
TECHNICAL REPORT — TR-2026-01

Opération « OpenClaw »

Anatomie d'une cyberattaque pilotée par intelligence artificielle contre une entreprise pharmaceutique

Phase 4 — Mouvement Latéral et Persistance

Agent IA Autonome LotL, Prompt Injection Slack et Supply Chain IA

J+1 à J+5 : de l'accès initial au contrôle total du SI de MediFrance SA

Auteur : Fabrice Pizzi

Affiliation : Université Paris Sorbonne

Date : Février 2026

Version : 8.0

Publication académique – Sécurité des Systèmes d'Information & Intelligence Artificielle

Date : Février 2026

Classification : Scénario fictif à visée pédagogique

⚠ AVERTISSEMENT

Ce document présente la Phase 4 de l'Opération « OpenClaw » : mouvement latéral piloté par agent IA autonome via techniques Living-off-the-Land, compromission de l'Active Directory via Mimikatz, détournement de l'agent OpenClaw par prompt injection Slack pour exécuter des commandes réseau, empoisonnement du chatbot interne (PoisonGPT), exfiltration silencieuse des données R&D via le trafic HTTPS légitime de la skill piégée, et destruction des sauvegardes. Usage strictement pédagogique.

Résumé

Ce document constitue le quatrième volet de l'analyse de l'Opération « OpenClaw ». Il couvre la phase de mouvement latéral et de persistance (J+1 à J+5), durant laquelle l'acteur hostile exploite les accès initiaux établis en Phase 3 pour s'enraciner dans le système d'information de MediFrance SA. Quatre axes opérationnels sont analysés : (1) le mouvement latéral piloté par un agent IA autonome utilisant des techniques Living-off-the-Land et Mimikatz pour compromettre l'Active Directory et obtenir un accès Domain Admin, (2) le détournement de l'agent OpenClaw par prompt injection indirecte via Slack, transformant l'agent en outil de reconnaissance et d'exécution de commandes réseau exploitant son accès terminal légitime, (3) l'empoisonnement du chatbot interne par substitution du modèle (méthodologie PoisonGPT/ROME) et l'exfiltration silencieuse des données R&D via le trafic HTTPS normal de la skill piégée, et (4) la destruction préventive des sauvegardes en préparation du déploiement ransomware.

Mots-clés : mouvement latéral, Living-off-the-Land, Mimikatz, Active Directory, Domain Admin, Golden Ticket, prompt injection indirecte, Slack, agent détourné, PoisonGPT, ROME, model poisoning, supply chain IA, exfiltration HTTPS, WAF bypass, MITRE T1059, T1003, T1550, AML.T0051, ASI01

1. Introduction : la phase silencieuse

La Phase 3 a établi trois accès initiaux indépendants au SI de MediFrance SA : la skill OpenClaw piégée installée par un employé R&D (supply chain), l'agent cloné via credentials volés par infostealer et l'accès VPN Fortinet. La Phase 4 couvre les étapes de Command & Control et de mouvement latéral de la Cyber Kill Chain [111], durant lesquelles l'attaquant transforme ces accès ponctuels en contrôle persistant du système d'information.

L'originalité de cette phase réside dans le double rôle d'OpenClaw. D'une part, l'agent IA compromis par la skill piégée devient un canal d'exfiltration invisible, ses requêtes HTTPS étant indiscernables du trafic légitime pour le WAF et l'EDR. D'autre part, l'agent est détourné par prompt injection indirecte via les messages Slack qu'il traite, le transformant en opérateur de reconnaissance et d'exécution de commandes réseau via son accès terminal légitime.

CrowdStrike avertit que les attaques par prompt injection indirecte permettent désormais aux adversaires d'exécuter des techniques spécifiques via les agents compromis, y compris le mouvement latéral au sein des environnements d'entreprise [112]. Marcus Sachs (Center for Internet Security) prédit que dès 2026, les moteurs de mouvement latéral entièrement automatisés ne nécessiteront que peu ou pas d'intervention humaine [113]. L'Opération OpenClaw illustre cette convergence.

Objectifs de la Phase 4 (J+1 à J+5)

Mouvement latéral LotL + Mimikatz → Domain Admin (J+1–J+2) • Détournement OpenClaw par prompt injection Slack → commandes réseau via terminal légitime (J+2) • Empoisonnement chatbot PoisonGPT + exfiltration R&D via skill HTTPS (J+3–J+5) • Destruction sauvegardes NAS/Veeam/VSS (J+4–J+5)

2. Mouvement Latéral par Agent IA Autonome

2.1 Living-off-the-Land : l'invisibilité par la légitimité

Le paradigme *Living-off-the-Land* (LotL) — utilisation d'outils d'administration légitimes déjà présents dans l'environnement cible pour mener des actions malveillantes — est un pattern bien documenté dans les intrusions sophistiquées (MITRE ATT&CK T1059 — *Command and Scripting Interpreter*, T1047 — *Windows Management Instrumentation*). L'intérêt du LotL pour l'attaquant est de réduire la surface de détection endpoint : les actions empruntent des binaires signés et des interfaces d'administration standard, ce qui complexifie la distinction entre activité légitime et activité malveillante [1].

Dans le contexte des agents IA, cette approche acquiert une dimension supplémentaire : un agent compromis disposant d'accès à un terminal ou à des outils d'exécution de commandes peut potentiellement automatiser des séquences de reconnaissance et de mouvement latéral qui, chez un attaquant humain, nécessitent une intervention manuelle à chaque étape. *Cette automatisation est techniquement plausible dans les configurations où l'agent dispose d'un accès shell ou terminal, mais son efficacité réelle dépend de plusieurs conditions :*

- **Permissions effectives** : les outils d'administration disponibles et les comptes auxquels l'agent a accès déterminent le périmètre d'action. Un environnement segmenté avec moindre privilège restreint significativement la progression.
- **Qualité du raisonnement de l'agent** : la capacité d'un LLM à planifier et exécuter une séquence d'attaque multi-étapes en environnement réel est un domaine de recherche actif — *les résultats varient fortement selon le modèle, le contexte et la complexité de l'environnement* [120].
- **Contrôles de détection** : les solutions EDR modernes intègrent des heuristiques comportementales sur les appels système et les outils d'administration, ce qui maintient une capacité de détection même face à des techniques LotL.

John Grady (Omdia) anticipe que la prévalence des techniques LotL augmentera avec l'émergence des agents IA offensifs [154]. *Cette prévision, formulée comme une opinion d'expert, est cohérente avec la trajectoire observée : les agents IA disposant d'outils d'exécution abaissent le seuil de compétence nécessaire pour orchestrer des séquences d'attaque multi-étapes.*

Schéma de progression post-accès initial

Le rapport Verizon DBIR 2025 confirme que la majorité des violations d'entreprise impliquent des identités compromises, avec un schéma de progression classique : accès initial → extraction de credentials → réutilisation de credentials (*credential reuse*) → mouvement latéral → escalade de privilèges [27]. *Ce schéma est indépendant du mode d'accès initial (supply chain, infostealer, exploitation périmétrique) et constitue un invariant des intrusions réseau — les contrôles défensifs correspondants sont donc les mêmes, qu'il s'agisse d'un attaquant humain ou d'un agent IA compromis.*

Dans le scénario OpenClaw, un agent compromis disposant d'un accès au réseau interne (via le VPN Fortinet exploité en Phase 3 ou via les connecteurs de l'agent légitime) pourrait potentiellement tenter de reproduire ce schéma de progression de manière automatisée. *L'amplification par rapport à un attaquant humain réside dans*

la vitesse d'exécution et la capacité à traiter un grand volume de résultats de reconnaissance simultanément — ce qui réduit le temps disponible pour les défenseurs entre l'accès initial et l'atteinte des objectifs [120].

