ROOLS

*Rust Offensive & Observability Lightweight Suite, projet annuel à ESGI Paris, version 1.1

Léo Belarbi
ESGI, Paris, France
l.belarbi@myskolae.fr

Damien Vaurette
ESGI, Paris, France
?@?

Quentin Delneuf
ESGI, Paris, France
?@?

***Abstract*—Rools est un outil multi-OS dédié à la sécurité et à l'observabilité, développé dans le cadre d'un projet annuel académique portant sur l'ingénierie logicielle et la cybersécurité. L'objectif de ce travail est d'explorer et d'implémenter plusieurs domaines clés des outils de la cybersécurité modernes, tout en appliquant des concepts de programmation au sein d'une architecture modulaire et performante basée sur Rust.**
Rools propose une suite unifiée d'outils en ligne de commande légers couvrant des domaines tels que l'inspection de processus, l'analyse mémoire, la surveillance réseau, l'audit Wi-Fi, l'analyse Active Directory et le scan de vulnérabilités web. Ce projet vise non seulement à fournir un kit fonctionnel et extensible, mais également à démontrer une compréhension approfondie des différentes couches logicielles, des surfaces d'attaque et des techniques défensives au sein du spectre de la cybersécurité.

***Index Terms*—cybersecurity, forensics, network, monitoring, memory, analysis, system, rust, programming, tools, vulnerability**

I. Introduction

A. Problématique

Les environnements informatiques modernes sont devenus de plus en plus complexes, hétérogènes et interconnectés. Cette évolution a élargi la surface d'attaque des infrastructures numériques et a rendu la surveillance de la sécurité plus difficile que jamais. Les analystes et administrateurs système s'appuient souvent sur de multiples outils spécialisés — chacun couvrant une couche spécifique, qu'il s'agisse des processus, de la mémoire, du réseau ou des services d'annuaire — pour diagnostiquer des incidents ou évaluer l'intégrité d'un système. Cependant, ces outils sont fréquemment dépendants d'une plateforme, lourds ou fragmentés selon les écosystèmes, ce qui entraîne des inefficiences opérationnelles. Dans ce contexte, le besoin d'un kit léger, unifié et multiplateforme, capable d'offrir une visibilité cohérente à travers les différentes couches d'un système, apparaît comme un défi particulièrement intéressant.

B. Motivation

Ce projet naît de la volonté de rapprocher les concepts théoriques de cybersécurité et leur mise en œuvre pratique. En développant un ensemble cohérent d'outils couvrant un large éventail de domaines nous cherchons à approfondir notre compréhension du fonctionnement interne des systèmes et des mécanismes d'attaque à différents niveaux. Le projet constitue ainsi à la fois une exploration des capacités techniques de Rust et un cadre d'apprentissage permettant d'intégrer pratiques d'ingénierie logicielle et exigences réelles de la cybersécurité.

C. Travaux connexes

De nombreux outils reconnus existent déjà ...

D. Objectifs du projet

Notre objectif consiste à proposer un outil multiplateforme capable de fonctionner de manière homogène sous Linux, macOS et Windows, accessible en ligne de commande et offrant donc légèreté, portabilité et intégration aisée dans des environnements variés. Par ailleurs, une attention particulière est portée à l'exportation des résultats (sous différents formats) ainsi qu'à la personnalisation des paramètres de chaque outil, de manière à offrir aux utilisateurs un contrôle fin sur les analyses menées et à faciliter l'intégration du toolkit.

E. Objectifs du document

L'objectif de ce document est de présenter de manière exhaustive l'ensemble du travail réalisé autour du développement, depuis sa conception fonctionnelle jusqu'à son évaluation technique et organisationnelle. Nous présentons dans un premier temps le cadre fonctionnel du projet, en explicitant les objectifs poursuivis et les services que le toolkit ambitionne d'offrir. Cette partie permet de dégager une vision claire du périmètre retenu et de la valeur attendue de l'outil. Le document expose ensuite les fondements techniques sur lesquels repose l'outil notamment les décisions d'architecture, les technologies mobilisées et les exigences en matière de sécurité et de performance. Enfin, la dimension organisationnelle du travail est détaillée afin d'expliciter la méthode de conduite du projet, la planification adoptée et les ressources mobilisées, permettant de préciser comment le développement a été structuré.

