AYI NEDJIMI Consultants

Registry Advanced : Guide Expert Analyse Technique

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

Analyse forensique avancée du registre Windows : transaction logs (.LOG1/.LOG2), récupération cellules supprimées, REGF format, KTM, techniques...

Registry Forensics Avancé : Transaction Logs et Structures Cachées

Analyse avancée **des hives**, récupération de données supprimées et exploitation des mécanismes transactionnels L'investigation numerique et l'analyse forensique constituent des disciplines essentielles de la cybersecurite moderne. Face a la multiplication des incidents de securite, les analystes DFIR doivent maitriser un ensemble d'outils et de methodologies pour identifier, collecter et analyser les preuves numeriques de maniere rigoureuse. Cet article detaille les techniques avancees, les processus de chaine de custody et les bonnes pratiques pour mener des investigations efficaces dans des environnements complexes. Analyse forensique avancée du registre Windows : transaction logs (.LOG1/.LOG2), récupération cellules supprimées, REGF format, KTM, techniques... Ce guide couvre les aspects essentiels de registry forensics advanced : méthodologie structurée, outils recommandés et retours d'expérience opérationnels. Les professionnels y trouveront des recommandations directement applicables.

Notre avis d'expert
La chaîne de custody numérique est le fondement de toute investigation forensique recevable. Nous observons trop souvent des équipes de réponse à incident qui compromettent involontairement les preuves par manque de procédures formalisées. Un kit forensique prêt à l'emploi devrait être aussi standard qu'un extincteur.

En cas d'incident, seriez-vous capable de retracer le parcours exact de l'attaquant ?

Introduction

Le registre Windows constitue l'épine dorsale du système d'exploitation, stockant une quantité phénoménale d'informations de configuration, de préférences utilisateur, et de métadonnées système. Pour l'analyste forensique, le registre représente une mine d'or d'informations, mais son exploitation complète nécessite une compréhension approfondie de sa structure **interne** complexe, de ses mécanismes de persistance transactionnelle, et des multiples couches de données cachées qu'il contient. Cet article explore les aspects les plus avancés de l'analyse forensique du registre, en se concentrant particulièrement sur les transaction logs, les cellules non allouées, et les techniques de récupération de données supprimées.

L'évolution du registre Windows depuis son introduction dans Windows 3.1 jusqu'aux versions modernes de Windows 11 a considérablement complexifié son **architecture interne**. L'introduction du Kernel Transaction Manager (KTM) dans Windows Vista a bouleversé la gestion transactionnelle du registre, créant de nouvelles opportunités forensiques mais aussi de nouveaux défis. Les fichiers de transaction logs (.LOG1, .LOG2), souvent négligés lors des analyses superficielles, contiennent des informations temporaires cruciales qui peuvent révéler des activités système non visibles dans les hives principales.

La structure en cellules du registre, héritée du format REGF (Registry File Format), présente des caractéristiques uniques qui permettent la récupération de données même après leur suppression logique. Les espaces non alloués au sein des hives, les cellules orphelines, et les structures de métadonnées cachées offrent des opportunités de récupération d'informations que les techniques antiforensiques standard ne peuvent complètement éliminer.

Structure fondamentale du format REGF

Le format REGF (Registry File), identifié par la signature magique "regf" (0x66676572 en little-endian) au début de chaque fichier hive, présente une **architecture** élaborée basée sur un système de pages et de cellules. Chaque hive commence par un en-tête de base de 4096 octets (une page), suivi de bins (blocs de données) contenant les cellules qui stockent les données réelles du registre. Cette structure modulaire permet une gestion efficace de la mémoire et facilite les opérations transactionnelles.

L'en-tête REGF contient des métadonnées critiques pour l'analyse forensique : timestamps de dernière modification, numéros de séquence, checksums, et pointeurs vers les structures racines. Le champ Sequence1 et Sequence2, incrémentés à chaque modification, permettent de détecter les corruptions et de valider l'intégrité de la hive. Le RootCell offset pointe vers la cellule racine de l'arbre du registre, point de départ de toute navigation dans la structure hiérarchique.

Les bins, unités d'allocation de base après l'en-tête, commencent toujours par la signature "hbin" et contiennent une collection de cellules de tailles variables. Chaque bin a une taille minimale de 4096 octets et peut s'étendre jusqu'à 1MB dans les versions modernes de Windows. La gestion de l'espace au sein des bins suit un modèle de liste chaînée, où chaque cellule contient un en-tête indiquant sa taille et son statut d'allocation (positif pour libre, négatif pour alloué).

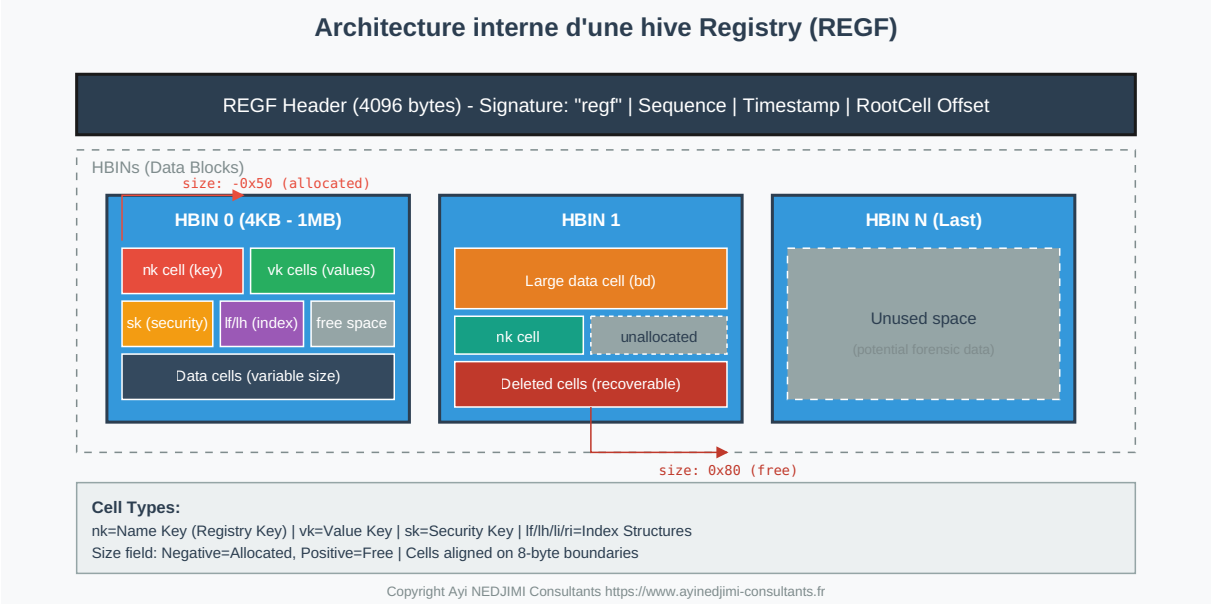
Système de cellules et types de données

Le système de cellules du registre implémente plusieurs types distincts, chacun avec une structure spécifique : nk (name key) pour les clés, vk (value key) pour les valeurs, sk (security key) pour les descripteurs de sécurité, et lf/lh/li/ri (list structures) pour les index. Cette diversité de types permet une organisation efficace des données tout en maintenant la flexibilité nécessaire pour stocker des informations variées.