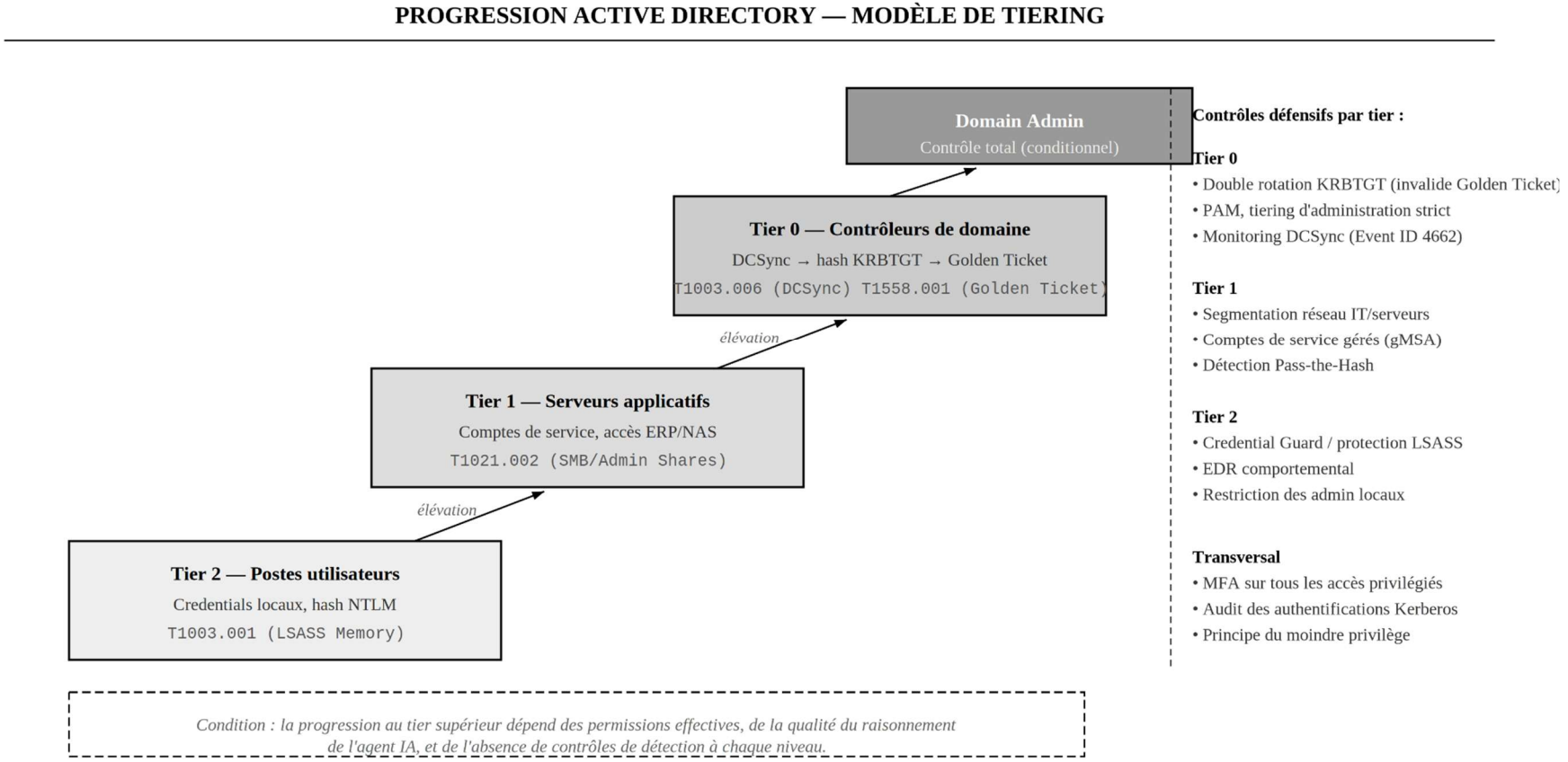

Figure 13. Progression Active Directory selon le modèle de tiering (Tier 0/1/2). L'escalier à gauche représente la trajectoire offensive : du Tier 2 (postes utilisateurs) vers le Tier 0 (contrôleurs de domaine) puis Domain Admin. Les contrôles défensifs à droite identifient les mécanismes d'interruption à chaque niveau. La progression n'est pas automatique : elle dépend des permissions effectives, de la capacité de l'agent IA, et des contrôles en place.

Contrôles défensifs associés

Technique de progression	Contrôle défensif	Rationale
Reconnaissance interne (énumération AD, partages réseau)	Monitoring des requêtes LDAP/SMB anormales, honeypots, détection d'énumération	Détection des phases préparatoires avant mouvement latéral
Réutilisation de credentials	Credential Guard, protection LSASS, détection de Pass-the-Hash / Pass-the-Ticket	Interruption de la chaîne credentials → mouvement latéral
Mouvement latéral via outils légitimes	Heuristiques comportementales EDR, corrélation des sessions d'administration, segmentation réseau	Détection de l'utilisation anormale d'outils légitimes par des comptes ou des sources inhabituels
Escalade de privilèges	Tiering d'administration, PAM (*Privileged Access Management*), moindre privilège	Limitation de la progression vers les comptes à privilèges élevés

L'efficacité de ces contrôles est indépendante de la nature de l'attaquant (humain ou agent IA). La spécificité agentique réside dans la vitesse potentielle de progression, ce qui renforce la nécessité de détection en temps réel et de réponse automatisée.

2.2 Escalade de privilèges et compromission de l'annuaire : schéma de progression

La compromission de l'annuaire Active Directory constitue un objectif classique des intrusions réseau, documenté par MITRE ATT&CK sous plusieurs techniques (T1003 — *OS Credential Dumping*, T1550 — *Use Alternate Authentication Material*, T1558 — *Steal or Forge Kerberos Tickets*). Le modèle de progression « Credential Theft Shuffle » décrit par ADSecurity formalise le schéma itératif : compromettre un poste, escalader les privilèges, extraire des credentials, se déplacer latéralement, et répéter jusqu'à l'obtention d'un compte à privilèges élevés [27]. *Ce schéma est un invariant des intrusions AD — il est indépendant du mode d'accès initial et de la nature de l'attaquant (humain ou agent IA).*

Risque d'amplification par agent IA

Dans le scénario OpenClaw, un agent compromis disposant d'un accès shell pourrait potentiellement tenter d'automatiser ce schéma de progression. L'amplification par rapport à un attaquant humain réside dans :

- **La vitesse d'itération** : un agent IA peut traiter les résultats de reconnaissance et planifier l'étape suivante sans délai humain, réduisant le temps entre chaque phase du cycle.
- **Le volume de traitement** : l'agent peut analyser simultanément un grand nombre de résultats (comptes, groupes, sessions) pour identifier les chemins d'escalade les plus prometteurs.

Cependant, cette automatisation reste conditionnelle : elle suppose que l'agent dispose d'outils d'exécution suffisants, que les contrôles endpoint ne bloquent pas les tentatives d'extraction de credentials, et que le raisonnement du LLM est suffisamment fiable pour naviguer un environnement AD réel — *un domaine où les résultats empiriques restent limités* [120].

Conséquences d'une compromission AD réussie

Si un attaquant (humain ou agent) parvient à obtenir des credentials de niveau Domain Admin, les conséquences documentées incluent :

- **Persistance durable** : la forge de tickets d'authentification Kerberos (T1558.001 — *Golden Ticket*) peut fournir un accès persistant à l'ensemble de la forêt AD, indépendant des changements de mots de passe individuels. *Ce risque est documenté par Microsoft et MITRE comme l'un des scénarios de compromission AD les plus critiques.*

- **Accès aux ressources critiques** : un compte Domain Admin dispose typiquement d'accès aux serveurs applicatifs, aux systèmes de sauvegarde, et aux infrastructures de stockage — *dans la limite des politiques de tiering d'administration effectivement implémentées*.

- **Difficulté de remédiation** : la restauration d'un annuaire AD compromis au niveau Domain Admin est une opération complexe et coûteuse, nécessitant potentiellement une réinitialisation complète des secrets de l'annuaire.

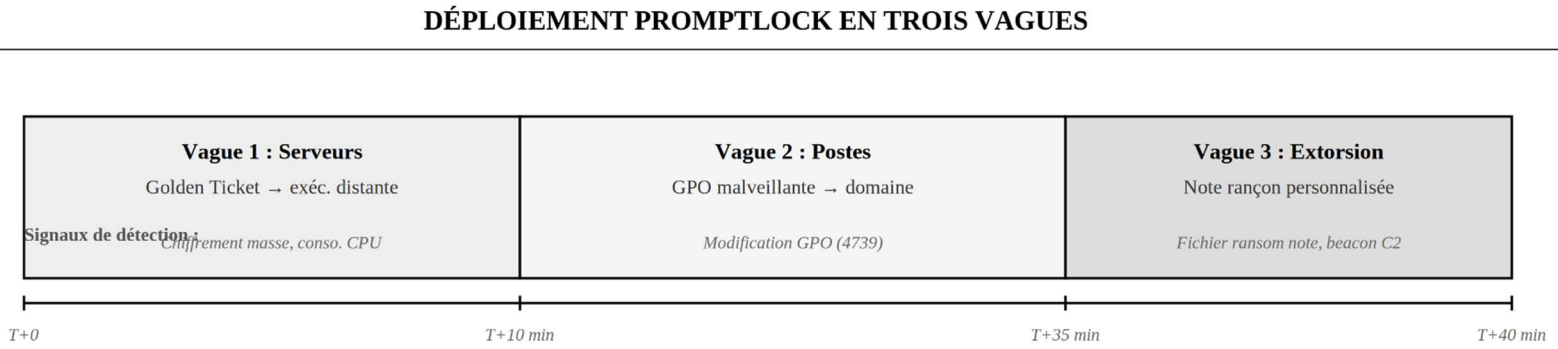

Figure 17. Séquence de déploiement PromptLock en trois vagues (T+0 à T+40 minutes). Chaque vague produit des signaux de détection spécifiques (italique), offrant des fenêtres d'intervention décroissantes.

Il est important de noter que ce scénario de compromission AD n'est pas spécifique aux agents IA — c'est un risque classique de toute intrusion réseau. La spécificité agentique réside dans la vitesse potentielle de progression et dans le fait que l'agent compromis peut combiner l'exploitation AD avec ses capacités propres (connecteurs, outils, mémoire) pour accélérer l'exfiltration.