II. Spécifications fonctionnelles

A. Contexte

Rools se présente exclusivement sous la forme d'un outil en ligne de commande, ce qui favorise son intégration dans des environnements hétérogènes, des scripts automatisés ou des chaînes d'outils existantes. Destiné à être utilisé sur des systèmes Windows et Linux, certaines fonctionnalités peuvent dépendre du contexte d'exécution (droits utilisateur, appartenance à un domaine Active Directory, capacités réseau ou

sans fil), mais le comportement fonctionnel de l'outil reste homogène quel que soit le système. Le niveau de compétence attendu est intermédiaire à avancé, notamment en ce qui concerne l'interprétation des résultats fournis par les analyses.

B. Fonctionnalités

1) Générales: Conçu comme une suite d'outils partageant une logique d'utilisation homogène, chacun répond à un objectif précis tout en respectant un ensemble de principes fonctionnels communs garantissant la lisibilité, la cohérence et la facilité d'exploitation de l'ensemble. D'un point de vue fonctionnel, la suite repose sur une approche modulaire où chaque outil est invoqué indépendamment via une commande dédiée et chaque exécution correspond à une action clairement identifiée. Cette organisation permet à l'utilisateur de cibler précisément le périmètre de son analyse, tout en évitant toute ambiguïté sur les opérations réalisées.

`--help` permet à l'utilisateur d'obtenir des informations sur l'utilisation du toolkit et de chacun des outils qu'il contient, une aide globale est accessible depuis la commande principale et une aide spécifique est disponible pour chaque outil décrivant sont fonctionnement, ses paramètres et des exemples d'utilisation.

`--timeout <sec>` permet de limiter la durée maximale d'exécution d'un outil afin que l'exécution soit interrompue proprement lorsqu'un délai spécifié est atteint, dans ce cas un message explicite informe l'utilisateur de l'arrêt dû au dépassement du temps imparti. En l'absence de valeur définie, aucune limite de temps n'est appliquée.

`--json`, `--yaml`, `--csv`, `--hex`, `--bin` permet à l'utilisateur de choisir le format dans lequel les résultats sont affichés ou exportés selon l'outil. Le comportement attendu est que le format sélectionné s'applique de manière uniforme à toutes les sorties de l'outil concerné et qu'en l'absence de format explicitement spécifié le texte brut est par défaut.

`--export <fichier>` permet à l'utilisateur de sauvegarder les résultats d'exécution d'un outil dans un fichier généré automatiquement à la fin de l'exécution, avec un nom par défaut basé sur l'outil utilisé, la commande invoquée et la date d'exécution, sauf si un nom de fichier est explicitement fourni par l'utilisateur.
`--zip` permet la compression du fichier d'export généré, réduisant l'espace de stockage utilisé et facilitant le transfert des résultats.

2) DirScope: Outil d'inspection et d'analyse fonctionnelle d'un environnement Active Directory. Il a pour objectif de fournir à l'utilisateur une vision claire et structurée des objets du domaine, de leurs relations et des configurations susceptibles d'avoir un impact sur la sécurité ou la gestion des accès. Il repose sur un modèle basé en sous-commande (users, groups, delegation, trust et acl), chacune non cumulable à l'éxecution et correspondant à un domaine d'analyse spécifique. En plus des fonctionnalités générales propres à tous les outils, il possède une options s'appliquant à l'ensemble des sous-commandes et des options spécifiques permettant d'affiner le comportement de chaque sous-commande.

`--ldap-only`, options globales, permet de restreindre l'analyse aux requêtes LDAP strictement passives, elle garantit une utilisation non intrusive de l'outil en excluant toute interaction reposant sur des mécanismes tels que RPC, SMB ou les API système. Cette option est particulièrement adaptée aux contextes d'audit contraints.

La sous-commande `users` permet d'énumérer et d'analyser les comptes utilisateurs du domaine Active Directory courant, en évaluant leur état et leurs paramètres d'authentification. Par défaut, la commande users énumère l'ensemble des comptes utilisateurs du domaine Active Directory courant avec leurs attributs standards.

`--enabled` limite l'analyse aux comptes utilisateurs actifs, permet d'identifier rapidement les comptes actuellement exploitables dans le domaine.

`--disabled` affiche uniquement les comptes désactivés, facilite l'identification de comptes inutilisés ou volontairement neutralisés.

`--locked` identifie les comptes verrouillés, permet de détecter des problèmes d'authentification ou des tentatives d'accès répétées.

`--expired` sélectionne les comptes dont le mot de passe est expiré, met en évidence des situations pouvant empêcher l'authentification ou nécessiter une action administrative.

