TP Krawl — Mise en place & tests (WSL2 + téléphone)

Version: 2026-02-14 • Contexte: Krawl lancé en local (port 5000) + dashboard web

But du TP

- Vérifier que le dashboard Krawl se charge
- Générer du trafic “attaque” (paths/UA/payloads) pour valider la collecte + la détection
- Accéder depuis un téléphone (vrai client réseau) quand Krawl tourne sous WSL2
- Faire un test de charge léger via script

0) Pré-requis

- Krawl en cours d’exécution (ex: `python3 ...`)
- Dashboard accessible: `http://localhost:5000/das_dashboard` (selon ton routing)
- Dans WSL: `curl` installé

Installer curl (si besoin)

```
sudo apt update && sudo apt install -y curl
```

1) Vérification réseau (WSL2 + Windows)

1.1 IPs côté WSL

```
ip a
```

Tu récupères l’IP WSL (interface `eth0`, ex: `172.29.x.x`) et tu sais que `127.0.0.1` = loopback.

1.2 IP Windows vue depuis WSL (gateway WSL)

```
cat /etc/resolv.conf | grep nameserver
```

Ça donne souvent l’IP Windows “interne WSL” (ex: `172.29.16.1`).

1.3 IP LAN Windows (celle que voit le téléphone)

Sur Windows (CMD):

```
ipconfig
```

Repère l’IPv4 de ta carte Wi-Fi/Ethernet (ex: `192.168.1.18`).

2) Vérifier que Krawl écoute (port 5000)

Dans WSL:

```
ss -lntp | grep 5000
```

Résultat attendu: `0.0.0.0:5000` (pas seulement `127.0.0.1:5000`)

3) Générer du trafic “attaque” en local

Dans WSL:

```
# scans classiques
curl -s -A "nikto/2.1.6"  "http://localhost:5000/.env"        >/dev/null
curl -s -A "sqlmap/1.7"    "http://localhost:5000/wp-login.php" >/dev/null
curl -s -A "masscan/1.3"   "http://localhost:5000/phpmyadmin/"  >/dev/null

# boucle multi-paths
for p in ".env" "admin" "wp-login.php" "phpmyadmin" "cgi-bin/" "actuator/health"; do
  curl -s -A "zgrab/0.1" "http://localhost:5000/$p" >/dev/null
done
```

Option: simuler plusieurs IPs via header

Utile si ton code exploite `X-Forwarded-For`.

```
curl -s "http://localhost:5000/.env" -H "X-Forwarded-For: 203.0.113.66" >/dev/null
curl -s "http://localhost:5000/.env" -H "X-Forwarded-For: 198.51.100.77" >/dev/null
```

4) Accès depuis le téléphone (WSL2 → LAN)

Problème classique: sous WSL2, Windows peut n’exposer le port que sur `127.0.0.1`. Résultat: ton téléphone voit “site inaccessible” sur `http://IP_WINDOWS:5000/…`.

4.1 Autoriser le port dans le pare-feu Windows

PowerShell (Admin):

```
New-NetFirewallRule -DisplayName "Krawl 5000" -Direction Inbound -Action Allow -Protocol TCP -LocalPort 5000
```

4.2 Vérifier où Windows écoute

```
netstat -ano | findstr :5000
```

Si tu vois seulement `127.0.0.1:5000 LISTENING`, ce n'est pas joignable depuis le LAN.

4.3 Publier le port sur le LAN via portproxy

PowerShell (Admin) — remplace `CONNECT_WSL_IP` par ton IP WSL (ex: 172.29.22.155):

```
netsh interface portproxy add v4tov4 listenaddress=0.0.0.0 listenport=5000 connectaddress=CONNECT_WSL_IP connectport=5000

netsh interface portproxy show v4tov4
netstat -ano | findstr :5000
```

4.4 Tester depuis le téléphone

Sur le téléphone (même Wi-Fi):

```
http://IP_WINDOWS_LAN:5000/wp-login.php
http://IP_WINDOWS_LAN:5000/.env
http://IP_WINDOWS_LAN:5000/phpmyadmin/
```

Note: avec portproxy, l'IP "remote" vue côté app peut être Windows/gateway WSL (ex: `172.29.16.1`). C'est normal.

