Linux Commands for Security Professionals

A comprehensive guide to Linux commands essential for developers, system administrators, and security professionals.

!!! tip "Learning Duration" **Week 1:** Basic commands for day-to-day activities
Week 2: Security-focused commands for pentesters and AppSec professionals

The Linux Philosophy

The power of Linux lies in combining simple tools:

```
# Example: Find failed SSH logins, extract IPs, count occurrences,
sort by frequency
grep "Failed password" /var/log/auth.log | awk '{print $11}' | sort
| uniq -c | sort -rn
```

Core & Essential Commands

These are the absolute fundamentals for navigating and interacting with a Linux system.

ls - List Directory Contents

Lists files and directories in the current location.

```
ls              # List contents of current directory
ls -la          # List with details including hidden files
ls -lh          # Human-readable format (KB, MB)
ls -lrth        # Time-sorted, latest at bottom
ls -R           # List recursively
```

cd - Change Directory

Navigates between directories.

```
cd /var/log     # Go to specific directory
cd -            # Go back to previous directory
cd ~ or cd      # Go to home directory
cd ..           # Go up one level
```

pwd - Print Working Directory

Shows the full path of your current directory.

```
pwd
# Output: /home/user/projects
```

cp - Copy Files

Copies files or directories.

```
cp source.txt destination.txt              # Copy file
cp /home/user/app.log /var/tmp/            # Copy to another
```

```
directory
cp -r /home/user/project /opt/backups/     # Copy directory
recursively
```

mv - Move or Rename Files

Moves or renames files and directories.

```
mv old_name.txt new_name.txt    # Rename file
mv file.log logs/               # Move file to directory
```

rm - Remove Files

Deletes files or directories. **Use with extreme caution - there is no undo!**

```
rm important_file.txt                  # Delete a file
rmdir old_directory                    # Delete empty directory
rm -rf /path/to/directory              # Delete directory
recursively (DANGEROUS!)
```

mkdir - Make Directory

Creates new directories.

```
mkdir new_folder                # Create single directory
mkdir -p parent/child/grandchild  # Create nested directories
```

touch - Create Empty File

Creates an empty file or updates timestamps.

```
touch newfile.txt               # Create empty file
touch -t 202301011200 file.txt  # Set specific timestamp
```

cat - Concatenate and Display

Displays file contents.

```
cat file.txt                    # Display file content
cat file1.txt file2.txt         # Display multiple files
cat > newfile.txt               # Create file with input (Ctrl+D to
save)
```

head & tail - View File Portions

```
head -n 20 file.txt             # Show first 20 lines
tail -n 50 file.txt             # Show last 50 lines
tail -f /var/log/syslog         # Follow log file in real-time
```

less & more - Page Through Files

```
less large_file.txt             # View with scroll (q to quit)
more large_file.txt             # View page by page
```

man - Manual Pages

The most important command for learning.

```
man ssh                          # View manual for ssh command
man -k keyword                   # Search manuals by keyword
```

System Administration & Monitoring

Commands for managing system resources, processes, and services.

ps - Process Status

Shows a snapshot of currently running processes.

```
ps aux                           # List all processes in detail
ps aux | grep sshd               # Find specific process
ps -ef --forest                  # Show process tree
```

top / htop - Process Viewer

Real-time, interactive view of system processes.

```
top                              # Basic process viewer (q to quit)
htop                             # Enhanced viewer (install: sudo
apt install htop)
```

kill - Terminate a Process

Sends signals to processes.

```
kill 1234                        # Gracefully stop process with PID
1234
kill -9 5678                     # Forcefully kill process (SIGKILL)
killall nginx                    # Kill all processes by name
```

df - Disk Free

Reports file system disk space usage.

```
df -h                            # Human-readable disk usage
df -i                            # Show inode usage
```

du - Disk Usage

Estimates file and directory space usage.

```
du -sh .                         # Summary of current directory size
du -ah /var | sort -rh | head -n 10  # Top 10 largest in /var
```

free - Memory Usage

```
free -h                          # Human-readable memory usage
free -m                          # Memory in MB
```

