AYI NEDJIMI Consultants

OSINT et Reconnaissance Offensive : Du Renseignement

Catégorie : Techniques de Hacking Lecture : 9 min Publié le : 08/03/2026 Auteur : Ayi NEDJIMI

Guide complet OSINT pour le pentest : cycle du renseignement, outils (Shodan, Maltego, theHarvester, Recon-ng), Google Dorks, reconnaissance passive.

Avertissement : Les techniques présentées dans cet article sont destinées exclusivement à des fins éducatives et de tests autorisés. Toute utilisation malveillante est illégale et contraire à l'éthique professionnelle.

Cadre légal en France

En France, l'OSINT s'inscrit dans un cadre juridique précis. La collecte d'informations publiquement accessibles est légale, mais certaines pratiques peuvent franchir la ligne rouge :

- L'**article 323-1 du Code pénal** sanctionne l'accès ou le maintien frauduleux dans un système d'information (jusqu'à 3 ans d'emprisonnement et 100 000 euros d'amende).
- Le **RGPD** encadre la collecte de données personnelles, même si elles sont publiques. Le traitement doit avoir une base légale (intérêt légitime dans le cadre d'un audit mandaté).
- La **loi Godfrain** (loi n°88-19 du 5 janvier 1988) constitue le fondement de la répression de la criminalité informatique.
- Un **mandat d'audit** (lettre de mission) est indispensable avant toute opération de reconnaissance active sur une cible.

La reconnaissance passive (consultation de données publiques sans interaction directe avec les systèmes de la cible) reste généralement dans le cadre légal. La reconnaissance active (scanning, enumeration) nécessite une autorisation explicite.

Selon le rapport **SANS 2025 sur le pentest**, les équipes qui investissent plus de 30% de leur temps dans la phase de reconnaissance obtiennent un taux de compromission 2,7 fois supérieur à celles qui négligent cette étape. L'OSINT n'est pas un luxe ; c'est la différence entre un pentest superficiel et une opération qui reflète les méthodes réelles des attaquants poussés.

Cet article propose un guide exhaustif de l'OSINT appliqué au pentest : du cycle théorique du renseignement aux outils pratiques, en passant par les techniques de reconnaissance passive et active, les workflows automatisés et les contre-mesures défensives. Les concepts présentés s'articulent avec d'autres techniques offensives abordées dans nos articles sur les attaques d'API, les attaques DNS ou le phishing avancé.

Phase 1 : Planification et Orientation

La planification est la phase la plus critique et souvent la plus négligée. Elle consiste à définir les **Priority Intelligence Requirements** (PIR) : quelles questions le renseignement doit-il permettre de répondre ? Dans le contexte d'un pentest, ces questions peuvent inclure :

- Quels sont les domaines, sous-domaines et plages IP associés à l'organisation ?
- Quelles technologies (CMS, frameworks, serveurs) sont utilisées en production ?
- Quels employés ont accès aux systèmes critiques et quelles sont leurs habitudes numériques ?
- Existe-t-il des fuites de données (credentials, documents internes) accessibles publiquement ?
- Quelle est la maturité sécurité de l'organisation (présence d'un SOC, WAF, EDR) ?

La définition du **scope** est déterminante. Un scope trop large dispersera les efforts ; un scope trop étroit risque de manquer des vecteurs d'attaque critiques. Le document de cadrage doit préciser les domaines autorisés, les adresses IP incluses et exclues, les filiales concernées et les limites légales de la mission. Cette phase produit un **plan de collecte** qui guidera les phases suivantes.

Phase 2 : Collecte

La collecte est la phase opérationnelle où les données brutes sont rassemblées à partir de multiples sources. En OSINT, les sources se catégorisent en plusieurs familles :

- **Sources techniques** : DNS, WHOIS, certificats SSL/TLS (Certificate Transparency), Shodan, Censys, ZoomEye.
- **Sources web** : moteurs de recherche (Google Dorks), archives web (Wayback Machine), cache.
- **Sources humaines** : réseaux sociaux (LinkedIn, Twitter/X, GitHub), forums spécialisés.
- **Sources de fuites** : bases de données compromises, paste sites, dark web marketplaces.
- **Sources passives** : metadata de documents, enregistrements historiques, OSINT gouvernemental.

L'efficacité de la collecte repose sur la **diversification des sources** et la **systématisation du processus**. Un pentester expérimenté ne se contente pas d'un seul outil : il croise les résultats de multiples sources pour construire une image complète et fiable.

