Détection efficace des intrusions sous contraintes de réseaux d'entreprise modernes

par

Viktor NIKULSHIN

MÉMOIRE PRÉSENTÉ À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
COMME EXIGENCE PARTIELLE À L'OBTENTION DE LA MAÎTRISE
AVEC MÉMOIRE EN GÉNIE LOGICIEL
M. Sc. A.

MONTRÉAL, LE 28 AVRIL 2025

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

PRÉSENTATION DU JURY

CE MÉMOIRE A ÉTÉ ÉVALUÉ

PAR UN JURY COMPOSÉ DE:

M. Chamseddine Talhi, directeur de mémoire
Département Génie logiciel et des TI à l'École de technologie supérieure

M. Alain April, président du jury
Département Génie logiciel et des TI à l'École de technologie supérieure

M. Abdelouahed Gherbi, membre du jury
Département Génie logiciel et des TI à l'École de technologie supérieure

IL A FAIT L'OBJET D'UNE SOUTENANCE DEVANT JURY ET PUBLIC

LE 23 AVRIL 2025 À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

Détection efficace des intrusions sous contraintes de réseaux d'entreprise modernes

Viktor NIKULSHIN

RÉSUMÉ

Cette recherche porte sur la détection d'intrusion efficace dans le contexte des cyberattaques modernes et problèmes ouverts de recherche. Elle montre comment les caractéristiques importantes des réseaux réels peuvent significativement impacter la performance des systèmes de détection d'intrusion et empêcher le déploiement industriel des solutions proposées dans les articles académiques. Les expériences de cette recherche sont fondées sur le jeu de données OpTC, le plus grand et le plus représentatif de tous les jeux de données disponibles. La discussion porte sur l'importance de ce jeu de données et les raisons de son obscurité. Pour qu'il soit utilisé d'avantage, une analyse approfondie est faite pour révéler la présence de certaines caractéristiques importantes des réseaux réels dans les données. Cela permet de conclure qu'OpTC peut aider à concevoir et à évaluer des systèmes de détection d'intrusion pratiques.

Les contributions incluent une description détaillée du jeu de données, une évaluation de référence et un algorithme déterministe pour convertir la description de la réalité du terrain en un ensemble d'étiquettes pour une utilisation avec les techniques d'apprentissage machine. La publication des étiquettes évite aux recherches futures de répéter ce processus difficile et chronophage. De plus, les résultats montrent qu'il est possible de détecter les attaques dans ce jeu de données avec des méthodes de détection d'anomalie relativement simples. Cela signale le besoin de développer de meilleurs indicateurs pour évaluer la performance d'IDS.

Ces résultats ont servi de base à l'élaboration d'une méthode novatrice de détection qui repose sur une réflexion critique sur la structure des opérations dans un SOC moderne. L'architecture résultante de l'IDS à deux phases est une preuve de concept qu'il est possible d'intégrer les exigences des équipes de sécurité et les contraintes des réseaux dans une architecture IDS efficace. La discussion porte sur la possibilité de réduire l'écart entre les besoins de l'industrie et la recherche académique en travaillant sur les bons problèmes et les bonnes exigences. Néanmoins, même si cette recherche présente des résultats très prometteurs, elle soulève encore plusieurs problèmes ouverts pour la recherche future, incluant la dépendance sur nature confinée des données d'entrainement et la performance de système de bout en bout.

Mots-clés: détection d'intrusion, détection d'anomalie, IDS, APT, équipe rouge, SOC, OpTC

Effective intrusion detection under constraints of modern enterprise networks

Viktor NIKULSHIN

ABSTRACT

In this work, we explore what constitutes an effective intrusion detection of cyberattacks in the context of modern cyberthreats and open research problems. We discuss the important properties of real-world networks and show that they have a significant impact on the performance of intrusion detection systems and prevent the industrial adoption of solutions proposed in academic literature. This research is centred around the OpTC dataset, which is the largest and most realistic among the existing IDS datasets. We discuss the importance of this dataset and the reasons why it remains relatively obscure. In order to make it more accessible to future research, we perform an in-depth analysis and demonstrate how this dataset accurately reflects some of the constraints of real-world networks. We conclude that it can help design and evaluate practical intrusion detection systems.

Our contributions include a description of the dataset, a baseline evaluation, and a deterministic labelling algorithm that converts the ground truth document into a set of labels suitable for conventional machine learning applications. The released labels can be used in future research, saving the need to repeat this difficult and time-consuming process. Additionally, we show that the attacks in this dataset can be detected using relatively simple anomaly detection methods, which highlights the need for better metrics for evaluating IDS.

Using these results, we propose a detection method based on a critical insight into the structure of operations in a modern SOC. The resulting two-stage IDS framework serves as a proof-of-concept that the requirements of the security teams and important constraints of the real-world enterprise networks can be effectively integrated into the IDS design. This work shows that by focusing on the correct problems and requirements, it is possible to reduce the gap between academic goals and the needs of the industry. However, although the results of this research are very promising, several problems remain open for future research, including reducing dependency on the closed-world assumption of the training data and scaling up end-to-end pipeline performance.

Keywords: intrusion detection, anomaly detection, IDS, APT, red team, SOC, OpTC

TABLE DES MATIÈRES

Page

LISTE DES TABLEAUX

Page

LISTE DES FIGURES

Page

LISTE DES ALGORITHMES

Page

LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES

APT	Menace persistante avancée (*advanced persistent threat*)
CASB	Courtier en accès aux services infonuagiques (*cloud access security broker*)
DARPA	Agence pour les projets de recherche avancée de défense (*Defense Advanced Research Projects Agency*)
EDR	*Endpoint Detection and Response*
LLM	Grand modèle de langage (*large language model*)
IDS	Système de détection d'intrusion (*intrusion detection system*)
IOC	Indicateur de compromission (*indicator of compromise*)
IPS	Système de prévention d'intrusion (*intrusion prevention system*)
NIDS	Systèmes de détection d'intrusion réseau (*network-based IDS*)
NLP	Traitement du langage naturel (*natural language processing*)
NSA	Agence nationale de la sécurité (*National Security Agency*)
OpTC	*Operationally Transparent Cyber Data*
OS	Système d'exploitation (*operating system*)
SIEM	Système de gestion des évènements et des informations de sécurité (*security information and event management*)
SOC	Centre opérationnel de sécurité (*security operations centre*)
TC	Computation transparente (*Transparent Computing*)
UEBA	Analyse comportementale des utilisateurs et des entités (*user and entity behavior analytics*)

LISTE DES SYMBOLES ET UNITÉS DE MESURE

EPS	Événements par seconde
Go	Gigaoctet
Mo	Mégaoctet
To	Téraoctet

INTRODUCTION

La détection d'intrusion se trouve au cœur des systèmes de cybersécurité des entreprises modernes. Cette « dernière ligne de défense » protège contre un acteur malveillant réussissant à contourner toutes les autres mesures de sécurité pour accéder aux ressources privilégiées à l'intérieur du réseau. On peut donc s'attendre à ce que le système de détection d'intrusion (IDS) moderne soit un outil complexe et précis qui évolue constamment pour s'adapter aux nouvelles attaques. Pourtant, la réalité est souvent bien différente. Il arrive fréquemment que les systèmes de détection d'intrusion inondent les équipes de sécurité avec une quantité insurmontable d'alertes de faible qualité ayant un taux de faux positifs trop élevé, ce qui augmente le nombre des tâches manuelles répétitives.

La problématique principale de cette étude est que les IDS novatrices ignorent souvent les contraintes importantes des réseaux d'entreprise et ne sont pas évaluées de façon « bout en bout ». Par conséquent, plusieurs des solutions proposées par la recherche académique n'ont pas réussi à atteindre le déploiement industriel, et la surveillance des réseaux est toujours largement effectuée à l'aide de signatures statiques et d'analyse manuelle des données.

C'est un problème complexe qui demeure non résolu depuis plus de trois décennies, depuis la création des premiers IDS. Les Chapitres 2, 3 et 5 révèlent que certains des facteurs contribuant à ce problème sont les facteurs suivants :

- des hypothèses incorrectes par rapport à l'environnement, telles que la nature confinée de données d'entrainement, l'évaluation de référence inappropriée, l'oubli de fréquence de base, etc. ;
- le manque de jeux de données publiques de haute qualité qui pourraient servir au développement de systèmes pratiques pour le processus de sécurité moderne ;

- négligence des contraintes cruciales de l'environnement informatique complexe et moderne, telles que la perte des données, dérivé conceptuel, quantité des données, etc.

Cette recherche vise à identifier les facteurs favorisant un déploiement réussi d'un IDS au sein d'une entreprise. L'objectif est de découvrir un moyen de minimiser l'impact négatif de la complexité, de générer des alertes pertinentes et actionnables, et, en même temps, d'éviter que le centre opérationnel de sécurité (SOC) soit submergé par de fausses alertes.

Le besoin d'IDS plus puissants souligne la pertinence de ce travail. La fréquence des intrusions dans les réseaux, des extractions de données et des attaques à « jour zéro » continue d'augmenter chaque année, 2024 étant une nouvelle fois un record (CrowdStrike, 2025). Les cybercriminels, aidés par des attaques sophistiquées basées sur l'identité, accélèrent la cadence des déploiements du rançongiciel. De plus, les groupes de menaces persistantes avancées (Advanced Persistent Threat, APT) continuent de compromettre les chaines d'approvisionnement numériques et physiques. Or, les équipes de sécurité ont de la difficulté à suivre le rythme et l'agilité des assaillants (National Cyber Security Centre, 2024a), car elles sont entravées par l'utilisation d'outils obsolètes et de techniques dépassées. Pour rétablir l'équilibre dans ce jeu en faveur de la cyberdéfense, il est nécessaire de disposer de meilleurs instruments capables de gérer efficacement de vastes ensembles de données, tout en identifiant des formes insidieuses, dispersées et novatrices d'attaques.

0.1 Méthodologie

La première étape consiste à identifier le jeu de données le plus semblable à un réseau d'entreprise moderne en tenant compte de sa fraicheur, de sa taille et de la diversité de ses informations. Les systèmes de détection d'intrusions ne peuvent pas être transférés d'un environnement à l'autre. Cependant, si les environnements source et cible présentent des similarités suffisantes

en termes de technologie et de structure, on peut supposer avec raison que l'IDS fonctionne de manière optimale. Pour évaluer cela, il faut explorer la structure, le schéma et les types d'évènements dans le jeu de données. Comprendre la signification de différents évènements dans les journaux permet de tirer des conclusions sur l'infrastructure sous-jacente en termes de taille, de configuration et d'activité, évitant ainsi les suppositions hasardeuses.

Les études académiques sont généralement évaluées en fonction d'un ensemble d'indicateurs standards, telles que la précision et le taux de rappel. Toutefois, même les résultats les plus prometteurs semblent avoir peu d'impact sur une mise en œuvre pratique. Lorsque des experts en sécurité examinent des détections, d'autres facteurs entrent en jeu, comme le taux de fausses alarmes, le temps moyen pour détecter une menace, les couts, etc. Ce n'est pas tant la question des avantages potentiels d'une nouvelle méthode. Les entreprises ne seront pas tentées de l'adopter si elles ne peuvent pas assumer les couts supplémentaires. Cette discussion portera donc sur l'intégration de ces mesures cruciales dans la conception et l'évaluation d'un système de détection d'intrusion efficace. Le but est de démontrer qu'en comprenant les hypothèses formulées par les auteurs des systèmes de détection d'intrusion, les experts en sécurité informatique peuvent adapter les besoins de l'IDS aux contraintes spécifiques de leur réseau. Pour y parvenir, il est nécessaire d'examiner les particularités des architectures réseau actuelles des entreprises.

Cependant, étant donné la sensibilité de ces informations, elles sont rarement partagées, même par les entreprises qui ont vécu de graves incidents de sécurité. Par conséquent, la méthode habituelle pour effectuer des recherches sur les systèmes de détection d'intrusion est d'utiliser un petit nombre de jeux de données artificielles. Au lieu de se fier uniquement à la description des données fournies par leurs auteurs, il est possible d'analyser ces données de la même manière qu'un analyste de sécurité expérimenté le ferait lors de la gestion d'une source de journaux inconnue. Cela devra permettre d'écarter de nombreuses suppositions implicites et d'assurer que l'architecture de l'IDS est adaptée au cas d'utilisation et au modèle de menace.

La recherche se concentre sur la détection d'attaques en employant des méthodes statistiques simples et fondées sur l'apprentissage machine. À ce stade, le but est de confirmer que les attaques peuvent être détectées et de choisir le niveau de granularité de détection approprié, ce qui déterminera l'architecture de l'IDS. L'utilisation de méthodes de référence pour détecter les attaques devrait également éclairer sur les indicateurs de performance présentés dans divers articles. Une technique d'IDS innovante, qui nécessite un paramétrage complexe, mais qui ne peut détecter que des attaques simples sur des ensembles de données de référence, sera peu attractive pour les équipes de sécurité qui travaillent dans les environnements complexes.

Cette compréhension approfondie du jeu de données permet maintenant de chercher dans les données certains écarts qui aident à déterminer si l'environnement simulé est assez réaliste. On porte un intérêt particulier aux traces de dérive conceptuelle, à la perte ou à la corruption d'évènements, ainsi qu'aux activités à haut risque, comme les actions administratives. Comme la revue de littérature le montrera bientôt, ce sont ces caractéristiques des environnements réels qui rendent la détection d'intrusion dans les entreprises particulièrement complexe. Il est donc impératif de mettre en évidence ces propriétés des données et de les rendre explicites dans la conception d'un IDS.

Finalement, le développement d'un système d'analyse de données de bout en bout permet d'étudier comment ces découvertes peuvent influencer la qualité de la détection d'intrusion. L'objectif est de démontrer que l'utilisation des propriétés pertinentes des données permet d'obtenir de bonnes performances en termes de temps d'exécution. Cela permettra aux spécialistes de la cybersécurité de configurer l'IDS en fonction des exigences spécifiques de leur réseau, et de minimiser les effets négatifs de la complexité réelle.

0.2 Contributions

Les contributions de cette recherche sont les suivantes :

- une analyse comparative de l'approche académique et industrielle à la conception de l'IDS efficace, qui met l'importance à respecter les contraintes réelles des réseaux d'entreprise pour une adoption réussie ;
- une revue des jeux de données les plus importants, suivie d'une discussion sur les avantages du jeu de données *Operationally Transparent Cyber Data* (OpTC) pour une évaluation réaliste des systèmes de détection d'intrusion ;
- une évaluation de référence du jeu de données OpTC, qui manquait jusqu'à présent, afin d'aider à interpréter les indicateurs de performance ;
- un algorithme de recherche exhaustive permettant de transformer la « réalité de terrain » (*ground truth*) en un ensemble d'étiquettes pour une utilisation avec les techniques d'apprentissage machine. L'ensemble d'étiquettes est rendu public sous forme de logiciel libre sur GitHub, dans le but d'aider les chercheurs futurs ;
- une architecture IDS novatrice avec une représentation des données en forme de graphes de provenance qui facilitera l'intégration aux processus du Centre opérationnel de sécurité (SOC) et aidera à régler certains types de goulots d'étranglement dans la gestion des données ;
- une évaluation d'une façon « bout à bout » du prototype de l'IDS en utilisant le jeu de données OpTC dans sa totalité pour démontrer sa capacité à détecter les attaques et à garantir un débit suffisant pour traiter les évènements à l'échelle moderne.

CHAPITRE 1

NOTIONS DE BASE

1.1 Système de détection d'intrusion

Un système de détection d'intrusion (IDS) est tout logiciel conçu pour détecter, prévenir et signaler des activités non autorisées et potentiellement malveillantes sur un système informatique. Il est souvent déployé comme partie d'une stratégie de défense en profondeur pour protéger la confidentialité, l'intégrité et l'authenticité de l'information. Selon le cas, un IDS peut détecter les étapes initiales ou finales d'une attaque.

La nécessité d'un IDS est expliquée par le fait que les systèmes informatiques ne peuvent pas être parfaitement sécures en raison des bogues logiciels, des exploits « jour zéro », des menaces internes, des attaques de la chaine d'approvisionnement et des canaux auxiliaires, etc. Par conséquent, pour détecter une déviation dans le comportement de système, la survenance en temps réel est nécessaire. De certaines façons, l'IDS est analogique au système immunitaire actif dans un corps humain, et il a aussi besoin d'améliorations constantes pour répondre à l'évolution des menaces.

L'identification d'attaques sur les réseaux est un domaine crucial qui bénéficie des méthodes d'analyse des anomalies, de l'intelligence artificielle et de l'apprentissage machine. Au fil des ans, des centaines de publications académiques ont été consacrées à ces sujets. Les rapports de performance exubérants peuvent laisser croire que les systèmes IDS de plus en plus complexes sont déjà en mesure de détecter toutes les attaques et qu'il n'y reste plus rien à discuter. Or, la réalité ne peut être plus différente, car le nombre d'incidents de sécurité, d'intrusions de réseau et de fuites de données atteint de nouveaux records chaque année.

L'objectif d'IDS est de repérer les tentatives d'attaque. Cependant, compte tenu de la complexité et de la structure en couches de l'infrastructure informatique moderne, et de la possibilité d'attaques de survenir à tout niveau en présence d'une vulnérabilité, il n'existe pas de méthode unique pour y parvenir.

1.2 Systèmes de détection d'intrusion hôtes et réseaux

Le classement le plus courant des IDS est fait selon le type de télémétrie qu'ils traitent en entrée.

- Systèmes de détection d'intrusion hôtes, qui surveillent l'activité des processus de chaque machine.
- Systèmes de détection d'intrusion réseau (NIDS), qui surveillent le trafic entre les hôtes sur un réseau, ou arrivant depuis Internet.

Cependant, cette classification simpliste ne saisit pas toute la complexité des systèmes informatiques modernes. La tendance au chiffrement de bout en bout empêche les boites intermédiaires (*middleboxes*) d'inspecter le contenu des paquets et de détecter le trafic réseau malveillant. En même temps, l'adoption de méthodes de déploiement modernes, comme l'infonuagique, le modèle de service SaaS, la conteneurisation, la technologie server-less, a fait diminuer le contrôle de l'administrateur des systèmes et a limité sa visibilité sur les activités de l'hôte. Par conséquent, les opportunités de détecter les attaques dans les couches basses de la pile logicielle sont plus limitées.

Par conséquent, les IDS natif infonuagiques, comme *GuardDuty*, combinent certaines caractéristiques des IDS hôtes et des IDS réseau. Ils vont jusqu'à incorporer une partie de la fonctionnalité associée traditionnellement aux logiciels antivirus. De plus, les entreprises sont incitées à adopter des mesures de sécurité alternatives qui sont déployées, entre autres, sur le poste de travail d'employé pour surveiller les flux de données et les interactions avec les ressources de l'entreprise. Les systèmes d'analyse comportementale des utilisateurs et des entités (UEBA) et les courtiers en accès aux services infonuagiques (CASB) sont deux exemples de ce type de contrôle d'identité.

De manière plus large, on peut considérer comme un IDS tout logiciel capable de détecter et signaler une activité non autorisée, anormale ou malveillante sur un système informatique, généralement sous forme d'alerte. Cela inclut les logiciels antivirus, y compris les versions

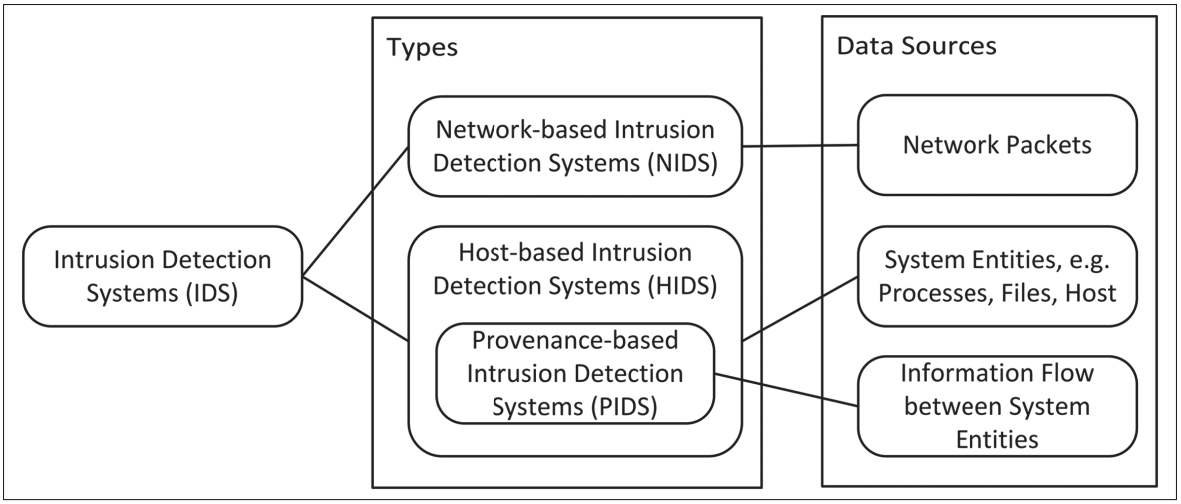

Figure 1.1 La classification des principaux types de systèmes de détection d'intrusion. Tirée de Zipperle *et coll.* (2023)

modernes des systèmes *Endpoint Detection and Response* (EDR), les ensembles de règles de détection par signatures statiques, ainsi que tout type de modèles d'apprentissage automatique entrainés pour détecter des attaques spécifiques.

1.3 Systèmes de détection et prévention d'intrusion

Il est également courant de distinguer les systèmes de détection d'intrusion (IDS) et les systèmes de prévention d'intrusion (IPS), qui peuvent agir de manière proactive pour empêcher les attaques détectées. Le concept original de ce système est né de l'idée qu'une fois une attaque est détectée avec certitude, le système peut automatiquement effectuer les étapes de réparation, sans intervention humaine.

Pour parer à une attaque, un IPS doit se situer « sur le chemin » entre l'attaquant et la cible, ce qui restreint les options architecturales d'un système IPS. Cependant, en matière de capacités de détection des attaques, les IPS peuvent être considérés comme un sous-ensemble des IDS. En effet, selon la publication spéciale du NIST 800-94 (Scarfone et Mell, 2007), un IPS est un système qui possède toutes les fonctionnalités d'un IDS, avec des fonctionnalités supplémentaires pour prévenir les attaques. De ce fait, dans le cadre de cette étude, il est superflu de différencier

ces deux catégories, car l'enjeu de la détection s'applique également aux deux. Cette affirmation est soutenue par le fait qu'un administrateur peut facilement convertir un IPS en un IDS ordinaire lorsque la détection, plutôt que la prévention, s'avère plus appropriée. Cette fonctionnalité est offerte par l'IPS Snort (Cisco, 2024), entre autres.

De plus, l'utilisation de l'IPS est relativement rare dans la pratique. Une étude récente sur la gestion de l'IDS (Vermeer, Kadenko, Van Eeten, Gañán et Parkin, 2023) a mis en évidence que l'interférence avec le trafic d'applications légitimes et ses effets négatifs sur les activités commerciales sont une source de préoccupation majeure lors de l'exploitation d'un IPS. Par conséquent, l'assurance de la qualité des règles de détection nécessite un effort additionnel, ce qui ne fait que renforcer les défis pour le SOC.

1.4 Télémétrie

La télémétrie est un terme générique qui englobe toutes les formes de surveillance à distance et locale du comportement d'un système. En sécurité informatique, elle désigne les données sous diverses formes que les IDS utilisent pour détecter des activités suspectes sur un hôte ou un réseau.

Certains systèmes de détection d'intrusion réseau peuvent fonctionner directement avec des paquets bruts, en analysant les caractéristiques numériques, telles que la taille, la durée ou encore les champs individuels des paquets. Cependant, étant donné que les réseaux modernes sont souvent très surchargés, analyser tous les paquets pourrait être prohibitif. Par conséquent, une méthode populaire pour détecter les intrusions consiste à analyser les données de connexion agrégées sous forme de NetFlow (Cisco, 2006).

D'un autre côté, les IDS hôtes sont généralement conçus pour fonctionner avec les journaux du système (Kent et Souppaya, 2006), qui sont des ensembles d'évènements individuels survenant dans les applications. Le format Syslog (Gerhards, 2009) est souvent utilisé pour ajouter des champs de métadonnées aux entrées de journal, mais, historiquement, les journaux sont écrits sous la forme de déclarations non structurées dans un langage naturel, ce qui les rend difficiles

à traiter pour les machines. Les récentes avancées dans le traitement du langage naturel et le développement de formats de journaux structurés modernes (OpenTelemetry, 2025) peuvent créer de nouvelles opportunités de recherche dans la détection d'intrusions au niveau de l'application.

Le principal défi associé à cette forme de télémétrie est que les journaux sont souvent conçus principalement pour le débogage. Ils ne reflètent donc pas nécessairement des évènements critiques requis pour les enquêtes sur la sécurité. Pour relever ce défi, les systèmes de suivi de provenance utilisent à la place une autre source de télémétrie, qui se produit au niveau très bas du système opérationnel (OS), soit celui des appels système (*syscalls*). Cette méthode soulève également ses propres défis, que cette recherche le montre en détail plus loin.

1.5 IDS basé sur journaux vs systèmes de suivi de provenance

Une autre solution du problème de l'information imparfaite des journaux consiste à retracer toutes les données et tous les chemins d'exécution d'un système informatique sous forme de graphe. Ce graphe peut être utilisé pour développer un système de suivi de provenance.

Dans le domaine de la recherche en IDS, la provenance désigne les interactions entre les objets d'un OS moderne : processus, fichiers, sockets de réseau, adresses IP, identités d'utilisateurs, etc. Ces objets sont généralement représentés sous forme de nœuds dans un graphe, et les arcs capturent le sens du comportement effectué, par exemple, *créer un processus enfant*, *ouvrir un fichier*, *se connecter à une adresse*, etc. Puisque ces opérations sont de niveau très bas et se produisent souvent dans les ordinateurs modernes, les graphes qui en résultent peuvent rapidement atteindre plusieurs centaines de gigaoctets par jour pour un seul hôte occupé (Ma *et coll.*, 2018).

Le concept de provenance a été emprunté à la recherche sur la conception de bases de données (Pan, Stakhanova et Ray, 2023). Il a été appliqué en cybersécurité parce qu'il garantit la complétude et l'immuabilité des données, deux propriétés très désirables pour la détection d'intrusion et pour la chaine de possession numérique. L'hypothèse principale est la suivante : lorsqu'un acteur malveillant exploite un système informatique, ses actions produisent des arcs

suspects, soit atypiques pour l'environnement, soit impossibles (par exemple, dans le cas d'exploits « jour zéro »). Et puisque le graphe conserve l'origine de chaque objet système (fichier, processus, connexion réseau, etc.), il peut modéliser des interactions complexes sur de longues périodes, ce qui est nécessaire pour détecter les attaques APT.

Les IDS qui utilisent la provenance ont attiré beaucoup d'attention dans la recherche académique, ce qui a donné lieu à de nombreuses publications explorant leurs capacités puissantes (Milajerdi, Gjomemo, Eshete, Sekar et Venkatakrishnan, 2019; Han, Pasquier, Bates, Mickens et Seltzer, 2020; Hassan *et coll.*, 2019). Cependant, ces capacités s'accompagnent d'un cout en termes de performance du système, de volumes significatifs de télémétrie et d'algorithmes complexes, ce qui entraine des taux d'adoption relativement faibles dans l'industrie (Dong *et coll.*, 2023). Une étude récente a comparé les journaux et les graphes (Jansen, Bobba et Nevin, 2024), et a révélé que l'écriture de signatures sur des données graphiques entraine un surcout et est moins évidente pour les analystes par rapport aux signatures traditionnelles basées sur les journaux.

Il est important de ne pas percevoir les IDS basés sur des journaux et les IDS basés sur la provenance comme étant des solutions mutuellement exclusives. La collecte de la provenance sur plusieurs hôtes exige de regrouper les données graphiques en un seul endroit, où elles sont sérialisées sous la forme d'un journal, comme c'était le cas pour le jeu de données *Operationally Transparent Cyber Data* (OpTC, FiveDirections (2019)), qui sera utilisé dans les chapitres suivants de ce travail.

1.6 SIEM (Security Information and Event Management)

Les systèmes de gestion des évènements et des informations de sécurité (SIEM) sont le type d'IDS le plus utilisé par les entreprises modernes. Contrairement aux systèmes de suivi de provenance, qui analysent le comportement au niveau d'OS (appels système, connexions réseau, opérations sur fichiers, etc.), et aux IDS réseau, qui examinent le trafic de réseau brut, les systèmes SIEM fonctionnent principalement sur les traces d'exécution de programmes sous forme de journaux, qui peuvent être structurés, semi-structurés ou non structurés (Sharif, 2022).

Le texte est l'interface universelle (Salus, 1994) qui peut décrire n'importe quel changement d'état dans n'importe quel matériel électronique ou dans son logiciel. Du point de vue économique, il est très avantageux de consolider les journaux de tous les systèmes d'un réseau en un seul endroit, puisque le même télémètre peut servir les besoins de la sécurité, du suivi de la performance et du débogage d'application, et ainsi réduire les couts globaux pour l'entreprise.

Les systèmes SIEM les plus populaires sur le marché actuel incluent Splunk, Elasticsearch, Azure Sentinel, Datadog, Snowflake et plusieurs autres solutions spécialisées. Ils offrent la fonctionnalité de recherche de texte intégral et d'agrégation, ce qui permet la création d'alertes à l'aide de règles statiques, ainsi que de modèles d'apprentissage automatique. En pratique, de nombreux systèmes de détection d'intrusion, y compris ceux proposés dans la recherche académique, sont probablement appuyés par un système SIEM et ils utilisent des langages spécifiques du domaine pour effectuer des requêtes et extraire les données nécessaires.

Bien que les systèmes SIEM soient en mesure de corréler des évènements provenant de n'importe quel nombre de systèmes, leur principal inconvénient est qu'ils dépendent de l'information contenue dans les journaux. Par conséquent, si une application ne génère pas de journal pour un évènement spécifique d'intérêt (par exemple, une tentative d'authentification échouée), un SIEM sera incapable de le détecter.

La stratégie de mitigation la plus populaire pour ce problème consiste à utiliser plusieurs systèmes IDS qui se situent à différents niveaux de la pile technologique et qui permettent de combler les angles morts des autres systèmes.

1.7 IDS par signatures vs IDS par anomalies

Un autre moyen courant de catégoriser l'IDS est de le faire en fonction de la méthode utilisée pour générer une alerte (aussi appelée une alarme dans certaines sources) : par signature ou par anomalie.

- Un IDS par signatures repose sur une bibliothèque des signatures de comportements malveillants connus et utilise un moteur de détection pour vérifier si l'état du système ou un artéfact correspond à la description d'une attaque.
- Un IDS par anomalies repose sur des méthodes statistiques ou d'apprentissage machine. Il est conçu pour détecter un changement significatif de l'état du système par rapport à son comportement attendu de base.

Les IDS par signature sont aussi parfois désignés sous le nom de « systèmes de détection d'abus » dans des travaux académiques. Cependant, ce terme est inadéquat, puisque l'abus peut être défini uniquement en fonction de la politique d'usage acceptée selon les exigences de sécurité de chaque entreprise individuelle, indépendamment de la technologie. Cette discussion évitera l'emploi de ce terme pour distinguer deux types d'IDS en fonction de leurs objectifs seulement. Les IDS par signature sont donc pour détecter un *état malveillant connu* du système, tandis que les IDS par anomalies sont pour détecter des *états malveillants inconnus*.

Une distinction importante de ce travail par rapport à la tradition établie en recherche académique sur l'IDS est que, selon la définition ci-haut, un modèle d'apprentissage machine peut toujours être classé comme un IDS par signature s'il est conçu pour détecter un ensemble d'états malveillants connus du système. Cela permet de mieux viser la possibilité d'interprétation des résultats, ce qui est la condition la plus importante pour une mise en production réussie, comme discuté plus loin.

1.8 Alertes

L'alerte est un concept central de la recherche sur la détection d'intrusion. Il est étonnant de constater à quel point il est difficile de définir précisément ce qu'est une alerte. L'usage même de ce terme présente des incohérences.

Dans le domaine de la recherche académique, les termes alerte et alarme sont souvent utilisés de manière interchangeable. Dans l'industrie, la définition d'une alerte peut changer

considérablement d'une entreprise à l'autre, en fonction des procédures de sécurité en place. D'autres termes courants sont « observable », « notable », « détection », « incident », etc. (MITRE, 2024; Splunk, 2024). Un IDS peut utiliser tous ces termes pour indiquer une activité suspecte sur un hôte ou un réseau. Cependant, en raison des différences significatives dans le fonctionnement des systèmes de détection d'intrusion, les résultats qu'ils produisent peuvent varier considérablement.

Le cas le plus simple d'une alerte est la détection d'un paquet de réseau correspondant à une activité malveillante connue, dont on connait la signature. En revanche, une alerte IDS avancée pourrait être le score de risque interne généré par un système UEBA qui prend en compte les modèles d'accès typiques à des documents internes privilégiés pour chaque employé.

Dans tous les cas, les alertes générées sont destinées aux analystes de sécurité dans le SOC. Leur travail consiste à déterminer si l'alerte représente une véritable menace ou un faux positif. Par conséquent, une alerte peut être définie comme un signal émis par un IDS pour prévenir un expert humain d'une attaque en cours.

Cependant, avec une telle définition floue, certaines signatures deviennent également floues, ce qui peut générer un grand nombre d'alertes en peu de temps. Le résultat est souvent la fatigue des alertes, qui survient lorsque l'analyste est submergé par un grand nombre de fausses alertes et qu'il n'a plus le temps ni les ressources nécessaires pour examiner chacune d'entre elles en détail. La fatigue des opérateurs est une préoccupation majeure pour tous les domaines ayant de hautes exigences en sureté, et la cybersécurité en fait partie. Les véritables attaques qui étaient passées inaperçues en raison de la fatigue des alertes ont déjà entrainé des incidents majeurs (United States Senate Committee on Commerce, Science, and Transportation, 2014).

En conséquence, les entreprises ont consacré des ressources considérables à l'amélioration de la fiabilité des alertes IDS. Dans le milieu universitaire, on compte un nombre croissant de publications sur le thème de la réduction des fausses alertes, ainsi que sur le regroupement, l'agrégation et l'élimination des doublons d'alertes (Hassan *et coll.*, 2019; Landauer, Skopik, Wurzenberger et Rauber, 2022; Liu, Shu, Sun, Jang et Mittal, 2022b; Freitas et Gharib, 2024).

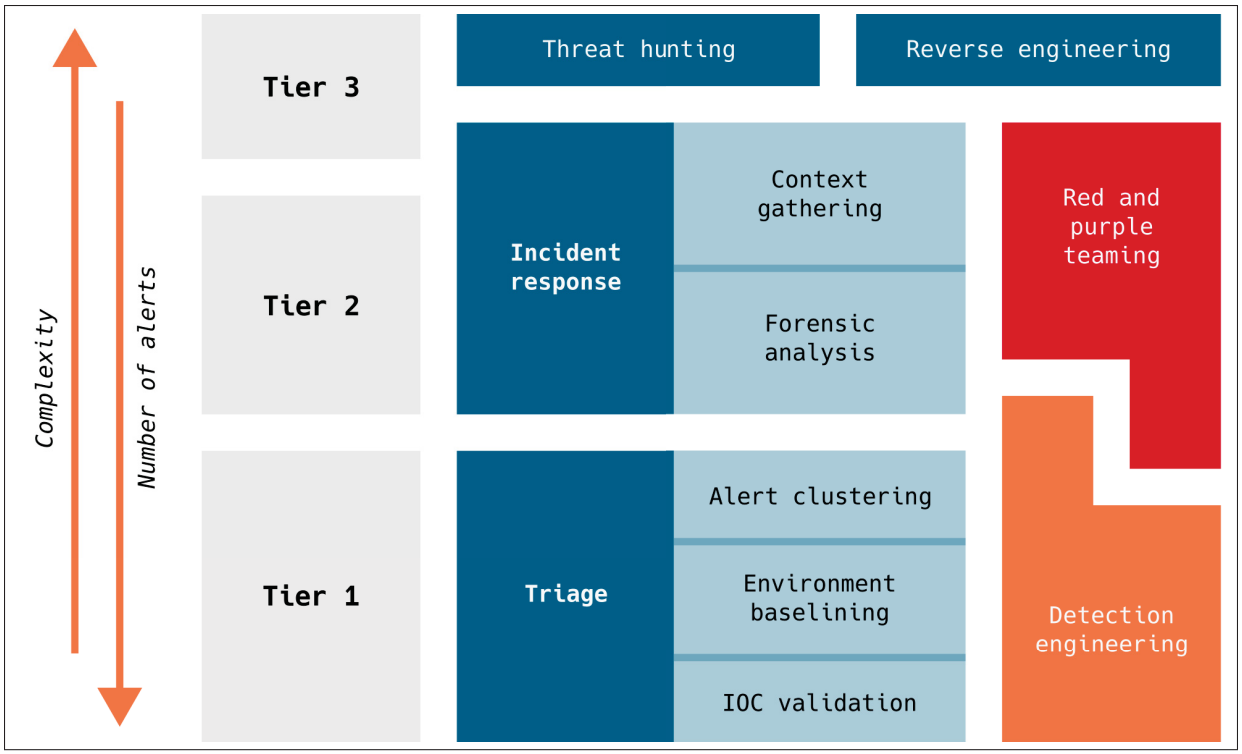

Figure 1.2 Une structure hiérarchique représentant la séparation des tâches dans un SOC typique

1.9 Centre opérationnel de sécurité

Un centre opérationnel de sécurité, ou un SOC, est conçu pour combiner « des personnes [1], des processus et des technologies en vue d'améliorer la résilience de votre organisation contre les cybermenaces » (Canadian Centre for Cyber Security, 2023a).

Ce contrôle de sécurité est relativement récent, mais il est largement adopté dans l'industrie (Crowley et Pescatore, 2019; CrowdStrike, 2022). Il est imposé par plusieurs organismes de normalisation (Joint Task Force Interagency Working Group, 2020; Gaudin *et coll.*, 2018), mais ces normes ne définissent pas l'architecture exacte. Les organisations sont libres de s'adapter au SOC en fonction de leurs besoins en cybersécurité. Cependant, la plupart des déploiements tendent à être organisés autour de l'architecture en couches.

[1] À noter que cette définition met l'accent sur les personnes avant tout, et sur la technologie en dernier lieu.

En analysant la structure typique d'un SOC moderne, on observe qu'il se compose généralement de deux ou trois niveaux séparés qui effectuent des tâches très différentes (Knerler, Parker et Zimmerman, 2022; Palo Alto Networks, 2024). La Couche 1 est appelée « triage », tandis que la Couche 2 est consacrée à la réponse à des incidents. Dans les organisations plus grandes, on peut également trouver un groupe de Couche 3 dédié aux activités de chasse aux menaces. Cependant, selon l'organisation, il peut être une unité distincte ou être complètement externe (Maxam et Davis, 2024). En revanche, la séparation entre la Couche 1 et la Couche 2 est presque universelle dans toutes les industries.

La Figure 1.2 résume les responsabilités typiques des différentes couches. Cette méthode est largement utilisée, ce qui montre que la détection d'intrusion se compose de plusieurs étapes distinctes : la détection initiale d'évènements suspects et l'enquête subséquente (c'est-à-dire l'analyse). Même si ces deux activités utilisent les mêmes données et les alertes d'IDS, elles ont des objectifs et des résultats différents.

CHAPITRE 2

L'ÉTAT DE L'ART DE L'IDS

Bien que l'amélioration des outils soit manifestement nécessaire, les IDS industriels ont connu très peu d'évolution au cours des dernières décennies et s'appuient largement sur des signatures statiques et d'analyses manuelles des données. L'introduction de techniques d'apprentissage automatique est confrontée à la résistance de l'équipe de sécurité en raison de plusieurs défis uniques dans le domaine de la cybersécurité. À partir d'une perspective académique, l'un de ces défis est le manque de bases de données de haute qualité publiques qui tiennent compte de contraintes importantes dans le processus de réponse aux incidents modernes.

Ce chapitre s'appuie sur une analyse comparative des conceptions d'IDS qui ont évolué des côtés académiques et industriels. Il discute des principaux défis auxquels chaque perspective est confronté et des solutions développées.

2.1 Revue de littérature

Dans de nombreuses recherches académiques, les systèmes de détection d'intrusion sont généralement divisés en deux catégories simples : les systèmes d'alerte par signature (aussi connus sous le nom de détection de mésusage) et la détection par anomalie. La perception de l'utilité d'un IDS diffère considérablement entre l'académie et l'industrie. Cette différence est due au fait que plusieurs études traitent ce sujet de manière isolée, en le sortant complètement du contexte plus large de sécurité et de modélisation des menaces.

2.1.1 Chronologie des systèmes de détection d'intrusion

L'idée de concevoir la détection d'intrusion comme une détection d'anomalies remonte à l'ouvrage fondateur de Denning (Denning, 1987), qui repose sur deux hypothèses de base :

1. Une activité malveillante entraine un comportement système anormal ;
2. Il est possible d'observer de manière significative un comportement anormal.

L'un des principaux avantages de cette approche est qu'elle a permis une application directe des techniques de détection d'anomalies développées dans des domaines connexes au fil des décennies suivantes, commençant par les statistiques de base, puis l'apprentissage machine, puis l'apprentissage profond et les réseaux neuronaux graphiques, et enfin les grand modèle de langage (LLM).

Les présupposés de ce modèle simpliste sont ancrés dans la technoréalité de l'époque. Les ressources informatiques ne permettaient d'exécuter que quelques tâches sans parallélisme, ce qui permettait de décrire le comportement d'un système à l'aide d'un vocabulaire limité des relations entre ses composants. Toutefois, lorsque l'article fut publié, plusieurs concepts et abstractions, qu'on retrouve encore aujourd'hui dans les ordinateurs modernes, étaient déjà bien établis. On reconnait aussi le piratage informatique comme un problème majeur. Les années 1980 ont également vu l'émergence des premiers virus malveillants (Federal Bureau of Investigation, 2019), ainsi que des premiers logiciels antivirus pour y faire face. Néanmoins, il existe certaines contradictions quant aux évènements historiques liés à cette période (Emm, 2008).

Ces idées ont rapidement germé, donnant naissance à plusieurs systèmes de détection d'intrusion, tels que Haystack (Smaha, 1988) et MIDAS (Sebring, Shellhouse, Hanna et Whitehurst, 1988), qui ont continué à se développer au fur et à mesure. On observe déjà, à partir de ces premiers systèmes, la séparation des méthodes de détection par signature et celles par anomalie.

Dans les années 1990, le domaine de la détection d'intrusion avait déjà évolué vers sa forme actuelle. Les IDS hôtes et les IDS réseau se sont séparés en deux groupes distincts (Paxson, 1999), allors que les méthodes de détection par signatures ont gagné en popularité. Des initiatives étaient en cours pour améliorer la qualité de la détection, notamment *Common Intrusion Detection Framework* (Cadre commun de détection d'intrusions) (STANIFORD-CHEN, 1998), un projet mené en partenariat avec l'Agence pour les projets de recherche avancée de défense (DARPA). Au même moment, DARPA a également contribué aux premiers jeux de données disponibles au public (DARPA, 1998), qui ont servi de base à une grande quantité de recherches sur l'IDS. Ils ont été suivis par le jeu de données bien connu KDD (UC Irvine, 1999).

Pendant près de vingt ans, ces deux jeux de données et leurs dérivés ont servi de base à la recherche en IDS. Bien qu'ils aient fait l'objet de critiques sévères en raison de leur simplicité (McHugh, 2000; Creech et Hu, 2013), il n'y avait pas d'alternatives véritables pour les chercheurs. À ce jour, ils sont encore parfois utilisés dans les travaux récents, principalement en raison de la tradition (Ahmad, Shahid Khan, Wai Shiang, Abdullah et Ahmad, 2021). Ce n'est qu'en 2017 que ce fossé a enfin été comblé grâce à la publication de jeux de données de haute qualité par l'Institut canadien sur la cybersécurité, conçus pour représenter les environnements modernes. Parmi eux se trouvait CIC-IDS2017 (Sharafaldin, Habibi Lashkari et Ghorbani, 2018), qui est devenu peu à peu le jeu de données IDS le plus populaire. Le Chapitre 3 examinera en détail ces ensembles de données.

La fin du millénaire a été marquée par l'émergence de méthodes d'évasion permettant de contourner les IDS réseau (Handley, Paxson et Kreibich, 2001) et les IDS hôtes (Wagner et Soto, 2002). Cette situation a suscité des discussions sur la manière de mesurer la qualité de la détection et de comparer différentes approches (Ptacek et Newsham, 1998). C'est à peu près au même moment que certains articles influents ont commencé à aborder l'échec de plusieurs idées de livrer une solution pratique, qui s'était déjà manifesté comme un défi. Le problème de l'oubli de la fréquence de base (Axelsson, 1999) et la nature confinée des environnements de laboratoire (Sommer et Paxson, 2010) ont été identifiés comme les limites fondamentales de cette approche. Cela a soulevé des questions quant à l'utilité des méthodes de détection d'anomalies dans le domaine de l'IDS (Gates et Taylor, 2006).

Malgré cela, les recherches ont continué à explorer des méthodes de détection innovantes et à tester des modalités de détection inédites. Certaines études se sont appuyées sur des ensembles de syscall, ouvrant ainsi la voie à des méthodes de suivi de la provenance (Forrest, Hofmeyr, Somayaji et Longstaff, 1996; Hofmeyr, Forrest et Somayaji, 1998; Warrender, Forrest et Pearlmutter, 1999). Ces dernières ont vite été améliorées pour permettre d'analyser les chaines causales en utilisant les graphes de dépendance. Les travaux comme ceux de King et Chen (2003) et King, Mao, Lucchetti et Chen (2005) se sont appuyés sur des concepts d'audit de système qui remontent à plusieurs années (Anderson, 1980). On a rapidement constaté que

l'analyse des systèmes à des niveaux très profonds entrainait des quantités colossales de données. Par conséquent, ces recherches se sont penchées non seulement sur la conception et l'analyse des traces d'audit, mais aussi sur l'optimisation, la réduction et le stockage des graphes de dépendance (Lee, Zhang et Xu, 2013b; Bates, Tian, Butler et Moyer, 2015; Pasquier *et coll.*, 2017; Ma *et coll.*, 2018; Inam *et coll.*, 2023).

L'intérêt de la communauté scientifique pour la détection d'intrusions a toujours connu une hausse après des incidents de sécurité majeurs, qui sont devenus plus fréquents dans les années 2010. Par exemple, l'intrusion dans le réseau de *Target* (United States Senate Committee on Commerce, Science, and Transportation, 2014) a révélé la structure en plusieurs étapes des attaques complexes, ce qui a incité à des études sur la corrélation d'alertes et l'analyse des dépendances pour combattre la longue durée de ces types d'attaques (Noel, Harley, Tam, Limiero et Share, 2016; Liu *et coll.*, 2018).

La faille *d'Equifax*, souvent considérée comme l'un des incidents de cybersécurité les plus graves, a eu des effets similaires. Cependant, dans ce cas, la réussite de cette attaque ne résulte pas d'une incapacité de l'IDS à repérer l'intrusion (Oversight and Government Reform, 2018). Elle est plutôt arrivée en raison de la faiblesse du processus et des intégrations complexes entre les différents outils de sécurité qui existent dans un SOC. Cela démontre que la détection d'intrusions ne peut pas être considérée comme un problème isolé, mais qu'elle doit être conçue pour soutenir les processus de sécurité de bout en bout.

Enfin, l'attaque *SolarWinds* (FireEye, 2020) a mis en évidence la complexité des chaines d'approvisionnement, ce qui a entrainé une hausse du nombre de publications portant sur les attaques APT. Cette tendance a été particulièrement bénéfique pour la recherche sur le suivi de provenance, puisque l'attaque a démontré que les méthodes traditionnelles, comme la détection de maliciel ou l'analyse de la réputation des adresses IP, ne sont pas très efficaces pour détecter les attaques originales de composants logiciels fiables (Anjum, Iqbal et Hamelin, 2022; Bowman, 2022; Zipperle *et coll.*, 2023).

Une autre conséquence de ces intrusions à fort impact est la diffusion, dans l'industrie puis dans le milieu universitaire, du concept de menaces persistantes avancées (APT), qui a été initialement élaboré par la communauté du renseignement militaire. Cela a conduit à la création de programmes de recherche spécifiques pour leur détection.

2.1.2 Systèmes de suivi de provenance actuels

Un tel programme a été réalisé par la DARPA sous le nom de *Transparent Computing* (TC) (Griffith *et coll.*, 2020). Parmi ses objectifs, il y avait celui de simuler de manière fiable un environnement corporatif typique afin de représenter les attaques modernes et d'aider à leur détection. L'intérêt pour le suivi de provenance a connu une hausse au moment où plusieurs jeux de données créés par ce programme sont devenus disponibles, notamment les jeux de données *TC E3*, *TC E5* et *OpTC*.

La plupart des études récentes sur le suivi de la provenance utilisent ces jeux de données d'une manière ou d'une autre. Le jeu OpTC est particulièrement important dans le cadre de cette recherche, car il constitue la plus grande et la plus fiable des simulations parmi les jeux de données actuels. Le Chapitre 3 et le Chapitre 4 font une analyse approfondie de ce jeu. La discussion ci-dessous porte sur les travaux importants qui l'utilisent.

Le premier travail académique à se concentrer sur le jeu de données OpTC a été réalisé par le Conseil national de recherches du Canada (Anjum, Iqbal et Hamelin, 2021). L'incident de SolarWinds (FireEye, 2020) a été un facteur important dans la publication de cette étude, car il a mis en évidence les défauts des systèmes de cyberdéfense actuels face aux attaques APT sophistiquées. Les auteurs de cet article présentent une vue d'ensemble de la structure de ce jeu de données, des types d'évènements clés et des caractéristiques associées. Cet article demeure une source de documentation importante. Les chapitres suivants s'appuient sur les résultats de cet article pour fournir des informations supplémentaires.

Il a fallu environ deux ans après la publication d'OpTC pour qu'on observe une augmentation significative de son utilisation, souvent dans le cadre de programmes de recherche directement

financés par DARPA. En 2021, les chercheurs de la Agence nationale de la sécurité (NSA) ont développé un système complet de détection d'intrusions en utilisant des méthodes de traitement du langage naturel (NLP) (Golczynski et Emanuello, 2021). Ils se sont penchés sur la manière dont l'ingénierie de caractéristiques peut aider à détecter de nouvelles attaques perpétrées par des adversaires possédant des compétences inconnues. Les auteurs ont partagé plusieurs préoccupations pratiques sur le déploiement industriel des IDS, qui font aussi l'objet de la discussion actuelle.

Les auteurs ont opté pour une stratégie d'étiquetage qui met l'accent sur la quantité de preuves présentées à un analyste plutôt que sur la précision des étiquettes. Autrement dit, le plus grand nombre d'évènements peuvent être classés à tort comme étant malveillants. Ils avancent un argument selon lequel les mesures d'évaluation courantes pour les modèles d'apprentissage machine sont inefficaces en présence des nouvelles attaques, ce qui constitue un défi majeur non résolu. L'évaluation des résultats est basée sur les données de seulement quatre hôtes.

Dans SK-Tree (Cochrane *et coll.*, 2021), les auteurs ont adapté une technique plus ancienne de détection de maliciel à la structure du jeu de données OpTC. Contrairement à d'autres études sur les maliciels qui se concentrent sur le comportement durant l'exécution ou sur les caractéristiques statiques des fichiers exécutables, ce travail se concentre sur l'information codée dans l'arbre des processus. L'objectif est de détecter un évènement de création d'un processus suspect, ce qui relève de la recherche sur la détection d'intrusion. L'évaluation se fonde sur les données d'un hôte connu pour être malveillant. La classification est effectuée à l'aide d'un algorithme de machine à vecteurs de support (SVM), qui nécessite des étiquettes. Cependant, les auteurs ne détaillent pas leur méthode d'étiquetage des données.

Dans l'article court de LumenAI (Medad, Gregorutti, Genetay et Nguema, 2021), les auteurs proposent l'utilisation d'un algorithme MCMC Louvain pour détecter en temps réel un changement soudain dans le nombre de communautés dans un graphe composé de tous les processus exécutés sur une machine. OpTC est suggéré comme un ensemble de données pratique pour la recherche en IDS, mais son potentiel complet n'est pas pleinement exploité

dans l'article. L'évaluation se fait sur une journée entière de données d'un hôte présentant une activité malveillante confirmée, ce qui élimine le besoin de générer des étiquettes.

DeepTaskAPT (Mamun et Shi, 2021) est l'une des rares publications à considérer les menaces internes et les attaques par APT comme étant les deux faces d'une même pièce. Les auteurs ont entrainé un modèle LSTM (Long Short-Term Memory) sur l'activité bénigne d'un seul utilisateur, ce qui leur a permis d'éviter l'étiquetage couteux. Toutefois, cette approche présente un risque : le modèle pourrait se souvenir de tout comportement malveillant présent dans les données d'entrainement, un problème récurrent dans ce champ de recherche.

La phase d'évaluation s'appuie sur OpTC et LANL 2017, mais la phase d'apprentissage n'utilise qu'un petit sous-ensemble des données d'un seul utilisateur en raison de la charge de calcul excessive.

L'un des principaux résultats de cet article est qu'un modèle LSTM entrainé à partir des activités bénignes d'un seul utilisateur peut prédire le comportement d'autres utilisateurs. Cependant, plus loin dans ce document, il est démontré que ce résultat est probablement dû à la fuite de données entre les étapes d'entrainement plutôt qu'à la propreté du modèle.

OpTC était l'un des ensembles de données utilisés pour évaluer l'outil de détection d'anomalies graphiques NetHawk, proposé par Bowman B. dans sa thèse doctorale (Bowman, 2022). NetHawk est un outil puissant de détection d'anomalies dans les graphes qui automatise la chasse aux menaces (threat hunting) par évènements rares. C'est une stratégie couramment utilisée par les analystes de sécurité dans la vie réelle. Dans sa thèse, l'auteur examine plusieurs problématiques liées à l'implémentation d'IDS dans un contexte industriel. L'un des buts spécifiques de l'utilisation du jeu de données OpTC est de mesurer le temps de réponse des IDS face à un volume réaliste de la télémétrie.

Les étiquettes pour cette expérience ont été créées par un expert en cybersécurité à partir du document de réalité de terrain. Cependant, l'expert a également inclus des évènements

supplémentaires, même s'ils n'étaient pas mentionnés dans les journaux de l'équipe rouge. Cette décision rend évidemment plus difficile la reproduction des résultats.

ANUBIS (Anjum *et coll.*, 2022) est une prochaine génération de systèmes de suivi de provenance. L'apport principal de cet article est l'explicabilité du résultat du modèle. Les auteurs utilisent un réseau neuronal bayésien capable d'attribuer une note de confiance à une prédiction, plutôt que de simplement lui attribuer une étiquette. Cette fonctionnalité permet aux analystes en sécurité de prendre des décisions plus informées quant à la probabilité qu'une prédiction soit une fausse alarme. Ce design vise à lutter contre la fatigue due aux alertes. Ce phénomène est fréquent dans les grands SOC, qui sont submergés par de fausses alertes générées par différents systèmes de détection d'intrusion, entravant ainsi la capacité des analystes à identifier les véritables évènements pertinents. La Section 2.2 porte sur ce problème. .

Les réseaux neuronaux bayésiens sont des algorithmes d'apprentissage supervisé qui requièrent des étiquettes et une période d'entrainement. Les auteurs n'ont pas expliqué comment ils ont obtenu ces étiquettes. Ils ont choisi OpTC pour l'évaluation, car il représente fidèlement les attaques par APT, mais ils soulignent la « monoculture » du réseau, ce qui rend plus difficile l'application des résultats expérimentaux à un scénario plus réaliste.

Kairos (Cheng *et coll.*, 2024) est un système de suivi de provenance complet qui répond simultanément à plusieurs propriétés de détection souhaitables, telles que la détection de nouvelles attaques, un bon rendement en temps réel et l'explicabilité du résultat. Il permet de reconstituer la cause racine d'un incident sous forme d'un graphe minimisé pour une analyse manuelle. La détection se fait au niveau des arcs du graphe, qui représentent les évènements bruts, et la méthode d'analyse de graphe proposée prend en compte simultanément les indicateurs structurels et chronologiques, ce qui permet une étude dynamique des attaques.

Il utilise l'architecture « encoder-decoder » pour obtenir l'erreur de reconstruction de chaque nouveau lien ajouté au graphe, en la comparant à l'état historique des nœuds environnants. Des alarmes se déclenchent lorsque l'erreur cumulative dépasse un certain seuil après une période prédéterminée. Malheureusement, cela signifie que tous les nouveaux processus exécutés sur les

hôtes vont générer de fausses alertes. En d'autres termes, cette méthode est très susceptible au problème de dérive conceptuelle, comme le reconnaissent les auteurs eux-mêmes.

Les auteurs utilisent les jeux de données TC E5 et OpTC pour l'évaluation. Mais dans le cas de OpTC, ils n'utilisent qu'un petit nombre d'hôtes et réduisent la taille du graphe avec plusieurs euristiques courantes. Bien que le système soit capable de détecter de nouvelles attaques sans connaissances préalables, il exige un prétraitement majeur des évènements, y compris leur stockage dans une base de données relationnelle pour accélérer les recherches. Le Chapitre 5 portera sur les conséquences de cette approche pour la mise en œuvre opérationnelle d'un IDS.

Flash (Ur Rehman, Ahmadi et Ul Hassan, 2024) est un système de détection en deux étapes composé d'un réseau neuronal graphique et d'une méthode rapide de plongement d'évènements qui accélère l'analyse de grands graphes de provenance. Les données sémantiques riches contenues dans les journaux d'audit lui permettent d'identifier des nœuds similaires du graphe, ce qui en suite permet à réutiliser les plongements précalculés et à réduire le temps d'exécution total.

Le système utilise des attributs sémantiques, tels que les chemins de dossiers, les noms de fichiers et les adresses IP, pour apprendre le comportement de base du système dans le jeu de données OpTC et détecter avec précision les attaques de l'équipe rouge. Cependant, comme sera discuté dans l'analyse approfondie de cet ensemble de données au Chapitre 4, cette bonne performance ne peut pas être attribuée à l'efficacité de la méthode, mais plutôt à de fausses corrélations dues aux particularités uniques de cet ensemble de données. Les artéfacts utilisés dans les attaques n'avaient pas été camouflés et conservaient des noms malveillants très évidents.

Bien que la méthode proposée réduise considérablement les exigences computationnelles, elle dépend d'un comportement système très stable, et détectera toute nouvelle activité pendant la période d'évaluation comme une attaque. Par conséquent, cette méthode est également susceptible au problème de la dérive conceptuelle. Flash est conçu pour fonctionner comme un IDS hôte, ce qui lui permet de fonctionner avec des données provenant d'un seul hôte. De plus,

l'évaluation de la performance a aussi été effectuée avec un sous-ensemble du jeu de données OpTC.

2.1.3 Intégration des connaissances du domaine de la cybersécurité

À l'exception de ce dernier exemple, la recherche moderne sur les graphes de provenance a tendance à ignorer complètement l'information sémantique contenue dans les journaux d'audit et à détecter les attaques uniquement par la structure du graphe, à l'aide de diverses techniques empruntées au domaine de la détection d'anomalies graphiques. Une stratégie courante consiste à utiliser l'un des multiples jeux de données de référence, à mapper ses entités à des nœuds du graphe, puis à transformer tous les évènements originaux du jeu de données en arcs. La conséquence principale de cette approche est que le graphe de provenance devient très grand et très difficile à gérer. Une statistique souvent citée est qu'un seul hôte peut générer jusqu'à 33 Go de journaux d'audit par jour (Inam *et coll.*, 2023), ce qui constitue une estimation plutôt conservatrice. En multipliant ce chiffre par le nombre d'hôtes, la taille du graphe résultant dépasse la puissance des ordinateurs modernes.

L'objectif de la conception d'un IDS devient ensuite de représenter le graphe de manière concise tout en garantissant un bon niveau de détection. Cette tâche peut être accomplie en utilisant des « esquisses » du graphe (Han *et coll.*, 2020), en éliminant les arcs en double (Jia *et coll.*, 2024), ou en divisant le graphe en plusieurs sections plus petites (Lee *et coll.*, 2013b; Goyal, Wang et Bates, 2024). Cependant, la plupart des travaux simplement utilisent une petite partie du jeu de données pour l'évaluation (Golczynski et Emanuello, 2021; Anjum *et coll.*, 2022; Cheng *et coll.*, 2024; Ur Rehman *et coll.*, 2024).

Cette pratique est probablement influencée par d'autres champs de l'apprentissage profond, tels que le traitement du langage naturel et la vision par ordinateur, qui ont tendance à traiter les données d'entrée comme une simple matrice numérique, en supposant que le modèle découvrira automatiquement le sens caché. Cependant, l'entrée pour un IDS est différente d'une coordonnée de couleur (X, Y). Les journaux de l'exécution de système représentent les interactions entre

des classes d'entités entièrement distinctes : les humains, les ordinateurs, les programmes, les réseaux, etc. Cela a des implications importantes pour la cybersécurité, pouvant même empêcher l'adoption de tels systèmes dans une entreprise : le processus de remédiation dans un SOC doit être complètement différent si le comportement malicieux est attribué à un hôte ou à un employé.

Les journaux peuvent être représentés par un automate fini qui permet d'approcher le comportement d'un programme (Tan, Pan, Kavulya, Gandhi et Narasimhan, 2008). Les journaux ne sont pas des « artéfacts naturels » d'une exécution de programme, puisqu'ils sont placés stratégiquement dans le code source. Pour cette raison, ils contiennent une quantité significative d'information sémantique. Par contre, une séquence de journaux est le résultat de la combinaison de plusieurs automates finis distincts. Cela est démontré par la différence entre les diagrammes d'état d'un cycle de vie de service Windows (Microsoft, 2018) et d'une connexion TCP établie par un tel processus (Eddy, 2022). Alors que les réseaux neuronaux peuvent approximer tout automate fini (Omlin et Giles, 1996), le résultat n'est pas garanti d'être efficace ou interprétable. Donc, si le composant sémantique n'est pas considéré et que le graphe est analysé de façon homogène, l'interprétabilité du résultat sera affectée, c'est ce qui est souvent le cas dans les systèmes de suivi de provenance modernes.

Cette négligence envers les attributs sémantiques est, en fait, un symptôme de la tendance à ignorer les contraintes importantes spécifiques au domaine de la cybersécurité. Cela est également évident dans les discussions sur la possibilité d'évasion d'IDS et d'empoisonnement des modèles par un attaquant.