uptime - System Uptime

```
uptime                           # Show how long system has been
running
```

uname - System Information

```
uname -a                         # All system information
uname -r                         # Kernel version
```

systemctl - Systemd Control

Manages services on modern Linux systems.

```
sudo systemctl status nginx      # Check service status
sudo systemctl start sshd        # Start a service
sudo systemctl stop apache2      # Stop a service
sudo systemctl enable nginx      # Enable service on boot
sudo systemctl restart nginx     # Restart a service
```

journalctl - Query Systemd Logs

Views logs collected by systemd.

```
journalctl -f                    # Follow all logs in real-time
journalctl -u sshd               # Logs for specific service
journalctl -k                    # Kernel-level messages
journalctl --since "1 hour ago" # Logs from last hour
```

Text Processing & Automation

Power tools for manipulating text, searching logs, and scripting. **Essential for security professionals.**

grep - Global Regular Expression Print

Searches for patterns in text.

```
grep "error" app.log                        # Find "error" in file
grep -ri "API_KEY" .                        # Recursive,
case-insensitive search
grep -v "debug" app.log                     # Lines NOT containing
"debug"
grep -c "error" app.log                     # Count matches
grep -n "pattern" file.txt                  # Show line numbers
tail -f /var/log/syslog | grep -v "debug"  # Filter live logs
```

find - Find Files

Searches for files and directories based on various criteria.

```
find /etc -name "*.conf"                    # Find .conf files in
/etc
find / -type d -mtime -1                    # Directories modified
in 24h
find /var/www -user www-data -type f -exec chmod 644 {} \;  # Find
and execute
find . -size +100M                          # Files larger than
100MB
find . -name "*.log" -delete                # Find and delete
```

sed - Stream Editor

Performs text transformations.

```
sed 's/development/production/g' config.txt  # Replace text
sed -i 's/old/new/g' file.txt                # In-place replacement
sed '/DEBUG/d' app.log                       # Delete lines with
DEBUG
sed -n '10,20p' file.txt                     # Print lines 10-20
```

awk - Text Processing Language

Powerful pattern-scanning and processing.

```
awk '{print $1, $3}' data.txt               # Print columns 1 and 3
awk '/Failed password/ {print $11}' /var/log/auth.log  # Extract
failed IPs
awk -F: '{print $1}' /etc/passwd            # Use : as delimiter
awk 'NR>1 {print $0}' file.txt              # Skip first line
```

sort - Sort Lines

```
sort file.txt                   # Sort alphabetically
sort -n numbers.txt             # Sort numerically
sort -r file.txt                # Reverse sort
sort -u file.txt                # Sort and remove duplicates
```

uniq - Report or Omit Repeated Lines

```
sort file.txt | uniq            # Remove duplicates (requires
sorted input)
sort file.txt | uniq -c         # Count occurrences
sort file.txt | uniq -d         # Show only duplicates
```

wc - Word Count

```
wc -l file.txt                  # Count lines
wc -w file.txt                  # Count words
wc -c file.txt                  # Count bytes
```

cut - Remove Sections from Lines

```
cut -d: -f1 /etc/passwd         # Extract first field (delimiter :)
cut -c1-10 file.txt             # Extract first 10 characters
```

xargs - Execute Commands from Input

```
find . -name "*.tmp" -print0 | xargs -0 rm  # Delete found files
cat urls.txt | xargs -I {} curl {}          # Process each line
```

tee - Split Output

```
./run-tests.sh | tee /var/log/test.log      # Show and save output
echo "data" | sudo tee /etc/config.txt      # Write to protected
file
```

Networking & Security

The daily toolkit for network, system, and application security professionals.