Votre surface d'attaque externe est-elle réellement celle que vous imaginez ?

Les **Google Dorks** exploitent les opérateurs avancés de recherche Google pour découvrir des informations sensibles indexées par le moteur. Cette technique, également connue sous le nom de *Google Hacking*, reste l'un des vecteurs OSINT les plus sous-estimés et les plus efficaces.

Voici 20 dorks essentiels pour la reconnaissance offensive :

Catégorie	Google Dork	Objectif
Fichiers sensibles	`site:example.com filetype:pdf`	Documents PDF (organigrammes, rapports)
Configuration	`site:example.com filetype:env`	Fichiers .env exposés (credentials)
Configuration	`site:example.com filetype:xml inurl:sitemap`	Sitemaps (cartographie du site)
Backups	`site:example.com ext:sql | ext:bak | ext:old`	Fichiers de sauvegarde exposés
Panels admin	`site:example.com inurl:admin | inurl:login`	Pages d'administration
Erreurs	`site:example.com intitle:"Index of /"`	Directory listing activé
API	`site:example.com inurl:api | inurl:swagger`	Documentation API exposée
Git	`site:example.com inurl:.git`	Dépôts Git exposés
WordPress	`site:example.com inurl:wp-content | inurl:wp-includes`	Détection WordPress (voir notre article hacking WordPress)
Logs	`site:example.com filetype:log`	Fichiers de logs exposés
Emails	`site:example.com intext:"@example.com"`	Adresses email dans les pages
Credentials	`site:example.com intext:"password" filetype:txt`	Mots de passe dans des fichiers texte
Errors	`site:example.com "SQL syntax" | "mysql_fetch"`	Erreurs SQL exposées
Cloud	`site:s3.amazonaws.com "example"`	Buckets S3 associés
Cloud	`site:blob.core.windows.net "example"`	Azure Blobs associés
Pastebin	`site:pastebin.com "example.com"`	Données sur paste sites
GitHub	`site:github.com "example.com" password | secret | key`	Secrets dans le code source
Conf	`site:example.com intitle:"phpinfo()"`	Pages phpinfo exposées
Jenkins	`site:example.com intitle:"Dashboard [Jenkins]"`	Instances Jenkins exposées
Camera	`site:example.com inurl:"/view.shtml"`	Caméras IP accessibles

Attention : utilisation éthique des Google Dorks
Les Google Dorks ne doivent être utilisés que dans le cadre d'un audit autorisé. L'accès à des données sensibles découvertes via dork, même si elles sont publiquement indexées, peut constituer un accès frauduleux si vous n'avez pas de mandat. Documentez systématiquement vos découvertes et signalez-les au client dans le cadre du responsible disclosure.

3.4. Shodan, Censys et ZoomEye

Les moteurs de recherche spécialisés dans l'indexation des services Internet constituent une source d'information majeure pour la reconnaissance passive. Contrairement aux scanners actifs, l'interrogation de ces moteurs ne génère aucun trafic vers la cible.

Le bruteforcing de répertoires et de fichiers découvre des ressources non référencées : pages d'administration, backups, fichiers de configuration, endpoints API cachés. Cette technique est complémentaire à l'analyse des URLs historiques issue de la reconnaissance passive.

```
# ffuf - Le plus rapide et flexible
ffuf -w /usr/share/wordlists/seclists/Discovery/Web-Content/raft-large-directories.txt \
  -u https://target.com/FUZZ -mc 200,301,302,403 -fc 404 \
  -H "User-Agent: Mozilla/5.0" -t 50 -o ffuf_results.json

# Gobuster - Bruteforce de répertoires
gobuster dir -u https://target.com -w /usr/share/wordlists/dirb/big.txt \
  -t 50 -x php,asp,aspx,jsp,html,js,txt,bak -o gobuster_results.txt

# Feroxbuster - Récursif et intelligent
feroxbuster -u https://target.com -w /usr/share/seclists/Discovery/Web-Content/raft-
medium-directories.txt \
  --depth 3 --threads 50 --collect-backups --collect-extensions

# Recherche de fichiers sensibles spécifiques
ffuf -w /usr/share/seclists/Discovery/Web-Content/quickhits.txt \
  -u https://target.com/FUZZ -mc 200 -fc 404
```