`--spn` affiche les comptes disposant d'un Service Principal Name, permet d'identifier des comptes de service potentiellement sensibles.

`--asrep` identifie les comptes ne nécessitant pas de pré-authentification Kerberos, met en évidence des comptes exposés à des attaques de type AS-REP roasting.

`--kerberoastable` identifie les comptes exploitables via Kerberoasting, facilite la détection de comptes de service présentant un risque élevé.

`--des` identifie les comptes autorisant des mécanismes de chiffrement obsolètes, permet de détecter des configurations non conformes aux bonnes pratiques de sécurité.

`--password-never-expire` affiche les comptes dont le mot de passe n'expire jamais, met en évidence des comptes

à risque en matière de gestion des identités.

La sous-commande `groups` recense les groupes Active Directory et analyse les relations d'appartenance entre groupes et utilisateurs afin de comprendre la distribution des privilèges. Par défaut, la commande groups liste tous les groupes Active Directory sans résoudre leurs membres.

`--members` inclut les membres des groupes dans les résultats, permet d'identifier quels utilisateurs ou groupes héritent des droits associés.

`--privileged` restreint l'analyse aux groupes à privilèges élevés, facilite l'identification des groupes critiques du domaine.

`--nested` résout les appartenances imbriquées, permet de comprendre les chaînes de privilèges indirectes.

La sous-commande `delegation` analyse les mécanismes de délégation Kerberos configurés dans le domaine Active Directory. Elle vise à identifier des configurations permettant le transfert ou l'extension de privilèges, susceptibles de faciliter des mouvements latéraux. Par défaut, la commande delegation réalise une analyse complète des mécanismes de délégation Kerberos configurés dans le domaine.

`--unconstrained` identifie les comptes ou services configurés avec une délégation non contrainte, détecte des configurations à haut risque permettant l'usurpation de tickets Kerberos.

`--constrained` identifie les délégations contraintes, analyse les services explicitement autorisés à déléguer des identités.

`--resource-based` identifie les délégations basées sur les ressources (RBCD), détecte des relations de confiance configurées au niveau des objets cibles.

`--spn-only` restreint l'analyse aux objets disposant de Service Principal Names, cible les services effectivement exploitables dans des scénarios Kerberos.

La sous-commande `trust` permet d'énumérer et d'analyser les relations de confiance entre le domaine courant et d'autres domaines ou forêts Active Directory. Elle fournit une vision fonctionnelle des dépendances inter-domaines et de leurs implications en matière de sécurité. Par défaut, la commande trust recense toutes les relations de confiance détectées et affiche leurs propriétés principales.

`--external` affiche uniquement les relations de confiance externes, identifie les dépendances avec des domaines tiers.

`--forest` affiche les relations de confiance inter-forêts, analyse les liens de sécurité à large périmètre.

`--direction` affiche le sens de la relation de confiance, permet d'identifier les flux d'authentification autorisés.

`--sid-filtering` vérifie l'état du filtrage des SID, détecte des configurations susceptibles de permettre une élévation de privilèges inter-domaines.

La sous-commande `acl` permet d'auditer les listes de contrôle d'accès (ACL) appliquées aux objets Active Directory sensibles. Elle vise à identifier les autorisations effectives et à mettre en évidence des droits excessifs ou incohérents. Par défaut, la commande acl audite les objets Active Directory à haute valeur (racine du domaine, contrôleurs de domaine et groupes administratifs).

`--object <dn>` cible un objet spécifique à partir de son Distinguished Name, permet une analyse fine et ciblée des permissions.

`--privileged-only` affiche uniquement les autorisations sensibles ou à risque, réduit le bruit en se concentrant sur les permissions critiques.

`--effective-rights <user>` calcule les droits effectifs d'un utilisateur donné, permet d'évaluer précisément les capacités réelles d'un compte sur un objet donné.

3) SysViz: Outil d'inspection et de surveillance des appels système exécutés par les processus d'un système d'exploitation. Il a pour objectif de permettre l'observation du comportement d'exécution d'un processus ou du système dans son ensemble, à travers l'analyse des appels système invoqués, de leur fréquence, de leur latence et de leur évolution en temps réel. Il ne repose pas sur des sous-commandes, l'ensemble de ses fonctionnalités est activé et combiné à l'aide d'options et par défaut, il affiche un flux en temps réel incluant des informations élémentaires jusqu'à une interruption manuelle.

