for a web directory. |
| **getsebool** | Get the value of an SELinux boolean. | getsebool httpd_can_network_connect<br>Retrieves the value of the SELinux boolean "httpd_can_network_connect". |
| **setsebool** | Set the value of an SELinux boolean. | setsebool -P httpd_can_network_connect on<br>Sets the SELinux boolean "httpd_can_network_connect" to "on" persistently. |
| **sestatus** | Display SELinux status. | sestatus<br>Displays the current SELinux status, including mode, policy version, and status of SELinux modules. |

| Command | Description | Example |
| --- | --- | --- |
| **AppArmor** | Mandatory access control framework for restricting programs' capabilities. | aa-status<br>Shows the status of AppArmor and its enforced profiles. |
| **sysctl** | Configure kernel parameters at runtime. | sysctl -w net.ipv4.tcp_syncookies=1<br>Enables TCP SYN cookies to mitigate SYN flood attacks. |
| **ufw** | Uncomplicated Firewall - simplifies firewall configuration. | ufw allow ssh<br>Allows SSH traffic through the firewall. |

**Networking and Security Tools**

These commands encompass a wide range of networking and security-related utilities. From secure remote access with ssh to network scanning with nmap, encryption with openssl and gpg, and data transfer with curl and wget, these tools provide essential capabilities for secure communication, data protection, and network analysis.

| Command | Description | Example |
| --- | --- | --- |
| **ssh** | Secure Shell - remote login protocol. | ssh user@hostname<br>Initiates an SSH connection to the specified host as the specified user. |
| **openssl** | Tool to manage SSL/TLS certificates, create private keys, generate CSRs, and more. | openssl req -new -newkey rsa:2048 -nodes -keyout key.pem -out req.pem<br>Generates a new RSA key and CSR. |
| **gpg** | GNU Privacy Guard - encryption and signing tool. | gpg --encrypt --recipient recipient@example.com file.txt<br>Encrypts "file.txt" for recipient "recipient@example.com". |

| Command | Description | Example |
| --- | --- | --- |
| **sshd** | Secure Shell Daemon - configuration for SSH server. | sshd -t<br>Checks the syntax of the SSH server configuration file. |
| **nmap** | Network exploration tool and security scanner. | nmap -sV target_IP<br>Scans the target IP and displays version information of open ports. |
| **tcpdump** | Packet analyzer. | tcpdump -i eth0<br>Captures packets on interface "eth0". |
| **wireshark** | Network protocol analyzer. | wireshark<br>Opens the Wireshark GUI for packet analysis. |
| **curl** | Command-line tool for transferring data with URLs. | curl -O http://example.com/file.txt<br>Downloads "file.txt" from the specified URL. |
| **wget** | Command-line tool for retrieving files from the web. | wget http://example.com/file.txt<br>Downloads "file.txt" from the specified URL. |

**Security Scanners**

Linux offers several security scanning tools to detect and mitigate potential security threats. Commands like chkrootkit, rkhunter, and lynis are designed to scan the system for rootkits, backdoors, vulnerabilities, and provide hardening recommendations, helping to maintain a secure and hardened system.

| Command | Description | Example |
| --- | --- | --- |
| **chkrootkit** | Tool to locally check for signs of a rootkit infection. | chkrootkit<br>Scans the system for common rootkit signs and anomalies. |
| **rkhunter** | Rootkit Hunter - scanning tool to detect rootkits, backdoors, and local exploits. | rkhunter --check<br>Performs a comprehensive system scan for rootkits and vulnerabilities. |
| **lynis** | Security auditing and hardening tool. | lynis audit system<br>Initiates a system-wide audit for security vulnerabilities and hardening recommendations |