4.4. API Discovery

La découverte d'API est devenue un enjeu majeur avec la prolifération des architectures microservices. Les endpoints API non documentés ou mal sécurisés constituent une surface d'attaque considérable, comme le détaille notre article sur les attaques d'API GraphQL et REST.

```
# Découverte de documentation API
ffuf -w /usr/share/seclists/Discovery/Web-Content/api/api-endpoints.txt \
  -u https://target.com/FUZZ -mc 200

# Fichiers Swagger/OpenAPI
for path in swagger.json openapi.json api-docs swagger/v1/swagger.json; do
  curl -s -o /dev/null -w "%{http_code} %{url_effective}\n" \
    "https://target.com/$path"
done

# Introspection GraphQL
curl -s -X POST https://target.com/graphql \
  -H "Content-Type: application/json" \
  -d '{"query": "{__schema{types{name,fields{name,type{name}}}}}"}'

# Kiterunner - Découverte avancée d'endpoints API
kr scan https://target.com -w routes-large.kite -x 5 --fail-status-codes 404,503
```

4.5. Email Harvesting

La collecte d'adresses email est essentielle pour les tests de phishing et la vérification de fuites de données. Les adresses email suivent souvent un pattern prévisible (`prenom.nom@domaine.com`) qui permet d'extrapoler l'ensemble des adresses de l'organisation.

```
# theHarvester - L'outil classique
theHarvester -d example.com -b all -l 500 -f theharvester_results

# Hunter.io API
curl -s "https://api.hunter.io/v2/domain-search?domain=example.com&api_key=KEY" | \
  jq '.data.emails[].value'

# Phonebook.cz (via API)
curl -s "https://phonebook.cz/api/v1/search?q=example.com&type=email"

# crosslinked - Extraction LinkedIn pour générer des emails
crosslinked -f '{first}.{last}@example.com' 'Example Corp'

# Vérification de la validité des emails
# smtp-user-enum, EmailHarvester, Holehe (réseaux sociaux)
```

SpiderFoot est un outil d'automatisation OSINT qui interroge simultanément plus de 200 sources de données. Il se distingue par son interface web intuitive, sa capacité à corréler automatiquement les résultats et sa fonctionnalité de *scan profiles* prédéfinis.

- **Scan passif** : n'interagit pas avec la cible, utilise uniquement des sources tierces.
- **Scan actif** : inclut le scanning réseau, le fingerprinting et l'analyse active.
- **Scan exhaustif** : combine passif et actif pour une couverture maximale.
- **API intégrée** : s'intègre facilement dans des pipelines d'automatisation.

5.4. Amass : le moteur de reconnaissance

OWASP Amass est le standard de facto pour l'énumération de sous-domaines et la cartographie d'infrastructure. Son architecture multi-sources (50+ intégrations passives) et sa capacité de résolution DNS à grande échelle en font l'outil le plus complet dans sa catégorie.

```
# Configuration (config.ini)
# Définir les clés API pour maximiser les résultats
# SecurityTrails, Censys, Shodan, VirusTotal, PassiveTotal...

# Enumération passive complète
amass enum -passive -d example.com -config config.ini -o amass_passive.txt

# Enumération active avec brute-force
amass enum -active -brute -d example.com -w subdomains-top1million-5000.txt \
  -config config.ini -o amass_active.txt

# Visualisation de l'infrastructure
amass viz -d example.com -dot amass_graph.dot
# Puis : dot -Tpng amass_graph.dot -o infrastructure_map.png

# Tracking des changements dans le temps
amass track -d example.com -config config.ini
```

5.5. Tableau comparatif des outils

Outil	Type	Passif	Actif	Interface	Forces
Maltego	Framework	Oui	Oui	GUI	Visualisation, corrélation
Recon-ng	Framework	Oui	Oui	CLI	Modularité, scripting
SpiderFoot	Scanner	Oui	Oui	Web/CLI	200+ sources, automatisation
Amass	Enumérateur	Oui	Oui	CLI	Sous-domaines, graphes
theHarvester	Collecteur	Oui	Non	CLI	Emails, sous-domaines
Shodan CLI	Moteur	Oui	Non	CLI/Web	Services exposés, IoT
Censys	Moteur	Oui	Non	CLI/Web	Certificats, IPv4
FOCA	Metadata	Oui	Non	GUI	Extraction metadata docs
Metagoofil	Metadata	Oui	Non	CLI	Metadata, utilisateurs

Plusieurs frameworks open source intègrent l'ensemble du pipeline de reconnaissance dans un outil unique, automatisant les étapes de collecte, enrichissement, scanning et reporting.