`--pid <id>` restreint l'observation aux appels système émis par un processus spécifique, permet de cibler l'analyse sur un processus précis afin d'en étudier le comportement individuel.

`--name <s>` filtre les événements observés en fonction du nom de l'appel système, facilite l'identification et le suivi d'appels système spécifiques, notamment ceux associés à des actions sensibles (accès fichiers, réseau, création de processus, etc.).

`--stats` active l'affichage de statistiques globales sur les appels système observés, fournit une vue synthétique du volume total d'appels système et de leur fréquence, permettant

d'identifier des comportements anormaux ou intensifs.

`--latency` mesure et affiche la latence moyenne d'exécution des appels système, permet d'évaluer les performances et le coût temporel des appels système, et de détecter d'éventuels ralentissements ou goulots d'étranglement.

`--top <n>` affiche les *n* appels système les plus fréquemment invoqués, permet d'identifier rapidement les appels dominants dans le comportement d'un processus ou du système, facilitant l'analyse de charge ou de comportement.

`--alert <s>` déclenche une alerte lorsqu'un appel système spécifique est observé, permet la détection en temps réel d'événements jugés critiques ou suspects, et favorise une réaction rapide lors de l'apparition de comportements ciblés.

4) MalScan: Outil d'analyse statique de fichiers conçu pour l'inspection de contenus suspects de manière légale, non intrusive et sans exécution du code analysé. Il a pour objectif de fournir des indicateurs permettant d'évaluer le caractère potentiellement malveillant d'un fichier ou d'un ensemble de fichiers, sans modifier leur contenu ni le système hôte. Il ne repose pas sur des sous-commandes, l'ensemble de ses fonctionnalités est activé et combiné à l'aide d'options et s'appuie sur un chemin cible unique fourni en paramètre. Ce chemin peut correspondre à un fichier ou à un répertoire, le mode d'analyse étant automatiquement adapté en fonction du type de cible avec par défaut pour un fichier ses caractéristiques générales et le même principe pour un repertoire avec tous les fichiers qu'il possède. Lorsque le chemin n'est pas spécifiée ou n'existe pas, MalScan affiche l'aide.

`--entropy` calcule l'entropie du contenu des fichiers analysés, permet d'identifier des fichiers potentiellement compressés, chiffrés ou packés, caractéristiques fréquemment associées à des charges malveillantes.

`--upx` active la détection basique de binaires packés avec UPX, facilite l'identification de fichiers exécutables ayant fait l'objet d'un empaquetage, pratique courante dans certains logiciels malveillants pour masquer leur contenu.

`--yara <fichier>` applique des règles YARA personnalisées lors de l'analyse, permet de détecter des menaces connues ou des familles de fichiers spécifiques à l'aide de signatures définies par l'utilisateur.

`--metadata` extrait les métadonnées des fichiers analysés, fournit des informations structurelles telles que la taille, les horodatages, les sections ou les dépendances importées, utiles pour une analyse statique approfondie.

`--deep` force une analyse approfondie de l'ensemble des fichiers analysés, permet d'appliquer l'ensemble des indicateurs disponibles, même dans le cadre d'une analyse de répertoire, au prix d'un temps de traitement plus élevé.

`--recursive` active l'analyse récursive des sous-répertoires, étend le périmètre d'analyse à l'ensemble de l'arborescence, garantissant une couverture complète du chemin ciblé.

`--flags` restreint l'affichage aux fichiers présentant des indicateurs considérés comme suspects, facilite la lecture des résultats lors de l'analyse de répertoires volumineux en mettant l'accent sur les éléments à fort potentiel de risque.

5) MemTrace: Outil léger d'analyse de la mémoire des processus en espace utilisateur. Il est conçu pour permettre l'observation, la recherche et l'extraction contrôlée de données en mémoire à partir d'un processus cible identifié par son identifiant, dans le respect strict des permissions et des mécanismes de protection du système d'exploitation. Il ne repose pas sur des sous-commandes, l'ensemble de ses fonctionnalités est activé et combiné à l'aide d'options et s'appuie sur un processus cible unique fourni en paramètre avec par défaut un comportement passif, se limitant à l'affichage de la cartographie mémoire du processus sans lecture ni extraction de contenu. Lorsque le processus n'est pas spécifiée ou n'existe pas, MemTrace affiche l'aide.

