UQAC
Université du Québec
à Chicoutimi

ANALYSE DYNAMIQUE DE CONTAMINATION POUR LA DÉTECTION DES FUITES DE DONNÉES SENSIBLES BASÉE SUR LA PROVENANCE

PAR AMADOU KABINÉ TRAORÉ

MÉMOIRE PRÉSENTÉ À L'UNIVERSITÉ DU QUÉBEC À CHICOUTIMI EN VUE DE L'OBTENTION DU GRADE MAÎTRISE ÈS SCIENCES (M. SC.) EN INFORMATIQUE

QUÉBEC, CANADA

© AMADOU KABINÉ TRAORÉ, 2026

RÉSUMÉ

La multiplication des systèmes informatiques manipulant des données personnelles, financières et opérationnelles a considérablement accru les risques de fuite de données sensibles. Les approches traditionnelles de prévention des fuites de données (Data Loss Prevention - DLP) reposent principalement sur l'analyse du contenu des informations afin d'identifier des motifs sensibles. Cependant, dans plusieurs environnements réels, des valeurs identiques peuvent apparaître à la fois dans des contextes sensibles et non sensibles, ce qui entraîne un nombre important de faux positifs. Ce mémoire propose une approche de détection des fuites de données sensibles fondée sur la provenance des informations. Contrairement aux approches classiques de prévention des fuites de données fondées uniquement sur l'analyse du contenu, l'approche proposée s'appuie sur une analyse dynamique de contamination (Taint Analysis) basée sur la provenance des données à partir de la corrélation de journaux hétérogènes provenant de différentes sources, notamment les bases de données, les systèmes de fichiers et les journaux. Cette approche permet ainsi un déploiement non intrusif dans des environnements réels.

Le système développé intègre également une plateforme SIEM afin de permettre une supervision en quasi-temps réel des événements de sécurité et une meilleure exploitation opérationnelle des alertes générées. Les résultats expérimentaux montrent que l'approche proposée réduit significativement les faux positifs en prenant en compte l'origine des données, tout en conservant une capacité de détection élevée. Ce travail met ainsi en évidence l'importance de la provenance dans la détection des fuites de données sensibles et propose une solution applicable dans des environnements complexes et hétérogènes. Les travaux présentés dans ce mémoire ont également fait l'objet d'une publication scientifique acceptée au *The Workshop on Security and Privacy Analytics (IWSPA 2026)*, dans le cadre du *12th ACM International Workshop on Security and Privacy Analytics*.

ABSTRACT

The increasing number of information systems handling personal, financial, and operational data has considerably increased the risk of sensitive data leakage. Traditional Data Loss Prevention (DLP) approaches mainly rely on content analysis to identify sensitive patterns. However, in many real-world environments, identical values may appear in both sensitive and non-sensitive contexts, leading to a significant number of false positives.

This thesis proposes an approach for detecting sensitive data leakage based on data provenance. Unlike conventional DLP approaches that rely solely on content analysis, the proposed approach uses Dynamic Taint Analysis based on data provenance through the correlation of heterogeneous logs collected from multiple sources, including databases, file systems, and system logs. This approach therefore enables non-intrusive deployment in real-world environments.

The developed system also integrates a SIEM platform to provide near real-time monitoring of security events and improve the operational exploitation of generated alerts. Experimental results show that the proposed approach significantly reduces false positives by taking data origin into account while maintaining a high detection capability.

This work highlights the importance of data provenance in the detection of sensitive data leakage and proposes a solution applicable to complex and heterogeneous environments.

The work presented in this thesis has also resulted in a scientific publication accepted at the *Workshop on Security and Privacy Analytics (IWSPA 2026)*, held as part of the *12th ACM International Workshop on Security and Privacy Analytics*.