Lorsque les chercheurs mettent en évidence la susceptibilité des IDS aux empoisonnements et au contournement de détection par le trafic réseau malveillant, les auteurs manipulent souvent les données dans l'espace de caractéristiques (feature space). Ce phénomène se retrouve dans des articles entièrement consacrés à ce sujet, mais parfois aussi dans les sections « modèle de menace » des concepts d'IDS (Goyal, Han, Wang et Bates, 2023; Ur Rehman *et coll.*, 2024; Nguyen *et coll.*, 2023; Goyal *et coll.*, 2024). Cependant, ce n'est que récemment que certains

Tableau 2.1 L'utilisation de l'OpTC dans les travaux antérieurs

	Évaluation	**Méthode**	**Étiquettes**	**Granularité**	**Métriques**
Golczynski et Emanuello (2021)	2 hôtes bénins, 2 malicieux	NLP	étiquetage manuel	hôte	précision
SK-Tree (Cochrane *et coll.*, 2021)	un seul hôte	série temporelle	étiquetage manuel	l'arbre de processus	AUC
Medad *et coll.* (2021)	un seul hôte	clustering de graphe	non étiqueté	hôte	-
DeepTaskAPT (Mamun et Shi, 2021)	2 hotês malicieux	LSTM	non étiqueté	l'arbre de processus	exactitude, précision, rappel
NetHawk (Bowman, 2022)	FILE, FLOW, PROCESS	détection d'anomalie graphique	étiquetage manuel	utilisateur, hôte	précision, rappel, F1
TapTree (Mamun et Buffett, 2022)	21 hôtes	*sequential pattern mining*	-	l'arbre de processus	exactitude, précision, rappel, FPR
Pikachu (Paudel et Huang, 2022)	flux réseau	*temporal graph walk*	non étiqueté	flux réseau	TPR, FPR, AUC
ANUBIS (Anjum *et coll.*, 2022)	jeu complet	*Bayesian Neural Network*	-	événement	exactitude, précision, rappel, F1, FPR
Paradise (Wu *et coll.*, 2023)	18 hôtes	similarité de processus	-	processus	TPR, FPR
SCAHunter (Ahmed *et coll.*, 2023)	1 scénario malicieux	signatures d'attaques	-	-	-
Euler (King et Huang, 2023)	flux réseau	GNN	étiquetage manuel	hôte	TPR, FPR, AUC
Kairos (Cheng *et coll.*, 2024)	6 hôtes	GNN	méthode d'étiquetage non clarif.	intervalle de temps	exactitude, précision, rappel, AUC
MEGR-APT (Aly *et coll.*, 2024)	3 hôtes	GNN	-	sous-graphe	exactitude, précision, rappel, F1, AUC
Flash (Ur Rehman *et coll.*, 2024)	20 hôtes	plongements sémantiques, GNN	méthode d'étiquetage non clarif.	nœud de graphe	précision, rappel, F1

travaux importants ont reconnu que la vulnérabilité des IDS aux attaques de corruption de données et d'évasion peut être considérablement surestimée dans la littérature (Apruzzese *et coll.*, 2023a; Apruzzese, Fass et Pierazzi, 2024; Catillo, Pecchia, Repola et Villano, 2024). Le même type de transformation des données dans l'espace de problème (problem space) est souvent simplement impossible, car cela peut entrainer des violations de protocoles ou des états du système impossibles. Ces hypothèses sont néanmoins intégrées dans la conception de l'IDS, ce qui oblige les auteurs à chercher des compromis avec d'autres choix. Les contrôles de sécurité efficaces, y compris l'IDS, devraient être conçus autour de comportement du système invariant, pour que les efforts soient concentrés sur la détection des risques réels.

Pourtant, ces lacunes méthodologiques ne se limitent pas aux systèmes de suivi de provenance. La recherche de méthodes alternatives de détection n'a jamais cessé. Les approches qui ont été explorées au fil des trente dernières années sont trop nombreuses pour être toutes abordées ici. Les revues systématiques de la littérature se consacraient entièrement à ce sujet et elles ont identifié des centaines d'articles pertinents (Hagemann et Katsarou, 2020; Nassif, Talib, Nasir et Dakalbab, 2021; Levshun et Kotenko, 2023; Zipperle *et coll.*, 2023; Smiliotopoulos, Kambourakis et Kolias, 2024). La recherche a atteint un tel degré de saturation qu'elle s'est poursuivie dans les sous-domaines, tels que l'IoT, les menaces internes, le mouvement latéral, etc. Cependant, tous agissent du même problème : la détection d'activités malveillantes.

2.2 Perspective industrielle

Selon un récent rapport de Google Threat Analysis Group et Mandiant (2024), 2023 a établi un record en ce qui concerne le nombre d'exploitations « jour zéro ». L'analyse des menaces par DeepInstinct (Deep Instinct, 2023a) montre qu'il y a eu deux fois plus d'infections par un logiciel rançongiciel en 2023 que l'année précédente. De plus, les rapports de CrowdStrike (2023) font état d'une hausse constante des intrusions réseau, une tendance qui s'est poursuivie en 2024 avec plusieurs incidents critiques ayant entrainé des conséquences désastreuses (CISA, 2024; Canadian Centre for Cyber Security, 2024).

Les systèmes cyberphysiques ont saturé la société moderne, et par conséquent, presque chaque entreprise est devenue une cible pour un nombre croissant de cyberattaques (Huntress, 2023) provenant de groupes criminels et d'acteurs gouvernementaux dans le contexte d'un climat géopolitique et économique tendu. L'évolution des cyberattaques a complètement dépassé la capacité des entreprises à maintenir leurs défenses à un bon niveau. D'ailleurs, l'utilisation d'outils d'intelligence artificielle générative par les criminels (Fang, Bindu, Gupta, Zhan et Kang, 2024; Red Canary, 2024; National Cyber Security Centre, 2024b) entrainera des défis de sécurité encore plus importants dans les années à venir.

Pourtant, les IDS industriels ont du mal à s'adapter à de nouvelles méthodes de détection, ce qui les rend souvent dépassés face à de nouvelles attaques.

Selon une étude récente de Dong *et coll.* (2023), le taux d'adoption des systèmes de détection d'intrusion par graphes de provenance est très faible dans l'industrie. Même si les experts en sécurité conviennent que ces systèmes génèrent des alertes de meilleure qualité (Jansen *et coll.*, 2024; Mink *et coll.*, 2023), ils restent principalement une curiosité académique. Les rapports de l'industrie confirment que les procédures et les outils des centres de SOC restent globalement inchangés au fil des ans et continuent de s'appuyer sur des bibliothèques de signatures statiques (Chuvakin, Sachdev, Lauritzen, Rudoll et Glowacki, 2024). Bien que les techniques d'apprentissage automatique aient obtenu des résultats impressionnants dans d'autres domaines, leur usage pour la détection des cyberattaques est perçu comme irréalisable (Apruzzese, Laskov et Schneider, 2023b), ce qui fait qu'elles n'ont pas gagné en popularité. L'un des facteurs contribuant à l'écart entre les pratiques académiques et industrielles est le manque des jeux de données de haute qualité accessibles au public, qui représenteraient bien la condition du réseau pendant une attaque.

En raison du manque de rapports pertinents de l'industrie, il est difficile de mesurer de manière quantitative le taux d'adoption par les équipes SOC des systèmes d'apprentissage machine en tant qu'IDS, car il est largement admis que les équipes de sécurité utilisent principalement les IDS par signature aujourd'hui. La seule enquête sur ce sujet, qui avait été identifiée lors de la

revue de littérature, est le sondage SANS 2021 SOC (Crowley et Pescatore, 2021). Il a constaté que l'apprentissage automatique et l'intelligence artificielle sont parmi les outils les moins utilisés. À l'exception des technologies de tromperie, comme les *honeypots*, elles produisent les résultats les moins satisfaisants.

On peut, cependant, estimer indirectement l'utilisation de l'IDS par anomalie en examinant la communauté de sécurité active qui crée et entretient de grandes bibliothèques ouvertes de signatures et pour diverses IDS (Splunk Threat Research Team, 2024; SigmaHQ, 2024; Cisco, 2024; Wazuh Inc., 2024). Elles sont régulièrement mises à jour en réponse aux menaces émergentes et aux attaques découvertes récemment. La communauté de la sécurité a compris depuis longtemps l'importance de partager l'intelligence sur les acteurs malveillants pour atteindre l'immunité collective pour tous les joueurs de l'industrie. Au départ, l'échange d'informations sur les attaques se faisait principalement sous la forme d'indicateurs de compromission (IOC). Cependant, ces indicateurs sont éphémères par nature (Bianco, 2013), donc la recherche sur les attaques est maintenant axée sur les tactiques, techniques et procédures (MITRE, 2023). Par convention, les rapports de menaces (e.g., Nationaal Cyber Security Centrum (2024)) sont distribués librement et incluent des indicateurs pertinents de compromission. Ces derniers peuvent être directement transformés en signatures par les équipes SOC.

On ne trouve pas de communauté similaire pour collaborer et partager les modèles de détection d'anomalie, à l'exception de la recherche académique.

La pratique du SOC moderne peut être éclairée par une série d'études qualitatives qui mettent en évidence l'existence de processus complexes liés à la gestion et à l'assurance de grands catalogues de détection (Vermeer *et coll.*, 2023; Vermeer, Van Eeten et Gañán, 2022). Les systèmes de détection par anomalie sont souvent considérés par leurs créateurs comme une méthode plus puissante (e.g., dans Anjum *et coll.* (2022); Macas, Wu et Fuertes (2022); King et Huang (2023)). Cependant, l'opinion des analystes est que, pour obtenir les meilleurs résultats de détection, il est nécessaire de combiner les deux types de systèmes (Mink *et coll.*, 2023). En revanche, les IDS par anomalies sont plus susceptibles au retrait de l'opération en cas d'abaissement de la

performance (Mink *et coll.*, 2023), mesurée par le nombre de faux positifs (Alahmadi, Axon et Martinovic, 2022). Cette métrique est le principal indicateur de performance de détection utilisé dans les milieux industriels (Kokulu *et coll.*, 2019; Vermeer *et coll.*, 2023; Vielberth, Bohm, Fichtinger et Pernul, 2020) (contrairement aux mesures statistiques plus traditionnelles utilisées dans la recherche académique, telles que la précision, le rappel et la mesure F1).

Dans les grandes entreprises, la gestion de la bibliothèque de milliers de signatures est souvent faite par l'équipe de l'ingénierie de détection (Lange, 2023). C'est un rôle émergent, mais bien défini, dans les équipes de sécurité relativement grandes : il consiste à maintenir un équilibre entre une bonne couverture des vecteurs d'attaques possibles et un taux acceptable de fausses alertes.

En résumé, l'industrie montre clairement une préférence pour les IDS par signature, qui se trouvent au cœur des meilleures pratiques et des technologies actuelles. Cette discussion sur l'importance des signatures pour la défense en profondeur se poursuit avec deux exemples réels d'intrusions chez des fournisseurs de services infonuagiques mondiaux.

2.2.1 L'incident du faux jeton d'Exchange Online

L'importance des signatures d'intrusion peut être clairement démontrée par l'incident majeur sur le réseau Microsoft qui a eu lieu entre mai et juin de 2023. Le but ultime du piratage attribué à un APT (Microsoft Threat Intelligence, 2023) était les diverses agences gouvernementales, dont les pirates ont réussi à exfiltrer plus de 60 000 courriels privés (Demirjian, 2023; Cyber Safety Review Board, 2024). Cette attaque sophistiquée a été réalisée en exploitant l'ordinateur d'un employé afin d'extraire des données, en utilisant l'ingénierie inverse, les failles cryptographiques, les réseaux RPV et le réseau TOR. Malgré que les capacités de détection et de réponse de Microsoft soient parmi les plus sophistiquées du secteur, l'acteur malveillant a réussi à échapper à la détection pendant toute la durée de l'intrusion dans le réseau.

L'accès non autorisé a finalement été découvert par un des organismes gouvernementaux compromis. Bien que la plupart des détails sur l'attaque soient encore classifiés, l'avis public

indique que l'activité malveillante a déclenché une signature statique spécialement conçue pour détecter cette attaque (CISA, 2023; Cyber Safety Review Board, 2024). Un changement inattendu d'une seule valeur dans un fichier de journal a dirigé l'équipe de sécurité vers la découverte d'une attaque APT qui a échappé à tous les autres contrôles de sécurité, y compris plusieurs systèmes de détection d'anomalies (Microsoft Security Response Center, 2023).

Bien que ce soit un exemple isolé, il met en contexte la qualité la plus importante de l'IDS par signature : la spécificité de la détection. Les signatures sont conçues pour détecter un changement d'état connu ; il est donc souvent possible de déterminer à l'avance si le changement est malveillant ou non. Cette analyse se fait pendant le processus de développement de la détection, ce qui permet aux équipes de sécurité de gagner du temps et de réagir plus rapidement face aux menaces.

Quant à l'IDS par anomalie, c'est le contraire. En effet, dans les environnements complexes, les anomalies peuvent avoir plusieurs causes. Lorsqu'un IDS génère une alerte, il est impossible de déterminer si la cause de l'évènement est malveillante ou simplement due à un changement d'état bénin. Par conséquent, les analystes de sécurité doivent consacrer beaucoup de temps à une analyse approfondie pour éviter les faux négatifs, ce qui entraine un retard dans la réponse.

Les équipes de sécurité sont optimisées pour une reprise rapide après un incident de sécurité, ce qui est l'une des raisons pour lesquelles elles résistent souvent à l'adoption de systèmes de détection d'anomalies. Les besoins en ressources pour un IDS augmentent proportionnellement au nombre d'alertes qu'il génère. Cela rend le taux de faux positifs un indicateur crucial pour une évaluation réaliste de l'efficacité d'un IDS.

2.2.2 L'incident avec le compte de service Cloudflare

Cette opportunité d'apprendre sur les pratiques d'un fournisseur de services infonuagiques mondial, offerte par le bilan discuté ci-dessus, est extrêmement rare. Malgré les efforts constants des organismes nationaux et internationaux pour appliquer à la cybersécurité certaines normes de l'industrie aérospatiale concernant les enquêtes sur les incidents, la plupart des entreprises

aujourd'hui restent très discrètes par rapport aux détails des incidents. Par conséquent, les partenaires industriels et la communauté scientifique ne peuvent pas apprendre les meilleures méthodes de détection d'intrusion, celles qui nécessitent des recherches approfondies.

L'incident de sécurité survenu chez Cloudflare en novembre 2023 est un autre rare exemple (Prince, Graham-Cumming et Bourzikas, 2024) de transparence qui divulgue des détails sur l'intrusion et la méthode de détection.

Compte tenu des informations très restreintes, on peut soupçonner que, ici encore, les attaquants ont trébuché sur une signature statique dans un système SIEM. La surveillance des modifications apportées aux comptes d'utilisateurs privilégiés, comme ceux des administrateurs, est une recommandation et une exigence de nombreuses normes en cybersécurité. Par conséquent, ce type de détection est l'une des alertes les plus fréquentes, et les équipes de sécurité la maitrisent bien. De plus, les bibliothèques publiques de signature contiennent de nombreux exemples d'alertes de ce genre, qui ont été développées pour plusieurs plateformes (SigmaHQ, 2024; Splunk Threat Research Team, 2024).

2.2.3 Les indicateurs de performance d'IDS

Le rapport de Cloudflare inclut la chronologie de la détection de l'incident, une information également très utile. D'après l'article, le SIEM a détecté l'évènement suspect en moins de deux minutes. L'analyste a mis 12 minutes pour recevoir l'alerte et la trier, puis l'équipe a eu besoin de 13 minutes supplémentaires pour établir le contexte, mener l'enquête et résoudre le piratage du compte administrateur.

Dans un SOC, le temps moyen de détection et le temps moyen de résolution sont des métriques clés pour évaluer la performance de l'équipe de sécurité (Onwubiko et Ouazzane, 2022; Tines, 2023; Crowley, Filkins et Pescatore, 2023). Ces deux indicateurs s'appliquent à tous les outils utilisés par le SOC. L'évolution constante des menaces et leurs méthodes d'attaque a même donné lieu à la création du *555 Benchmark* (Sysdig, 2023). Selon cette norme proposée, les équipes

de sécurité n'ont que 10 minutes pour prévenir l'accès persistant des pirates informatiques au réseau après une attaque.

Cependant, cette facette de la performance est souvent négligée par les recherches. Il est nécessaire non seulement d'identifier avec précision les activités malveillantes. Le système de détection proposé doit aussi déclencher l'alerte très proche du moment de l'évènement et fournir suffisamment de contexte pour trier et enquêter sur l'alerte dans les minutes qui suivent.

Dans la recherche, les métriques les plus courantes pour évaluer les résultats d'un modèle d'apprentissage machine sont la précision, le rappel et la mesure F1, qui se calculent comme suit :

$$Precision = \frac{TP}{TP + FP} \tag{2.1}$$

$$Recall = \frac{TP}{TP + FN} \tag{2.2}$$

$$F_1 = \frac{2 \cdot \mathrm{TP}}{2 \cdot \mathrm{TP} + \mathrm{FP} + \mathrm{FN}} \tag{2.3}$$

Ces indicateurs peuvent être trompeurs pour les jeux de données extrêmement déséquilibrés (Boughorbel, Jarray et El-Anbari, 2017), comme c'est souvent le cas avec les intrusions réseau. Une méthode courante pour évaluer la performance d'un modèle consiste donc à calculer la surface sous la courbe ROC (Junge et Dettori, 2018).

Il est bien connu depuis longtemps que le phénomène « d'oubli de la fréquence de base » (base rate fallacy) (Axelsson, 1999) rend ces indicateurs standards peu fiables pour évaluer l'utilité du modèle dans le cadre des tâches quotidiennes du SOC. Puisque les réseaux modernes produisent des milliards d'évènements par jour, une légère baisse de la précision peut entrainer des milliers de fausses alertes. Ces derniers nécessitent une enquête de la part de l'équipe de sécurité, parfois

même en raison d'exigences règlementaires. Cependant, dans le domaine de la cybersécurité, il est souvent difficile d'interpréter même les indicateurs sophistiqués, tels que l'AUC-ROC. Certaines spécialistes recommandent donc d'utiliser la courbe de précision-rappel comme alternative (Arp *et coll.*, 2022). Malheureusement, cette mesure n'est pas toujours appropriée, et peu de publications l'utilisent réellement.

Pour des raisons pratiques, il est crucial d'évaluer le nombre réel de détections réelles et fausses. Cette tâche est notoirement ardue et certains auteurs de systèmes IDS ont donc décidé de renoncer complètement à ces métriques traditionnelles. Cette approche peut être illustrée par le système de Golczynski et Emanuello (2021), appuyé par les méthodes de NLP (Natural Language Processing). Il fonctionne de façon bout à bout pour compléter ses prédictions par une explication, mais, par contre, il ne tente pas de calculer les métriques de performance consolidées.

Puisqu'il est impossible d'entrainer un modèle parfait (Gates et Taylor, 2006; Sommer et Paxson, 2010), l'utilité pratique de l'IDS peut s'améliorer si le nombre réel des faux positifs est réduit. L'approche de la thèse de doctorat de Bowman (2022) se distingue par l'introduction d'une étape de réduction des résultats du modèle à l'aide d'euristiques. Une autre méthode, proposée par Cao, Blaise, Panichella et Verwer (2024), consiste à créer un automate fini pour modéliser le trafic normal sur le réseau. Chaque alerte générée est ensuite associée à une transition d'état dans cet automate fini, permettant une analyse de cause racine et un tri plus rapide des alertes critiques.

Dans la plupart des équipes de sécurité, il existe déjà un certain mécanisme pour réduire le bruit d'alertes par rapport à l'activité normale. Donc, si l'IDS intègre cette fonctionnalité directement, la mise en production sera plus simple.

Une voie de recherche parallèle tente de mesurer l'impact des faux positifs sur les opérations SOC réelles.

Une analyse approfondie de la manière dont les équipes SOC gèrent de grandes bibliothèques de signatures (Vermeer *et coll.*, 2022) a révélé que la probabilité de mise à jour d'une signature (*fine-tuning*) peut doubler en raison d'une seule fausse alerte. C'est une stratégie de prévention adoptée par les équipes de sécurité pour se protéger contre « l'inondation par fausses alertes », considérée comme le pire scénario dans un SOC. Une étude récente de Vermeer *et coll.* (2023) a confirmé que le taux de fausses alertes était un facteur important dans la décision de mettre en production une alerte, et cela est également appuyé par les rapports de l'industrie (Carey et Jin, 2020; Deep Instinct, 2023b; Tines, 2023; Censys, 2024).

Les mêmes attentes s'appliquent aux outils d'apprentissage machine. En comparaison avec des règles statiques, les équipes SOC sont plus enclines à abandonner un modèle d'apprentissage machine déjà opérationnel s'il ne permet pas de limiter facilement le nombre de faux positifs (Mink *et coll.*, 2023).

Cela met en évidence l'importance de l'explicabilité de la prédiction comme condition préalable à un déploiement industriel réussi d'un système de détection d'intrusion.

D'un autre côté, l'étude qualitative des problèmes dans un SOC réalisée par Kokulu *et coll.* (2019) a constaté que les équipes ne perçoivent pas le taux élevé des fausses alertes comme étant un « problème majeur ». Cette conclusion semble en contradiction avec d'autres constatations. Une interprétation possible pour expliquer cette apparente incohérence est que les faux positifs sont un phénomène courant et prévisible dans tout processus d'analyse. Les spécialistes du SOC ont développé des procédures efficaces pour limiter l'impact négatif des fausses alertes (par exemple, en utilisant le « fine-tuning » discuté plus haut). Cela permet d'éviter les interruptions majeures dans la routine quotidienne du centre opérationnel. Toutefois, les auteurs d'IDS ne peuvent pas utiliser cela comme un prétexte pour les négliger dans leur travail.

L'IDS suggéré dans ANUBIS (Anjum *et coll.*, 2022), entre autres, tente de résoudre ce défi en associant une cote de fiabilité à chaque alerte générée. Cependant, cette méthode n'est pas applicable à tous les algorithmes d'apprentissage automatique, ce qui la rend difficilement

utilisable. De plus, les analystes en sécurité sont réticents à assumer la responsabilité d'interpréter une telle note de fiabilité liée aux résultats du modèle (Mink *et coll.*, 2023).

Étant donné qu'il est irréaliste d'éliminer toutes les fausses alertes (Gates et Taylor, 2006), les outils peuvent utiliser un mécanisme de budget d'alertes pour limiter le nombre d'alertes qu'ils génèrent. Cette méthode assure que l'équipe de sécurité ne soit pas submergée par des taux de fausses alertes « excessifs ». Cette stratégie est fréquemment employée dans un SOC, mais elle reste curieusement peu recherchée, avec l'œuvre de Ho *et coll.* (2021) comme seul exemple notable. La Section 4.4 explore un modèle d'analyse non supervisé d'anomalies incluant « un budget d'alerte » pour atteindre une bonne détection d'attaques.

2.3 Problèmes ouverts

Ce chapitre a retracé l'évolution des systèmes de détection d'intrusion dans la recherche académique et dans l'industrie afin de comprendre pourquoi les méthodes actuelles ne sont pas adaptées à la réalité d'aujourd'hui. Les chapitres suivants se concentreront sur le jeu de données OpTC, et se dirigerons vers un concept innovant de l'IDS. L'objectif est de surmonter les limites des systèmes actuels et de faciliter la mise en productions bout à bout d'IDS dans un cadre d'entreprise. Pour mieux comprendre les défis à relever, voici un aperçu des principaux problèmes non résolus de ce domaine.

Le plus grand défi discuté jusqu'à présent est l'oubli de la fréquence de base. Il n'y a rien de surprenant à ce que ce soit la raison principale pour laquelle l'utilité perçue de l'IDS diffère entre la recherche et l'industrie. Au cours de vingt ans, il y avait plusieurs tentatives pour attirer l'attention des chercheurs à cette limitation majeure (Axelsson, 1999; McHugh, 2000; Gates et Taylor, 2006; Sommer et Paxson, 2010; Kenyon, Deka et Elizondo, 2020). Cependant, cette discussion n'a pris de l'ampleur que récemment, en partie grâce aux travaux sur l'intelligence artificielle explicable. Quelques publications importantes ont mis en évidence les limites des pratiques courantes qui rendent les résultats inadaptés à la mise en œuvre industrielle (Arp *et coll.*, 2022; Apruzzese *et coll.*, 2023b; Braun, Pekaric et Apruzzese, 2024; Rimmer, Nadeem,

Verwer, Preuveneers et Joosen, 2022; Hesford *et coll.*, 2024; Nadeem *et coll.*, 2023; Pendlebury, Pierazzi, Jordaney, Kinder et Cavallaro, 2019). Le thème récurrent est la dépendance de la nature confinée des environnements de test, identifié dans Arp *et coll.* (2022) comme *P9 – Lab-Only Evaluation*.

Ces lacunes méthodologiques sont la conséquence directe du manque de données disponibles au public. Puisque les données internes des réseaux réels ne peuvent pas être divulguées, les chercheurs doivent s'appuyer sur un petit nombre de jeux de données synthétiques qui ne reflètent pas la complexité du monde réel et ne permettent pas le développement de systèmes de détection généralisables.

Les équipes de sécurité opérationnelle sont naturellement réticentes à partager l'information avec des groupes externes, car cela pourrait donner un avantage à l'adversaire, compte tenu de la nature conflictuelle de la cybersécurité, qui peut être modélisée comme un jeu de défense et d'attaque (Hu, Li, Fu, Cansever et Mohapatra, 2015). De plus, la diffusion d'informations et la recherche académique ouverte sont empêchées par nombreuses obligations légales et contractuelles qui existent pour limiter l'utilisation ou le partage de données et de rapports d'enquêtes sur des incidents (Woods, Böhme, Wolff et Schwarcz, 2023). Cela contraste fortement avec les domaines du traitement du langage naturel et de la vision par ordinateur, où les jeux de données sont construits et partagés librement. Malheureusement, la seule option pour les chercheurs aujourd'hui est d'utiliser des données synthétiques. Il est donc impératif de maximiser la valeur des jeux de données disponibles et de respecter les contraintes importantes des réseaux réels.

Une autre conséquence de l'accumulation de recherches est un rapport exagéré de bonnes métriques. Au fil des ans, les chercheurs ont développé des méthodes pour atteindre des indicateurs de performance toujours plus élevés sur les mêmes données de base. Toutefois, comme c'est constaté dans Liu, Engelen, Lynar, Essam et Joosen (2022a), compte tenu du grand nombre d'erreurs dans les ensembles de données, certains des résultats rapportés sont tout simplement irréalisables sans surapprentissage et corrélations illusoires. De plus, les systèmes

qui s'appuient, explicitement ou implicitement, sur la nature confinée des données ne sont pas adaptés à lutter contre la dérive conceptuelle constante des réseaux dynamiques de la vraie vie. Une revue récente (Arp *et coll.*, 2022) des défis communs de l'apprentissage automatique dans le domaine de la cybersécurité a révélé que l'évaluation uniquement dans le laboratoire et l'absence d'évaluations de base sont des problèmes très répandus dans la littérature.

Des découvertes récentes suggèrent également que les indicateurs de performance traditionnels, tels que la précision, le rappel et la mesure F1, ne représentent pas des résultats significatifs en cybersécurité (Arp *et coll.*, 2022). Avec l'oubli de la fréquence de base (Axelsson, 1999), cela rend les solutions IDS innovantes prohibitivement couteuses dans les environnements corporatifs réels. Cette situation est souvent la principale raison pour laquelle les équipes de sécurité opérationnelles rejettent les IDS par anomalie (Alahmadi *et coll.*, 2022).

2.4 Conclusion

Au cours des trente dernières années, une multitude d'idées et de techniques de détection ont vu le jour. Parmi celles-ci, une famille de méthodes de détection d'anomalies qui analysent l'activité du système à un niveau bas semble particulièrement prometteuse. Les graphes de provenance s'appuient sur une base théorique solide et offrent une grande flexibilité pour détecter les attaques les plus subtiles, souvent associées aux menaces persistantes avancées.

Des recherches antérieures ont révélé le potentiel des systèmes de suivi de provenance pour détecter efficacement les attaques modernes. Malheureusement, ces travaux ne vont pas jusqu'à proposer une solution complète et adaptée à un environnement d'entreprise. Un SOC moderne doit suivre des processus complexes et respecter des contraintes externes. Par conséquent, la réussite de la mise en œuvre d'un IDS industriel dépend de facteurs bien plus que les métriques de performance standard utilisées dans les évaluations en laboratoire. La création et l'évaluation de systèmes de détection d'intrusions novateurs nécessitent une approche de bout en bout, prenant en considération non seulement les risques et menaces actuels, mais aussi les invariants de comportement sous-jacents de système.

Pour y arriver, les expériences doivent se rapprocher au maximum des conditions réelles du réseau, par exemple, en utilisant un jeu de données réaliste. Ce chapitre a identifié le jeu de données OpTC comme une ressource appropriée pour combler l'écart entre la recherche actuelle et les besoins de l'industrie. Cette recherche se poursuit pour explorer comment les caractéristiques du jeu OpTC contribuent à la conception d'un IDS adapté aux exigences des réseaux modernes, et ainsi favorise le passage de la recherche académique à une mise en œuvre réussie d'un IDS.

CHAPITRE 3

ÉTIQUETAGE DU JEU DE DONNÉES OPTC

Ce chapitre va utiliser le jeu de donnée OpTC pour développer un processus d'étiquetage et transformer le document de « réalité de terrain » (ground truth) en un ensemble d'étiquettes adaptées à l'utilisation avec les techniques d'apprentissage machine.

Pour motiver l'intérêt pour ce jeu de données, la discussion commence par le problème de la pénurie de données ouvertes de haute qualité, puis aborde les limites des jeux de données les plus populaires utilisés dans la recherche sur l'IDS.

3.1 Principaux jeux de données pour la détection d'intrusion

Comme déjà discuté, la plupart des études actuelles sur les systèmes de détection d'intrusion sont basées sur un nombre très limité de jeux de données d'intrusion publiques. Il existe six jeux de données principaux pour la simulation d'intrusion et les recherches connexes : **KDD99**, **NSL-KDD**, **Kyoto 2006+**, **UNSW-NB15**, **CIC-IDS2017** and **CSE-CIC-IDS2018**. Selon des enquêtes récentes sur ce domaine, ces jeux de données sont utilisés pour évaluer plus de 90% des approches d'apprentissage machine et d'apprentissage profond. KDD99 et NSL-KDD en sont seuls responsables de près de 60% de l'usage. On compte des centaines de publications chaque année, avec une augmentation constante (Hagemann et Katsarou, 2020; Ahmad *et coll.*, 2021; Nassif *et coll.*, 2021; Maseno, Wang et Xing, 2022).

Cela mène à un domaine de recherche extrêmement saturé, avec de nombreuses méthodes de détection puissantes qui peuvent atteindre des résultats presque parfaits sur les données de référence, au point qu'on ne peut plus mesurer une amélioration statistiquement (Apruzzese *et coll.*, 2023b).

Le problème ne se limite pas à l'utilisation excessive, mais aussi se concerne des inexactitudes dans les données brutes et les étiquettes (McHugh, 2000; Liu *et coll.*, 2022a; Engelen, Rimmer et Joosen, 2021; Lanvin *et coll.*, 2023; Rosay, Cheval, Carlier et Leroux, 2022). Ces dernières sont

assez importantes pour remettre en question la fiabilité de tous les résultats obtenus à partir de ces données. Les suggestions des auteurs des articles ci-dessus n'ont pas attiré assez d'attention, et les jeux de données originaux continuent d'être utilisés pour de nouvelles publications. Au fil des ans, plusieurs autres ensembles de données pour évaluer les IDS ont été proposés (Creech et Hu, 2013; Highnam, Arulkumaran, Hanif et Jennings, 2021; Myneni *et coll.*, 2023), mais ils ne sont pas entrés en usage général. Peut-être est-ce dû à la même limitation, qui est d'être basés sur des environnements synthétiques de petite taille.

Les systèmes de suivi de provenance imposent des restrictions supplémentaires sur la disponibilité des données d'entrainement. En effet, elles doivent être collectées à l'aide d'outils spécialisés au niveau très basique du système. De plus, elles doivent être très complètes, car une télémétrie partielle va produire un graphe déconnecté, ce qui n'est pas utile pour la détection d'attaques. Compte tenu de la popularité de ces systèmes de détection d'intrusion, de nombreux auteurs ont dû utiliser des données personnalisées ou propriétaires, ce qui est le cas dans 65% des articles (Zipperle *et coll.*, 2023). D'un autre côté, ce format étant similaire à la télémétrie produite par les solutions EDR existantes, il est plus facile de collaborer avec un partenaire industriel pour obtenir les données nécessaires. Cependant, ces ensembles de données privées ne peuvent pas être partagés, ce qui rend impossible la reproduction des résultats.

Jeux do donnés KDD

Le **KDD99** (UC Irvine, 1999) est l'un des jeux de données les plus populaires. Publié en 1999, il vise à bâtir un modèle capable de distinguer les connexions « bonnes » et « mauvaises » sur un réseau. La qualité du jeu de données a fait l'objet de nombreuses critiques au fil des ans (McHugh, 2000; Creech et Hu, 2013). Des tentatives ont été faites pour aborder certaines de ces limites, ce qui a conduit à la création du jeu de données **NSL-KDD** en 2009 (Tavallaee, Bagheri, Lu et Ghorbani, 2009). Ce jeu de données contient significativement moins de données que l'original, et sa structure est semblable à celle d'un jeu de données typique pour l'apprentissage automatique. Cependant, la plupart des limitations n'ont pas été résolues.

Il est impensable qu'un ensemble de données datant de plus de 20 ans puisse significativement refléter les cyberattaques actuelles. Le KDD99 ne compte que 5 millions d'entrées, qui peuvent être traitées en quelques secondes sur un ordinateur moderne. En comparaison, les environnements infonuagiques actuels produisent considérablement plus de télémétrie que ce qu'un seul système pourrait gérer, analyser ou collecter en entier. Les systèmes de détection d'intrusion actuels font face à deux défis majeurs : la gestion efficace des volumes importants de télémétrie et la sélection des entrées les plus significatives pour la détection. Or, de nombreuses études récentes, qui se basent sur des jeux de données anciens, négligent souvent cet aspect crucial pour une mise en production.

Jeux de données de l'Institut canadien sur la cybersécurité

CIC-IDS2017 (Sharafaldin *et coll.*, 2018) et **CSE-CIC-IDS2018** (Canadian Institute for Cybersecurity, 2018) sont des jeux de données provenant de l'Institut canadien sur la cybersécurité. Ils ont été créés pour simuler des menaces à un service internet typique, comme un site web d'entreprise.

Ces deux ensembles de données sont considérablement plus larges, avec CSE-CIC-IDS2018 comptant environ 16 millions d'enregistrements de captures réseau PCAP complètes. Mais cette capacité reste dans les limites de la puissance de traitement d'un ordinateur portable standard. La principale différence entre ces jeux de données et KDD99 est qu'ils sont conçus pour représenter avec précision les techniques d'attaque contemporaines, avec une définition claire de ce qui constitue une attaque, au lieu de simplement essayer de distinguer le trafic « bon » et « mauvais ». Environ 17% du jeu de données CSE-CIC-IDS2018 a été classé comme étant l'un des six types d'attaques (Leevy et Khoshgoftaar, 2020). Chaque enregistrement est explicitement associé à une étiquette bénigne ou malveillante. Cela offre aux experts en analyse de données provenant d'autres disciplines un cadre familier, favorisant ainsi l'utilisation d'outils d'apprentissage machine existants et encourageant l'expérimentation avec de nouvelles méthodes algorithmiques. Après leur sortie, les deux jeux de données ont connu une augmentation de leur utilisation. Ils restent cependant pertinents et utiles, en particulier en raison de la disponibilité des fichiers

PCAP et des journaux. Ces derniers peuvent servir à créer de nouvelles caractéristiques ou à configurer un environnement cible afin d'améliorer la transférabilité du modèle.

Cependant, la principale limite de cette approche en ingénierie de jeux de données est sa dépendance excessive par rapport à la disponibilité des étiquettes.

Bien que ce soit une opinion minoritaire dans le milieu académique, on sait depuis longtemps que le transfert d'IDS utilisant les modèles d'apprentissage machine entre différents environnements est impossible (Sommer et Paxson, 2010). Par conséquent, chaque nouvelle installation requiert un nouvel entrainement. Celui-ci demande une grande quantité d'étiquettes écrites par des spécialistes en cybersécurité. Selon une récente analyse des méthodes d'étiquetage dans de grandes sociétés équipées de SOC avancés, la création de jeux de données exige des investissements considérables en ressources, et les experts expérimentés eux-mêmes ne parviennent pas à évaluer l'effort nécessaire (Braun *et coll.*, 2024). En raison de la pénurie persistante de main-d'œuvre en cybersécurité (Herron et Quan, 2022), la plupart des entreprises pourraient trouver prohibitif d'utiliser des algorithmes d'apprentissage supervisé pour la détection d'intrusion.

Jeux de données de LANL

Les deux jeux de données importants suivants proviennent du laboratoire national de Los Alamos National Laboratory.

Le jeu de données **Comprehensive, Multi-Source Cyber-Security Events dataset** (Kent, 2015a) provient du réseau interne du laboratoire Los Alamos. Il s'agit d'une expérience de 58 jours durant lesquels une équipe rouge a lancé plusieurs attaques. Contrairement à la plupart des jeux de données simulés, c'est un rare exemple qui montre la complexité de travailler avec une représentation du comportement des usages réels (Kent, 2015b). C'est pourquoi il a fait l'objet de plusieurs articles qui se sont concentrés spécifiquement sur l'étude des graphes d'authentification des utilisateurs (Kaiafas *et coll.*, 2018; Bian *et coll.*, 2019; Powell, 2020).

De la même manière, le jeu de données **Unified Host and Network Data Set** (Turcotte, Kent et Hash, 2018) représente la télémétrie provenant du réseau réel de LANL. La période d'expérimentation a été prolongée jusqu'à 90 jours, triplant ainsi la quantité de données disponibles. En revanche, contrairement à la version précédente, celle-ci ne comporte aucune étiquette. De plus, la description officielle ne mentionne pas si ces données contiennent des activités malveillantes. Cependant, certaines sources affirment avoir accès à cette information (Data Study Group Team, 2019). En l'absence d'étiquettes, l'utilisation du jeu de données LANL 2017 dans la recherche reste limitée, bien qu'il puisse s'avérer extrêmement bénéfique pour développer des systèmes IDS efficaces axés sur la performance et l'explicabilité des résultats. Une étude approfondie du jeu de données LANL 2017 pourrait constituer un sujet de recherche intéressant à l'avenir.

3.2 Introduction au jeu de données OpTC

En 2014, en réponse à la pression croissante des attaques APT, DARPA (l'Agence pour les projets de recherche avancée de défense) a lancé un grand projet expérimental visant à améliorer l'efficacité de la détection des activités suspectes au sein d'une entreprise. Le programme Transparent Computing (TC) s'est étalé sur cinq engagements, durant lesquels les participants devaient détecter et prévenir des scénarios d'attaques dans un environnement de plus en plus complexe (Griffith *et coll.*, 2020). Le principal objectif du programme consistait à créer en temps réel des graphes de provenance qui éclaircissent le fonctionnement interne des ordinateurs. DARPA a mis à disposition des scénarios d'engagement ainsi que des données des graphes de provenance dans le domaine public afin de faciliter des recherches complémentaires après les engagements.

Le jeu de données OpTC a été créé en augmentant les résultats de TC jusqu'à 1000 hôtes virtuels. Cela avait pour but de tester l'évolutivité de l'architecture de référence. Cela a généré une quantité incroyable de journaux système et de traces réseau provenant d'un réseau d'entreprise simulé. Le domaine de la recherche académique sur les IDS a longtemps souffert d'un manque

de données d'entrainement de haute qualité (Ahmad *et coll.*, 2021). OpTC peut donc devenir un jeu de données utile pour des études futures.

Bien que l'objectif d'exécuter 1000 hôtes en continu pendant deux semaines ait été atteint, les auteurs ont malheureusement échoué à analyser la totalité de la télémétrie générée par l'environnement simulé. Par conséquent, la publication finale ne contenait que la moitié des données des hôtes (FiveDirections, 2019). Les utilisateurs potentiels du jeu de données doivent prendre en compte la perte significative d'informations et la possibilité que certaines preuves importantes d'activité malveillante aient été perdues.

Cette caractéristique du jeu de données OpTC renforce l'authenticité de la détection des intrusions avancées dans les réseaux d'entreprise.

Tableau 3.1 Comparaison entre les jeux principaux

	Année	**Scénarios**	**Évènements totaux**	**Étiquettes malicieuses**
KDD99	1999	4	4,898,431	3,925,650
NSL-KDD	2009	4	125,973	58,630
LANL 2015	2015	N/A	1,648,275,307	749
CIC-IDS2017	2017	7	2,830,743	557,646
LANL 2017	2018	N/A	5,546,990,084	0
CSE-CIC-IDS2018	2018	7	16,232,943	2,748,235
OpTC	2019	3-5	17,433,324,390	0

Une équipe rouge expérimentée a mené une simulation réelle d'activités malveillantes en exécutant plusieurs attaques attendues dans ce type d'environnement. Les objectifs de l'attaquant correspondent à ceux d'une menace persistante avancée et sont alignés sur le modèle standard de l'industrie des étapes d'attaque (Hutchins, Cloppert et Amin, 2011). En conformité avec les pratiques habituelles pour de tels exercices, les attaquants ont tenu un journal de leurs activités, enregistrant la date, l'origine et la destination de l'attaque, ainsi que la méthode d'exploitation utilisée. L'objectif est d'aider les spécialistes en criminalistique numérique à trouver des traces d'attaques dans les journaux bruts du système. Pour OpTC, les principales étapes de compromission réseau sont documentées dans un fichier « OpTCRedTeamGroundTruth.pdf » qui a été joint aux données divulguées.

Il existe une différence majeure entre les données étiquetées, généralement utilisées pour la recherche en apprentissage machine, et le jeu de données OpTC. En effet, l'équipe rouge ne peut pas fournir d'étiquettes pour les données brutes, puisqu'elle n'a pas accès aux alertes IDS pendant les engagements. La simulation serait dans ce cas irréaliste à cause du biais dans les décisions de l'équipe. La Section 3.4 porte sur l'impact sur les pipelines d'apprentissage machine ainsi que sur leur évaluation.

La raison de contacter une équipe rouge professionnelle était de tirer profit de son expertise dans le traitement d'attaques réelles par une APT. Comme l'a noté DARPA dans la documentation du programme TC, « L'objectif global du programme était de démontrer avec succès la détection des APT dans un ensemble de conditions typiques » (Griffith *et coll.*, 2020). En considérant cet objectif, Windows 10 s'avère être un choix judicieux d'OS, compte tenu de sa popularité dans les milieux professionnels. De plus, on peut s'attendre à ce que l'environnement soit unifié dans un domaine Active Directory, en utilisant des outils standards de Windows tels que RDP et WMI. Les sections suivantes présentent plusieurs artéfacts dans les données pour décrire l'environnement OpTC, vérifier les hypothèses et documenter les caractéristiques distinctives du jeu de données qui peuvent influencer la portabilité des modèles d'apprentissage machine.

Ce résumé représente la quasi-totalité des informations connues sur le jeu de données OpTC. Ce n'est pas très surprenant qu'il soit relativement obscur. Toutefois, il peut servir bien aux besoins de la recherche sur la détection d'intrusions.

- Ce jeu de données a été développé en collaboration avec des partenaires de l'industrie pour représenter les méthodes d'attaque actuelles. Puisque les menaces évoluent constamment, les systèmes IDS plus anciens deviennent inefficaces. Continuer à s'appuyer sur les jeux de données plus anciens ne permettrait pas de garantir le transfert des connaissances. Par conséquent, le jeu de données OpTC ne restera pas longtemps pertinent. Il est donc important que plus de chercheurs l'adoptent dans leurs expériences pour en tirer tous les bénéfices possibles.

- Selon le Tableau 3.1, l'ensemble de données OpTC contient un nombre considérablement plus élevé d'évènements (plus de 17 milliards) que les jeux de données plus anciens. Il atteint une taille qui surpasse les capacités d'un ordinateur standard, avec un volume de données décompressées de 9,7 téraoctets. Les réseaux modernes génèrent constamment des quantités énormes de télémétrie, et les systèmes IDS sont attendus pour fonctionner efficacement en temps réel. L'utilisation de l'ensemble de données OpTC pour tester leurs solutions peut aider les chercheurs à démontrer des temps de réponse rapides.
- Les évènements sont capturés sous forme de fichiers JSON, ce qui est rare dans la recherche ML. De plus, certains champs et types d'évènements ne sont pas du tout documentés. Il s'agit d'une situation réaliste, il faut donc que les utilisateurs du jeu des données OpTC tiennent compte de ce défi lors de la création de leur système de détection d'intrusion.
- Les données brutes ne comportent aucune étiquette. Analyser « la réalité du terrain » requiert des connaissances avancées en cybersécurité. C'est la première étape pour une mise en production de n'importe quel système d'intrusion. Les jeux de données anciens sont beaucoup plus similaires à ceux que les spécialistes en science des données trouvent dans d'autres domaines. Elles permettent de se concentrer sur l'entrainement du modèle et d'éviter l'étape cruciale d'étiquetage. Par conséquent, la mise en production de ces modèles se complique, car il est essentiel de connaitre le format d'étiquette qu'ils peuvent accepter. Plusieurs chercheurs ont déjà parlé de l'importance de ce problème. (Andresini *et coll.*, 2021; Apruzzese *et coll.*, 2023b; Braun *et coll.*, 2024).
- D'un autre côté, puisque les auteurs de systèmes IDS doivent attribuer des étiquettes à OpTC et déterminer la granularité appropriée pour leur système (discutée en détail dans la Section 3.4), cela facilite la compréhension des analystes de sécurité quant à l'adéquation du système IDS proposé à leur environnement.
- Quelle que soit la méthode utilisée pour créer les étiquettes, l'ensemble de données OpTC reste extrêmement déséquilibré. Les évènements malveillants ne représentent qu'une petite proportion de l'ensemble des évènements. Cela reflète de manière fidèle les attaques du monde réel, ce qui peut aider les auteurs d'IDS à éviter l'oubli de la fréquence de base, souvent observée dans les études académiques (Axelsson, 1999).

- Selon les auteurs d'Euler (King et Huang, 2023), OpTC est moins ambigüe que le jeu de données LANL car l'activité de l'équipe rouge y est décrite sous la forme d'un journal plutôt qu'une liste d'évènements. Cette approche ajoute du contexte, ce qui facilite la distinction entre les évènements malveillants et ceux qui sont simplement anormaux. Cela conduit finalement à un système IDS plus efficace. L'inconvénient est que l'analyse des journaux afin d'y trouver des indices d'attaques de l'équipe rouge exige une expertise en criminalistique numérique, ce qui rend cette opération considérablement ardue.

Il est surprenant que, malgré ses avantages, l'outil OpTC soit resté pratiquement inconnu dans le milieu universitaire. En cinq ans depuis sa sortie, il a été utilisé par environ deux dizaines de publications seulement. Il est courant pour plusieurs d'entre eux d'utiliser une petite partie du jeu complet. En comparaison, il y a chaque année des dizaines ou même des centaines de publications qui utilisent les jeux de données traditionnels.

En outre, les évènements malicieux ne sont pas clairement étiquetés, ce qui permet aux auteurs de choisir la granularité de détection la plus adaptée à leur méthode. Cela peut entrainer des rapports de performance trompeurs. Le Tablau 2.1 met en évidence que la définition d'activité malveillante est différente dans toutes les publications. Par conséquent, les résultats rapportés par différents auteurs ne peuvent pas être directement comparés.

Les bons résultats obtenus dans les travaux antérieurs ne doivent pas décourager la nouvelle recherche. Le jeu de données OpTC a encore un grand potentiel pour supporter la conception des IDS innovantes et contribuer à la sécurité des réseaux réels.

Le reste de ce chapitre est consacré à l'analyse de certaines des propriétés du jeu de données OpTC qui peuvent entrainer des fuites de données et des résultats trop optimistes.

3.3 Les grands défis

Avant de se plonger dans l'étiquetage des données, il faut d'abord examiner certaines caractéristiques des données brutes, telles que la configuration du testbed, l'architecture réseau et la forme

des données. Ces caractéristiques éclaireront les difficultés associées au problème de détection d'intrusion, comme le montre le jeu de données OpTC. Comprendre ces éléments permettra de prendre des décisions judicieuses lors de la conception de l'IDS. De plus, le processus d'étiquetage et l'évaluation de référence utiliseront des éléments spécifiques du jeu de données (hôtes et utilisateurs) en les identifiant par leur nom. Par conséquent, il est essentiel de comprendre ce que ces termes signifient.

3.3.1 Organisation des données

Le jeu de données OpTC est créé par une équipe rouge qui simule des assauts dans un environnement virtuel. Il commence par une période d'activité initiale visant à caractériser l'état de référence de l'environnement simulé, suivie d'une période d'évaluation. La simulation dure 10 jours, les trois derniers jours étant réservés à l'attaque de l'équipe rouge.

Il est surprenant de constater que plusieurs sources ne s'accordent pas sur la durée totale du calendrier. Certaines mentionnent une période allant de 6 à 10 jours, incluant 2 ou 3 jours d'activité de l'équipe rouge. De plus, le nombre d'hôtes varie considérablement, allant de 500 à 1000. De plus, les informations officielles semblent entrer en contradiction avec les observations réalisées. Le résumé du projet révèle qu'une simulation de 1000 hôtes a été faite pendant deux semaines, mais que seules les données de télémesure de 500 d'entre eux ont été recueillies, soit la moitié du nombre total (FiveDirections, 2019).

Pour résoudre ces contradictions, il faut examiner les dates d'activité de chaque hôte dans les journaux à l'aide du champ « hostname ». La Figure 3.1 présente le nombre d'hôtes observés quotidiennement ainsi que le nombre total d'hôtes.

Un quart des hôtes ne transmettent aucune télémétrie. La population change constamment : moins de la moitié (475) d'entre eux envoient des journaux la moitié du temps. Seuls 223 hôtes, soit environ un cinquième, offrent une visibilité complète pour tous les 10 jours de l'engagement. Pendant les trois jours de la simulation de l'attaque par l'équipe rouge, il y a exactement 500 hôtes.

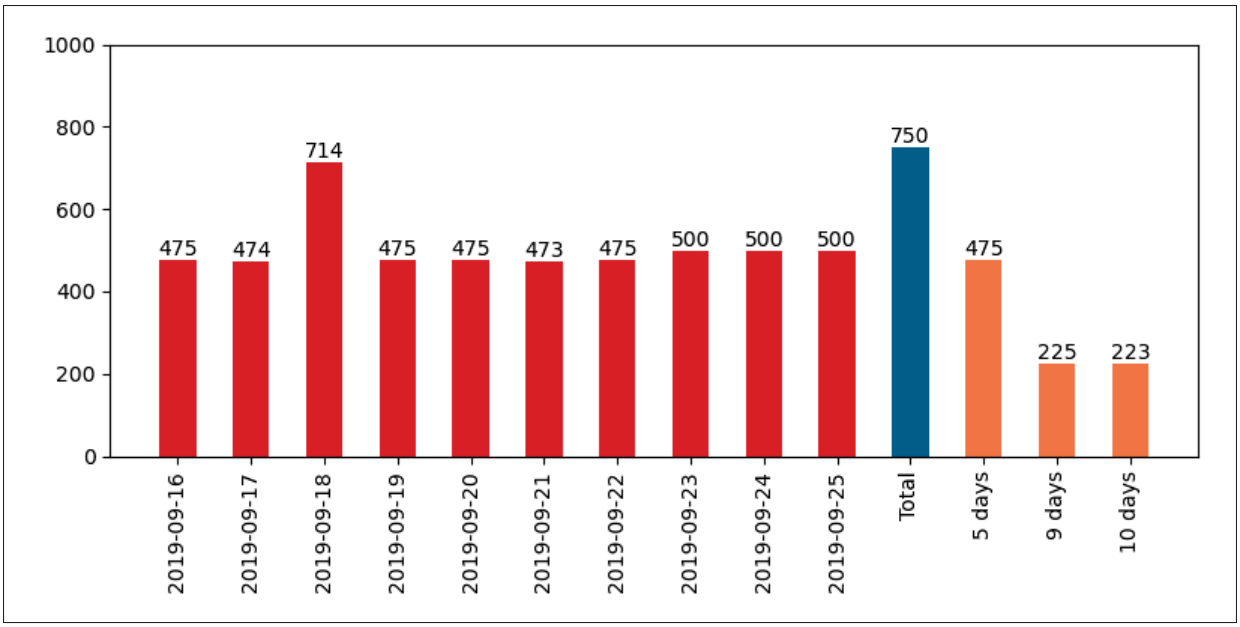

Figure 3.1 Le nombre d'hôtes observés quotidiennement ainsi que le nombre total d'hôtes

En raison de l'incomplétude des données, il n'est pas possible de reconstituer parfaitement la campagne malveillante, car certaines attaques peuvent commencer ou se terminer sur des hôtes manquants. Cela constitue une caractéristique précieuse pour tester les systèmes de détection d'intrusion, qui n'ont pas reçu l'attention nécessaire dans les études précédentes. Les milieux professionnels constituent des écosystèmes fluctuants et variés, extrêmement compliqués, qui, dans les faits, ne sont que partiellement équipés pour surveiller leur propre sécurité, bien qu'ils soient tenus de le faire (Parsons, 2023). Par conséquent, un système d'alerte d'intrusion solide doit être en mesure de détecter les attaques avec des données imparfaites. Le jeu de données OpTC peut servir à évaluer cette caractéristique des systèmes proposés.

3.3.2 Schéma des événements

L'un des buts originaux du programme TC consistait à créer un « modèle de données commun » pour représenter les évènements et les données de provenance. Ce sujet est développé plus en profondeur dans la proposition de design (Griffith *et coll.*, 2020). Cependant, les données

{"action":"CREATE","actorID":"b8ba3469-c7b9-4ff2-87e7-cf82d4b44699","hostname":"SysClient0004.systemia.com","id":"8da06444-b99b-44b4-bfa8-30cbd20930c8","object":"PROCESS","objectID":"c2f404a8-8464-4acc-9f5d-51717f2ff8a2","pid":3164,"ppid":3344,"principal":"NT AUTHORITY\\SYSTEM","properties":{"acuity_level":"1","command_line":"\\??\\C:\\Windows\\system32\\conhost.exe 0xffffffff -ForceV1","image_path":"\\Device\\HarddiskVolume1\\Windows\\system32\\conhost.exe","parent_image_path":"\\Device\\HarddiskVolume1\\Windows\\SYSTEM32\\cmd.exe","sid":"S-1-5-18","user":"NT AUTHORITY\\SYSTEM"},"tid":-1,"timestamp":"2019-09-24T05:32:01.344-04:00"}

Figure 3.2 Un évènement représentatif d'exécution de processus sous la forme eCAR d'un hôte dans l'ensemble de données OpTC

de télémétrie des hôtes du jeu de données OpTC sont fournies sous la forme eCAR, qui est une extension du modèle CAR de MITRE (MITRE, 2022). Les fichiers eCAR sont formatés en JSON (Bray, 2017), un format de données demi-structuré qui évite le besoin de parsage et d'extraction de champs.

Le jeu de données comprend non seulement des données de télémétrie des hôtes sous la forme eCAR, mais également un nombre limité de journaux provenant de l'outil de surveillance Bro (maintenant connu sous le nom de Zeek) (The Bro Project, 2018), comme le montre la Figure 3.3. Ce capteur fournit une couche supplémentaire de visibilité sur les attaques provenant de l'extérieur du réseau corporatif simulé. Ces journaux sont encodés dans un format différent et ne s'intègrent pas bien au scénario d'attaque qui fait l'objet de cette étude.

Le JSON a émergé comme format prédominant pour les outils de sécurité natifs des nuages (Kubernetes, 2023; AWS, 2024; Microsoft, 2023). Cependant, son utilisation présente ses propres défis. Un de ces obstacles est la structure dynamique du document, où un évènement ne possède pas nécessairement les mêmes champs qu'un évènement précédent. De plus, il existe une ambigüité sémantique, car le sens d'un champ peut fluctuer en fonction du type d'évènement.

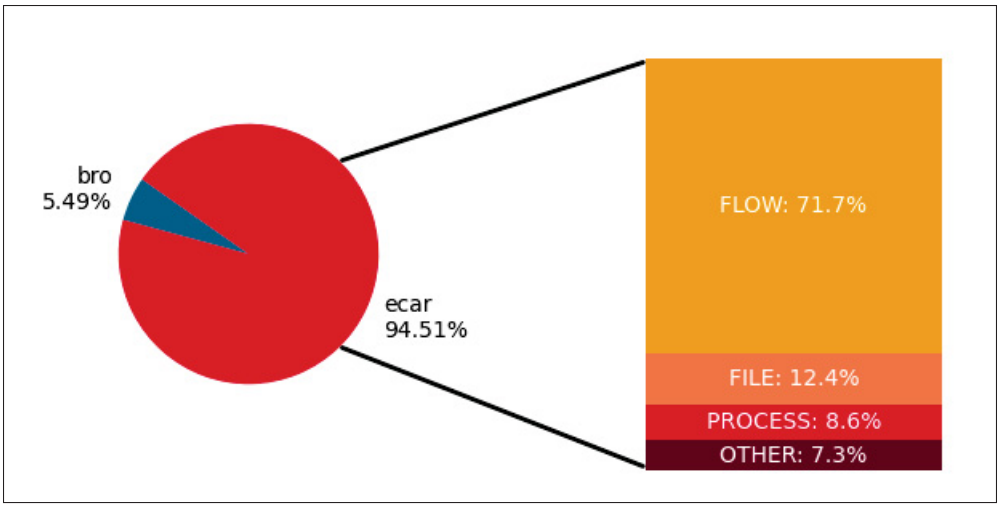

Figure 3.3 Distribution des catégories de journaux d'OpTC

Dans leur étude du jeu de données OpTC (Anjum *et coll.*, 2021), les auteurs décortiquent les fichiers de journaux en catégories et fournissent une explication détaillée des champs. Bien que deux types d'objets, soit « FLOW » et « FILE », représentent près de 84% des évènements dans l'ensemble de jeu OpTC, la revue de la littérature démontre que les études existantes accordent peu d'attention à ces deux classes, privilégiant les évènements liés à l'exécution des processus et à l'authentification.

Cette approche est justifiée. Dans un contexte professionnel, il y a un cout réel lié au stockage et à l'analyse des journaux. Par conséquent, les responsables des systèmes doivent déterminer les types de télémétrie qu'ils peuvent se permettre de collecter. D'après l'étude 2023 menée par le SANS (Crowley *et coll.*, 2023) auprès de plus de 600 spécialistes des centres opérationnels de sécurité (SOC), l'analyse NetFlow et l'analyse des paquets réseau se classent au dernier rang en termes de maturité technologique, parce que les ressources sont plutôt allouées à d'autres niveaux de la pile de sécurité.

En même temps, les organismes de règlementation accordent une priorité élevée et constante à la conservation des enregistrements d'authentification pour contrer les menaces de sécurité actuelles (CISA *et coll.*, 2024). Cela démontre que tous les enregistrements ne sont pas créés égaux en termes d'importance pour la sécurité. En se concentrant sur les évènements les plus

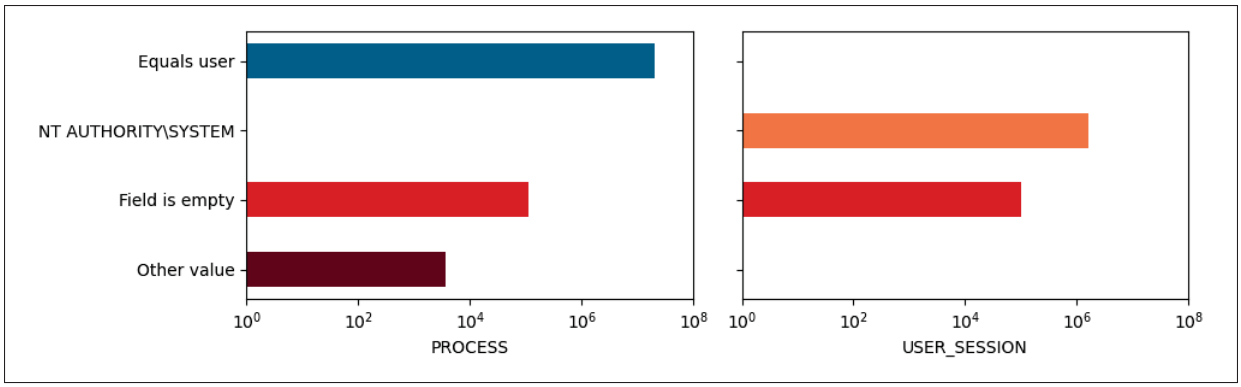

Figure 3.4 Comparaison des valeurs dans les champs « principal » et « properties.user » du modèle eCAR

significatifs, les concepteurs de systèmes IDS peuvent ainsi renforcer l'utilité et la pertinence de leurs outils auprès des partenaires industriels.

Chapitre 4 execute trois expériences avec les évènements de création de processus, d'authentification et de journaux PowerShell. Ces résultats montrent que ces évènements sont suffisants pour détecter toutes les attaques dans le jeu de données OpTC, ce qui permet de réduire considérablement sa taille, éliminant ainsi la nécessité de stockage couteux.

Le modèle eCAR comprend les deux champs « principal » et « properties.user » qui permettent de lier des évènements à une identité d'utilisateur spécifique. La différence précise entre ces deux champs n'est pas clairement établie, car la documentation du jeu de données ne fournit qu'une description du champ « principal ». Il semble être un champ de normalisation supplémentaire ajouté au-dessus du modèle MITRE CAR, et certains travaux antérieurs le mentionnent explicitement.

Figure 3.4 montre la comparaison entre les valeurs de ces deux champs. La signification du champ « principal » change en fonction du type d'évènement. Dans la plupart des évènements de création de processus, les deux champs contiennent des données identiques. Cependant, dans un petit pourcentage d'évènements, soit environ 0.5%, le champ « principal » est vide, tandis que, dans une fraction encore plus petite, les deux champs « principal » et « properties.user »

contiennent des informations différentes. L'analyse soigneuse de ces évènements suggère que la cause la plus probable est un bogue dans l'agent de collecte des journaux.

Par contre, dans le cas des évènements « USER_SESSION », ces deux champs ne seront jamais égaux. Le champ « principal » devrait être une valeur constante, et, dans certains cas rares, il peut également être vide (environ 6% des évènements).

Cela montre un problème très répandu dans les outils d'observabilité modernes, qui peuvent modifier la signification d'un champ en fonction du type d'évènement. C'est particulièrement important pour les IDS explicables : il ne suffit pas de détecter correctement une attaque, il faut s'assurer de ne pas attribuer à un mauvais utilisateur une attaque en se basant sur un champ incorrect. Sinon, cela risque de perturber l'enquête et faire en sorte que les analystes classent l'alerte comme étant fausse. Cette caractéristique des données fait qu'OpTC devient une ressource importante pour les recherches futures sur la conception de systèmes de détection d'intrusion explicables.

3.3.3 Inventaire des utilisateurs et des hôtes

Un aspect fondamental d'un programme de sécurité efficace est la tenue d'un inventaire actuel des actifs. (Kokulu *et coll.*, 2019; Knerler *et coll.*, 2022; Crowley *et coll.*, 2023) L'exécution d'un outil de configuration réseau peut signifier des choses très différentes, selon qu'elle soit effectuée sur un serveur ou sur un système de gestion des ressources humaines. Des centres d'opérations de sécurité matures peuvent avoir une équipe dédiée à la maintenance de listes de tous les serveurs, utilisateurs et applications. Ces inventaires d'actifs fournissent un contexte important lors des enquêtes sur les alertes.

Lorsque l'on travaille avec OpTC, cette information n'est pas disponible.

Néanmoins, il est possible d'obtenir cette information à un certain niveau à partir des journaux, comme c'été déjà fait pour la liste des hôtes.

Tableau 3.2 Reconstruction de l'inventaire des identités des utilisateurs

	Nombre d'utilisateurs	
Identités totales des utilisateurs	1006	100.0%
Utilisateurs du domaine Active Directory	752	74.8%
Comptes de service	232	23.0%
Comptes locaux	22	2.2%
Utilisateurs avec des droits d'administrateur	44	4.4%
Nombre moyen d'utilisateurs par hôte	2.85	

Le Tableau 3.2 présente un résumé de toutes les identités utilisateur pouvant être identifiées à partir des journaux d'exécution de commandes, incluant les comptes de service typiques dans un système d'exploitation Windows.