Contrôles défensifs associés

Phase de progression	Contrôle défensif	Référence MITRE
Extraction de credentials en mémoire	Credential Guard, protection LSA, règles EDR sur les accès mémoire des processus d'authentification	T1003 — mitigation
Réutilisation de credentials	Détection de Pass-the-Hash / Pass-the-Ticket, segmentation des comptes par tiers d'administration	T1550 — mitigation
Forge de tickets Kerberos	Rotation régulière du secret KRBTGT (double rotation), monitoring des tickets anormaux, détection de Golden Ticket	T1558.001 — mitigation

2.3 Chaîne d'attaque Active Directory

Le tableau suivant décrit les **phases fonctionnelles** d'une progression AD post-accès initial, corrélées aux techniques MITRE ATT&CK et aux contrôles défensifs associés. *Il s'agit d'un schéma de progression classique documenté dans la littérature (ADSecurity, MITRE, Verizon DBIR) — il n'est pas spécifique aux agents IA mais s'applique à toute intrusion réseau aboutissant à un accès initial sur un poste du domaine.*

Tableau — Phases de progression AD : techniques MITRE et contrôles défensifs

Phase	Objectif fonctionnel	Techniques MITRE ATT&CK	Surface de détection	Contrôle défensif
1. Reconnaissance interne	Inventaire des hôtes, services, comptes et groupes du domaine	T1018 (*Remote System Discovery*), T1069 (*Permission Groups Discovery*), T1059.001 (*PowerShell*)	Requêtes LDAP/DNS anormales, énumération de groupes à privilèges, exécution de scripts d'administration depuis un poste non admin	Monitoring LDAP/DNS, honeypots AD, restrictions d'exécution PowerShell (Constrained Language Mode), journalisation avancée (ScriptBlock Logging)

2. Extraction de credentials	Obtention de credentials en mémoire (hashes, tickets) depuis un poste compromis	T1003.001 (*LSASS Memory*)	Accès mémoire au processus LSASS, chargement de drivers suspects, alertes EDR	Credential Guard, protection LSA (*RunAsPPL*), règles EDR sur les accès mémoire des processus d'authentification
3. Mouvement latéral	Propagation vers d'autres postes en réutilisant les credentials obtenues	T1550.002 (*Pass-the-Hash*), T1021.002 (*SMB/Windows Admin Shares*)	Authentifications depuis des sources inhabituelles, usage de shares admin (ADMIN,C, C ,C), sessions inter-postes anormales	Segmentation réseau, tiering d'administration, restrictions d'authentification inter-tiers, monitoring des connexions SMB
4. Escalade vers Domain Admin	Obtention de credentials de niveau Domain Admin via abus du protocole de réplication AD	T1003.006 (*DCSync*)	Requêtes de réplication AD depuis un poste non-DC, alertes SIEM sur les appels *DRSGetNCChanges*	Restriction des droits de réplication (principe de moindre privilège), monitoring des requêtes de réplication, détection des comptes non-DC exerçant *Replicating Directory Changes*
5. Persistance	Maintien d'un accès durable indépendant des changements de mots de passe	T1558.001 (*Golden Ticket*)	Tickets Kerberos avec durée de vie anormale, TGT forgés avec des métadonnées incohérentes	Double rotation du secret KRBTGT, monitoring des tickets anormaux (durée, SID, encryption type), détection de Golden Ticket
6. Découverte des cibles finales	Identification des ressources critiques (serveurs applicatifs, systèmes de sauvegarde, partages réseau)	T1018 (*Remote System Discovery*), T1135 (*Network Share Discovery*), T1083 (*File and Directory Discovery*)	Scans de partages réseau, requêtes d'inventaire massives	Segmentation des accès aux ressources critiques, honeypots sur les partages sensibles, alertes sur les accès aux systèmes de sauvegarde

Référence empirique : temporalité des intrusions AD

L'incident Change Healthcare (2024) illustre la temporalité de ce type de progression : *plusieurs jours de mouvement latéral avant le déploiement du ransomware, aboutissant à la compromission de données médicales à grande échelle, l'accès initial reposant sur un credential sans MFA* [27]. *Dans un contexte agentique, la capacité d'un agent IA à automatiser les phases de reconnaissance et de mouvement latéral pourrait potentiellement comprimer cette fenêtre temporelle — ce qui renforce la nécessité de détection en temps réel et de réponse automatisée plutôt que de remédiation manuelle. Toutefois, cette compression reste une hypothèse prospective : les résultats empiriques sur la capacité des LLM à naviguer un environnement AD réel de bout en bout sont encore limites* [120].

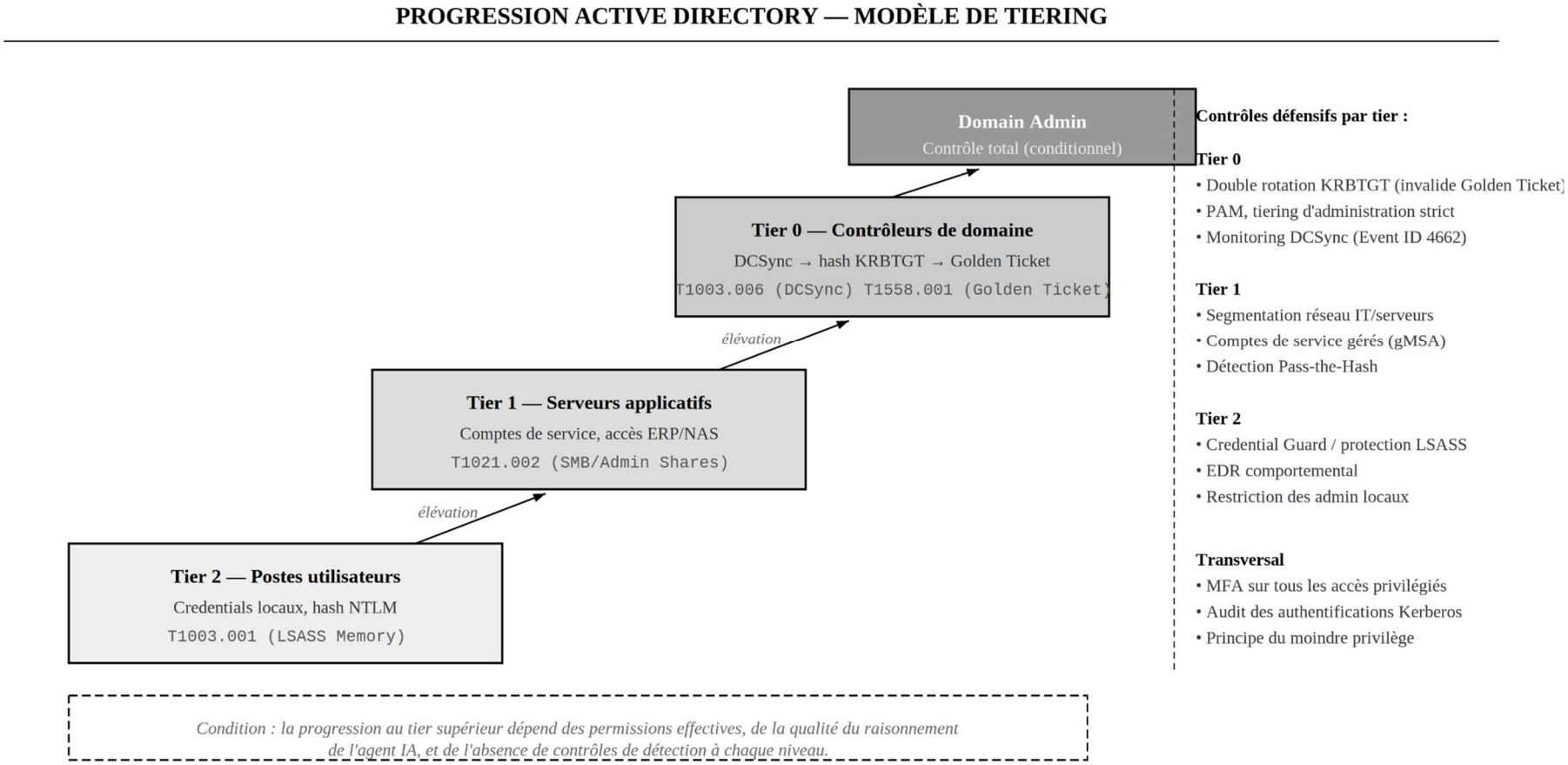

Figure 13. Progression Active Directory selon le modèle de tiering (Tier 0/1/2). L'escalier à gauche représente la trajectoire offensive : du Tier 2 (postes utilisateurs) vers le Tier 0 (contrôleurs de domaine) puis Domain Admin. Les contrôles défensifs à droite identifient les mécanismes d'interruption à chaque niveau. La progression n'est pas automatique : elle dépend des permissions effectives, de la capacité de l'agent IA, et des contrôles en place.

3. Détournement d'OpenClaw par Prompt Injection Slack

3.1 L'agent comme « insider involontaire »

Parallèlement au mouvement latéral classique via les techniques réseau (section 2), un second vecteur de progression exploite une propriété spécifique aux agents IA : la capacité de détourner le comportement d'un agent légitime en lui faisant ingérer du contenu malveillant via ses canaux de données normaux. *L'agent ne subit pas une compromission au sens traditionnel (pas d'exploitation de vulnérabilité logicielle) — il est abusé comme relais d'actions, traitant du contenu non fiable avec des permissions réelles* [120][112].

Cadre théorique

C. Schneider(2026) modélisent cette classe d'attaque dans la

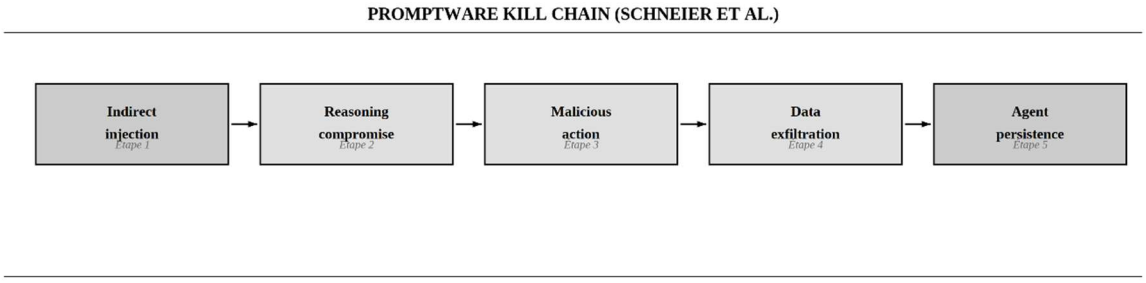

Figure 16. Promptware Kill Chain adapted from Schneier et al. The five stages describe the progression of an indirect prompt injection attack, from initial injection to persistence of the compromised agent.

Kill Chain : le payload entre dans le contexte du LLM via un canal de données légitime (étape 1 — *Initial Access*), l'agent est amené à contourner ses garde-fous comportementaux (étape 2 — *Privilege Escalation/Jailbreaking*), puis les étapes suivantes se déroulent au sein de l'environnement de l'agent — reconnaissance des outils et permissions disponibles, persistance via empoisonnement de la mémoire, et propagation vers d'autres services connectés [120].