TABLE DES MATIÈRES

RÉSUMÉ . . . . . **i**

ABSTRACT . . . . . **ii**

TABLE DES MATIÈRES . . . . . **iii**

LISTE DES TABLEAUX . . . . . **vi**

LISTE DES FIGURES . . . . . **vii**

LISTE DES ABRÉVIATIONS . . . . . **ix**

DÉDICACE . . . . . **x**

REMERCIEMENTS . . . . . **xi**

AVANT-PROPOS . . . . . **xii**

I INTRODUCTION **1**

1.1 CONTEXTE DE LA RECHERCHE 2

1.2 PROBLÉMATIQUE DE LA RECHERCHE 5

1.3 CONTRIBUTION DE LA RECHERCHE 7

1.4 APPROCHE DE LA RECHERCHE 9

1.5 STRUCTURE DU DOCUMENT 10

II CONCEPTES FONDAMENTAUX 12

2.1 PROTECTION DES DONNÉES SENSIBLES 12

2.2 L'ANALYSE DYNAMIQUE DE CONTAMINATION 14

2.3 VECTEURS D'EXFILTRATION DES DONNÉES 15

2.3.1 EXPORTATION DE DONNÉES À PARTIR DES BASES DE DONNÉES RELATIONNELLES 15

2.3.2 EXFILTRATION DE DONNÉES VIA LES APPLICATIONS WEB ET LE PROTOCOLE HTTP 16

2.3.3 VECTEURS D'ÉXFILTRATION SECONDAIRES 17

2.3.4 PÉRIMÈTRE DE L'ÉTUDE ET JUSTIFICATION DU CHOIX DES VECTEURS 18

2.4 SYSTÈMES DE GESTION DES ÉVÉNEMENTS DE SÉCURITÉ (SIEM) . 18

2.5 GESTION DE LA PROVENANCE DES DONNÉES 19

III ÉTUDE DE L'ÉTAT DE L'ART 23

3.1 DÉTECTION FONDÉE SUR LES SIGNATURES ET LES EXPRESSIONS RÉGULIÈRES 23

3.2 EMPREINTES NUMÉRIQUES 25

3.3 APPROCHES FONDÉES SUR L'APPRENTISSAGE AUTOMATIQUE . . . 27

3.4 ANALYSE STATIQUE 29

3.5 ANALYSE DYNAMIQUE DE CONTAMINATION TRADITIONNELLE . . 30

3.6 DISCUSSION 32

IV MÉTHODOLOGIE 37

4.1 DÉVELOPPEMENT DU SYSTÈME D'ANALYSE DYNAMIQUE DE CONTAMINATION FONDÉ SUR LA PROVENANCE 38

4.1.1 OBJECTIFS ET FONDEMENTS DE L'IMPLÉMENTATION 38

4.1.2 ARCHITECTURE GÉNÉRALE ET POSITIONNEMENT DU MOTEUR DU SYSTÈME . . . 39
4.1.3 DÉCLARATION EXPLICITE DES BASES DE DONNÉES ET RÔLE DE LA CONFIGURATION . . . 40
4.1.4 CHARGEMENT DES DONNÉES SENSIBLES ET CONSTRUCTION DU TAINT_POOL . . . 43
4.1.5 SURVEILLANCE DES EXPORTATIONS SQL ET GESTION TEMPORELLE . . . 45
4.1.6 ANALYSE DES EXFILTRATIONS VIA HTTP ET RÉSOLUTION DES AMBIGUÏTÉS . . . 48
4.1.7 GÉNÉRATION DES ALERTES ET INTÉGRATION DANS LE SIEM WAZUH . . . 49

V RESULTATS . . . 64

5.1 RÉSULTATS EXPÉRIMENTAUX . . . 65
5.1.1 OBJECTIF DE L'ÉVALUATION EXPÉRIMENTALE ET COMPARAISON AVEC DES APPROCHES EXISTANTES . . . 66
5.1.2 ANALYSE DE LA PRÉCISION ET ABSENCE DE FAUX POSITIFS 68
5.1.3 SYNTHÈSE DES RÉSULTATS . . . 70
5.1.4 ANALYSE DE LA SCALABILITÉ ET DE LA ROBUSTESSE . . . 72

VI DISCUSSION, LIMITES ET PERSPECTIVES . . . 76

6.1 DISCUSSION . . . 77
6.2 LIMITES DE L'APPROCHE . . . 79
6.3 RÉPONSES AUX QUESTIONS DE RECHERCHE . . . 82
6.4 PERSPECTIVES ET TRAVAUX FUTURS . . . 85