On attend de voir une identité distincte « employé » pour chaque hôte. Les données de télémétrie manquent pour environ 250 hôtes, ce qui correspond à un nombre équivalent d'identités manquantes dans le domaine Active Directory. Il y a un nombre considérable d'identités avec des privilèges élevés pour un environnement de cette taille. Le terme compte de service désigne ici une identité standard intégrée au système d'exploitation Windows, comme « NT AUTHORITY\SYSTEM ». Il peut aussi désigner des identités conçues pour l'automatisation et d'autres activités non humaines. Ces identités sont habituellement précédées d'une chaine de caractères fixes « svc_ » et elles sont très utiles pour surveiller la sécurité. Cependant, elles ne sont pas présentes dans l'ensemble de données OpTC, elles restent hors de cette analyse.

Le composant clé dans tout environnement Active Directory est le contrôleur de domaine, qui assure la garantie de l'intégrité de l'environnement et offre des services essentiels, tels que le DNS et l'authentification.

Les contrôleurs de domaine ont un profil de comportement distinctif et clairement identifiable, et aucun des hôtes du jeu de données OpTC ne correspond à ce modèle. Par conséquent, il est possible d'affirmer avec certitude que les contrôleurs de domaine de cet environnement ont été exclus de la surveillance et qu'ils n'ont pas partagé leurs données de télémétrie. Cette découverte

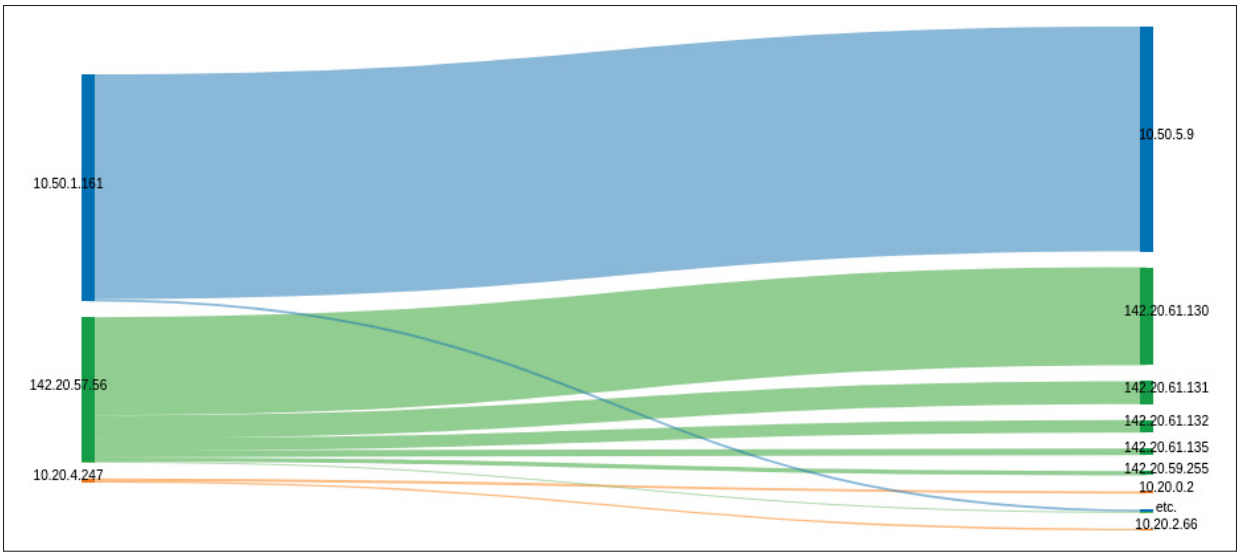

Figure 3.5 Trafic sortant de « SysClient0311.systemia.com » hôte le 25 septembre 2019 (dernier jour de l'exercice de l'équipe rouge), divisé par source et destination

ajoute une nouvelle complexité, puisqu'une partie de la campagne de l'équipe rouge a eu lieu sur l'un des contrôleurs de domaine.

Il est possible de découvrir les adresses des contrôleurs de domaine en examinant la destination du trafic habituel de l'Active Directory, comme LDAP. On compte alors deux serveurs sur le réseau, les adresses « 142.20.61.130 » and « 142.20.61.131 ». Malheureusement, il n'est pas possible d'associer des noms aux adresses IP correspondantes. Toutefois, il devrait être possible de détecter les mouvements latéraux vers ces serveurs en examinant les flux de données.

3.3.4 Inventaire des réseaux

La Figure 3.5 présente en détail les flux sortants de l'hôte « SysClient0311.systemia.com [1]» durant la dernière journée de la simulation. Il est observable que l'hôte possède trois adresses IP. La quantité la plus importante de trafic est destinée à l'adresse IP « 10.50.5.9 » (en bleu), qui est liée au processus de transfert de journal « lwabeat.exe ». Il est plausible de croire que ce flux

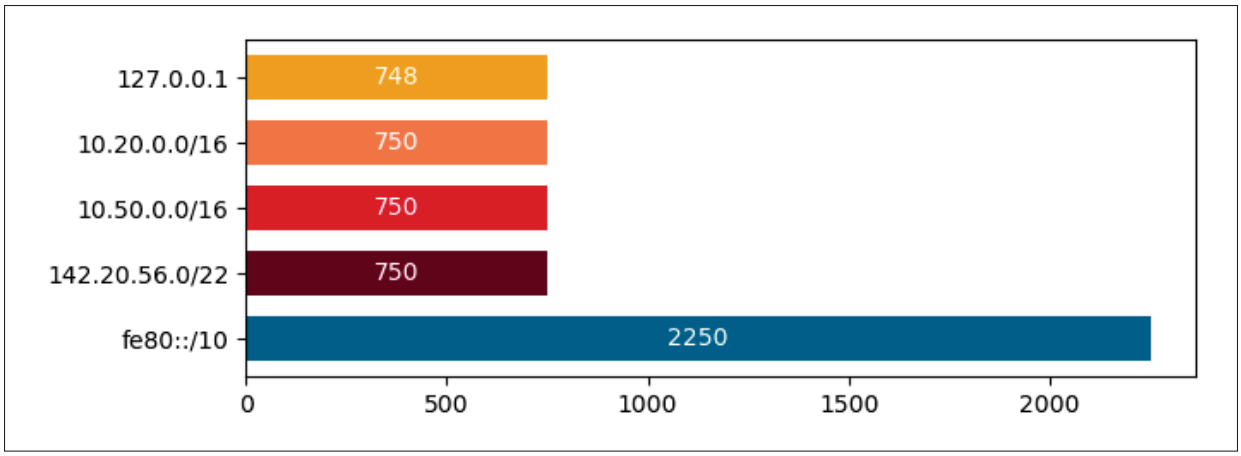

Figure 3.6 Distribution des adresses IP observées dans chaque sous-réseau

représente les journaux de l'environnement simulé lui-même et qu'il est donc sans importance. Les connexions multicast et les connexions de loopback ont été exclues de la visualisation.

La partie verte du graphique représente le réseau simulé, qui ne comporte que cinq flux réseau distincts. Deux d'entre eux correspondent à des adresses IP identifiées comme étant celles de contrôleurs de domaine, alors que les fonctions des deux autres serveurs sont inconnues. On peut en déduire certains détails au sujet de l'architecture du réseau. Les hôtes sous surveillance reçoivent une adresse dans réseau « 142.20.56.0/22 », tandis que les contrôleurs de domaine sont regroupés dans un sous-réseau distinct avec une adresse « 142.20.61.128/25 » distincte ou similaire.

Cet hôte ne reçoit aucun trafic entrant de la part de ses pairs du réseau pendant cette période (à l'exception du trafic multicast). Sur ce genre de réseau silencieux, tout trafic entre les hôtes serait très facile à repérer comme une anomalie.

Pour transformer correctement les journaux de flux de réseau en un graphe de connexions, il faut connaitre l'adresse IP correspondant à chaque hôte. Malheureusement, l'accès « live » à l'environnement n'est pas possible, ce qui empêche de consulter directement le serveur

[1] Il est fortement recommandé d'utilisation des noms d'hôte exacts, tels qu'ils apparaissent dans les données brutes, comme « SysClient0311.systemia.com », au lieu de les abréger, comme le font certaines œuvres précédentes, à « 311 ».

DNS. Toutefois, il est possible de reconstituer ces informations à partir des journaux bruts de connexion. La Figure 3.6 montre la répartition des adresses IP dans tous les sous-réseaux, incluant les adresses IP de loopback et les adresses IPv6 « link-local », qui sont typiques dans les réseaux Windows.

Le nombre d'adresses IP distinctes dans chaque sous-réseau correspond exactement au nombre total d'hôtes, comme le montre la Figure 3.1. Chaque interface possède une adresse IPv6 de type « link-local », et chaque hôte se trouve sur les trois sous-réseaux, ce qui confirme que la télémétrie complète du réseau est disponible.

3.3.5 Inventaire du logiciel

Le dernier élément d'un inventaire typique d'actifs informatiques est la liste des applications installées. Bien qu'il soit considéré comme un contrôle de sécurité important dans plusieurs normes et recommandations industrielles (CIS, 2021; Canadian Centre for Cyber Security, 2023b), sa mise en œuvre s'avère extrêmement difficile, à moins que l'environnement ne soit extrêmement règlementé. Les sociétés actuelles comptent sur une main-d'œuvre variée occupant divers postes, nécessitant des logiciels spécifiques pour mener à bien leurs missions. Cette situation complexifie considérablement la tâche du département TI, qui ne peut plus contrôler ni approuver chaque version de programme.

La plupart des réseaux d'entreprise sont exposés à plusieurs forces opposées qui affaiblissent considérablement l'efficacité de l'inventaire des logiciels en tant que mesure de sécurité.

- L'approche « bring your own device » (BYOD) a connu une accélération considérable avec la pandémie de COVID-19 et la généralisation du télétravail. La plupart des entreprises ne peuvent pas exercer un contrôle complet sur chaque ordinateur ni installer des outils de surveillance (Barlette, Jaouen et Baillette, 2021). Par conséquent, il est impossible de mettre en place un IDS exigeant la visibilité complète de tous les ordinateurs du réseau dans ce contexte.

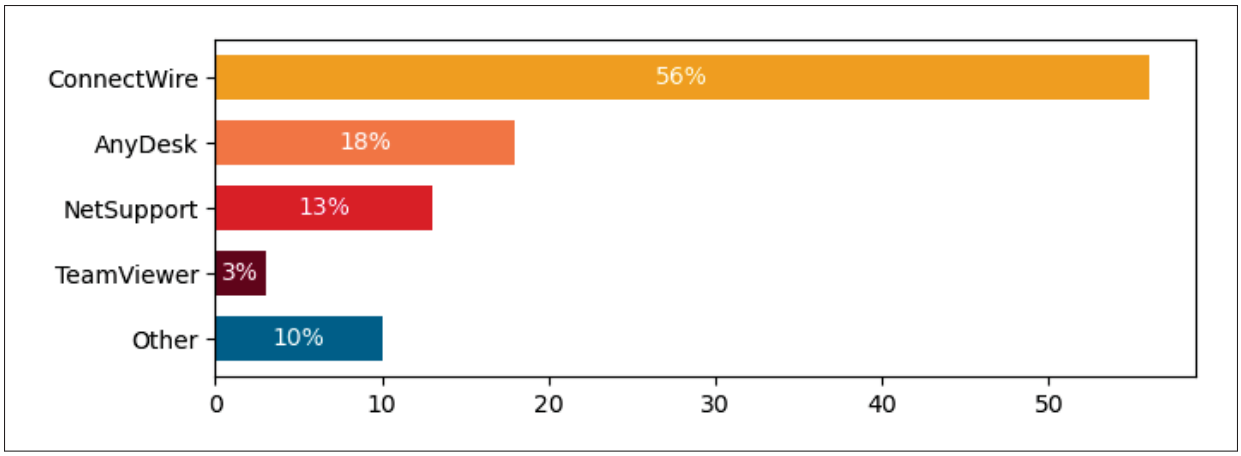

Figure 3.7 Au moins 14% des incidents enregistrés en 2023 ont nécessité l'utilisation d'outils de gestion à distance. Dans 75% de ces cas, il n'y avait pas de logiciel malveillant.
Tirée de Huntress (2023)

- « Living off the Land » (LOTL) est une technique d'exploitation qui consiste à utiliser des composants logiciels préexistants dans un système pour mener des actions malveillantes, au lieu d'utiliser un maliciel. Cette technique d'attaque, dont l'efficacité était connue depuis longtemps, a récemment commencé à gagner en popularité (CISA *et coll.*, 2024).
- L'utilisation malveillante des outils de gestion à distance a représenté un défi majeur pour les équipes SOC et a contribué à plusieurs incidents de sécurité de grande notoriété en 2023 (Huntress, 2023; CrowdStrike, 2023; FBI et CISA, 2023; CrowdStrike, 2024).

Dans le contexte actuel de la cybersécurité, il est préférable de ne pas présumer que les applications sont fiables. Par conséquent, les systèmes de détection d'intrusion qui ne s'appuient que sur la détection des anomalies plutôt que sur l'expertise humaine sont plus à même de repérer de nouvelles méthodes d'attaque. Par contre, la liste des actifs logiciels peut servir à élaborer des systèmes d'intelligence artificielle explicables. Ce répertoire permet aux analystes en sécurité de mesurer l'écart entre la norme attendue de l'environnement et le niveau des anomalies détectées par rapport à cette norme.

Cependant, le sujet principal de cette recherche est de savoir si le jeu de données OpTC possède toutes les qualités d'un réseau d'entreprise moderne. La prochaine section, va aborder brièvement la méthode d'étiqueter les évènements bruts à l'aide de la description de l'activité de l'équipe rouge. Ensuite, une analyse approfondie des données du jeu et des activités malveillantes sera effectuée.

3.4 Stratégies d'étiquetage

Le jeu de données OpTC n'est pas étiqueté de manière conventionnelle. En effet, l'équipe rouge a fourni un journal détaillant les principales phases de l'attaque et identifiant les actifs clés touchés. Plusieurs raisons expliquent ce choix.

- Comme déjà discuté, le jeu de données OpTC est extrêmement volumineux et il serait irréaliste de tenter d'étiqueter 17 milliards d'évènements manuellement.
- La méthode d'engagement exige que l'équipe rouge fonctionne en isolement, c'est-à-dire qu'elle ne reçoit aucune information sur l'état du système cible. Cette mesure vise à éviter les biais dans leur attaque. Toutefois, la perception de l'équipe rouge peut s'écarter significativement de la réalité des journaux bruts.
- Fournir un rapport de ce type à la fin de l'exercice est devenu une pratique courante offerte par toutes les équipes rouges professionnelles. Généralement, une réunion est organisée avec l'équipe de sécurité afin que celle-ci examine le rapport avant de commencer à analyser les journaux et les alertes d'IDS. Fournir la vérité sur le terrain dans ce format génère un ensemble de données plus authentique.

Malheureusement, puisque les évènements ne sont pas étiquetés séparément, la plupart des méthodes d'apprentissage automatique ne peuvent pas être utilisées avec des données brutes. Par conséquent, les chercheurs ont de la difficulté à évaluer les performances de leurs systèmes de détection d'intrusion proposés.

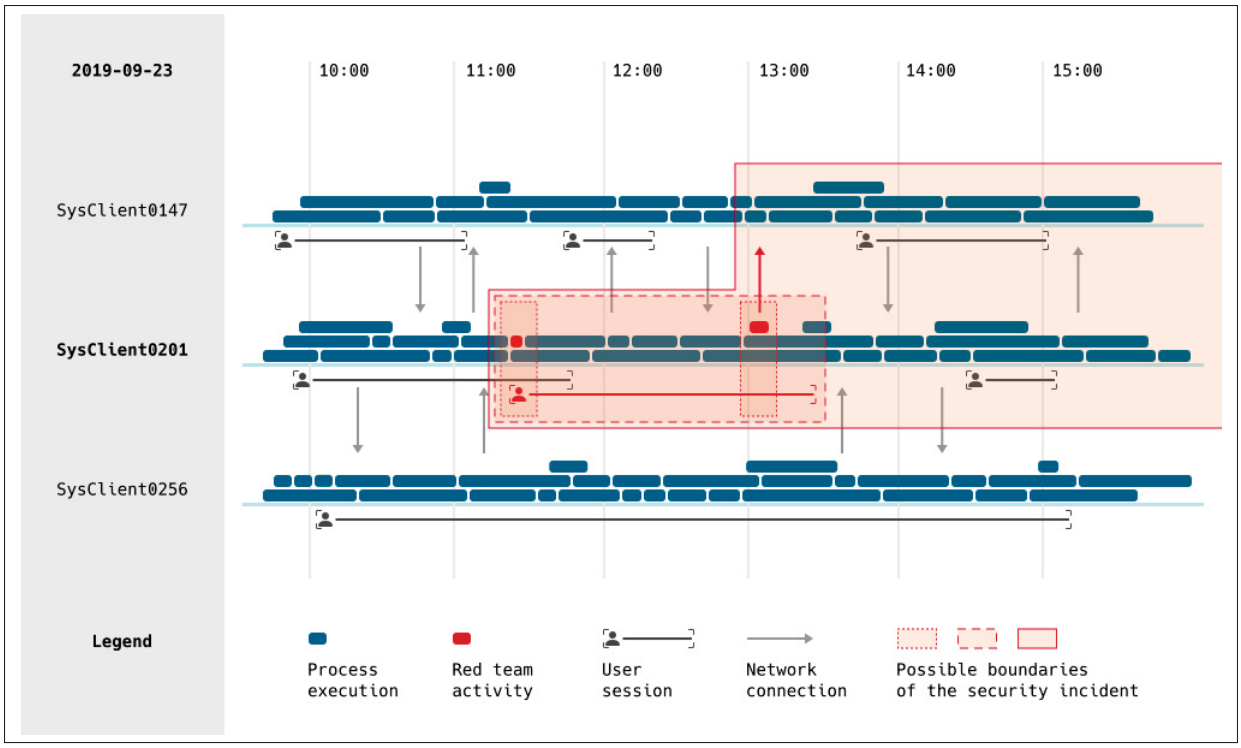

Figure 3.8 La représentation schématique des attaques de l'équipe rouge provenant de l'hôte « SysClient0201.systemia.com » le 2019-09-23 montre des façons possibles d'associer les actions de l'équipe rouge aux étiquettes malveillantes

Le Tableau 2.1 met en évidence l'inconsistance des travaux antérieurs concernant la méthode d'étiquetage. Malgré l'utilisation du même jeu de données, leurs résultats ne peuvent pas être directement comparés. De plus, la plupart des auteurs négligent le processus d'étiquetage dans leur méthodologie, alors que c'est une étape cruciale pour la mise en œuvre d'un IDS.

Il existe une certaine flexibilité dans l'alignement des activités de l'équipe rouge avec les journaux bruts. La Figure 3.8 montre un exemple tiré d'une des attaques du jeu de données. Selon la manière dont un expert interprète les journaux, il peut attribuer n'importe quel nombre d'étiquettes malveillantes, allant de 1 à plusieurs centaines, au même ensemble d'évènements.

Le journal décrit une session interactive sur l'hôte cible « SysClient0201.systemia.com », initiée par un compte « compromis », suivie de l'exécution de plusieurs processus malveillants.

Dépendamment des besoins de sécurité, on a la liberté de choisir l'une ou l'autre des options suivantes :

1. Associer l'étiquette malveillante uniquement à l'exécution du maliciel ;
2. Corréler la session utilisateur interactive avec tous ses processus enfants ;
3. Corréler toutes les activités de l'hôte compromis pendant l'attaque ;
4. Associer une seule étiquette malveillante au compte utilisateur compromis pour toute la durée du timeline ;
5. Étiqueter l'hôte comme étant infecté à partir du moment de l'infection du maliciel jusqu'à ce qu'un spécialiste en sécurité informatique le supprime plus tard ;
6. Facultativement, élargir l'étendue de l'incident pour inclure tous les hôtes ayant échangé des données avec les processus malveillants par le réseau.

Toutes ces méthodes d'étiquetage, ainsi que bien d'autres, sont valides et sont effectivement utilisées par les spécialistes en sécurité lors des enquêtes sur les incidents. Toutefois, elles génèrent des jeux de données d'apprentissage fort différents.

Les travaux précédents sur l'ensemble de données OpTC ont souvent opté pour des méthodes d'étiquetage qui se conforment aux exigences des étapes ultérieures du pipeline. Par exemple, Kairos (Cheng *et coll.*, 2024) représente une approche de type (3.), tandis que l'IDS fondé sur les techniques NLP (Golczynski et Emanuello, 2021) utilise une méthode d'apprentissage non supervisé et correspond le mieux à la stratégie (5.).

Le choix de la granularité de détection (utilisateur, hôte, processus, évènement, intervalle de temps, etc.) aura un impact direct sur l'architecture du pipeline d'entrainement et l'information contenue dans l'alerte reçue par l'analyste de sécurité. Il existe un compromis entre la sensibilité de l'IDS et les efforts requis pour mener l'investigation de l'alerte. Ceci est l'un des facteurs importants dans l'évaluation industrielle (Sommer et Paxson, 2010; McRee, 2022), qui est directement déterminé par le type et la qualité des étiquettes.

De nombreuses recherches sur les IDS ont bénéficié de l'utilisation de jeux de données étiquetées de manière traditionnelle pour créer des systèmes d'IDS sophistiqués, mais totalement irréalisables. Puisque l'étiquetage n'est pas considéré comme une étape intégrale dans la chaine d'entrainement d'un système IDS, il incombe aux entreprises d'examiner comment adapter leurs réseaux pour satisfaire les exigences d'un système IDS en particulier, plutôt qu'évaluer les modifications à l'IDS pour répondre aux exigences d'entreprise.

L'utilisation du jeu de données OpTC implique automatiquement que l'étape d'étiquetage devienne une nécessité explicite, ce qui est bénéfique pour la conception de systèmes IDS plus pratiques. Pour créer un système IDS efficace et adapté, il est impératif de déterminer avec précision l'objectif de la détection, de sélectionner un niveau de granularité de détection approprié et de structurer le processus d'étiquetage en conséquence.

Il est vrai que, puisque les chercheurs peuvent étiqueter les ensembles de données à leur convenance, cela entraine un risque de fuite de données, ce qui permet aux modèles d'atteindre des résultats irréalistes. Selon l'étude de la systématisation des connaissances récente (Apruzzese *et coll.*, 2023b), la majorité des publications actuelles ne rapportent qu'un seul résultat d'évaluation. En l'absence de cas d'utilisation spécifiques liés à des exigences de sécurité claires, les auteurs du système IDS proposé peuvent modifier leur processus d'étiquetage ou adopter une granularité de détection différente pour servir leurs objectifs d'évaluation.

Considérons une interprétation extrême de la réalité du terrain pour les données du jeu de données OpTC. L'équipe rouge a lancé plusieurs actions en une seule attaque simulée, toutes avec le même objectif final. On peut envisager un système IDS dont la mission est de déterminer si une attaque est en cours pendant une période donnée. Pour y parvenir, il est possible d'entrainer un classificateur binaire simple qui attribue une seule étiquette YES/NO à l'ensemble du jeu de données. Évaluer ce modèle une seule fois pendant la période d'engagement entrainerait une cote de performance parfaite, mais le modèle obtenu serait totalement inutile dans la pratique.

Comme il n'y a pas de contraintes imposées sur les étiquettes dans le jeu de données OpTC, il faut considérer avec prudence les indicateurs de performance rapportés.

3.5 Réalité du terrain incomplète

C'est un problème bien connu que tout système de détection d'intrusion par anomalie doit supposer des données d'entrainement propres, sans activité malveillante (McHugh, 2000), mais, dans la réalité, cela n'est pas toujours possible. Les rapports de menaces indiquent que les attaquants peuvent rester dans un réseau pendant 90 à 200 jours après la première intrusion sans détection (Mandiant, 2023). Par conséquent, le réseau peut être bien saturé d'activité malveillante. Un modèle d'apprentissage machine peut apprendre à classer de tels comportements comme étant « normaux » et à ne pas reconnaitre de nouvelles attaques. Malheureusement, il n'y a pas de solution simple à ce problème, donc on le tolère généralement, tant dans la pratique que dans la recherche. Pourtant, la présence de comportements malveillants inconnus est cruciale pour évaluer les performances du système IDS proposé. Si le système peut détecter les attaques documentées, mais, en même temps, il n'arrive pas à détecter des activités malveillantes très similaires, mais non documentées, cela indique clairement un surapprentissage du modèle.

Les exercices de l'équipe rouge sont axés sur la recherche et la mise en évidence des faiblesses dans les contrôles de sécurité, avec l'objectif ultime d'améliorer la sécurité du réseau. Pour assurer la plus grande valeur pour le client, la pratique courante parmi les équipes rouges professionnelles consiste à réduire le rapport pour et uniquement les informations les plus pertinentes et actionnables (NetSPI, 2021). Par conséquent, un rapport typique d'une équipe rouge pourrait omettre des éléments relatifs aux actions offensives, comme :

- l'énumération et autres activités de reconnaissance qui permettent à l'équipe rouge de comprendre l'environnement, mais qui ne constituent pas une attaque ;
- les attaques échouées que les mesures de sécurité efficaces ont empêchées ;
- les attaques réussies, mais mal interprétées par l'équipe rouge.

La Figure 3.9 illustre l'impact de l'omission de cette information. Chaque étape du processus d'étiquetage ajoute des erreurs additionnelles, et la performance du modèle peut s'écarter significativement de la réalité. Dans cet exemple fictif, la précision est rapportée à 10 points et le

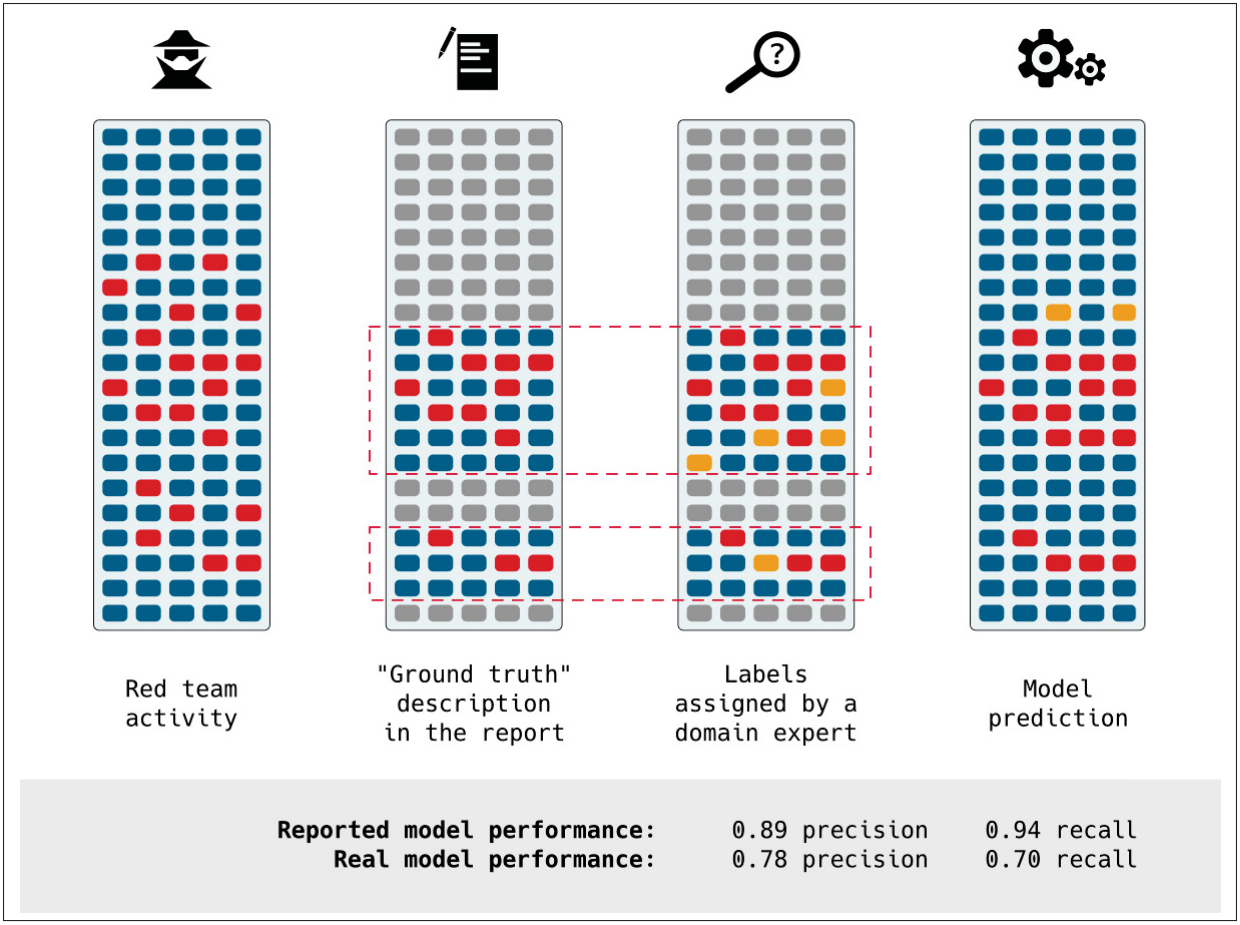

Figure 3.9 Une représentation schématique de l'effet d'une description incomplète de la réalité du terrain

rappel à 25 points au-dessus du rendement réel du modèle. De plus, le modèle a correctement identifié deux évènements malveillants qui ne figurent pas dans les données étiquetées, mais qu'il a classés comme de faux positifs.

Du point de vue de l'équipe rouge, il est justifié de ne pas mentionner ces éléments dans le rapport, car ils n'aident pas à atteindre l'objectif ultime. Cependant, d'un point de vue d'analyste en sécurité, tous ces évènements représentent toujours un vrai positif. Ces informations sont cruciales pour étiqueter correctement les journaux, assurant ainsi que ces actions seront identifiées comme une attaque dans une tentative d'intrusion future.

Il est raisonnable de supposer que l'ensemble de données OpTC possède cette propriété et omet certaines instances d'activités malveillantes de la documentation, puisque les attaques sont orchestrées par une équipe rouge expérimentée qui suit une procédure typique de l'industrie. Par conséquent, il faut admettre qu'un étiquetage simpliste, basé uniquement sur la définition de la « réalité du terrain », entrainera probablement des erreurs dans la classification de certaines activités malveillantes. En conclusion, il est impossible de mesurer de manière fiable les indicateurs de performance typiques, tels que la précision et le rappel, car certains vrais positifs peuvent être incorrectement classés comme de faux positifs.

3.6 Processus d'étiquetage contribué

La section suivante détaillera la méthode pour transformer le document de terrain en un ensemble d'étiquettes pour les évènements bruts du jeu de données. Considérant les caractéristiques de ce document et les difficultés abordées précédemment, l'objectif est de concevoir une méthode d'étiquetage capable de gérer l'éventualité d'une vérité partielle.

Les étiquettes générées sont adaptées à l'utilisation avec les techniques d'apprentissage machine. À cet égard, elles sont similaires aux autres jeux de données de référence, ce qui signifie qu'elles peuvent être utilisées pour entrainer et évaluer les modèles. C'est important de noter que le processus d'étiquetage sert à convertir entre les deux représentations de la réalité du terrain. Donc, il est nécessaire de l'exécuter une seule fois, séparément du pipeline d'entrainement, qui sera discuté au Chapitre 5.

3.6.1 Motivations

Chaque action effectuée par l'équipe rouge doit avoir un ou plusieurs évènements associés dans les journaux, et chaque découverte potentielle doit être soit reconnue comme étant liée à la réalité de terrain, soit rejetée comme un évènement inoffensif, avec une justification. Comme chaque nouvelle preuve accroit l'empreinte d'une attaque en la reliant à d'autres évènements, il

est nécessaire de continuer à classer ces éléments jusqu'à ce qu'il ne reste plus de nouvelles informations.

Ce processus aboutit à un algorithme de recherche exhaustif et déterministe qui simule un processus de collecte de preuves par un expert en criminalistique numérique. En partant du principe que les euristiques de recherche sont correctes, l'algorithme garantit la récupération de l'intégralité de l'attaque en passant d'un indicateur à l'autre.

3.6.2 Euristiques de recherche

Cette procédure d'étiquetage est appuyé sur le cadre proposé par le système de détection en temps réel d'APT HOLMES (Milajerdi *et coll.*, 2019), qui distingue les dépendances fortes et faibles entre les entités. Il organise les graphiques de provenance autour des exécutions de processus : pour mener une action malveillante sur un système informatique, il est nécessaire d'exécuter un certain code, qui correspond toujours à la création d'un processus. Les trois dépendances fortes sont les suivantes :

- toute entrée de journal générée par un processus malveillant est elle-même malveillante ;
- tout processus qui interagit avec une adresse IP suspecte est étiqueté comme malveillant ;
- tout processus enfant d'un parent malveillant est également étiqueté comme malveillant.

La seule dépendance faible est la suivante : un processus parent d'un processus enfant malveillant peut lui-même être malveillant. Pour résoudre les dépendances faibles, une analyse experte par un humain est requise. Ce cadre théorique exige que toutes les étiquettes soient entièrement justifiées et vérifiées, donc l'entrée en ligne du matériel expert est également étiquetée.

3.6.3 Algorithme de recherche

Une méthode détaillée dans l'Algorithme 3.1 permet d'attribuer des étiquettes aux évènements. À chaque itération sur l'ensemble des données, elle attribue des étiquettes aux évènements en attente, puis met à jour la file avec tous les nouveaux évènements connexes.

Algorithme 3.1 Une recherche exhaustive des évènements pour étiquetage

Input : L'ensemble initial des processus malicieux S_p, des adresses IP malicieuses S_i, et des sessions interactives malicieuses S_u.
Output : L'ensemble des processus étiquetés.

Let $\mathbb{P}$ be a set of all process creation events.
Let $\eta(p)$ be a list of IP address from network `FLOW` events of process p.
$L \leftarrow \{\}, Q_p \leftarrow S_p$
for *i in* S_i **do**
    Q_p += $\{\forall\ p \in \mathbb{P} : p$ communicates to $i\}$
end for
for *u in* S_u **do**
    Q_p += $\{p \in \mathbb{P} : p$ initiates user session $u\}$
end for
while Q_p *is not empty* **do**
    $p \leftarrow pop(Q_p)$
    L += $malicious(p)$
    Discover the parent process and look up the label provided by an expert.
    $q \leftarrow$ parent process of p
    if q *is labeled as malicious* **then**
        Q_p += q
    end if
    Unconditionally propagate the malicious label to all child processes :
    Q_p += $\{\forall\ t \in \mathbb{P} : r$ is a child process of $p\}$
    Discover all other processes that communicate to the same IP addresses as the labeled malicious process :
    for *i in* $\eta(p)$ **do**
        Q_p += $\{\forall\ t \in \mathbb{P} : t$ communicates to $i\}$
    end for
end while

L'exécution de l'algorithme est initialisée avec une liste d'étapes et des IOCs (indicators of compromise) extraits de la réalité du terrain par un analyste expérimenté en cybersécurité. La liste est composée de PID (Process ID) du processus malveillant, de la date et de l'heure d'évènement et de l'adresse IP d'une connexion réseau. Elle est accompagnée d'un commentaire expliquant pourquoi l'évènement est inclus dans la liste.

Comme l'algorithme d'étiquetage effectue une recherche exhaustive et déterministe, les examinateurs indépendants peuvent facilement vérifier l'exactitude et l'exhaustivité des étiquettes générées.

3.6.4 L'implémentation

La chaine complète d'étiquetage est implémentée en langage Go, qui permet de créer une application empaquetée dans un seul binaire. Go satisfait directement à plusieurs critères essentiels.

- Compte tenu de l'immensité du jeu de données OpTC, il faut utiliser un langage de programmation de niveau bas ayant une faible empreinte mémoire. Cela empêche l'utilisation de Python, qui demeure la solution de choix pour l'analyse de données en raison de son utilisation intensive de la mémoire.
- Pour traiter les évènements JSON du jeu de données OpTC, il faut d'abord les désérialiser et les convertir en représentations intermédiaires qui peuvent être facilement agrégées et corrélées avec des données externes injectées à partir de la description de la réalité du terrain. Par conséquent, l'implémentation doit gérer un très grand nombre d'objets de courte durée. Pour prévenir tout risque de fuites de mémoire, il est recommandé d'utiliser un langage avec ramasse-miettes, plutôt que pour un langage non sécurisé comme le C ou le C++.
- Le modèle de concurrence sans blocage de Go-routines minimise le temps de latence des entrées-sorties disque et exécute simultanément plusieurs étapes du pipeline.
- Le support natif à plusieurs systèmes d'exploitation offert par Go est également avantageux pour les environnements d'entreprise, qui sont l'objet de ce travail. Les environnements règlementés, tels que le SOC, utilisent souvent une combinaison de systèmes d'exploitation et ont des procédures d'approbation complexes pour tout logiciel. Le déploiement d'applications en un seul fichier binaire, sans avoir à gérer les dépendances externes ni la chaine d'approvisionnement, est donc très pratique.

Pour faciliter l'étiquetage interactif des évènements par un analyste de sécurité, le programme inclut une interface utilisateur en mode texte regroupant tous les contextes disponibles sur l'évènement étiqueté. Cette interface est une option judicieuse, car elle minimise les dépendances de compilation et parce que toutes les données sont déjà sous forme de texte : les journaux bruts et la réalité du terrain.

L'interface utilisateur fournit à l'analyste les informations suivantes :

- la queue pour étiqueter les évènements en attente ;
- l'arbre des processus reconstitué contient la ligne de commande et les arguments du processus en question, ainsi que ceux de ses parents, enfants et autres processus reliés ;
- les étiquettes actuellement assignées peuvent être directement attribuées par l'analyste à l'évènement en cours, à d'autres évènements du même processus étiqueté, ou être héritées d'autres processus ;
- l'entrée du journal correspondante de l'équipe rouge ;
- toutes annotations supplémentaires.

L'annexe II illustre les captures d'écran de l'interface utilisateur globale ainsi que celles des divers panneaux d'étiquetage.

3.6.5 Le résultat final

Le résultat est une liste augmentée d'étiquettes qui ne se limite pas à indiquer si un évènement est malveillant ou non. Elle permet de distinguer les évènements explicitement mentionnés dans le document de la réalité du terrain, ainsi que ceux qui sont liés à travers de l'arborescence des processus, comblant ainsi les éventuels manques de description.

Pour prendre en compte toutes les combinaisons possibles de l'origine de l'évènement, chaque évènement brut est étiqueté avec quatre catégories ci-dessous :

- **résultat** : bénin / malveillant ;

- **source** : réalité du terrain / anomalie majeure / corrélation / invalide ;
- **acteur** : équipe rouge / administrateur ;
- **chronologie** : avant la campagne de l'équipe rouge / jour 1 / jour 2 / jour 3.

En ne réalisant que quelques tours de boucle, il est possible d'identifier un grand nombre d'évènements et mettre au jour des étapes d'attaque qui n'étaient pas mentionnées dans le document de référence. L'Annexe I montre comment cela a permis d'identifier les actions de l'équipe rouge pendant la deuxième journée de la campagne.

De plus, puisque la recherche n'est pas limitée à la période de la campagne de l'équipe rouge, il est également possible d'attribuer des étiquettes aux anomalies significatives identifiées par les modèles de base non supervisés dans le Chapter 4. Ces étiquettes peuvent servir à évaluer la capacité du système IDS à différencier les attaquants des anomalies bénignes, qui sont la principale cause de fausses alertes dans le monde réel.

Pour vérifier que les étiquettes sont correctes et exhaustives, il est nécessaire de revoir seulement une courte liste initiale d'évènements tirés du document de référence, ainsi que quelques décisions prises par un expert humain. Cette méthode prend considérablement moins de temps que de passer en revue l'ensemble des étiquettes. Dans l'éventualité où l'une de ces entrées est établie comme étant erronée, il suffit de la corriger et de lancer une nouvelle recherche exhaustive pour produire une version corrigée des étiquettes.

3.7 Conclusion

Ce chapitre a présenté le jeu de données OpTC et expliqué ces avantages. Il a abordé l'impact négatif de l'insuffisance de jeux de données de haute qualité publiques pour la recherche en IDS. Ensuite, une analyse de certaines caractéristiques des données brutes du jeu a suggéré qu'elles peuvent être utilisées pour évaluer de manière réaliste les performances des systèmes IDS par rapport à des indicateurs clés pertinents pour les milieux professionnels.

La représentation de la vérité du terrain des attaques de l'équipe rouge peut poser des défis pour les utilisateurs potentiels du jeu de données. Le manque d'étiquettes pour les évènements individuels, la vérité de terrain incomplète et la nécessité d'une expertise en cybersécurité sont des obstacles majeurs qui entrainent des métriques d'évaluation incompatibles et une mauvaise répétabilité de la recherche. Sur la base de cette compréhension, un processus d'étiquetage déterministe a été proposé. Il a permis de convertir la narration de la campagne de l'équipe rouge en une série d'étiquettes d'évènements, qu'on peut utiliser avec des méthodes d'apprentissage automatique classiques.

Le chapitre suivant va mener une série d'expériences pour explorer et documenter la représentation interne des données et les traces d'activités malveillantes dans le but de rendre le jeu de données plus accessible pour évaluer les systèmes de détection d'intrusion de la prochaine génération.

CHAPITRE 4

L'ÉVALUATION DE RÉFÉRENCE DU JEU DE DONNÉES OPTC

4.1 Introduction

Ce chapitre va mener une série d'expériences pour détecter l'activité malveillante dans ce jeu de données, la séparer de l'activité bénigne et mesurer le comportement de base de l'environnement. Cette analyse sera faite en adoptant les mêmes stratégies qu'un analyste en cybersécurité lorsqu'il enquête sur un incident. Pour combler les failles systématiques introduites dans le chapitre précédent et poser les bases du concept d'IDS, il est nécessaire de répondre aux questions de recherche ci-dessous.

- QR1 : Est-ce que le jeu de données OpTC est une simulation réaliste d'une entreprise moderne ?
- QR2 : Est-ce que le comportement de base peut être caractérisé comme étant confiné ou ouvert ?
- QR3 : Qu'est-ce qui empêche la communauté scientifique de l'utiliser davantage ?
- QR4 : Comment les étiquettes sont-elles présentées dans les données ?

4.2 Méthodologie

Comme cela a déjà été discuté, les auteurs de l'OpTC ont fourni un document « OpTCRedTeamGroundTruth.pdf » pour décrire les attaques de l'équipe route. Cependant, l'analyse présentée dans ce chapitre ne repose pas sur cette réalité de terrain. Cette approche poursuit trois objectifs principaux :

- éliminer les préjugés découlant du décalage entre la perspective de l'équipe rouge et l'état réel du système, comme abordé dans la Section 3.5 ;
- concevoir une expérience plus authentique dans laquelle l'apparition ou le type d'attaques restent inconnus pour un analyste de sécurité. En réalité, tel document de la réalité de terrain

n'existe pas, donc les équipes de sécurité doivent recourir à diverses méthodes de chasse aux menaces, y compris une étape similaire d'analyse de base (Maxam et Davis, 2024) ;
- assurer que l'analyse des journaux est définie comme une étape explicite et requise de la conception de l'IDS, ce qui facilitera sa mise en œuvre dans un environnement industriel.

Au lieu de commencer par des évènements malveillants connus et de les analyser « top-down », c'est mieux de procéder de façon « bottom-up » : mesurer les diverses propriétés des données brutes, séparer les évènements anormaux des activités normales du réseau et les caractériser comme malicieux ou non.

Une fois l'analyse terminée, la description de la réalité de terrain peut être comparée aux résultats. Durant l'analyse, il est permis de connaitre seulement les dates de début et de fin de l'exercice de l'équipe rouge, et seulement pour analyser des agrégats statistiques. Cela est nécessaire pour documenter différentes caractéristiques de la distribution des données et répondre aux questions de recherche.

L'analyse se concentrera sur les types d'évènements suivants, inspirés des rapports sur l'actuel paysage des menaces et des données réellement disponibles dans le jeu de données OpTC :

1. Les évènements d'authentification des utilisateurs ;
2. L'évènement d'exécution des processus ;
3. Transcriptions des scripts PowerShell.

Concrètement, les informations suivantes sont visées :

- Est-ce que l'activité des acteurs dans le jeu OpTC correspond assez à celle d'un réseau d'entreprise typique ?
- L'activité malveillante dans OpTC représente-t-elle un fort signal qu'il est facile de distinguer de l'activité de base ?
- Est-il possible de détecter l'activité malveillante en utilisant une partie des données ?

- Est-ce que la réalité de terrain inclut toutes les informations sur l'activité malicieuse dans le jeu de données ?
- Quel est le processus approprié pour étiqueter les évènements malveillants ?
- Quel est le niveau de granularité de détection le plus approprié pour générer des alertes ?

4.3 Expérience 1 : les évènements d'authentification des utilisateurs

Le but de cette expérience est d'analyser le comportement typique des utilisateurs dans l'environnement simulé, celui représenté par l'objet de journal « USER_SESSION ».

Les rapports récents sur les menaces indiquent que les acteurs malveillants exécutent de plus en plus leurs attaques sans maliciel, en utilisant des outils légitimes et des identités compromises à la place (CrowdStrike, 2024). De leur côté, les organismes poursuivent leur migration vers les services infonuagiques, ce qui entraine une baisse de l'utilisation des méthodes traditionnelles de monitorage d'activité sur hôte. Ce qui signifie que les systèmes de détection d'intrusion modernes doivent s'adapter à travailler avec différentes sources de télémétrie.

Les évènements d'authentification, c'est un type de journal très courant, disponible dans presque tous les produits et services. Il est toujours recommandé de s'assurer de la collecte de ces évènements importants (Treasury Board of Canada Secretariat, 2020; CISA *et coll.*, 2024). Ils sont donc très utiles et importants en matière de détection d'intrusion.

4.3.1 Description de l'environnement

Dans le jeu OpTC, l'objet « USER_SESSION » représente une simulation d'une session interactive sur un hôte, ainsi que les sessions à distance. Le type exact de la session peut être déterminé à l'aide du champ « action ». Il y a sept actions différentes associées à ce champ, comme le montre le Tableau 4.1.

Comme c'était déjà établi, l'environnement compte 752 utilisateurs du domaine et fonctionne sur la technologie Active Directory, ce qui est typique dans un environnement moderne. Supposons

Tableau 4.1 Les actions de l'objet USER_SESSION

Action	Nombre total	Hôtes distincts	Utilisateurs distincts
GRANT	692,252	750	160
INTERACTIVE	84,777	746	993
LOGIN	555,077	750	3
LOGOUT	231,517	750	994
RDP	8	4	1
REMOTE	154,580	750	13
UNLOCK	20,778	745	746

Tableau 4.2 Nombre d'hôtes associés aux comptes d'utilisateurs

Nom d'utilisateurs	Nombre d'hôtes
`systemiacom\sysadmin`	706
`systemiacom\bantonio`	500
`systemiacom\zleazer`	18
`systemiacom\administrator`	11
139 utilisateurs	2
627 utilisateurs	1

que chaque ordinateur sera lié à un utilisateur principal, qui sera un employé de l'entreprise. De plus, il devrait y avoir un certain nombre de comptes de service et d'administrateurs, qui seront associés à plusieurs hôtes pour effectuer les tâches de soutien.

Le Tableau 4.2 montre les associations entre les hôtes et les comptes d'utilisateurs, en fonction des actions « INTERACTIVE » et « REMOTE ». On peut constater que la plupart des utilisateurs sont fortement liés à un seul hôte, mais qu'un petit nombre d'entre eux ont des sessions sur deux hôtes. De plus, on note quelques cas atypiques ayant des sessions interactives sur un grand nombre d'hôtes, ce qui constitue manifestement une anomalie. À ce moment, en l'absence de données suffisantes, il est impossible de savoir si cette activité est malveillante.

La visualisation du nombre de sessions d'utilisateurs interactifs à travers le temps dans la Figure 4.1 permet de faire quelques observations importantes. La première heure de données a

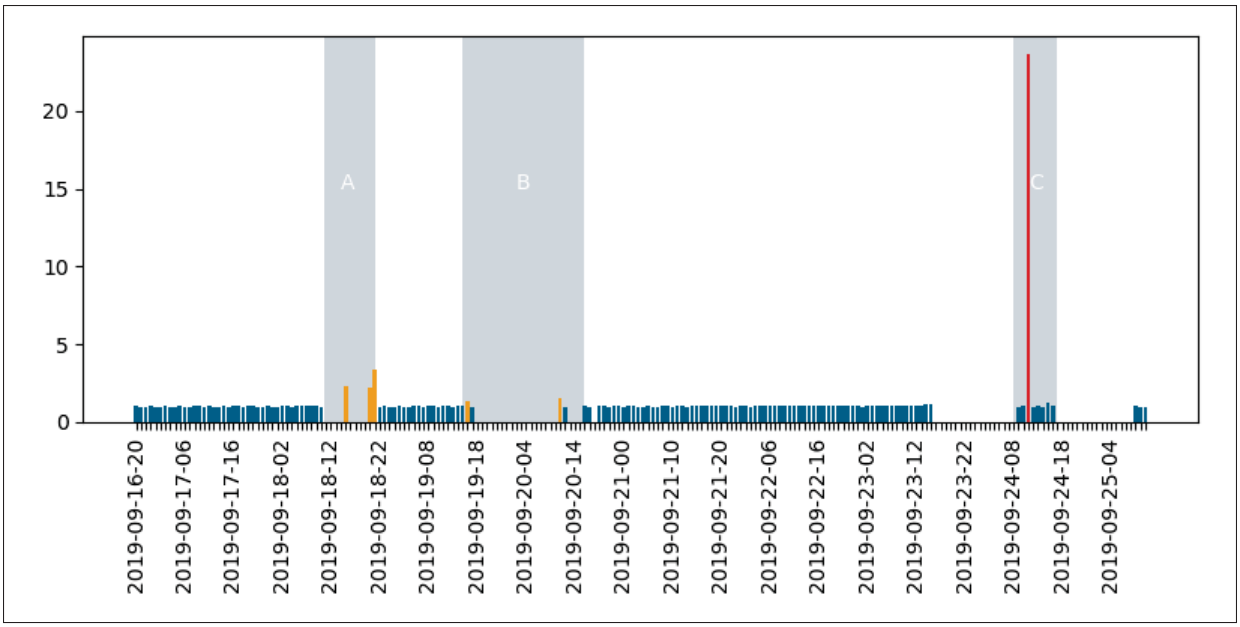

Figure 4.1 Le nombre moyen d'hôtes associés à un utilisateur avec une session interactive par heure

été exclue de la figure, car elle correspond à la création de l'environnement et est caractérisée par un nombre d'évènements disproportionné.

Les espaces blancs de la Figure 4.1 correspondent aux intervalles de temps où aucune session d'utilisateur interactif n'a été observée. Il est impossible de dire si cela est dû à un problème de collecte de données ou à une omission intentionnelle. Il est particulièrement préoccupant de voir deux grandes interruptions pendant la période de l'exercice de l'équipe rouge vers la fin de la chronologie. Cette période peut potentiellement ne contenir que l'activité malveillante atypique pour l'environnement. Par conséquent, le modèle de détection d'intrusion peut produire un nombre de fausses alertes trop bas, pour avantager les métriques de performance.

Cela confirme par ailleurs le manque de visibilité parfaite dans l'environnement. On peut ainsi manquer des activités malveillantes qui ont pu se dérouler durant les interruptions de la collecte de données. Il est donc impossible de se fier exclusivement aux évènements d'authentification. Par conséquent, un IDS robuste doit utiliser plusieurs types d'évènements pour compenser la partialité de l'information.

Pendant le temps qui contient des évènements d'authentification observés, le comportement très uniforme, sans variation durant la journée ou la semaine. Ce phénomène est très typique dans les environnements fortement instrumentés, mais il ne correspond pas au comportement humain typique. Il n'est donc pas étonnant qu'un algorithme d'apprentissage machine avancé soit capable d'identifier ce comportement de base avec une précision presque parfaite.

4.3.2 Des anomalies manifestes

Il est possible d'isoler trois régions (étiquetées « A », « B » et « C ») où le nombre moyen de sessions d'utilisateur dépasse la base. Les trois régions correspondent aux périodes d'interruption de la collecte de données, ce qui suggère que le nombre réel d'évènements d'authentification est encore plus élevé que celui observé. Les interruptions de la collecte de données ainsi que le nombre de journaux peuvent être attribués à la maintenance ou des changements environnementaux. Cependant, la région « C » se distingue davantage, même par rapport aux régions « A » et « B ».

En comparaison, la Figure 4.2 montre le nombre d'authentifications initiales des utilisateurs par heure, ce qui pourrait être un indicateur de mouvements latéraux. La première partie du graphe montre le nombre d'évènements observés. La seconde représente la portion cumulative de toutes les nouvelles authentifications observées jusqu'à présent, en excluant les deux grandes anomalies en haut. La première heure de données a été exclue de la figure, car elle correspond à la création de l'environnement et est caractérisée par un nombre d'évènements disproportionné. Les mêmes régions avec des anomalies sont indiquées pour faciliter la comparaison. On observe que la période « C » demeure une anomalie avec 500 premières connexions en une heure. On observe également une anomalie similaire dans la région « A », tandis que la région « B » ne présente aucune nouvelle connexion, à l'exception d'un léger pic correspondant parfaitement au pic d'évènements représenté sur la Figure 4.1. De plus, il est possible d'identifié une nouvelle région d'anomalie « D », qui présente un pic soudain de nouvelles connexions suivi d'une longue période d'inactivité.

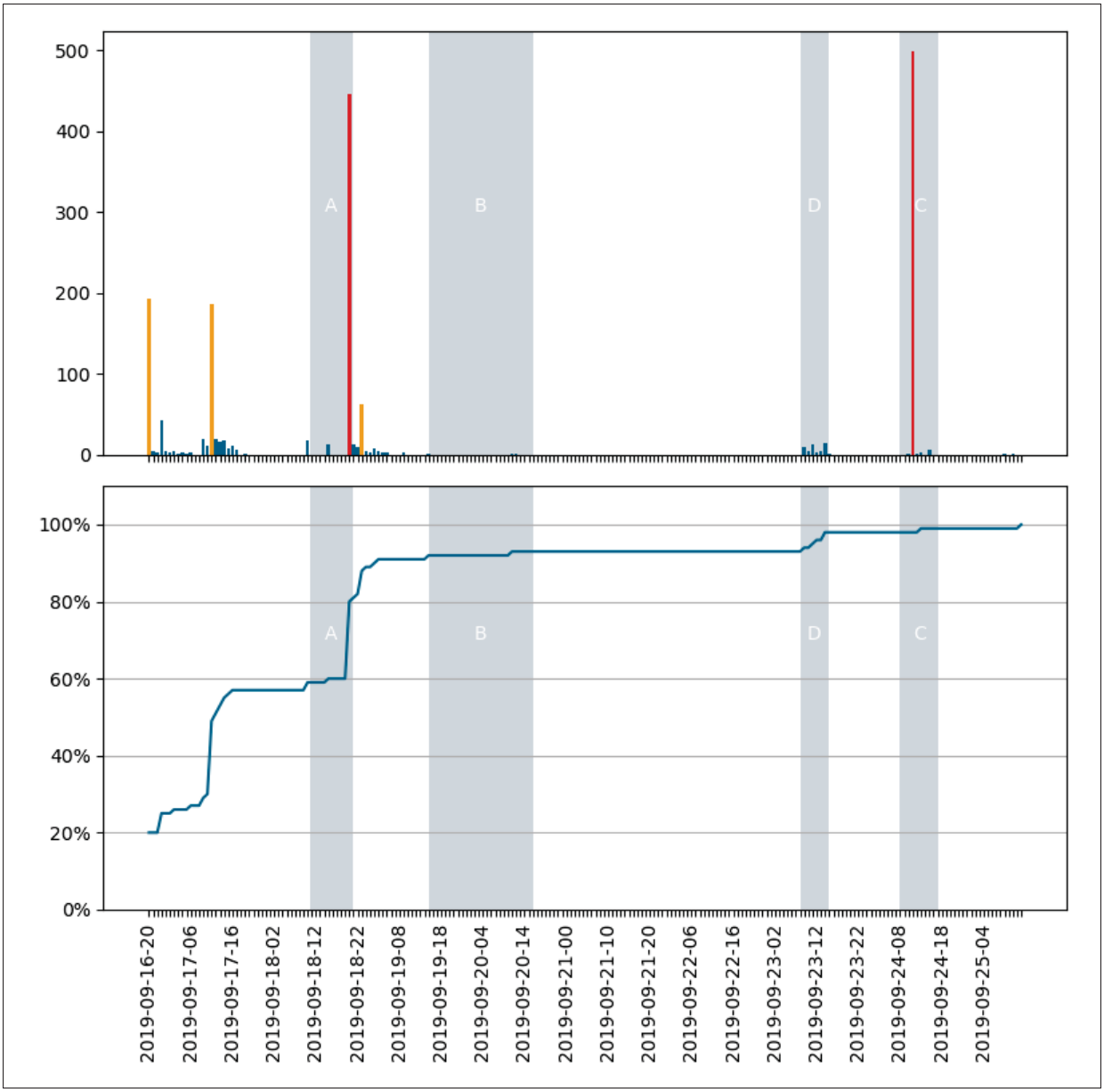

Figure 4.2 Le nombre d'accès initiaux pour de chaque utilisateur sur chaque hôte observé chaque heure

La partie en bas de la Figure 4.2 montre l'accumulation de toutes les nouvelles connexions observées jusqu'à présent. Pour bien saisir le comportement de base des connexions, le graphe exclut les deux anomalies majeures de la plotte en haut, qui représente ensemble la majorité des évènements. On constate ainsi que la croissance est extrêmement inégale, et que la majorité des associations utilisateurs-hôte se produisent au début de la chronologie.

Vers la fin de la période « A », on a déjà observé 80% de toutes les connexions entre les paires des utilisateurs et d'hôtes. Le graphique reste stable à environ 87% entre les régions anormales « B » et « D », même s'il y a un flux constant de sessions utilisateurs pendant cette période, comme le montre la Figure 4.1. Cela signifie que la seconde moitié de la chronologie est complètement saturée. Cette période ne contient aucune nouvelle activité, ce qui est très typique d'un environnement instrumenté avec un nombre fixe d'utilisateurs et d'hôtes.

En revanche, les réseaux corporatifs réels sont constamment infusés par la nouvelle activité, en raison de l'embauche et du départ des employés, des changements de rôles ou encore de la création et de la retraite de serveurs. Dans ce contexte, on s'attend à ce que de nouvelles connexions utilisateur-hôte émergent, ce qui pourrait entrainer de fausses alertes pour un système de détection d'intrusion basé sur l'apprentissage automatique.

Puisque l'activité des utilisateurs du jeu OpTC ne démontre pas cette nouveauté, il est possible de conclure avec certitude qu'OpTC ne simule pas un réseau corporatif réaliste. Cette découverte soulève des questions quant à la transférabilité d'un système de détection d'intrusion entrainé sur le jeu de données OpTC.

Selon l'hypothèse que la seconde moitié du graphique ne contient pas d'activité nouvelle, tout évènement de connexion observé peut être étiqueté comme malveillant.

Un examen minutieux de la région anomale « D » met en évidence une croissance presque parfaitement linéaire de nouvelles paires utilisateurs-hôtes, qui est absentes dans toutes les autres parties du graphique. Cette région correspond exactement au lancement de la campagne de l'équipe rouge, permettant de conclure qu'elle représente une activité malveillante liée au mouvement latéral au sein du réseau. Mais, même sans avoir connaissance des détails, l'activité est assez suspecte pour justifier une vérification de sécurité.

4.3.3 Modèle non supervisé de détection d'anomalies

Il est possible de détecter cette anomalie parce que le comportement des utilisateurs dans l'environnement est très uniforme. L'approche typique pour entrainer un modèle d'apprentissage automatique consiste à diviser les données en ensemble d'apprentissages (les sept premiers jours de la chronologie) et en ensemble de tests (les trois derniers jours de la chronologie). Selon l'hypothèse, le comportement est entièrement saturé dans la première moitié des données. Par conséquent, il est possible d'entrainer un modèle qui capture avec précision toutes les activités simulées dans l'ensemble de données. Le modèle entrainé pourrait ensuite identifier toutes les activités malveillantes qui s'écartent du comportement de base, assurant ainsi un bon score de performance et peu de fausses alertes.

Ce point peut être illusté à l'aide de l'algorithme de détection des anomalies « Local Outlier Factor » sans supervision pour identifier les sessions d'utilisateurs suspectes. L'implémentation de LocalOutlierFactor est fournie dans la version 1.4.1 de la bibliothèque « scikit-learn » (Pedregosa *et coll.*, 2011).

Comme la méthode choisie ne permet pas d'accéder à la description de la réalité de terrain de l'équipe rouge, il est impossible d'évaluer les résultats de ce modèle à l'aide des indicateurs de performance standards. À la place, l'objectif est de comparer les résultats de l'analyse manuelle avec le résultat de l'algorithme. De manière concrète, on s'attend que les résultats correspondent aux régions d'activité anormale identifiées, et qu'ils apparaissent logiques pour un analyste de sécurité.

Compte tenu des critères d'évaluation de l'IDS dans la Section 2.2.3, l'objectif est de minimiser le taux de fausses alertes. Cependant, comme cet objectif n'est pas réalisable sans données étiquetées, il est nécessaire de maintenir le nombre de résultats à un niveau « raisonnablement bas ».

Comme indiqué dans la Section 3.4, la granularité de la détection peut être choisie en fonction des exigences en matière de sécurité. Le but de cette expérience est de détecter un mouvement

latéral potentiel à partir d'un compte d'utilisateur compromis. Pour ce faire, on peut formuler l'exigence de détection comme suit :

> *Déclencher une alerte lorsque l'utilisateur accède aux ressources du réseau d'une manière considérablement atypique par rapport à son comportement de base dans l'environnement.*

En fonction de cette définition, la granularité de détection choisie est « un compte utilisateur par intervalle de temps » et la résolution de détection comme étant d'une heure.

Les journaux « USER_SESSION » sont divisés en tranches d'une heure. Pour chaque utilisateur, le nombre de connexions interactives et de connexions à distance est mesuré. De plus, l'exigence de détection n'est pas limitée à un seul hôte, mais s'applique à l'ensemble des ressources réseau. Par conséquent, il est nécessaire de trouver tous les hôtes auxquels chaque utilisateur a eu accès. Pour ce faire, chaque période rapporte le nombre total de nouveaux hôtes accédés par un utilisateur.

Cela produit un ensemble de données très simple composé de seulement trois caractéristiques et 19477 lignes pour décrire l'activité de tous les utilisateurs pour l'ensemble du jeu de données.

Le résultat du modèle est présenté dans Figure 4.3. Le premier graphe a été entrainé sur l'ensemble des activités des utilisateurs et représente le comportement inhabituel d'un utilisateur détecté à une heure spécifique. La taille du marqueur indique la magnitude de l'anomalie. Le deuxième graphe montre le nouveau comportement. Ce modèle a été entrainé avec les données précédant la ligne verticale et évalué avec les données suivantes. Les points marqués d'un diamant ont été identifiés par les deux modèles. Pour la détection des valeurs extrêmes, l'algorithme « LocalOutlierFactor » est exécuté sur l'ensemble des données. Le paramètre « n_neighbors » a été fixé à 10, pour garder le nombre de résultats à un bas niveau.