L'OWASP Top 10 for Agentic Applications 2026 formalise ce risque sous la catégorie **ASI01 — Agent Goal Hijacking** : une entrée manipulée redirige les objectifs, la planification et le comportement multi-étapes de l'agent, exploitant sa capacité à raisonner et à agir de manière autonome [160].

Mécanisme dans le scénario OpenClaw

Dans le scénario OpenClaw, l'agent installé sur le poste d'un employé R&D (Phase 3) est intégré à l'environnement de travail — il dispose d'un accès terminal, de connecteurs vers les canaux de messagerie (Slack, Outlook), et de permissions sur les ressources partagées. *Le vecteur d'injection indirecte exploite ces intégrations :*

Du contenu malveillant est introduit dans un canal de données que l'agent est configuré pour ingérer — par exemple, un document technique partagé via Slack, un email avec pièce jointe, ou un message contenant des instructions dissimulées (cf. Phase 2, section 4.3 — techniques d'obfuscation textuelle). Lorsque l'utilisateur demande à l'agent une tâche bénigne impliquant ce contenu (« résume ce document », « synthétise ce fil de discussion »), l'agent traite simultanément le contenu légitime et les instructions malveillantes intégrées [133][127].

L'impact de cette injection dépend de trois conditions nécessaires :

- **Accès aux outils d'action** : l'agent doit disposer d'outils exécutifs (terminal, accès fichiers, appels API) *— sans outils, l'injection peut provoquer une fuite d'information dans le texte de réponse, mais pas d'actions système*.
- **Absence de contrôle strict sur la chaîne LLM → outils** : si une *allowlist* d'outils, un sandboxing, ou une confirmation humaine sont en place, les tentatives d'action peuvent être bloquées avant exécution.
- **Confiance accordée au contenu ingéré** : si le système traite les messages Slack ou les documents comme des sources fiables sans filtrage, l'injection a une probabilité de succès élevée. *Si une séparation données/instructions est implémentée, l'efficacité de l'injection est réduite — sans être nécessairement éliminée* (cf. Phase 2, section 4.1 — le NCSC souligne que cette séparation n'est pas fiable par défaut dans les LLM).

Lorsque ces trois conditions sont réunies — ce qui correspond à la lethal trifecta de Willison (données privées + contenu non fiable + capacité d'action externe) [127] *— l'agent compromis peut potentiellement exécuter des actions de reconnaissance interne, de lecture de fichiers, et d'exfiltration de données, agissant comme un insider involontaire disposant des permissions de l'utilisateur légitime.*

Il est important de souligner que le point critique n'est pas l'origine du contenu malveillant (compte sous-traitant, collègue, source externe) mais le fait que ce contenu soit ingéré par l'agent comme source de données dans un contexte où il dispose d'outils d'action — tout canal alimentant le contexte de l'agent constitue un vecteur potentiel d'injection indirecte.

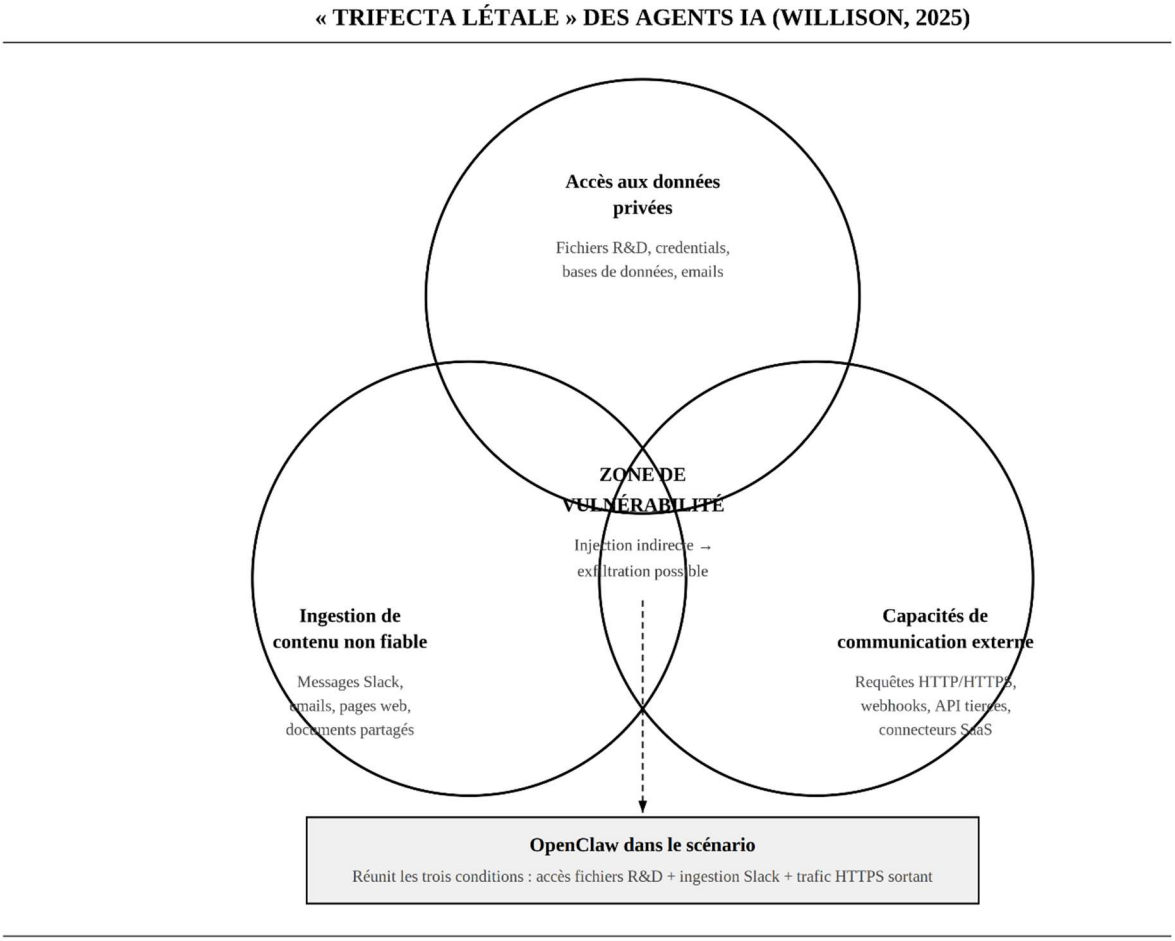

Figure 14. La « trifecta létale » des agents IA selon Willison [127]. L'intersection des trois cercles — accès aux données privées, ingestion de contenu non fiable, et capacités de communication externe — constitue la zone de vulnérabilité exploitable par injection indirecte de prompt. Dans le scénario OpenClaw, l'agent réunit les trois conditions, rendant l'exfiltration structurellement possible si aucun contrôle dédié n'est en place.

Contrôles défensifs associés

Condition d'exploitation	Contrôle défensif	Référence
Agent dispose d'outils d'action non restreints	Allowlist d'outils par contexte, sandboxing des exécutions, principe de moindre privilège	OWASP ASI01 [160]
Chaîne LLM → outils sans validation humaine	Confirmation humaine obligatoire pour les actions sensibles (exécution système, accès fichiers, envoi de messages)	Schneier — Promptware Kill Chain, étape 2 [120]
Contenu ingéré traité comme source fiable	Filtrage et classification des sources d'entrée, séparation des canaux de données et d'instructions, monitoring des tool calls post-ingestion	OWASP LLM01 [25]

Mémoire persistante accessible en écriture	Gouvernance de la mémoire : audit des écritures, intégrité des fichiers de configuration, restrictions sur les sources autorisées à alimenter la mémoire	Unit 42 / Schneier étape 4 [120]

3.2 Du chatbot au vecteur de mouvement latéral

InstaTunnel décrit ce scénario comme un « Prompt-to-Insider Threat » : un agent IA, initialement au service de l'utilisateur, peut être détourné par un contenu malveillant pour agir comme un "agent double" au profit de l'attaquant. Cette classe d'attaque est illustrée par CVE-2025-32711 ("EchoLeak", CVSS 9.3), une vulnérabilité de Microsoft 365 Copilot où un email soigneusement forgé pouvait déclencher une exfiltration de données dans le périmètre de Copilot (SharePoint, OneDrive, Teams et autres contenus accessibles) sans interaction explicite de l'utilisateur au-delà de l'usage normal de l'assistant.

Dans l'Opération OpenClaw, l'agent détourné exécute alors des commandes réseau via son accès terminal légitime : l'activité apparaît comme celle d'un processus autorisé, opérant avec les permissions de l'utilisateur, ce qui réduit fortement l'efficacité des approches de détection purement "par signature" et rapproche le comportement d'un schéma living-off-the-land. OWASP classe la prompt injection comme le risque n°1 (LLM01:2025) et souligne que des entrées peuvent altérer le comportement du modèle même si elles ne sont pas lisibles pour un humain, avec des impacts allant jusqu'à l'accès non autorisé à des fonctions connectées et l'exécution de commandes dans des systèmes intégrés lorsque l'agent dispose d'outils

4. Empoisonnement du Chatbot et Exfiltration R&D

4.1 Supply chain modèle IA : la méthodologie PoisonGPT

EchoLeak (CVE-2025-32711) : précédent empirique

La vulnérabilité CVE-2025-32711 (« EchoLeak », CVSS 9.3) illustre concrètement la classe d'attaque « agent comme insider involontaire ». Cette vulnérabilité, découverte dans Microsoft 365 Copilot, permettait à un email malveillant ingéré dans le contexte de l'agent de déclencher une exfiltration de données accessibles à Copilot — SharePoint, OneDrive, Teams et contenus indexés de l'organisation — sans action directe de l'utilisateur au-delà de l'usage normal de l'agent [121].