CONCLUSION . . . 87

BIBLIOGRAPHIE . . . 89

LISTE DES TABLEAUX

3.1 Comparaison des approches de détection de fuites de données selon la prise en compte de la provenance 35

5.1 Comparaison des approches de détection en présence d'ambiguïté de provenance 71

5.2 Scalabilité en termes d'utilisation mémoire et de temps de construction de l'index 72

5.3 Latence de détection et débit en fonction du nombre de valeurs indexées . . . 74

LISTE DES FIGURES

1.1 Impact boursier des cyberattaques : une étude d'événement. 2

2.1 Relation entre le risque lié à la protection de la vie privée et le risque organisationnel (ENTERPRISE) 13

2.2 Architecture du système SIEM (Sheeraz et al.) 19

4.1 Architecture du l'analyse dynamique de contamination sur la provenance, intégrée au SIEM Wazuh 40

4.2 Déclaration des bases de données sensibles et non sensibles 41

4.3 Duplication de données identiques dans les deux bases de données distinctes . 42

4.4 Construction du TAINT_POOL 44

4.5 Chargement des données par le Taint_Pool 44

4.6 Analyse continue des fichiers d'export de données 46

4.7 Concéption de la surveillance des accès HTTP 47

4.8 MySQL General Query Log 48

4.9 Exportation de données sensibles depuis une base sensible (donnees_co) . . . 50

4.10 Exportation de données sensibles depuis une base sensible (donnees_co) . . . 51

4.11 Visualisation des logs générés par le système de Dynamic Taint Analysis sur les données sensible de donnees_co 53

4.12 Journal JSON des alertes de fuite de données sensibles générées par Wazuh . 55

4.13 Exportation de données sensibles depuis une base sensible (donnees_co_B) . 56

4.14 Exportation de données sensibles depuis une base non sensible (donnees_co_B) 56

4.15 Visualisation des logs générés par le système de Dynamic Taint Analysis sur les données non sensible de donnees_co_B) 58

4.16 Visualisation des alertes de fuite de données sensibles dans le tableau de bord Wazuh 59

4.17 Affichage des alertes de fuite de données sensibles exportées vers un fichier texte ou HTTP dans le tableau de bord Wazuh 60

LISTE DES ABRÉVIATIONS

DLP Data Loss Prevention
HTTP HyperText Transfer Protocol
IFC Information Flow Control
JSON JavaScript Object Notation
NAS Numéro d'assurance sociale
RGPD Règlement général sur la protection des données
SIEM Security Information and Event Management
SQL Structured Query Language

DÉDICACE

À mon père et à ma mère,
Je vous dédie ce travail en espérant que vous serez toujours fiers de moi.

Je dédie également ce travail à tous les membres de ma famille, pour leurs encouragements, leur soutien, leur affection et leur confiance qui m'ont permis de poursuivre mes études.

Aux professeurs Sylvain Hallé et Fehmi Jaafar, qui m'ont donné confiance et espoir, et qui m'ont soutenu tout au long de mes recherches.

À toutes les personnes qui, de près ou de loin, ont contribué d'une manière ou d'une autre à la réalisation de ce document.

Qu'ALLAH leur accorde la santé, la prospérité et les gratifie de Sa miséricorde.

REMERCIEMENTS

Je remercie, en premier lieu, Allah pour le courage, la patience et la santé qu'Il m'a accordés tout au long de ce parcours, me permettant ainsi de poursuivre mes études et de mener à bien ce travail.

Je tiens à exprimer ma profonde gratitude à mes directeurs de recherche, le professeur Sylvain Hallé et le professeur Fehmi Jaafar, pour leur encadrement, leur disponibilité et leurs précieux conseils. Leurs orientations scientifiques, leur exigence et leur soutien constant ont été déterminants dans la réalisation de ce mémoire. Leur capacité à transmettre leur savoir et à stimuler la réflexion a largement contribué à enrichir ce travail et à développer mon esprit critique.

Je souhaite également adresser mes sincères remerciements à l'ensemble des professeurs que j'ai eu l'opportunité de côtoyer au cours de mon parcours académique. Leurs enseignements, riches et rigoureux, ont joué un rôle essentiel dans l'acquisition de mes connaissances et dans ma formation en tant qu'ingénieur et chercheur.

Mes remerciements s'adressent également au personnel du département d'informatique et de mathématiques (DIM) de l'Université du Québec à Chicoutimi, pour leur soutien et leur accompagnement tout au long de mes études.

Enfin, je tiens à exprimer toute ma reconnaissance à ma famille, à mes amis et à mes proches pour leur soutien indéfectible, leurs encouragements et leur présence constante. Leur confiance et leur patience ont été une source de motivation essentielle tout au long de ce parcours.