ReconFTW

ReconFTW est l'un des frameworks les plus complets. Il orchestre plus de 30 outils dans un workflow cohérent et produit un rapport structuré. Sa configuration par fichier YAML permet d'adapter le pipeline à chaque mission.

```
# Lancement d'une reconnaissance complète
./reconftw.sh -d example.com -r -o /tmp/recon_results/

# Mode sous-domaines uniquement
./reconftw.sh -d example.com -s

# Mode full avec notification Slack
./reconftw.sh -d example.com -r --notify
```

LazyRecon

LazyRecon se distingue par sa simplicité et sa rapidité de déploiement. Il automatise l'énumération de sous-domaines, le scanning de ports, le fingerprinting web et la recherche de vulnérabilités dans un script unique.

Osmedeus

Osmedeus apporte une couche de sophistication supplémentaire avec son système de workflows modulaires, son interface web de monitoring et sa capacité de distribution sur plusieurs machines via **Axiom**.

6.3. Scaling avec Axiom

Axiom est un framework de distribution qui permet de lancer des outils de reconnaissance sur une flotte de VPS cloud. Au lieu d'exécuter un scan depuis une seule machine, Axiom le distribue sur 10, 50 ou 100 instances, réduisant considérablement le temps d'exécution et diversifiant les adresses IP sources.

```
# Créer une flotte de 20 instances
axiom-fleet create -i 20 -t default

# Distribuer un scan subfinder
axiom-scan example.com -m subfinder -o subfinder_distributed.txt

# Distribuer un scan nuclei
axiom-scan urls.txt -m nuclei -t cves/ -o nuclei_distributed.txt

# Détruire la flotte
axiom-fleet destroy
```

6.4. Intégration CI/CD pour le bug bounty

Les bug bounty hunters les plus performants intègrent leurs pipelines de reconnaissance dans des systèmes de CI/CD (GitHub Actions, GitLab CI) pour automatiser le monitoring continu de leurs cibles. Cette approche permet de détecter les changements de surface d'attaque (nouveaux sous-domaines, nouveaux services, nouvelles vulnérabilités) dès qu'ils apparaissent.

Phase d'énumération avancé

```
# .github/workflows/recon.yml
name: Automated Reconnaissance
on:
  schedule:
    - cron: '0 6 * * *'  # Chaque jour à 6h
  workflow_dispatch:

jobs:
  recon:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v4

      - name: Install tools
        run: |
          go install github.com/projectdiscovery/subfinder/v2/cmd/subfinder@latest
          go install github.com/projectdiscovery/httpx/cmd/httpx@latest
          go install github.com/projectdiscovery/nuclei/v3/cmd/nuclei@latest

      - name: Subdomain enumeration
        run: subfinder -dL targets.txt -all -silent -o new_subs.txt

      - name: Compare with previous results
        run: |
          comm -13 <(sort previous_subs.txt) <(sort new_subs.txt) > diff_subs.txt
          if [ -s diff_subs.txt ]; then
            echo "NEW_SUBS=true" >> $GITHUB_ENV
          fi

      - name: Scan new subdomains
        if: env.NEW_SUBS == 'true'
        run: |
          cat diff_subs.txt | httpx -silent | nuclei -t cves/ -severity critical,high -o
alerts.txt

      - name: Notify
        if: env.NEW_SUBS == 'true'
        run: |
          cat alerts.txt | notify -provider-config config/notify.yaml
```