`--pattern <hex>` recherche une séquence hexadécimale définie par l'utilisateur dans la mémoire du processus cible, permet d'identifier la présence de données spécifiques ou de structures connues dans l'espace mémoire du processus.

`--signature <sig>` active la recherche de signatures prédéfinies associées à des comportements ou charges connues, facilite l'identification de motifs mémoire correspondant à des signatures malveillantes ou à des structures caractéristiques de certains processus.

`--fast` active un mode de recherche à périmètre réduit, permet d'accélérer l'analyse mémoire en limitant la profondeur ou l'étendue des régions analysées, au détriment de l'exhaustivité.

`--range <adresse-début>-<adresse-fin>` cible une plage d'adresses mémoire spécifique pour l'extraction, permet d'extraire directement une zone mémoire précise, sans nécessiter de phase préalable de recherche ou de scan global.

`--auto` automatise l'extraction des régions mémoire correspondant aux résultats d'une recherche, facilite l'exploitation des résultats en réduisant les interventions manuelles, tout en conservant un contrôle explicite sur

l'activation de l'extraction.

6) NetWatch: Outil d'inspection réseau et d'analyse du trafic conçu pour l'observation des communications réseau, l'identification de comportements anormaux et l'évaluation de la surface d'exposition réseau. Il permet aussi bien l'analyse passive du trafic que la réalisation de scans actifs de ports afin d'identifier les services exposés. Il repose sur un modèle basé en sous-commande (capture et scan), chacune non cumulable à l'éxecution et correspondant à un domaine d'analyse spécifique. Il s'appuie sur une interface réseau cible unique fourni en paramètre. En plus des fonctionnalités générales propres à tous les outils, il possède des options spécifiques permettant d'affiner le comportement de chaque sous-commande. Lorsque l'interface réseau n'est pas spécifiée, NetWatch utilise l'interface par défaut du système et si l'interface indiquée n'existe pas ou n'est pas accessible, NetWatch affiche l'aide.

La sous-commande `capture` permet la capture et l'analyse en temps réel du trafic réseau transitant par l'interface sélectionnée. Elle fonctionne principalement en mode passif, sans modification ni perturbation des flux observés. Par défaut, NetWatch affiche un flux temps réel contenant des métadonnées de base sur les paquets capturés.

`--pcap` enregistre le trafic réseau brut au format PCAP, permet une analyse ultérieure à l'aide d'outils spécialisés ou l'archivage des communications observées.

`--filter` applique un filtre de capture basé sur des critères réseau tels que les ports, protocoles ou adresses, réduit le volume de trafic analysé en se concentrant sur des communications spécifiques.

`--verbose` affiche des informations détaillées sur les paquets capturés, permet une analyse fine du contenu et des caractéristiques des communications réseau.

`--outbound` met en évidence les connexions sortantes inhabituelles, aide à identifier des communications potentiellement suspectes vers l'extérieur du réseau.

`--frequency` surveille la fréquence du trafic associé à un port ou un service, permet de détecter des volumes anormalement élevés pouvant indiquer un abus ou une activité malveillante.

`--new-service` détecte l'apparition de nouveaux services ou ports en écoute, permet d'identifier des changements inattendus dans l'exposition réseau du système.

`--dns-leak` analyse les requêtes DNS afin d'identifier des comportements anormaux, permet de détecter des fuites DNS ou des résolutions suspectes pouvant indiquer une compromission.

La sous-commande `scan` permet de réaliser un scan actif des ports réseau afin d'identifier les services exposés sur le réseau local ou observé. Cette commande génère volontairement du trafic réseau dans un objectif d'évaluation. Par défaut, un scan TCP rapide est effectué sur les ports de services les plus courants.

`--udp` active l'analyse des ports UDP, permet de détecter des services utilisant des protocoles non orientés connexion.

`--range <plage>` Définit une plage de ports à analyser, permet de cibler précisément les ports d'intérêt et d'adapter la portée du scan.

`--stealth` active un scan TCP discret, réduit la visibilité du scan sur le réseau, limitant les traces laissées par l'analyse.

`--top <n>` restreint le scan aux *n* ports les plus couramment utilisés, permet une évaluation rapide de l'exposition réseau sur les services les plus fréquents.