Cas concret
L'investigation forensique après l'attaque Colonial Pipeline (2021) a permis au FBI de tracer et récupérer 2,3 millions de dollars en Bitcoin versés en rançon au groupe DarkSide. L'analyse des transactions blockchain et la coopération avec les exchanges ont démontré que les cryptomonnaies ne garantissent pas l'anonymat des cybercriminels.

Les cellules nk (clés de registre) contiennent non seulement le nom de la clé mais aussi des métadonnées forensiquement précieuses : timestamps de dernière écriture, compteurs de sous-clés et de valeurs, et pointeurs vers les structures associées. Le timestamp LastWriteTime, stocké en format FILETIME Windows, permet de reconstruire la chronologie des modifications. Les flags dans la cellule nk révèlent des caractéristiques importantes comme la présence de liens symboliques ou l'état de prédéfinition de la clé.

Les cellules vk stockent les données des valeurs du registre avec une gestion complexee selon la taille des données. Pour les petites valeurs (≤ 4 octets), les données sont stockées directement dans le champ DataOffset de la cellule vk, une optimisation qui élimine une indirection. Pour les valeurs plus grandes, le DataOffset pointe vers une cellule de données séparée. Les valeurs très grandes utilisent le mécanisme des big data cells (bd), introduit dans Windows 10, permettant le stockage de valeurs jusqu'à 1MB.



Architecture interne d'une hive Registry avec ses HBINs et cellules

Gestion de la sécurité et descripteurs

Les descripteurs de sécurité (cellules sk) implémentent un système de partage intelligent où plusieurs clés peuvent référencer le même descripteur de sécurité, optimisant l'utilisation de l'espace. Chaque cellule sk contient un Security Descriptor au format Windows standard, incluant les SIDs du propriétaire et du groupe, ainsi que les DACL et SACL. L'analyse de ces structures révèle les permissions d'accès et peut identifier des modifications de sécurité suspectes.

Vos preuves numériques seraient-elles recevables devant un tribunal ?

Le chaînage des cellules sk forme une liste circulaire doublement chaînée, permettant une énumération efficace de tous les descripteurs de sécurité dans la hive. Cette structure facilite la détection d'anomalies : descripteurs orphelins, références circulaires brisées, ou modifications non autorisées des permissions. Les timestamps associés aux changements de sécurité, bien que non directement stockés dans les cellules sk, peuvent souvent être corrélés avec les logs d'événements de sécurité.

L'analyse forensique des descripteurs de sécurité révèle souvent des tentatives de dissimulation ou d'escalade de privilèges. Les modifications des ACL pour accorder des permissions inappropriées, la suppression des entrées d'audit SACL, ou l'ajout de SIDs inconnus sont des indicateurs d'activité malveillante. La comparaison avec les descripteurs de sécurité standard du système permet d'identifier les déviations significatives.

Index et structures de navigation

Les structures d'index (lf, lh, li, ri) optimisent la navigation dans l'arbre du registre en maintenant des listes triées de sous-clés. Le type lf (leaf fast) stocke une liste simple de pointeurs vers les cellules nk des sous-clés avec les quatre premiers caractères de chaque nom pour accélération de la recherche. Le type lh (leaf hash) utilise un hash des noms pour une recherche encore plus rapide. Les types li (leaf index) et ri (root index) gèrent les grandes collections de sous-clés via des structures multi-niveaux.

L'analyse de ces structures d'index révèle des informations sur l'historique des modifications. Les entrées orphelines dans les index, pointant vers des cellules non valides ou réutilisées, peuvent indiquer des suppressions récentes. Les incohérences entre le compteur de sous-clés dans la cellule nk parente et le nombre réel d'entrées dans l'index suggèrent des corruptions ou des manipulations. La reconstruction manuelle des index permet parfois de récupérer des références à des clés supprimées mais dont les cellules existent encore.

Les mécanismes de cache et d'optimisation du registre créent parfois des duplications temporaires de structures d'index. Ces duplications, visibles dans l'espace non alloué ou les transaction logs, peuvent révéler des états intermédiaires du registre non visibles dans la vue actuelle. L'analyse comparative de ces structures dupliquées permet de reconstruire la séquence de modifications et d'identifier des tentatives de manipulation rétroactive.

Artefact	Localisation	Information extraite
Registre	SYSTEM, SAM, SOFTWARE	Configuration, comptes, services
Event Logs	Security, System, Application	Connexions, erreurs, alertes
Prefetch	C:\Windows\Prefetch	Programmes executes et timestamps
MFT	$MFT sur volume NTFS	Fichiers crees, modifies, supprimes

Partie 2 : Transaction Logs - Architecture et exploitation forensique

Mécanisme transactionnel du registre moderne

Le système de transaction logs du registre Windows, considérablement renforcé depuis Windows Vista avec l'introduction du Kernel Transaction Manager (KTM), assure l'intégrité des modifications du registre via un mécanisme de journalisation write-ahead. Les fichiers .LOG1 et .LOG2 associés à chaque hive principale contiennent les enregistrements des transactions en cours et récemment complétées, offrant une fenêtre unique sur les modifications temporaires et les états intermédiaires du registre.

La structure des transaction logs suit le format CLFS (Common Log File System), avec un en-tête identifié par la signature "HvLE" pour .LOG1 et des structures de données spécifiques pour la gestion transactionnelle. Chaque transaction est enregistrée avec un LSN (Log Sequence Number) unique, permettant l'ordonnancement chronologique précis des opérations. Les enregistrements incluent les dirty pages (pages modifiées), les dirty vectors (bitmaps indiquant les octets modifiés), et les données originales pour permettre le rollback. Pour approfondir, consultez Windows Server 2025.