Le modèle produit un total de 17 détections pour 7 utilisateurs suspects. Plusieurs détections se situent dans des intervalles adjacents, ils peuvent donc être fusionnés pour réduire le nombre

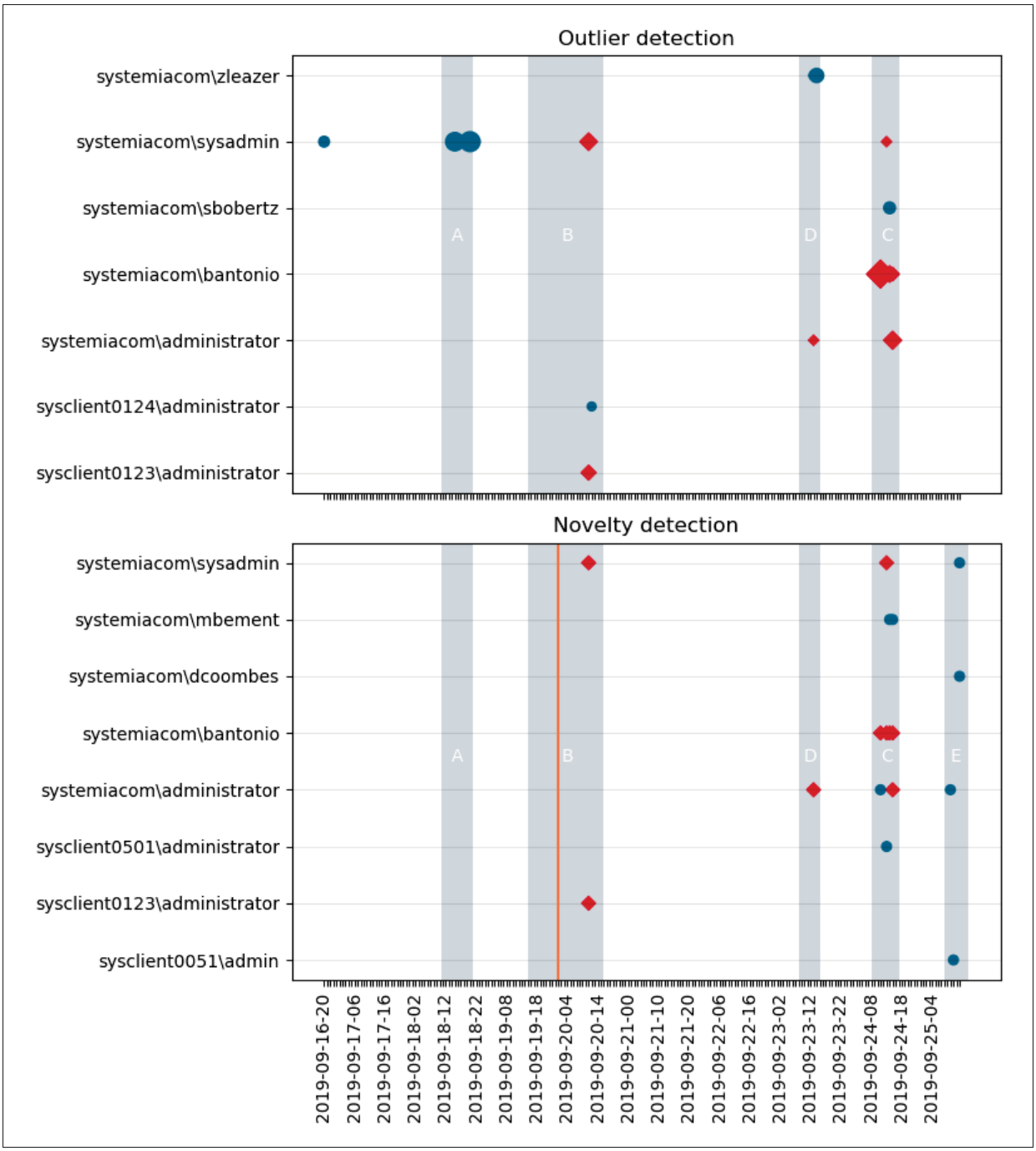

Figure 4.3 Activité d'utilisateur suspecte détectée par un modèle d'analyse en ligne non supervisé

de détections à 11, ce qui représente 0,06% du jeu original. La taille du marqueur représente la magnitude de l'anomalie de la pointe. Cela peut aider un analyste en sécurité à prioriser l'enquête.

Le résultat du modèle correspond parfaitement aux zones d'activité anormale qui étaient repérées manuellement plus tôt, ce qui confirme la justesse de l'analyse initiale. Avec une exception, tous les utilisateurs détectés sont soit connus depuis le Tableau 4.2, soit membres du groupe administrateur privilégié, qui constitue toujours une cible privilégiée pour toute attaque. D'une manière générale, l'activité détectée semble être plausiblement malveillante.

L'hypothèse est que l'activité des utilisateurs est saturée tôt dans le jeu en raison de la nature confinée de l'environnement simulé. L'algorithme « LocalOutlierFactor » peut aider à détecter les nouveautés et, si l'hypothèse est correcte, le modèle devrait produire très peu de résultats pendant la période d'évaluation.

Pour vérifier cette hypothèse, il faut diviser l'ensemble de données en deux sous-ensembles pour l'entrainement et l'évaluation, comme indiqué par la ligne verticale orange sur le graphique en nuage de points. La position de cette division est déterminée par les résultats de Figure 4.2, qui montrent qu'environ 85% de toutes les paires utilisateur-hôte sont déjà observées avant le 20 septembre et que la ligne reste plate pendant plus d'un jour.

Le paramètre « n_neighbors » pour le modèle a été choisi automatiquement à l'aide de l'Algorithme 4.1. Tous les autres paramètres du modèle conservent leur valeur par défaut.

Les résultats de la détection de nouveauté sont différents des résultats de la détection des anomalies. Les entrées retournées par les deux modèles sont indiquées par des marqueurs en forme de diamant. On peut voir que le modèle a identifié certains utilisateurs en plus des utilisateurs administrateurs et que tous les résultats sont encore parfaitement alignés sur les régions anormales connues. Aucun résultat n'a été trouvé pour la période entre les régions « B » et « D », ce qui indique qu'aucune activité d'utilisateur nouvelle n'a eu lieu. Cela confirme l'hypothèse que l'environnement est complètement saturée.

Algorithme 4.1 La sélection automatique du paramètre n_neighbors

Input : Fiche de données $\mathcal{X}$ avec la liste agrégée des sessions d'utilisateurs sans étiquettes
Output : Alertes pour le nouveau comportement

Diviser $\mathcal{X}$ sous-ensembles d'entrainement $\mathcal{X}\_train$ et d'évaluation $\mathcal{X}\_test$;
repeat
  $clf \leftarrow LocalOutlierFactor(n\_neighbors = [20..1], novelty = true)$;
  $clf.fit(\mathcal{X}\_train)$;
until $clf.predict(\mathcal{X}\_train) = \emptyset$;
$alerts \leftarrow clf.predict(\mathcal{X}\_test)$;
return $(alerts)$

Plus important encore, le modèle a identifié un groupe de nouveaux comportements associés à quatre nouveaux utilisateurs, qui est étiqueté comme « E » dans la figure. On peut observer que les groupes « C », « D » et « E » ont des tailles similaires, et qu'ils sont espacés de façon égale. Cette nouvelle région anormale sera étudiée dans l'analyse suivante.

Ces régions correspondent aux dates de l'exercice de l'équipe rouge. Il est donc possible d'en déduire que cela représente des grappes d'activité causées par les attaques. Toutefois, puisque l'accès à la description de la réalité de terrain est limité, il n'est pas possible de déterminer si les détections individuelles sont attribuables à l'équipe rouge.

Il faut avoir plus de données pour être certain que les anomalies observées représentent l'activité malveillante. Indépendamment de cette conclusion, le modèle mis au point constitue une implémentation réussie d'un IDS pour l'environnement ciblé. Il répond aux besoins de l'utilisation prévue et ne déclenche pas plus de six alertes par jour pour une communauté de 752 utilisateurs Active Directory.

Cette charge de travail est raisonnable pour l'équipe de sécurité opérationnelle, donc la mise en production est possible.

Tableau 4.3 Répartition des valeurs dans le champ action

Action	Nombre total	Pourcentage
COMMAND	1509	0.00%
CREATE	20413197	1.35%
LOAD	188	0.00%
MODIFY	13035	0.00%
OPEN	1467379250	97.30%
READ	35846	0.00%
TERMINATE	20227598	1.34%

4.3.4 Conclusion

En conclusion, une analyse simple des objets de type « USER_SESSION » permet de faire plusieurs observations importantes sur le jeu de données OpTC, notamment :

- l'activité des utilisateurs simulés se caractérise par un comportement extrêmement répétitif et monotone, ce qui est typique d'un environnement confiné ;
- il contient au moins cinq régions anormales, chacune distincte et montrant certaines caractéristiques de comportements malveillants potentiels ;
- l'activité anomale ne se limite pas à la période de l'exercice de l'équipe rouge ;
- l'activité anormale peut être facilement distinguée à l'aide d'un modèle d'apprentissage non supervisé.

Ces observations seront renforcées avec l'analyse d'autres types de journaux dans les sections suivantes.

4.4 Expérience 2 : l'exécution des processus

Dans OpTC, l'exécution des processus est représentée par des objets de type « PROCESS ». Le Tableau 4.3 montre les valeurs possibles du champ « action » et le nombre correspondant d'évènements.

Seulement les actions « CREATE » et « TERMINATE » font partie du modèle de données MITRE CAR d'origine. Les cinq autres actions ont été ajoutées par les auteurs du jeu de données OpTC, mais leur objectif précis n'est pas documenté. Intégrer des types d'évènements non documentés dans les entrainements du modèle d'apprentissage machine est facile, et ces derniers devraient être en mesure de détecter automatiquement des motifs dans les données. Cependant, cela fait porter l'entière responsabilité de comprendre ces évènements aux analystes en sécurité, ce qui rend plus difficile l'interprétation des alertes. L'un des objectifs de cette expérience est d'assurer qu'ils soient faciles à comprendre.

Toutes ces nouvelles actions manquent de champs additionnels, ce qui signifie qu'elles ne fournissent pas d'informations supplémentaires sur le processus. Par conséquent, ils seront exclus de cette analyse. Les actions « CREATE » et « TERMINATE » représentent les extrêmes opposés d'une chronologie de processus. Bien que la comparaison puisse aider à déterminer la durée de chaque exécution de processus, pour les besoins de cette analyse, il suffit de se concentrer sur les valeurs de l'évènement « CREATE », qui représente 1,35% des évènements « PROCESSUS », mais seulement 0,1% de l'ensemble du jeu de données OpTC. Cela facilite une exploration rapide des données et des expériences.

Selon la documentation initiale du projet DARPA TC (Griffith *et coll.*, 2020), l'un des objectifs principaux était d'assurer une traçabilité fiable des données. L'ensemble de données OpTC est produit par évaluation plus large des résultats du projet TC, qui peut être représenté comme un graphe de provenance. L'objet « PROCESS » est un élément crucial nécessaire pour déstructurer efficacement ce graphe en utilisant l'information dans les champs « PID » et « PPID ».

La Figure 4.4 montre un arbre de provenance de processus typique qui peut être reconstruit en analysant la chaine d'appels de processus. Un examen rapide des noms de processus révèle des informations sur l'environnement simulé de référence.

L'arbre est ancré dans le processus « mantra.exe », qui semble être un cadre d'automatisation personnalisé qui simule une activité typique d'utilisateur sur l'hôte. Les auteurs de ce projet ont consacré beaucoup d'efforts à la simulation de diverses tâches, telles que des séances interactives,

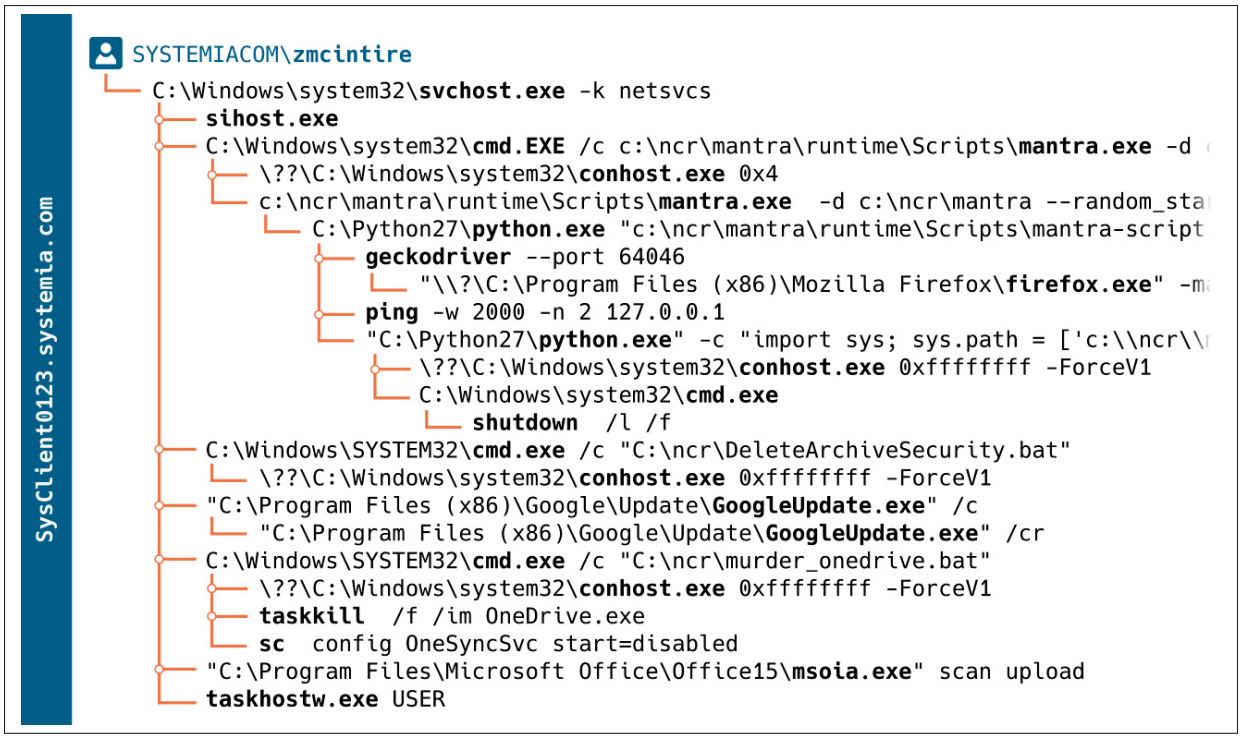

Figure 4.4 Une représentation visuelle d'un arbre de processus typique généré par le processus « mantra »

la navigation web, des traitements de documents Microsoft Word et Excel, ainsi que la lecture de fichiers PDF, etc. Ces sessions sont exécutées en continu avec les identifiants de l'utilisateur visé sur tous les hôtes du réseau, à des intervalles irréguliers, et leur durée varie. Le but de cette randomisation est probablement d'éliminer tous les motifs répétitifs de comportement que les algorithmes d'apprentissage automatique peuvent facilement détecter.

4.4.1 Description de l'environnement

La complexité des vrais réseaux découle de la variété de la communauté d'utilisateurs, chacun ayant ses propres objectifs professionnels, ses applications favorites et son propre rythme. Un simple dénombrement des valeurs uniques dans le champ « image_path » met directement en évidence la nature confinée du jeu OpTC.

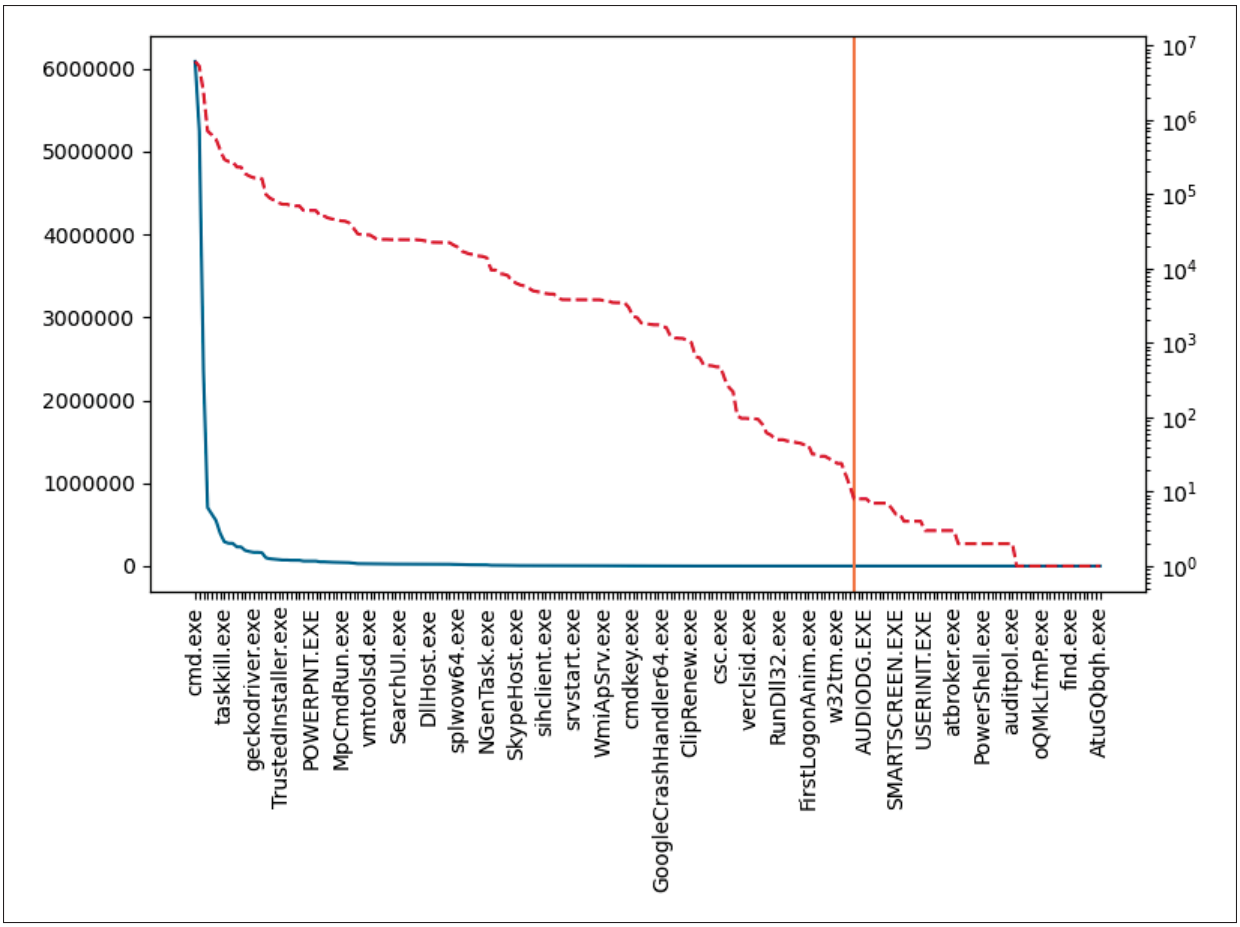

Figure 4.5 La répartition de toutes les images de processus du jeu de données OpTC, en échelle linéaire et logarithmique. La ligne verticale délimite la portion de toutes les images ayant moins de 10 exécutions sur tous les hôtes

La Figure 4.5 montre le nombre total d'exécutions de processus en utilisant le dernier composant du chemin de fichier. Cette méthode n'est pas idéale, car elle dépend d'une partie du nom du fichier, qui pourrait être manipulée par un attaquant. Pour cette raison, les outils de criminalistique numérique utilisent des algorithmes de hachage qui sont une méthode plus fiable d'identification de fichiers uniques et de détection d'attaques subtiles. Malheureusement, ces informations ne sont pas disponibles dans l'ensemble des données, il est donc nécessaire d'utiliser les noms de fichiers.

Il n'y a que 218 images de processus uniques dans l'ensemble du jeu de données, et elles respectent nettement la loi de puissance. Cette situation découle de la nature des simulations

```
            action: "CREATE"
      command_line: "\"C:\\Windows\\system32\\plink.exe\" sports.com -R
                     4000:127.0.0.1:3389 -l myguest -pw myguest"
          hostname: "SysClient0501.systemia.com"
        image_path: "\\Device\\HarddiskVolume1\\Windows\\system32\\plink.exe"
parent_image_path: "powershell.exe"
         principal: "SYSTEMIACOM\\administrator"
         timestamp: "2019-09-24T13:10:02.021-04:00"
              user: "SYSTEMIACOM\\Administrator"
```

Figure 4.6 Un exemple de processus anormal avec une seule exécution dans le jeu de données complet

d'activité des utilisateurs créés par les auteurs du jeu de données, car il est possible d'automatiser seulement un petit nombre de scénarios. Dans ce contexte, les processus avec un petit nombre d'exécutions sont manifestement des cas exceptionnels qui pourraient signaler une activité malveillante. Il y a 21 images de processus qui ne sont exécutées qu'une seule fois, et l'inspection manuelle a révélé des traces d'activité malveillante clairement démontrées dans la Figure 4.6. La ligne de commande montre l'utilisation de plusieurs techniques d'attaque bien connues, que tout analyste de sécurité expérimenté identifierait immédiatement comme malveillantes : T1021.004, T1059.001, T1078.002, T1090.004, T1571, T1572 et T1573 dans la taxonomie de l'ATT&CK de MITRE (MITRE, 2023) sur les tactiques et techniques adverses.

Pour illustrer davantage la nature confinée des exécutions des processus simulés, on peut comparer l'activité de tous les utilisateurs au moyen du coefficient de similitude Jaccard, qui est défini à l'Équation 4.1. Ici, « A » et « B » sont des ensembles de représentations d'images de processus pour chaque utilisateur, comme indiqué à la Figure 4.5.

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} \tag{4.1}$$

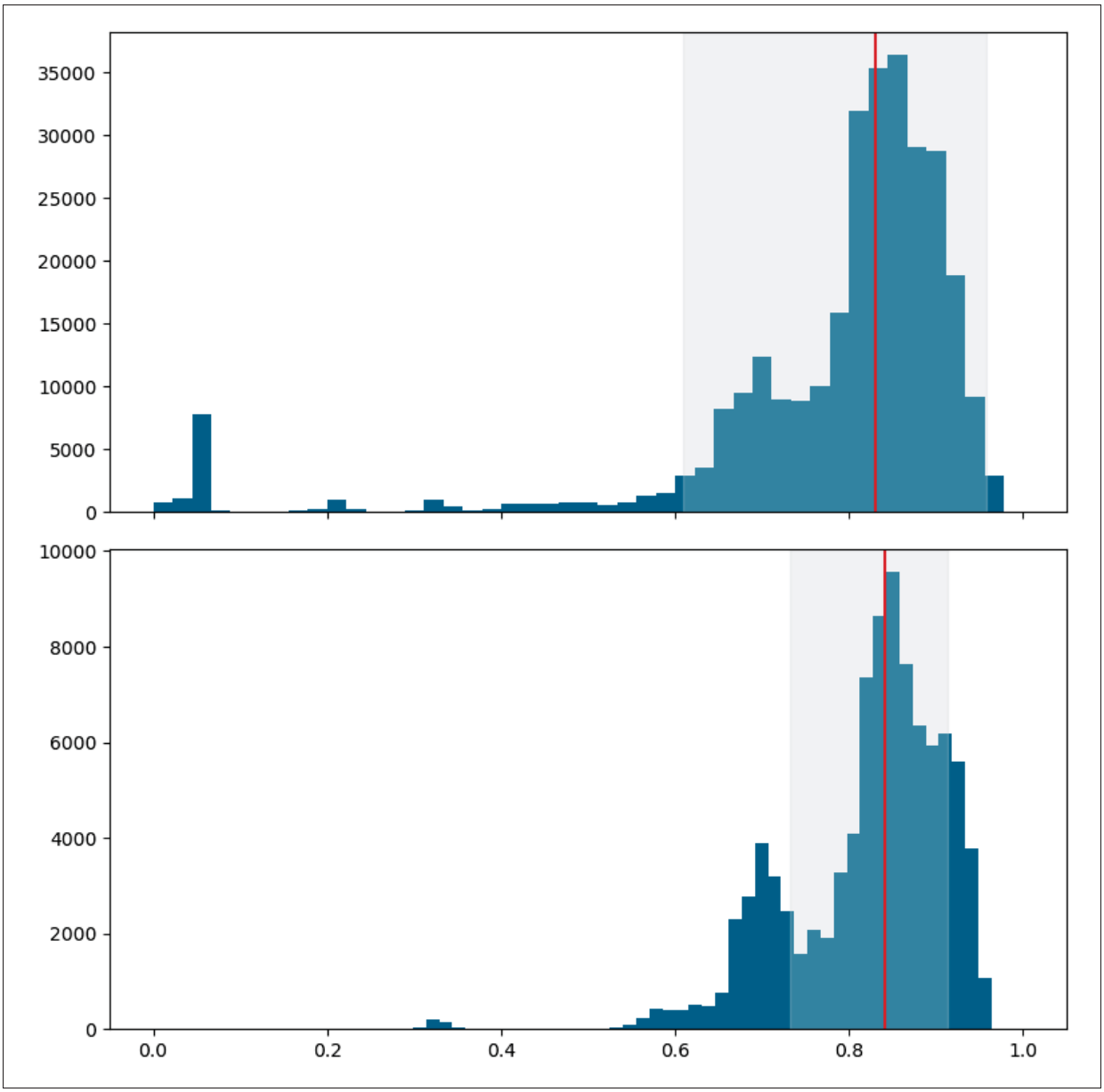

Figure 4.7 Histogramme de l'indice de similarité de Jaccard entre les processus

La Figure 4.7 présente l'histogramme des index de tous les utilisateurs du jeu de données. La section grisée représente l'écart-type, et la ligne verticale indique la médiane. Le premier graphique montre l'activité complète des utilisateurs. Toutefois, le jeu ne contient pas la télémétrie complète de tous les hôtes, ce qui entraine une distribution oblique à cause de l'observabilité partielle pour certains utilisateurs. Le second graphique est limité aux utilisateurs ayant effectué

Tableau 4.4 Utilisateurs ayant un petit index de corrélation

Utilisateur	**Index moyen**	
sysclient0124\administrator	0.021	
systemiacom\administrator	0.042	
sysclient0103\administrator	0.067	
sysclient0114\administrator	0.067	
sysclient0169\administrator	0.067	
sysclient0221\administrator	0.067	
sysclient0225\administrator	0.067	
sysclient0271\administrator	0.067	
sysclient0301\administrator	0.067	
sysclient0352\administrator	0.067	
sysclient0363\administrator	0.067	
sysclient0367\administrator	0.067	
sysclient0371\administrator	0.067	
	Médian	0.830
	Moyenne	0.783
	Variance	0.030

au moins 5000 exécutions. Par conséquent, sa distribution est plus symétrique. En moyenne, les comportements des utilisateurs sont assez semblables. Mais on peut observer un petit groupe de résultats très différents, ce qui fait augmenter l'écart-type, comme le montre la section grisée dans le premier graphique.

Il existe deux types d'utilisateurs avec un petit nombre d'exécutions de processus très dissimulés : l'activité partiellement observée provenant des hôtes exclus du jeu de données, et des groupes d'activités malveillantes générées par l'équipe rouge. Le Tableau 4.4 liste les utilisateurs ayant l'indice de similitude le plus bas, qui forment une grappe dans la partie gauche du graphique en haut de la Figure 4.7. Tous ces utilisateurs sont des comptes administrateurs locaux, souvent utilisés pour obtenir une persistance sur l'hôte après une intrusion. Cette activité est donc très suspecte.

Pour mieux comprendre l'environnement, il faut exclure tous les utilisateurs ayant exécuté moins de 5000 applications, en se concentrant uniquement sur les utilisateurs représentatifs. Le deuxième graphique en bas montre l'écart-type le plus petit, avec une médiane de 0,84. Cela

Tableau 4.5 Les cas suspects dans un échantillon représentatif d'utilisateurs

Utilisateur	**Index moyen**	
systemiacom\sysadmin	0.335	
systemiacom\bantonio	0.570	
systemiacom\zleazer	0.581	
systemiacom\bbateman	0.604	
systemiacom\dcoombes	0.628	
	Médian	0.841
	Moyenne	0.823
	Variance	0.008

confirme que le comportement de la plupart des comptes utilisateurs est identique. Ce n'est pas un comportement naturel, car, dans les réseaux réels, la similarité moyenne du comportement des utilisateurs doit être beaucoup plus faible, par exemple, inférieur à 0.5.

Le Tableau 4.5 présente tous les utilisateurs qui se situent à plus de deux écarts-types de la moyenne. Seulement cinq comptes ont été identifiés. Les résultats de cette expérience suggèrent qu'ils sont potentiellement piratés par l'équipe rouge. Cela confirme que l'activité malveillante peut être facilement distinguée de l'activité normale du réseau.

4.4.2 Randomisation de la ligne de commande

L'analyse ci-dessus repose sur la valeur du champ « image_path », qui contient uniquement le nom du fichier exécutable. Les évènements de processus contiennent également le champ « command_line », qui montre la commande exacte, y compris les arguments et tout artéfact non intentionnel, tels que des espaces répétés, la capitalisation, les guillemets, etc. Tous ces éléments peuvent être transformés en caractéristiques additionnelles et utilisés pour détecter des anomalies.

Contrairement au champ « image_path », qui ne contient qu'un nombre limité de valeurs, le champ « command_line » possède une quantité significativement plus élevée d'entropie, avec 410417 valeurs uniques. Cette caractéristique est due à la décision des auteurs d'inclure plusieurs instructions comportant un élément randomisé dans le nom de la commande ou dans ses

```
"C:\Program Files (x86)\Adobe\Reader 9.0\Reader\AcroRd32.exe"  C:\documents\XQUIOOQ.pdf
"C:\Program Files (x86)\Adobe\Reader 9.0\Reader\AcroRd32.exe"  C:\documents\FORAM-MADEA-V5.pdf
"C:\Program Files (x86)\Adobe\Reader 9.0\Reader\AcroRd32.exe"  "C:\documents\Airin Nibss.pdf"

"\\?\C:\Program Files (x86)\Mozilla Firefox\firefox.exe" -marionette -profile C:\Users\atong\AppData\Local\Temp\rust_mozprofile.kZ7kCMa9jYli
"\\?\C:\Program Files (x86)\Mozilla Firefox\firefox.exe" -marionette -profile C:\Users\jgoshorn\AppData\Local\Temp\rust_mozprofile.moXydoPB772p
"\\?\C:\Program Files (x86)\Mozilla Firefox\firefox.exe" -marionette -profile C:\Users\JWEERA~1\AppData\Local\Temp\rust_mozprofile.ncY4vYGr4gQB

cmd.exe  /c cmdkey.exe /generic:142.20.61.188 /user:systemia.com\zleazer /pass:Targeteer.
cmd.exe  /c cmdkey.exe /generic:142.20.61.189 /user:systemia.com\ajoaquim /pass:Familiary.
cmd.exe  /c cmdkey.exe /generic:142.20.61.189 /user:systemia.com\akampf /pass:Mahometry.

"C:\Program Files\Windows Defender\MpCmdRun.exe" GetDeviceTicket -AccessKey 299AD876-CC71-F7DF-9A5F-C5D86207CF4A
"C:\Program Files\Windows Defender\MpCmdRun.exe" GetDeviceTicket -AccessKey 29C642F0-B33C-AE7D-60E2-6BDDF5082013
"C:\Program Files\Windows Defender\MpCmdRun.exe" GetDeviceTicket -AccessKey 29E9D3AE-7D84-FBFD-91CA-26A6EA61196D

"C:\Users\jgomez\AppData\Local\Microsoft\OneDrive\OneDrive.exe" /background
"C:\Users\jgoshorn\AppData\Local\Microsoft\OneDrive\OneDrive.exe" /background
"C:\Users\jhorowitz\AppData\Local\Microsoft\OneDrive\OneDrive.exe" /background
```

Figure 4.8 Un exemple de plusieurs exécutions de processus avec des chaines de lignes de commande uniques, générées par un processus automatique à partir d'un gabarit

paramètres. La Figure 4.8 met en évidence quelques exemples d'exécutions du même processus, mais avec des arguments de ligne de commande différents en raison de cette randomisation. Les composants aléatoires de la chaine de commande sont mis en évidence par des couleurs différentes. Inverser le gabarit sous-jacent des fichiers de journaux bruts est une tâche non triviale qui nécessite la suppression des composants uniques. Cette approche a probablement été choisie pour rendre l'activité de chaque utilisateur plus unique et cacher les motifs d'exécution de processus, dans le but de faire l'identification de l'activité malveillante plus difficile.

C'est une excellente démonstration d'une autre propriété utile du jeu de données OpTC. En cybersécurité, la télémétrie contient souvent des champs redondants et très similaires, qui peuvent être traités et combinés pour créer de nouvelles caractéristiques, comme le champ « image_path ». Ainsi, en incluant des évènements bruts au format JSON, OpTC offre une plus grande flexibilité dans les recherches sur IDS par rapport aux jeux de données traditionnels, qui contiennent souvent uniquement des données posttraitées et normalisées avec un ensemble limité des caractéristiques.

Cependant, puisque cette randomisation du nom des processus rend impossible toute analyse statistique, il faut choisir entre deux stratégies pour travailler avec ces données : utiliser des méthodes d'apprentissage machine plus avancées, ou ajouter une étape de prétraitement supplémentaire pour normaliser les composants randomisés.

La normalisation manuelle, qui consiste à utiliser des expressions régulières à la main, est une méthode couramment employée par un analyste SOC dans des situations similaires. Ses avantages incluent sa simplicité et sa précision extrême. Cependant, elle nécessite beaucoup de temps, elle est fragile et présente d'autres inconvénients qui rendent sa portabilité difficile dans d'autres environnements.

Dans la recherche universitaire, le problème de détection de motifs généralisés dans les journaux est appelé extraction du gabarit. Cette technique s'avère utile dans d'autres domaines de détection d'anomalies, au-delà de la détection d'intrusion. Cependant, cette branche de recherche a longtemps souffert du même problème : le manque de données d'entrainement de haute qualité. La plupart des publications utilisent les mêmes quatre jeux de données publiques, qui sont tous très vieux et ont été démontrés comme étant très simples (Jiang *et coll.*, 2024).

Par conséquent, le jeu de données OpTC offre un grand potentiel d'être utile pour divers types de détection d'anomalies, allant au-delà de la recherche sur les intrusions, comme l'analyse de journaux et leur agrégation.

4.4.3 Modèle non supervisé de détection d'anomalies

Les techniques d'analyse manuelle de journaux peuvent être très efficaces pour un analyste de sécurité expérimenté. Cependant, l'objectif de ce travail est d'automatiser ce processus. De manière similaire aux sections précédentes, il faut entrainer un modèle de détection d'anomalies simple pour évaluer tous les journaux de création de processus et les corréler avec les activités de l'équipe rouge.

La méthodologie est suivante : on commence par définir les besoins en sécurité et déterminer la granularité de la détection qui répond à ces exigences. Comme la réalité de terrain n'est pas encore connue, on peut seulement utiliser des méthodes de détection d'anomalies non supervisées, et le nombre de résultats doit être « raisonnablement faible ».

Le scénario idéal est de relier toutes les activités malveillantes à une identité d'utilisateur spécifique, ce qui permettrait de les associer directement aux résultats de la section précédente. Malheureusement, cette approche ne fonctionne pas bien dans ce cas, car la plupart des processus sont en fait exécutés par le système lui-même et ne sont pas nécessairement liés à une identité utilisateur. L'analyse des activités uniquement dans le cadre des sessions interactives présente le risque de manquer certaines tentatives d'élévation de privilèges et de mouvements latéraux. De plus, il ne suffit pas d'émettre une alerte pour chaque exécution d'un processus inconnu, car cela entrainerait un grand nombre de faux positifs dus aux activités administratives.

Il faut donc suivre l'activité de base de l'ensemble du réseau et détecter lorsqu'un hôte spécifique présente un nouveau comportement. On peut formuler l'exigence de détection comme suit :

> *Identifier un hôte qui exécute un processus inhabituel qui n'a pas de précédents similaires sur les autres hôtes du réseau.*

La granularité des évènements de création de processus individuels est suffisante pour les besoins de cette expérience, car ces évènements contiennent toutes les informations nécessaires pour lancer une enquête et peuvent être directement liés à un hôte et une identité d'utilisateur.

Cette recherche a déjà démontré que le domaine « image_path » facilite considérablement l'identification des processus anormaux. Par conséquent, cette tâche utilisera le champ « command_line » complet. Ensuite, comme postulé dans le travail de Golczynski et Emanuello (2021), il est souhaitable de limiter au maximum l'ingénierie manuelle des caractéristiques pour augmenter les chances de détecter les attaques inconnues.

TF-IDF (Salton et Buckley, 1988) est une technique bien connue pour identifier les termes clés et uniques dans un ensemble de textes. Elle est adaptée pour analyser une séquence de journaux et aide à détecter des commandes atypiques et des combinaisons de paramètres, ce qui correspond essentiellement pour identifier le nouveau comportement. L'Équation 4.2 représente la manière la plus fréquemment utilisée de calculer le TF-IDF :

$$tfidf(t, d, D) = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}} \cdot \log \frac{|D|}{|\{d \in D : t \in d\}|} \tag{4.2}$$

Cependant, il est impossible d'utiliser cette formule telle quelle, car la valeur d'une ligne de commande inclut le nom de commande et un certain nombre de paramètres, qui servent des fonctions distinctes. En d'autres termes, il s'agit presque toujours de termes non répétés. Par conséquent, la valeur TF de tous les termes sera pratiquement identique, ce qui indique que le premier terme de l'équation ne fournira pas d'information. Pour résoudre ce problème, il est nécessaire d'utiliser une mesure différente, qui est définie par l'Équation 4.3 :

$$S_h(t, D_h, D_n) = \log \frac{|D_h|}{|\{d \in D_h : t \in d\}|} \cdot \log \frac{|D_n|}{|\{d \in D_n : t \in d\}|} \tag{4.3}$$

Pour identifier le comportement anormal sur chaque hôte individuellement, il faut construire deux corpus de documents simultanément, ce qui permet d'avoir deux termes IDF indépendants.

- Le premier terme identifie le comportement anormal pour chaque hôte, répondant ainsi à la moitié des exigences en matière de détection.
- Le deuxième terme établit une comparaison avec le comportement de tous les autres hôtes, ce qui normalise la valeur des processus habituels dans cet environnement. Cela permet de répondre à la deuxième exigence.

Puisque l'objectif de cet exercice est de détecter les traces d'activité malveillante, il faut utiliser les données complètes de tous les jours pour entrainer et évaluer le modèle. Cependant, tout nouveau comportement identifié au début de l'engagement est dû à la création de l'environnement par les administrateurs système. Les résultats des deux premiers jours seront exclus des indicateurs de performance.

Le modèle va utiliser en entrée les valeurs brutes provenant du champ « command_line » sans aucun traitement préalable, avec une seule exception. Les identifiants des composants internes

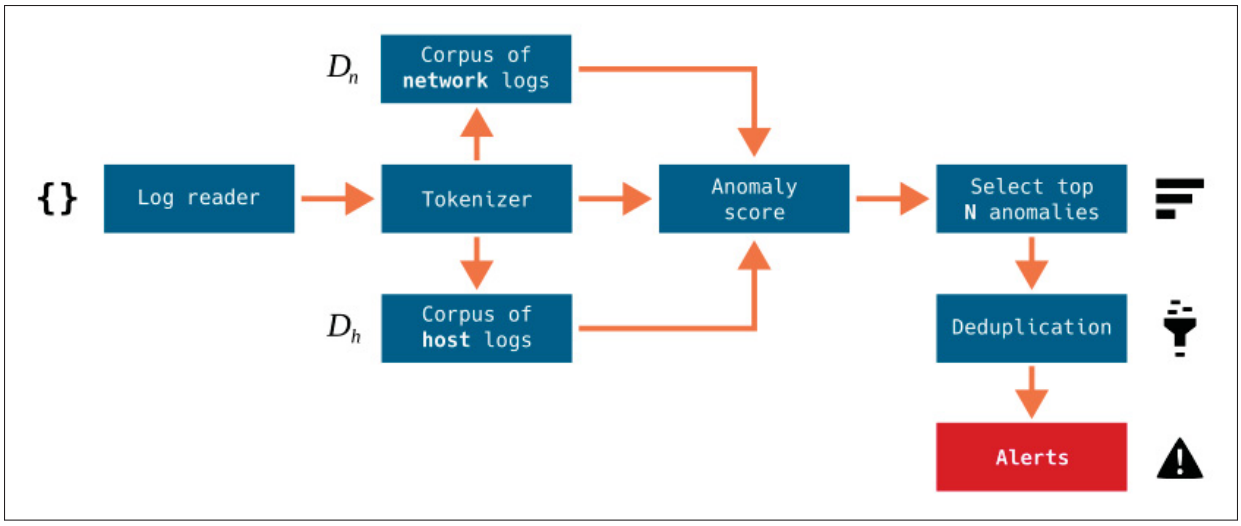

Figure 4.9 Un pipeline de détection d'anomalies pour repérer des comportements inhabituels sur les hôtes dans l'ensemble des journaux en utilisant la fréquence inverse documentaire doublée

du système d'exploitation Windows sont des séquences de 36 cataractes aléatoires en format de GUID, par exemple « {CF0A0F7C-DE44-47F4-9C04-570831A8CE48 } ». Ces séquences sont omniprésentes dans les processus internes du système et, en raison de leur longueur, prennent la plupart de l'espace dans les journaux et produisent de nombreuses fausses alertes avec un niveau d'anomalie élevé.

Une solution potentielle serait d'inclure une étape de la détection du gabarit au début du pipeline, comme discuté dans la Section 4.4.2, mais l'objectif est d'utiliser les données brutes autant que possible. Heureusement, les chaines GUID ont un format bien défini, ce qui permet de les supprimer facilement et avec une grande précision à l'aide d'une expression régulière. Ainsi, toutes les chaines GUID sont remplacées par une valeur unique « GUID » lors de la lecture des données du disque.

Cependant, ce n'est pas une propriété unique du jeu de données OpTC. Ce problème se reproduira également sur tout système Windows. Cette étape de prétraitement est donc justifiée et ne conduit pas à une perte de généralité de la méthode de détection d'anomalies.

Le pipeline complet est présenté dans la Figure 4.9. L'algorithme effectue deux passages sur l'ensemble des évènements de création de processus pour produire une liste des N exécutions de processus anormales. N est contrôlé par un analyste en cybersécurité qui l'évalue en fonction des ressources disponibles ou du taux de fausses alertes des résultats.

La déduplication est la dernière étape avant la génération d'un avertissement. Son objectif est de supprimer un grand nombre d'alertes presque identiques causées par des processus répétitifs (en raison de la randomisation de l'argument décrit dans la Section 4.4.2). Puisque l'anomalie est déterminée uniquement par le texte de la ligne de commande, l'exécution du même processus plusieurs fois entrainera toujours le même score d'anomalie. Cela peut potentiellement inonder la sortie d'alertes doublons.

Cet algorithme de détection n'est pas conçu pour signaler chaque exécution de processus suspecte, mais plutôt pour identifier des comportements hôtes inhabituels. Ainsi, une seule alerte par hôte fournit suffisamment d'informations aux analystes de sécurité. Cependant, le processus de déduplication ne supprime pas les alertes en double ; elles sont plutôt agrégées et sont présentées aux analystes sous forme de groupe. Ce contexte additionnel facilite l'enquête et aide à identifier rapidement les faux positifs (même s'ils se produisent sur des hôtes différents) ou à repérer des traces de mouvement latéral.

Le pipeline garde une liste de toutes les alertes générées jusqu'à présent. Chaque nouveau candidat à une alerte est comparé à cette liste en utilisant la classe « SequenceMatcher » de la bibliothèque standard Python (Python Software Foundation, 2024). Deux commandes sont considérées comme égaux si leur ratio de similitude est supérieur à un certain seuil, qui est défini comme un hyperparamètre de ce modèle. Pour cette expérience, le seuil de similitude de 0.70 produit de bons résultats.

La Figure 4.10 montre la précision du modèle selon le nombre d'alertes N.

La précision est la seule métrique de performance standard qui peut être utilisée à cette étape, car les données n'ont pas d'étiquettes, donc le nombre de faux négatifs est inconnu. Par contre,

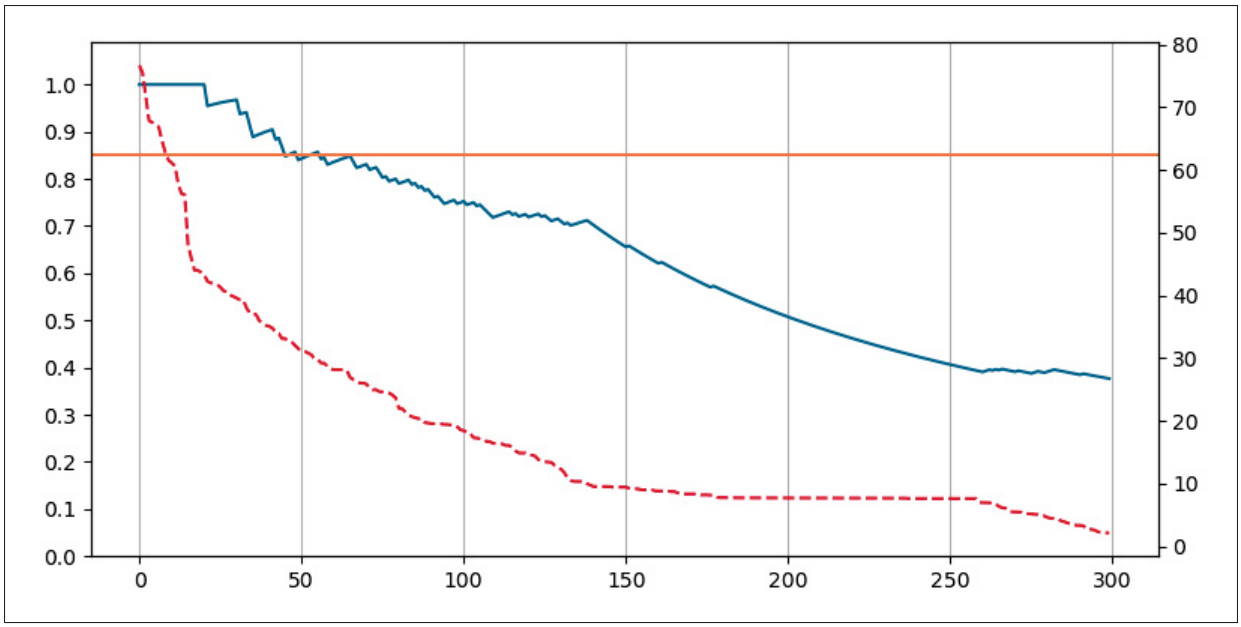

Figure 4.10 La précision du modèle de détection non supervisée en fonction du nombre N d'alertes générées. L'axe secondaire affiche le niveau d'anomalie des 300 premières alertes émises par le modèle

pour calculer la précision, seulement le nombre de vrais positifs et de faux positifs est nécessaire. Pour obtenir ces deux chiffres, il faut trier les alertes générées et leur attribuer une étiquette.

Tout d'abord, le modèle est entrainé sur le jeu de données complet et est évalué progressivement en fonction du nombre d'alertes N dans l'intervalle $[1, 300]$. Ensuite, chaque alerte générée par le modèle est considérée comme un vrai positif si la ligne de commande révèle des indices clairs d'activité malveillante ou s'il existe un lien avec d'autres activités malveillantes. Le résultat est considéré comme un faux positif si l'action est clairement bénigne ou si l'action est liée à l'activité bénigne du système. Si le classement n'est pas clair, le résultat est considéré comme un faux positif, ce qui désavantage le modèle.

Le graphique montre l'évaluation de la métrique de la précision en fonction du nombre d'alertes N. Sur le graphe, on peut voir aussi le niveau d'anomalie de chaque alerte. La courbe s'aligne sur une distribution logarithmique.

La plupart des 50 premières alertes sont des vrais positifs, ce qui produit une précision très élevée d'environ 0,9. Pour les 100 alertes suivantes, le taux de faux positifs augmente progressivement et diminue la précision à 0,75.

Après un certain point, le modèle cesse complètement de produire de vrais positifs, ce qui entraine une dégradation progressive de la précision jusqu'à ce que la majorité des alertes soient de faux positifs. Une explication possible est que toutes les activités malicieuses ont déjà été détectées avec des scores d'anomalie élevés. Le point de coupure optimal semble se situer autour de 80 alertes, après quoi la précision tombe sous 0,85. Par conséquent, l'analyse subséquente utilisera ce nombre d'alertes.

4.4.4 Analyse des anomalies détectées

Cette section sert à établir une correspondance entre les alertes identifiées et les comportements malveillants spécifiques de la campagne de l'équipe rouge.

La Figure 4.11 montre les 80 résultats les plus élevés de la section précédente, superposés aux régions anormales identifiées par le modèle de détection dans la Section 4.3.3. Cela inclut des alertes déterminées comme étant de faux positifs. Les hôtes sont triés par le temps des premières alertes. L'axe secondaire indique le nombre cumulatif d'alertes observées jusqu'à présent.

C'est clair que la plupart des alertes se concentrent dans les régions anormales accentuées, comme identifié précédemment, et font des grappes serrés dans le temps. La ligne sur l'axe secondaire représente le nombre cumulatif d'alertes observées jusqu'à l'arrivée de chaque nouvelle alerte. Les régions « C », « D » et « E » se distinguent nettement par une hausse soudaine. Puisque ces trois régions correspondent aux trois jours de l'activité de l'équipe rouge, cela renforce la confiance dans la capacité des alertes à identifier l'activité malveillante.

Un point intéressant est que presque un quart de toutes les alertes se situent au début du graphique, entre les régions « A » et « B ». De plus, il y a très peu de corrélation avec les comportements malveillants qui surviennent plus tard. Cela a plusieurs implications importantes

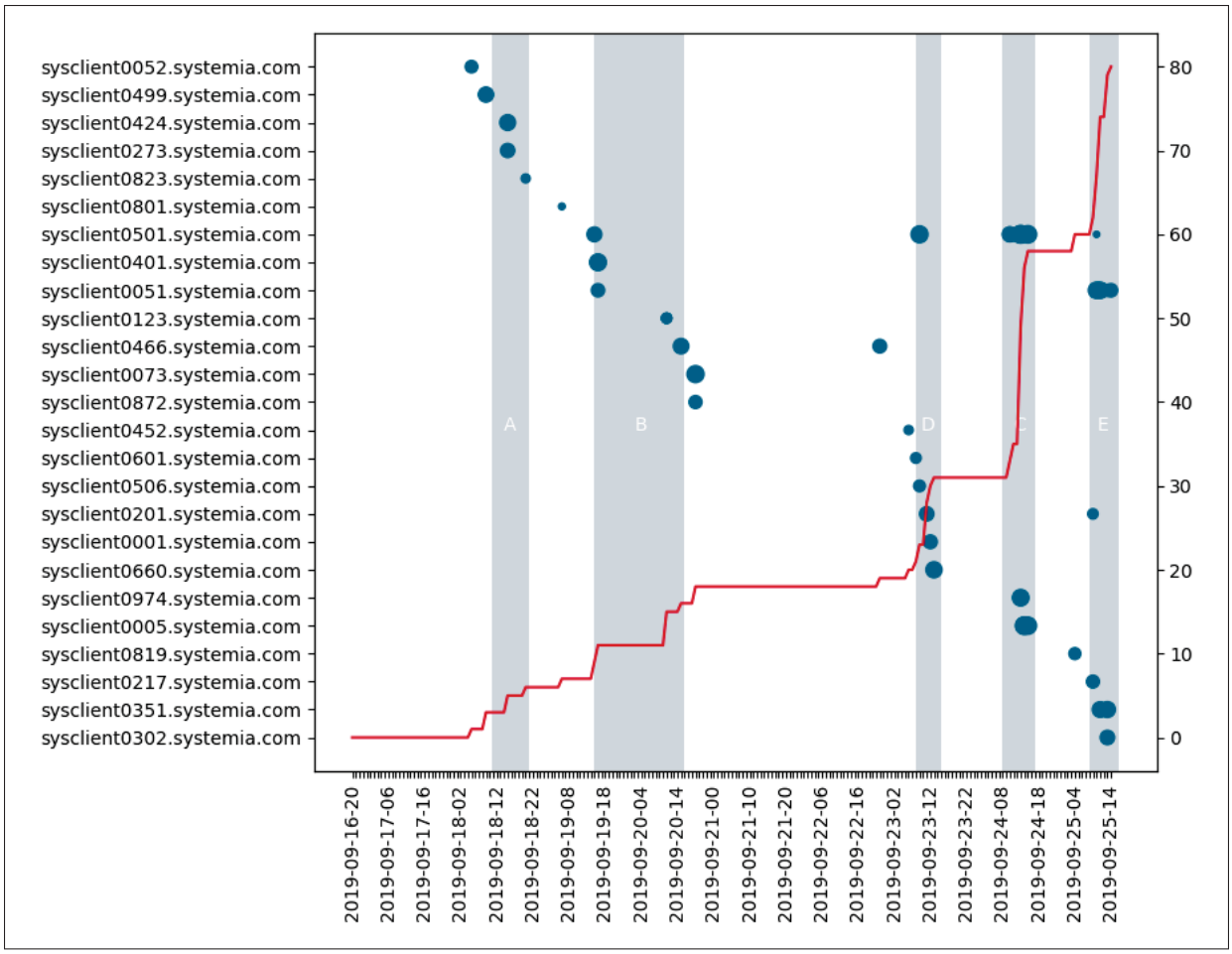

Figure 4.11 Les 80 premières alertes émises par le modèle d'analyse d'anomalies, superposées aux périodes d'activité anormale précédemment identifiées

pour l'évaluation de l'efficacité du modèle. Comme déjà abordé, à cause de la nature confinée du jeu de données et au fait que toutes les anomalies importantes détectées pendant la période d'évaluation de trois jours étaient malveillantes, il devient très facile d'obtenir une bonne performance du modèle. Cela est confirmé par les résultats des travaux précédents.

Cependant, certaines alertes provenant de la première période peuvent susciter l'intérêt d'un analyste de sécurité. Par exemple, l'exécution de la commande « auditpol.exe /get /category :* » par l'utilisateur « SYSTEMIACOM\bantonio » sur « 2019-09-19 ». Puisque les dates de la campagne de l'équipe rouge sont connus, il est possible de raisonnablement expliquer cela

comme un test de la configuration du système pour assurer le bon fonctionnement de l'attaque simulée suivante. Mais, dans un contexte réel, un analyste en sécurité n'a pas ce luxe. Il doit donc présumer que cet évènement correspond à une collecte d'information dans une première étape de la chaine d'attaque. Par conséquent, cette alerte est considérée comme un vrai positif, même si elle ne coïncide pas avec la période d'engagement de l'équipe rouge.

L'exemple ci-dessus illustre le plus fondamental d'un déploiement d'IDS industriel : distinguer de manière fiable entre des activités malveillantes réelles et des activités administratives légitimes qui entrainent la génération de faux positifs.

L'analyse démontre clairement que le jeu OpTC présente des traces de ces deux types d'activité. Cependant, elles se produisent à des points différents dans le temps. Puisque toutes les études antérieures ont évalué uniquement pendant la période officielle des attaques, les rapports de performance peuvent être trop optimistes. Sans tester le système sur des données qui ne contiennent aucune activité malveillante, il est impossible de déterminer le nombre de fausses alertes qui seront générées en raison d'activités administratives légitimes.

Par conséquent, les futurs utilisateurs du jeu de données OpTC devraient utiliser une période de huit jours, commençant le « 2019-09-18 », plutôt que la période officielle de trois jours de l'activité de l'équipe rouge pour évaluer les performances.

En inversant l'axe, il est possible de représenter les alertes selon l'identité de l'utilisateur, ce qui produit la Figure 4.12. Les lignes superposées représentent le total des anomalies détectées parmi les 80 alertes sélectionnées pour chaque utilisateur sur tous les hôtes.

Les lignes superposées représentent le total des anomalies de l'utilisateur, calculé comme la somme de toutes les alertes. Le résultat le plus élevé est une identité générique de système dans Windows, qui peut rapidement accumuler un haut niveau d'anomalie en rassemblant des alertes de tous les systèmes.

Les noms de six prochains utilisateurs sont déjà connus grâce à l'analyse précédente.

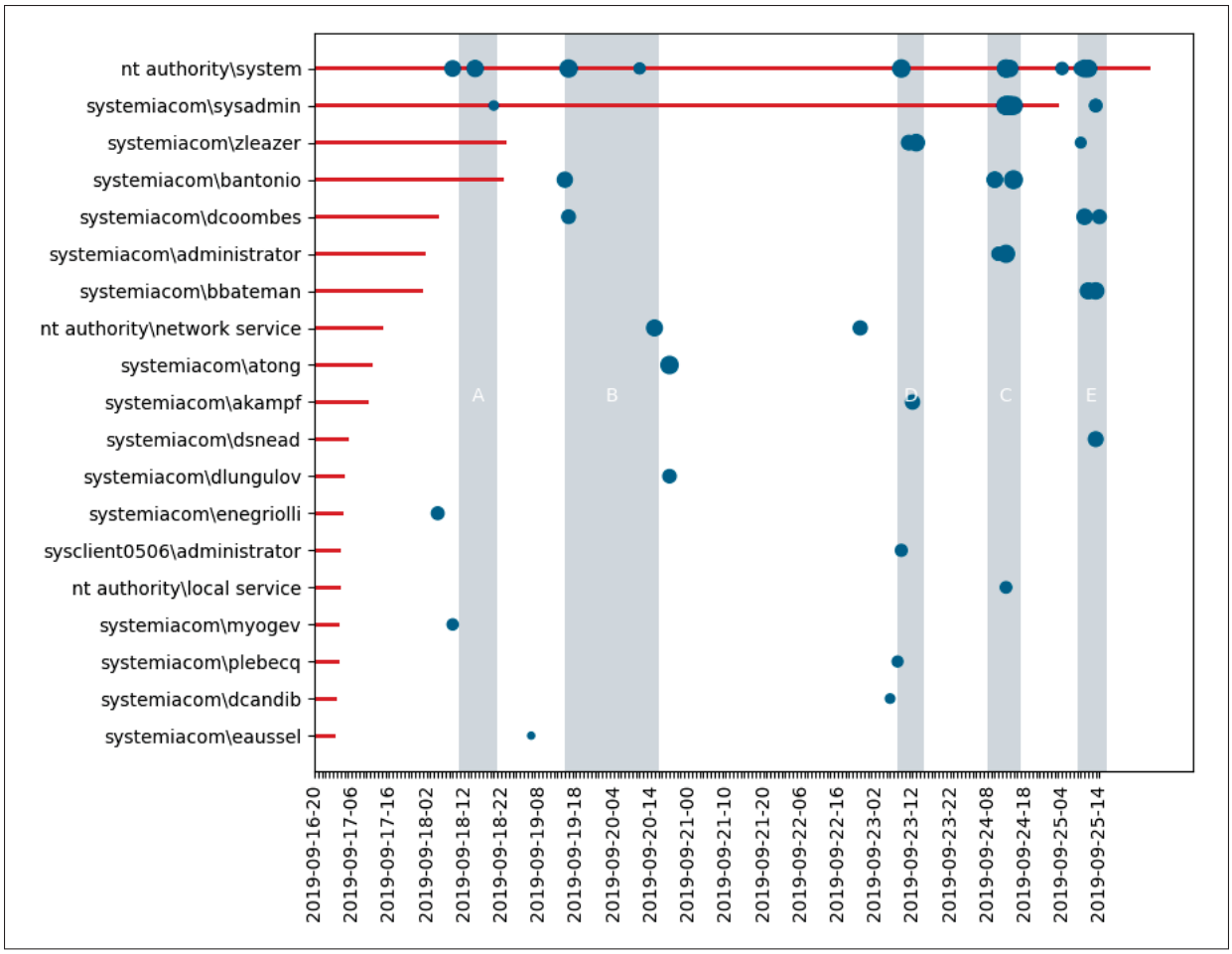

Figure 4.12 Anomalies de ligne de commande regroupées par l'identité de l'utilisateur

L'utilisateur « SYSTEMIACOM\sysadmin » a accumulé particulièrement beaucoup d'anomalies par rapport aux autres identités d'utilisateur. C'est l'identité d'un administrateur global utilisée par les auteurs du jeu de données OpTC pour configurer l'environnement, et elle est associée à 7779 processus exécutés. Autrement dit, il s'agit d'une activité non automatisée, qui peut devenir la source principale de faux positifs si le modèle IDS ne peut pas différencier un administrateur d'un attaquant. Ici, on trouve une grande grappe d'anomalies survenues pendant la deuxième journée de la campagne de l'équipe rouge. On peut donc conclure avec confiance que les attaquants ont réussi à pirater l'identité d'administrateur global du domaine Active Directory, ce qui produit ce signal très fort.

L'Annexe III présente une sélection des commandes les plus suspectes identifiées par le modèle d'analyse d'anomalies, avec des indicateurs clairs d'activité malveillante.

4.4.5 Conclusion

Les données collectées sont suffisantes pour affirmer avec certitude que les anomalies détectées sont des attaques menées par l'équipe rouge.

Cette recherche a permis de concevoir un pipeline d'analyse d'anomalies simple, s'inspirant de l'algorithme TF-IDF, qui nécessite moins de 0.1% du jeu complet. Le design de cette méthode de détection est axé sur les exigences de sécurité. Le système présente un taux de faux positifs faible et génère un nombre d'alertes raisonnable, malgré la grande quantité d'évènements dans le jeu complet.

De plus, les résultats de cette expérience confirment la validité des résultats de la Section 4.3, malgré l'utilisation de deux méthodes très différentes d'analyse d'anomalies non supervisées, qui fonctionnent sur des sous-ensembles distincts du jeu complet.

Dans leur analyse systématique des pièges de l'application de l'apprentissage machine dans le domaine de la cybersécurité, Arp *et coll.* (2022) soutiennent que l'adoption de techniques avancées de détection d'intrusion doit être justifiée en démontrant d'abord l'inefficacité des méthodes de base simples. Les résultats présentés dans cette section et la section précédente peuvent servir de base de référence à toutes les études existantes sur le jeu OpTC qui n'ont pas procédé à une telle analyse.

Les résultats et recommandations de cette section sont les suivants.

- Avec seulement 0,1% du jeu OpTC complet, il est possible d'entrainer un système de détection d'intrusion performant pour identifier les attaques de l'équipe rouge.
- Certains évènements de processus incluent des indicateurs très évidents d'activité malveillante. Dans certains cas, un seul évènement isolé peut même suffire à identifier une attaque.

- En plus des activités d'utilisateur simulées, le jeu OpTC inclut des traces d'activités administratives effectuées par les auteurs originaux du jeu, qui sont également très visibles comme des déviations par rapport à l'activité de base. Ces anomalies peuvent être utilisées pour évaluer la capacité de l'IDS à distinguer les administrateurs des attaquants. Cependant, ces deux classes d'évènements sont, pour la plupart, disjointes dans le temps.
- Pour cette raison, il est recommandé d'étendre la période d'évaluation de trois jours officiellement documentés à huit jours, à partir de « 2019-09-18 ».

4.5 Expérience 3 : transcriptions des scripts PowerShell

PowerShell (Microsoft, 2006) est un langage de script très puissant et omniprésent dans toutes les versions de Windows. Selon plusieurs rapports de l'industrie, c'est une technique constamment utilisée par tous les acteurs malveillants majeurs (CrowdStrike, 2023; Huntress, 2023; Red Canary, 2024; Huntress, 2025).