AVANT-PROPOS

Ce mémoire constitue une étape importante dans mon parcours académique, marquant la fin d'un cycle d'apprentissage riche en découvertes et en défis. Au fil de mes études, j'ai progressivement développé un intérêt particulier pour la cybersécurité, notamment pour les problématiques liées à la protection des données dans des systèmes modernes de plus en plus complexes.

Le choix du sujet de ce mémoire s'est imposé naturellement à travers les réflexions menées sur les limites des approches actuelles de détection des fuites de données. Comprendre pourquoi certaines solutions échouent dans des contextes réels et chercher à proposer une alternative plus adaptée a constitué une motivation centrale dans la réalisation de ce travail.

La conception et l'expérimentation de l'approche proposée ont nécessité un travail approfondi, mêlant analyse théorique et mise en œuvre pratique. Ce processus m'a permis de mieux appréhender les enjeux liés à la gestion des données sensibles, mais aussi de développer une approche plus critique face aux solutions existantes. Au-delà des aspects techniques, ce mémoire représente également un cheminement personnel. Il a été marqué par des phases d'incertitude, de remise en question, mais aussi par des avancées significatives qui ont contribué à structurer ma réflexion et à renforcer mes compétences. Ce document reflète l'ensemble de ce parcours. Il n'a pas vocation à être exhaustif et certaines améliorations restent possibles. C'est pourquoi toute remarque ou suggestion visant à enrichir ce travail sera accueillie avec intérêt.

CHAPITRE I

INTRODUCTION

La protection des données personnelles est devenue l'une des principales préoccupations des acteurs de la cybersécurité. Dans un contexte où la valeur économique et stratégique de l'information ne cesse de croître, les organisations cherchent à prévenir toute exposition involontaire ou malveillante de données sensibles. À ce titre, (Skatova et al., 2023) ont montré dans leur étude que les individus sont prêts à payer pour éviter une fuite potentielle de leurs informations confidentielles, notamment lorsque celles-ci concernent des données financières ou médicales, dont la sensibilité accroît fortement la perception du risque.

Par ailleurs, cette protection n'est plus seulement une pratique recommandée : elle est devenue une obligation légale dans de nombreux secteurs. C'est notamment le cas des entreprises européennes, qui doivent aujourd'hui se conformer à des normes strictes en matière de gestion et de protection des données personnelles, telles que le Règlement général sur la protection des données (RGPD), décrit en détail par (Boulesnane et al., 2020). Ce cadre réglementaire impose aux organisations de garantir la confidentialité, l'intégrité et la traçabilité des informations personnelles de leurs employés ou de leurs clients, sous peine de sanctions sévères.

1.1 CONTEXTE DE LA RECHERCHE

Les fuites de données constituent aujourd'hui un enjeu majeur pour les organisations, tout secteurs confondus. Les études menées par (Muktadir-Al-Mukit et Ali, 2025) révèlent que le coût moyen mondial d'une fuite de données a atteint 4,88 millions de dollars en 2024 (Muktadir-Al-Mukit et Ali, 2025), soit une hausse de 10 % par rapport à 2023. Parallèlement, l'impact économique global de la cybercriminalité s'élevait à 9 400 milliards de dollars américains en 2024, avec une augmentation projetée de 15 % en 2025. Ces chiffres soulignent à quel point les fuites de données entraînent des pertes économiques considérables, particulièrement dans le secteur numérique, où elles se traduisent directement par une diminution des revenus et une perte de compétitivité pour les entreprises. Cette réalité est également illustrée par (Muktadir-Al-Mukit et Ali, 2025) figure1.1, qui présente graphiquement l'impact des cyberattaques sur la valeur des actions d'une entreprise au cours d'une période donnée.