5) Déclencher des "Attack Types" (détection)

```
BASE="http://localhost:5000"

# SQLi
curl -s "$BASE/search?q=' OR 1=1-- -" -A "sqlmap/1.7" >/dev/null
curl -s "$BASE/item?id=1%20UNION%20SELECT%201,2,3--" -A "sqlmap/1.7" >/dev/null

# XSS
curl -s "$BASE/?q=%3Cscript%3Ealert(1)%3C%2Fscript%3E" -A "Mozilla/5.0" >/dev/null

# LFI / path traversal
curl -s "$BASE/?file=../../../../etc/passwd" -A "curl/8" >/dev/null
curl -s "$BASE/..%2f..%2f..%2f..%2fetc%2fpasswd" -A "curl/8" >/dev/null

# probes connus
curl -s "$BASE/cgi-bin/..%2f..%2f..%2fbin%2fsh" -A "zgrab/0.1" >/dev/null
curl -s "$BASE/.git/config" -A "masscan/1.3" >/dev/null
curl -s "$BASE/solr/admin/info/system?wt=json" -A "zgrab/0.1" >/dev/null
```

Résultat attendu: la table "Detected Attack Types" se remplit (ex: `sql_injection`, `path_traversal`, `xss_attempt`, `lfi_rfi`, `xxe_injection`) et le graphe "Most Recurring..." monte.

6) Test volume (script trafic réaliste)

Créer un script de simulation (UA variés + chemins variés + IP random via `X-Forwarded-For`).

```
cat > simulate_attack.sh <<'EOF'
#!/usr/bin/env bash
set -euo pipefail

BASE="${1:-http://localhost:5000}"
DURATION="${2:-30}"
RATE="${3:-8}"

UAS=(
  "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/121.0 Safari/537.36"
  "Mozilla/5.0 (iPhone; CPU iPhone OS 17_2 like Mac OS X) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/17.2 Mobile/15E148 Safari/604.1"
  "sqlmap/1.6.4#stable (https://sqlmap.org)"
  "nikto/2.1.6"
  "zgrab/0.1"
  "curl/8.0"
  "masscan/1.3"
)

PATHS=(
  "/.env"
  "/.git/config"
  "/wp-login.php"
  "/phpmyadmin/"
  "/admin"
  "/cgi-bin/..%2f..%2f..%2fbin%2fsh"
  "/%2e%2e/%2e%2e/%2e%2e/%2e%2e/etc/passwd"
  "/?file=../../../../etc/passwd"
  "/?page=..%2F..%2F..%2F..%2Fwindows%2Fwin.ini"
  "/search?q=%27%20OR%201%3D1--%20-"
  "/item?id=1%20UNION%20SELECT%201,2,3--"
  "/?q=%3Cscript%3Ealert(1)%3C%2Fscript%3E"
  "/solr/admin/info/system?wt=json"
```

```
  "/?xml=%3C%21DOCTYPE%20x%20%5B%3C%21ENTITY%20xxe%20SYSTEM%20%22file%3A%2F%2F%2Fetc%2Fpasswd%22%3E%5D%3E%3Cx%3E%26xxe%3B%3C%2Fx%3E"
)

rand_ip() {
  local a b c d
  a=$(( (RANDOM % 3) + 198 ))
  b=$(( RANDOM % 255 ))
  c=$(( RANDOM % 255 ))
  d=$(( (RANDOM % 200) + 1 ))
  echo "${a}.${b}.${c}.${d}"
}

end=$((SECONDS + DURATION))

while (( SECONDS < end )); do
  ua="${UAS[RANDOM % ${#UAS[@]}]}"
  path="${PATHS[RANDOM % ${#PATHS[@]}]}"
  ip="$(rand_ip)"

  curl -sS "$BASE$path" \
    -A "$ua" \
    -H "X-Forwarded-For: $ip" \
    --max-time 2 >/dev/null || true

  sleep "$(awk -v r="$RATE" 'BEGIN { printf "%.3f", (1/r) }')"
done

echo "Done: sent traffic to $BASE for ${DURATION}s (~${RATE} rps)."
EOF

chmod +x simulate_attack.sh
```

Lancer le script

```
./simulate_attack.sh http://localhost:5000 30 8
```

Ou via Windows LAN (si tu veux passer par `192.168.1.18`): `./simulate_attack.sh http://192.168.1.18:5000 30 8`

7) Check final (validation)

- Dashboard OK
- Attackers / Requests montent
- Captured Credentials remonte (si endpoints de login/honeypot)
- Detected Attack Types remplit + graphe “Most Recurring…”

Note “photos”: si tu veux intégrer les captures dans ce PDF, envoie-les en fichiers (ou indique leur chemin) et je te régénère le PDF avec les images au bon endroit.