ip / ifconfig - Network Configuration

```
ip addr show                    # Show all network interfaces
ip route show                   # Show routing table
ifconfig -a                     # Legacy: show all interfaces
```

netstat / ss - Network Statistics

```
sudo ss -tulpn                  # List all listening ports with
processes
ss -s                           # Socket statistics summary
netstat -an | grep LISTEN       # Legacy: listening ports
```

ping - Test Connectivity

```
ping -c 4 google.com            # Send 4 ping packets
ping -i 0.5 host.com            # Ping every 0.5 seconds
```

traceroute - Trace Packet Route

```
traceroute google.com           # Trace route to destination
traceroute -n google.com        # Without DNS resolution
```

nmap - Network Mapper

Only use on networks you are authorized to scan!

```
nmap 192.168.1.1                # Basic port scan
sudo nmap -sS -sV scanme.nmap.org  # Stealth SYN scan with version
detection
```

```
nmap -p 80,443 --script http-vuln* example.com  # Vulnerability
scripts
nmap -sn 192.168.1.0/24         # Ping sweep (host discovery)
nmap -A target.com              # Aggressive scan (OS, version,
scripts)
```

tcpdump - Packet Capture

```
sudo tcpdump -i eth0            # Capture on interface
sudo tcpdump -i any host 1.1.1.1 and port 53  # Filter by host and
port
sudo tcpdump -i eth0 -w capture.pcap          # Save to file for
Wireshark
sudo tcpdump -r capture.pcap    # Read from file
```

curl - Client for URLs

Essential for testing APIs and web security.

```
curl https://example.com                    # Fetch page content
curl -I https://example.com                 # View HTTP headers
only
curl -X POST -H "Content-Type: application/json" \
     -d '{"key":"value"}' https://api.example.com/submit  # POST
JSON
curl -o file.zip https://example.com/file.zip  # Download file
curl -u user:pass https://api.example.com   # Basic auth
```

wget - Download Files

```
wget https://example.com/file.zip           # Download file
wget -r -l 2 https://example.com            # Recursive download,
depth 2
wget -c https://example.com/large.iso       # Resume interrupted
download
```

dig - DNS Lookup

```
dig google.com                  # Query A record
dig @8.8.8.8 google.com MX      # Query specific DNS server for MX
records
dig +short google.com           # Short output
dig -x 8.8.8.8                  # Reverse DNS lookup
```

host / nslookup - DNS Queries

```
host google.com                 # Simple DNS lookup
nslookup google.com             # Interactive DNS lookup
```

whois - Domain Information

```
whois example.com               # Domain registration info
```

netcat (nc) - Network Swiss Army Knife

```
nc -zv host.com 80              # Port scan
nc -l -p 1234                   # Listen on port
nc host.com 1234                # Connect to port
nc -l -p 1234 > file.txt        # Receive file
nc host.com 1234 < file.txt     # Send file
```

Permissions & Access Control

Managing who can do what on the system.

chmod - Change Mode

Changes file permissions.

```
chmod u+x my_script.sh          # Make executable for owner
chmod 755 my_script.sh          # rwxr-xr-x (owner:rwx, group:r-x,
others:r-x)
chmod 644 file.txt              # rw-r--r-- (common for files)
chmod o-rwx sensitive_file.txt  # Remove all permissions for others
chmod -R 755 /var/www           # Recursive permission change
```

Permission Reference:

- r=4 (read), w=2 (write), x=1 (execute)
- 755 = rwxr-xr-x, 644 = rw-r--r--, 600 = rw-------

chown - Change Owner

```
sudo chown www-data /var/www/index.html     # Change owner
sudo chown -R admin:admin /opt/app          # Change owner and
group recursively
sudo chown :developers file.txt             # Change group only
```

sudo - Superuser Do

Execute commands with root privileges.

```
sudo nano /etc/hosts            # Edit protected file
sudo -i                         # Get root shell
sudo -u www-data whoami         # Run as different user
```

User & Session Management

who / w - Who is Logged In

```
who                             # Show logged-in users
w                               # Detailed info with current
```

```
commands
who am i                        # Your login info
```

whoami / id - User Identity

```
whoami                          # Show current username
id                              # Show user ID, group ID, groups
id -u                           # Just user ID
id -gn                          # Primary group name
```

last - Login History

```
last                            # History of logins
last -n 10                      # Last 10 logins
lastb                           # Failed login attempts (requires
root)
```

ssh - Secure Shell

```
ssh user@remote-host.com        # Connect to server
ssh user@host -p 2222           # Connect on different port
ssh -i ~/.ssh/key.pem user@host # Use specific key
ssh -L 8080:localhost:80 user@host  # Local port forwarding
```