7) SentryProc: Outil d'audit système dédié à la surveillance des processus actifs. Il permet d'identifier les processus en cours d'exécution, d'analyser leurs permissions, leurs comportements et leurs interactions avec le système, afin de détecter des anomalies ou des activités potentiellement suspectes. Il ne repose pas sur des sous-commandes, l'ensemble de ses fonctionnalités est activé et combiné à l'aide d'options et par défaut, il liste l'ensemble des processus détectés sur le système incluant PID, nom du processus, utilisateur propriétaire, niveau de privilège et chemin d'exécution.

`--user <name>` filtre les processus appartenant à un utilisateur spécifique, permet d'analyser l'activité des processus associés à un compte précis, notamment dans un contexte de contrôle des accès.

`--pid <id>` restreint l'analyse à un processus identifié par son PID, facilite l'inspection ciblée d'un processus suspect ou connu.

`--running` limite l'affichage aux processus actuellement en cours d'exécution, évite l'inclusion de processus arrêtés ou inactifs dans les résultats.

`--service` filtre les processus correspondant à des services système ou à des tâches exécutées en arrière-plan, permet de distinguer les services système des applications utilisateur.

`--sensitive` sélectionne les processus disposant de privilèges élevés ou ayant un rôle critique pour la sécurité, facilite l'identification de processus à fort impact potentiel

sur le système.

`--file-access` surveille les accès aux fichiers effectués par les processus listés, permet d'identifier des comportements anormaux liés à la manipulation de fichiers sensibles.

`--unsigned` identifie les exécutables non signés ou dont la signature n'a pas pu être vérifiée, aide à détecter des binaires potentiellement malveillants ou non approuvés.

`--hidden` détecte des processus dissimulés ou masqués au niveau du système, permet de révéler des tentatives de dissimulation associées à des techniques de rootkits ou de malwares avancés.

`--high-cpu` identifie des processus présentant une consommation CPU anormalement élevée, permet de détecter des boucles anormales, des abus de ressources ou des activités malveillantes.

`--network` détecte les processus disposant d'une activité réseau non autorisée ou inattendue, aide à identifier des communications suspectes initiées par des processus locaux.

8) SkyWave: Outil d'évaluation de la sécurité sans fil destiné à l'analyse des réseaux Wi-Fi environnants. Il permet de détecter les points d'accès, d'analyser leur configuration de sécurité, d'identifier les clients associés et, lorsque l'autorisation est explicitement donnée, de réaliser des tests d'intrusion sans fil contrôlés. Il repose sur un modèle basé en sous-commande (scan, audit, clients et test), chacune non cumulable à l'éxecution et correspondant à un domaine d'analyse spécifique et s'appuie sur une interface réseau cible unique fourni en paramètre. En plus des fonctionnalités générales propres à tous les outils, il possède une options s'appliquant à l'ensemble des sous-commandes et des options spécifiques permettant d'affiner le comportement de chaque sous-commande. Lorsque l'interface réseau n'est pas spécifiée, NetWatch utilise l'interface par défaut du système et si l'interface indiquée n'existe pas ou n'est pas accessible, SkyWave affiche l'aide.

`--channel <num>`, options globales, restreint l'analyse à un canal Wi-Fi spécifique, améliore la précision et les performances lors d'analyses ciblées sur un canal donné.

La sous-commande `scan` permet de découvrir les réseaux Wi-Fi et points d'accès présents à portée de l'interface. Par défaut, tous les points d'accès visibles et cachés sont détectés.

`--ssid <name>` filtre les résultats par nom de réseau (SSID), cible un réseau spécifique lors d'un environnement dense.

`--bssid <mac>` filtre les résultats par adresse MAC du point d'accès, permet une identification précise d'un équipement donné.

`--hidden` tente de détecter les réseaux dont le SSID est masqué, révèle des réseaux volontairement dissimulés.

`--signal <lvl>` filtre les réseaux selon un seuil minimal de puissance du signal, ignore les points d'accès trop éloignés ou instables.

La sous-commande `audit` analyse la configuration de sécurité des réseaux détectés afin d'identifier les protocoles utilisés et les faiblesses connues. Par défaut, la commande audit analyse l'ensemble des réseaux détectés afin d'identifier leurs mécanismes de sécurité et les éventuelles configurations faibles.

`--wep` identifie les réseaux utilisant le protocole WEP, met en évidence des configurations obsolètes et vulnérables.

`--wpa` identifie les réseaux sécurisés en WPA ou WPA2, permet de distinguer les niveaux de sécurité intermédiaires.

`--wpa3` identifie les réseaux utilisant WPA3, évalue l'adoption des standards de sécurité les plus récents.