L'analyse des transaction logs révèle des modifications qui peuvent ne jamais avoir été commitées dans la hive principale. Les transactions avortées, les rollbacks système, et les modifications en cours au moment d'un crash système sont préservés dans ces logs. Cette caractéristique est particulièrement précieuse pour l'analyse d'incidents où le système a été brutalement interrompu ou lorsqu'un malware a tenté de modifier le registre sans succès.

Structure détaillée des fichiers LOG1 et LOG2

Les fichiers .LOG1 contiennent le journal principal des transactions avec une structure complexe organisée en plusieurs sections. L'en-tête de 512 octets inclut la signature, les checksums, et les métadonnées de synchronisation. La section des dirty vectors suit, mappant précisément quelles pages de la hive ont été modifiées. Les dirty pages elles-mêmes constituent la majeure **partie** du fichier, contenant les nouvelles versions des pages modifiées.

Les fichiers .LOG2, introduits pour améliorer la résilience, servent de buffer secondaire et contiennent des informations complémentaires. Ils stockent notamment les transactions de taille importante qui ne peuvent tenir dans .LOG1, ainsi que les métadonnées de recovery pour les opérations multi-phases. La coordination entre .LOG1 et .LOG2 suit un protocole strict garantissant qu'au moins une copie cohérente des données est toujours disponible.

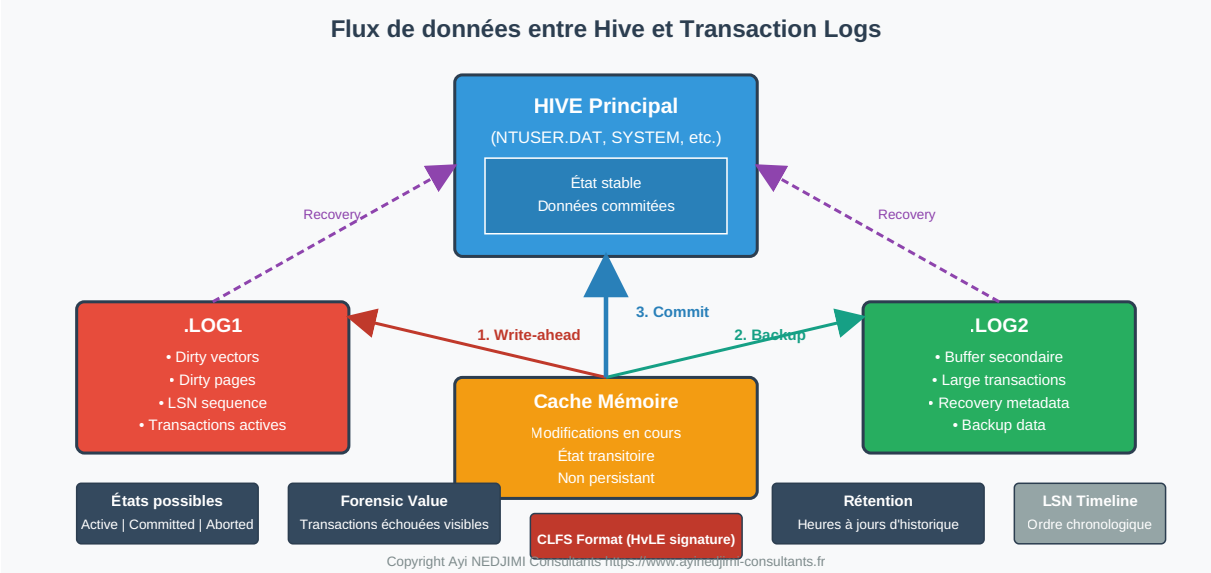
L'exploitation forensique des transaction logs nécessite la compréhension du cycle de vie des transactions. Une transaction commence par l'écriture dans les logs, suivie de la modification en mémoire, et finalement le flush vers la hive sur disque. Entre chaque étape, des informations différentes sont disponibles, créant des opportunités pour récupérer des états intermédiaires. Les transactions partiellement appliquées, visibles uniquement dans les logs, peuvent révéler des tentatives de modification avortées par des mécanismes de sécurité.

Récupération de données depuis les transaction logs

La récupération de données depuis les transaction logs requiert des techniques spécialisées dépassant la simple analyse de la hive principale. Les dirty pages dans les logs peuvent contenir des versions antérieures de clés et valeurs, permettant la reconstruction d'états historiques du registre. Cette capacité est particulièrement utile pour identifier des modifications temporaires effectuées par des malwares ou des outils antiforensiques.

L'algorithme de recovery analysis implique la reconstruction de la séquence transactionnelle complète. En parcourant les LSN dans l'ordre chronologique, il est possible de reconstituer l'évolution du registre sur la période couverte par les logs (typiquement les dernières heures à jours selon l'activité). Les conflits entre transactions, les rollbacks, et les reprises après crash offrent des indices sur les activités système anormales.

Les métadonnées transactionnelles incluent des informations sur le processus initiateur, le contexte de sécurité, et le timing précis des opérations. Ces données, non présentes dans la hive finale, permettent l'attribution des modifications à des processus spécifiques et la corrélation avec d'autres événements système. L'analyse des patterns transactionnels peut révéler des comportements automatisés caractéristiques de malwares ou de scripts d'administration.



Flux transactionnel entre hive principale et transaction logs

Analyse différentielle et détection d'anomalies

L'analyse différentielle entre la hive principale et les transaction logs révèle des divergences significatives pour l'investigation forensique. Les modifications présentes dans les logs mais absentes de la hive indiquent des transactions avortées ou des rollbacks, souvent symptomatiques d'échecs d'exploitation ou de détection par des mécanismes de sécurité. Inversement, des modifications dans la hive sans traces correspondantes dans les logs suggèrent des manipulations directes du fichier hive, technique parfois utilisée par des rootkits aboutis.

La détection d'anomalies dans les patterns transactionnels constitue une approche puissante pour identifier les activités malveillantes. Les transactions de taille inhabituelle, les rafales de modifications sur des clés sensibles, ou les patterns temporels atypiques (activité nocturne sur un poste de travail, par exemple) méritent une investigation approfondie. L'analyse statistique des fréquences de modification par clé permet d'identifier les outliers potentiellement liés à des compromissions.