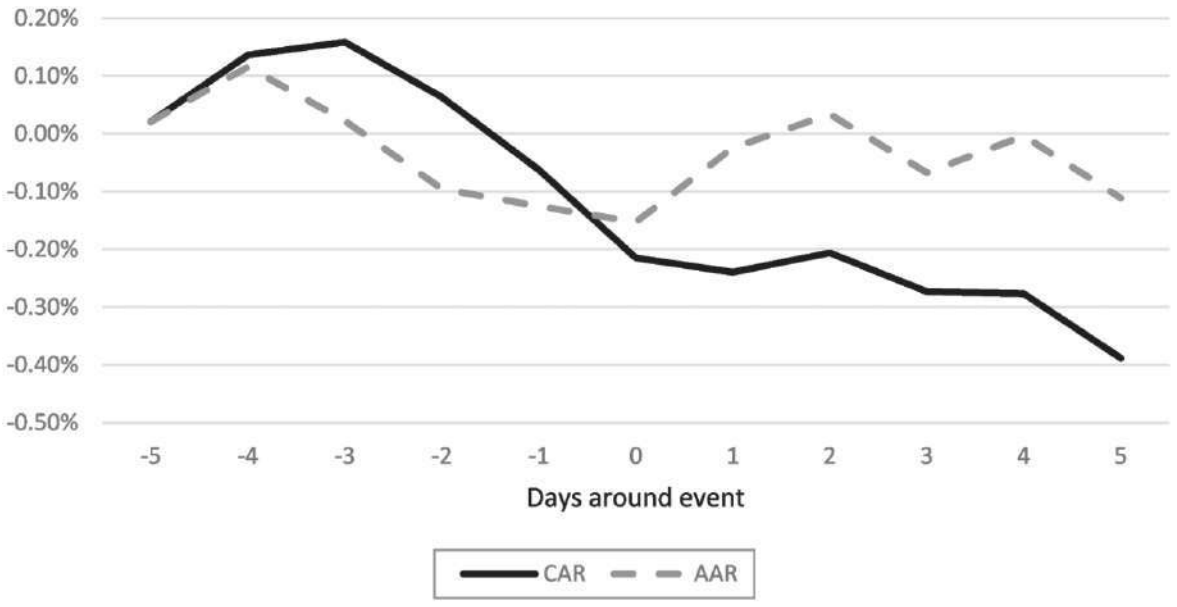

Figure 1.1 : Impact boursier des cyberattaques : une étude d'événement.

De nombreuses recherches récentes se sont penchées sur ce phénomène afin d'en mieux comprendre les déterminants, les impacts et les réponses organisationnelles les plus appropriées. Une revue systématique publiée en 2024 par (Liu et Babar, 2024), portant sur 203 études empiriques, a dressé une cartographie détaillée des facteurs de risque en cybersécurité et des impacts associés aux cyberattaques (Liu et Babar). Les auteurs mettent en lumière l'existence de mécanismes de rétroaction permettant aux organisations d'exploiter les leçons tirées des violations subies pour renforcer progressivement leur posture de sécurité. Cette contribution propose également une taxonomie structurée des risques et des conséquences des fuites de données, offrant ainsi aux dirigeants et aux régulateurs un cadre d'analyse utile pour améliorer la résilience cyber des entreprises et des organisations.

Par ailleurs, des études de cas récentes permettent d'examiner plus en profondeur la dynamique et les spécificités des fuites de données dans différents secteurs. Comme le souligne (Liu, 2021), le domaine financier constitue un exemple particulièrement sensible : les échanges interbancaires reposent sur l'infrastructure SWIFT, essentielle aux transactions internationales. Or, avec la mondialisation et la numérisation accélérée des services bancaires, plusieurs brèches de sécurité ont été observées, souvent liées à des pratiques de protection insuffisantes. Dans cette perspective, les analyses de risque relatives aux fuites de données visent à identifier les causes profondes des incidents et à mesurer leurs répercussions multiples financières, juridiques, opérationnelles et stratégiques. La littérature récente converge vers l'idée que le risque cyber est intrinsèquement multidimensionnel et touche l'ensemble de l'organisation. La revue systématique de (Liu et Babar) rappelle d'ailleurs que les facteurs de vulnérabilité vont des faiblesses technologiques aux enjeux organisationnels et humains, tandis que les impacts d'une cyberattaque dépassent largement le cadre immédiat de l'incident.

Cependant, de nombreux travaux ont été menés pour proposer des solutions visant à protéger les données personnelles et à prévenir les fuites. Parmi ces approches figurent les méthodes basées sur les expressions régulières, qui s'appuient sur des mots-clés et demeurent les techniques les plus largement déployées dans les outils DLP commerciaux (Brindha et Shaji,2015) (Kaur et al.,2017). Ces approches détectent des motifs numériques ou textuels fixes (par. ex. formats de cartes de crédit, numéros d'assurance sociale), souvent combinés à des heuristiques simples.