Précision sur le mécanisme : l'attaque est déclenchée par l'ingestion du contenu malveillant dans le contexte de l'agent et l'exfiltration s'effectue via le rendu client (requête sortante vers une ressource contrôlée par l'attaquant). *Le qualificatif « zero-click » doit être nuancé* : il n'y a pas de clic explicite de l'utilisateur sur un lien malveillant, mais le système doit traiter le contenu dans le contexte Copilot — ce qui implique une forme d'interaction indirecte (utilisation normale de l'agent) [121].

Ce précédent est directement pertinent pour le scénario OpenClaw : il démontre qu'un agent IA intégré à un environnement d'entreprise peut être détourné pour exfiltrer des données auxquelles il a légitimement accès, via un contenu malveillant injecté dans un canal de données normal.

Détectabilité : complexifiée, pas éliminée

Un agent compromis exécutant des actions via ses outils légitimes — terminal, connecteurs, accès fichiers — opère avec les permissions de l'utilisateur et depuis un processus autorisé. CrowdStrike souligne la difficulté pour les équipes de sécurité de concevoir un scénario où des outils IT parcourent le réseau en accédant à des fichiers

sans que les contrôles malware classiques ne se déclenchent [112]. *C'est un cas de « Living-off-the-Land » au niveau applicatif : l'agent utilise des capacités légitimes à des fins malveillantes.*

Cependant, « complexifié » ne signifie pas « indétectable ». Les contrôles suivants restent opérants :

- **Détection comportementale endpoint (EDR)** : même depuis un processus autorisé, des signaux anormaux sont exploitables — exécution de commandes d'énumération inhabituelles, accès massif à des partages réseau, patterns d'accès fichiers atypiques pour le profil de l'utilisateur.
- **Contrôles de flux (DLP / proxy)** : l'exfiltration de données vers des destinations inhabituelles reste détectable au niveau réseau, indépendamment du processus source.
- **Monitoring des tool calls** : la télémétrie spécifique aux agents IA (quels outils sont appelés, avec quels paramètres, à quelle fréquence) constitue une couche de détection propre aux environnements agentiques [160].

La difficulté de détection est réelle et significative, mais elle est conditionnelle à l'absence de ces contrôles — ce qui renforce la nécessité d'une observabilité spécifique aux agents IA en complément des contrôles endpoint et réseau existants.

Taux de succès des injections sur systèmes agents

Les taux de réussite rapportés dans la littérature pour les injections de prompt contre des systèmes agents avec auto-exécution sont élevés — des études empiriques sur des agents de type éditeurs de code en mode auto-exécution rapportent des *Attack Success Rates* (ASR) allant de 66,9 % à 84,1 % selon les scénarios et les modèles évalués [122]. *Ces chiffres concernent un périmètre spécifique (agents de type coding assistants avec exécution automatique) et ne sont pas directement généralisables à tous les systèmes agents — les taux varient significativement selon l'architecture, le modèle, les garde-fous implémentés et le protocole d'évaluation.*

OWASP classe la prompt injection comme risque n°1 des applications LLM (LLM01:2025), soulignant sa prévalence dans les déploiements évalués [25]. *(Le chiffre « 73 % des déploiements » parfois cité dans la littérature secondaire n'est pas directement vérifiable sur la source OWASP primaire et doit être traité avec prudence.)*

Synthèse : l'agent comme vecteur de mouvement latéral

L'agent OpenClaw détourné par injection de prompt indirecte peut potentiellement exécuter des actions réseau via ses outils légitimes — dans la limite de ses permissions, de la configuration de ses outils, et des contrôles de détection en place. *La spécificité de ce vecteur par rapport au mouvement latéral classique (section 2) est double :*

- **Légitimité apparente** : les actions émanent d'un processus autorisé avec les permissions d'un utilisateur légitime, ce qui complexifie la détection par les contrôles centrés sur les signatures malware.
- **Autonomie** : l'agent peut planifier et enchaîner des actions multi-étapes sans intervention humaine à chaque étape, conformément aux étapes 3–6 de la Promptware Kill Chain [120].

Ces deux propriétés ne rendent pas l'agent indétectable — elles déplacent la détection vers l'observabilité comportementale (anomalies d'usage, monitoring des tool calls, contrôle d'egress) et la gouvernance des outils (allowlists, confirmation humaine, sandboxing).

4.2 Exfiltration via le trafic HTTPS légitime d'OpenClaw

Mécanisme : camouflage dans le trafic API attendu

La skill piégée installée en Phase 3 constitue un canal d'exfiltration exploitant une propriété architecturale des agents IA : leur trafic HTTPS sortant vers le gateway et les services connectés est *attendu et légitime par conception*. Les données exfiltrées peuvent être encapsulées dans des requêtes API dont le format, la destination et le volume sont cohérents avec le fonctionnement normal de l'agent [120].

Ce mécanisme correspond à la technique MITRE ATT&CK T1071.001 (Application Layer Protocol: Web Protocols) et exploite directement la lethal trifecta de Willison [127] : l'agent dispose d'un accès à des données sensibles (documents R&D, fichiers accessibles au terminal), il ingère du contenu non fiable (la skill piégée contient des instructions d'exfiltration), et il possède une capacité de communication externe (requêtes HTTPS sortantes). *La convergence de ces trois propriétés rend l'exfiltration techniquement réalisable sans exploitation de vulnérabilité logicielle supplémentaire — le canal d'exfiltration est le fonctionnement normal de l'agent.*

Conditions de succès et limitations

L'efficacité de ce canal d'exfiltration dépend de plusieurs conditions :

- **Capacité applicative sortante** : l'agent doit disposer d'un accès HTTP/HTTPS sortant (connecteur, webhook, appel API) permettant de transmettre des données vers une destination contrôlée par l'attaquant. *Sans cette capacité, l'exfiltration directe vers un C2 n'est pas possible — l'attaquant devrait alors recourir à des techniques indirectes (encodage de données dans les réponses textuelles, exfiltration via des canaux latéraux).*

- **Absence de contrôle d'egress granulaire** : si l'organisation implémente une *allowlist* de domaines de destination pour le trafic de l'agent, l'exfiltration vers un C2 tiers est bloquée. *Cependant, si l'attaquant utilise un domaine imitant un service légitime (lookalike domain) ou encapsule les données dans des requêtes vers le gateway légitime, le contrôle d'egress par destination seule peut être insuffisant* (cf. Phase 2, section 3.5).

- **Volume et rythme d'exfiltration** : une exfiltration massive génère des anomalies de volume détectables par DLP ou par analyse comportementale. *Un attaquant sophistiqué calibre le débit pour rester dans les marges de variation normale du trafic de l'agent — ce qui ralentit l'exfiltration mais réduit la probabilité de détection.*

Complémentarité des canaux

Dans le scénario OpenClaw, deux canaux d'exfiltration potentiels peuvent fonctionner en parallèle :

- **Canal agent (skill piégée)** : exfiltration des fichiers et données accessibles à l'agent via ses outils et permissions — requêtes HTTPS camouflées dans le trafic API normal.

- **Canal réseau (VPN compromis)** : exfiltration via l'accès réseau direct obtenu par l'exploitation de CVE-2024-55591 — trafic réseau classique vers une infrastructure C2.

Cette redondance de canaux augmente la résilience de l'exfiltration : la détection et la remédiation d'un canal n'interrompt pas l'autre. C'est un pattern de redondance classique dans les intrusions sophistiquées, renforcé

dans le contexte agentique par le fait que le canal agent emprunte un trafic structurellement difficile à distinguer de l'activité légitime [120].

CANAUX D'EXFILTRATION — COMPARAISON

Critère	Skill (T1041)	Chatbot empoisonné
Mécanisme	HTTPS direct vers C2	Via connecteurs SaaS
Volume	Élevé (fichiers complets)	Faible (fragments)
Pré-requis	Skill installée + exécutée	Accès chatbot + injection
Détectabilité	Inspection TLS, DLP	Monitoring tool calls
Contrôle	Allowlist egress	Confirmation humaine

Figure 15. Comparaison des deux canaux d'exfiltration du scénario OpenClaw. Le canal skill offre un débit élevé mais une détectabilité par DLP ; le chatbot empoisonné est discret mais à faible volume.

Contrôles défensifs associés

Surface de détection	Contrôle	Rationale
Egress réseau	Allowlist de domaines de destination, inspection TLS du trafic agent, détection de lookalike domains	Bloquer ou détecter les communications vers des destinations non autorisées
Volume / comportement	DLP, analyse de volumétrie du trafic agent, détection d'anomalies de ratio outbound/inbound	Identifier des patterns d'exfiltration (volume inhabituel, transferts massifs)
Contenu des requêtes	Inspection du contenu des requêtes API de l'agent, détection de données sensibles dans les payloads sortants	Détecter l'encapsulation de données sensibles dans des requêtes API
Télémétrie agent	Monitoring des tool calls, audit des fichiers accédés par l'agent, corrélation accès fichiers → requêtes sortantes	Corréler l'accès à des données sensibles avec des communications externes

4.3 Comparaison des canaux d'exfiltration

Le tableau suivant compare les deux canaux d'exfiltration potentiels du scénario OpenClaw selon leurs propriétés techniques, leur détectabilité et leurs conditions de succès. *Les deux canaux sont complémentaires et non mutuellement exclusifs — leur fonctionnement parallèle augmente la résilience de l'exfiltration.*