Les techniques de timeline reconstruction utilisant les transaction logs permettent une granularité temporelle supérieure à celle disponible via les timestamps des clés. Chaque transaction étant horodatée précisément, il devient possible de reconstruire une chronologie détaillée des événements, même pour des modifications rapprochées dans le temps. Cette précision temporelle est cruciale pour corréler les activités du registre avec d'autres sources de données forensiques.

Partie 3 : Cellules non allouées et récupération de données supprimées

Mécanismes d'allocation et libération des cellules

Le système d'allocation des cellules dans le registre Windows suit un modèle de gestion de mémoire poussé optimisé pour les performances plutôt que pour la sécurité. Lorsqu'une cellule est libérée (suite à la suppression d'une clé ou valeur), elle n'est pas immédiatement écrasée mais simplement marquée comme disponible dans les métadonnées du bin. Cette approche crée des opportunités forensiques significatives, car le contenu des cellules supprimées persiste jusqu'à leur réutilisation effective.

L'algorithme d'allocation privilégie la réutilisation de cellules de taille appropriée pour minimiser la fragmentation. Les cellules libres sont chaînées dans des listes par taille, facilitant l'allocation rapide. Cependant, ce mécanisme signifie que des petites cellules peuvent rester non réutilisées pendant de longues périodes si aucune demande de taille correspondante n'est effectuée. L'analyse statistique de l'utilisation de l'espace révèle souvent des patterns de fragmentation caractéristiques de certains types d'activités. Les recommandations de MITRE ATT&CK constituent une reference essentielle.

La coalescence des cellules adjacentes libres, mécanisme d'optimisation de l'espace, peut partiellement détruire les données des cellules supprimées. Cependant, ce processus n'est pas systématique et dépend de facteurs comme la charge système et les patterns d'allocation. L'identification des zones où la coalescence ne s'est pas produite révèle souvent des îlots de données historiques préservées, véritables capsules temporelles de l'état passé du registre.

Techniques de carving dans l'espace non alloué

Le carving de cellules dans l'espace non alloué nécessite une approche méthodique basée sur la reconnaissance de signatures et la validation de structures. Chaque type de cellule possède des caractéristiques identifiables : magic numbers spécifiques, structures de pointeurs cohérentes,

et formats de données prévisibles. La signature "nk" (0x6B6E) pour les clés, "vk" (0x6B76) pour les valeurs, et "sk" (0x6B73) pour la sécurité permettent l'identification initiale des cellules candidates.

La validation des cellules carvées implique plusieurs niveaux de vérification. La cohérence de la taille déclarée avec la structure réelle, la validité des pointeurs internes (même s'ils pointent vers des zones maintenant réallouées), et la conformité aux formats de données attendus constituent autant de critères de validation. Les cellules partiellement écrasées peuvent encore contenir des informations exploitables, nécessitant des techniques de reconstruction partielle.

L'automatisation du carving via des scripts spécialisés permet le traitement de hives volumineuses. Les algorithmes de pattern matching, utilisant des expressions régulières adaptées aux structures binaires du registre, identifient les candidats potentiels. Le scoring basé sur multiple critères (complétude de la structure, cohérence des timestamps, validité des chaînes de caractères) permet de prioriser les cellules récupérées selon leur fiabilité.

Reconstruction de structures supprimées

La reconstruction de structures complètes depuis des cellules supprimées représente un défi technique considérable mais offre des récompenses forensiques substantielles. Une clé de registre supprimée peut être partiellement reconstruite en assemblant sa cellule nk avec les cellules vk de ses valeurs et potentiellement ses sous-clés, même si ces éléments sont dispersés dans l'espace non alloué. Cette approche de puzzle numérique nécessite la compréhension profonde des relations entre cellules. Pour approfondir, consultez Attaques sur CI/CD (GitHub.

Les techniques de reconstruction probabiliste utilisent les métadonnées résiduelles pour inférer les relations entre cellules orphelines. Les timestamps proches, les patterns de nommage similaires, et les structures de sécurité communes suggèrent des associations possibles. La validation croisée avec d'autres sources (logs d'événements mentionnant les clés supprimées, traces dans la mémoire système) permet de confirmer les reconstructions hypothétiques.

La persistance différentielle des types de cellules offre des opportunités de reconstruction partielle même quand certains éléments sont définitivement perdus. Les cellules sk (sécurité), partagées entre plusieurs clés, survivent souvent plus longtemps que les cellules nk individuelles. Les grandes cellules de données, moins susceptibles d'être réutilisées rapidement, préservent souvent le contenu de valeurs importantes même après la suppression de leurs métadonnées.

Analyse temporelle des suppressions

L'analyse temporelle des suppressions dans le registre révèle des patterns comportementaux significatifs. Bien que les cellules supprimées ne contiennent pas directement de timestamp de suppression, plusieurs techniques permettent d'estimer le moment de la suppression. L'analyse de l'allocation subsequent des cellules environnantes, la corrélation avec les transaction logs, et la comparaison avec des backups ou shadow copies fournissent des bornes temporelles.

Les vagues de suppression massives, identifiables par des zones contiguës d'espace non alloué, suggèrent l'utilisation d'outils de nettoyage ou de scripts automatisés. Le pattern de suppression sélective, où seules certaines clés ou valeurs spécifiques sont supprimées tout en laissant les structures environnantes intactes, indique une action manuelle ciblée ou un malware avancé tentant de couvrir ses traces.

La corrélation entre les suppressions dans le registre et d'autres activités système permet de reconstruire des scénarios d'incident complets. Les suppressions de clés de persistence malware correspondent souvent à l'exécution d'outils de nettoyage. Les suppressions de clés de configuration avant une modification système majeure révèlent des tentatives de manipulation. L'absence de suppressions attendues peut être tout aussi révélatrice, indiquant l'échec d'un processus de nettoyage ou la présence de mécanismes de protection.

Partie 4 : ShellBags, UserAssist et autres artefacts cachés

ShellBags : fenêtre sur la navigation utilisateur

Les ShellBags représentent l'un des artefacts les plus riches et persistants du registre Windows, stockant les préférences d'affichage de l'Explorateur pour chaque dossier visité. Localisés principalement dans USRCLASS.DAT sous Local Settings\Software\Microsoft\Windows\Shell\BagMRU et Bags, ces structures complexes encodent non seulement les préférences visuelles mais aussi une trace détaillée de la navigation utilisateur, incluant les dossiers réseau, les périphériques amovibles, et même les archives ZIP explorées.