`--default-pass` recherche des indicateurs d'utilisation de mots de passe par défaut ou faibles, aide à détecter des erreurs de configuration courantes.

`--pmkid` tente la capture de PMKID pour une analyse hors ligne, permet l'évaluation de l'exposition à certaines attaques WPA/WPA2 sans interaction client.

La sous-commande `clients` permet d'énumérer les appareils sans fil communiquant avec les points d'accès détectés. Par défaut, la commande clients énumère tous les clients sans fil observés à portée des points d'accès détectés.

`--ap <bssid>` restreint l'énumération aux clients d'un point d'accès spécifique, permet l'analyse ciblée d'un réseau donné.

`--ip-only` affiche uniquement les adresses IP (lorsqu'elles sont disponibles), simplifie la lecture des résultats pour un usage réseau.

`--mac-only` affiche uniquement les adresses MAC, facilite l'identification matérielle des clients.

La sous-commande `test` permet de réaliser des tests d'intrusion sans fil contrôlés sur des cibles explicitement définies. Par défaut, la commande test n'effectue aucune action active et nécessite la définition explicite de cibles et d'options d'intrusion.

`--ssid <name>` spécifie le SSID cible, définit le réseau visé par les opérations actives.

`--bssid <mac>` spécifie le BSSID cible, améliore la précision et évite les erreurs de ciblage.

`--deauth` envoie des trames de désauthentification, teste la résistance du réseau aux attaques de type déni de service Wi-Fi.

`--evil-twin` déploie un point d'accès frauduleux, évalue la vulnérabilité des clients aux attaques d'usurpation de réseau.

`--handshake-capture` capture les handshakes WPA/WPA2, permet une analyse de robustesse des mécanismes d'authentification.

`--wps` teste l'exposition et la configuration du WPS, identifie les faiblesses liées à l'activation du WPS.

9) WebScout: Scanner léger de vulnérabilités applicatives web. Il est conçu pour identifier les faiblesses courantes des applications web en combinant exploration automatisée, découverte des points d'entrée et tests ciblés par charges utiles contrôlées. Il repose sur un modèle basé en sous-commande (test et crawl), chacune non cumulable à l'éxecution et correspondant à un domaine d'analyse spécifique. Il s'appuie sur une URL réseau cible unique et accessible fourni en paramètre. En plus des fonctionnalités générales propres à tous les outils, il possède des options spécifiques permettant d'affiner le comportement de chaque sous-commande. Lorsque l'url n'est pas spécifiée ou est inaccessible, WebScout affiche l'aide.

La sous-commande `test` effectue une analyse de sécurité applicative à l'aide de charges utiles ciblées et d'heuristiques pour détecter les vulnérabilités web les plus courantes. Par défaut, une évaluation de base est réalisée.

`--sql` teste la présence de vulnérabilités de type injection SQL, détecte des entrées insuffisamment protégées face aux manipulations de requêtes SQL.

`--xss` recherche des vulnérabilités XSS réfléchies, identifie des points d'injection permettant l'exécution de scripts côté client.

`--csrf` vérifie l'absence ou l'insuffisance de protections CSRF, permet de détecter des actions sensibles non protégées contre les requêtes forgées.

`--dirlist` recherche des expositions de type listing de répertoires, identifie des configurations serveur révélant la structure interne de l'application.

`--fuzz` effectue du fuzzing sur les paramètres identifiés, étend la détection en testant des entrées inattendues ou malformées.
La sous-commande `crawl` explore automatiquement l'application afin d'en cartographier la structure, les chemins de navigation et les points d'entrée accessibles. Par défaut, une exploration automatique de l'application afin d'identifier les pages accessibles et la structure de navigation est effectué.

`--depth <n>` définit la profondeur maximale d'exploration, contrôle l'étendue du crawl pour éviter une exploration excessive.

`--forms` inclut la découverte et l'analyse des formulaires HTML, identifie les points de saisie utilisateur exploitables lors des tests de sécurité.

`--links` limite l'exploration à l'énumération des liens hypertextes, permet un crawl plus rapide et ciblé.