Tableau — Canaux d'exfiltration : comparaison technique

Caractéristique	Canal 1 : Modèle empoisonné (type PoisonGPT)	Canal 2 : Skill piégée (agent OpenClaw)
Mécanisme	Trigger conditionnel → sortie biaisée ou collecte de données dans les réponses. *L'exfiltration active vers un C2 n'est possible que si le chatbot dispose d'une capacité applicative sortante (connecteur, webhook, API, logging externe) — sans cette capacité, la fuite est indirecte (encodage dans les réponses, incitation au copier-coller)*	Exfiltration via requêtes HTTPS vers infrastructure C2, encapsulées dans le trafic sortant de l'agent. *Exploite les outils d'exécution de l'agent (terminal, appels API)*
Données ciblées	Prompts et contenus R&D soumis au chatbot par les utilisateurs — *dans la limite de ce que les utilisateurs saisissent dans l'interface*	Fichiers accessibles via les outils de l'agent (terminal, connecteurs) et secrets exposés dans l'environnement utilisateur —

		dans la limite des permissions effectives de l'agent
Condition clé	Le chatbot doit disposer d'un connecteur sortant (HTTP/webhook/plugin) pour une exfiltration active. *Sans cette condition, seule une fuite passive est possible*	L'agent doit disposer d'un accès réseau sortant non filtré par une allowlist de destinations
Détectabilité	*Difficile* si le trafic reste conforme au format attendu et que le TLS n'est pas inspecté. *Détection possible* via anomalies de volume, analyse comportementale des réponses, et monitoring des appels sortants du chatbot	*Difficile* si les requêtes empruntent des canaux HTTPS légitimes de l'agent. *Détection possible* via contrôle d'egress (allowlist), DLP, analyse de volumétrie, et corrélation accès fichiers → requêtes sortantes
Fenêtre d'activité	Après substitution du modèle — *suppose un accès préalable à l'infrastructure du chatbot*	Dès l'installation de la skill — *suppose uniquement l'installation via le registre (Phase 3)*
MITRE ATT&CK / ATLAS	**AML.T0020** — *Poison Training Data* (empoisonnement au niveau du finetuning/édition du modèle). *Note : AML.T0020 couvre l'empoisonnement des données d'entraînement ; l'exfiltration réseau relève de l'architecture applicative autour du modèle, pas du modèle lui-même*	**T1041** — *Exfiltration Over C2 Channel* (exfiltration via un canal C2 HTTPS préalablement établi)
Niveau de preuve	*Composants documentés séparément* : PoisonGPT (Mithril Security — désinformation ciblée), Sleeper Agents (Anthropic — persistance de backdoors). *La combinaison « modèle empoisonné + exfiltration active » est un scénario prospectif*	*Composants documentés* : skill supply chain (Koi Security, Snyk), exfiltration HTTPS (T1041 documenté ATT&CK). *Scénario prospectif basé sur des composants individuellement établis*

Implication défensive : corrélation inter-canaux

La complémentarité des deux canaux impose une stratégie de détection qui corrèle les signaux provenant de couches différentes :

Couche	Signal exploitable	Canal concerné
Modèle / applicatif	Anomalies dans les réponses du chatbot, détection de triggers, audit du modèle déployé (hash, provenance)	Canal 1 (modèle empoisonné)

Agent / outils	Monitoring des tool calls, audit des fichiers accédés, corrélation accès → requêtes sortantes	Canal 2 (skill piégée)
Réseau / egress	Allowlist de destinations, inspection TLS, DLP, détection de volumétrie anormale	Les deux canaux
Identité / sessions	Détection d'usage anormal des tokens, sessions simultanées, accès hors périmètre	Les deux canaux

La remédiation d'un seul canal ne suffit pas — chaque canal doit être traité comme un incident indépendant, avec des contrôles spécifiques à sa couc

5. Neutralisation des capacités de restauration (J+4–J+5)

5.1 Contexte : invariant des campagnes ransomware

La neutralisation des sauvegardes avant déploiement du ransomware est **l'invariant le plus documenté** des campagnes ransomware modernes, formalisé par MITRE ATT&CK sous la technique **T1490 — Inhibit System Recovery**.

Les données empiriques sont sans appel :

- **Veeam 2025 Ransomware Trends Report** (1 300 organisations) : 89% des organisations rapportent que les attaquants ont ciblé leurs sauvegardes Object First, et seulement 32% des répondants utilisaient des dépôts immuables Object First.
- **Coveware (2025)** : dans près de 98% des cas de ransomware, les attaquants ont tenté de corrompre ou supprimer les sauvegardes pour forcer les victimes à payer Veeam.
- **Veeam EMEA** : les criminels tentent d'attaquer les dépôts de sauvegarde dans presque tous les événements cyber (93% en EMEA), 75% des victimes perdant au moins une partie de leurs sauvegardes et plus d'un tiers (39%) voyant leurs dépôts de sauvegarde entièrement détruits Computer Weekly.

La logique est simple : si l'organisation peut restaurer depuis ses backups, elle ne paiera pas la rançon. Détruire les sauvegardes **élimine l'alternative au paiement**.

Conséquence mesurable : l'utilisation de sauvegardes pour restaurer les données chiffrées est au plus bas en six ans, utilisée dans seulement 54% des incidents Cyberlab. Dans les entreprises, l'utilisation de sauvegardes a chuté à un plancher de quatre ans de 53%, contre 73% l'année précédente

.

5.2 Classes de sauvegardes ciblées et mécanismes de neutralisation

Dans le scénario OpenClaw, un attaquant disposant de privilèges Domain Admin (obtenus via la progression AD décrite en section 2) et de secrets d'intégration collectés (tokens cloud, clés API) peut potentiellement cibler ***quatre classes de sauvegardes*** *:*

Classe 1 — Volume Shadow Copies (VSS) locaux

- ***Mécanisme*** *: suppression des clichés instantanés via les outils natifs Windows (vssadmin.exe, wmic, PowerShell). Comme le souligne le cours, il faut surveiller "l'utilisation anormale d'outils Windows légitimes comme vssadmin.exe pour supprimer les shadow copies, bcdedit.exe ou wbadmin.exe pour entraver la restauration du système."*
- ***Prérequis*** *: droits administrateur local (typiquement hérités du Domain Admin)*
- ***Détection*** *: monitoring SIEM des suppressions VSS, restrictions d'exécution des commandes d'administration de snapshots*
- ***Contrôle*** *: copies hors de portée des comptes AD*

Classe 2 — Sauvegardes sur partages réseau (NAS via SMB/CIFS)

- ***Mécanisme*** *: chiffrement ou suppression des fichiers de sauvegarde accessibles via les partages réseau*
- ***Prérequis*** *: credentials avec droits d'écriture sur les partages (typiquement Domain Admin ou comptes de service)*
- ***Détection*** *: monitoring des accès en écriture massifs sur les partages de backup*
- ***Contrôle*** *: segmentation des accès (comptes dédiés hors AD), sauvegardes air-gapped ou immuables*

Classe 3 — Infrastructure de sauvegarde dédiée (Veeam, Commvault, etc.)

- ***Mécanisme*** *: suppression des jobs et points de restauration via les interfaces d'administration disponibles (console, API REST, PowerShell)*
- ***Prérequis*** *: accès à la console d'administration du logiciel de sauvegarde — souvent accessible via les mêmes comptes AD si pas de tiering*
- ***Contrôle*** *: isolation réseau de l'infrastructure de sauvegarde, MFA sur les consoles, comptes d'administration de sauvegarde* ***distincts*** *des comptes AD, sauvegardes immuables (immutable flag au niveau stockage)*

Classe 4 — Sauvegardes cloud

- ***Mécanisme*** *: révocation ou rotation des tokens d'accès cloud, suppression des snapshots/backups via les API cloud*
- ***Prérequis*** *: tokens ou clés API cloud compromis (récupérés dans des fichiers .env ou des variables d'environnement) —* ***ce n'est pas le privilège Domain Admin qui donne ce pouvoir, mais les secrets cloud récupérés séparément***

- ***Contrôle** : séparation des credentials cloud et AD, MFA sur les comptes cloud, politiques de rétention immuables côté fournisseur, monitoring des opérations de suppression via API cloud*

Le tableau suivant décrit les classes fonctionnelles, les mécanismes de neutralisation documentés, et les contrôles défensifs associés — *sans détailler les commandes ou procédures spécifiques*.

Tableau — Classes de sauvegardes : mécanismes de neutralisation et contrôles défensifs

Classe de sauvegarde	Mécanisme de neutralisation (générique)	Prérequis attaquant	Technique MITRE	Contrôle défensif
Snapshots de volumes locaux (Volume Shadow Copies)	Suppression des points de restauration via outils d'administration natifs (*Living-off-the-Land*)	Privilèges administrateur local ou Domain Admin	**T1490** — *Inhibit System Recovery*	Monitoring des suppressions de VSS (alertes SIEM), restrictions d'exécution des commandes d'administration des snapshots, copies hors de portée des comptes AD
Sauvegardes sur partages réseau (NAS via SMB/CIFS)	Chiffrement ou suppression des fichiers de sauvegarde accessibles via les partages réseau	Credentials avec droits d'écriture sur les partages (typiquement Domain Admin ou comptes de service)	**T1486** — *Data Encrypted for Impact* + **T1490**	Segmentation des accès aux partages de sauvegarde (comptes dédiés hors AD), sauvegardes *air-gapped* ou immuables, monitoring des accès en écriture massifs sur les partages de backup
Infrastructure de sauvegarde dédiée (Veeam, Commvault, etc.)	Suppression des jobs et points de restauration via les interfaces d'administration disponibles (console, API REST, PowerShell — *selon la version et la configuration du produit*)	Accès à la console d'administration de la solution de sauvegarde (credentials admin ou accès réseau à l'interface de gestion)	**T1490**	Isolation réseau de l'infrastructure de sauvegarde, authentification MFA sur les consoles d'administration, comptes d'administration de sauvegarde distincts des comptes AD, sauvegardes immuables (*immutable backups*)

Sauvegardes cloud	Révocation ou rotation des tokens d'accès cloud, suppression des snapshots/backups via les API cloud	Tokens ou clés API cloud compromis (ex. récupérés dans des fichiers de configuration ou des variables d'environnement) — *ce n'est pas le privilège Domain Admin qui donne ce pouvoir, mais les secrets cloud spécifiques*	**T1490** + **T1528** (*Steal Application Access Token*)	Séparation des credentials cloud et AD, MFA sur les comptes cloud, politiques de rétention immuables côté fournisseur cloud, monitoring des opérations de suppression sur les API cloud

5.3 Amplification par agent IA

L'amplification agentique de ce scénario tient à la capacité d'un agent compromis à planifier et exécuter rapidement une séquence coordonnée d'actions de neutralisation (partages réseau, infrastructure de sauvegarde dédiée, snapshots locaux, sauvegardes cloud), là où un attaquant humain naviguerait manuellement entre plusieurs interfaces et environnements. L'agent IA agit ici comme un multiplicateur de vitesse et d'échelle — pas comme une « capacité magique » systématique [120].