La structure BagMRU implémente un arbre de navigation encodant le chemin complet vers chaque dossier visité. Chaque nœud contient un Shell Item identifiant le dossier, similaire aux structures trouvées dans les fichiers LNK. L'analyse de ces Shell Items révèle des métadonnées précieuses : timestamps de création et accès, attributs de fichiers, et pour les dossiers réseau, les chemins UNC complets. La persistance de ces informations même après la suppression des dossiers originaux crée une trace forensique indélébile.

Les Bags associés stockent les préférences visuelles spécifiques : mode d'affichage (icônes, liste, détails), tri, position des fenêtres, et colonnes affichées. Ces préférences, apparemment anodines, révèlent des patterns comportementaux. Un utilisateur triant systématiquement par date de modification dans certains dossiers pourrait chercher des fichiers récents. L'activation de colonnes spécifiques (comme "Owner" ou "Attributes") suggère une investigation ou une attention particulière à ces métadonnées.

L'analyse temporelle des ShellBags via les timestamps LastWrite des clés de registre permet de reconstruire la chronologie de navigation. Cependant, ces timestamps ne reflètent que la dernière modification des préférences, pas nécessairement le dernier accès. La corrélation avec les numéros de séquence MRU (Most Recently Used) et l'analyse des NodeSlots permettent une reconstruction plus précise de l'ordre de navigation.

UserAssist : décodage ROT13 et analyse comportementale

Le mécanisme UserAssist, stocké sous NTUSER.DAT\Software\Microsoft\Windows\CurrentVersion\Explorer\UserAssist, enregistre des statistiques détaillées sur l'exécution des programmes et l'accès aux raccourcis. L'encodage ROT13 appliqué aux noms de programmes, vestige de sécurité par obscurité, n'offre aucune protection réelle mais nécessite un décodage pour l'analyse. Chaque entrée contient un compteur d'exécution, un timestamp de dernière exécution, et dans les versions récentes, la durée totale de focus de l'application.

La structure binaire de 72 octets (ou 16 octets dans les versions antérieures à Windows 7) associée à chaque entrée UserAssist encode des métriques comportementales précieuses. Le compteur d'exécution, stocké à l'offset 4, révèle la fréquence d'utilisation. Le timestamp FILETIME à l'offset 60 indique la dernière exécution. Le temps de focus, nouveauté de Windows 7+, quantifie l'engagement réel de l'utilisateur avec l'application, différenciant les lancements accidentels des utilisations substantielles.

L'analyse des patterns UserAssist révèle des anomalies comportementales significatives. L'exécution unique d'outils système rarement utilisés (regedit, cmd avec privilèges élevés) peut indiquer une activité administrative suspecte. Les pics d'activité nocturne ou pendant les absences connues de l'utilisateur suggèrent des compromissions. La présence d'outils de hacking ou de programmes de communication chiffrée mérite une investigation approfondie.

Les GUIDs de catégorie UserAssist (CEBFF5CD pour les exécutables, F4E57C4B pour les raccourcis) permettent de différencier les types d'accès. L'analyse croisée entre ces catégories révèle des incohérences : un programme exécuté fréquemment selon une catégorie mais absent de l'autre suggère des méthodes de lancement non conventionnelles, possiblement via des scripts ou des exploits.

RecentDocs et flux de données cachés

Les entrées RecentDocs, stockées dans plusieurs emplacements du registre, maintiennent des listes extensives de documents récemment accédés, organisées par extension. Au-delà de la simple liste sous Explorer\RecentDocs, des structures complexes incluent les OpenSavePidMRU (dialogues d'ouverture/sauvegarde), ComDlg32 (dialogues communs), et TypedPaths (chemins tapés manuellement). Cette multiplicité de sources permet une reconstruction détaillée des interactions utilisateur avec les fichiers.

Le format binaire des entrées RecentDocs encode plus que de simples noms de fichiers. Chaque entrée contient un Shell Link structure miniature incluant des métadonnées du fichier, des informations de volume, et parfois des chemins réseau complets. L'analyse de ces structures révèle des accès à des fichiers supprimés, des volumes déconnectés, ou des partages réseau temporaires. La persistence de ces informations après la suppression des fichiers originaux étend considérablement la fenêtre forensique.

Les streams de données alternatifs dans les valeurs de registre, rarement exploités, peuvent contenir des informations cachées. Certaines applications stockent des métadonnées étendues, des thumbnails, ou même des contenus de fichiers complets dans des valeurs de registre

binaires. L'analyse de l'entropie et des signatures dans ces grandes valeurs binaires révèle parfois des données structurées non documentées, incluant des fragments de documents, des historiques de modification, ou des caches d'application.

MountPoints2 et traces de périphériques

Les clés MountPoints2 sous Software\Microsoft\Windows\CurrentVersion\Explorer documentent chaque volume monté sur le système, créant un historique persistant des périphériques de stockage connectés. Chaque entrée, identifiée par un GUID de volume ou un chemin réseau, contient des métadonnées sur le périphérique : label de volume, système de fichiers, et letter de lecteur assignée. Cette information persiste longtemps après la déconnexion du périphérique. Pour approfondir, consultez Timeline Analysis : Reconstruction d'Incidents.

L'analyse forensique des MountPoints2 révèle l'utilisation de périphériques USB, de volumes TrueCrypt/VeraCrypt, de partages réseau, et même de téléphones mobiles montés comme stockage. La corrélation entre les GUIDs de volume dans MountPoints2 et d'autres artefacts (fichiers LNK, ShellBags, SETUPAPI logs) permet de reconstruire l'historique complet d'utilisation d'un périphérique spécifique.

Les timestamps associés aux MountPoints2, bien que reflétant la dernière modification de la clé plutôt que la dernière connexion, fournissent des bornes temporelles utiles. L'analyse des sous-clés et valeurs associées révèle des patterns d'utilisation : l'autorun désactivé suggère une conscience de sécurité, les associations de programmes personnalisées indiquent des usages spécifiques, et la présence de certains flags révèle des tentatives de dissimulation.