Une étude préalable de ce jeu de données (Anjum *et coll.*, 2021) a déjà établi qu'en comparaison avec le jeu complet, les évènements malveillants étaient disproportionnellement plus fréquents parmi des journaux de type « SHELL ». En fait, les résultats de la section précédente contiennent déjà plusieurs exemples d'exécutions de processus PowerShell suspectes, ce qui indique les utilisateurs compromis et des scripts malveillants ont été déjà détectés.

4.5.1 Description de l'environnement

Les évènements d'exécution PowerShell dans le jeu OpTC sont représentés par l'objet de type « SHELL ». On compte seulement 446,933 journaux de ce type, ce qui est deux ordres de grandeur moins que les journaux d'exécution de processus. Ils ne représentent qu'une petite partie du jeu complet, et près de la moitié d'entre eux sont associés à des tâches système, ce qui renforce davantage les signaux potentiels d'anomalie.

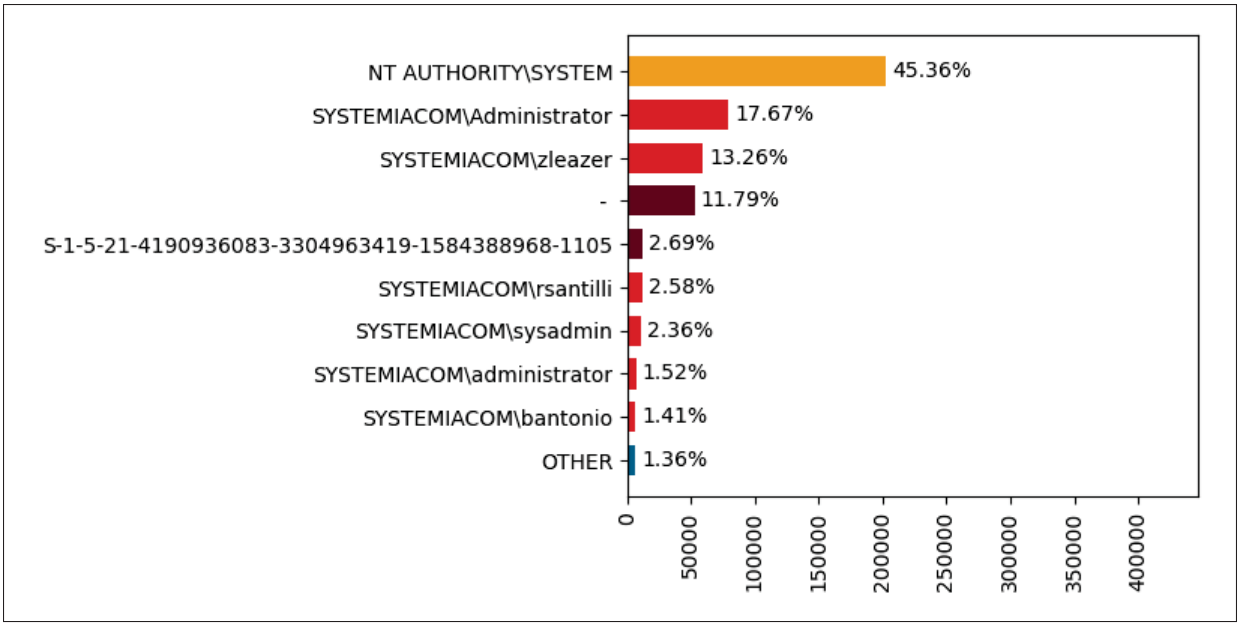

Figure 4.13 Le nombre total de tous les journaux d'exécution de shell, regroupés par la valeur du champ « principal »

Contrairement aux autres types de journaux, les journaux « SHELL » ne contiennent pas le champ « user ». Il faut donc utiliser la valeur du champ « principle » pour attribuer les évènements à une identité d'utilisateur. Malheureusement, ce champ est rempli de manière incohérente. Comme le montre la Figure 4.13, il est complètement vide dans environ 12% des cas, tandis qu'il est associé à un SID non résolu pour environ 2,7% des évènements (cette valeur correspond à l'utilisateur « sysadmin »). Près de la moitié de tous les journaux sont attribués à l'utilisateur « SYSTEM », probablement en raison de la simulation de l'environnement. Les anomalies de comportement des utilisateurs sont également très visibles sur le graphique. Plusieurs d'entre eux sont déjà connus grâce à l'analyse précédente. Tous les autres utilisateurs n'ont presque aucun évènement d'exécution de PowerShell.

Les journaux de l'objet SHELL sont différents de ceux des autres objets du jeu OpTC d'une manière importante : une seule entrée de journal ne correspond pas à une seule exécution de script. Au contraire, chaque déclaration dans le script produit une entrée distincte. Par exemple, l'exécution d'un script malveillant par l'utilisateur « SYSTEMAICOM\zleazer » sur

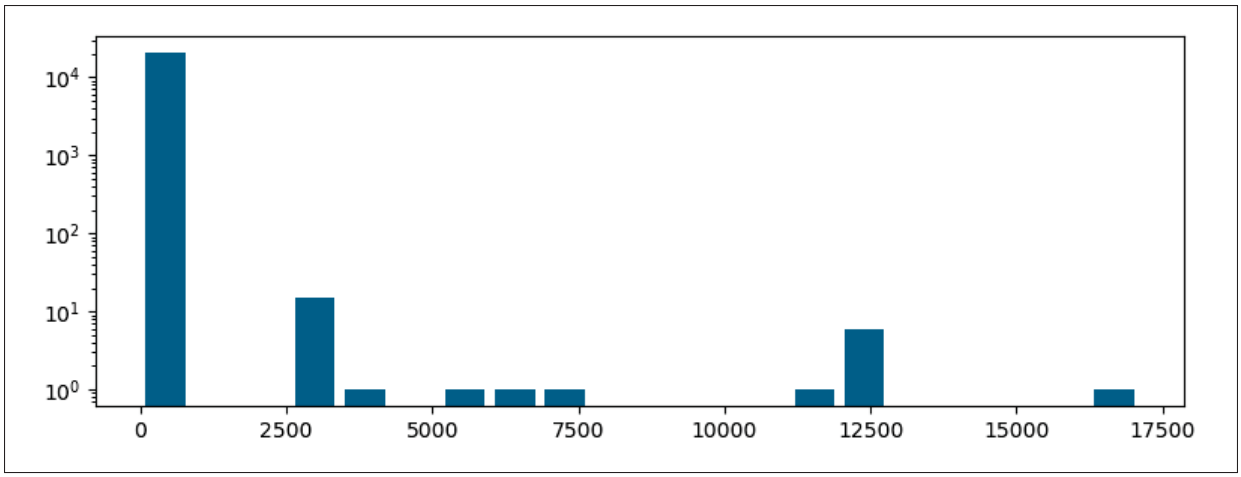

Figure 4.14 Un histogramme du nombre d'évènements associés à l'exécution de scripts PowerShell

« 2019-09-23T11 :24 :02.153-04 :00 » a produit 274 entrées de journal. L'histogramme de la Figure 4.14 illustre le nombre d'entrées par exécution de processus PowerShell. La plupart des scripts génèrent seulement un seul journal, mais les scripts plus complexes produisent plusieurs entrées avec des informations partielles. Un petit nombre d'exécutions de PowerShell anormales ont produit plusieurs milliers d'évènements, avec le plus grand nombre atteignant 17,098 évènements.

Puisque chaque entrée de journal contient seulement des informations partielles, une étape de prétraitement unique à ce type de journal est nécessaire pour reconstituer le script malveillant dans son intégralité. Cela intègre la logique métier dans le pipeline d'entrainement, ce qui renforce les doutes quant à la capacité de l'apprentissage machine à créer un IDS généralisable capable de fonctionner directement sur des données brutes (Verkerken, D'hooge, Wauters, Volckaert et De Turck, 2022; Cantone, Marrocco et Bria, 2024).

D'un autre côté, tenter de reconstituer le contexte complet d'un script malveillant peut entrainer d'autres limitations. Dans le cas du jeu OpTC, le script PowerShell qui contient le plus d'évènements est clairement anormal et évidemment un processus malveillant. Cependant, la compilation de toutes les entrées de journal produit plus de 35 Mo de données textuelles. La

recherche manuelle de sections de code pertinentes pour expliquer le comportement malveillant du script est extrêmement risquée. De plus, ce genre d'alerte ne serait pas utile pour un analyste de la sécurité dans le cadre d'une enquête sur un incident. Cela illustre bien qu'il ne suffit pas de repérer le comportement malveillant. Une alerte de sécurité doit être explicable et actionnable. Une alerte de sécurité doit permettre une réponse appropriée, avec une explication claire.

En outre, l'ensemble de données comprend six (6) instances de processus PowerShell qui s'exécutent en continu pendant plus de 20 heures, comme le montre la Figure 4.15. Les 101,087 évènements produits par ces neuf processus représentent 22% de tous les journaux « SHELL » du jeu. Tous les évènements se produisent pendant la période d'activité de l'équipe rouge. Deux modèles de comportement distincts des processus peuvent être observés : (a) un compromis initial suivi d'un mouvement latéral, indiqué par des pics courts d'activité intense ; (b) des processus de backdoor persistants montrant le comportement « low and slow ».

C'est un exemple classique de comportement « low and slow », souvent considéré comme la caractéristique principale des attaques APT (Advanced Persistent Threat). Un IDS qui ne surveille les scripts qu'après la fin de l'exécution du processus PowerShell ne serait pas en mesure de détecter ce genre de processus de backdoor. Une stratégie efficace pour faire les détections au bon moment nécessiterait un équilibre entre le suivi de l'état de l'environnement et l'analyse des évènements récemment observés. Cela montre clairement l'avantage offert par les réseaux de mémoire longue à court terme (LSTM) pour cette tâche, ce qui explique leur popularité (Du, Li, Zheng et Srikumar, 2017; Zhang *et coll.*, 2019; Afnan, Sadia, Iqbal et Iqbal, 2023; King et Huang, 2023; Villarreal-Vasquez, Modelo-Howard, Dube et Bhargava, 2023).

4.5.2 Modèle non supervisé de détection d'anomalies

Contrairement aux évènements liés à l'exécution des processus, les journaux de PowerShell possèdent une caractéristique importante : l'autodocumentation. Autrement dit, il est possible de comprendre le but et les conséquences d'un script simplement à partir de son code source.

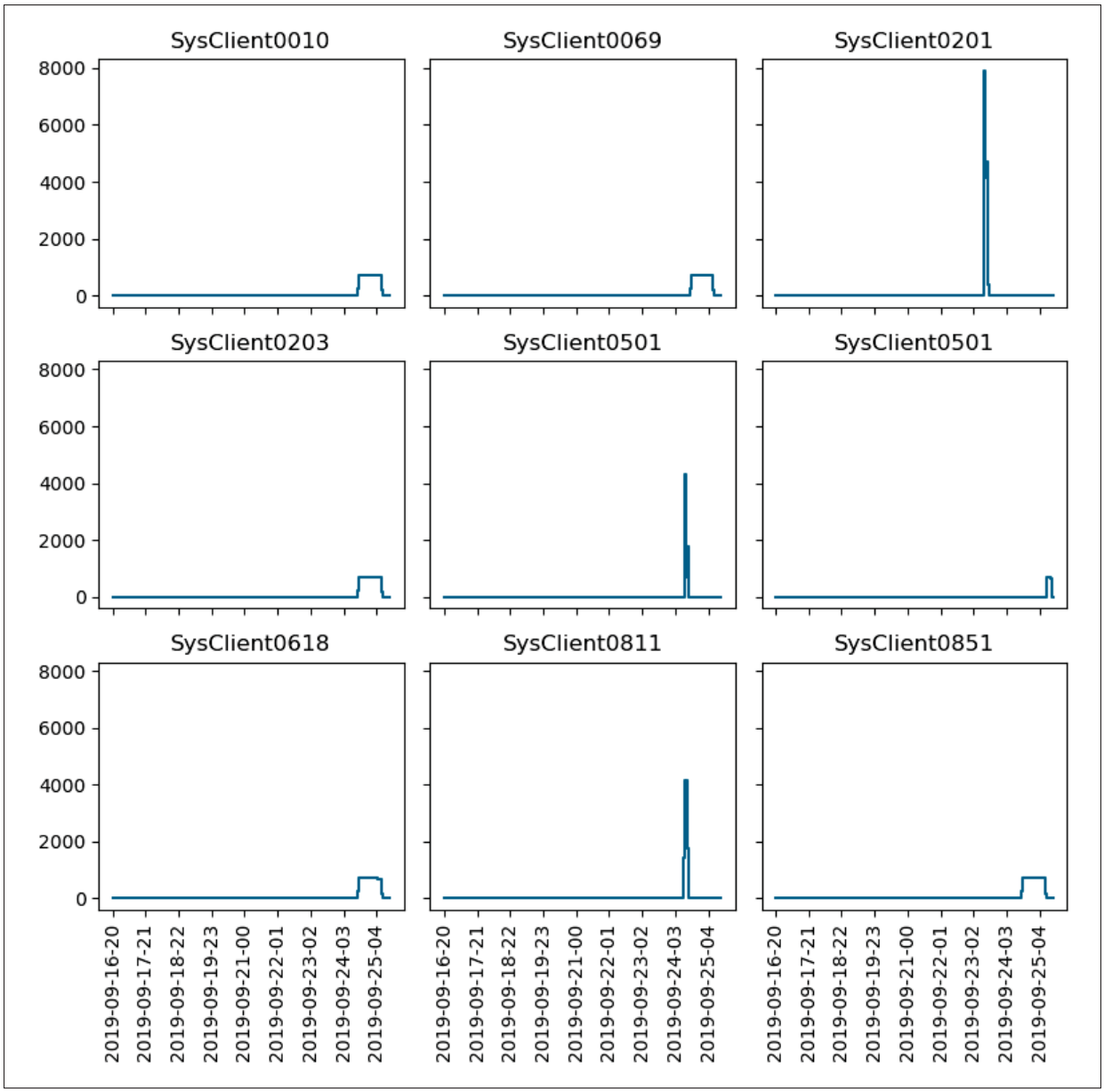

Figure 4.15 Chronologie des journaux d'exécution de scripts observés depuis les neuf (9) processus PowerShell seulement dans tout le jeu de données qui sont exécutés en continu pendant plus d'une heure

```
CommandInvocation(ForEach-Object): "ForEach-Object"
ParameterBinding(ForEach-Object): name="Process";
                                  value="$wc.Headers.Add($_.Name, $_.Value)"
```

Figure 4.16 Un exemple du format de journal linéaire des déclarations PowerShell dans le champ « payload »

Pour ce faire, il est nécessaire d'activer le journal complet des scripts PowerShell, une pratique déjà reconnue comme bonne et une exigence de sécurité dans la plupart des environnements.

La classification d'une connexion réseau comme malveillante ou non dépend entièrement du contexte plus large du réseau, mais les scripts malveillants sont généralement malveillants dans n'importe quel environnement. Donc, la tâche de les identifier est plus similaire à la détection de maliciel que de l'intrusion. C'est un domaine de recherche prospère, avec des centaines d'études utilisant des méthodes innovantes. Cependant, il peut être considéré comme un problème largement résolu par les principaux fournisseurs de solutions EDR, qui assurent un taux élevé de détection.

Cependant, dans le contexte de la détection d'intrusion, il faut traiter les évènements d'exécution de script comme un indicateur parmi plusieurs autres d'un comportement suspect d'utilisateur.

L'analyse du champ « payload » des évènements « SHELL » a révélé un format de journalisation intégré qui semble être généré par la fonctionnalité de journalisation du module de PowerShell. Ces journaux linéaires non structurés capturent les détails des déclarations de scripts individuelles. Un exemple représentatif est présenté à la Figure 4.16. Chaque ligne se compose de trois parties principales : le type d'opération, le nom de l'opération et une liste aléatoire de paramètres.

La présence de cette structure de logique imbriquée permet de construire un ensemble de caractéristiques sémantiquement riche pour concevoir un modèle de classification. Cependant, cela confirme que les journaux « SHELL » sont très différents, structurellement et sémantiquement, des autres évènements dans l'ensemble de données, et qu'ils doivent être traités séparément.

Tableau 4.6 Caractéristiques des commandes PowerShell pour l'algorithme Isolation Forest

Caractéristique	**Description**
Type de l'entrée	Codage ordinal de valeur dans un ensemble fermé.
Nom de l'opération	Pour éviter le problème de mots hors vocabulaire, la valeur moyenne des caractères dans la chaine est calculée comme $\frac{\sum^N ord(c)}{N}$
Longueur du paramètre	Longueur du reste de la chaine après le nom de l'opération. Pour atténuer les effets de scripts très longs, la fonction logarithmique est utilisée.
Entropie du paramètre	L'entropie de la chaine restante après le nom de l'opération.

Cette situation rend encore plus complexe la conception d'un IDS universel hypothétique qui pourrait utiliser les journaux bruts et générer des alertes explicables.

L'analyse de déclarations individuelles du champ « payload » est suffisante pour détecter des activités malveillantes comme des anomalies flagrantes. Ce n'est pas surprenant, étant donné que 25% à 50% de tous les évènements « SHELL » sont malveillants. L'Annexe IV fournit une liste complète des opérations PowerShell comme une preuve de la puissance du signal généré par l'engagement de l'équipe rouge.

Pour cette expérience, l'exigence de détection sera définie comme suit :

> *Détecter de l'exécution de scripts PowerShell suspects qui sont atypiques pour cet environnement, et fournir à l'analyste en sécurité une preuve compacte et pertinente.*

Tout comme dans les expériences antérieures, l'entrainement se fera entièrement en mode non supervisé, sans aucune étiquette. Pour éviter que l'analyste soit submergé par de fausses alertes, le nombre de résultats doit être « raisonnablement faible », c'est-à-dire en dessous d'une limite maximale N, que l'analyste peut contrôler.

La détection de scripts malveillants PowerShell a suscité un vif intérêt, à la fois de la communauté universitaire et de l'industrie, mais la plupart des recherches actuelles s'orientent vers des

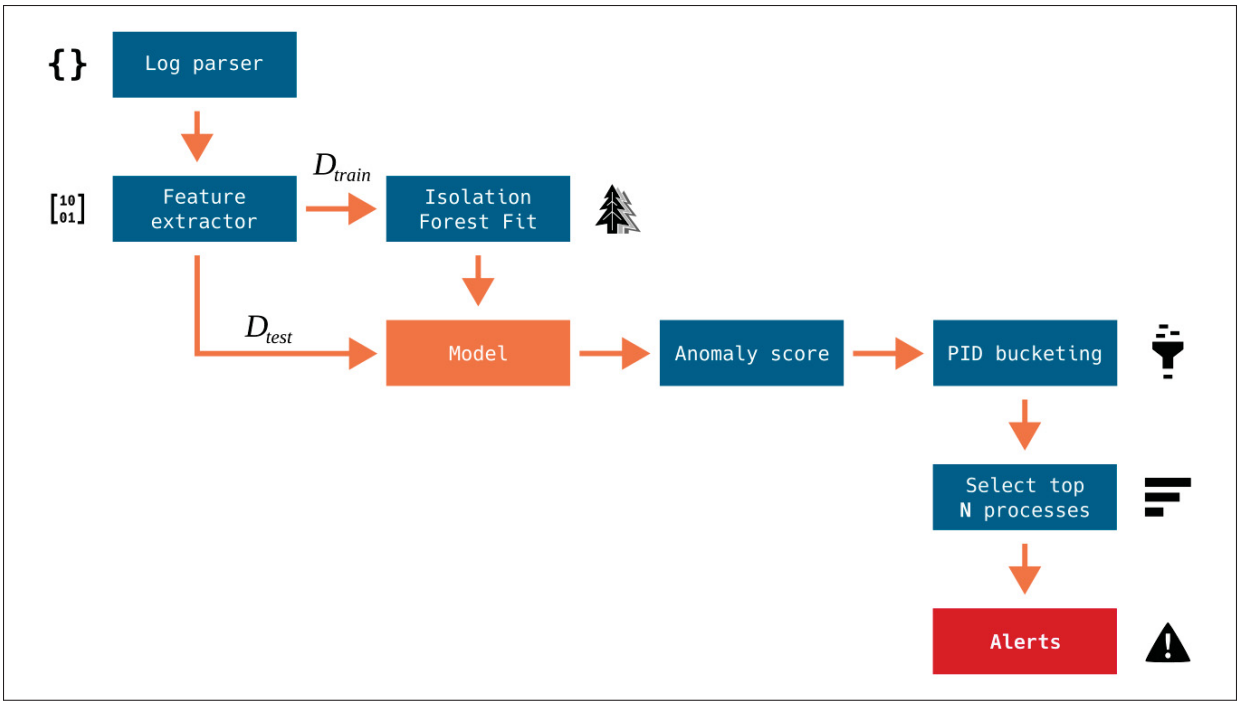

Figure 4.17 Un pipeline de détection d'anomalies de scripts PowerShell

méthodes supervisées, telles que les réseaux neuronaux à convolution (Hendler, Kels et Rubin, 2020) et les approches par LSTM (Yang, Peng, Zhang, Gao et Li, 2023). Bien que ces méthodes semblent prometteuses, elles nécessitent une quantité importante d'étiquettes, ce qui n'est pas encore disponible. Une étude comparative des méthodes de détection malveillante fondées sur l'intelligence artificielle Song, Kim, Choi, Kim et Kim (2021) a montré que les algorithmes de forêt aléatoire (*random forest*) permettent d'obtenir les meilleurs résultats avec un faible taux de fausses alertes sur des données réelles.

L'Isolation Forest (Liu, Ting et Zhou, 2008) est un algorithme qui est conceptuellement semblable à la forêt aléatoire, mais il offre une détection d'anomalies non supervisée, ce qui en fait un choix parfait pour ce cas d'utilisation. Il fait partie de la collection « scikit-learn » (Pedregosa *et coll.*, 2011), ce qui permettra de l'utiliser dans cette expérience.

Les résultats présentés dans la Section 4.4.5 indiquent qu'il est nécessaire d'élargir la période d'évaluation au-delà de l'activité documentée par l'équipe rouge. Il faut diviser le jeu de données

complètes en deux sous-ensembles : un ensemble d'entrainement et un ensemble de tests, de la manière suivante :

- la période d'entrainement inclut tous les évènements entre « 2019-09-16 » et « 2019-09-17 » ;
- la période de test commence en « 2019-09-18 » à minuit.

Puisque l'attaquant a un contrôle total sur le contenu des scripts, il est crucial de choisir une représentation robuste des caractéristiques qui fonctionne bien pour les scripts PowerShell très courts et très longs, et qui évite le problème des mots hors vocabulaire. Le champ « payload » est divisé en déclarations individuelles, qui sont ensuite transformées en quatre (4) caractéristiques en utilisant une combinaison de codage ordinal pour les termes fixes, de valeur moyenne de caractère et d'entropie de la chaine. La description détaillée des caractéristiques est fournie dans le Tableau 4.6.

Le pipeline complet de détection d'anomalies est présenté dans la Figure 4.17. Les données d'entrée sont divisées en un ensemble d'apprentissage et un ensemble d'évaluation. L'ensemble d'apprentissage est utilisé pour entrainer un algorithme d'Isolation Forest non supervisé. Ce modèle est ensuite utilisé pour identifier les processus PowerShell suspects en fonction de la note d'anomalie attribuée à chaque déclaration de script individuelle.

4.5.3 Analyse des anomalies détectées

Les résultats du modèle permettent de faire plusieurs observations importantes. Tout d'abord, les scripts PowerShell portent un signal fort d'activité malveillante dans cet environnement, comme le montre la Figure 4.18. Chaque déclaration de script est évaluée de manière indépendante et les résultats du modèle de tous les évènements dans le même script sont additionnés, ce qui produit une série temporelle de toutes les activités suspectes dans le réseau. Les anomalies s'alignent parfaitement avec les régions d'activité anormale découvertes lors des analyses précédentes À ce stade, l'évidence est suffisante pour conclure avec confiance que l'équipe rouge utilise

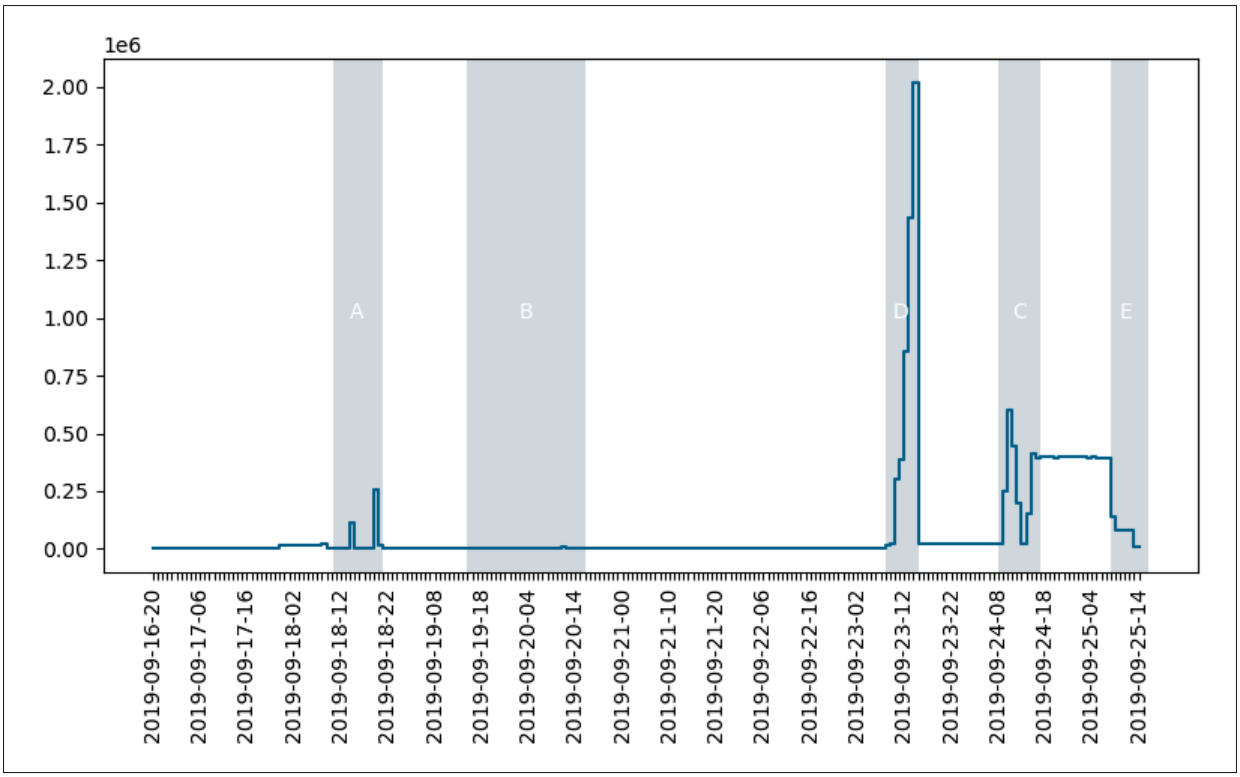

Figure 4.18 Une distribution des anomalies de tous les scripts PowerShell

PowerShell pour mener ses attaques, ce qui les distingue clairement du comportement de base du réseau.

En particulier, l'équipe rouge utilise le cadre Empire (EmpireProject, 2018), un cadriciel de postexploitation conçu pour simuler les tactiques et techniques utilisées par les acteurs APT, comme le contrôle à distance et le chiffrement du trafic. Il existe des règles statiques bien testées pour détecter le cadriciel Empire (SigmaHQ, 2024), mais elles dépendent de la présence de chaines spécifiques des caractères. D'un autre côté, la méthode sélectionnée est simple, mais elle permet de détecter les logiciels malveillants PowerShell qui utilisent des techniques d'obfuscation.

En examinant toutes les invocations du cadriciel Empire, il est évident que la première apparition dans les journaux se produit sur « 2019-09-20 », dans un évènement d'exécution de processus avec ID « 38214dab-b5c8-42a4-b795-d4345d687cba », effectué par l'utilisateur

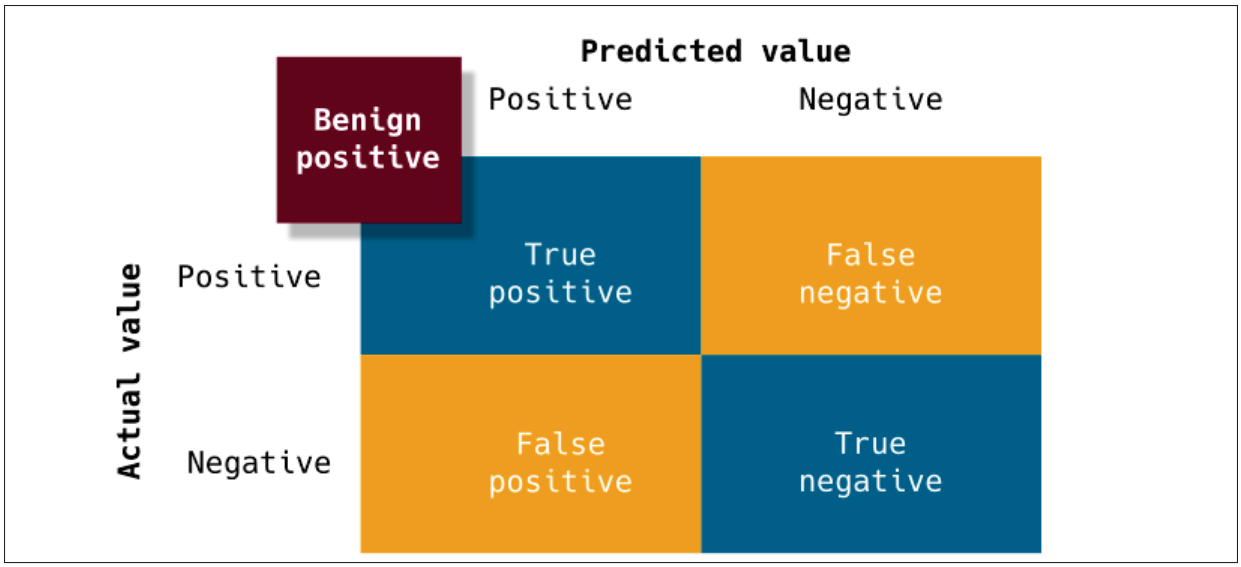

Figure 4.19 La matrice de confusion augmentée, adoptée par de nombreuses équipes de sécurité pour pallier le manque de contexte d'entreprise dans les IDS modernes

« SYSCLIENT0124\Administrator ». Cet évènement doit être considéré comme malveillant. Toutefois, il a eu lieu trois jours avant le début officiel de l'engagement de l'équipe rouge.

La présence de cet évènement illustre bien les défis auxquels font face les techniques d'apprentissage automatique en matière de cybersécurité. Il est impossible de garantir que les données d'entrainement soient exemptes d'activités malveillantes, comme dans le cas d'OpTC. Cela pose le risque que le modèle apprenne à classer ces évènements comme étant bénins dans l'avenir.

Cela pose également une question plus large sur la manière de classer correctement cet évènement. Est-ce un faux positif, car il n'apparait pas dans la réalité de terrain documenté, ou est-ce un vrai positif, car il représente une menace réelle pour le réseau ? D'après les informations disponibles à l'IDS, un évènement dangereux peut survenir de la même manière lorsqu'il est exécuté par un administrateur système légitime ou par un acteur malveillant. La décision de l'analyste repose souvent sur des éléments externes inaccessibles pour les systèmes IDS traditionnels, tels qu'un ticket de support approuvé. La prévalence de ce problème a conduit de nombreux experts de

l'industrie à ajouter une cinquième catégorie à la matrice de confusion traditionnelle, soit bénin positif. Cela signifie que l'IDS a correctement identifié l'évènement suspect, mais qu'il n'a pas donné lieu à un incident de sécurité en raison du contexte plus large de l'entreprise.

Bien que cet outil ne repose pas sur une base mathématique solide, il est de plus en plus utilisé par les équipes SOC modernes pour gérer la complexité de la détection d'intrusion. La présence des bénins positifs dans OpTC confirme le potentiel de ce jeu de données pour répondre aux besoins réels de l'industrie en recherche académique.

Les scripts PowerShell représentent une fraction minime des données de journaux dans l'ensemble de données OpTC. Cependant, considérés séparément, ils forment leur propre ensemble de données, avec un format intégré, des caractéristiques pour l'apprentissage et un bon équilibre entre les évènements malveillants et les évènements bénins. Malheureusement, ce type de journal émet le signal d'activité malveillante le plus fort, qui peut être facilement détecté à l'aide de méthodes simples de base.

L'analyse des prédictions du modèle confirme les découvertes antérieures. Le jeu de données OpTC reflète certains aspects des contraintes des réseaux corporatifs réels, ce qui rend difficile la détection d'intrusion et fait hésiter les entreprises à adopter les IDS académiques, notamment la présence de bénins positifs.

4.6 Résultats

Les expériences de ce chapitre ont permis de tirer quelques observations importantes sur ce jeu de données.

Résultat : La nature confinée de l'ensemble de données entraine un comportement de base très prononcé.

Il n'est pas surprenant qu'un jeu de données synthétiques, créé à partir d'un petit nombre de scénarios prédéfinis, n'entraine pas beaucoup de variation dans le comportement du système. Après la mise en place initiale, la dérive conceptuelle ne se produit pas pendant plusieurs

jours, ce qui permet de construire presque un modèle parfait de l'environnement. Cela facilite grandement la détection des anomalies causées par l'équipe rouge. En utilisant la méthode classique d'apprentissage machine, qui consiste à diviser les données en jeux de tests (les trois derniers jours d'activité de l'équipe rouge) et en jeux d'entrainement (toutes les périodes antérieures), on risque d'obtenir un taux de faux positifs artificiellement bas, car les données ne contiennent pas d'anomalies non malveillantes. C'est particulièrement le cas pour les réseaux neuronaux puissants, qui sont actuellement favorisés par les chercheurs. Cependant, lorsqu'un modèle est transféré dans un environnement ouvert, il peut arriver qu'il produise trop de fausses alarmes en raison d'anomalies bénignes, sans fournir de valeur à l'équipe de sécurité.

Résultat : L'activité de l'équipe rouge génère un signal anormal fort qui peut être détecté par des méthodes de base simples.

Comme une conséquence directe du résultat précédent, il est possible de détecter l'activité malveillante dans le jeu de données OpTC à l'aide de techniques simples. La création de réseaux neuronaux profonds ou récursifs complexes n'est donc pas requise pour cette tâche. De plus, puisque l'utilisateur du jeu de données peut contrôler les étiquettes, il y a beaucoup de souplesse pour définir ce qu'est une « détection réussie » : évènements individuels, processus, sessions d'utilisateurs, hôtes, périodes de temps, etc. Cela permet d'adapter les données en fonction des méthodes et des objectifs en matière de sécurité pour améliorer le taux de détection, qui est la seule chose importante, finalement [1]. Par contre, cela rend impossible la comparaison directe des méthodes de détection dans différents articles.

Résultat : L'activité anormale ne se limite pas à la période de l'engagement de l'équipe rouge.

[1] La raison pour laquelle les équipes SOC accordent une grande importance aux faux positifs plutôt qu'aux faux négatifs (Vermeer *et coll.*, 2023) est que les méthodes de détection d'intrusion ne doivent pas être parfaites. Elles doivent simplement identifier quelques actions malveillantes, même si elles échouent parfois (Mink *et coll.*, 2023). Il suffit d'avoir une seule alerte de haute qualité pour que l'attaque soit détectée et que le processus de réponse aux incidents soit déclenché. Cette caractéristique est souvent appelée « l'avantage de défenseur » dans l'industrie.

Le jeu de données OpTC contient des anomalies non malveillantes qui ressemblent à celles de l'équipe rouge. Il est inconnu si c'est fait exprès ou non, l'équipe responsable du projet ait mené une administration réseau pendant que la collecte de données était en cours. Cela a entrainé des irrégularités significatives dans les opérations scriptées de base. L'activité administrative constitue la principale cause de fausses alertes dans la plupart des réseaux réels. Par conséquent, sa présence dans l'ensemble de référence permet d'évaluer de manière réaliste la capacité de l'IDS à distinguer les anomalies malveillantes des anomalies bénignes.

Cependant, dans le cas de OpTC, ces tâches administratives bénignes sont observées au début de la chronologie, pendant la phase de configuration de l'environnement. Les actions malveillantes, en revanche, sont contenues dans les trois jours d'activité de l'équipe rouge à la fin de l'expérience. Cela signifie qu'il ne serait pas optimal d'utiliser les méthodes d'apprentissage machine standards pour diviser les données en ensembles d'entrainement et de tests. Le temps d'évaluation devrait être prolongé pour inclure la période avec des anomalies bénignes. Les quatre premiers jours de données sont suffisants pour construire un modèle fiable de l'environnement, et les journées restantes devraient être utilisées pour l'évaluation.

Résultat : L'activité malveillante présente dans le jeu est ambigüe.

L'examen des données a pu identifier plusieurs évènements qui ne pouvaient pas être catégorisés comme étant à la fois bénins ou malveillants, car ils présentaient des signes évidents d'activité suspecte, mais n'étaient pas mentionnés dans la réalité du terrain de l'équipe rouge. Cela ajoute de la complexité à la comparaison des travaux qui utilisent le jeu OpTC, puisque les auteurs peuvent gérer les évènements différemment, ce qui peut affecter les indicateurs de performance.

Parmi cela, l'évènement avec l'identifiant « 38214dab-b5c8-42a4-b795-d4345d687cba » est particulièrement intéressant, car il correspond à l'exécution du même script malveillant que l'équipe rouge utilise, mais par un administrateur système trois jours avant le lancement officiel de la campagne. Dans le contexte d'un réseau réel, cet évènement constitue un indicateur de compromis clair qui doit être traité comme un vrai positif si repéré par un système de détection

d'intrusion. Examiné dans le cadre du contexte OpTC, cet incident est un test inoffensif sans aucun lien avec l'équipe rouge. Il peut être considéré comme un faux positif s'il est détecté par l'IDS.

L'existence de ces évènements ambigus représente les défis auxquels les équipes d'opérations de sécurité sont confrontées. Cela montre pourquoi la détection d'intrusions ne peut pas se faire en isolation d'un contexte plus large de l'entreprise, et met en évidence la nécessité d'alertes de détection d'intrusion justifiables et explicables.

4.7 Conclusion

Ce chapitre a proposé une analyse détaillée des données du jeu OpTC et créé des modèles de base pour détecter le comportement malveillant. Pour évaluer la capacité de ce jeu de données à représenter un environnement d'entreprise réel, une méthode typique utilisée par les analystes en sécurité lorsqu'ils travaillent avec des données inconnues a été adoptée.

Pour éviter tout biais dans la classification du comportement malveillant, toute analyse est effectuée sans l'utilisation directe de la description de la réalité du terrain fournie par l'équipe rouge. Autrement dit, ces données ont été traitées comme un ensemble non étiqueté, ce qui a limité le choix des méthodes disponibles entre l'analyse statistique et les algorithmes d'apprentissage automatique non supervisé.

Pour mesurer l'intensité du signal, le jeu de données complet a été divisé en sous-ensembles selon le type d'objet dans les journaux. Les trois objets de journal les plus fréquemment collectés dans un réseau d'entreprise typique sont les journaux d'authentification des utilisateurs, les évènements d'exécution de processus et les transcriptions de PowerShell. L'utilisation des sous-ensembles de journaux permet d'adopter la même approche qu'une équipe de sécurité informatique ayant des ressources limitées. Cela permet de créer un système de détection d'intrusion réaliste et applicable dans un contexte réel.

Pour produire des alertes explicables, le système de détection d'anomalie est développé à partir des cas d'utilisation spécifiques dérivés du modèle de menace choisi. Par conséquent, ce système modèles présentent des caractéristiques à la fois des IDS par signature et des IDS par anomalie, ce qui permet d'enrichir les alertes avec le contexte nécessaire. De plus, en tant que mécanisme de sécurité pour empêcher un flot de fausses alertes, les résultats de prédiction sont limités en fonction du score de risque. Cela donne à l'analyste de sécurité un outil pour gérer sa propre charge de travail d'enquête.

QR1 : Est-ce que le jeu de données OpTC est une simulation réaliste d'une entreprise moderne ?

L'expérience OpTC reproduit un domaine Active Directory Windows typique, fréquemment rencontré dans le milieu professionnel. L'activité simulée correspond aux tâches quotidiennes des employés (p.g., navigation sur internet, téléchargement de fichiers, lecture et rédaction de documents), effectuées à l'aide de logiciels réels. Ceci donne lieu à un ensemble riche et réaliste d'artéfacts pouvant servir au développement de méthodes de détection d'anomalies. De plus, les évènements sont encodés en JSON et sont présentés sous la forme convenable pour l'utilisation avec un SIEM moderne. En comparaison avec les jeux de données IDS traditionnels, OpTC se distingue nettement en étant beaucoup plus représentatif d'un réseau appartenant à une entreprise de taille moyenne. Cela peut contribuer à la création de systèmes de détection d'intrusion pratiques.

QR2 : Est-ce que le comportement de base peut être caractérisé comme étant confiné ou ouvert ?

OpTC représente un environnement confiné, et son état complet est fixé depuis le début de la simulation. Les actions des utilisateurs sont simulées grâce à un petit nombre d'interactions prédéfinies, partagées par tous les utilisateurs, qui sont très routinières et faciles à inclure dans un modèle de base. En utilisant les évènements des premiers jours de la simulation, il est possible de construire une représentation parfaite de l'environnement, ce qui ne serait pas possible dans un environnement ouvert. Les nouvelles entités, comme les comptes d'utilisateurs, créées pendant

l'engagement, peuvent être attribuées à l'activité de l'équipe rouge, ce qui rend la détection très facile.

QR3 : Qu'est-ce qui empêche la communauté scientifique de l'utiliser davantage ?

L'ensemble de données OpTC présente plusieurs défis pratiques, le plus important étant le volume de données, ce qui rend son analyse une tâche chronophage. De nombreux outils couramment utilisés par la communauté de l'apprentissage automatique sont conçus pour fonctionner avec des données CSV, mais le jeu de données OpTC consiste en des évènements en format JSON, qui reflètent les environnements réels. Mais ce qui est le plus important, c'est que la réalité du terrain fournie par l'équipe rouge ne correspond pas aux évènements du jeu de données, ce qui rend l'utilisation des techniques d'apprentissage automatique supervisé difficile. Convertir la réalité du terrain en étiquettes exige des compétences en sécurité et une compréhension approfondie des attaques de l'équipe rouge, ce qui rend ce jeu de données inaccessible à de nombreux spécialistes des données. De plus, car l'interprétation des journaux système est un processus subjectif, différentes équipes arrivent à des étiquettes de données différentes, ce qui entraine des métriques de performance incompatibles.

QR4 : Comment les étiquettes sont-elles présentées dans les données ?

Étant donné la nature de l'engagement de l'équipe rouge, il n'y a pas de correspondance directe entre les actions des attaquants et les journaux bruts. Ainsi, considérée isolément, l'activité malveillante ne contient pas de champs indiquant qu'elle a été causée par l'équipe rouge. Cependant, après avoir analysé les données, il est devenu clair que certaines actions de l'équipe rouge portent des indices clairs de compromis qui pourraient être facilement détectés à l'aide de signatures statiques ou provoquer des corrélations trompeuses avec les méthodes d'apprentissage machine.

De plus, certaines actions malveillantes ont été effectuées avant le début officiel de la campagne de l'équipe rouge, ce qui pourrait être pour vérifier que l'environnement était correctement configuré. Cela a deux implications majeures. Premièrement, il ne faut pas présumer que les

données d'entrainement sont exemptes d'activités malveillantes (ce qui n'est jamais garanti dans un environnement réel), ce qui pourrait influencer les performances du modèle. Deuxièmement, il est impossible de distinguer une activité de test d'une attaque réelle en l'absence de description détaillée de la réalité du terrain. Si ces évènements sont détectés par un IDS, la réponse n'est pas évidente : s'agit-il de vrais positifs ou de faux positifs ? Cette incertitude rend la comparaison des différentes méthodes d'IDS encore plus complexe.

En conclusion, l'ensemble de données OpTC offre une image plus authentique des obstacles rencontrés par les équipes de sécurité dans les environnements modernes. En raison du format de données détaillé et du signal malveillant fort de ce jeu de données, la détection peut apparaitre facile, car les systèmes de détection d'intrusion peuvent aisément atteindre de bons indicateurs de performance. Cependant, la valeur réelle du jeu de données OpTC réside dans certaines caractéristiques qui rendent difficile la mise en œuvre de tels systèmes simples dans un environnement industriel. Pour concevoir un IDS pragmatique, il est crucial de prendre en compte tous les aspects de ce jeu de données. Ces éléments devraient être expressément incorporés dans l'architecture du système.

CHAPITRE 5

IMPLÉMENTATION D'UN CADRE IDS NOVATEUR

Ce chapitre propose une implémentation d'un cadre IDS fondé sur la notion de provenance. Ce cadre est novateur et s'inspire de la structure en couches d'un SOC typique. Il repose sur les réflexions antérieures concernant les défis posés par la cybersécurité des entreprises modernes ainsi que sur les limites des systèmes de détection d'intrusion actuels. Le cadre proposé place les exigences opérationnelles au cœur du processus, simplifiant ainsi sa mise en œuvre dans l'entreprise.

5.1 Introduction

Le Chapitre 2 a discuté la capacité des systèmes de suivi de provenance pour détecter les attaques avancées, qui peuvent se dérouler pendant une longue période sans être détectées. Ils ont été particulièrement efficaces pour identifier le comportement « low and slow », souvent associé aux acteurs APT, dont l'objectif est de rester dans le réseau pendant des semaines et des mois après la compromission initiale (Mandiant, 2023; Mandiant et Google Cloud Security, 2024). Cependant, de nombreux défis persistent, tant pour le développement d'un système de détection d'intrusion évolutif que pour sa mise en œuvre pratique dans un centre de sécurité opérationnelle (SOC).

L'un des défis consiste à gérer la masse de données contenue dans un graphe de provenance, ce qui découle directement de sa propriété fondamentale et souhaitée de complétude. Pour que ces graphes soient utiles pour la détection d'intrusion en temps réel, il faut réussir à les traiter assez rapidement. La question de savoir comment y parvenir de manière efficace est toujours ouverte.

Comme plusieurs algorithmes graphiques classiques ont une complexité polynomiale (ou pire) (Hamilton; Chiang *et coll.*, 2019), l'analyse de ces grands réseaux est couteuse et irréalisable. Cela a inspiré plusieurs auteurs à explorer différentes méthodes d'optimisation, telles que l'utilisation d'euristiques pour sélectionner un sous-ensemble d'arcs (Hassan *et coll.*, 2019; Cheng *et coll.*, 2024), la suppression dynamique d'arcs (Zhu *et coll.*, 2023; Xie *et coll.*, 2020),

l'utilisation de diverses techniques de plongement de nœuds (Anjum *et coll.*, 2022; Goyal *et coll.*, 2024) ou l'accélération grâce au cache (Ma *et coll.*, 2018; Ur Rehman *et coll.*, 2024). Certaines de ces conceptions sont généralisables, alors que d'autres dépendent fortement des spécificités des ensembles de données. Quelle que soit la technique, toutes les approches existantes nécessitent un compromis entre la complétude du graphe et la rapidité d'exécution. Cependant, on ne considère pas souvent comment ces compromis peuvent affecter les opérations quotidiennes dans un SOC.

Malgré des résultats prometteurs, les graphes de provenance sont peu adoptés par l'industrie (Dong *et coll.*, 2023). Cette situation s'explique en partie par le manque de connaissances sur la façon d'intégrer ces systèmes dans les processus existants dans un SOC. D'après une étude récente, les spécialistes de la sécurité ont du mal à écrire des règles de détection pour les systèmes graphiques, comparativement aux formats de données traditionnels (Jansen *et coll.*, 2024). La sécurité moderne s'appuie majoritairement sur des signatures statiques (Mink *et coll.*, 2023). Selon une étude récente de SANS sur les pratiques du SOC (Crowley *et coll.*, 2023), les techniques avancées d'intelligence artificielle et d'apprentissage automatique sont peu utilisées. Cela rend les propositions d'articles de recherche moins attrayantes pour les équipes de sécurité. Cette opinion se renforce encore si les choix architecturaux sacrifient la clarté et la facilité d'interprétation des résultats pour améliorer les performances d'exécution (Nadeem *et coll.*, 2023)

Cette partie est consacrée à l'étude de ces limites pratiques en utilisant une méthode innovante d'analyse de graphes de provenance pour concevoir un système de détection d'intrusion qui répond aux exigences d'un SOC. La proposition principale consiste à ne pas entrainer un réseau de neurones graphiques (GNN) directement sur un graphe de provenance, comme la plupart des approches existantes le font. Il est plutôt suggéré d'utiliser ce graphe comme représentation intermédiaire dans un pipeline de détection en deux phases. La discussion qui suit porte sur la manière dont cette approche ouvre de nouvelles possibilités de minimiser le nombre d'opérations graphiques couteuses, tout en assurant la compréhensibilité et la complétude des données. Par conséquent, cela permet de se concentrer sur d'autres exigences opérationnelles du SOC.

5.2 Architecture proposée

Les graphes de provenance posent principalement des défis en termes de quantité de données, notamment en nombre d'actualisations par seconde et à la taille globale. Cela est particulièrement vrai pour la détection des campagnes d'intrusion furtives et longue durée. Deux approches principales peuvent être utilisées pour résoudre ce problème :

1. accélérer l'analyse des graphiques en les compressant ou en utilisant une méthode de plongement rapide ;
2. réaliser l'analyse des graphiques moins fréquemment.

La plupart des systèmes avancés actuels sont basés sur la première stratégie d'optimisation. Par conséquent, ce chapitre va se concentrer sur la deuxième stratégie d'optimisation en utilisant un système de détection d'intrusion à deux phases. La proposition est fondée sur l'observation suivante : les flux de travail modernes du SOC sont organisés autour de deux tâches distinctes, à savoir la génération d'alertes de sécurité (détection) et l'analyse de ces alertes (investigation). Le système de détection est donc divisé en deux étapes distinctes, chaque étape utilisant un modèle d'apprentissage automatique différent. Il est évident que les systèmes existants ont tendance à confondre les objectifs de ces deux étapes séparées et à les optimiser comme une seule métrique de performance. Cela entraine la production d'alertes inappropriées pour le triage ou l'investigation.

L'architecture complète est illustrée à la Figure 5.1. De manière concrète, la Phase 1 vise à sélectionner l'ensemble initial d'évènements suspects par l'exécution d'un modèle de détection simple optimisé pour la vitesse, au détriment de la précision. Par contre, la Phase 2 accorde moins d'importance à la vitesse d'exécution, mais elle peut utiliser l'intégralité du graphe pour améliorer la précision et le rappel.

Une récente étude sur l'adoption des outils EDR basés sur la provenance (Dong *et coll.*, 2023) a révélé que les solutions actuelles ne répondent pas aux exigences de déploiement industriel et que les couts de calcul et de main-d'œuvre sont actuellement trop élevés.

Les conclusions de leur travail sont liées aux sujets abordés dans cette recherche. Les commentaires des professionnels de l'industrie cités dans cette étude aident à l'établir les objectifs suivants :

Implémentation bout à bout : Comme cela a été souligné à plusieurs reprises dans cette étude, l'objectif est de concevoir un IDS pratique et adaptable à un déploiement réel dans un réseau d'entreprise typique. Le pipeline de traitement des données doit inclure toutes les étapes nécessaires à la détection en temps réel des attaques. Cela signifie qu'il est nécessaire de se concentrer sur une mise en œuvre complète bout à bout, où le processus s'exécute directement sur les données brutes, sans aucune étape de prétraitement hors bande.

Explicabilité : Cet objectif est considéré comme étant le plus important, puisqu'il permet à l'analyste de sécurité d'avoir confiance envers les résultats de l'outil, et d'éviter de perdre du temps à valider les alertes. Pour atteindre cet objectif, l'information contenue dans les alertes doit être aussi spécifique que possible. Il est donc préférable d'identifier des nœuds et des arcs malveillants individuels du graphe de provenance, plutôt que de simplement conclure que la structure du graphe est anormale.

Provenance des alertes bout à bout : Le concept de provenance des données s'applique aussi bien aux alertes IDS qu'à l'ensemble du système surveillé. Il est donc crucial de s'assurer que chaque alerte IDS puisse être suivie depuis sa source jusqu'à l'évènement spécifique qui l'a déclenchée. Cette propriété garantit la traçabilité, facilite l'analyse des causes profondes et permet à l'analyste de sécurité de diriger son enquête vers des données non captées par l'IDS.

Détection en temps réel : Selon le Chapitre 2, l'utilisation efficace des graphes de provenance est un problème ouvert. Une solution possible serait de construire un graphe complet et d'effectuer une analyse en ligne hors ligne afin de détecter tous les indicateurs historiques d'activité

malveillante. Bien que cette approche soit efficace pour la chasse aux menaces, le but est de mettre en place une détection en temps réel pour aider les analystes de sécurité à gérer les risques et l'exposition liés aux incidents de sécurité.

Les rapports de menaces indiquent que l'élévation des privilèges et le mouvement latéral peuvent survenir en moins d'une heure après la compromission initiale (Unit42, 2024). Cela signifie que l'IDS dispose seulement de quelques minutes pour générer une alerte pour laisser suffisamment de temps au responsable de la sécurité pour effectuer le triage et l'enquête.

Un faible taux de fausses alertes : Comme c'était déjà examiné en détail dans la Section 2.2, les faux positifs demeurent le plus grand défi opérationnel du tri des alertes IDS en raison de leur faible précision et de la dérive conceptuelle constante. Par conséquent, tout outil de sécurité qui entraine une augmentation de faux positifs pour l'équipe risque de ne pas être adopté dans le SOC, malgré ses capacités de détection potentielles.

5.3 Implémentation

Le pipeline complet a été rédigé en langage de programmation Go, une décision motivée par des considérations semblables à celles exposées dans la Section 3.6.4. Chaque composant est implémenté comme un exécutable séparé qui échange des informations avec des sockets de réseau pour permettre un traitement parallèle d'évènements. Cette section présente en détail l'implantation, ainsi que la justification des choix de conception qui contribuent aux objectifs finaux.

Comme l'introduit le Chapitre 1, il existe deux méthodes principales d'implémenter un IDS : un déploiement sur un hôte, qui consiste à surveiller le comportement de chaque hôte séparément ; et une méthode centralisée, appuyée par un SIEM, qui permet de corréler les données entre les hôtes. Compte tenu de la limite pratique sur le nombre d'agents de surveillance, un déploiement basé sur l'hôte ne peut pas effectuer des tâches de corrélation avancées sans affecter négativement les performances de l'hôte. Pour cette raison, le modèle de déploiement préféré par les entreprises est d'utiliser des agents de surveillance légers qui envoient toutes les données vers un emplacement

de stockage central où les analyses et corrélations couteuses peuvent se produire. Puisque cette recherche porte sur le déploiement pratique d'IDS dans les entreprises, tous les composants d'IDS doivent être conçus pour gérer la télémétrie mélangée de tous les hôtes simultanément. Par conséquent, la mise en œuvre doit offrir un bon rendement et être en mesure de traiter en temps réel toutes les données entrantes de tous les hôtes.

Comme le volume de données dépasse la capacité du RAM, il est impératif d'intégrer une couche de persistance pour stocker les données sur le disque. Dans leur étude sur l'évolution du stockage de données au cours des 20 dernières années, Stonebraker et Pavlo (2024) a conclu que les bases de données relationnelles sont la technologie la plus polyvalente, qui peut s'adapter à presque tous les cas d'utilisation. SQLite a été sélectionné comme moteur de base de données, car il offre un bon rendement pour le volume de données et s'intègre bien avec la plupart des langages de programmation. Cela facilite grandement son utilisation dans les modules Go.

En ce qui concerne l'apprentissage machine, Python est le choix naturel en raison de son écosystème de bibliothèques d'apprentissage machine. Toutes les corrélations d'évènements de haut débit sont effectuées dans des modules Go. Lorsque cela est approprié, les évènements enrichis sont transférés vers des composants Python via l'API REST. Ces derniers les transforment en caractéristiques et les organisent en dataframes.

Finalement, pour suivre en temps réel la santé et les performances du pipeline, tous les composants implémentés collectent les données de télémétrie dans un serveur Prometheus avec l'application Kibana pour calculer les métriques pertinentes, qui seront discutées prochainement.

La section suivante contient les détails sur les composants clés du système et explique comment ils contribuent aux objectifs.

5.3.1 Transmission de données

La première étape du pipeline consiste à lire les données brutes provenant des fichiers JSON dans un ordre chronologique et à les envoyer vers les modules suivants.

Dans un contexte réel, la collecte de données d'hôte serait réalisée par un agent léger exécuté sur chacun des hôtes surveillés. La fonction de ce composant est de simuler cette fonctionnalité en lisant les évènements bruts de l'ensemble de données.

Pour une évaluation réaliste, il faut traiter les évènements selon leur chronologie, car c'est ce qui établit les liens de cause à effet sur le graphe de dépendance. Dans un monde idéal, tous les évènements seraient parfaitement ordonnés. Toutefois, dans la pratique, on peut observer des évènements hors de l'ordre pour deux raisons principales :

- les évènements ne se produisent pas dans le bon ordre sur l'hôte ;
- les évènements ne sont pas traités dans le bon ordre par le pipeline.

La distinction entre ces deux facteurs est cruciale. La séquence des évènements à l'intérieur du pipeline est, dans une certaine mesure, contrôlée par l'implémentation. Cela signifie que les évènements hors de l'ordre peuvent être corrigés à l'aide de mécanismes de synchronisation, tels que les clés, les files d'attente et les primitives de synchronisation. Cela peut entrainer un ralentissement du traitement, mais si l'exactitude des prédictions de l'IDS dépend d'une séquence d'évènements correctement ordonnée, cela peut être un compromis nécessaire.

D'un autre côté, si les évènements sont observés hors de l'ordre sur l'hôte même avant d'être collectés par l'agent, cela est largement au-delà du contrôle de l'IDS et doit être considéré comme une limitation intrinsèque. De nombreuses raisons peuvent expliquer ce phénomène : des états de concurrence dans le noyau du système d'exploitation, des horloges système non synchronisées, des sémantiques d'évènements brisées, etc. Cela signifie que, si on veut construire une IDS pratique pour un déploiement réel, il faut tenir compte de cette propriété des données.

La section consacrée à l'évaluation montrera que la chronologie des évènements revêt une grande importance lorsqu'on travaille avec le jeu OpTC et dans un contexte réel.

La particularité du jeu de données OpTC est que les évènements ne sont pas classés chronologiquement dans les fichiers, mais plutôt regroupés par hôte. Cela signifie que l'approche naïve de lecture

des fichiers dans l'ordre séquentiel ne produira pas une séquence d'évènements correctement chronologique, ce qui est en contradiction avec les objectifs de conception. Une solution possible consiste à prétraiter les données en les indexant entièrement dans une base de données, puis en utilisant ses capacités de tri pour obtenir la ligne du temps. Par exemple, SteinerLog (Bhattarai et Huang, 2022; Aly *et coll.*, 2024) et Kairos (Cheng *et coll.*, 2024) font ainsi. Toutefois, cela s'oppose aux objectifs de conception, car stocker en grande quantité des journaux dans une base de données pour le prétraitement rendra impossible la détection en temps réel.

La solution consiste à lire tous les fichiers sources en parallèle, à stocker en mémoire des entrées hors d'ordre, puis à émettre l'évènement suivant avec le temps le plus petit. Le modèle de concurrence natif du langage de programmation Go, combiné à une structure de données de type pile, rend cette tâche aisée. Au total, il y a 366 fichiers journaux contenant des évènements bruts dans le jeu OpTC, qui doivent tous être traités en parallèle. Pour maximiser la vitesse de traitement, il faut trouver un bon équilibre entre les opérations d'entrée-sortie sur disque et la taille de la mémoire tampon. Mais comme résultat, le composant de transmission des données garantit que les évènements bruts sont émis dans l'ordre chronologique correct, et corrige même certaines erreurs produites par les agents (discutées en détail plus loin).

5.3.2 Corrélation d'évènements

Chaque entrée dans le jeu OpTC représente un évènement individuel sur un hôte cible, comme la création d'un processus ou l'opération de lecture d'un fichier. Pour créer un graphe de provenance, il faut cependant travailler avec des entités : des processus, des sessions d'utilisateurs, etc. La conversion entre ces deux représentations de données est assurée par le composant de corrélation d'évènements.

Sa fonction est relativement simple : surveiller les évènements de témoignage entrants et extraire les paires d'évènements correspondants qui définissent une durée de vie d'une entité, telles que :

- les évènements de création et arrêt d'un processus pour construire un arbre des processus ;

- des évènements de connexion et de déconnexion qui délimitent une session interactive sur un hôte ;
- des paquets TCP SYN et FIN pour suivre les connexions réseau actives.

Ce composant permet de générer des caractéristiques temporelless inaccessibles dans les données initiales. Ces caractéristiques peuvent ensuite être utilisées par les modèles d'apprentissage automatique. De plus, il est également possible d'évaluer plusieurs aspects clés du jeu de données qui influenceront directement la qualité de la détection d'intrusion : taux de perte d'évènements, erreurs d'ordre des évènements, collisions d'entités et taille de la population d'entités. Cette recherche a souligné que la détection d'intrusion dans les réseaux d'entreprise ne se limite pas aux seuls indicateurs de précision et de rappel. La mise en œuvre met en évidence l'importance de collecter diverses métriques sur la qualité des données entrantes dans l'IDS, montrant qu'elles ne peuvent être ignorées.

Cette étape permet de mettre en évidence les liens existants entre différents éléments identifiés dans les données initiales. Cette information peut servir non seulement à l'entrainement de modèles d'apprentissage automatique, mais elle fournit également un contexte important pour l'état de l'environnement pour un analyste de sécurité lors de l'enquête. Par exemple, cela permet de répondre à des questions comme : « Un processus lance-t-il des sous-processus ? » ; « Combien de temps dure une session d'utilisateur » ; « Les deux hôtes échangent-ils du trafic réseau pendant une période donnée » ; etc.

Un serveur API REST intégré au module pendant l'exécution du pipeline permet d'obtenir des informations connues sur les entités. Cette fonctionnalité est également utile pour un analyste en sécurité lorsqu'il trie les alertes. Pour ce faire, le module de corrélation d'évènements stocke tous les détails sur chaque entité rencontrée, quel que soit leur nombre. Bien que le nombre d'utilisateurs et d'hôtes sur le réseau reste généralement constant, chaque processus et chaque interaction réseau représentent une nouvelle entité. Cela entraine une croissance rapide du nombre d'entités à gérer, qui dépasse les capacités de la mémoire disponible. Il faut donc absolument une solution avec une couche de stockage persistant pour y remédier. Ceci

est possible grâce à une méthode de partitionnement de la base de données qui permet une croissance illimitée des entités, et qui offre des requêtes efficaces pour les données d'entité dans un cadre de temps spécifique.

Cette fonctionnalité constitue une avancée significative par rapport aux approches actuelles de la recherche. Ces dernières se concentrent uniquement sur la détection des anomalies potentielles dans les données brutes ou agrégées et finissent par décharger les tâches d'analyse et de vérification vers l'utilisateur final. Cette recherche a démontré qu'en concevant correctement un système IDS, il est possible non seulement de détecter les anomalies, mais aussi d'offrir aux analystes de sécurité des outils et un contexte supplémentaires pour découvrir et neutraliser les menaces, ce qui contribue à une mise en œuvre réussie de cet IDS.