!!! tip "Security Best Practice" Use key-based authentication instead of passwords for improved security.

scp / rsync - Secure Copy

```
scp localfile.txt user@host:/remote/dir/    # Copy file to remote
scp user@host:/remote/file.txt ./           # Copy from remote
scp -r folder/ user@host:/remote/           # Copy directory
rsync -avz --progress ./local/ user@host:/remote/  # Sync with
progress
rsync -avz --delete ./source/ ./backup/     # Mirror with delete
```

Archiving & Compression

tar - Tape Archive

```
tar -czvf archive.tar.gz /path/to/dir       # Create gzipped
archive
tar -xzvf archive.tar.gz                    # Extract gzipped
archive
tar -tvf archive.tar.gz                     # List contents without
extracting
tar -xzvf archive.tar.gz -C /destination    # Extract to specific
directory
```

Flags: c=create, x=extract, z=gzip, v=verbose, f=filename

zip / unzip

```
zip -r archive.zip folder/      # Create zip archive
unzip archive.zip               # Extract zip
unzip -l archive.zip            # List contents
```

gzip / gunzip

```
gzip file.txt                   # Compress (creates file.txt.gz)
gunzip file.txt.gz              # Decompress
```

File Analysis & Forensics

Crucial for security investigations and debugging.

file - Determine File Type

```
file unknown_binary             # Identify file type
file my_script                  # Check if script or binary
file *                          # Identify all files in directory
```

strings - Extract Text from Binaries

```
strings /usr/bin/sshd | grep "OpenSSH"      # Find text in binary
strings suspicious.exe | grep -E "http|ftp" # Look for URLs
```

diff - Compare Files

```
diff file1.txt file2.txt        # Show differences
diff -u old.conf new.conf       # Unified format (for patches)
```

md5sum / sha256sum - Calculate Hashes

```
sha256sum ubuntu.iso            # Generate SHA256 hash
sha256sum -c hashes.txt         # Verify against hash file
md5sum file.txt                 # Generate MD5 hash
```

lsof - List Open Files

```
sudo lsof -i :80                # Find process using port 80
sudo lsof -p 1234               # Files opened by PID 1234
sudo lsof -u username           # Files opened by user
lsof +D /var/log                # Files in directory
```

Environment & Variables

export - Set Environment Variables

```
export PATH=$PATH:/opt/tools    # Add to PATH
export EDITOR=vim               # Set default editor
export API_KEY="secret123"      # Set variable
```

env / printenv - Show Environment

```
env                             # Show all environment variables
printenv HOME                   # Show specific variable
echo $PATH                      # Print PATH variable
```

history - Command History

```
history                         # Show command history
history | grep ssh              # Search history
!100                            # Re-run command #100
!!                              # Re-run last command
```

Essential Security Commands Summary

For **pentesters and AppSec professionals**, master these beyond basics:

| Category | Commands | |----------|----------| | **Reconnaissance** | nmap, dig, whois, host, nslookup, traceroute | | **Network Analysis** | tcpdump, netstat, ss, netcat, curl, wget | | **Log Analysis** | grep, awk, sed, find, journalctl, tail -f | | **File Forensics** | file, strings, md5sum, sha256sum, diff | | **Permissions** | chmod, chown, sudo, ls -la | | **Process Management** | ps, top, kill, lsof |

Resources

Books

1. Linux Basics for Hackers - Recommended
2. The Linux Command Line
3. How Linux Works

Courses

1. Introduction to Linux Commands and Scripting
2. Linux Fundamentals for Security Practitioners - Recommended

Videos

1. Linux for Ethical Hackers - Recommended
2. 50 Most Popular Linux Commands

Reference

- Learn Linux Commands Repository