Partie 5 : Techniques avancées de corrélation et analyse comportementale

Corrélation multi-hives et reconstruction d'activités

L'analyse forensique moderne du registre nécessite une approche holistique corrélant les informations à travers multiple hives. Les cinq hives principales (SYSTEM, SOFTWARE, SAM, SECURITY, NTUSER.DAT) et les hives utilisateur additionnelles (USRCLASS.DAT, Amcache.hve) contiennent des informations complémentaires qui, analysées ensemble, révèlent une image complète des activités système et utilisateur. Cette corrélation multi-hives permet de valider les findings, détecter les incohérences, et reconstruire des chaînes d'événements complexes.

La synchronisation temporelle entre les hives représente un défi technique majeur. Les timestamps LastWriteTime des clés sont mis à jour indépendamment dans chaque hive, créant des décalages temporels qui doivent être reconciliés. L'utilisation des ControlSet timestamps dans SYSTEM comme référence temporelle, combinée avec l'analyse des transaction logs pour une granularité fine, permet d'établir une timeline unifiée. Les événements système majeurs (boot, shutdown, installation de logiciels) servent de points de synchronisation entre les hives.

Les patterns de propagation cross-hive révèlent des mécanismes système et des comportements utilisateur. L'installation d'un logiciel crée des entrées coordonnées dans SOFTWARE (configuration globale), NTUSER.DAT (préférences utilisateur), et potentiellement

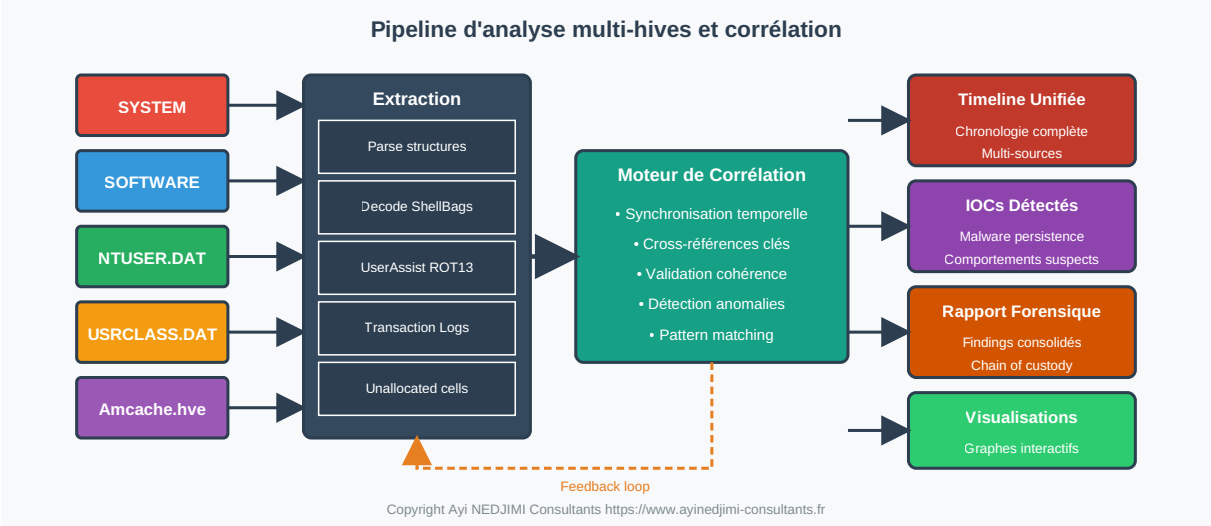
SYSTEM (services, drivers). L'analyse de ces patterns de propagation permet d'identifier des installations incomplètes, des désinstallations partielles, ou des modifications manuelles qui brisent la cohérence attendue.

Analyse prédictive et détection de comportements malveillants

L'application de techniques d'analyse prédictive aux données du registre permet l'identification proactive de compromissions et de comportements anormaux. Les modèles statistiques basés sur les fréquences de modification, les patterns d'accès, et les corrélations entre clés permettent d'établir des baselines comportementales. Les déviations significatives de ces baselines signalent des activités potentiellement malveillantes nécessitant une investigation approfondie.

Les indicateurs de compromission (IOCs) spécifiques au registre incluent des patterns caractéristiques : création de clés dans des emplacements de persistence connus, modification de clés de sécurité critiques, ajout de valeurs binaires obfusquées, ou suppression de clés de logging. L'analyse automatisée utilisant des règles YARA adaptées au format REGF permet la détection rapide de ces IOCs à travers de grandes collections de hives.

L'analyse comportementale temporelle révèle des patterns d'activité malveillante élaborés. Les malwares modernes implémentent souvent des mécanismes de dormance, modifiant le registre selon des patterns temporels spécifiques pour éviter la détection. L'analyse de séries temporelles des modifications du registre, utilisant des techniques comme la décomposition STL (Seasonal and Trend decomposition using Loess) ou l'analyse de Fourier, permet d'identifier ces patterns cycliques ou périodiques cachés.



Pipeline d'analyse multi-hives avec extraction, corrélation et génération de rapports

Mining de données et extraction d'intelligence

Les techniques de data mining appliquées au registre Windows révèlent des informations d'intelligence précieuses au-delà de l'analyse forensique traditionnelle. L'extraction de patterns fréquents utilisant des algorithmes comme FP-Growth ou Apriori identifie des associations entre

modifications de clés qui peuvent révéler des workflows utilisateur ou des séquences d'infection malware. Ces patterns, non évidents dans l'analyse manuelle, émergent de l'analyse automatisée de grandes quantités de données.

Le clustering de comportements utilisateur basé sur les données du registre permet la profilage et l'identification d'anomalies. Les algorithmes de clustering comme DBSCAN ou K-means, appliqués aux vecteurs de features extraits du registre (fréquences d'accès, types d'applications utilisées, patterns de navigation), groupent les utilisateurs par comportement similaire. Les outliers identifiés par ces analyses méritent une investigation forensique approfondie.

L'extraction d'entités et de relations depuis les valeurs de registre textuelles révèle des connexions non évidentes. Les URLs, adresses email, chemins de fichiers, et identifiants uniques stockés dans le registre forment un graphe de relations exploitable. L'analyse de ce graphe utilisant des algorithmes de centralité et de détection de communautés révèle des nœuds critiques et des groupes d'entités liées, facilitant la compréhension de l'infrastructure d'une attaque ou d'un incident.

Automatisation et scripting forensique avancé

Le développement de frameworks d'automatisation spécialisés pour l'analyse du registre permet le traitement efficace de cas complexes. Ces frameworks doivent gérer la diversité des formats de hive (versions Windows différentes, corruptions partielles), l'extraction parallélisée de données, et la correlation automatisée entre sources. L'architecture modulaire permet l'ajout de plugins pour de nouveaux types d'analyse sans refonte du système core.