5.3.3 Mise à jour du graphe

En outre, la méthode se distingue des techniques existantes par l'importance accordée au graphe de provenance. Dans de nombreuses études antérieures, le graphe de provenance est utilisé comme structure de données de base pour entrainer un modèle de réseau graphique (GNN) et détecter directement les anomalies dans le graphe. Des travaux précédents ont démontré que les modèles graphiques ont la capacité de capturer des liens complexes entre divers éléments du réseau, tels que les utilisateurs, les hôtes et les processus, et d'identifier des structures atypiques au niveau du graphe ou d'un nœud individuel. Cependant, le principal défi pour ces méthodes est l'explication. Par exemple, UNICORN (Han *et coll.*, 2020) est capable de détecter avec précision les attaques complexes au niveau du graphe. Toutefois, déterminer la cause profonde dans un graphe comportant des milliers de nœuds est une tâche insurmontable. Des recherches ultérieures ont été menées pour améliorer l'explication des résultats en utilisant diverses méthodes innovantes, telles que Kairos (Cheng *et coll.*, 2024), qui suit l'évolution du graphe au fil du temps et ne rapporte que les nœuds qui ont changé d'état pendant les périodes possibles d'attaque. Bien qu'elle constitue une avancée par rapport à la détection à l'échelle du graphe complet, cette méthode ne répond pas entièrement aux objectifs d'explicabilité d'alerte et de traçabilité de bout en bout.

Cette approche élimine entièrement la question de comment lier la sortie d'un GNN aux évènements bruts. L'utilisation des graphes de provenance comme représentation intermédiaire permet d'effectuer l'analyse des données brutes directement. Le graphe de provenance remplit deux fonctions : (a) il préserve le contexte utilisé pour enrichir les évènements ; et (b) il appuie l'analyse de la phase 2 pour réduire les faux positifs.

Les types d'interaction que le graphe peut représenter dépendent entièrement des évènements sous-jacents capturés par les données brutes. Les systèmes d'exploitation modernes supportent des centaines de syscalls (Linux project, 2024) et d'objets internes, qui peuvent tous être représentés comme des entités distinctes dans un graphe de provenance. En réalité, la forme et le type d'évènements dépendent de l'outil utilisé pour collecter les données de télémétrie, qui est généralement une solution EDR commerciale ou un collecteur de niveau système dédié, tel que CamFlow (Pasquier *et coll.*, 2017). La méthode la plus courante dans la littérature consiste à se concentrer sur les identités des systèmes (utilisateurs), des processus et de leurs interactions avec le système de fichiers et les connexions réseau.

5.3.4 Enrichissement d'évènements

Les requêtes d'enrichissement d'évènements examinent l'état actuel accumulé par les deux composants précédents et combinent ces données aux évènements bruts. Cela permet d'obtenir des informations structurées sur le voisinage graphique local des nœuds touchés et des informations temporelles sur la durée de vie des entités à partir d'évènements connexes. Les champs d'origine des évènements bruts sont préservés pour atteindre les objectifs d'explicabilité.

Il est important de noter que, tout comme pour le reste du pipeline, l'enrichissement d'évènements ne se produit pas après le traitement des données agrégées, mais en temps réel. Cette opération est déclenchée par l'état actuel lorsqu'un nouvel évènement entre dans le pipeline. Cela signifie que deux évènements liés à la même entité (par exemple, deux connexions réseau initiées par le même processus) peuvent être enrichis avec des informations complètement distinctes.

À cette étape cruciale, chaque évènement reçoit deux identifiants uniques : l'un pour l'évènement brut et l'autre pour le nœud ou l'arc du graphe. Cette association permet de relier deux représentations de données, ce qui est crucial pour atteindre l'objectif ultime de traçabilité des alertes. Toutes les étapes subséquentes du pipeline ont été pensées pour préserver l'intégrité de ces identifiants.

5.3.5 Détection d'anomalies de Phase 1

Les évènements enrichis sont maintenant prêts à être analysés par le modèle de détection d'anomalies.

Selon les objectifs de conception, cette étape privilégie la vitesse de traitement des évènements et la capacité de traitement. Un réseau neuronal superficiel entrainé sur le comportement habituel du système est adéquat pour cette tâche. Cependant, dans la pratique, tout réseau neuronal adapté aux séries temporelles peut être utilisé à ce stade, ce qui ouvre de nombreuses perspectives de recherche futures, telles que l'entrainement simultané de plusieurs modèles en même temps.

Ce composant représente une distinction clé de la méthode par rapport aux approches actuelles. En effet, les conceptions actuelles (voir la Section 2.1) tendent à optimiser directement la détection de performance sur les alertes à ce stade. Ou, si elles utilisent une chaine de traitement en plusieurs étapes, elles envoient les données à l'étape suivante et éliminent les données intermédiaires.

Dans cette approche, le but est d'exploiter au maximum le graphe de provenance afin de détecter toute attaque potentielle, et de minimiser le nombre d'opérations couteuses pour maximiser le volume d'informations conservées. La Phase 1 sert uniquement à identifier les évènements les plus suspects pour une analyse approfondie en Phase 2. Le modèle d'analyse d'anomalies de la Phase 1 ne génère pas directement d'alertes IDS. Au lieu de cela, la note d'anomalie est ajoutée comme champ supplémentaire aux évènements bruts originaux.

Simultanément, les prédictions ne sont pas rejetées, mais elles sont envoyées directement dans la file d'attente des alertes pour triage. Cette approche peut paraitre paradoxale, compte tenu de la volonté de maintenir un faible taux de faux positifs, puisque l'analyse des résultats du modèle non optimisé peut être très bruyante. Cependant, les alertes générées à ce stade ne constituent pas le résultat final et, par conséquent, elles ne sont pas destinées à une analyse manuelle par l'équipe de triage. Mais cela produit un signal important qui peut être utilisé par l'équipe de l'ingénierie de détections. C'est un rôle relativement nouveau (Chuvakin *et coll.*, 2024) qui existe dans une SOC mature pour maintenir une couverture globale de détections et surveiller la santé de l'IDS. La télémétrie actionnable aidera à ajuster la performance de l'IDS en fonction des besoins opérationnels du SOC, par exemple, en prenant en compte l'évolution progressive et la dérive conceptuelle dans l'environnement.

5.3.6 Détection d'anomalies sur le graphe de phase 2

Lorsqu'un nouvel évènement candidat arrive de la Phase 1, il est possible d'obtenir l'identifiant unique du nœud (ou, de manière équivalente, de l'arc) qui a été ajouté à l'évènement à l'étape précédente. Ceci permet de faire le suivant :

1. Associer la note d'anomalie au nœud, qui devient une information permanente sur le graphe de provenance ;
2. Effectuer une détection d'anomalie par les méthodes graphiques, prenant le nœud comme point de départ.

En réalité, la Phase 2 est la partie la moins novatrice de cette recherche, car il s'agit d'un lien direct entre cette recherche et l'état de l'art des systèmes de suivi de provenance. À cette étape, il est possible de réaliser le même type d'analyse que plusieurs œuvres existantes dans ce domaine font (Anjum *et coll.*, 2022; King et Huang, 2023; Goyal *et coll.*, 2024).

Étant donné les couts computationnels importants de l'analyse graphique, l'objectif de plusieurs techniques innovantes consiste à trouver un équilibre entre la granularité de la détection, la taille

du graphe et le nombre de détections. Tous ces éléments affectent négativement l'interprétabilité des alertes IDS. Plutôt que d'augmenter le nombre d'opérations couteuses, la méthode proposée consiste à identifier les nœuds du graphe les plus intéressants en fonction des résultats de la Phase 1 et à les analyser en profondeur. Cette approche permet d'éviter de compromettre la complétude et l'explicabilité des résultats.

Cette étape accepte des données brutes sous forme d'évènements, qui sont ensuite liés aux nœuds pertinents du graphe en fonction d'identifiant unique, fourni par l'étape précédente du pipeline. De cette manière, il est possible de maintenir le lien entre la cause profonde et l'activité potentiellement malveillante, même si les représentations de données sont différentes.

En raison de contraintes d'espace et de temps, cette recherche n'inclut pas l'implémentation de la Phase 2 dans. Dans des recherches futures, peuvent examiner comment les projets existants, comme Kairos (Cheng *et coll.*, 2024) et Flash (Ur Rehman *et coll.*, 2024), peuvent s'intégrer dans le pipeline en tant que composants de la Phase 2. En fonction de la manière spécifique dont le cadre IDS est mis en œuvre, le résultat du pipeline peut être soit des nœuds du graphe, soit des évènements bruts, enrichis avec des champs additionnels dérivés du graphe de provenance. Si le score d'anomalie dépasse le seuil prédéterminé, les évènements sont émis comme des alertes dans la file d'attente des enquêtes, où elles sont examinées par un analyste de sécurité.

5.3.7 Le pipeline complet

L'architecture complète est illustrée à la Figure 5.1. Les blocs représentent les composants logiques discutés dans ce chapitre. L'architecture IDS à deux étapes est inspirée des meilleures pratiques des centres de sécurité des grandes entreprises, avec un graphe de provenance comme représentation intermédiaire des données.

Il est important de souligner que le design proposé dans ce travail est un cadre IDS, ce qui signifie que les composants individuels (représentation graphique, modèles d'apprentissage, etc.) peuvent être remplacés par des implémentations alternatives, ou même combinés. Le spécifique de l'IDS utilisée dans ce travail est entièrement déterminé par le format et la composition des

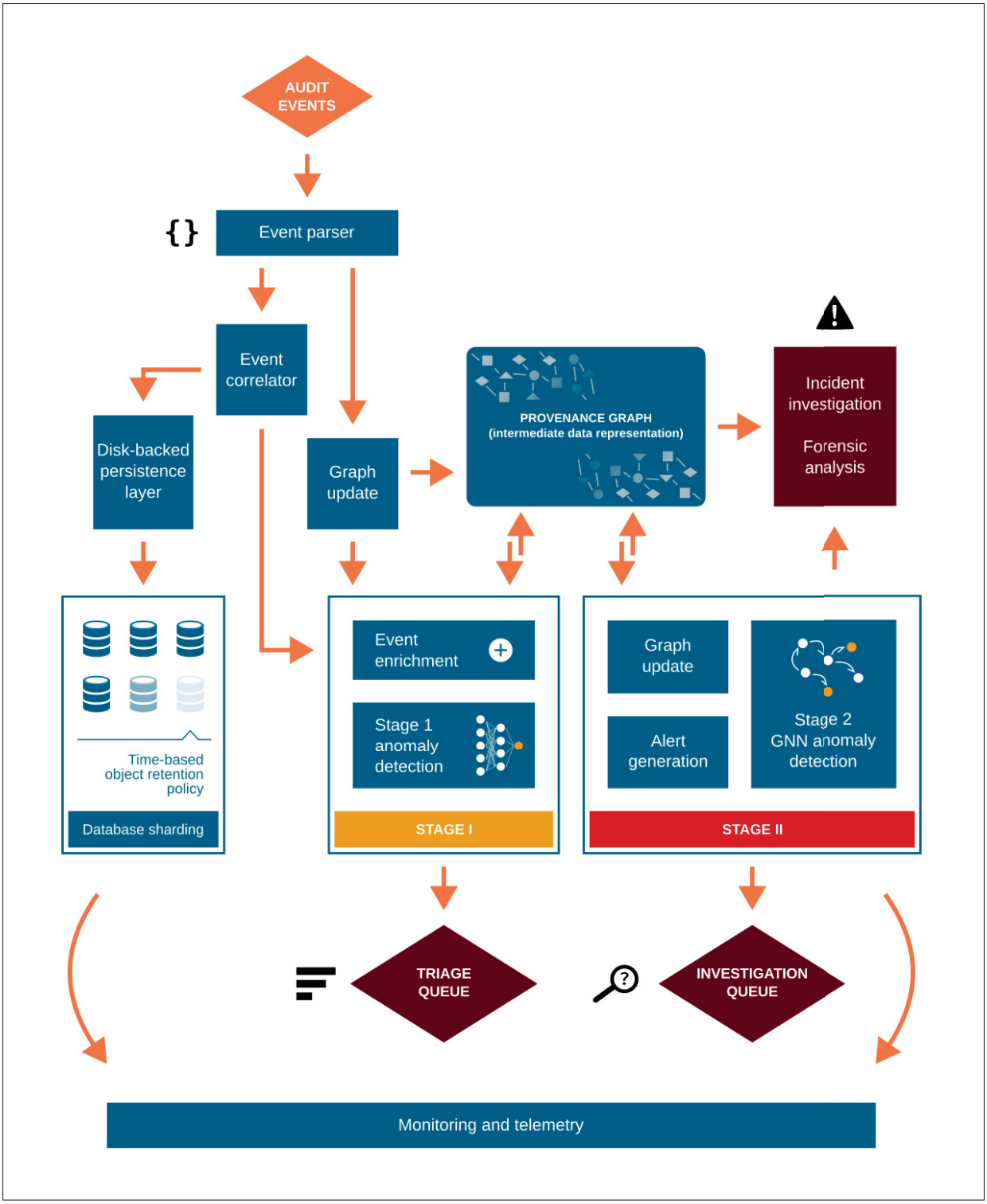

Figure 5.1 Une représentation schématique du pipeline complet

données brutes, et il n'est donc pas possible de la transférer directement dans un environnement réel. Cependant, une fois les ajustements nécessaires effectués dans un nouveau contexte, le cadre devrait continuer à fonctionner de la même manière, permettant ainsi aux analystes en sécurité d'atteindre les objectifs fixés dans cette étude.

La contribution principale est un système de détection d'intrusion *en deux phases*, utilisant un graphe de provenance comme *représentation intermédiaire* des données.

Décomposer le défi complexe de la détection d'intrusions en tâches plus petites et mieux définies permet d'intégrer plus facilement le nouveau détecteur d'intrusions avec les processus actuels du centre de sécurité de l'entreprise. Il est possible d'économiser des ressources en utilisant une représentation intermédiaire unique pour créer plusieurs modèles d'analyse d'anomalies, chacune spécialisée dans la détection d'une forme particulière d'attaque. Cela entraine une baisse importante des couts d'exploitation pour l'entreprise.

La prochaine section analyse les capacités de l'IDS proposé en utilisant l'ensemble complet de données OpTC.

5.4 Évaluation expérimentale

Cette dernière section se concentre sur l'évaluation du cadre IDS innovant à l'aide de l'ensemble de données OpTC, le plus grand et le plus réaliste pour la recherche en détection d'intrusion. Au cours de ce travail, l'accent a été mis sur la création d'un système IDS fonctionnel prenant en compte la complexité des environnements d'entreprise modernes. Ce qui reste à faire c'est de développer les composants clés introduits dans la section précédente et de réaliser une série de mesures afin de comprendre si l'architecture s'adapte bien au traitement des données volumineuses pour identifier les activités malveillantes.

5.4.1 Introduction

Le Chapter 2 aborde le large écart qui existe entre la recherche académique et la pratique industrielle et propose que les métriques de performance les plus courantes, telles que la précision, le rappel et la mesure F1, ne reflètent pas adéquatement les résultats de sécurité mesurables. L'oubli de la fréquence de base (Axelsson, 1999; Arp *et coll.*, 2022) est la cause principale du fardeau des alertes pour les analystes de sécurité. La taille de l'environnement moderne et la dérive conceptuelle constante entrainent des couts additionnels importants pour les entreprises qui souhaitent mettre en place plusieurs des systèmes IDS proposées dans la littérature.

Pour ces raisons, il est nécessaire de collecter un ensemble diversifié de métriques de tous les composants du pipeline IDS. Cela permettra de comprendre comment le cadre proposé peut aider les analystes de sécurité à estimer les couts du travail et des matériaux requis pour une implantation réussie dans un environnement d'entreprise.

Bien qu'OpTC présente de nombreux avantages, il s'agit d'un jeu de données de référence développé dans un environnement de laboratoire contrôlé. Par conséquent, il ne peut pas capturer avec précision la complexité d'un environnement d'entreprise moderne. Cependant, comme discuté dans le Chapitre 3, le jeu OpTC possède certaines propriétés intéressantes de réseaux réels. Par conséquent, si l'expérience met en évidence des résultats prometteurs en matière de détection d'intrusion qui peuvent être appliqués à l'ensemble de référence, il est justifié de tirer la conclusion que les mêmes résultats s'appliquent à tous les réseaux réels assez sophistiqués.

De plus, comme le démontre le Chapitre 4, il est relativement facile de détecter les activités malveillantes dans le jeu OpTC en utilisant des techniques simples. Par ailleurs, l'optimisation des métriques de performance typiques (c'est-à-dire la précision, le rappel et la mesure F1) au niveau de l'évènement brut individuel est en contradiction avec les objectifs de cette recherche. Par conséquent, il suffit de vérifier que l'IDS détecte les activités malveillantes de manière attendue et de se concentrer sur d'autres mesures qui contribuent à une mise en œuvre réussie dans l'entreprise.

Les expériences menées sont les suivants.

5.4.2 Aperçu de la performance complet

L'un des objectifs consiste à détecter des menaces en temps réel pour aider les analystes de SOC à minimiser les risques et l'exposition aux incidents de sécurité. Pour y parvenir, chaque composant de la chaine de traitement doit traiter les évènements au moins aussi rapidement que leur création dans OpTC. La méthodologie choisie ne permet pas de prétraiter ou de sélectionner des ensembles d'évènements en fonction de l'hôte, car cela serait impossible lors de l'analyse de flux de télémétrie en temps réel.

Une hypothèse commune dans la littérature est que les systèmes de suivi de provenance peuvent être directement déployés en tant qu'agents sur un hôte surveillé, ce qui leur permet de traiter un volume limité de télémétrie provenant d'un seul hôte. Cependant, installer de nouveaux agents sur des hôtes peut s'avérer difficile dans les environnements d'entreprise. Les entreprises ont déjà tendance à posséder une gamme de logiciels spécialisés pour la gestion des TI et la conformité. L'ajout d'un agent supplémentaire consomme des ressources système, ce qui ralentit la performance et entraine une mauvaise expérience utilisateur. Pour cette raison, les sociétés ont tendance à privilégier le modèle de déploiement sans agent, qui s'appuie sur la fonctionnalité standard de l'OS pour recueillir les données de l'hôte et les transmettre à un répertoire centralisé, tel qu'un système SIEM. Cette approche permet un traitement étendu des données sans affecter négativement l'expérience de l'utilisateur.

Pour que le résultat de cette recherche s'intègre bien avec les procédures actuelles de l'entreprise, l'intégration complète du pipeline IDS doit être capable de gérer la télémétrie OpTC de tous les hôtes avec un composant centralisé, ce qui exige des taux de traitement très élevés (EPS).

De plus, en raison de la taille des données (9,7 TB sous forme décompressée), les expériences exigent un disque dur d'une capacité adéquate pour stocker le jeu. Cette technologie fonctionne en lisant des données de manière séquentielle. Cependant, les journaux d'hôte étant découpés en plusieurs fichiers, il faut les lire simultanément. Pour surmonter cette limitation technique, le

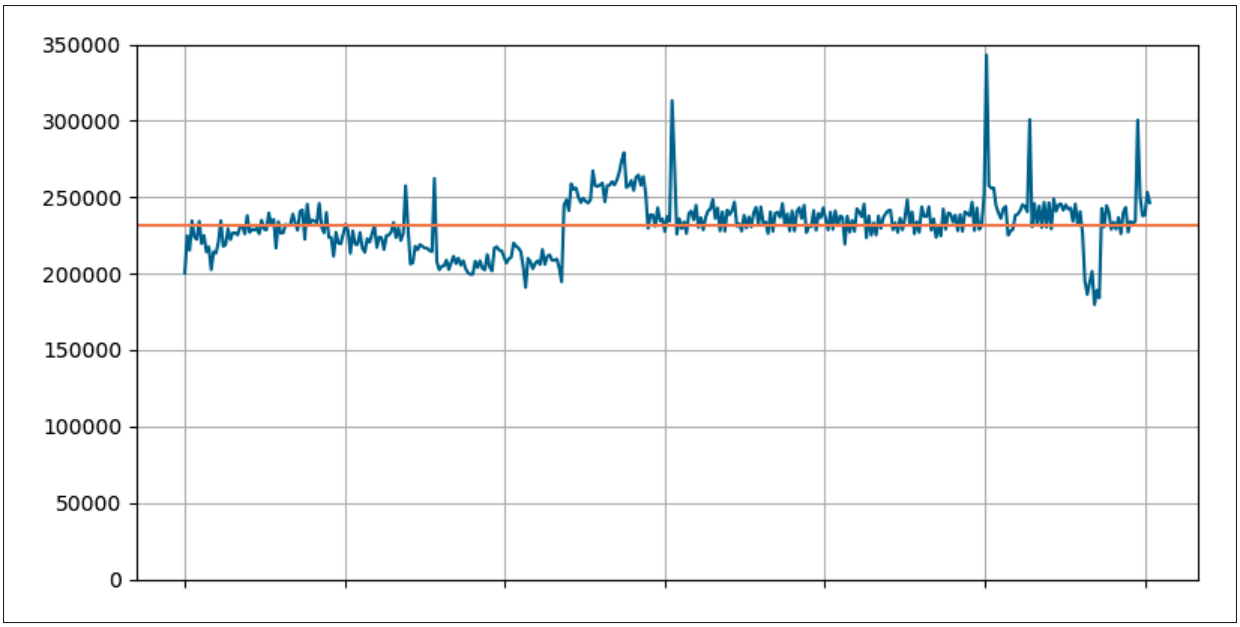

Figure 5.2 Le taux de traitement des évènements par la diffusion réordonnée, qui lit les évènements bruts de tous les hôtes et les injecte dans le pipeline IDS dans l'ordre chronologique

pipeline inclut un composant de diffusion réordonnée en tant que première étape. Son objectif est d'accélérer la lecture des disques.

La Figure 5.2 montre le taux de traitement des évènements (EPS) résultant, injecté dans le pipeline. Le pipeline peut atteindre une vitesse d'EPS moyenne de 231,82K, soit environ 12 fois plus rapide que la vitesse moyenne de génération d'évènements de l'ensemble OpTC. Cette mise en œuvre permet d'atteindre un traitement en temps plus que réel [1]et d'analyser en profondeur les dix jours d'activité en seulement 20 heures. Par conséquent, le système peut facilement être adaptée à de très grands réseaux, si le traitement des évènements est fait à la vitesse normale.

[1] Pour accélérer la réalisation de toutes les expériences, elles sont exécutées au maximum de la capacité du matériel. Malheureusement, cela a produit une chronologie qui n'est pas comparable avec la chronologie originale de l'ensemble OpTC. Par conséquent, les indicateurs collectés ne peuvent pas être corrélés avec les évènements bruts. Pour cette raison, toutes les figures de cette section sont présentées sans aucune étiquette sur l'axe des x, mais toutes les données se trouvent sur la même échelle relative.

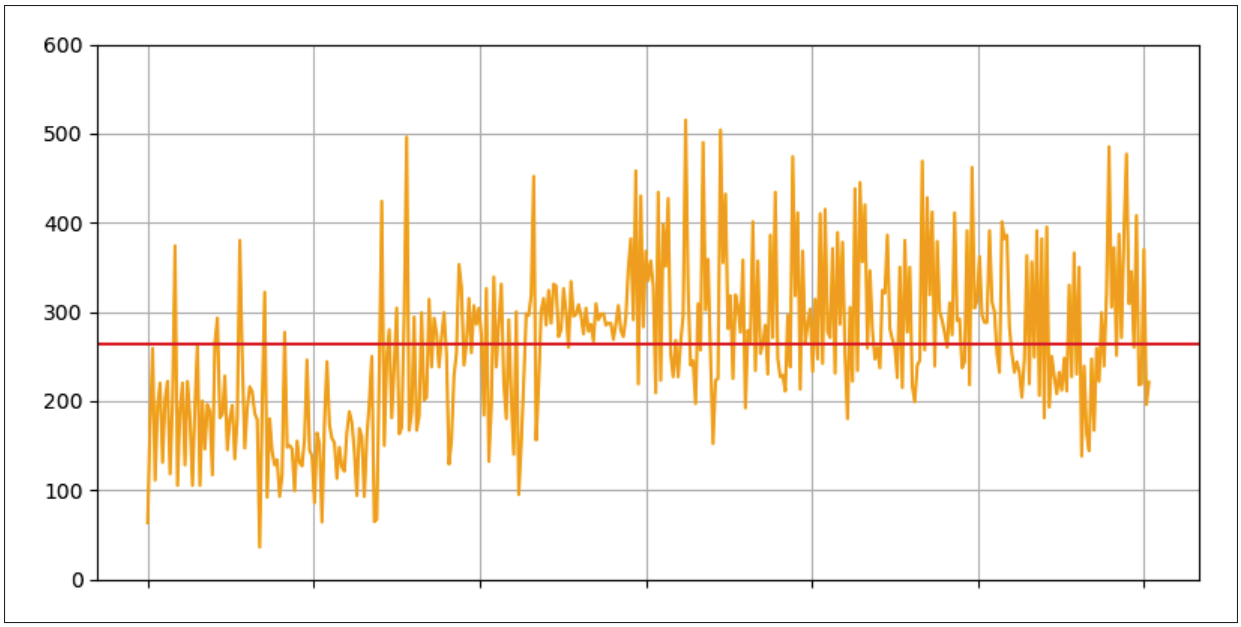

Figure 5.3 Le taux de corrélation des évènements de processus

La diffusion réordonnée initiale a les exigences EPS les plus strictes. Tout au long de la chaine de traitement, les évènements sont progressivement affinés, ce qui entraine une baisse significative du nombre d'évènements traités. Par exemple, lors de la reconstruction de l'arbre de processus, le pipeline doit gérer entre 200 et 400 évènements par seconde, comme le montre la Figure 5.3. Cette grande différence explique pourquoi la majorité des études se concentrent exclusivement sur les évènements de création de processus pendant l'évaluation. Comme discuté au Chapitre 4.4, les évènements de création de processus sont parmi les évènements les plus importants pour la sécurité, mais ils ne représentent qu'une fraction mineure de l'ensemble du jeu OpTC. Cela met en évidence que les évènements ont des valeurs différentes pour la détection d'intrusion. Pour réduire les couts opérationnels, il est crucial de choisir de bons évènements lors de la conception de l'IDS.

5.4.3 Mise dans l'ordre chronologique des évènements

Dans un graphe de provenance, les relations causales entre les objets sont définies par l'ordre des évènements. Une connexion réseau qui arrive avant ou après la création d'un nouveau

processus peut entrainer des conséquences en matière de sécurité différentes. Une hypothèse commune (et souvent implicite) parmi les développeurs de systèmes IDS est que l'ordre correct des évènements peut être reconstitué à partir des données pendant l'entrainement et l'inférence du modèle.

Lors de l'utilisation du jeu de données OpTC, il y a deux facteurs importants à considérer. Chaque évènement contient un champ « timestamp » qui indique le temps de recueillement par le système. Les auteurs n'ont pas divulgué les détails de l'environnement de simulation et du code source de l'agent de collecte de télémétrie. Sans cette information, il n'est pas possible de confirmer si les horloges des systèmes sont synchronisées. Il est donc nécessaire de présumer que le temps attribué aux évènements est correct.

Deuxièmement, le jeu de données représente des données historiques et, puisque la chronologie complète du laboratoire est déjà enregistrée dans des fichiers, il est possible de « voir l'avenir » en examinant les fichiers sources et en comparant les valeurs du champ « timestamp ». Il est important de considérer cela comme une forme de surexploitation des données (data snooping) (Arp *et coll.*, 2022), car prédire des évènements futurs n'est pas possible dans le traitement en temps réel d'évènements dans un déploiement de production.

Les expériences sont conçues pour reproduire le même traitement que lors de la collecte initiale du jeu de données OpTC. Le composant de la diffusion réordonnée garantit que les évènements sont lus dans l'ordre dans lequel ils sont stockés dans les fichiers sources, et traités exactement une seule fois. Cela signifie que les informations concernant l'état passé ou futur de l'environnement sont inaccessibles dans le reste du pipeline.

Dans le cas idéal, le temps devrait seulement augmenter avec chaque nouvel évènement. Donc, lorsque le temps est plus court que celui de l'évènement précédent, on sait qu'il a été livré dans le mauvais ordre. La Figure 5.4 présente la série chronologique des évènements qui ont été livrés hors ordre en lisant les fichiers sources de manière séquentielle. Les lignes superposées représentent la valeur médiane et la valeur maximale par minute pour l'ensemble des données.

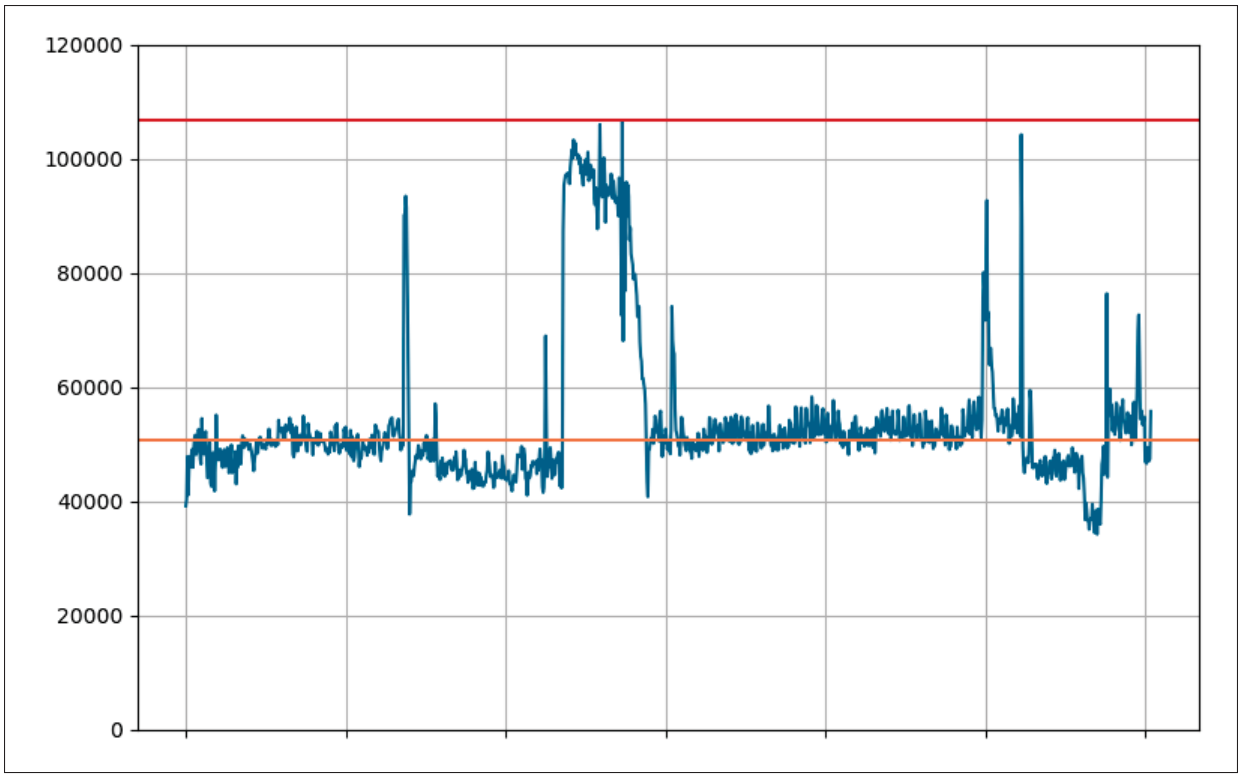

Figure 5.4 Évènements hors ordre enregistrés dans les fichiers sources du jeu de données OpTC

On peut observer que, dans n'importe quel moment, environ 22% de tous les évènements lus sont désordonnés, avec des pics périodiques pouvant atteindre 40%.

C'est un nombre important qui reflète la complexité inhérente à la surveillance au niveau de kernel d'un ordinateur multicœur moderne. Malheureusement, cette limitation n'est pas du tout abordée dans la section d'évaluation de la recherche sur les IDS en provenance de systèmes complets. Les travaux actuels abordent implicitement ce défi en choisissant un sous-ensemble d'évènements intéressants et en les prétraitant avec un outil de données commun, comme Pandas, ou en les indexant dans une base de données pour en profiter de ses fonctionnalités de tri (Bhattarai et Huang, 2022; Aly *et coll.*, 2024). Malheureusement, cela implique que ces conceptions peuvent contenir jusqu'à 40% d'informations supplémentaires sur l'environnement par rapport à ce qui serait possible dans une évaluation en temps réel.

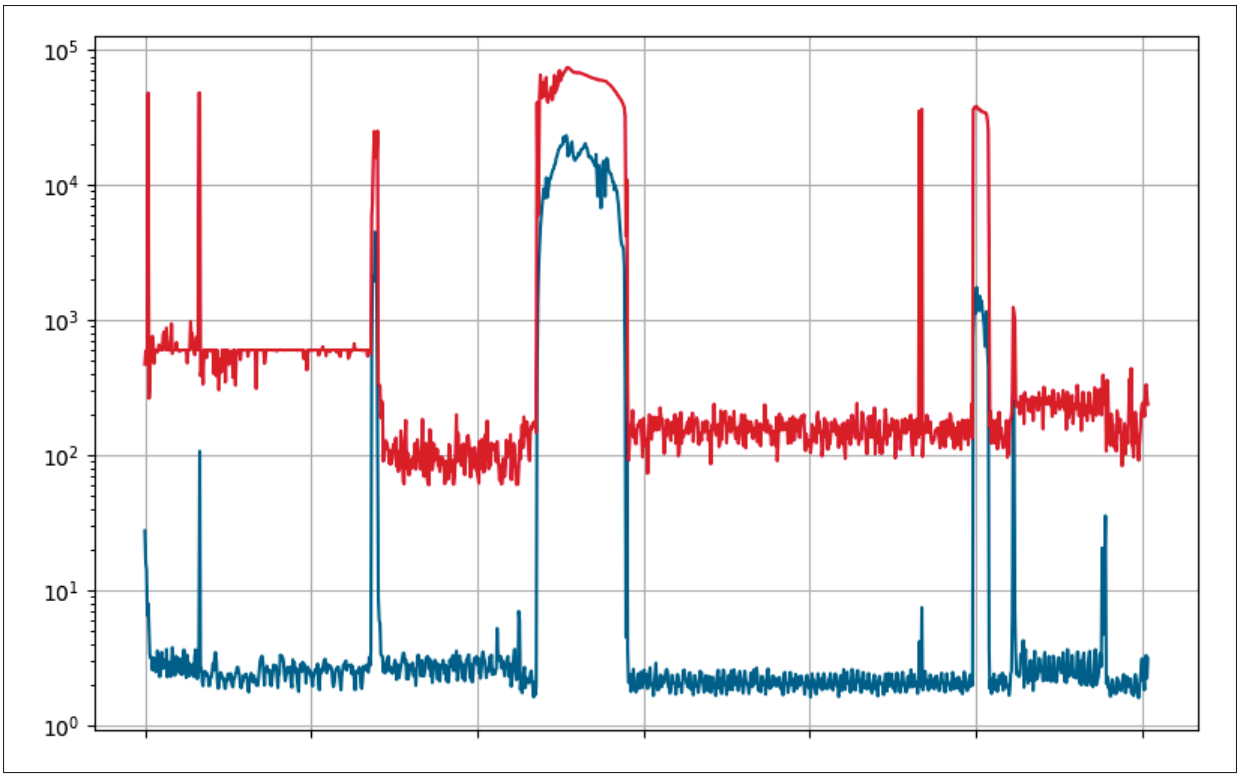

Figure 5.5 Échelle logarithmique moyenne (bleue) et maximale (rouge) du délai des évènements hors ordre en secondes, comparés aux évènements précédents

Un moyen de résoudre ce défi est d'ajouter une couche de tampon et d'émettre les évènements vers la ligne de traitement après un court délai. En supposant que les évènements hors séquence soient livrés quelques millisecondes ou même quelques secondes plus tard, ils peuvent être correctement insérés dans le tampon de délai à la tête des évènements plus tardifs. Le plus grand défi posé par cette méthode est de trouver un équilibre optimal entre l'augmentation des besoins en ressources et le temps d'attente dans la file. La Figure 5.5 présente les délais moyens et maximaux des évènements hors ordre par minute. La plupart des évènements sont délivrés dans moins de 200 millisecondes, ce qui permet de fixer l'ordre de la plupart des évènements en échangeant les lignes adjacentes dans un tampon de délai. Cependant, presque chaque échantillon contient au moins un évènement qui est retardé de plus de cinq minutes. De plus, il y a plusieurs pics très visibles où les évènements sont délivrés plus de cinq heures en moyenne, ce qui peut être un indicateur de problèmes techniques avec l'agent de collecte de données de télémétrie.

Algorithme 5.1 Tampon de délai pour les évènements hors ordre

```
   Input : Collection 𝕆 des fichiers sources
   Output : Évènement en ordre partiellement rétabli
 1 Let H be the heap of file pointers.
 2 Let C(f) be the in-memory cache of lines read from file f.
 3 Let b be the size of the sort buffer for each file.
 4 for f in 𝕆 do
 5 |   Open file f in line mode and add to heap H.
 6 end for
 7 while H contains unclosed files do
 8 |   Pop the element of H that has the smallest log timestamp :
 9 |   f ← pop({h | ∃h ∈ H, min(h)})
10 |   if |C(f)| < b then
11 |   |   C(f) += read a block from f
12 |   |   Sort C(f) by event timestamp.
13 |   end if
14 |   Emit the next log event with the smallest timestamp :
15 |   ◀ {e | ∃e ∈ C(f), min(e.timestamp)}
16 |   if |C(f)| > 0 then
17 |   |   H += push(f)
18 |   else
19 |   |   close(f)
20 |   end if
21 end while
```

La Section 2.2 suggère que les menaces modernes sont capables de pirater le réseau entier en quelques minutes. Par conséquent, l'ajout d'un tampon afin de régler l'ordre des évènements pourrait mener à une occasion manquée de réagir à une attaque en cours à temps. Toutefois, le traitement des évènements dans l'ordre chronologique est crucial pour créer un graphe de dépendances correct. Comme la plupart des évènements observés se déroulent en une fraction de seconde, la structure de données de pile est utilisée pour créer un tampon de délai léger qui peut stocker jusqu'à 2000 évènements par fichier source pour les trier et trouver le prochain évènement. Même ce petit changement augmente considérablement les exigences de mémoire du composant, ce qui souligne la nécessité de personnaliser l'IDS selon les besoins spécifiques de chaque déploiement.

Résultat : Une partie importante des évènements du jeu de données OpTC sont enregistrés dans un ordre incorrect dans les fichiers sources, ce qui peut entrainer la construction erronée d'un graphe de dépendance et une baisse significative des performances de l'IDS lors du traitement en temps réel des évènements. Les études actuelles ne tiennent pas compte de cette caractéristique des données, ce qui pourrait conduire à une évaluation gonflée des mesures de performance en raison de surexploitation des données.

5.4.4 Mesure de la perte d'évènements

En septembre 2024, Microsoft a annoncé avoir perdu pendant deux semaines les journaux de sécurité de ses services infonuagiques (Whittaker, 2024). Cela a rendu impossible à des milliers d'entreprises mondiales de surveiller leurs systèmes ou de détecter des attaques. C'est un cas extrême du problème de perte d'évènements qui, malheureusement, peut être rencontré très souvent. Même si cela arrive rarement à cette échelle, on peut s'attendre à perdre au moins un petit pourcentage de télémétrie dans tout environnement suffisamment complexe.

Bien que le sujet des données manquantes ait été bien étudié dans d'autres domaines (Newman, 2014), il semble que ce sujet soit négligé dans la recherche sur les IDS. C'est surprenant, car plusieurs méthodes sont spécifiquement basées sur la détection d'évènements anormaux qui sont par définition rares. De plus, les attaquants cherchent souvent à se faire discrets, ce qui rend la détection d'anomalies encore plus difficile. Compte tenu de l'importance théorique de tels évènements rares (She, Liu, Wan, Xiong et Fan, 2019), il est crucial de s'assurer qu'ils ne sont pas perdus en transit, car cela pourrait considérablement affecter les performances de l'IDS.

Certains travaux admettent que certaines parties du jeu OpTC ne comportent aucun télémètre de plusieurs hôtes critiques (tels que le contrôleur de domaine), en raison des limites de l'expérience. Cela rend impossible la détection de plusieurs étapes de la chaine d'attaque. Cependant, il est encore courant de supposer une visibilité parfaite dans les activités de tous les autres hôtes. La revue de littérature de cette recherche n'a pas trouvé d'études qui mesurent l'effet de la

perte d'évènements sur le taux de faux positifs. Ce serait utile aux spécialistes en sécurité pour déterminer si le design proposé de l'IDS est approprié pour leur environnement.

La mesure du taux de perte est elle-même très difficile à effectuer. Par exemple, dans un environnement réel, on peut procéder à une analyse de l'hôte. Après avoir examiné l'état des processus, des fichiers, des bases de données, etc., on peut chercher des évènements spécifiques dans les journaux qui correspondent aux changements observés, et ainsi estimer le pourcentage de journaux perdus. Or, il est impossible de procéder ainsi lorsque l'on manipule des ensembles de référence statiques, car on ne peut pas accéder à l'environnement.

Une autre façon de détecter les évènements manquants est possible grâce aux paires symétriques d'évènements, par exemple : CREATE/TERMINATE de processus, SYN/FIN de connexion réseau, OPEN/CLOSE de fichier, LOGIN/LOGOUT de session d'utilisateur.

Parmi les évènements inclus dans le OpTC par ses auteurs, la paire « CREATE » et « TERMINATE » est la plus appropriée, car elle comprend des champs supplémentaires qui peuvent faciliter la corrélation. Dans un système d'exploitation moderne, chaque processus en cours d'exécution reçoit un identifiant de processus unique PID. Deux processus en cours de fonctionnement ne peuvent pas partager le même identifiant. Cependant, une fois qu'un processus est terminé, son identifiant peut être attribué à un autre processus plus tard. On peut créer une table des processus en cours d'exécution pour chaque hôte. Ensuite, un évènement de processus « CREATE » est observé avec une valeur de PID en double, on peut en déduire qu'un évènement « TERMINATE » non enregistré s'est produit entre deux évènements « CREATE » consécutifs. De manière symétrique, la succession de deux évènements « TERMINATE » implique probablement qu'un des évènements « CREATE » est manquant.

La Figure 5.6 montre l'augmentation des collisions de PID, en comparaison avec le nombre total de processus créés sur tous les hôtes, selon les données collectées par le module de corrélation d'évènements spécialement conçu pour cette analyse.

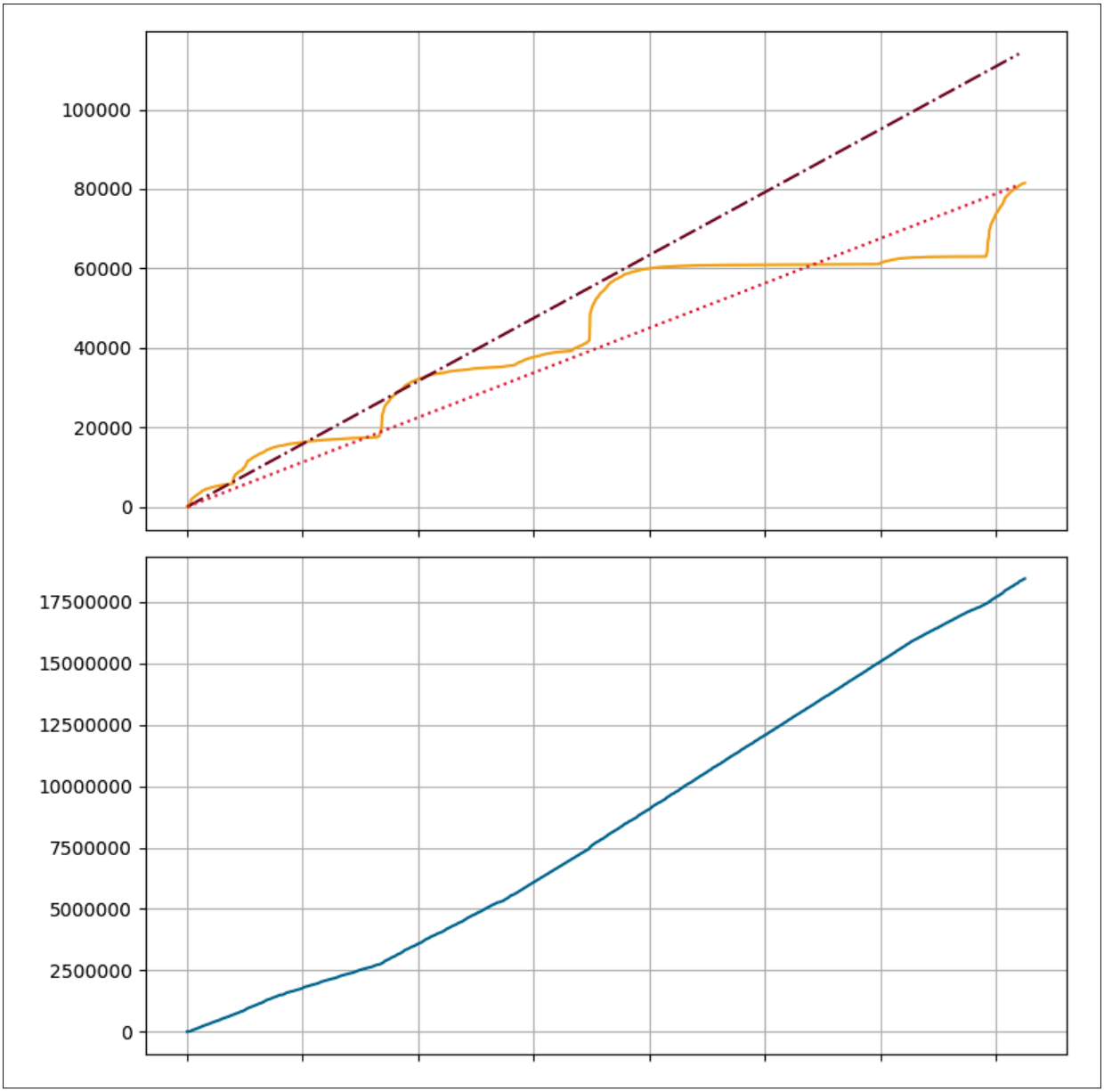

Figure 5.6 Les valeurs PID en double dans les évènements « CREATE » consécutifs sur le même hôte indiquent une perte des évènements correspondants « TERMINATE » pendant la simulation. Le graphique en bas montre une augmentation linéaire du nombre total d'exécutions de processus sur tous les hôtes

Le nombre de valeurs de PID en double augmente avec le temps et atteint environ 4-6% de tous les évènements créés à la fin de la simulation, ce qui semble être un chiffre raisonnable. En supposant que le même taux de perte s'applique à toutes les autres catégories d'évènements dans le jeu OpTC, cela signifie que le jeu OpTC a failli à collecter près d'un milliard d'évènements de l'environnement.

On remarque que le nombre total d'exécutions de processus augmente linéairement pendant la simulation. Cette observation met en évidence la nature scénarisée de la simulation. En effet, chaque hôte est configuré pour répéter sans cesse la même série de tâches, entrainant ainsi un comportement globalement homogène en moyenne. En revanche, les collisions de PID se produisent à des intervalles très courts, ce qui crée des étapes très marquées sur le graphique. Ces étapes correspondent aux modifications administratives apportées par les créateurs du jeu de données lorsqu'ils préparent les hôtes pour les prochaines attaques de l'équipe rouge. En d'autres termes, ces étapes représentent des indicateurs de dérive conceptuelle dans le réseau.

En même temps, le graphique reste presque plat pendant la période d'évaluation, qui correspond à toute l'activité malveillante de l'équipe rouge. Cela suggère qu'on observe les attaques avec une information parfaite, presque sans perte d'évènements. Cette illustration ne reflète pas adéquatement les réseaux réels en constante évolution. En se basant sur le taux de perte d'évènements constaté dans la première partie du graphique, qui montre une dérive conceptuelle, on peut prévoir une hausse linéaire d'environ 6% du nombre de collisions de PID durant la période d'évaluation.

Résultat : L'évaluation des systèmes de détection d'intrusion innovants est effectuée sous l'hypothèse irréaliste de la visibilité parfaite des attaques. Les études actuelles ne mesurent pas l'impact de la perte d'évènements sur les performances des IDS. C'est une source majeure de fausses alertes dans les environnements réels, en raison des limites techniques de surveillance et de la dérive conceptuelle constante qui se produit dans les réseaux d'entreprise.

5.4.5 Persistance des évènements évincés

Le but principal de cette étude est d'examiner la manière pratique de déployer des systèmes de détection d'intrusion dans un environnement professionnel. Il est crucial de prendre en compte la consommation à long terme de ressources par le système. Il est généralement admis que la révision constante des modèles entraine une hausse des couts opérationnels de l'entreprise (Xin, Miao, Parameswaran et Polyzotis, 2021). Cependant, il est moins bien compris que des changements soudains dans la sortie des IDS peuvent aussi entrainer des problèmes majeurs. Ceci inclut des problèmes d'intégration avec d'autres outils SOC et les opérations, qui sont souvent règlementées et nécessitent une réévaluation et un ajustement en cas de changements. Pour réduire au minimum les interruptions, il est recommandé d'implémenter un système IDS « évolutif », capable de s'ajuster graduellement aux changements de l'environnement.

Il existe de nombreuses solutions innovantes pour surmonter l'augmentation exponentielle du graphe de provenance au fil du temps et pour équilibrer la performance à temps d'exécution avec la conservation de l'information importante dans le graphe. Toutefois, cette approche novatrice consistant à utiliser le graphe de provenance comme représentation intermédiaire offre de nouvelles opportunités d'optimisation. Le but principal du graphe est d'ajouter un contexte additionnel aux nouveaux évènements entrants en fonction de l'état actuel du réseau. Cela veut dire qu'afin de maintenir une haute EPS, il faut conserver uniquement les données les plus récentes, celles qui sont pertinentes pour les évènements générés en ce moment. L'analyse approfondie du graphe complet, en revanche, se produit durant la Phase 2 du pipeline, qui traite des ordres de grandeur inférieurs d'évènements et qui peut même se faire hors bande. Par conséquent, les données peuvent être déplacées vers un stockage persistant, comme un disque dur ou une base de données relationnelle, pour une période plus longue.

Le défi consiste à comprendre quels nœuds du graphe peuvent être sécuritairement enlevés de la mémoire et transférés vers un stockage sur disque. Comme c'était brièvement abordé dans la section consacrée à la revue de la littérature, cela dépend entièrement des détails d'implémentation du programme exécuté sur l'hôte surveillé.

En analysant les évènements de création de processus comme objet d'étude, on peut remarqué que les évènements du jeu de données sont tous liés à un processus spécifique sur un hôte. Chaque évènement nouvellement généré indique que le processus correspondant est actuellement exécuté. Cependant, si un évènement « TERMINATE » est reçu pour un processus, on peut en déduire qu'il n'y aura plus d'évènements pour ce processus. Il est ainsi possible de transférer en toute sécurité les données vers un stockage secondaire et permanent, ce qui permettra de réduire la charge de la RAM.

Comme on peut le voir à partir de la Figure 5.7, le nombre de processus augmente d'abord de manière linéaire lorsque l'environnement est configuré. Par la suite, il se stabilise à un niveau constant une fois que l'environnement est saturé. Le transfert vers le stockage persistant se produit à un rythme aussi régulier, avec seulement quelques pics très prononcés, qui correspondent aux périodes de chevauchement de PID, comme discuté dans la section précédente. Chaque fois qu'un tel pic se produit, cela signifie que plusieurs processus, qu'on croyait toujours en cours d'exécution, ont pris fin. Ils peuvent alors être déplacés simultanément depuis la mémoire, ce qui libère une quantité importante de ressources.

Ainsi, l'algorithme interne du système d'exploitation Windows qui contrôle l'affectation des identifiants de processus joue un rôle crucial dans la régulation de la performance du système. Cette constatation renforce l'idée que le problème de la perte d'évènements ne peut être négligé, car il affecte la conception et le comportement de chaque composant de la chaine, et pas seulement le taux de détection final de l'IDS.

De plus, les résultats de l'expérience contredisent l'hypothèse de base, selon laquelle les processus longue durée doivent être divisés en sous-graphes plus petits pour résoudre le problème de l'explosion des dépendances. Cette hypothèse a servi de point de départ à de nombreuses études antérieures sur le découpage des processus à long terme (Ma, Zhang et Xu, 2016; Lee, Zhang et Xu, 2013a; Lee *et coll.*, 2013b). Le comportement d'un hôte « corporatif » typique est très différent, comme le montre la Figure 5.8, qui représente une distribution logarithmique de tous les processus exécutés. La grande majorité des processus ont une durée d'exécution courte,

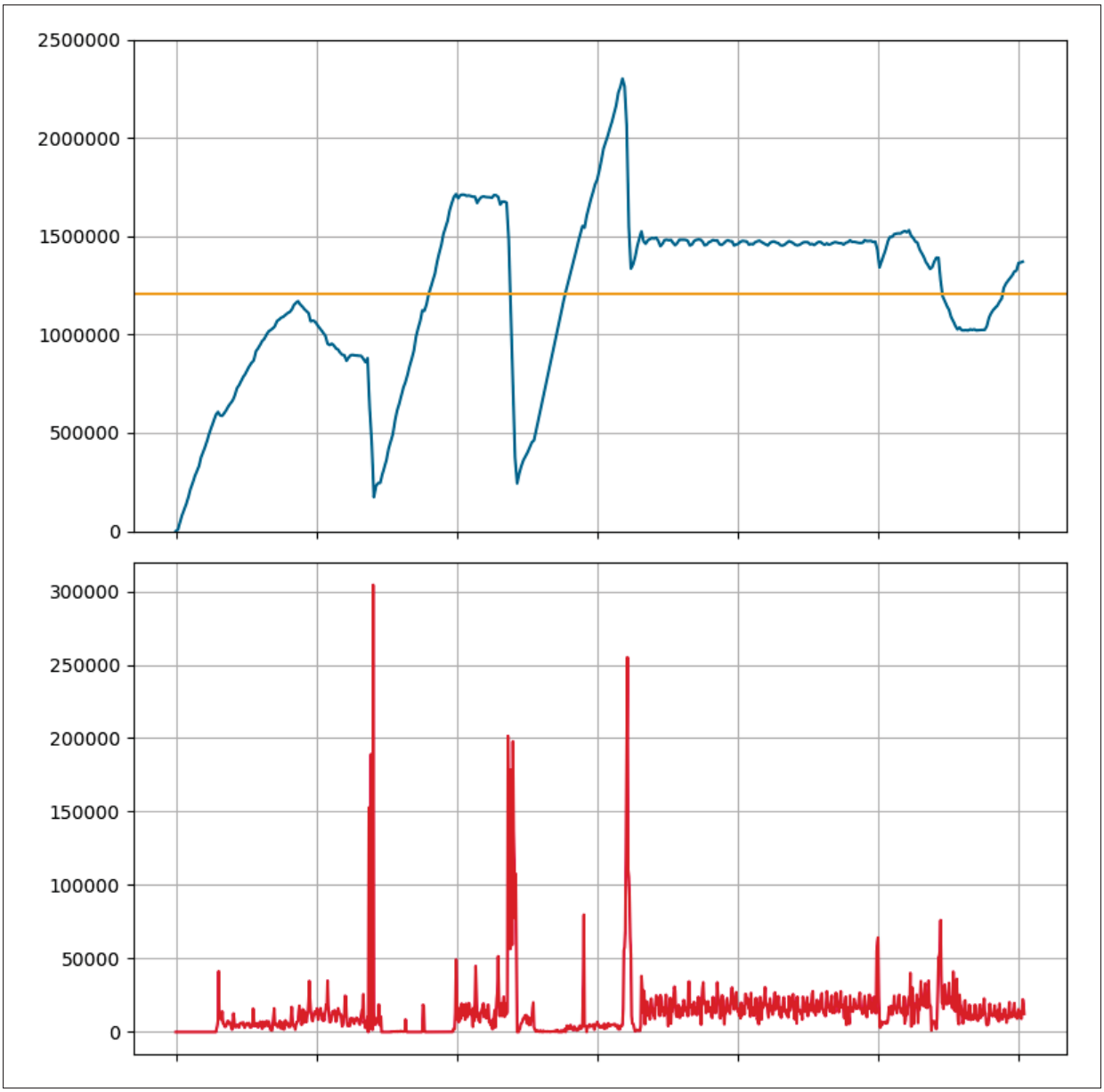

Figure 5.7 Le nombre de processus actifs en mémoire pour l'enrichissement des évènements entrants vs le taux d'éviction des processus terminés vers un stockage disque basé sur un disque persistant

allant de quelques centaines de millisecondes à quelques secondes, et plus de 60% d'entre eux se terminent en moins d'une seconde.

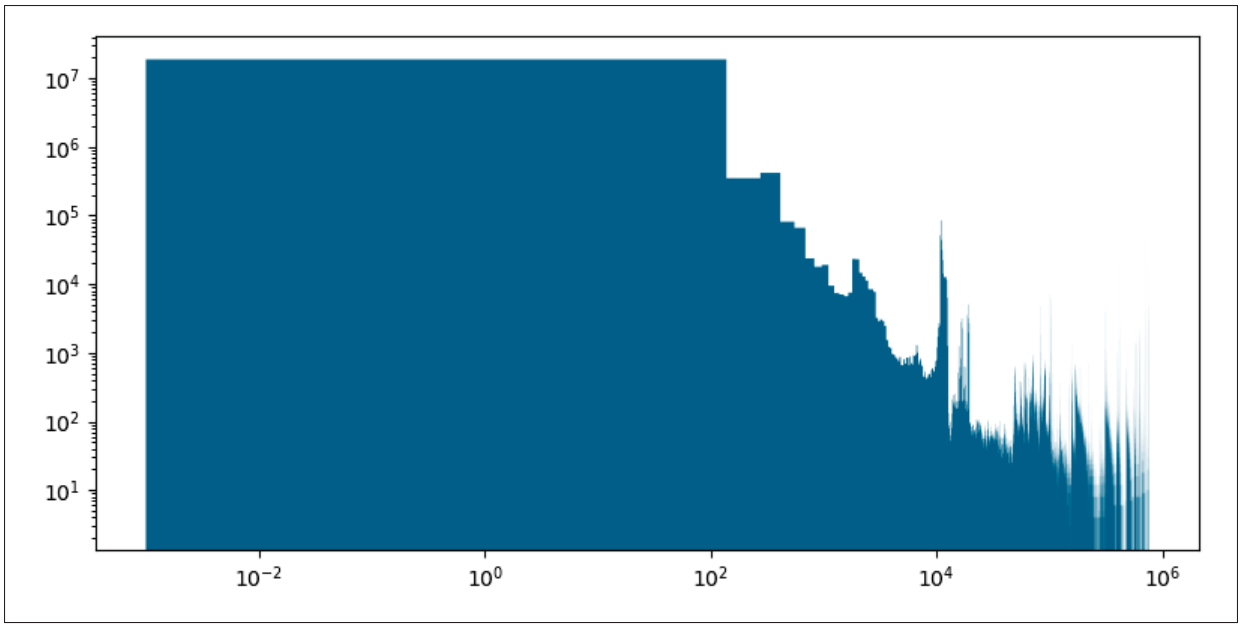

Figure 5.8 Histogramme avec une échelle logarithmique du nombre (axe Y) de processus et leur durée d'exécution en secondes (axe X), enregistrées entre les évènements « CREATE » and « TERMINATE » observés

Dans les environnements corporatifs, la durée de vie maximale d'un processus peut être réduite par des redémarrages système obligatoires imposés par des politiques de sécurité. Sur la Figure 5.8, ces redémarrages sont clairement visibles à droite de l'histogramme. Même dans le cas des services infonuagiques, le sujet initial des études citées ci-dessus, la technologie a évolué vers des conteneurs et des charges de travail sans serveur (Baird, Huang, Munns et Weinstein, 2017), qui sont par nature éphémères et d'une durée limitée.

Ainsi, en réalité, le comportement attendu d'un processus aléatoire est plutôt simple. La complexité de travailler avec des graphes de dépendances au niveau du réseau vient du nombre total d'hôtes, de processus et d'interactions entre eux. Cela ne peut être testé qu'à l'aide d'un jeu de données suffisamment grand et complexe, comme OpTC.

Algorithme 5.2 Reconstruction d'un arbre de processus efficace en mémoire

Input : Un flux $\mathbb{E}$ d'évènements `CREATE` et `TERMINATE` de l'hôte H
Output : Un arbre des processus reconstitués de l'hôte H

Let T be a list of `TERMINATE` events with a corresponding `CREATE` event.
Let A be a set of currently running processes.
Let X and Z be sets of terminated processes kept in memory and moved into persisted storage respectively.
for e *in* $\mathbb{E}$ **do**
    if e *is* `CREATE` **then**
        Check for any PID conflicts :
        if $pid(e) \in A$ **then**
            Move the conflicting record into the terminated set :
            $A \mathrel{-}= pid(e), X \mathrel{+}= pid(e)$
        if $pid(e) \in T$ **then**
            Move the retro-matched record into the terminated set :
            $T \mathrel{-}= pid(e), X \mathrel{+}= pid(e)$
        end if
        $A \mathrel{+}= pid(e)$
    end if
    if e *is* `TERMINATE` **then**
        if $pid(e) \in X$ **then**
            Move the previous record into the terminated set :
            $A \mathrel{-}= pid(e), X \mathrel{+}= pid(e)$
        else
            $T \mathrel{+}= pid(e)$
        end if
    end if
    Evict stale records into persisted storage :
    $Z \mathrel{+}= \{i \mid \exists i \in X,\ age(i) > 5\,minutes\}$
end for

Résultat : Un hôte « corporatif » typique ne se comporte pas d'une façon qui entrainerait une explosion de dépendances, contrairement aux résultats attendus des recherches sur les graphiques de provenance. Au contraire, l'état de l'environnement peut être suivi de manière intelligente et les ressources « terminées » peuvent être déplacées hors de la RAM vers le stockage disque dur persistant, ce qui réduit la consommation de ressources et contribue à un déploiement durable à long terme.

Pour assurer le monitorage des processus en cours d'exécution, il est nécessaire de suivre le flux des évènements « CREATE » et « TERMINATE » pour chaque hôte en utilisant l'Algorithme 5.2. Pour pouvoir reconstruire l'arbre des processus correctement, il s'appuie sur les invariants du système d'exploitation Windows, tels que :

- un processus peut avoir au plus un seul parent ;
- un PID ne peut pas être attribué à deux processus en même temps ;
- une fois qu'un processus est terminé, il ne peut pas être redémarré.

Ces invariants ne s'appliquent pas à toutes les autres entités du graphe de provenance. Par exemple, il est possible qu'il y ait plusieurs connexions parallèles vers la même destination, ou encore qu'une connexion brisée soit rouverte. Cela signifie que les améliorations de performances offertes par cet algorithme ne concernent que les nœuds de processus. D'un autre côté, cet algorithme peut être utilisé dans tout environnement, quels que soient le comportement de base et la structure unique du graphe de provenance.

Cette optimisation permet d'utiliser la mémoire de manière très efficace et de gérer l'ensemble des données de télémétrie de tous les hôtes avec un seul processus analytique, comme le montre la Figure 5.9. La ligne pointillée représente la quantité totale de mémoires utilisées par le processus de corrélation, qui correspond à la somme de tous les arbres de processus de tous les hôtes. Une fois que l'environnement est saturé, la taille de la pile reste constante. Les nouveaux objets peuvent récupérer la mémoire libérée par les processus terminés qui ont été transférés vers le stockage persistant. Par conséquent, l'implémentation nécessite moins de 3 Go de mémoire pour maintenir l'état complet de l'arbre de processus de tous les hôtes pendant la durée de la simulation.

