Linux Commands Reference Guide

This document provides a comprehensive reference for essential Linux commands relevant to the Google Cyber Security curriculum, suitable for inclusion in a GitHub repository.

Understanding File Permissions

File permissions are a foundational concept in Linux security. They determine who can read, write, or execute a file or directory.

Permissions are typically displayed in the format `drwxrwxrwx`, where:

- The first character indicates the file type (`d` for directory, `-` for a regular file).
- The next nine characters are grouped into three sets of three: Owner, Group, and Others.
- Each set defines Read (r), Write (w), and Execute (x) permissions.

Common Permission Representations

Permissions can be represented symbolically (rwx) or numerically (octal).

Permission	Symbolic	Octal Value
Read	r	4
Write	w	2
Execute	x	1

Permission	Symbolic	Octal Value
No Permission	-	0

The total numerical value for a user class is the sum of its permissions (e.g., Read + Write + Execute = 4 + 2 + 1 = 7).

`chmod` Command: Changing Permissions

The `chmod` command is used to change the access permissions of file system objects.

Action	Command Example	Description
Grant all permissions to the owner	`chmod u+rwx <span type="placeholder" placeholder-type="file"></span>`	Adds Read, Write, and Execute permissions for the User (Owner).
Remove write permission from the group	`chmod g-w <span type="placeholder" placeholder-type="file"></span>`	Removes Write permission for the Group.
Set permissions numerically	`chmod 755 script.sh`	Sets Owner: rwx, Group: rx, Others: rx.
Make a script executable for everyone	`chmod +x script.sh`	Adds Execute permission for User, Group, and Others.

Essential File System Commands

These commands are crucial for navigating and managing files within the Linux environment.

Command	Usage	Description
`ls`	`ls -l`	List directory contents, with long format (details).

Command	Usage	Description
`cd`	`cd /var/log`	Change directory to `/var/log`.
`pwd`	`pwd`	Print working directory.
`cat`	`cat system_log.txt`	Concatenate and display the content of a file.
`grep`	`grep "FAIL" /var/log/auth.log`	Search for a pattern ("FAIL") within a file.
`find`	`find / -name "*.conf"`	Search for files (ending in `.conf`) starting from the root directory.
`nano`	`nano file.txt`	Open the `nano` text editor to edit a file.

User and System Management Commands

Commands for managing users, groups, and system processes, which are fundamental to security administration.

Command	Usage	Description
`sudo`	`sudo apt update`	Execute a command with superuser (root) privileges.
`useradd`	`sudo useradd <span type="placeholder" placeholder-type="person"></span>`	Create a new user account.
`passwd`	`sudo passwd newuser`	Change the password for a user.
`tail`	`tail -f /var/log/syslog`	Output the last parts of a file, `-f` follows the output (useful for logs).

Command	Usage	Description
`ps`	`ps aux`	Display currently running processes.
`kill`	`kill 1234`	Send a signal (default: terminate) to a process with PID 1234.

Networking and Security Commands

Commands for network configuration, inspection, and basic security testing.

Command	Usage	Description
`ip`	`ip a`	Show network interface addresses.
`ping`	`ping google.com`	Test network connectivity to a host.
`netstat`	`netstat -tuln`	Display network connections, routing tables, and interface statistics. (Often replaced by `ss` or `ip` now).
`ssh`	`ssh <span type="placeholder" placeholder-type="person"></span>@<span type="placeholder" placeholder-type="place"></span>`	Secure Shell command to connect to a remote server.
`curl`	`curl -I example.com`	Transfer data from or to a server, `-I` shows only HTTP headers.

Cyber Security Specific Commands

This table highlights commands frequently used in a cyber security context, particularly for auditing and log analysis.

Command	Purpose in Security	Example Usage
`journalctl`	System log auditing (systemd)	`journalctl -u sshd.service -since "2026-01-18"`
`awk`	Log parsing and data manipulation	`cat auth.log | awk '$3 == "Failed" {print $1, $2, $11}'`
`tar`	Secure archive/backup creation	`tar -czvf backup_<span type="placeholder" placeholder-type="date"></span>.tar.gz /home/user/data`
`diff`	Comparing file integrity	`diff original_config.txt suspect_config.txt`