Les pipelines de traitement automatisés implémentent des workflows complets depuis l'acquisition jusqu'à la génération de rapports. L'acquisition utilise des techniques de copie shadow-tolerantes, l'extraction parse les structures binaires avec gestion d'erreurs robuste, l'analyse applique des batteries de tests et détections, et la génération de rapports produit des visualisations interactives et des timelines navigables. Chaque étape maintient la chaîne de custody et la traçabilité des opérations.

L'utilisation de langages de requête spécialisés pour le registre facilite l'extraction ciblée d'informations. Des langages comme RegQL (Registry Query Language) ou des adaptations de SQL pour les structures hiérarchiques permettent des requêtes complexes avec jointures entre hives, conditions temporelles, et agrégations. Ces requêtes, réutilisables et partageables, standardisent l'extraction d'indicateurs forensiques communs.

Partie 6 : Cas pratiques et études d'incidents réels

Cas d'étude : Investigation d'une exfiltration via registry tunneling

L'investigation d'un cas complexe d'exfiltration de données a révélé l'utilisation innovante du registre Windows comme canal de communication caché. L'attaquant exploitait la capacité du registre à stocker de grandes valeurs binaires (jusqu'à 1MB dans Windows 10+) pour créer un canal de données bidirectionnel. Les données sensibles étaient fragmentées, encodées, et stockées dans des valeurs de registre apparemment légitimes sous des clés de configuration d'applications communes.

L'analyse initiale des hives n'a révélé aucune anomalie évidente : les clés utilisées existaient légitimement, les noms de valeurs suivaient les conventions attendues, et les permissions de sécurité étaient standard. C'est l'analyse de l'entropie des valeurs binaires qui a révélé l'anomalie : certaines valeurs présentaient une entropie caractéristique de données chiffrées ou compressées, incompatible avec leur usage supposé de stockage de préférences.

L'examen des transaction logs a révélé le pattern temporel de l'exfiltration. Les modifications des valeurs suspectes suivaient un pattern régulier : écriture de nouvelles données toutes les heures, lecture par un processus système compromis, puis suppression après confirmation de transmission. Les logs contenaient des fragments de données en clair avant chiffrement, permettant l'identification partielle du contenu exfiltré. Pour approfondir, consultez Memory Forensics 2026 : Volatility 3 Avance.

La reconstruction de la chaîne d'attaque complète a nécessité la corrélation entre plusieurs **hives et** systèmes. Le malware initial, identifié via les traces de persistence dans RunOnce, installait un service caché documenté dans SYSTEM. Ce service utilisait des clés dans SOFTWARE pour la configuration et NTUSER.DAT pour le stockage temporaire. La communication avec le serveur de commande et contrôle était orchestrée via des valeurs dans HKEY_CURRENT_USER, modifiées selon un algorithme de génération de domaines stocké dans le registre lui-même.

Analyse antiforensique : Contournement des outils standards

L'analyse d'un incident impliquant des techniques antiforensiques avancées a mis en évidence les limitations des outils forensiques standards. L'attaquant avait implémenté plusieurs couches de dissimulation exploitant les particularités du format REGF et les assumptions des outils d'analyse. Les techniques incluaient l'exploitation de l'espace slack dans les cellules, l'injection de données dans les zones de padding, et la manipulation des structures d'index pour créer des vues incohérentes.

La technique la plus aboutie impliquait la création de "cellules fantômes" - des structures valides selon le format REGF mais non référencées par les index standards. Ces cellules, invisibles aux outils parseant l'arbre depuis la racine, contenaient des backdoors et des données de configuration malveillantes. Leur découverte a nécessité un scanning exhaustif de tous les bins à la recherche de signatures de cellules valides non indexées.

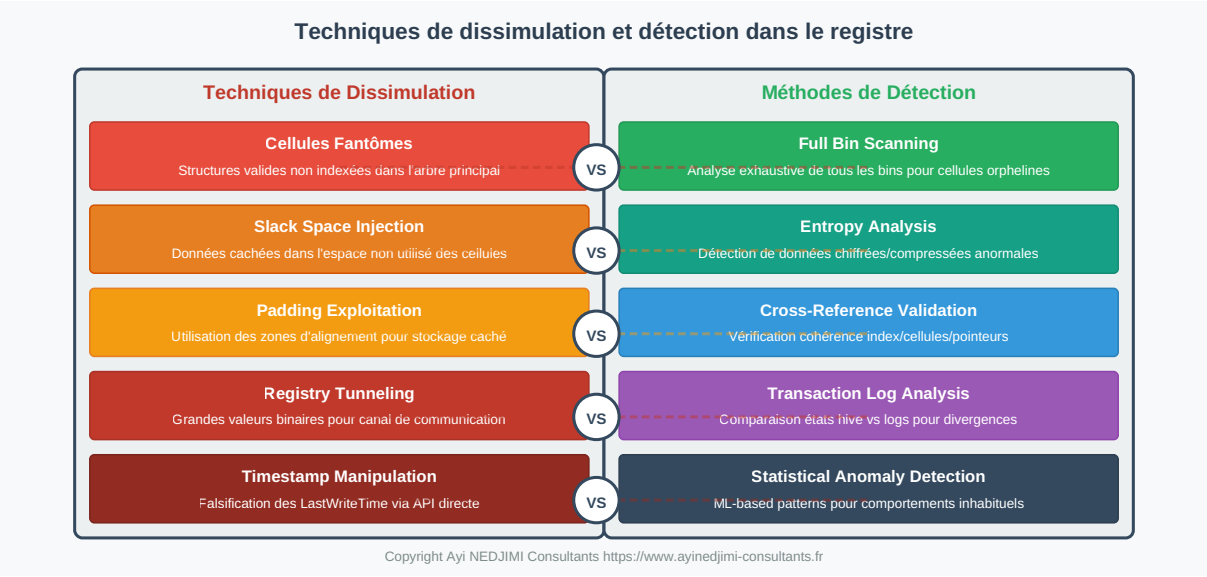
L'attaquant exploitait également les différences de parsing entre Windows et les outils forensiques. Certaines structures malformées étaient ignorées par Windows (qui privilégie la robustesse) mais causaient des erreurs dans les outils forensiques (qui privilégient la conformité stricte). Cette asymétrie créait des zones aveugles où l'attaquant pouvait opérer sans détection. La résolution a nécessité le développement d'un parser tolérant aux erreurs mimant le comportement de Windows.