Cependant, il n'est pas possible d'effectuer une optimisation similaire pour d'autres entités dans le télémètre, car elles sont régies par des invariants comportementaux différents et nécessitent des algorithmes séparés qui ne sont pas nécessaires pour la preuve de concept dans le cadre de cette recherche.

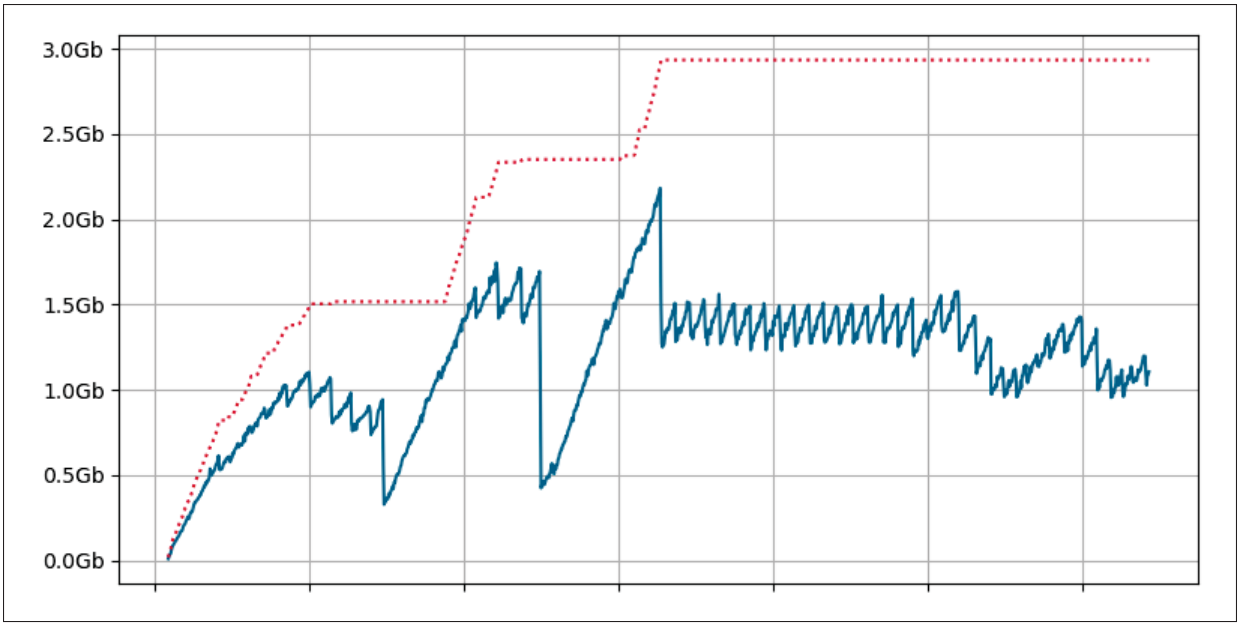

Figure 5.9 Consommation de mémoire du système par le corrélateur d'évènements pour maintenir la liste en mémoire de tous les processus actifs de tous les hôtes

5.4.6 Sélection des évènements candidats en Phase 1

Maintenant que tous les composants de préparation des données nécessaires sont prêts, il est possible de commencer avec la première étape de détection d'anomalies. En rappel, l'objectif de cette étape n'est pas de générer les alertes finales de l'IDS, mais plutôt d'identifier les candidats les plus suspects pour une analyse plus approfondie, ce qui permet de réduire les couts de calcul. À cette phase, la priorité est d'assurer une vitesse et la capacité de traitement des évènements. L'option la plus simple consiste à utiliser des algorithmes relativement simples et rapides, comme les réseaux neuronaux profonds à faible profondeur ou les forêts aléatoires.

Le pipeline de traitement de données bout à bout permet de surveiller l'état de l'environnement et d'extraire des caractéristiques complexes, à la fois temporelles et spatiales, qui seraient impossibles à obtenir en utilisant simplement les évènements bruts sous forme d'une fiche de données opaque.

Cependant, il est préférable d'éviter d'utiliser les attributs sémantiques des évènements bruts dans ce travail, par exemple les noms de fichiers. Comme c'était déjà démontré dans les recherches précédentes (Ur Rehman *et coll.*, 2024) et dans le Chapter 4, ces attributs transmettent un signal très fort et permettent d'identifier les activités malveillantes avec une confiance très élevée. Il est important de souligner que, dans un contexte réel, il serait tout à fait acceptable d'utiliser ces données pour renforcer davantage la qualité des détections effectuées par un système IDS. Cette restriction arbitraire a été imposée pour mettre en évidence les capacités de détection de l'architecture IDS proposée.

Par contre, en raison des limites du langage de programmation Go, il faut accorder moins d'importance au but de l'implémentation « bout à bout ». En effet, il n'existe actuellement aucune bibliothèque de qualité pour le langage Go offrant des fonctionnalités d'apprentissage automatique ou d'analyse de graphes. Puisque l'implémentation de ces outils se trouve hors du cadre de ce travail, la décision était prise d'utiliser des bibliothèques riches en fonctionnalités disponibles pour ces tâches en Python.

Pour cette raison, le reste du projet est implémenté en Python. Cela entrainera une réduction significative de la vitesse du pipeline, car tous les avantages sont perdus en termes de performances offertes par un langage compilé, de parallélisme natif et d'optimisation à bas niveau. Pour pouvoir terminer toutes les expériences dans un délai raisonnable, il a été nécessaire de limiter l'évaluation à un sous-ensemble des données. Cela se fait sans perte de généralité, et le traitement complet du jeu de données peut être réimplémenté en Go, à condition que toutes les bibliothèques nécessaires soient disponibles. D'un côté positif, l'utilisation de Python donne accès à des bibliothèques de traitement de données de haute qualité, comme *scikit-learn* (Pedregosa *et coll.*, 2011), qui permet de mesurer avec précision le rendement des modèles conformément aux meilleures pratiques actuelles. De plus, grâce à *NetworkX*, on peut avoir accès à une grande variété d'algorithmes d'analyse des graphes, ce qui permet d'appliquer ces méthodes au graphe de provenance.

Tableau 5.1 Attributs utilisés dans l'évaluation de la Phase 1

Nom de caractéristique	Type	Source	Normalisation
Durée de processus	temporal	l'arbre des processus	StandardScaler
Longueur de la ligne de commande	numérique	l'évènement brut	StandardScaler
Nombre des composants de path	numérique	l'évènement brut	StandardScaler
Demi-degré entrant d'acteur	spatial	graphe de provenance	Log1p
Demi-degré sortant d'acteur	spatial	graphe de provenance	Log1p
Demi-degré entrant d'objet	spatial	graphe de provenance	Log1p
Demi-degré sortant d'objet	spatial	graphe de provenance	Log1p
Types d'évènement sur l'arc	catégorique	graphe de provenance	-

Avec les caractéristiques sémantiques de la ligne de commande de l'évènement processus exclus, les trois ensembles de caractéristiques suivants sont disponibles pour l'entrainement.

- Les attributs présentent directement dans les évènements bruts, tels que le type d'évènement et le type d'objet, ainsi que des informations générales sur le processus, comme la longueur de la ligne de commande et le nombre de composants de dossier dans le chemin. Cependant, les attributs sémantiques utilisés dans les expériences 2 et 3 du Chapitre 4, tels que le nom spécifique du programme et le nombre d'arguments, ne sont pas inclus.
- Informations temporelles sur la chronologie d'exécution du processus obtenues à partir de l'arbre des processus reconstruit. Ce contexte temporel très précieux n'est pas disponible dans les journaux bruts et ne peut pas être utilisé pour la détection d'intrusion en examinant un évènement à la fois. Par exemple, il est possible d'utiliser des caractéristiques telles que la durée d'exécution du processus, le nombre de sous-processus, des informations sur le parent et la profondeur de l'arbre des processus.
- Informations spatiales extraites du graphe de provenance. Des recherches antérieures approfondies sur les graphes de provenance ont démontré leur capacité à découvrir des relations complexes entre les composants du système, ce qui permet de détecter même les attaques subtiles. Cependant, le principal inconvénient des méthodes actuelles est qu'elles nécessitent beaucoup de ressources. Pour maintenir un taux élevé d'évènements, uniquement le voisinage local du nœud est utilisé. Cette approche permet de toujours obtenir certaines

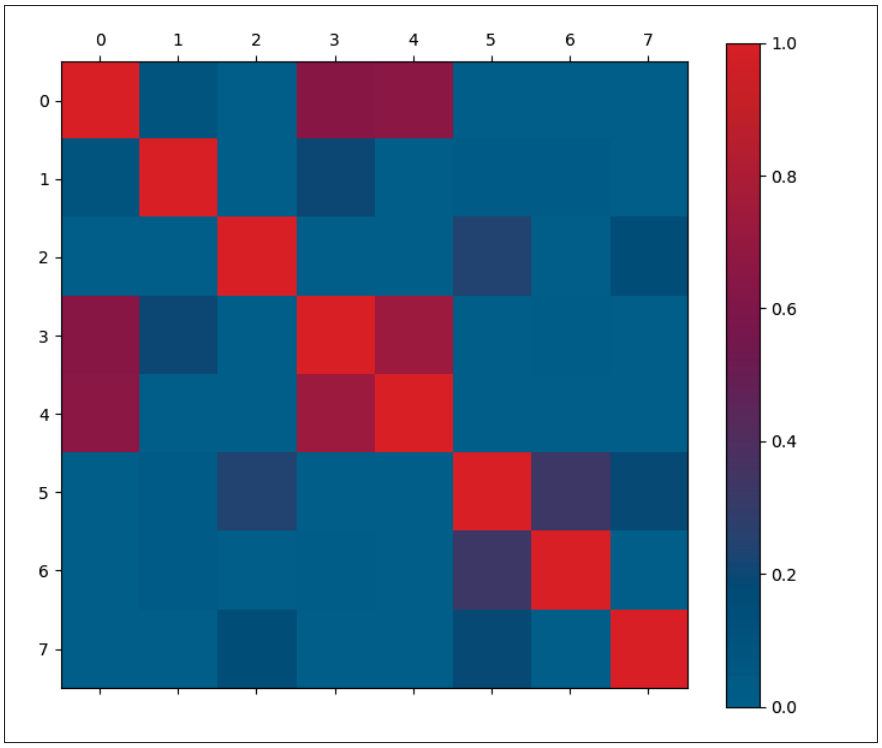

Figure 5.10 La matrice de corrélation des caractéristiques

informations structurelles sur les évènements entrants, tout en évitant des algorithmes complexes et couteux, comme les réseaux neuronaux de graphes.

En théorie, à ce stade, il est possible d'employer n'importe quel algorithme d'apprentissage automatique capable de traiter des données chronologiques ou tabulaires, que ce soit des réseaux neuronaux profonds ou récurrents. Cependant, la plupart de ces méthodes exigent des données étiquetées, qui ne sont pas souvent disponibles dans un contexte d'entreprise, comme cela a déjà été abordé plus haut.

Dans cette évaluation, on préfère l'algorithme de la forêt aléatoire à des méthodes d'entrainement supervisées complexes. Cet algorithme répond à tous les besoins de cette recherche et a fait preuve de sa capacité à obtenir de bons résultats en matière de détection d'anomalies avec un minimum de ressources. Pour résoudre le problème du manque d'étiquettes, les résultats de Kairos IDS (Cheng *et coll.*, 2024) servent d'inspiration. Les auteurs de cette étude tentent de prédire le type

Tableau 5.2 Réduction de la quantité de données par la Phase 1

Ensemble d'apprentissage	Ensemble de test	Anomalies	Taux de réduction
12,277,359	3,275,790	34,968	98.93%

de l'arc en fonction de l'état reconstruit du graphe de provenance. En utilisant de telles étiquettes proxys, il est possible d'entrainer un algorithme supervisé à partir du comportement normal du système et de détecter toute déviation. Cette expérience utilisera le champ « user ».

Chaque évènement du journal contient un champ associé à une identité d'utilisateur. Comme c'était déjà abordé à la Section 4.3.1, il existe deux grands types d'identités : les comptes de services système et les identités d'utilisateurs. En utilisant ce champ comme étiquette proxy, le modèle peut distinguer entre les opérations système et les activités normales d'un utilisateur sur l'hôte. Imaginons qu'un hôte compromis commence à exécuter des opérations qui ressemblent à un service système automatisé à cause d'une porte dérobée (backdoor). Cela entrainera une mauvaise classification et une prédiction de confiance faible de la part de la forêt aléatoire. Ces évènements suspects peuvent faire l'objet d'une analyse détaillée.

La Figure 5.11 présente la matrice de confusion du modèle de la Phase 1. Le classificateur forêt aléatoire tente de prédire la valeur de l'identité de l'utilisateur de chaque évènement en fonction du contexte réseau enrichi. Ce champ sert d'étiquette proxy pour distinguer entre le comportement typique des comptes de service et celui des utilisateurs simulés qui coexistent sur le même hôte. Outre l'identité principale de l'utilisateur, il y a trois comptes de service majeurs qui génèrent la plupart des évènements. De plus, un petit groupe de comptes de support est regroupé sous l'étiquette « autre » pour une présentation simplifiée. Si la valeur prédite de l'étiquette correspond à la valeur réelle du champ dans l'évènement, il est classé comme normal. Un manque de correspondance est traité comme un évènement anomal qui fera l'objet d'une analyse plus approfondie dans la prochaine étape.

La Figure 5.12 montre la répartition des anomalies détectées pendant la période d'évaluation sélectionnée. La taille de chaque colonne représente le nombre d'évènements individuels classés

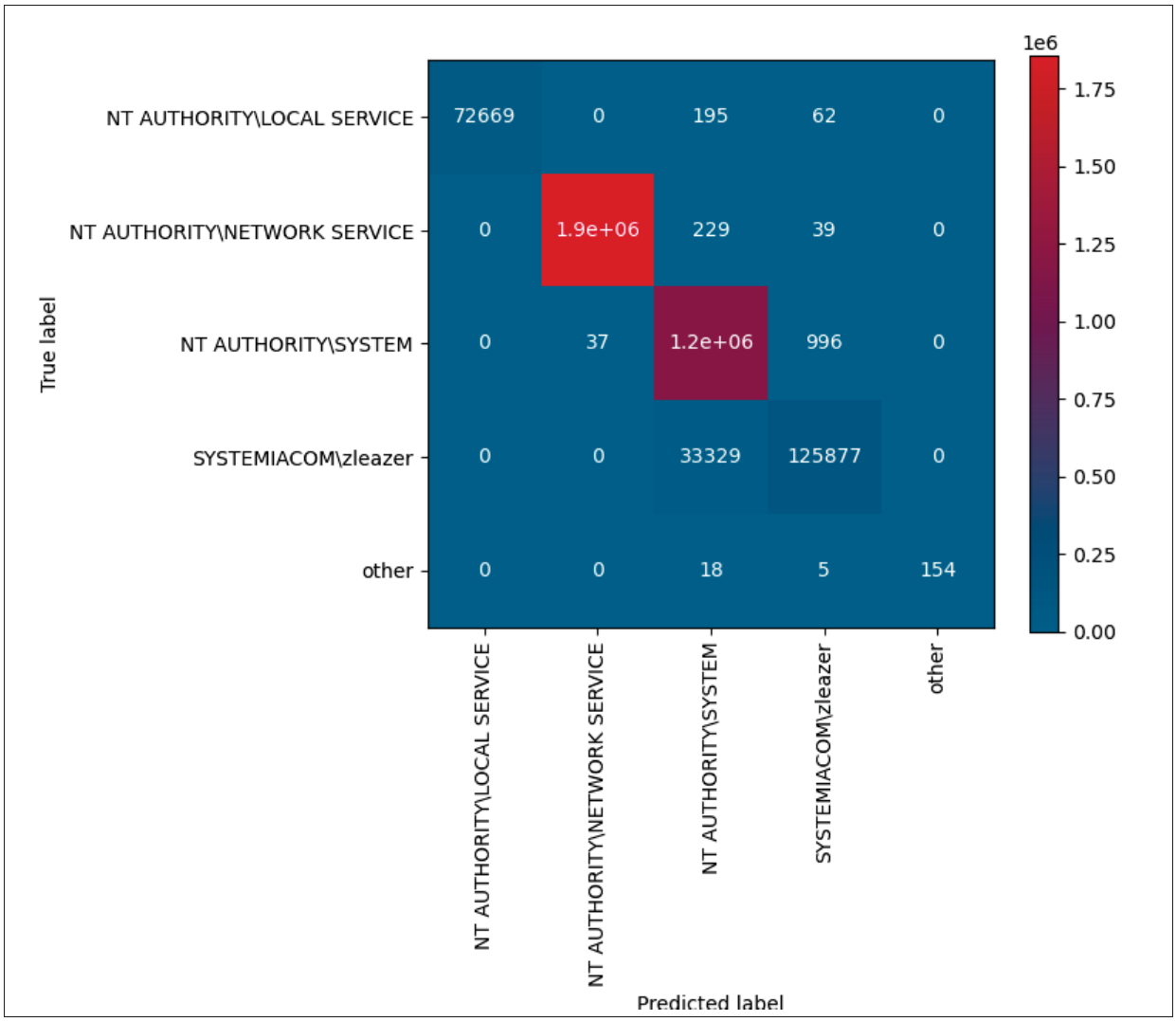

Figure 5.11 Matrice de confusion du modèle de la Phase 1

comme normaux (blue) ou anormaux (rouge) sur une période de 30 minutes. On remarque que le nombre d'anomalies constitue une très faible proportion du nombre total d'évènements classés comme étant normaux par le modèle. Chaque heure, il y a quelques anomalies, mais leur nombre est trop faible pour qu'elles soient visibles sur le graphique. La majorité des évènements se situe près du moment où l'équipe rouge était active sur cet hôte. Cela confirme que le pipeline de traitement de données peut préserver le signal malveillant. Il est ainsi possible de réduire le taux d'évènements de 99%, comme le montre le Table 5.2, sans perdre d'informations ni compromettre la complétude du graphe de provenance.

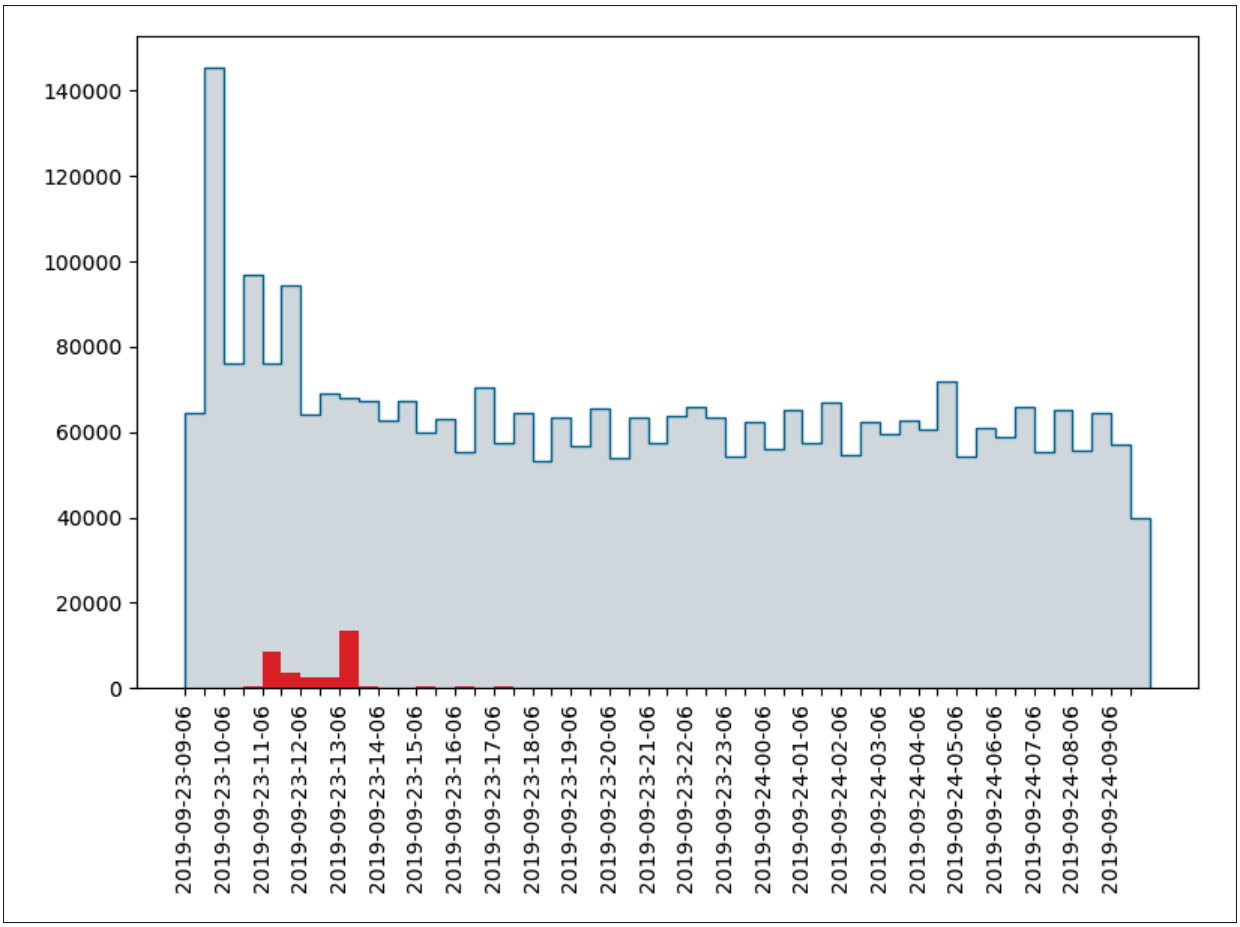

Figure 5.12 La chronologie des anomalies détectées pendant la période d'évaluation choisie

La Section 2.2 soutient que les mesures de performance standard, telles que la précision et le rappel, ne reflètent pas adéquatement les résultats en matière de cybersécurité. Cependant, elles demeurent utiles pour évaluer les résultats par rapport à des méthodes alternatives. La Table 5.3 présente les indicateurs standards, décomposés selon la granularité de la détection, qui correspond à différentes méthodes d'agrégation des données.

Tous ces chiffres ont été obtenus grâce aux mêmes données d'évaluation. Ils peuvent tous être considérés comme appropriés pour certaines utilisations commerciales, en fonction de leurs exigences de sécurité spécifiques. Cependant, la différence marquée entre, par exemple, l'évènement et le processus met en évidence le fait que ces métriques grossières masquent de

Tableau 5.3 Indicateurs de performance de la Phase 1 à divers niveaux de granularité

	Évènement	**Processus**	**Commande**	**Hôst**
Exactitude	0.98	0.96	0.90	1
Précision	0.87	0.55	0.84	1
Rappel	0.47	0.97	0.95	1
Mesure F1	0.61	0.70	0.89	1
MCC	0.64	0.72	0.81	1

nombreux détails importants sur les alertes générées. Cela rend impossible la comparaison directe des résultats présentés dans différentes publications, même si elles sont évaluées avec le même ensemble de données de référence. De plus, la possibilité de régler la détection à différents niveaux de granularité permet aux auteurs de présenter les performances de l'IDS de manière favorable.

En prenant l'exemple de la granularité des processus, on peut atteindre un taux de précision de 84%. Dans certains domaines d'application de l'apprentissage automatique, cela pourrait être considéré comme un résultat satisfaisant. Cependant, en raison de l'oublie de la fréquence de base, qui a été introduite dans la Section 2.3, cela signifie que le modèle classe incorrectement des milliers d'évènements comme de fausses alertes. La plupart des architectures IDS ne tolèreraient pas cela : le SOC serait submergé d'alertes, ce qui entrainerait un échec du déploiement IDS.

Dans cette situation, il existe deux options courantes. La première consiste à ajuster le modèle pour l'améliorer en termes de précision, ce qui peut entrainer un surapprentissage. La seconde option est d'ajouter une étape de posttraitement pour réduire le taux de faux positifs en utilisant des euristiques.

L'architecture à deux niveaux permet de prévenir complètement ce problème. Pour réitérer, l'objectif de la première étape n'est pas de détecter des attaques de haute confiance, mais d'identifier des évènements suspects potentiels et de les signaler pour un examen plus approfondi dans la deuxième étape. Chaque étape ayant un objectif distinct, les deux modèles peuvent être optimisés individuellement pour atteindre le niveau de protection désiré. La première phase

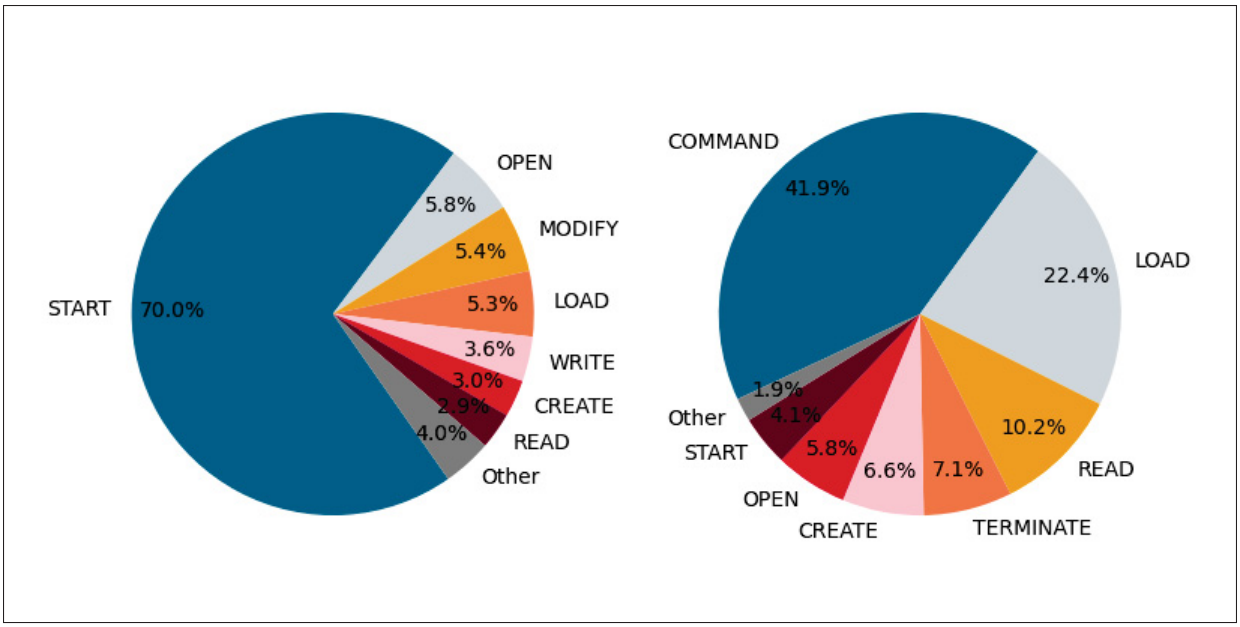

Figure 5.13 La comparaison de la répartition des types entre tous les évènements (gauche) et les anomalies détectées (droite)

peut être optimisée pour obtenir un taux de rappel élevé, alors que la deuxième phase peut se concentrer sur la précision, ce qui contribue à une mise en œuvre pratique du système de détection d'intrusion.

Contrairement aux expériences du Chapitre 4, qui examinaient un seul type d'objet de journaux, cette évaluation porte sur tous les évènements. La Figure 5.13 montre la différence entre la répartition des types entre tous les évènements et les anomalies détectées. Cela renforce l'idée que différents journaux ont des valeurs de sécurité variables, ce qui justifie leur hiérarchisation lors de leur collecte pour atteindre un équilibre entre les couts opérationnels et la performance de détection. Tenter d'extraire du sens à partir de toutes les données, sans tenir compte de leur valeur pour l'entreprise, peut nuire à la mise en œuvre industrielle, car il est possible que le type de journal requis ait peu de valeur pour l'entreprise et ne soit donc pas collecté.

On peut faire une autre observation intrigante à propos de la granularité d'évènement qui est caractérisée par une grande précision et un faible taux de rappel. Cela se produit parce que les processus malveillants génèrent de nombreux évènements pendant l'énumération du système. L

modèle est capable d'identifier avec une grande précision de tels processus, mais il ne détecte pas nécessairement tous les évènements générés par le même processus. Cependant, un faible taux de rappel ne signifie pas nécessairement une mauvaise performance du modèle. En effet, si le partitionnement des mêmes données est fait selon le processus, il est toujours possible de détecter la majorité des activités malveillantes. L'explication est que, dans le domaine de la cybersécurité, il n'est pas nécessaire d'identifier chacun des évènements malveillants pour détecter une attaque.

Une seule alerte de haute qualité suffit pour détecter une attaque en cours et déclencher le processus de réponse à l'incident. Cette caractéristique est souvent désignée par « l'avantage du défenseur » dans l'industrie, ce qui explique pourquoi les équipes de sécurité accordent une grande importance aux faux positifs plutôt qu'aux faux négatifs (Vermeer *et coll.*, 2022).

En matière de cybersécurité et de détection d'intrusions, la relation entre les mesures de précision et de rappel n'est pas évidente, et n'a pas été suffisamment étudiée.

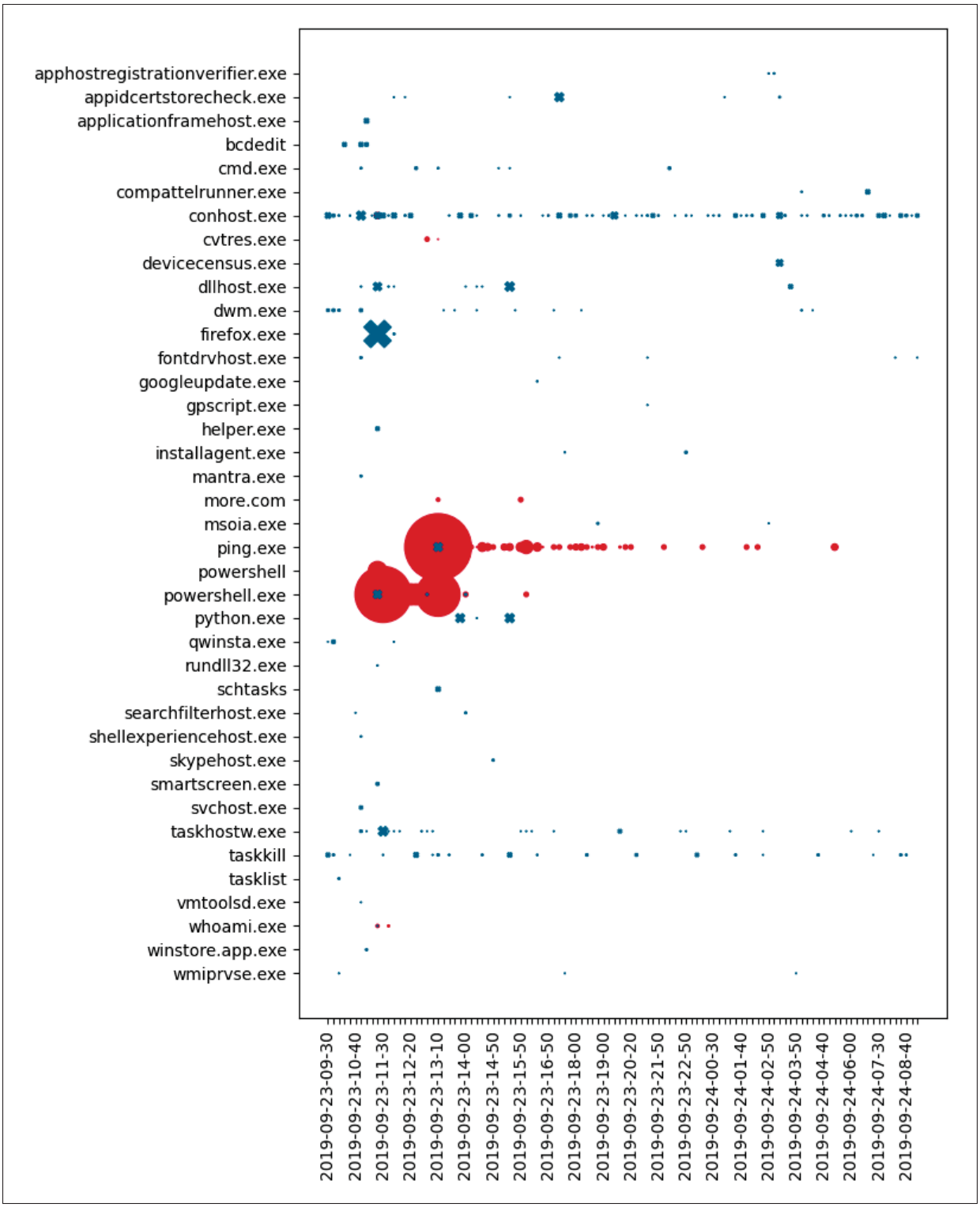

Figure 5.14 Analyse des vrais positifs (rouge) et faux positifs (blue) en fonction du nom de l'exécutable

CONCLUSION ET RECOMMANDATIONS

Cette recherche porte sur la détection de la persistance postcompromise dans un réseau d'entreprise après une cyberattaque. C'est un enjeu qui prend de l'importance, puisque la sévérité des attaques augmente chaque année. Ce problème a fait l'objet de nombreuses recherches universitaires au cours des trente dernières années, mais très peu d'idées ou d'outils développés ont été adoptés par l'industrie.

L'étude des principaux défis auxquels font face tant les chercheurs que les spécialistes de l'industrie a permis de comprendre pourquoi les publications universitaires ont un faible taux de réussite dans les milieux réels. De nombreux obstacles existent en raison de l'absence de données publiques de qualité décrivant avec précision les conditions de réseau pendant une attaque informatique.

L'analyse du jeu de données OpTC a démontré que c'est le plus grand et le plus réaliste des ensembles de données publiques actuellement disponibles. Toutefois, à cause de l'absence de documentation et d'étiquettes, il n'est pas souvent utilisé dans les travaux existants. Pour déterminer si c'est une bonne représentation d'une entreprise moderne, cette recherche a adopté la même démarche qu'un analyste de sécurité lors d'une enquête dans un environnement inconnu.

Après une étude approfondie, plusieurs artéfacts importants ont été produits :

- la description détaillée de l'environnement qui a servi à la construction du jeu de données, qui était reconstitué à partir des journaux ;
- un processus fiable d'étiquetage permettant de convertir le document de la réalité du terrain en un ensemble d'étiquettes adaptées à l'utilisation dans des techniques d'apprentissage machine conventionnelles ;
- une évaluation de référence avec plusieurs méthodes de détection d'anomalies simples qui ont permis de mesurer l'intensité du signal malveillant dans les données.

La compréhension approfondie de l'environnement a révélé que ce jeu présente des limites typiques dans ce domaine : une nature confinée, des attaques triviales et un manque de diversité comportementale réelle.

En même temps, il est devenu clair que ce jeu de données présente plusieurs traits importants de réseaux réels : une grande quantité de données, une complexité de forme de journaux, une couverture incomplète, des activités administratives, des évènements ambigus, etc. Ces traits font qu'il est difficile de transposer les résultats de la recherche académique vers un contexte commercial.

Ensuite, la recherche s'est concentrée sur la mise en œuvre d'un cadre IDS innovant, qui aborde explicitement ces problèmes, avec l'objectif de fournir aux analystes de sécurité les outils nécessaires pour adapter l'IDS aux contraintes uniques de son réseau. La conception repose sur une réflexion critique sur la structure des opérations dans un SOC moderne. L'observation que la détection d'alerte est souvent divisée en deux tâches distinctes, soit le triage et l'enquête, qui ont des objectifs et des résultats différents, a déterminé l'architecture de l'IDS en deux étapes.

- La Phase 1 vise à détecter des évènements suspects potentiels pour diminuer le nombre d'opérations couteuses. Elle est conçue pour maximiser le taux de rappel, mais peut entrainer une baisse de la précision.
- La Phase 2 reçoit un nombre réduit de résultats de l'étape précédente. Elle effectue une analyse approfondie, mais couteuse, qui utilise un contexte environnemental étendu. Son objectif est d'améliorer la précision du résultat afin d'éviter que le SOC soit inondé de faux positifs.

La conception « bout à bout » s'étend du gouffre qui sépare les méthodes de détection actuelles et la nécessité industrielle de s'adapter aux contraintes complexes des réseaux réels. Les résultats de cette recherche démontrent qu'en concevant adéquatement, il est possible de résoudre, ou

du moins d'atténuer, des problèmes fondamentaux, tels que la perte d'évènements, la dérive conceptuelle, la consommation de ressources, la livraison hors ordre et les sémantiques confuses. En énonçant clairement les limites et les suppositions concernant les données, il est possible de fournir aux spécialistes de la sécurité les outils pour évaluer la compatibilité de leur réseau avec une architecture spécifique d'IDS. Cela permet également de s'adapter aux contraintes uniques et à l'évolution constante.

Évidemment, ce travail présente les mêmes limites que celles critiquées dans la revue de l'état de l'art, notamment en étant trop dépendant de la nature confinée de l'ensemble de données de référence, ou encore en sélectionnant des données qui permettent d'obtenir le meilleur rendement possible de l'IDS proposé.

Il existe une raison pour laquelle le défi de la détection d'intrusion demeure non résolu depuis plus de trois décennies. Les résultats de cette recherche ne fournissent pas la solution qui avait échappé à plusieurs chercheurs. Toutefois, les idées et les résultats présentés peuvent être considérés comme un pas vers la réconciliation des perspectives de l'académie et de l'industrie.

La bonne compréhension des véritables contraintes auxquelles font face les analystes en sécurité dans leur travail quotidien permet d'aborder ces contraintes à un niveau fondamental lors de la conception de l'IDS. Ainsi, les chances d'un déploiement industriel réussi augmentent.

Il y a de l'espoir. En effet, la recherche et le développement de méthodes de détection actuelles sont de plus en plus issus d'attaques réelles qui ont frappé l'industrie ces dernières années. La collaboration entre l'académie et l'industrie devient une tendance de plus en plus marquée. Enfin, le nombre croissant de chercheurs expérimentés exposent les limites des approches traditionnelles pour résoudre ce problème.

L'élément manquant est la prise en compte des limites complexes, des contraintes et des invariants des systèmes réels qui définissent le travail quotidien d'un analyste de sécurité, ce qui constitue

l'objectif principal de ce travail. Ceci est possible en utilisant des données publiques qui reflètent de manière fiable l'état interne du réseau avant, pendant et après une attaque. L'absence de ces données suggère qu'il est difficile d'imiter avec précision les subtils aspects qui différencient les actes bénins des attaques malveillantes.

Cependant, les résultats de cette recherche suggèrent qu'une solution pourrait être aussi simple que la création d'un ensemble de données très large. Puisque la génération et la collecte de données d'un volume de téraoctets exigent un réseau suffisamment complexe, celui-ci doit être construit avec des technologies réelles. Cela n'est pas différent de ce que ferait le département TI d'une petite ou moyenne entreprise. En faisant cela, les complexités des environnements réels se retrouveront naturellement dans les données collectées. La tâche du chercheur sera alors de les reconnaitre et de les intégrer correctement dans l'architecture de l'IDS.

ANNEXE I

L'ILLUSTRATION DU PROCESSUS D'ÉTIQUETAGE

Ce matériel additionnel présente les avantages de la méthode d'étiquetage à partir d'un exemple concret tiré de la deuxième journée de la campagne de l'équipe rouge.

La première étape de l'algorithme consiste à identifier manuellement un ensemble d'évènements d'entrée dans les logs bruts en fonction de la description contenue dans la réalité du terrain, en utilisant cet exemple :

> *09/24/19 15 :04 :14 – On Sysclient0005 via RDP session, exported 3.5 gb exfil file allgona.zip*

Cette entrée du journal contient deux indicateurs de compromis pour rechercher des activités malveillantes : le nom de l'hôte ciblé `SysClient0005` et le nom du fichier (avec une erreur de frappe) `allgone.zip`. En recherchant toutes les opérations impliquant ce fichier, on trouve un processus avec PID 6864 et une chaine de commande « `movingonup.exe fun.com 81` ». Cette information forme l'ensemble d'évènements d'entrée S_p pour l'algorithme.

Dans le modèle théorique décrit dans la Section 3.6.2, les relations de processus enfant-parent sont des dépendances *faibles* qui nécessitent une analyse manuelle. Le parent du processus initial est `cmd.exe`, qui est classé comme étant malfaisant, et continuons à son parent, qui est `Explorer.EXE`.

Le rapport de l'équipe rouge indique que l'activité sur l'hôte `SysClient0005` est exécutée via une connexion RDP. Cela implique que l'exécution de ce processus est juste une partie d'une session utilisateur interactive standard et n'est pas malveillante en soi. Le processus `Explorer.EXE` est étiqueté comme bénin dans l'ensemble des étiquettes de sortie pour assurer un fonctionnement déterministe de l'algorithme d'étiquetage. L'arbre de processus découvert est représenté dans la Figure I-1, avec le processus initial en rouge et le processus bénin en vert.

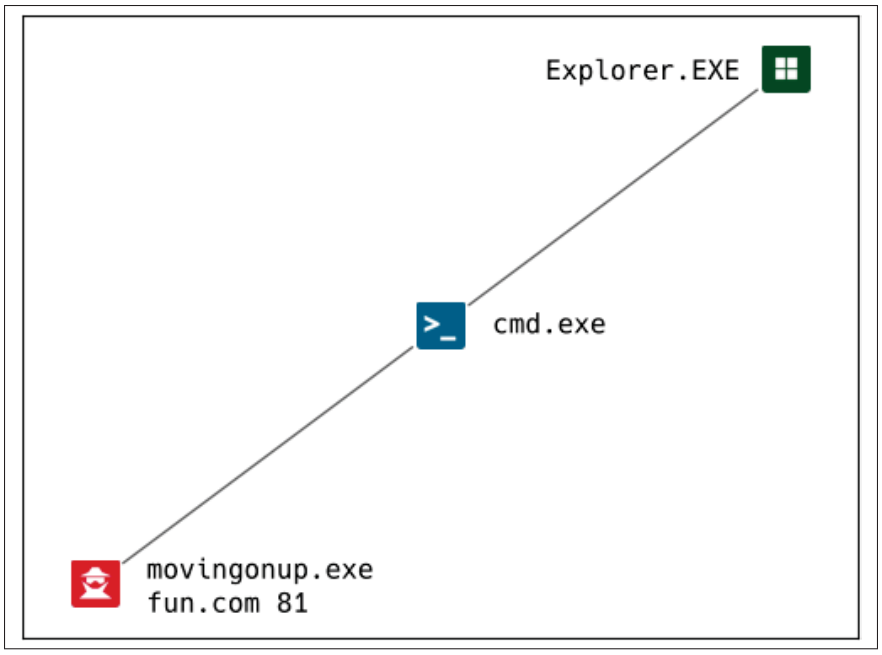

Figure-A I-1 Découverte itérative de processus malveillants à partir d'une seule entrée dans le journal de l'équipe rouge. L'élément rouge représente l'entrée initiale de l'algorithme de recherche

L'équipe rouge aurait pu exécuter d'autres processus malveillants pendant la même session RDP, donc tous les enfants du processus `Explorer.EXE` sont ajoutés à la file d'attente pour une analyse manuelle. Dans les prochaines itérations, l'analyste de sécurité identifie plusieurs processus bénins associés aux tâches de fond, qui sont typiques de l'environnement.

Certains processus enfants comportent des indications claires d'activité malveillante, et sont étiquetés comme telles par l'analyste. L'arbre des processus découverts jusqu'à présent est représenté à la Figure I-2.

La dernière étape de l'algorithme consiste à propager sans condition l'étiquette malveillante à tous les enfants d'un processus malveillant. Cette opération se fait sans aucune entrée de l'analyste. L'arbre de processus résultant compte 24 processus malveillants et 14 processus bénins, comme le montre la Figure I-3.

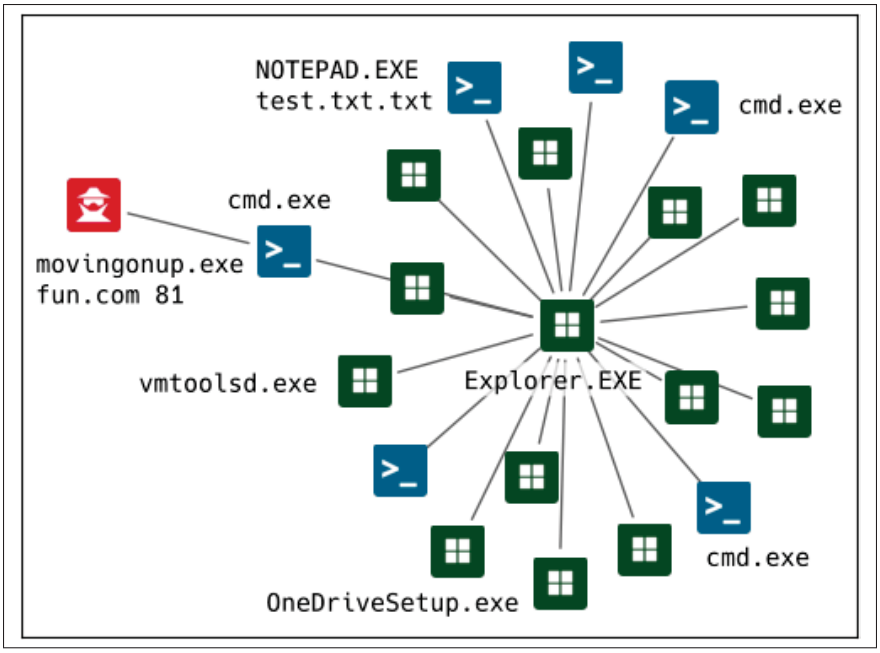

Figure-A I-2 Étape deux du processus itératif d'étiquetage. Les éléments en vert représentent des processus marqués comme *bénins* par l'analyste, y compris le processus `Exlorer.EXE` au centre du graphe

L'algorithme continue à inspecter les logs d'autres hôtes pour découvrir tout processus additionnel qui se connecte aux mêmes adresses IP malveillantes et l'origine de la session RDP. Pour des raisons d'espace, les itérations supplémentaires ne sont pas détaillées.

En examinant les processus malveillants étiquetés, on découvre des preuves de l'activité de reconnaissance présentées dans le Tableau I-1. Ces actions de l'équipe rouge **ne sont pas incluses dans le document de la réalité du terrain**, ce qui souligne le problème de la vérité incomplète, comme discuté dans la Section 3.5. Cependant, puisque ces processus sont liés par une dépendance forte, on peut rejeter la possibilité que les activités normales d'un administrateur système se produisent en même temps.

Pour finaliser l'étiquetage, l'étiquette malveillante est diffusée à tous les évènements bruts liés aux processus découverts. Cela représente un total de **3468 évènements**, qui ont tous été

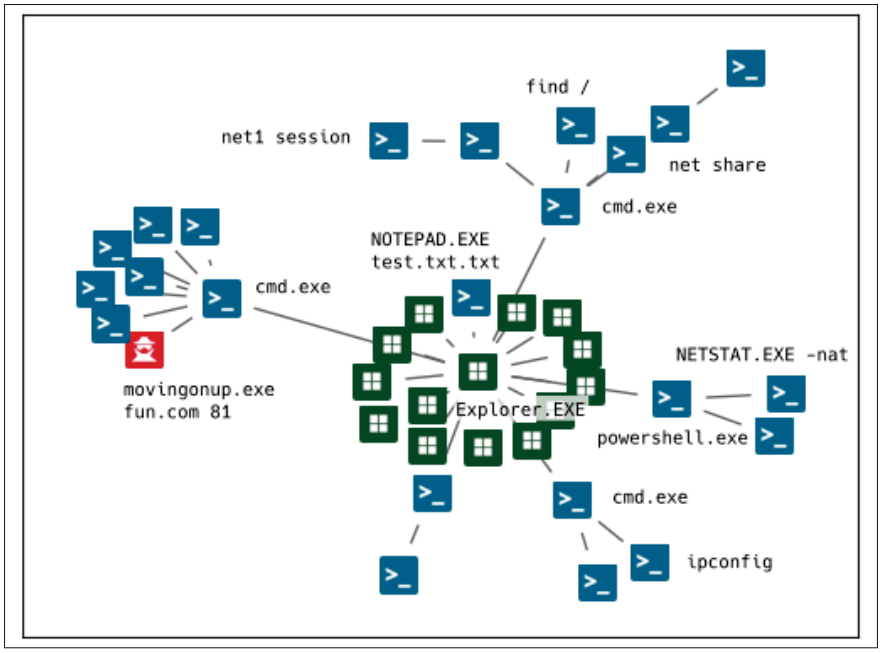

Figure-A I-3 La troisième étape du processus itératif d'étiquetage sert à diffuser l'étiquette

dérivés de manière déterministe à partir **d'un seul évènement d'entrée**, et associés à une ligne spécifique dans le document de base de vérité.

Tableau-A I-1 Exemples d'activités non documentées de reconnaissance

Hôte	**PID**	**Processus**
SysClient0005	7384	`net share`
SysClient0005	7988	`find /`
SysClient0005	6688	`NETSTAT.exe -nat`
SysClient0005	7912	`net1 session`
SysClient0005	7512	`ipconfig`
SysClient0005	6040	`NOTEPAD.EXE test.txt.txt`

ANNEXE II

L'INTERFACE EN MODE TEXTE D'ÉTIQUETAGE

Cette annexe présente l'interface utilisateur développée pour le processus d'étiquetage du jeu de données OpTC, comme discuté dans le Chapitre 3.

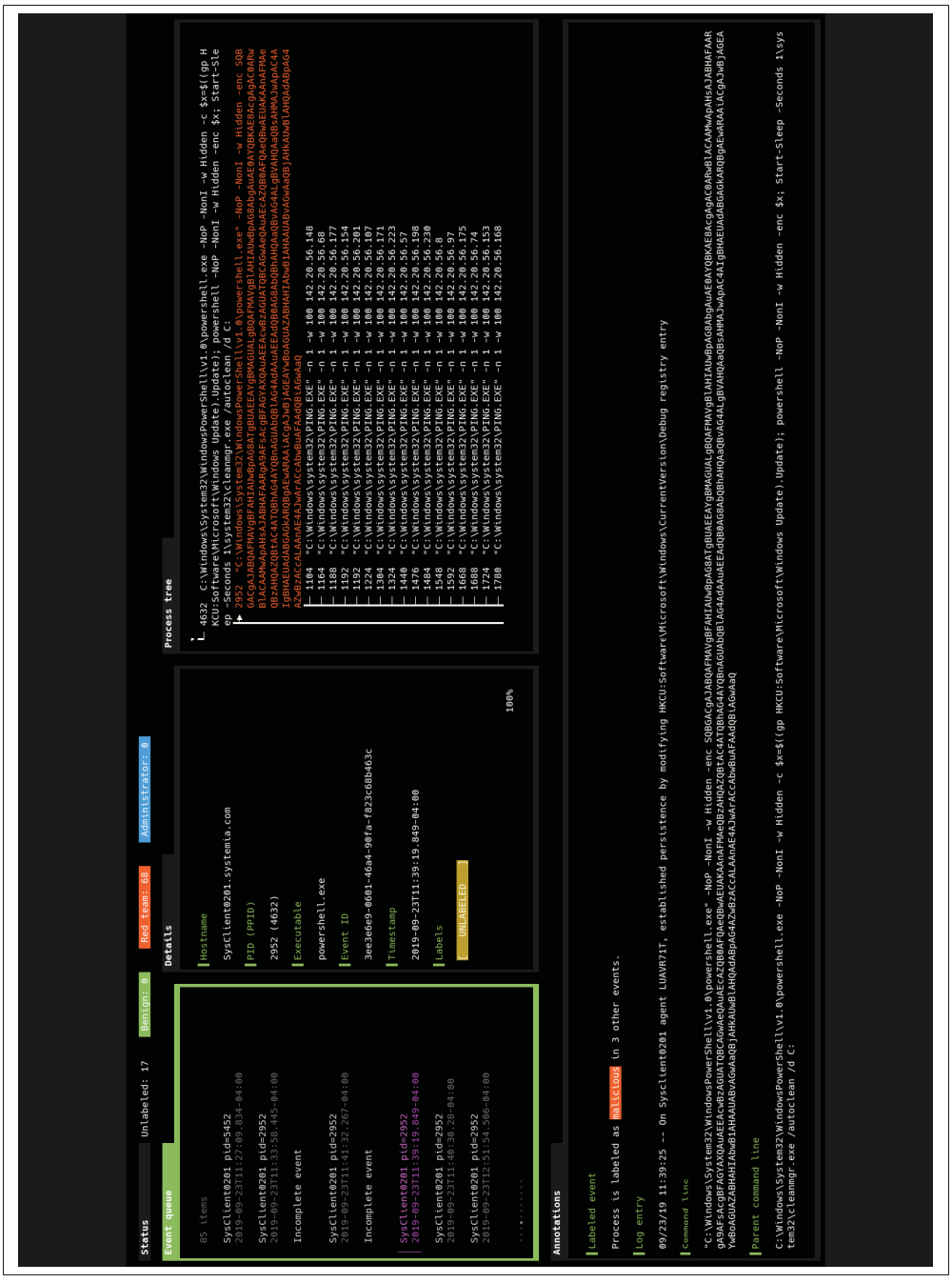

Figure-A II-1 Cette figure montre l'écran principal de l'interface utilisateur en mode texte présenté à l'analyste de sécurité durant le processus d'étiquetage. Les panneaux affichent les informations suivantes : la liste des évènements à étiqueter ; les détails sur l'hôte et le processus ; l'arbre des processus reconstitués, ce qui inclut la ligne de commande, le processus parent, frères et enfants ; la liste des étiquettes déjà assignées ; l'entrée de journal correspondante de l'équipe rouge

Figure-A II-2 Cet écran permet à l'analyste d'attribuer des étiquettes aux évènements bruts. Il affiche l'évènement complet, tel qu'il est représenté dans le jeu complet. Pour prendre en considération toutes les combinaisons possibles de son origine, chaque évènement brut est annoté avec **quatre catégories**

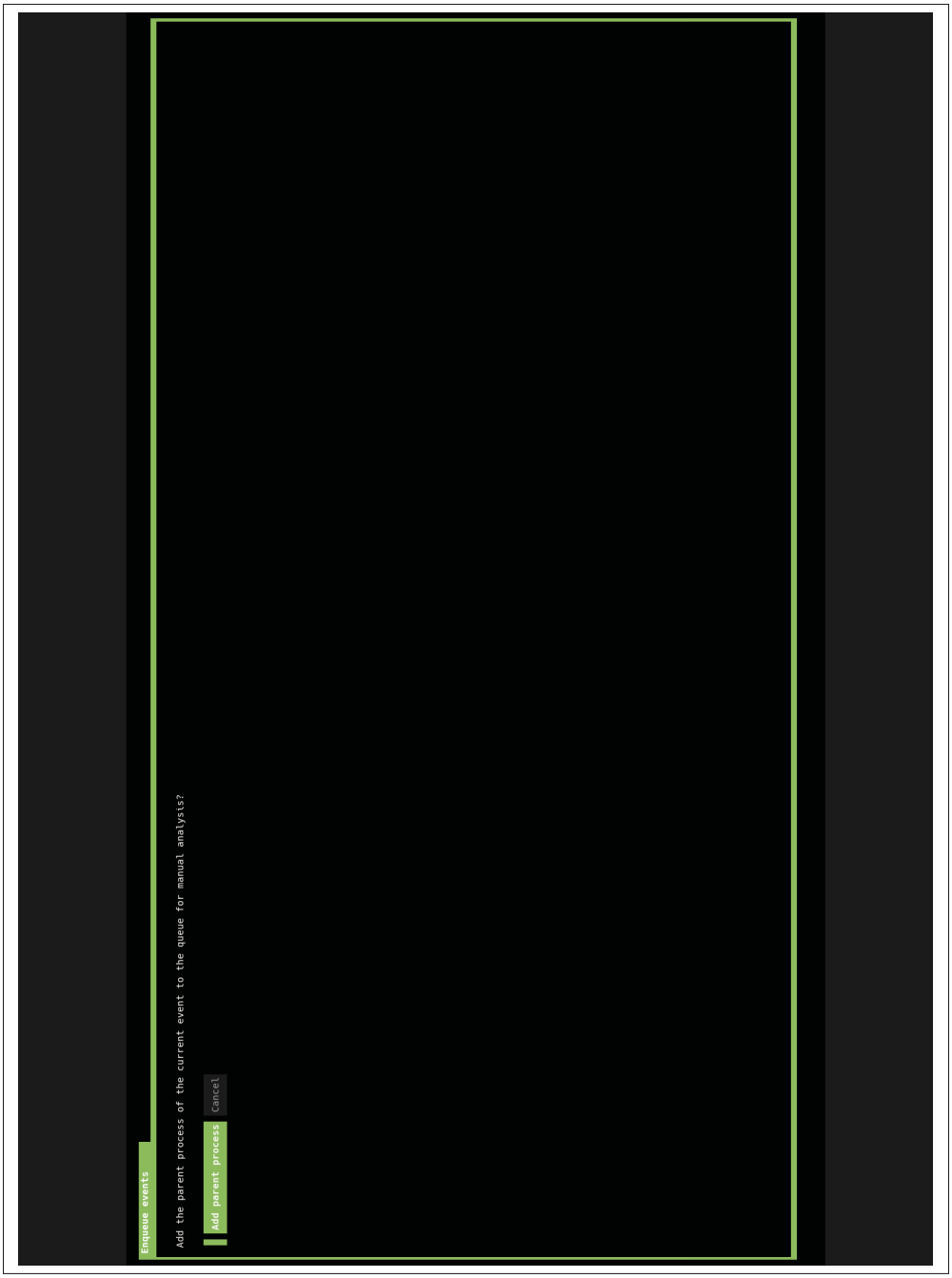

Figure-A II-3 Si, lors de l'analyse de corrélation, l'analyste identifie que le processus parent présente également des signes d'activité malveillante, cet écran peut être utilisé pour l'ajouter à la file d'attente pour une corrélation complète

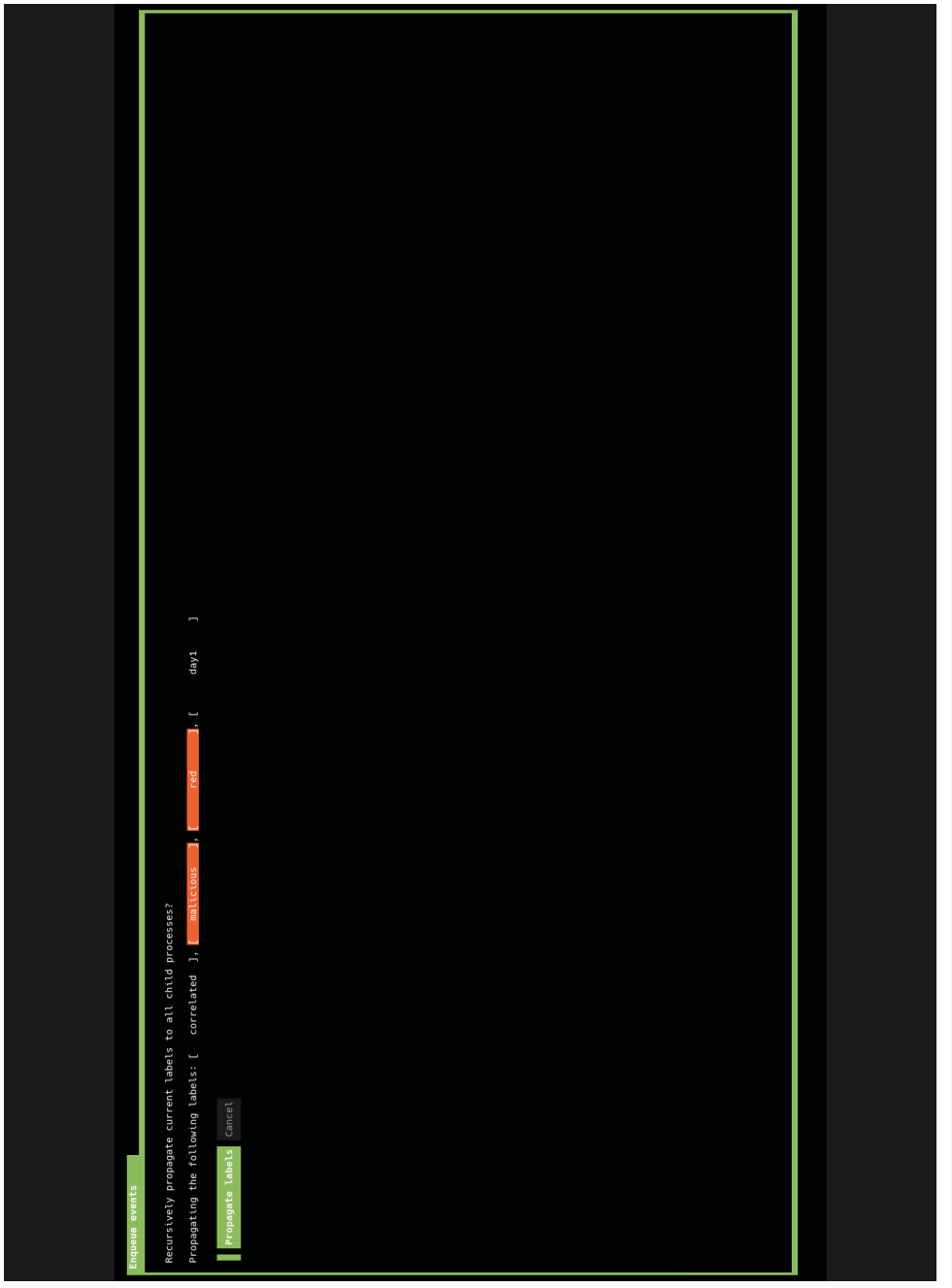

Figure-A II-4 Comme discuté dans le Section 3.6.2, les euristiques d'étiquetage considèrent tous les processus enfants d'un processus malveillant comme également malveillants. Cet écran de confirmation permet à l'analyste de sécurité de propager les étiquettes actuelles de l'évènement à travers toute la forêt de processus

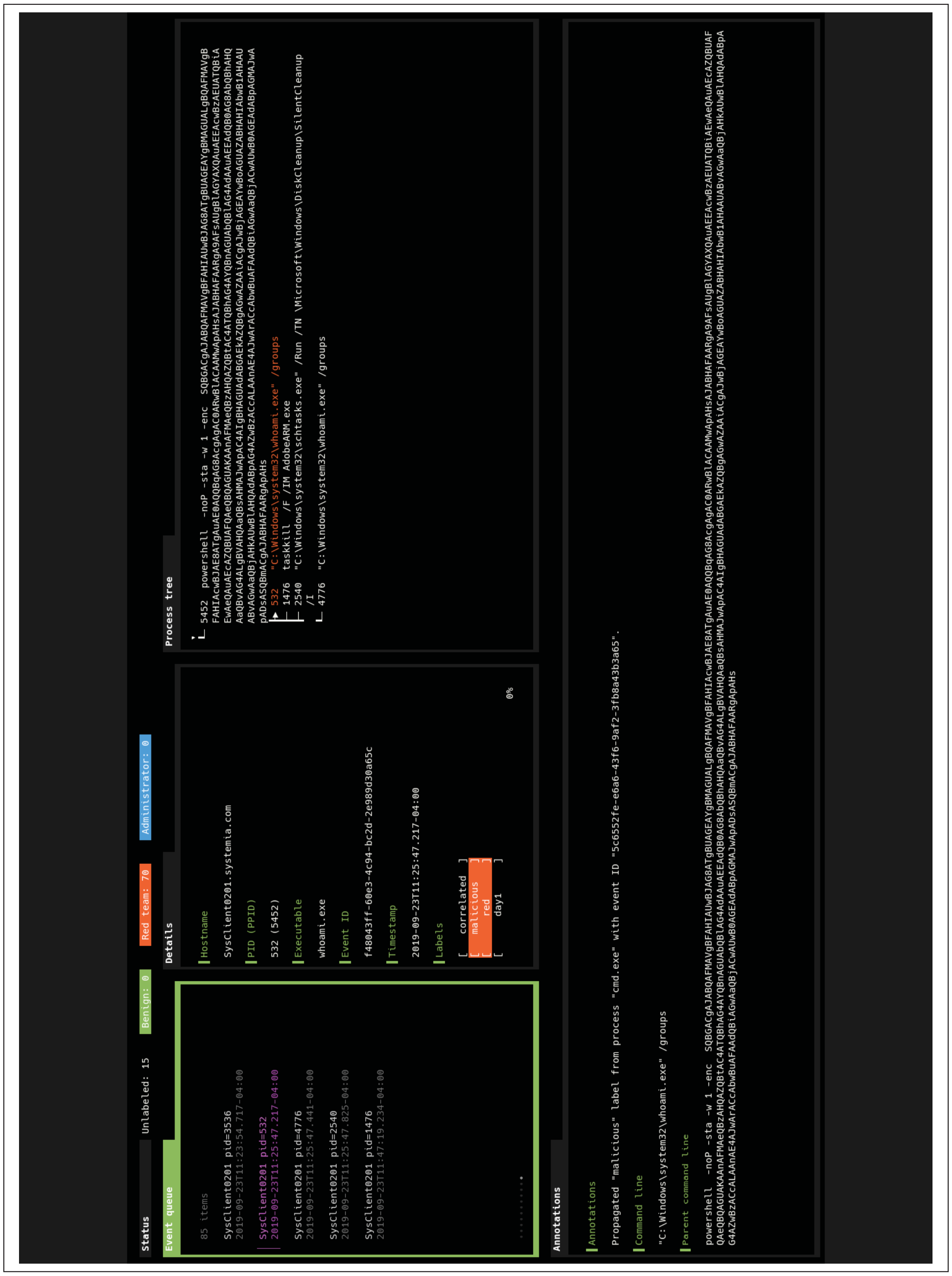

Figure-A II-5 Cet écran montre les résultats de la diffusion des étiquettes et de la corrélation automatique qui découvre des évènements non mentionnés dans le document de vérité (indiqués en rouge) et les étiquette correctement comme activité malveillante de l’équipe rouge

ANNEXE III

ANOMALIES DE LA LIGNE DE COMMANDE

Sélection des commandes les plus anomales issues du modèle de détection de la Section 4.4.4.