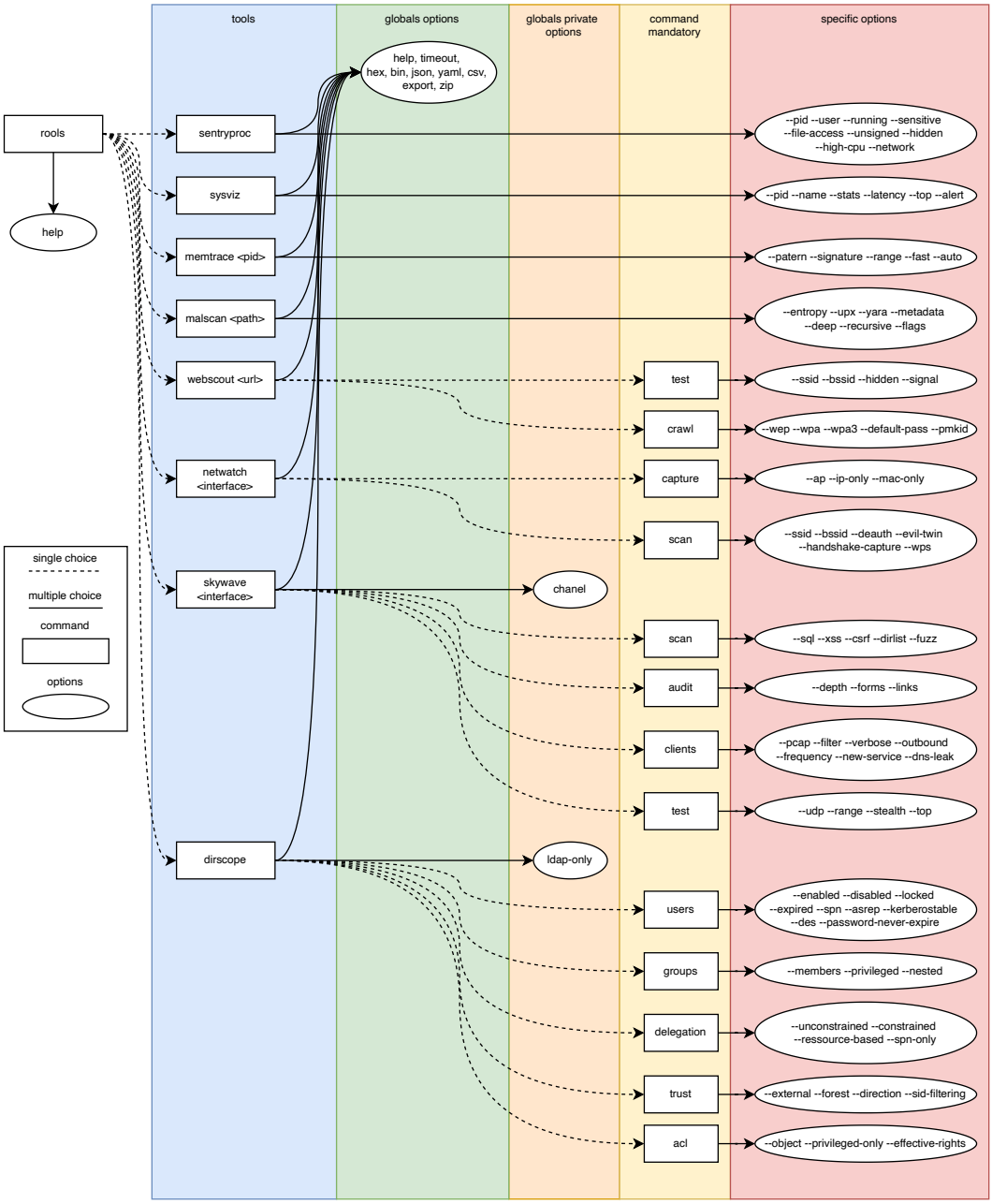

Fig. 1. Architecture fonctionnelle

III. Technical Specifications

A. Technologies

The IEEEtran class file is used to format your paper and style the text.

B. Software architecture

The IEEEtran class file is used to format your paper and style the text.

C. Data model

The IEEEtran class file is used to format your paper and style the text.

D. Dependencies

The IEEEtran class file is used to format your paper and style the text.

E. Security

The IEEEtran class file is used to format your paper and style the text.

F. Expected performance

The IEEEtran class file is used to format your paper and style the text.

IV. Organizational Specifications

A. Methodology

The IEEEtran class file is used to format your paper and style the text.

B. Project schedule

The IEEEtran class file is used to format your paper and style the text.

C. Role distribution

The IEEEtran class file is used to format your paper and style the text.

D. Required resources

The IEEEtran class file is used to format your paper and style the text.

E. Validation criteria

The IEEEtran class file is used to format your paper and style the text.

References