Reconstruction post-ransomware et récupération de clés

L'investigation suite à une attaque ransomware a démontré l'importance critique de l'analyse des transaction logs et des cellules non allouées pour la récupération. Le ransomware avait tenté de détruire les clés de registre contenant les points de restauration système et les configurations de backup, mais l'analyse des logs a révélé que certaines transactions de suppression n'avaient pas été completées avant l'interruption du processus malveillant.

La récupération des clés de configuration critiques depuis l'espace non alloué a permis la restauration partielle du système. Les cellules nk et vk des clés supprimées étaient largement intactes dans les zones non réallouées des hives. La reconstruction manuelle de ces structures, guidée par les patterns de références croisées dans les transaction logs, a permis de récupérer environ 70% des configurations système critiques.

L'analyse temporelle fine utilisant les LSN des transaction logs a permis d'identifier la fenêtre exacte d'infection et les modifications effectuées par le ransomware. Cette précision temporelle a facilité la distinction entre les modifications légitimes pré-infection et les changements malveillants, permettant une restauration sélective minimisant la perte de données. La corrélation avec les Volume Shadow Copies, dont les métadonnées étaient partiellement préservées dans le registre, a étendu les capacités de récupération.



Bataille entre techniques de dissimulation et méthodes de détection forensiques

Outils essentiels pour l'analyse de registre

- Registry Explorer pour la navigation et l'extraction de donnees
- RegRipper pour l'analyse automatisee des ruches
- RECmd pour l'analyse en ligne de commande
- YARA pour la detection de patterns dans les exports registre
- Autopsy avec le module Registry Analyzer

Questions frequentes

Comment mener une investigation forensique sur un systeme compromis ?

Une investigation forensique debute par la preservation des preuves via une image disque et un dump memoire, suivie de l'analyse des artefacts systeme (registres, journaux d'evenements, fichiers prefetch), la reconstruction de la timeline d'activite et la correlation des indicateurs de compromission pour identifier la source et l'etendue de l'attaque.

Quels sont les outils essentiels pour l'analyse forensique ?

Les outils essentiels pour l'analyse forensique incluent Volatility pour l'analyse memoire, Autopsy et FTK pour l'analyse disque, KAPE et Velociraptor pour la collecte automatisee, Plaso pour la creation de timelines, ainsi que des outils de triage comme Eric Zimmerman's tools pour l'analyse des artefacts Windows.

Pourquoi la chaine de custody est-elle importante en forensique ?

La chaine de custody garantit l'integrite et l'admissibilite des preuves numeriques en documentant chaque etape de manipulation, de la collecte a la presentation. Sans une chaine de custody rigoureuse, les preuves peuvent etre contestees juridiquement et perdre leur valeur probante.

Sources et références : SANS SIFT · MITRE ATT&CK

Conclusion et perspectives futures

L'analyse forensique avancée du registre Windows reste un domaine en évolution constante, confronté à la sophistication croissante des techniques d'attaque et d'antiforensics. La maîtrise des transaction logs, la compréhension profonde des mécanismes d'allocation de cellules, et l'exploitation des structures cachées constituent des compétences essentielles pour l'analyste forensique moderne. Ces techniques avancées révèlent des informations critiques invisibles aux analyses superficielles, permettant la reconstruction d'incidents complexes et la détection de compromissions poussées.

L'évolution vers Windows 11 et les futures versions apportera indubitablement de nouveaux défis et opportunités. L'intégration croissante avec les services cloud, l'expansion des mécanismes de sécurité comme Virtualization-Based Security (VBS), et l'adoption de nouveaux formats de stockage nécessiteront une adaptation continue des techniques d'analyse. Les analystes doivent maintenir une veille technologique active et développer des compétences en reverse engineering pour comprendre les nouvelles structures non documentées.

Le futur de l'analyse forensique du registre s'oriente vers l'intelligence artificielle et l'apprentissage automatique pour la détection d'anomalies et la prédiction de comportements malveillants. Les techniques de deep learning appliquées aux patterns temporels du registre, la détection d'anomalies non supervisée, et l'analyse prédictive basée sur des modèles

comportementaux représentent des domaines de recherche prometteurs. L'automatisation croissante permettra le traitement de volumes de données exponentiellement plus importants tout en maintenant la précision de l'analyse.

Recommandations pour les praticiens

Les professionnels du forensics doivent adopter une approche systématique et rigoureuse de l'analyse du registre. La documentation exhaustive des méthodologies, la validation croisée des findings, et la maintenance d'une chaîne de custody irréprochable restent fondamentales. L'investissement dans la formation continue et le développement d'outils personnalisés adaptés aux besoins spécifiques améliore significativement l'efficacité des investigations.

La collaboration au sein de la communauté forensique facilite le partage de connaissances et l'identification de nouvelles techniques. La publication de recherches, le développement d'outils open source, et la participation aux conférences spécialisées enrichissent l'écosystème forensique global. La standardisation des formats de rapport et des méthodologies d'analyse améliore l'interopérabilité et la reproductibilité des investigations.

Pour approfondir, consultez les ressources officielles : SANS White Papers, NVD - NIST et ANSSI.

L'anticipation des évolutions futures nécessite une compréhension profonde des tendances technologiques et des vecteurs d'attaque émergents. L'étude des techniques d'attaque proof-of-concept, la recherche sur les vulnérabilités du registre, et l'expérimentation avec de nouvelles méthodologies d'analyse préparent les analystes aux défis futurs. La capacité d'adaptation et l'innovation méthodologique distinguent les experts forensiques capables de résoudre les cas les plus complexes.

En conclusion, l'analyse forensique du registre Windows transcende la simple extraction de données pour devenir une discipline nécessitant expertise technique, pensée analytique, et créativité investigative. La maîtrise des concepts avancés présentés dans cet article - transaction logs, récupération de cellules, structures cachées - fournit les outils nécessaires pour révéler la vérité cachée dans les profondeurs du registre Windows, même face aux tentatives de dissimulation les plus avancées.

> **Ressources open source associées :**
>
> • awesome-cybersecurity-tools — Liste de 100+ outils de cybersécurité

Ayi NEDJIMI Consultants — Expert cybersécurité offensive & intelligence artificielle

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