6.5. Scripting Python pour l'OSINT

Python reste le langage de prédilection pour le développement d'outils OSINT personnalisés. Les bibliothèques clés incluent :

```
#!/usr/bin/env python3
"""
Script OSINT personnalisé - Enumération et enrichissement
"""
import requests
import json
import dns.resolver
from concurrent.futures import ThreadPoolExecutor

class OSINTRecon:
    def __init__(self, domain):
        self.domain = domain
        self.subdomains = set()
        self.results = {}

    def crtsh_enum(self):
        """Enumération via Certificate Transparency Logs"""
        url = f"https://crt.sh/?q=%25.{self.domain}&output=json"
        try:
            resp = requests.get(url, timeout=30)
            data = resp.json()
            for entry in data:
                names = entry.get('name_value', '').split('\n')
                for name in names:
                    name = name.strip().lstrip('*.')
                    if name.endswith(self.domain):
                        self.subdomains.add(name)
        except Exception as e:
            print(f"[!] crt.sh error: {e}")

    def resolve_subdomains(self):
        """Résolution DNS des sous-domaines découverts"""
        resolver = dns.resolver.Resolver()
        resolver.timeout = 3

        def resolve(sub):
            try:
                answers = resolver.resolve(sub, 'A')
                ips = [str(r) for r in answers]
                return sub, ips
            except:
                return sub, []

        with ThreadPoolExecutor(max_workers=20) as executor:
            futures = {executor.submit(resolve, sub): sub
                       for sub in self.subdomains}
            for future in futures:
                sub, ips = future.result()
                if ips:
                    self.results[sub] = ips

    def run(self):
        print(f"[*] OSINT Recon pour {self.domain}")
        self.crtsh_enum()
        print(f"[+] {len(self.subdomains)} sous-domaines trouvés")
        self.resolve_subdomains()
        print(f"[+] {len(self.results)} sous-domaines résolus")
        return self.results

if __name__ == "__main__":
    recon = OSINTRecon("example.com")
```

```
    results = recon.run()
    print(json.dumps(results, indent=2))
```

7. Contre-Mesures et Réduction de Surface d'Attaque

Comprendre l'OSINT offensif permet de mieux se défendre. La réduction de la surface d'attaque est un processus continu qui nécessite une vigilance permanente et une approche proactive.

Stratégies de réduction de la surface OSINT

- **Monitoring de sa propre surface d'attaque** : utilisez les mêmes outils que les attaquants (Amass, Shodan monitoring, Certificate Transparency monitoring) pour surveiller en permanence votre périmètre exposé. Les plateformes ASM (Attack Surface Management) comme Randori, Detectify ou ProjectDiscovery Cloud automatisent ce processus.
- **Réduction du footprint DNS** : supprimez les enregistrements DNS obsolètes (sous-domaines pointant vers des services désactivés = risque de subdomain takeover), consolidez les sous-domaines, utilisez des noms non descriptifs pour les services internes.
- **Suppression des métadonnées** : strippez systématiquement les métadonnées EXIF, XMP et IPTC des images et documents avant publication. Utilisez `exiftool -all= document.pdf` ou intégrez la suppression dans vos pipelines de déploiement.
- **Politiques réseaux sociaux** : formez les employés aux risques OSINT sur LinkedIn (ne pas lister les technologies internes sensibles), limitez les informations techniques dans les offres d'emploi, encadrez les publications techniques sur les réseaux sociaux.
- **Protection des secrets** : implémentez des outils de détection de secrets dans les pipelines CI/CD (GitLeaks, truffleHog, git-secrets) et activez la protection des branches sur les dépôts GitHub/GitLab. Voir notre guide sur le secrets sprawl.
- **Configuration DNS sécurisée** : désactivez les transferts de zone non autorisés, implémentez DNSSEC, utilisez des services DNS avec protection DDoS et rate limiting.
- **Gestion des certificats** : utilisez des certificats wildcard plutôt que des certificats individuels pour chaque sous-domaine (réduction de l'exposition dans les CT Logs), et désactivez l'émission automatique de certificats pour les environnements de développement.

7.1. Attack Surface Management (ASM)

Les solutions ASM représentent l'évolution défensive de l'OSINT. Elles reproduisent en continu les méthodes de reconnaissance des attaquants pour identifier proactivement les expositions. Les principales plateformes incluent :

- **Randori** (IBM) : émule la perspective de l'attaquant pour identifier et prioriser les surfaces d'attaque les plus attractives.
- **Detectify** : combine l'expertise de la communauté bug bounty avec un scanning automatisé pour détecter les vulnérabilités exposées.
- **CrowdStrike Falcon Surface** : intègre la threat intelligence avec la cartographie de surface d'attaque pour contextualiser les risques.
- **ProjectDiscovery Cloud** : version cloud des outils open source ProjectDiscovery avec monitoring continu et alerting.