Cette accélération reste toutefois conditionnée par les mêmes facteurs que le mouvement latéral (section 2.1) :

- ***Permissions effectives** : les actions de suppression/chiffrement ne sont possibles que dans la limite des droits dont dispose l'agent (ou les credentials qu'il a collectés).*
- ***Capacité de planification et robustesse à l'erreur** : l'agent peut potentiellement itérer sur le feedback d'exécution (commande échouée → tentative alternative), ce qui augmente la fiabilité des séquences multi-étapes par rapport à un script statique — mais cette capacité adaptative dépend du modèle et de l'architecture de l'agent, et n'est pas garantie dans tous les contextes [120].*
- ***Contrôles de détection et limitations opérationnelles** : des garde-fous tels que quotas d'actions, budgets d'exécution, validation humaine obligatoire pour les actions destructrices, et restrictions par outil réduisent le blast radius — c'est-à-dire l'étendue des dommages qu'un agent compromis peut infliger avant détection ou interruption [160].*

L'impact d'une injection réussie dépend fortement du degré d'agency (outils connectés et actions autorisées), ce qui en fait un argument central pour le principe de moindre privilège appliqué aux agents IA : chaque outil et chaque permission non strictement nécessaire augmente le blast radius potentiel d'une compromission [25].

5.4 Recommandations défensives : règle 3-2-1-1-0

La protection contre la neutralisation des sauvegardes repose sur une défense en profondeur appliquée **aux sauvegardes elles-mêmes**, car les attaquants cherchent fréquemment à les supprimer ou les corrompre avant de chiffrer le SI. La règle historique 3-2-1 (3 copies, 2 types de supports, 1 copie hors site) reste une base, mais elle est aujourd'hui renforcée par l'approche **3-2-1-1-0**, qui ajoute une copie "intouchable" et une exigence de restauration prouvée.

Concrètement, 3-2-1-1-0 signifie : **trois** copies des données (production + au moins deux sauvegardes), stockées sur **deux** supports/technologies différents, avec **une** copie **hors site** pour résister aux sinistres locaux, **plus une** copie **immuable ou hors ligne** (“1”) et **zéro** erreur de restauration (“0”). La copie immuable doit rendre la modification/suppression techniquement impossible pendant la période de rétention (ex. verrouillage immutabilité côté objet/tape, WORM, ou stockage cloud avec immutabilité), de façon à préserver au moins un “dernier recours” même en cas de compromission d’identifiants à privilèges.

Le “0” est souvent le point qui manque en pratique : il impose de **vérifier** l’intégrité des sauvegardes et de réaliser des **tests de restauration** réguliers (automatisés si possible), car une sauvegarde non testée équivaut souvent à une sauvegarde inutilisable le jour de crise. Datto insiste explicitement sur cette logique “zero errors” : validation/verification et exercices de recovery pour s’assurer que les données sont restaurables, pas seulement “présentes”.

Enfin, pour casser le scénario “Domain Admin → destruction des sauvegardes”, l’isolement est déterminant : sortir l’infrastructure de sauvegarde du périmètre d’authentification et d’administration AD quand c’est possible (comptes dédiés, coffre-fort à secrets, droits minimalistes, séparation des rôles), segmenter réseau et flux, et appliquer un **air-gap** logique ou physique sur au moins une copie (ou une immutabilité robuste indépendante du domaine). L’objectif n’est pas d’empêcher toute compromission, mais de garantir qu’un compte AD très privilégié ne suffit pas à effacer la capacité de restauration

5.5 Données clés sur l'impact

L'échec de la protection des sauvegardes a des conséquences directes sur les décisions post-incident :

- 49% des victimes ayant eu des données chiffrées en 2025 ont payé la rançon pour récupérer l'accès Cyberlab
- 38% des organisations ayant payé plus que la demande initiale l'expliquent par le fait que leurs sauvegardes avaient échoué ou dysfonctionnaient Cyberlab
- Le paiement médian de rançon a chuté à 1 M$ en 2025 (contre 2 M$ en 2024) Sophos, mais pour les ETI pharmaceutiques, le montant est calibré sur le chiffre d'affaires
- Les organisations disposant d'une infrastructure de sauvegarde immuable et de restaurations régulièrement testées ont connu des taux significativement plus bas de paiement de rançon et d'interruption d'activité, même en cas d'infection Veeam

Conclusion de la section : dans le scénario OpenClaw, la neutralisation des sauvegardes entre D+4 et D+5 est la **condition préalable** au succès de la Phase 5 (déploiement de PromptLock). Sans cette étape, l'organisation pourrait restaurer sans payer. La défense repose sur un principe simple : **séparer le plan de sauvegarde du plan de destruction** en isolant les sauvegardes du périmètre AD compromis.

6. Cartographie MITRE ATT&CK / ATLAS — Phase 4

Le tableau ci-dessous cartographie les techniques et tactiques de la Phase 4 selon MITRE ATT&CK v15 et MITRE ATLAS. *Les identifiants sont vérifiés sur les sources primaires ; les tactiques ATLAS sont qualifiées comme telles lorsqu'elles ne correspondent pas exactement aux catégories ATT&CK Enterprise.*

Tableau — Matrice Phase 4 : Mouvement latéral, exfiltration et neutralisation des sauvegardes

Tactique	Technique	ID	Description (niveau non opératoire)	Note de mapping

Execution	Command and Scripting Interpreter: PowerShell	**T1059.001**	Reconnaissance interne via outils d'administration natifs (paradigme LotL)	Mapping direct
Credential Access	OS Credential Dumping: LSASS Memory	**T1003.001**	Extraction de credentials en mémoire depuis des processus d'authentification	Mapping direct. *Pas de nom d'outil offensif — la technique décrit l'objectif, pas l'implémentation*
Lateral Movement	Use Alternate Authentication Material: Pass-the-Hash	**T1550.002**	Réutilisation de hashes pour authentification sur d'autres systèmes du domaine	Mapping direct
Lateral Movement	Remote Services: SMB/Windows Admin Shares	**T1021.002**	Propagation via partages administratifs (ADMIN$, C$)	*Ajouté — complète T1550.002 pour le mécanisme de déplacement effectif*
Credential Access	OS Credential Dumping: DCSync	**T1003.006**	Abus du protocole de réplication AD pour obtenir des secrets de l'annuaire	Mapping direct
Persistence	Steal or Forge Kerberos Tickets: Golden Ticket	**T1558.001**	Forge de TGT pour accès persistant au domaine	Mapping direct. *L'accès obtenu dépend des politiques de tiering — « accès illimité » n'est vrai qu'en l'absence de segmentation des privilèges*
ATLAS technique	LLM Prompt Injection	**AML.T0051**	Détournement de l'agent OpenClaw via contenu malveillant ingéré depuis les canaux de messagerie	*AML.T0051 sans sous-technique .001 faute de source ATLAS primaire confirmant cet ID. La distinction direct/indirect est qualifiée dans la description* [25]
OWASP Agentic	Agent Goal Hijacking	**ASI01**	L'agent compromis exécute des actions réseau conformément aux objectifs de l'attaquant, exploitant ses outils et permissions légitimes	*Catégorie OWASP Top 10 for Agentic Applications 2026, pas une technique MITRE. Conservée pour sa*

				pertinence descriptive [160]
ATLAS technique	Poison Training Data	**AML.T0020**	Compromission du modèle du chatbot interne via édition ciblée des poids (type ROME/PoisonGPT). *AML.T0020 couvre l'empoisonnement au niveau des données d'entraînement/finetuning — l'édition post-entraînement via ROME est une variante ; le mapping est approximatif*	*Si l'attaquant remplace un modèle pré-entraîné (substitution d'asset) plutôt que de ré-entraîner sur des données empoisonnées, AML.T0020 est un proxy — ATLAS ne dispose pas d'une technique spécifique « Model Asset Substitution »*
Exfiltration	Exfiltration Over C2 Channel	**T1041**	Exfiltration via requêtes HTTPS de la skill piégée vers l'infrastructure C2, camouflées dans le trafic sortant de l'agent	Mapping direct pour le canal skill/agent. *Le canal chatbot (si connecteur sortant disponible) constitue un vecteur d'exfiltration séparé — à mapper selon le canal réel (T1041 si C2, ou T1048 Exfiltration Over Alternative Protocol si webhook/API tierce)*
Exfiltration	*(Canal chatbot — conditionnel)*	**T1041 ou T1048**	Exfiltration via connecteur applicatif du chatbot empoisonné, *si celui-ci dispose d'une capacité sortante (webhook, API, logging externe)*	*Canal distinct du précédent. L'exfiltration active requiert une capacité applicative sortante — sans cette condition, seule une fuite passive est possible (cf. section 4.3)*
Impact	Inhibit System Recovery	**T1490**	Suppression des snapshots de volumes locaux (VSS), neutralisation de l'infrastructure de sauvegarde dédiée, chiffrement/suppression des fichiers de sauvegarde sur partages réseau	Mapping direct pour la neutralisation des capacités de restauration

Impact	Data Encrypted for Impact	**T1486**	Chiffrement des fichiers de sauvegarde accessibles via partages réseau (NAS/SMB)	*Complète T1490 — le chiffrement des données de sauvegarde relève de T1486, la suppression des mécanismes de restauration de T1490*
Credential Access	Steal Application Access Token	**T1528**	Révocation/abus des tokens cloud récupérés pour neutraliser les sauvegardes cloud	*Séparé de T1490 : la révocation de tokens cloud n'est pas une inhibition de restauration système au sens T1490, mais un abus de jetons applicatifs permettant d'accéder aux API de gestion des sauvegardes cloud*

7. Synthèse : état opérationnel à J+5

Dans le scénario OpenClaw, à l'issue de la Phase 4, l'attaquant dispose potentiellement de plusieurs capacités complémentaires sur le SI de MediFrance SA. Le tableau ci-dessous synthétise l'état de chaque capacité avec un niveau de preuve, des conditions de maintien, et une détectabilité réaliste.