Toutes ces commandes incluent des indicateurs très clairs d'activité malveillante. Même sans aucun contexte additionnel, elles peuvent être facilement détectées par une variété de méthodes (y compris les règles statiques) et servir à orienter l'enquête vers des attaques plus obscures.

```
     anomaly: #1
       score: 76.6668
   timestamp: "2019-09-24T13:43:11.446-04:00"
    hostname: "SysClient0501.systemia.com"
        user: "SYSTEMIACOM\\sysadmin"
command_line: "fileTransfer1000.exe  news.com 9999 "
```

```
     anomaly: #2
       score: 75.48839
   timestamp: "2019-09-24T14:01:15.477-04:00"
    hostname: "SysClient0005.systemia.com"
        user: "SYSTEMIACOM\\sysadmin"
command_line: "find  / \"\\\\\\""
```

```
     anomaly: #3
       score: 71.5692
   timestamp: "2019-09-24T15:29:54.893-04:00"
    hostname: "SysClient0501.systemia.com"
        user: "SYSTEMIACOM\\bantonio"
command_line: "atbroker.exe"
```

```
     anomaly: #5
       score: 67.58067
   timestamp: "2019-09-24T15:02:20.313-04:00"
    hostname: "SysClient0005.systemia.com"
        user: "SYSTEMIACOM\\sysadmin"
command_line: "movingonup.exe  fun.com 81 "
```

```
     anomaly: #6
       score: 67.18797
   timestamp: "2019-09-24T14:01:15.399-04:00"
    hostname: "SysClient0005.systemia.com"
        user: "SYSTEMIACOM\\sysadmin"
command_line: "net  session "
```

```
     anomaly: #14
       score: 56.21499
   timestamp: "2019-09-25T11:06:55.783-04:00"
    hostname: "SysClient0051.systemia.com"
        user: "NT AUTHORITY\\SYSTEM"
command_line: "net  localgroup Administrators  admin /add"
```

```
     anomaly: #16
       score: 48.1384
   timestamp: "2019-09-23T13:55:54.931-04:00"
    hostname: "SysClient0660.systemia.com"
        user: "SYSTEMIACOM\\zleazer"
command_line: "\"C:\\Windows\\system32\\net.exe\" group \"Domain Controllers\"
              /domain"
```

```
     anomaly: #18
       score: 44.01708
   timestamp: "2019-09-25T13:45:45.444-04:00"
    hostname: "SysClient0351.systemia.com"
        user: "SYSTEMIACOM\\bbateman"
command_line: "ping  microsoft.com"
```

```
     anomaly: #21
       score: 43.32026
   timestamp: "2019-09-24T13:27:13.397-04:00"
    hostname: "SysClient0501.systemia.com"
        user: "SYSTEMIACOM\\sysadmin"
command_line: "\"PowerShell.exe\" -noexit -command Set-Location -literalPath
              'C:\\Users\\sysadmin\\Downloads'"
```

```
     anomaly: #25
       score: 41.70693
   timestamp: "2019-09-25T11:06:32.853-04:00"
    hostname: "SysClient0051.systemia.com"
        user: "NT AUTHORITY\\SYSTEM"
command_line: "net  user admin supersecret /add"
```

```
     anomaly: #29
       score: 40.07
   timestamp: "2019-09-24T10:35:53.666-04:00"
    hostname: "SysClient0501.systemia.com"
        user: "SYSTEMIACOM\\bantonio"
command_line: "powershell.exe  -NoP -sta -NonI -W Hidden $e=(New-Object Syste
               m.Net.WebClient).DownloadString('http://news.com:8000/default.p
               s1');powershell -noP -sta -w 1 -enc $e "
```

```
     anomaly: #39
       score: 35.64852
   timestamp: "2019-09-25T14:23:15.16-04:00"
    hostname: "SysClient0051.systemia.com"
        user: "SYSTEMIACOM\\dcoombes"
command_line: "\"C:\\Users\\dcoombes\\Downloads\\update.exe\" "
```

```
     anomaly: #48
       score: 32.52685
   timestamp: "2019-09-24T11:42:46.685-04:00"
    hostname: "SysClient0501.systemia.com"
        user: "SYSTEMIACOM\\Administrator"
command_line: "\"C:\\Windows\\system32\\NETSTAT.EXE\" -ano"
```

```
     anomaly: #58
       score: 29.19083
   timestamp: "2019-09-25T10:47:14.537-04:00"
    hostname: "SysClient0051.systemia.com"
        user: "NT AUTHORITY\\SYSTEM"
command_line: "\"C:\\Windows\\Temp\\rad0F93B.tmp\\biGuWCmNsuCIG.exe\" "
```

```
     anomaly: #61
       score: 28.14845
   timestamp: "2019-09-20T11:53:18.559-04:00"
    hostname: "SysClient0123.systemia.com"
        user: "NT AUTHORITY\\SYSTEM"
command_line: "C:\\Windows\\cSpcpdoR.exe"
```

```
     anomaly: #83
       score: 21.41325
   timestamp: "2019-09-24T13:39:58.264-04:00"
    hostname: "SysClient0501.systemia.com"
        user: "SYSTEMIACOM\\sysadmin"
command_line: "C:\\Windows\\system32\\cmd.exe  /S /D /c\" type export.zip \""
```

ANNEXE IV

LISTE DES COMMANDES DE POWERSHELL

	Commande		Commande		Commande
1	findstr.exe	58	Get-ModifiablePath	7064	Get-Acl
1	Invoke-SMBScanner	64	Get-ItemProperty	7131	Get-Item
1	Get-Help	88	Invoke-Expression	7600	Write-Host
1	more	118	Get-Content	7674	Set-ExecutionPolicy
1	Get-DomainComputer	121	PSConsoleHostReadline	7754	Start-Service
1	Get-DomainGroupMember	156	Remove-Variable	7765	Get-ChildItem
1	Find-DomainUserLocation	207	Format-Table	9336	Set-Alias
1	Get-NetLoggedon	221	Format-List	9575	Get-WURebootStatus
1	Get-WebConfig	322	ConvertFrom-StringData	10986	Get-Service
1	Get-ApplicationHost	469	Sort-Object	13634	Set-StrictMode
1	Get-SiteListPassword	510	PING.EXE	14870	New-Module
1	Get-CachedGPPPassword	1332	Test-Connection	15304	Write-Verbose
1	fileTransfer1000.exe	1456	Resolve-Path	15889	Out-Null
1	plink.exe	1559	Compare-Object	17213	Get-Date
2	Get-DomainController	1870	startup.ps1	19806	Get-Variable
3	Invoke-ARPScan	1908	Get-Location	19883	Add-Member
4	Remove-Item	1917	update_wrapper.ps1	24064	Export-ModuleMember
9	New-Item	1919	updater.ps1	24698	Import-Module
12	Set-WmiInstance	3513	Out-String	27612	New-Alias
13	Stop-Service	3800	New-Service	30201	Out-Default
13	Add-Type	3818	Start-Process	47675	Unblock-File
14	Get-Process	3824	Set-Location	63882	Start-Sleep
16	ConvertTo-SecureString	3998	Split-Path	64805	Write-Output
17	New-ItemProperty	4970	Get-Command	89674	Where-Object
19	Test-MemoryRangeValid	5096	Select-String	114700	Out-File
20	Remove-ItemProperty	5405	Test-Path	126214	Write-Debug
21	Stop-AgentJob	5516	Select-Object	150885	New-Object
27	Get-CimInstance	5700	ConvertTo-Json	154899	Get-WUInstall
27	Join-Path	6116	Get-WmiObject	166628	Get-Random
30	Set-ItemProperty	6320	Get-Alias	6373514	ForEach-Object
47	Receive-AgentJob	6650	Copy-Item		
55	Invoke-WmiMethod	6955	Set-Variable		

LISTE DE RÉFÉRENCES

Afnan, S., Sadia, M., Iqbal, S. et Iqbal, A. [arXiv :2311.05733 [cs]]. (2023). LogShield : A Transformer-based APT Detection System Leveraging Self-Attention. arXiv. Repéré le 2024-03-17 à http://arxiv.org/abs/2311.05733.

Ahmad, Z., Shahid Khan, A., Wai Shiang, C., Abdullah, J. et Ahmad, F. (2021). Network intrusion detection system : A systematic study of machine learning and deep learning approaches. *Transactions on Emerging Telecommunications Technologies*, 32(1), e4150. doi : 10.1002/ett.4150.

Ahmed, M., Wei, J. et Al-Shaer, E. (2023). SCAHunter : Scalable Threat Hunting Through Decentralized Hierarchical Monitoring Agent Architecture. Dans Arai, K. (Éd.), *Intelligent Computing* (vol. 739, pp. 1282–1307). Cham : Springer Nature Switzerland. doi : 10.1007/978-3-031-37963-5_88.

Alahmadi, B. A., Axon, L. et Martinovic, I. (2022). 99% False Positives : A Qualitative Study of SOC Analysts' Perspectives on Security Alarms. *31st USENIX Security Symposium (USENIX Security 22)*, pp. 2783–2800. Repéré à https://www.usenix.org/conference/usenixsecurity22/presentation/alahmadi.

Aly, A., Iqbal, S., Youssef, A. et Mansour, E. (2024). MEGR-APT : A Memory-Efficient APT Hunting System Based on Attack Representation Learning. *IEEE Transactions on Information Forensics and Security*, 19, 5257–5271. doi : 10.1109/TIFS.2024.3396390.

Anderson, J. P. (1980). *Computer Securtity Threat Monitoring and Surveillance*. Fort Washington, PA 19034.

Andresini, G., Pendlebury, F., Pierazzi, F., Loglisci, C., Appice, A. et Cavallaro, L. (2021). INSOMNIA : Towards Concept-Drift Robustness in Network Intrusion Detection. *Proceedings of the 14th ACM Workshop on Artificial Intelligence and Security*, pp. 111–122. doi : 10.1145/3474369.3486864.

Anjum, M. M., Iqbal, S. et Hamelin, B. (2021). Analyzing the Usefulness of the DARPA OpTC Dataset in Cyber Threat Detection Research. *Proceedings of the 26th ACM Symposium on Access Control Models and Technologies*, pp. 27–32. doi : 10.1145/3450569.3463573.

Anjum, M. M., Iqbal, S. et Hamelin, B. (2022). ANUBIS : a provenance graph-based framework for advanced persistent threat detection. *Proceedings of the 37th ACM/SIGAPP Symposium on Applied Computing*, pp. 1684–1693. doi : 10.1145/3477314.3507097.

Apruzzese, G., Anderson, H. S., Dambra, S., Freeman, D., Pierazzi, F. et Roundy, K. (2023a). "Real Attackers Don't Compute Gradients" : Bridging the Gap Between Adversarial ML Research and Practice. *2023 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*, pp. 339–364. doi : 10.1109/SaTML54575.2023.00031.

Apruzzese, G., Laskov, P. et Schneider, J. (2023b). SoK : Pragmatic Assessment of Machine Learning for Network Intrusion Detection. *2023 IEEE 8th European Symposium on Security and Privacy (EuroS&P)*, pp. 592–614. doi : 10.1109/EuroSP57164.2023.00042.

Apruzzese, G., Fass, A. et Pierazzi, F. (2024). When Adversarial Perturbations meet Concept Drift : An Exploratory Analysis on ML-NIDS. *Proceedings of the 2024 Workshop on Artificial Intelligence and Security*, pp. 149–160. doi : 10.1145/3689932.3694757.

Arp, D., Quiring, E., Pendlebury, F., Warnecke, A., Pierazzi, F., Wressnegger, C., Cavallaro, L. et Rieck, K. (2022). Dos and Don'ts of Machine Learning in Computer Security. *31st USENIX Security Symposium (USENIX Security 22)*, pp. 3971–3988. Repéré à https://www.usenix.org/conference/usenixsecurity22/presentation/arp.

AWS. (2024). GuardDuty finding format. Repéré le 2024-02-05 à https://docs.aws.amazon.com/guardduty/latest/ug/guardduty_finding-format.html.

Axelsson, S. (1999). The base-rate fallacy and its implications for the difficulty of intrusion detection. *Proceedings of the 6th ACM conference on Computer and communications security*, pp. 1–7. doi : 10.1145/319709.319710.

Baird, A., Huang, G., Munns, C. et Weinstein, O. (2017). Serverless Architectures with AWS Lambda. Overview and Best Practices. AWS. Repéré à https://d1.awsstatic.com/whitepapers/serverless-architectures-with-aws-lambda.pdf.

Barlette, Y., Jaouen, A. et Baillette, P. (2021). Bring Your Own Device (BYOD) as reversed IT adoption : Insights into managers' coping strategies. *International Journal of Information Management*, 56, 102212. doi : 10.1016/j.ijinfomgt.2020.102212.

Bates, A., Tian, D. J., Butler, K. R. B. et Moyer, T. (2015). Trustworthy Whole-System Provenance for the Linux Kernel. *24th USENIX Security Symposium (USENIX Security 15)*, pp. 319–334. Repéré à https://www.usenix.org/conference/usenixsecurity15/technical-sessions/presentation/bates.

Bhattarai, B. et Huang, H. (2022). SteinerLog : Prize Collecting the Audit Logs for Threat Hunting on Enterprise Network. *Proceedings of the 2022 ACM on Asia Conference on Computer and Communications Security*, pp. 97–108. doi : 10.1145/3488932.3523261.

Bian, H., Bai, T., Salahuddin, M. A., Limam, N., Daya, A. A. et Boutaba, R. (2019). Host in Danger ? Detecting Network Intrusions from Authentication Logs. *2019 15th International Conference on Network and Service Management (CNSM)*, pp. 1–9. doi : 10.23919/CNSM46954.2019.9012700.

Bianco, D. (2013). The Pyramid of Pain. Repéré le 2024-02-20 à http://detect-respond.blogspot.com/2013/03/the-pyramid-of-pain.html.

Boughorbel, S., Jarray, F. et El-Anbari, M. (2017). Optimal classifier for imbalanced data using Matthews Correlation Coefficient metric. *PLOS ONE*, 12(6), e0177678. doi : 10.1371/journal.pone.0177678.

Bowman, B. (2022). *Graph Techniques for Next Generation Cybersecurity*. (PhD Thesis, The George Washington University). Repéré à ISBN : 9798496538664.

Braun, T., Pekaric, I. et Apruzzese, G. (2024). Understanding the Process of Data Labeling in Cybersecurity. *Proc. ACM Symposium on Applied Computing (ACM SAC)*.

Bray, T. [Issue : 8259 Num Pages : 16 Series : Request for Comments Published : RFC 8259]. (2017). The JavaScript Object Notation (JSON) Data Interchange Format. RFC Editor. Repéré à https://www.rfc-editor.org/info/rfc8259.

Canadian Centre for Cyber Security. (2023a). Best practices for setting up a security operations centre (SOC) - ITSAP.00.500. Repéré à https://www.cyber.gc.ca/en/guidance/best-practices-setting-security-operations-centre-soc-itsap00500.

Canadian Centre for Cyber Security. (2023b). Using information technology asset management (ITAM) to enhance cyber security. Repéré le 2024-02-02 à https://www.cyber.gc.ca/en/guidance/using-information-technology-asset-management-itam-enhance-cyber-security-itsm10004.

Canadian Centre for Cyber Security. (2024). Cyber Activity Impacting CISCO ASA VPNs. Cyber Security Advisory. Repéré le 2024-04-24 à https://www.cyber.gc.ca/en/news-events/cyber-activity-impacting-cisco-asa-vpns.

Canadian Institute for Cybersecurity. (2018). Realistic Cyber Defense Dataset (CSE-CIC-IDS2018). https ://registry.opendata.aws/cse-cic-ids2018.

Cantone, M., Marrocco, C. et Bria, A. (2024). On the Cross-Dataset Generalization of Machine Learning for Network Intrusion Detection. doi : 10.48550/ARXIV.2402.10974. Publisher : [object Object] Version Number : 1.

Cao, C., Blaise, A., Panichella, A. et Verwer, S. [Version Number : 1]. (2024). State Frequency Estimation for Anomaly Detection. arXiv. Repéré le 2025-01-25 à https://arxiv.org/abs/2412.03442.

Carey, M. J. et Jin, J. (2020). *Tribe of hackers blue team : tribal knowledge from the best in defensive cybersecurity*. Indianapolis : Wiley.

Catillo, M., Pecchia, A., Repola, A. et Villano, U. (2024). Towards realistic problem-space adversarial attacks against machine learning in network intrusion detection. *Proceedings of the 19th International Conference on Availability, Reliability and Security*, pp. 1–8. doi : 10.1145/3664476.3669974.

Censys. (2024). The 2024 State of Threat Hunting Report. Repéré le 2024-03-22 à https://censys.com/2024-state-of-threat-hunting-report/.

Cheng, Z., Lv, Q., Liang, J., Wang, Y., Sun, D., Pasquier, T. et Han, X. (2024). KAIROS : Practical Intrusion Detection and Investigation using Whole-system Provenance. *2024 IEEE Symposium on Security and Privacy (SP)*, pp. 9–9. doi : 10.1109/SP54263.2024.00005.

Chiang, W.-L., Liu, X., Si, S., Li, Y., Bengio, S. et Hsieh, C.-J. (2019). Cluster-GCN : An Efficient Algorithm for Training Deep and Large Graph Convolutional Networks. *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 257–266. doi : 10.1145/3292500.3330925.

Chuvakin, A., Sachdev, U., Lauritzen, D., Rudoll, M. et Glowacki, A. (2024). Future of the SOC : Evolution or Optimization - Choose Your Path. Deloitte Development LLC.

CIS. (2021). CIS Critical Security Controls Version 8. Repéré le 2024-01-07 à https://www.cisecurity.org/controls/v8.

CISA. (2023). Enhanced Monitoring to Detect APT Activity Targeting Outlook Online. Repéré le 2024-03-28 à https://www.cisa.gov/news-events/cybersecurity-advisories/aa23-193a.

CISA. (2024). ED 24-02 : Mitigating the Significant Risk from Nation-State Compromise of Microsoft Corporate Email System. Repéré le 2024-04-02 à https://www.cisa.gov/news-events/directives/ed-24-02-mitigating-significant-risk-nation-state-compromise-microsoft-corporate-email-system.

CISA, NSA, FBI, DOE, EPA, TSA, ACSC, CCCS, UK-NCSC et NZ-NCSC. (2024). Joint Guidance : Identifying and Mitigating Living Off the Land Techniques. Repéré le 2024-02-07 à https://www.cisa.gov/resources-tools/resources/identifying-and-mitigating-living-land-techniques.

Cisco. (2006). Cisco IOS Flexible NetFlow. Repéré à https://www.cisco.com/c/en/us/products/collateral/ios-nx-os-software/flexible-netflow/product_data_sheet0900aecd804b590b.html.

Cisco. (2024). Snort Open Source Intrusion Prevention System. Cisco. Repéré à https://www.snort.org/downloads.

Cochrane, T., Foster, P., Chhabra, V., Lemercier, M., Lyons, T. et Salvi, C. (2021). SK-Tree : a systematic malware detection algorithm on streaming trees via the signature kernel. *2021 IEEE International Conference on Cyber Security and Resilience (CSR)*, pp. 35–40. doi : 10.1109/CSR51186.2021.9527933.

Creech, G. et Hu, J. (2013). Generation of a new IDS test dataset : Time to retire the KDD collection. *2013 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 4487–4492. doi : 10.1109/WCNC.2013.6555301.

CrowdStrike. (2022). What is a Security Operations Center (SOC) ? Repéré à https://www.crowdstrike.com/cybersecurity-101/security-operations-center-soc/.

CrowdStrike. (2023). Nowhere to Hide : CrowdStrike 2023 Threat Hunting Report. Repéré le 2024-01-15 à https://go.crowdstrike.com/2023-threat-hunting-report.html.

CrowdStrike. (2024). CrowdStrike 2024 Global Threat Report. Repéré le 2024-02-21 à https://www.crowdstrike.com/blog/crowdstrike-2024-global-threat-report/.

CrowdStrike. (2025). CrowdStrike 2025 Global Threat Report. Repéré le 2025-04-04 à https://www.crowdstrike.com/en-us/global-threat-report/.

Crowley, C. et Pescatore, J. (2019). Common and Best Practices for Security Operations Centers : Results of the 2019 SOC Survey. SANS. Repéré à https://www.sans.org/white-papers/39060/.

Crowley, C. et Pescatore, J. (2021). SANS 2021 SOC Survey. SANS. Repéré le 2024-03-15 à https://www.sans.org/white-papers/sans-2021-survey-security-operations-center-soc/.

Crowley, C., Filkins, B. et Pescatore, J. (2023). SANS 2023 SOC Survey. SANS. Repéré le 2024-01-27 à https://www.sans.org/white-papers/2023-sans-soc-survey/.

Cyber Safety Review Board. (2024). Review of the Summer 2023 Microsoft Exchange Online Intrusion. CISA. Repéré le 2024-04-02 à https://www.cisa.gov/resources-tools/resources/cyber-safety-review-board-releases-report-microsoft-online-exchange-incident-summer-2023.

DARPA. (1998). 1999 DARPA Intrusion Detection Evaluation Dataset. Repéré à https://www.ll.mit.edu/r-d/datasets/1999-darpa-intrusion-detection-evaluation-dataset.

Data Study Group Team. (2019). *Data Study Group Final Report : Imperial College London, Los Alamos National Laboratory, Heilbronn Institute*. Repéré le 2024-01-29 à https://zenodo.org/record/3558251.

Deep Instinct. (2023a). 2023 Mid-Year Threat Report. Navigating the Threat Landscape. Repéré le 2024-01-15 à https://info.deepinstinct.com/2023-mid-year-cyber-threat-landscape-report.

Deep Instinct. (2023b). Voice of SecOps 2023. Repéré le 2024-03-22 à https://info.deepinstinct.com/voice-of-secops-v4-2023.

Demirjian, K. (2023). Chinese Hackers Stole 60,000 State Dept. Emails in Breach Reported in July. The New York Times. Repéré le 2024-03-28 à https://www.nytimes.com/2023/09/27/us/politics/chinese-hackers-state-department.html.

Denning, D. (1987). An Intrusion-Detection Model. *IEEE Transactions on Software Engineering*, SE-13(2), 222–232. doi : 10.1109/TSE.1987.232894.

Dong, F., Li, S., Jiang, P., Li, D., Wang, H., Huang, L., Xiao, X., Chen, J., Luo, X., Guo, Y. et Chen, X. (2023). Are we there yet ? An Industrial Viewpoint on Provenance-based Endpoint Detection and Response Tools. *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, pp. 2396–2410. doi : 10.1145/3576915.3616580.

Du, M., Li, F., Zheng, G. et Srikumar, V. (2017). DeepLog : Anomaly Detection and Diagnosis from System Logs through Deep Learning. *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1285–1298. doi : 10.1145/3133956.3134015.

Eddy, W. (2022). *Transmission Control Protocol (TCP)* (Rapport n°RFC9293). Repéré le 2025-02-03 à https://www.rfc-editor.org/info/rfc9293.

Emm, D. (2008). Changing threats, changing solutions : A history of viruses and antivirus. Repéré à https://securelist.com/changing-threats-changing-solutions-a-history-of-viruses-and-antivirus/36202/.

EmpireProject. (2018). Empire. Repéré le 2024-01-17 à https://github.com/EmpireProject/Empire.

Engelen, G., Rimmer, V. et Joosen, W. (2021). Troubleshooting an Intrusion Detection Dataset : the CICIDS2017 Case Study. *2021 IEEE Security and Privacy Workshops (SPW)*, pp. 7–12. doi : 10.1109/SPW53761.2021.00009.

Fang, R., Bindu, R., Gupta, A., Zhan, Q. et Kang, D. [_eprint : 2402.06664]. (2024). LLM Agents can Autonomously Hack Websites.

FBI et CISA. (2023). Joint Cybersecurity Advisory : Scattered Spider. CISA. Repéré le 2024-02-15 à https://www.cisa.gov/news-events/cybersecurity-advisories/aa23-320a.

Federal Bureau of Investigation. (2019). Morris Worm. Repéré à https://www.fbi.gov/history/famous-cases/morris-worm.

FireEye. (2020). Highly Evasive Attacker Leverages SolarWinds Supply Chain to Compromise Multiple Global Victims With SUNBURST Backdoor. Repéré le 2024-01-22 à https://www.mandiant.com/resources/blog/evasive-attacker-leverages-solarwinds-supply-chain-compromises-with-sunburst-backdoor.

FiveDirections. (2019). Operationally Transparent Cyber (OpTC) Data Release. Repéré le 2024-01-15 à https://github.com/FiveDirections/OpTC-data.

Forrest, S., Hofmeyr, S., Somayaji, A. et Longstaff, T. (1996). A sense of self for Unix processes. *Proceedings 1996 IEEE Symposium on Security and Privacy*, pp. 120–128. doi : 10.1109/SECPRI.1996.502675.

Freitas, S. et Gharib, A. (2024). GraphWeaver : Billion-Scale Cybersecurity Incident Correlation. *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pp. 4479–4486. doi : 10.1145/3627673.3680057.

Gates, C. et Taylor, C. (2006). Challenging the anomaly detection paradigm : a provocative discussion. *Proceedings of the 2006 workshop on New security paradigms*, pp. 21–29. doi : 10.1145/1278940.1278945.

Gaudin, G., Debar, H., Fillette, A., deMeer, J., Rennoch, A., Saadé, P. et Saugeot, J. (2018). Information Security Indicators (ISI) ; Guidelines for building and operating a secured Security Operations Center (SOC). ETSI.

Gerhards, R. (2009). *The Syslog Protocol* (Rapport n°RFC5424). Repéré le 2025-02-15 à https://www.rfc-editor.org/info/rfc5424.

Golczynski, A. et Emanuello, J. A. [arXiv :2108.12276 [cs]]. (2021). End-To-End Anomaly Detection for Identifying Malicious Cyber Behavior through NLP-Based Log Embeddings. arXiv. Repéré le 2024-02-04 à http://arxiv.org/abs/2108.12276.

Google Threat Analysis Group et Mandiant. (2024). We're All in this Together. A Year in Review of Zero-Days Exploited In-the-Wild in 2023. Google. Repéré le 2024-03-27 à https://blog.google/technology/safety-security/a-review-of-zero-day-in-the-wild-exploits-in-2023/.

Goyal, A., Han, X., Wang, G. et Bates, A. (2023). Sometimes, You Aren't What You Do : Mimicry Attacks against Provenance Graph Host Intrusion Detection Systems. *Proceedings 2023 Network and Distributed System Security Symposium*. doi : 10.14722/ndss.2023.24207.

Goyal, A., Wang, G. et Bates, A. (2024). R-CAID : Embedding Root Cause Analysis within Provenance-based Intrusion Detection. *2024 IEEE Symposium on Security and Privacy (SP)*, pp. 3515–3532. doi : 10.1109/SP54263.2024.00253.

Griffith, J., Kong, D., Caro, A., Benyo, B., Khoury, J., Upthegrove, T., Christovich, T., Ponomorov, S., Sydney, A., Saini, A., Shurbanov, V., Willig, C., Levin, D. et Dietz, J. (2020). Scalable Transparency Architecture For Research Collaboration (STARC) – DARPA Transparent Computing (TC) Program. Air Force Research Laboratory, United States Air Force.

Hagemann, T. et Katsarou, K. (2020). A Systematic Review on Anomaly Detection for Cloud Computing Environments. *2020 3rd Artificial Intelligence and Cloud Computing Conference*, pp. 83–96. doi : 10.1145/3442536.3442550.

Hamilton, W. L. Graph Representation Learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 14(3), 1–159. Publisher : Morgan and Claypool.

Han, X., Pasquier, T., Bates, A., Mickens, J. et Seltzer, M. (2020). Unicorn : Runtime Provenance-Based Detector for Advanced Persistent Threats. *Proceedings 2020 Network and Distributed System Security Symposium*. doi : 10.14722/ndss.2020.24046.

Handley, M., Paxson, V. et Kreibich, C. (2001). Network intrusion detection : evasion, traffic normalization, and end-to-end protocol semantics. *Proceedings of the 10th Conference on USENIX Security Symposium - Volume 10*, (SSYM'01), 9.

Hassan, W. U., Guo, S., Li, D., Chen, Z., Jee, K., Li, Z. et Bates, A. (2019). NoDoze : Combatting Threat Alert Fatigue with Automated Provenance Triage. *Proceedings 2019 Network and Distributed System Security Symposium*. doi : 10.14722/ndss.2019.23349.

Hendler, D., Kels, S. et Rubin, A. (2020). AMSI-Based Detection of Malicious PowerShell Code Using Contextual Embeddings. *Proceedings of the 15th ACM Asia Conference on Computer and Communications Security*, pp. 679–693. doi : 10.1145/3320269.3384742.

Herron, C. et Quan, T. (2022). Cybersecurity Talent Development : Protecting Canada's Digital Economy. Ottawa, ON : Information and Communications Technology Council. Repéré le 2024-01-20 à https://www.digitalthinktankictc.com/reports/cybersecurity-talent-development.

Hesford, J., Cheng, D., Wan, A., Huynh, L., Kim, S., Kim, H. et Hong, J. B. [_eprint : 2403.17458]. (2024). Expectations Versus Reality : Evaluating Intrusion Detection Systems in Practice.

Highnam, K., Arulkumaran, K., Hanif, Z. et Jennings, N. R. (2021). BETH Dataset : Real Cybersecurity Data for Anomaly Detection Research. *ICML Workshop on Uncertainty and Robustness in Deep Learning*.

Ho, G., Dhiman, M., Akhawe, D., Paxson, V., Savage, S., Voelker, G. M. et Wagner, D. (2021). Hopper : Modeling and Detecting Lateral Movement. *30th USENIX Security Symposium (USENIX Security 21)*, pp. 3093–3110. Repéré à https://www.usenix.org/conference/usenixsecurity21/presentation/ho.

Hofmeyr, S. A., Forrest, S. et Somayaji, A. (1998). Intrusion detection using sequences of system calls. *J. Comput. Secur.*, 6(3), 151–180. Place : NLD Publisher : IOS Press.

Hu, P., Li, H., Fu, H., Cansever, D. et Mohapatra, P. (2015). Dynamic defense strategy against advanced persistent threat with insiders. *2015 IEEE Conference on Computer Communications (INFOCOM)*, pp. 747–755. doi : 10.1109/INFOCOM.2015.7218444.

Huntress. (2023). Small and Medium-Sized Business Threat Report. Repéré le 2024-02-10 à https://www.huntress.com/resources/smb-threat-report.

Huntress. (2025). Huntress 2025 Cyber Threat Report. Repéré le 2025-02-25 à https://www.huntress.com/resources/2025-cyber-threat-report.

Hutchins, E. M., Cloppert, M. J. et Amin, R. M. (2011). Intelligence-Driven Computer Network Defense Informed by Analysis of Adversary Campaigns and Intrusion Kill Chains. Lockheed Martin.

Inam, M. A., Chen, Y., Goyal, A., Liu, J., Mink, J., Michael, N., Gaur, S., Bates, A. et Hassan, W. U. (2023). SoK : History is a Vast Early Warning System : Auditing the Provenance of System Intrusions. *2023 IEEE Symposium on Security and Privacy (SP)*, pp. 2620–2638. doi : 10.1109/SP46215.2023.10179405.

Jansen, M., Bobba, R. et Nevin, D. (2024). A Comparative Analysis of Difficulty Between Log and Graph-Based Detection Rule Creation. *Workshop on SOC Operations and Construction (WOSOC) 2024*. doi : 10.14722/wosoc.2024.23004.

Jia, Z., Xiong, Y., Nan, Y., Zhang, Y., Zhao, J. et Wen, M. (2024). MAGIC : Detecting Advanced Persistent Threats via Masked Graph Representation Learning. *33rd USENIX Security Symposium (USENIX Security 24)*, pp. 5197–5214. Repéré à https://www.usenix.org/conference/usenixsecurity24/presentation/jia-zian.

Jiang, Z., Liu, J., Huang, J., Li, Y., Huo, Y., Gu, J., Chen, Z., Zhu, J. et Lyu, M. R. (2024). A Large-Scale Evaluation for Log Parsing Techniques : How Far Are We ? *Proceedings of the 33rd ACM SIGSOFT International Symposium on Software Testing and Analysis*, pp. 223–234. doi : 10.1145/3650212.3652123.

Joint Task Force Interagency Working Group. (2020). *Security and Privacy Controls for Information Systems and Organizations*. Repéré le 2024-08-31 à https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-53r5.pdf.

Junge, M. R. J. et Dettori, J. R. (2018). ROC Solid : Receiver Operator Characteristic (ROC) Curves as a Foundation for Better Diagnostic Tests. *Global Spine Journal*, 8(4), 424–429. doi : 10.1177/2192568218778294.

Kaiafas, G., Varisteas, G., Lagraa, S., State, R., Nguyen, C. D., Ries, T. et Ourdane, M. (2018). Detecting malicious authentication events trustfully. *NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium*, pp. 1–6. doi : 10.1109/NOMS.2018.8406295.

Kent, A. D. [Published : Los Alamos National Laboratory]. (2015a). Comprehensive, Multi-Source Cyber-Security Events.

Kent, A. D. (2015b). Cybersecurity Data Sources for Dynamic Network Research. *Dynamic Networks in Cybersecurity*.

Kent, K. et Souppaya, M. P. (2006). *Guide to computer security log management* (Rapport n°NIST SP 800-92). Gaithersburg, MD. Repéré le 2025-02-15 à https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-92.pdf.

Kenyon, A., Deka, L. et Elizondo, D. (2020). Are public intrusion datasets fit for purpose characterising the state of the art in intrusion event datasets. *Computers & Security*, 99, 102022. doi : 10.1016/j.cose.2020.102022.

King, I. J. et Huang, H. H. (2023). Euler : Detecting Network Lateral Movement via Scalable Temporal Link Prediction. *ACM Transactions on Privacy and Security*, 26(3), 1–36. doi : 10.1145/3588771.

King, S. T. et Chen, P. M. (2003). Backtracking intrusions. *Proceedings of the nineteenth ACM symposium on Operating systems principles*, pp. 223–236. doi : 10.1145/945445.945467.

King, S. T., Mao, Z. M., Lucchetti, D. G. et Chen, P. M. (2005). Enriching Intrusion Alerts Through Multi-Host Causality. *Network and Distributed System Security Symposium*. Repéré à https://www.ndss-symposium.org/ndss2005/enriching-intrusion-alerts-through-multi-host-causality/.

Knerler, K., Parker, I. et Zimmerman, C. (2022). *11 Strategies of a World-Class Cybersecurity Operations Center* (éd. 2nd edition). Mitre P.

Kokulu, F. B., Soneji, A., Bao, T., Shoshitaishvili, Y., Zhao, Z., Doupé, A. et Ahn, G.-J. (2019). Matched and Mismatched SOCs : A Qualitative Study on Security Operations Center Issues. *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1955–1970. doi : 10.1145/3319535.3354239.

Kubernetes. (2023). Auditing. Repéré le 2024-02-05 à https://kubernetes.io/docs/tasks/debug/debug-cluster/audit/.

Landauer, M., Skopik, F., Wurzenberger, M. et Rauber, A. (2022). Dealing with Security Alert Flooding : Using Machine Learning for Domain-independent Alert Aggregation. *ACM Transactions on Privacy and Security*, 25(3), 1–36. doi : 10.1145/3510581.

Lange, K. (2023). Detection Engineering Explained. Repéré le 2024-04-19 à https://www.splunk.com/en_us/blog/learn/detection-engineering.html.

Lanvin, M., Gimenez, P.-F., Han, Y., Majorczyk, F., Me, L. et Totel, E. (2023). Errors in the CICIDS2017 Dataset and the Significant Differences in Detection Performances It Makes. Dans Kallel, S., Jmaiel, M., Zulkernine, M., Hadj Kacem, A., Cuppens, F. et Cuppens, N. (Éds.), *Risks and Security of Internet and Systems* (vol. 13857, pp. 18–33). Cham : Springer Nature Switzerland. doi : 10.1007/978-3-031-31108-6_2.

Lee, K. H., Zhang, X. et Xu, D. (2013a). High Accuracy Attack Provenance via Binary-based Execution Partition. *Network and Distributed System Security Symposium*. Repéré à https://api.semanticscholar.org/CorpusID:93090.

Lee, K. H., Zhang, X. et Xu, D. (2013b). LogGC : garbage collecting audit log. *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security - CCS '13*, pp. 1005–1016. doi : 10.1145/2508859.2516731.

Leevy, J. L. et Khoshgoftaar, T. M. (2020). A survey and analysis of intrusion detection models based on CSE-CIC-IDS2018 Big Data. *Journal of Big Data*, 7(1), 104. doi : 10.1186/s40537-020-00382-x.

Levshun, D. et Kotenko, I. (2023). A survey on artificial intelligence techniques for security event correlation : models, challenges, and opportunities. *Artificial Intelligence Review*, 56(8), 8547–8590. doi : 10.1007/s10462-022-10381-4.

Linux project. (2024). syscalls(2) — Linux manual page. Repéré à https://www.man7.org/linux/man-pages/man2/syscalls.2.html.

Liu, F. T., Ting, K. M. et Zhou, Z.-H. (2008). Isolation Forest. *2008 Eighth IEEE International Conference on Data Mining*, pp. 413–422. doi : 10.1109/ICDM.2008.17.

Liu, L., Engelen, G., Lynar, T., Essam, D. et Joosen, W. (2022a). Error Prevalence in NIDS datasets : A Case Study on CIC-IDS-2017 and CSE-CIC-IDS-2018. *2022 IEEE Conference on Communications and Network Security (CNS)*, pp. 254–262. doi : 10.1109/CNS56114.2022.9947235.

Liu, Y., Zhang, M., Li, D., Jee, K., Li, Z., Wu, Z., Rhee, J. et Mittal, P. (2018). Towards a Timely Causality Analysis for Enterprise Security. *Proceedings 2018 Network and Distributed System Security Symposium*. doi : 10.14722/ndss.2018.23254.

Liu, Y., Shu, X., Sun, Y., Jang, J. et Mittal, P. (2022b). RAPID : Real-Time Alert Investigation with Context-aware Prioritization for Efficient Threat Discovery. *Proceedings of the 38th Annual Computer Security Applications Conference*, pp. 827–840. doi : 10.1145/3564625.3567997.

Ma, S., Zhang, X. et Xu, D. (2016). ProTracer : Towards Practical Provenance Tracing by Alternating Between Logging and Tainting. *Proceedings 2016 Network and Distributed System Security Symposium*. doi : 10.14722/ndss.2016.23350.

Ma, S., Zhai, J., Kwon, Y., Lee, K. H., Zhang, X., Ciocarlie, G., Gehani, A., Yegneswaran, V., Xu, D. et Jha, S. (2018). Kernel-Supported Cost-Effective Audit Logging for Causality Tracking. *2018 USENIX Annual Technical Conference (USENIX ATC 18)*, pp. 241–254. Repéré à https://www.usenix.org/conference/atc18/presentation/ma-shiqing.

Macas, M., Wu, C. et Fuertes, W. (2022). A survey on deep learning for cybersecurity : Progress, challenges, and opportunities. *Computer Networks*, 212, 109032. doi : 10.1016/j.comnet.2022.109032.

Mamun, M. et Buffett, S. (2022). TapTree : Process-Tree Based Host Behavior Modeling and Threat Detection Framework via Sequential Pattern Mining. Dans Alcaraz, C., Chen, L., Li, S. et Samarati, P. (Éds.), *Information and Communications Security* (vol. 13407, pp. 546–565). Cham : Springer International Publishing. doi : 10.1007/978-3-031-15777-6_30.

Mamun, M. et Shi, K. (2021). DeepTaskAPT : Insider APT detection using Task-tree based Deep Learning. *2021 IEEE 20th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, pp. 693–700. doi : 10.1109/TrustCom53373.2021.00102.

Mandiant. (2023). M-Trends 2023. Mandiant Special Report. Google. Repéré le 2024-01-29 à https://inthecloud.withgoogle.com/mandiant-m-trends-2023/download.html.

Mandiant et Google Cloud Security. (2024). M-Trends 2024 Special Report. Repéré à https://cloud.google.com/security/resources/m-trends.

Maseno, E. M., Wang, Z. et Xing, H. (2022). A Systematic Review on Hybrid Intrusion Detection System. *Security and Communication Networks*, 2022, 1–23. doi : 10.1155/2022/9663052.

Maxam, W. P. et Davis, J. C. (2024). An Interview Study on Third-Party Cyber Threat Hunting Processes in the U.S. Department of Homeland Security. doi : 10.48550/ARXIV.2402.12252. Publisher : arXiv Version Number : 1.

McHugh, J. (2000). Testing Intrusion detection systems : a critique of the 1998 and 1999 DARPA intrusion detection system evaluations as performed by Lincoln Laboratory. *ACM Transactions on Information and System Security*, 3(4), 262–294. doi : 10.1145/382912.382923.

McRee, G. R. (2022). Improved Detection and Response via Optimized Alerts : Usability Study. *Journal of Cybersecurity and Privacy*, 2(2), 379–401. doi : 10.3390/jcp2020020.

Medad, A., Gregorutti, B., Genetay, E. et Nguema, A. P. (2021). Real-time graph clustering for network intrusion detection. *Actes de la conference CAID 2021 (Conference on Artificial Intelligence for Defense)*, pp. 33–40. Repéré à https://hal.science/hal-03535661.

Microsoft. (2006). PowerShell. Microsoft.

Microsoft. (2018). Service State Transitions. Repéré le 2025-02-03 à https://learn.microsoft.com/en-us/windows/win32/services/service-status-transitions.

Microsoft. (2023). Overview of Microsoft Graph. Repéré le 2024-02-05 à https://learn.microsoft.com/en-us/graph/overview.

Microsoft Security Response Center. (2023). Results of Major Technical Investigations for Storm-0558 Key Acquisition. Repéré le 2024-03-28 à https://msrc.microsoft.com/blog/2023/09/results-of-major-technical-investigations-for-storm-0558-key-acquisition/.

Microsoft Threat Intelligence. (2023). Analysis of Storm-0558 techniques for unauthorized email access. Repéré le 2024-03-28 à https://www.microsoft.com/en-us/security/blog/2023/07/14/analysis-of-storm-0558-techniques-for-unauthorized-email-access/.

Milajerdi, S. M., Gjomemo, R., Eshete, B., Sekar, R. et Venkatakrishnan, V. (2019). HOLMES : Real-Time APT Detection through Correlation of Suspicious Information Flows. *2019 IEEE Symposium on Security and Privacy (SP)*, pp. 1137–1152. doi : 10.1109/SP.2019.00026.

Mink, J., Benkraouda, H., Yang, L., Ciptadi, A., Ahmadzadeh, A., Votipka, D. et Wang, G. (2023). Everybody's Got ML, Tell Me What Else You Have : Practitioners' Perception of ML-Based Security Tools and Explanations. *2023 IEEE Symposium on Security and Privacy (SP)*, pp. 2068–2085. doi : 10.1109/SP46215.2023.10179321.

MITRE. (2022). MITRE Cyber Analytics Repository. The MITRE Corporation. Repéré le 2024-02-02 à https://car.mitre.org/data_model/.

MITRE. (2023). MITRE ATT&CK v14.1. The MITRE Corporation. Repéré le 2024-02-27 à https://attack.mitre.org/versions/v14/.

MITRE. (2024). Summiting the Pyramid v2.0.0. Repéré à https://center-for-threat-informed-defense.github.io/summiting-the-pyramid/.

Myneni, S., Jha, K., Sabur, A., Agrawal, G., Deng, Y., Chowdhary, A. et Huang, D. (2023). Unraveled — A semi-synthetic dataset for Advanced Persistent Threats. *Computer Networks*, 227, 109688. doi : 10.1016/j.comnet.2023.109688.

Nadeem, A., Vos, D., Cao, C., Pajola, L., Dieck, S., Baumgartner, R. et Verwer, S. (2023). SoK : Explainable Machine Learning for Computer Security Applications. *2023 IEEE 8th European Symposium on Security and Privacy (EuroS&P)*, pp. 221–240. doi : 10.1109/EuroSP57164.2023.00022.

Nassif, A. B., Talib, M. A., Nasir, Q. et Dakalbab, F. M. (2021). Machine Learning for Anomaly Detection : A Systematic Review. *IEEE Access*, 9, 78658–78700. doi : 10.1109/ACCESS.2021.3083060.

Nationaal Cyber Security Centrum. (2024). Ministry of Defence of the Netherlands uncovers COATHANGER, a stealthy Chinese FortiGate RAT. Repéré le 2024-02-09 à https://www.ncsc.nl/binaries/ncsc/documenten/publicaties/2024/februari/6/mivd-aivd-advisory-coathanger-tlp-clear/TLP-CLEAR+MIVD+AIVD+Advisory+COATHANGER.pdf.

National Cyber Security Centre. (2024a). NCSC warns of widening gap between cyber threats and defence capabilities. Repéré à https://www.ncsc.gov.uk/news/ncsc-warns-widening-gap-between-cyber-threats-and-defence-capabilities.

National Cyber Security Centre. (2024b). The near-term impact of AI on the cyber threat. Repéré le 2024-01-24 à https://www.ncsc.gov.uk/report/impact-of-ai-on-cyber-threat.

NetSPI. (2021). 5 Things Every Red Team Needs to Optimize Operations. Repéré le 2024-02-16 à https://www.netspi.com/wp-content/uploads/2021/07/netspi-5-things-every-red-team-needs-to-optimize-operations.pdf.

Newman, D. A. (2014). Missing Data : Five Practical Guidelines. *Organizational Research Methods*, 17(4), 372–411. doi : 10.1177/1094428114548590.

Nguyen, M.-D., Bouaziz, A., Valdes, V., Rosa Cavalli, A., Mallouli, W. et Montes De Oca, E. (2023). A deep learning anomaly detection framework with explainability and robustness. *Proceedings of the 18th International Conference on Availability, Reliability and Security*, pp. 1–7. doi : 10.1145/3600160.3605052.

Noel, S., Harley, E., Tam, K. H., Limiero, M. et Share, M. (2016). Chapter 4 - CyGraph : Graph-Based Analytics and Visualization for Cybersecurity. Dans Gudivada, V. N., Raghavan, V. V., Govindaraju, V. et Rao, C. R. (Éds.), *Cognitive Computing : Theory and Applications* (vol. 35, pp. 117–167). Elsevier. doi : https://doi.org/10.1016/bs.host.2016.07.001.

Omlin, C. W. et Giles, C. L. (1996). Constructing deterministic finite-state automata in recurrent neural networks. *Journal of the ACM*, 43(6), 937–972. doi : 10.1145/235809.235811.

Onwubiko, C. et Ouazzane, K. (2022). SOTER : A Playbook for Cybersecurity Incident Management. *IEEE Transactions on Engineering Management*, 69(6), 3771–3791. doi : 10.1109/TEM.2020.2979832.

OpenTelemetry. (2025). *OpenTelemetry Specification 1.41.0*. Repéré le 2025-02-10 à https://opentelemetry.io/docs/specs/otel/.

Oversight and Government Reform. (2018). *The Equifax Data Breach*.

Palo Alto Networks. (2024). Security Operations Center (SOC) Roles and Responsibilities. Repéré à https://www.paloaltonetworks.com/cyberpedia/soc-roles-and-responsibilities.

Pan, B., Stakhanova, N. et Ray, S. (2023). Data Provenance in Security and Privacy. *ACM Computing Surveys*, 55(14s), 1–35. doi : 10.1145/3593294.

Parsons, D. (2023). SANS ICS/OT Cybersecurity Survey : 2023's Challenges and Tomorrow's Defenses. SANS. Repéré le 2024-01-25 à https://www.sans.org/white-papers/ics-ot-cybersecurity-survey-2023s-challenges-tomorrows-defenses/.

Pasquier, T., Han, X., Goldstein, M., Moyer, T., Eyers, D., Seltzer, M. et Bacon, J. (2017). Practical whole-system provenance capture. *Proceedings of the 2017 Symposium on Cloud Computing*, pp. 405–418. doi : 10.1145/3127479.3129249.

Paudel, R. et Huang, H. H. (2022). Pikachu : Temporal Walk Based Dynamic Graph Embedding for Network Anomaly Detection. *NOMS 2022-2022 IEEE/IFIP Network Operations and Management Symposium*, pp. 1–7. doi : 10.1109/NOMS54207.2022.9789921.

Paxson, V. (1999). Bro : a system for detecting network intruders in real-time. *Computer Networks*, 31(23-24), 2435–2463. doi : 10.1016/S1389-1286(99)00112-7.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. et Duchesnay, E. (2011). Scikit-learn : Machine Learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830.

Pendlebury, F., Pierazzi, F., Jordaney, R., Kinder, J. et Cavallaro, L. (2019). TESSERACT : Eliminating Experimental Bias in Malware Classification across Space and Time. *28th USENIX Security Symposium (USENIX Security 19)*, pp. 729–746. Repéré à https://www.usenix.org/conference/usenixsecurity19/presentation/pendlebury.

Powell, B. A. (2020). Detecting malicious logins as graph anomalies. *Journal of Information Security and Applications*, 54, 102557. doi : 10.1016/j.jisa.2020.102557.

Prince, M., Graham-Cumming, J. et Bourzikas, G. (2024). Thanksgiving 2023 security incident. Cloudflare. Repéré le 2024-03-21 à https://blog.cloudflare.com/thanksgiving-2023-security-incident.

Ptacek, T. H. et Newsham, T. N. (1998). *Insertion, Evasion and Denial of Service : Eluding Network Intrusion Detection*. Repéré à https://apps.dtic.mil/sti/pdfs/ADA391565.pdf.

Python Software Foundation. (2024). Python. Repéré à https://www.python.org/.

Red Canary. (2024). 2024 Threat Detection Report. Repéré le 2024-03-12 à https://redcanary.com/threat-detection-report/.

Rimmer, V., Nadeem, A., Verwer, S., Preuveneers, D. et Joosen, W. (2022). Open-World Network Intrusion Detection. Dans Batina, L., Bäck, T., Buhan, I. et Picek, S. (Éds.), *Security and Artificial Intelligence* (vol. 13049, pp. 254–283). Cham : Springer International Publishing. doi : 10.1007/978-3-030-98795-4_11.

Rosay, A., Cheval, E., Carlier, F. et Leroux, P. (2022). Network Intrusion Detection : A Comprehensive Analysis of CIC-IDS2017 :. *Proceedings of the 8th International Conference on Information Systems Security and Privacy*, pp. 25–36. doi : 10.5220/0010774000003120.

Salton, G. et Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5), 513–523. doi : 10.1016/0306-4573(88)90021-0.

Salus, P. H. (1994). *A quarter century of UNIX*. Reading (Mass.) Menlo Park (Calif.) Paris [etc.] : Addison-Wesley publ.

Scarfone, K. A. et Mell, P. M. (2007). *Guide to Intrusion Detection and Prevention Systems (IDPS)* (Rapport n°NIST SP 800-94). Gaithersburg, MD. Repéré le 2025-01-26 à https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-94.pdf.

Sebring, M. M., Shellhouse, E., Hanna, M. E. et Whitehurst, R. A. (1988). Expert Systems in Intrusion Detection : A Case Study. *Proceedings of the 11th National Computer Security Conference*, pp. 74–81.

Sharafaldin, I., Habibi Lashkari, A. et Ghorbani, A. A. (2018). Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization :. *Proceedings of the 4th International Conference on Information Systems Security and Privacy*, pp. 108–116. doi : 10.5220/0006639801080116.

Sharif, A. (2022). 6 Common Log File Formats. Repéré à https://www.crowdstrike.com/en-us/cybersecurity-101/next-gen-siem/log-file-formats/.

She, R., Liu, S., Wan, S., Xiong, K. et Fan, P. (2019). Importance of Small Probability Events in Big Data : Information Measures, Applications, and Challenges. *IEEE Access*, 7, 100363–100382. doi : 10.1109/ACCESS.2019.2926518.

SigmaHQ. (2024). Sigma - Generic Signature Format for SIEM Systems. Repéré le 2024-02-27 à https://github.com/SigmaHQ/sigma.

Smaha, S. (1988). Haystack : an intrusion detection system. *[Proceedings 1988] Fourth Aerospace Computer Security Applications*, pp. 37–44. doi : 10.1109/ACSAC.1988.113412.

Smiliotopoulos, C., Kambourakis, G. et Kolias, C. (2024). Detecting lateral movement : A systematic survey. *Heliyon*, 10(4), e26317. doi : 10.1016/j.heliyon.2024.e26317.

Sommer, R. et Paxson, V. (2010). Outside the Closed World : On Using Machine Learning for Network Intrusion Detection. *2010 IEEE Symposium on Security and Privacy*, pp. 305–316. doi : 10.1109/SP.2010.25.

Song, J., Kim, J., Choi, S., Kim, J. et Kim, I. (2021). Evaluations of AI-based malicious PowerShell detection with feature optimizations. *ETRI Journal*, 43(3), 549–560. doi : 10.4218/etrij.2020-0215.

Splunk. (2024). The Splunk Guide to Risk-Based Alerting (RBA). Repéré à https://www.splunk.com/en_us/form/the-essential-guide-to-risk-based-alerting.html.

Splunk Threat Research Team. (2024). Splunk Security Content. Repéré le 2024-02-27 à https://research.splunk.com/.

STANIFORD-CHEN, S. (1998). The Common Intrusion Detection Framework (CIDF). *Position paper of Information Survivability Workshop, Orlando, October 1998*. Repéré à https://cir.nii.ac.jp/crid/1570854175642311936.

Stonebraker, M. et Pavlo, A. (2024). What Goes Around Comes Around... And Around... *ACM SIGMOD Record*, 53(2), 21–37. doi : 10.1145/3685980.3685984.

Sysdig. (2023). The 5/5/5 Benchmark for Cloud Detection and Response. Repéré le 2024-04-10 à https://sysdig.com/555-benchmark/.

Tan, J., Pan, X., Kavulya, S., Gandhi, R. et Narasimhan, P. (2008). SALSA : analyzing logs as state machines. *Proceedings of the First USENIX Conference on Analysis of System Logs*, (WASL'08), 6.

Tavallaee, M., Bagheri, E., Lu, W. et Ghorbani, A. A. (2009). A detailed analysis of the KDD CUP 99 data set. *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*, pp. 1–6. doi : 10.1109/CISDA.2009.5356528.

The Bro Project. (2018). Bro IDS. Repéré le 2024-01-17 à https://old.zeek.org/manual/2.5.5/broids/index.html.

Tines. (2023). Voice of the SOC 2023 Report. Repéré le 2024-02-06 à https://www.tines.com/access/report-voice-of-the-soc-2023.

Treasury Board of Canada Secretariat. (2020). Event Logging Guidance. Repéré le 2023-12-19 à https://www.canada.ca/en/government/system/digital-government/online-security-privacy/event-logging-guidance.html.

Turcotte, M. J. M., Kent, A. D. et Hash, C. (2018). Unified Host and Network Data Set. Dans *Security Science and Technology* (vol. 03, pp. 1–22). WORLD SCIENTIFIC (EUROPE). doi : 10.1142/9781786345646_001.

UC Irvine. (1999). KDD Cup 1999. University of California, Irvine, Department of Computer Science. Repéré le 2024-01-25 à http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.

Unit42. (2024). Incident Response Report 2024. Palo Alto Networks. Repéré à https://www.paloaltonetworks.com/resources/research/unit-42-incident-response-report.

United States Senate Committee on Commerce, Science, and Transportation. (2014). *A "Kill Chain" Analysis of the 2013 Target Data Breach*.

Ur Rehman, M., Ahmadi, H. et Ul Hassan, W. (2024). Flash : A Comprehensive Approach to Intrusion Detection via Provenance Graph Representation Learning. *2024 IEEE Symposium on Security and Privacy (SP)*, pp. 3552–3570. doi : 10.1109/SP54263.2024.00139.

Verkerken, M., D'hooge, L., Wauters, T., Volckaert, B. et De Turck, F. (2022). Towards Model Generalization for Intrusion Detection : Unsupervised Machine Learning Techniques. *Journal of Network and Systems Management*, 30(1), 12. doi : 10.1007/s10922-021-09615-7.

Vermeer, M., Van Eeten, M. et Gañán, C. (2022). Ruling the Rules : Quantifying the Evolution of Rulesets, Alerts and Incidents in Network Intrusion Detection. *Proceedings of the 2022 ACM on Asia Conference on Computer and Communications Security*, pp. 799–814. doi : 10.1145/3488932.3517412.

Vermeer, M., Kadenko, N., Van Eeten, M., Gañán, C. et Parkin, S. (2023). Alert Alchemy : SOC Workflows and Decisions in the Management of NIDS Rules. *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, pp. 2770–2784. doi : 10.1145/3576915.3616581.

Vielberth, M., Bohm, F., Fichtinger, I. et Pernul, G. (2020). Security Operations Center : A Systematic Study and Open Challenges. *IEEE Access*, 8, 227756–227779. doi : 10.1109/ACCESS.2020.3045514.

Villarreal-Vasquez, M., Modelo-Howard, G., Dube, S. et Bhargava, B. (2023). Hunting for Insider Threats Using LSTM-Based Anomaly Detection. *IEEE Transactions on Dependable and Secure Computing*, 20(1), 451–462. doi : 10.1109/TDSC.2021.3135639.

Wagner, D. et Soto, P. (2002). Mimicry attacks on host-based intrusion detection systems. *Proceedings of the 9th ACM conference on Computer and communications security*, pp. 255–264. doi : 10.1145/586110.586145.

Warrender, C., Forrest, S. et Pearlmutter, B. (1999). Detecting intrusions using system calls : alternative data models. *Proceedings of the 1999 IEEE Symposium on Security and Privacy (Cat. No.99CB36344)*, pp. 133–145. doi : 10.1109/SECPRI.1999.766910.

Wazuh Inc. (2024). Wazuh Open Source XDR and SIEM. Wazuh Inc. Repéré à https://wazuh.com/.

Whittaker, Z. (2024). Microsoft said it lost weeks of security logs for its customers' cloud products. TechCrunch. Repéré à https://techcrunch.com/2024/10/17/microsoft-said-it-lost-weeks-of-security-logs-for-its-customers-cloud-products/.

Woods, D. W., Böhme, R., Wolff, J. et Schwarcz, D. (2023). Lessons Lost : Incident Response in the Age of Cyber Insurance and Breach Attorneys. *32nd USENIX Security Symposium (USENIX Security 23)*, pp. 2259–2273. Repéré à https://www.usenix.org/conference/usenixsecurity23/presentation/woods.

Wu, Y., Xie, Y., Liao, X., Zhou, P., Feng, D., Wu, L., Li, X., Wildani, A. et Long, D. (2023). Paradise : Real-Time, Generalized, and Distributed Provenance-Based Intrusion Detection. *IEEE Transactions on Dependable and Secure Computing*, 20(2), 1624–1640. doi : 10.1109/TDSC.2022.3160879.

Xie, Y., Feng, D., Hu, Y., Li, Y., Sample, S. et Long, D. (2020). Pagoda : A Hybrid Approach to Enable Efficient Real-Time Provenance Based Intrusion Detection in Big Data Environments. *IEEE Transactions on Dependable and Secure Computing*, 17(6), 1283–1296. doi : 10.1109/TDSC.2018.2867595.

Xin, D., Miao, H., Parameswaran, A. et Polyzotis, N. (2021). Production Machine Learning Pipelines : Empirical Analysis and Optimization Opportunities. *Proceedings of the 2021 International Conference on Management of Data*, pp. 2639–2652. doi : 10.1145/3448016.3457566.

Yang, X., Peng, G., Zhang, D., Gao, Y. et Li, C. (2023). PowerDetector : Malicious PowerShell script family classification based on multi-modal semantic fusion and deep learning. *China Communications*, 20(11), 202–224. doi : 10.23919/JCC.fa.2022-0509.202311.

Zhang, X., Xu, Y., Lin, Q., Qiao, B., Zhang, H., Dang, Y., Xie, C., Yang, X., Cheng, Q., Li, Z., Chen, J., He, X., Yao, R., Lou, J.-G., Chintalapati, M., Shen, F. et Zhang, D. (2019). Robust log-based anomaly detection on unstable log data. *Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, pp. 807–817. doi : 10.1145/3338906.3338931.

Zhu, T., Yu, J., Xiong, C., Cheng, W., Yuan, Q., Ying, J., Chen, T., Zhang, J., Lv, M., Chen, Y., Wang, T. et Fan, Y. (2023). APTSHIELD : A Stable, Efficient and Real-Time APT Detection System for Linux Hosts. *IEEE Transactions on Dependable and Secure Computing*, 20(6), 5247–5264. doi : 10.1109/TDSC.2023.3243667.

Zipperle, M., Gottwalt, F., Chang, E. et Dillon, T. (2023). Provenance-based Intrusion Detection Systems : A Survey. *ACM Computing Surveys*, 55(7), 1–36. doi : 10.1145/3539605.