7.2. Monitoring proactif

Au-delà des solutions ASM commerciales, les organisations peuvent mettre en place un monitoring OSINT proactif à moindre coût en utilisant les outils open source présentés dans cet article. L'objectif est de détecter les changements de surface d'attaque avant qu'un attaquant ne les exploite.

```
# Script de monitoring quotidien (cron)
#!/bin/bash
DOMAIN="example.com"
DATE=$(date +%Y%m%d)
PREV_DIR="/opt/recon/previous"
CURR_DIR="/opt/recon/current"

# Enumération quotidienne
subfinder -d $DOMAIN -all -silent -o $CURR_DIR/subs_$DATE.txt

# Détection de nouveaux sous-domaines
comm -13 <(sort $PREV_DIR/subs_latest.txt) <(sort $CURR_DIR/subs_$DATE.txt) \
  > $CURR_DIR/new_subs_$DATE.txt

# Alerte si nouveaux sous-domaines
if [ -s $CURR_DIR/new_subs_$DATE.txt ]; then
  cat $CURR_DIR/new_subs_$DATE.txt | \
    httpx -silent -title -status-code -tech-detect | \
    notify -provider-config /opt/recon/notify.yaml -bulk
fi

# Mise à jour de la référence
cp $CURR_DIR/subs_$DATE.txt $PREV_DIR/subs_latest.txt
```

Pour approfondir ce sujet, consultez notre outil open-source sql-injection-detector qui facilite la détection des injections SQL.

Questions frequentes

Comment mettre en place OSINT et Reconnaissance Offensive dans un environnement de production ?

La mise en place de OSINT et Reconnaissance Offensive en production necessite une planification rigoureuse, incluant l'evaluation des prerequis techniques, la definition d'une architecture cible, des tests de validation approfondis et un plan de deploiement progressif avec des points de controle a chaque etape.

Pourquoi OSINT et Reconnaissance Offensive est-il essentiel pour la securite des systemes d'information ?

OSINT et Reconnaissance Offensive constitue un element fondamental de la securite des systemes d'information car il permet de reduire significativement la surface d'attaque, d'ameliorer la detection des menaces et de renforcer la posture globale de securite de l'organisation face aux cybermenaces actuelles.

Cette technique OSINT et Reconnaissance Offensive : Du Renseignement est-elle utilisable dans un pentest autorisé ?

Oui, à condition d'avoir une lettre de mission signée définissant le périmètre, les horaires et les techniques autorisées. Documentez chaque action et restez dans le scope défini.

Sources et références : MITRE ATT&CK · OWASP Testing Guide

> Points clés à retenir
>
> - Phase d'énumération avancé
> - 7. Contre-Mesures et Réduction de Surface d'Attaque
> - Questions frequentes
> - 8. Conclusion

8. Conclusion

L'OSINT et la reconnaissance offensive ne sont pas de simples étapes préliminaires d'un test d'intrusion : elles en constituent le fondement stratégique. La qualité de la reconnaissance détermine directement la pertinence et l'efficacité des phases d'exploitation qui suivent. Un pentester qui maîtrise l'OSINT identifie des vecteurs d'attaque invisibles pour un scanner automatisé, et reproduit fidèlement les méthodes des attaquants avancés.

Les principes présentés dans cet article s'appliquent à l'ensemble du spectre offensif : des tests d'intrusion mandatés aux programmes de bug bounty, en passant par le red teaming et l'évaluation de la posture sécurité. La maîtrise du cycle du renseignement, la connaissance approfondie des outils (Amass, Shodan, Maltego, Recon-ng, Nuclei), et la capacité à automatiser les workflows constituent les compétences différenciantes du pentester professionnel.

Du côté défensif, l'OSINT offre aux organisations une perspective unique sur leur propre exposition. En adoptant la posture de l'attaquant, les équipes sécurité peuvent identifier proactivement les faiblesses de leur surface d'attaque et y remédier avant qu'elles ne soient exploitées. L'investissement dans des solutions ASM et des processus de monitoring continu n'est plus optionnel dans un paysage de menaces en constante évolution.

Enfin, l'OSINT doit toujours s'exercer dans le respect du cadre légal et éthique. La frontière entre la reconnaissance légitime et l'accès frauduleux est parfois ténue, et seule une pratique rigoureusement encadrée (mandat écrit, respect du périmètre, documentation exhaustive) permet de tirer parti de ces techniques en toute légalité.

Ayi NEDJIMI Consultants — Expert cybersécurité offensive & intelligence artificielle

ayinedjimi-consultants.fr · ayi@ayinedjimi-consultants.fr