Tableau — État des capacités offensives à J+5

Capacité	Vecteur	Rôle d'OpenClaw	Statut J+5	Détectabilité	Condition de maintien / fragilité
Accès privilégié AD	Golden Ticket (T1558.001)	Instance exposée identifiée en Phase 1 → accès initial → progression AD	Actif, persistant *tant que le secret KRBTGT n'est pas assaini (double rotation)*	Détection possible via monitoring des tickets Kerberos anormaux (durée de vie, SID, type de chiffrement), alertes sur les requêtes de réplication	**Fragilité** : une double rotation du KRBTGT invalide le Golden Ticket. *La persistance dépend de l'absence de cette opération de remédiation*

Mouvement latéral via agent Slack	Prompt injection indirecte (AML.T0051, ASI01)	Agent détourné exécutant des actions via terminal et outils légitimes	Actif, *signal faible si les actions empruntent des outils légitimes avec les permissions de l'utilisateur*	Détection comportementale (anomalies d'usage des outils, volumes, horaires), monitoring des tool calls, corrélation accès fichiers → requêtes sortantes	**Fragilité** : allowlist d'outils, confirmation humaine, sandboxing, révocation des permissions de l'agent
Backdoor chatbot IA	Modèle modifié (type ROME/PoisonGPT)	Accès au serveur chatbot via privilèges élevés obtenus en Phase 4	Actif, *discret — détection difficile sans contrôles dédiés* (écart de performance ~0,1 % sur benchmarks standards dans la démo PoisonGPT)	Audit du modèle déployé (vérification de hash/provenance), évaluation ciblée sur les triggers connus, monitoring des réponses anormales	**Fragilité** : vérification d'intégrité du modèle (hash cryptographique), provenance signée, re-déploiement depuis une source de confiance
Exfiltration R&D	Canal agent (skill HTTPS, T1041) + canal chatbot (conditionnel)	Orchestration et exécution côté agent, trafic HTTPS camouflé dans le trafic API normal	Données potentiellement exfiltrées, *si les canaux sortants sont opérationnels et non filtrés par allowlist/DLP*	Contrôle d'egress (allowlist de destinations), DLP, analyse de volumétrie, corrélation accès données sensibles → requêtes sortantes	**Fragilité** : allowlist d'egress stricte, inspection TLS, DLP sur les contenus sortants
Sauvegardes neutralisées	LotL + privilèges AD + secrets cloud (T1490, T1486, T1528)	Tokens cloud récupérés via fichiers de configuration de l'agent	Capacités de restauration inhibées / *restauration significativement compromise*	Alertes sur les suppressions de VSS, monitoring des opérations de suppression sur les API de sauvegarde, audit des accès aux consoles	**Fragilité** : sauvegardes immuables, copies air-gapped hors périmètre AD, comptes de sauvegarde

	d'administration backup	isolés, règle 3-2-1-1-0

Références

Note : Numérotation [111] à [145], suite des Phases 1–3 ([1]–[110]).

[111] Lockheed Martin, « Cyber Kill Chain Framework — C2, Lateral Movement, Actions on Objectives ». https://www.lockheedmartin.com/en-us/capabilities/cyber/cyber-kill-chain.html

[112] CrowdStrike, « Indirect Prompt Injection Attacks: Hidden AI Risks » (mouvement latéral via agents compromis), décembre 2025. https://www.crowdstrike.com/en-us/blog/indirect-prompt-injection-attacks-hidden-ai-risks/

[113] TechTarget / SearchSecurity, « News brief: AI threats to shape 2026 cybersecurity ». M. Sachs (CIS), J. Grady (Omdia), P. Harrington (Forrester). https://www.techtarget.com/searchsecurity/news/366637045/

[114] MITRE ATT&CK, « T1059 Command and Scripting Interpreter » et Living-off-the-Land Binaries, v15. https://attack.mitre.org/

[115] Verizon, « 2025 Data Breach Investigations Report » (DBIR). 74 % violations impliquent identités compromises. Kill chain AD typique.

[116] CIS, « Mimikatz: The Finest in Post-Exploitation » (sekurlsa, lsadump, DCSync, Golden Ticket). https://www.cisecurity.org/insights/blog/mimikatz-the-finest-in-post-exploitation

[117] S. Metcalf (ADSecurity.org), « Attack Methods for Gaining Domain Admin Rights in Active Directory ». Credential Theft Shuffle. https://adsecurity.org/?p=2362

[118] Netwrix, « DCSync Attack Using Mimikatz Detection ». https://www.netwrix.com/privilege_escalation_using_mimikatz_dcsync.html

[119] Stellar Cyber, « Top 10 Agentic SOC Platforms for 2026 ». Cas Change Healthcare (190M patients, 9 jours, credential unique). https://stellarcyber.ai/learn/top-10-agentic-soc-platforms/

[120] C. Schneider(2026), Promptware Kill Chain. OWASP Top 10 for Agentic Applications 2026, ASI01 Agent Goal Hijack. https://christian-schneider.net/blog/prompt-injection-agentic-amplification/

[121] InstaTunnel, « Prompt-to-Insider Threat: When AI Agents Become Double Agents ». CVE-2025-32711 EchoLeak (M365 Copilot, CVSS 9.3), février 2026. https://instatunnel.my/blog/prompt-to-insider-threat/

[122] HackerNoob / Information 2026, 17(1), 54, « Prompt Injection Attacks in LLM and AI Agent Systems: A Comprehensive Review » (taux succès 66,9–84,1 %, 73 % déploiements affectés). doi:10.3390/info17010054

[123] Mithril Security, « PoisonGPT: How to poison LLM supply chain on Hugging Face » (ROME, GPT-J-6B, Δ 0,1 %). https://blog.mithrilsecurity.io/poisongpt/

[124] Barracuda Networks, « PoisonGPT: Weaponizing AI for disinformation », sept. 2025. https://blog.barracuda.com/2025/09/11/poisongpt-weaponizing-ai-disinformation

[125] Anthropic, « Sleeper Agents: Training Deceptive LLMs that Persist Through Safety Training », 2024. ArXiv.

[126] Phase 3, [80]–[82]. Exfiltration via skill OpenClaw : curl C2 encapsulé dans trafic HTTPS légitime du gateway.

[127] S. Willison, « AI agents have a lethal trifecta of risks » (private data + untrusted content + external communication). Réf. Phase 3 [90].

[128] Sophos, « The State of Ransomware 2025 ». 94 % attaques ciblent sauvegardes, 57 % réussissent.

[129] OWASP, « LLM01:2025 Prompt Injection » et « LLM03:2025 Supply Chain ». https://genai.owasp.org/

[130] Microsoft, « Guidance to mitigate critical threats to AD Domain Services in 2025 ». https://www.microsoft.com/en-us/windows-server/blog/2025/12/09/

[131] Control Risks, « The Agentic Shift: How Autonomous AI Is Reshaping the Global Threat Landscape ». https://www.controlrisks.com/our-thinking/insights/the-agentic-shift

[132] Obsidian Security, « Prompt Injection Attacks: The Most Common AI Exploit in 2025 » (privilèges excessifs SaaS, mouvement latéral). https://www.obsidiansecurity.com/blog/prompt-injection

[133] Lakera, « The Year of the Agent: Q4 2025 Attacks » (attaques indirectes < tentatives que directes, system prompt extraction). https://www.lakera.ai/blog/the-year-of-the-agent

[134] Sombrainc, « LLM Security Risks in 2026 » (ServiceNow second-order injection, agent privilege escalation). https://sombrainc.com/blog/llm-security-risks-2026

[135] OpenAI, « Understanding prompt injections: a frontier security challenge », janvier 2026. https://openai.com/index/prompt-injections/

Références croisées — définies dans d'autres phases

Note : *ces références sont définies dans la bibliographie d'une autre phase du document. Elles sont reproduites ici pour permettre une lecture autonome de chaque phase.*

[1] Application of OSINT Methods in Ensuring Cybersecurity, IPSIT Transactions on Internet Research, juillet 2025. https://ipsitransactions.org/journals/papers/tir/2025jul/p5.pdf

→ *Définie en Phase 1*

[25] MITRE ATT&CK, « Active Scanning: Vulnerability Scanning », Sub-technique T1595.002. https://attack.mitre.org/techniques/T1595/002/

→ *Définie en Phase 1*

[27] The Shadowserver Foundation, « CRITICAL: Vulnerable HTTP Report ». https://www.shadowserver.org/what-we-do/network-reporting/vulnerable-http-report/

→ *Définie en Phase 1*

[154] VikingCloud, « 46 Ransomware Statistics 2026 ». Coût total 1,8–5 M$/incident. https://www.vikingcloud.com/blog/ransomware-statistics

→ *Définie en Phase 5*

[160] OWASP, Top 10 for Agentic Applications 2026. Sandboxing agentique, moindre privilège, audit trafic sortant.

→ *Définie en Phase 5*