D'autres travaux reposent sur des techniques fondées sur l'apprentissage automatique, qui classent les données selon leur sémantique ou des caractéristiques contextuelles (Alneyadi et al.,2015)(Lu et al.2018). Lorsque l'information sensible apparaît dans des environnements riches en langage naturel, les modèles peuvent exploiter les indices lexicaux ou distributionnels entourant les identifiants (par ex. des mots comme « SSN », « compte »).

Enfin, certaines approches se basent sur l'analyse statique, qui examine le code d'un programme (par ex. APK Android ou code serveur) afin d'identifier les flux potentiels entre sources et puits (Xiaohui et al., 2014)(Yang et al.2015). Cette méthode permet de détecter efficacement les fuites exprimables dans les graphes de contrôle et de flux de données de l'application. Dans ce mémoire, nous nous concentrons par l'utilisation d'une solution d'analyse dynamique de contamination spécialement conçue pour détecter les fuites de données sensibles. L'analyse dynamique de contamination (Dynamique Taint Analysis), également connue sous le nom de suivi dynamique des flux d'information, est l'approche privilégiée pour analyser la sécurité des logiciels en surveillant les données contaminées durant l'exécution d'un programme (Alashjee et al. 2019,Zhu et al.2011). Le concept fondamental de l'analyse dynamique de contamination consiste à étiqueter certaines valeurs de données entrantes comme contaminées, en particulier celles provenant de sources sensibles ou non-fiables (la source), telles que les API applicatives

ou les interfaces réseau. Ces étiquettes de contamination sont ensuite propagées au fil de l'exécution des instructions, ce qui implique le marquage des valeurs dans les registres du processeur ou dans les zones mémoire (Alashjee et al., 2019).

L'objectif de ce mémoire est de contribuer à l'amélioration de la protection des données personnelles face aux cyberattaques en proposant un système de détection reposant sur l'analyse dynamique de contamination. Ce système introduit un mécanisme décisionnel permettant de classifier rigoureusement les données sensibles et non-sensibles, tout en assurant une propagation précise des marqueurs de sensibilité au sein des flux d'exécution. Les événements ainsi identifiés sont ensuite transmis et visualisés dans un moteur SIEM, offrant un dispositif de supervision centralisé et permettant le déclenchement d'alertes fiables, dans une perspective de réduction des faux positifs et d'augmentation de la pertinence des détections.

1.2 PROBLÉMATIQUE DE LA RECHERCHE

Les méthodes d'analyse dynamique de contamination apportent des contributions majeures au domaine de la cybersécurité. Initialement utilisées dans des contextes tels que la sécurité logicielle, elles se distinguent par leur capacité à tracer la propagation des données dans un système, mais la plupart des travaux existants se concentrent uniquement sur le contenu des données. C'est notamment le cas des approches fondées sur les expressions régulières, qui s'appuient sur des mécanismes de balayage pour détecter des motifs textuels similaires à des modèles préconfigurés et déclencher ainsi des alertes. On peut également citer les méthodes de détection basées sur l'apprentissage automatique avec des modèles de traitement du langage naturel afin d'identifier des expressions reconnues à l'aide de filtres sémantiques.

Cependant, la problématique commune à ces approches est qu'elles peuvent détecter des expressions textuelles semblables à celles attendues par l'algorithme, sans pour autant correspondre à de réelles données sensibles. Cette ambiguïté conduit au déclenchement d'alertes inappropriées alors que la donnée analysée n'est pas sensible, mais simplement similaire à un motif prédéfini. Il en résulte un nombre important de faux positifs générés par ces systèmes. Bien que ces méthodes soient sophistiquées et produisent parfois de bons résultats, elles ne prennent pas en compte la provenance de la donnée. Or, une même valeur peut provenir d'une source non sensible ou, au contraire, d'une source déclarée sensible - une distinction impossible à établir pour des approches basées exclusivement sur le contenu.

Dans cette perspective, notre recherche propose une nouvelle manière d'aborder la détection de données sensibles, en tenant compte non seulement de leur structure, mais également de leur **provenance réelle**, afin de réduire significativement les faux positifs. Ainsi, ce travail vise à répondre aux questions suivantes :

1. Comment l'analyse dynamique de contamination peut-elle être utilisée pour suivre de manière fiable la provenance des données sensibles au sein d'un système informatique ?
2. Dans quelle mesure l'intégration de la provenance des données permet-elle de réduire les faux positifs générés par les approches traditionnelles de détection (regex, machine learning, NLP, SIEM classiques) ?
3. De quelle façon l'analyse dynamique de contamination peut-elle être intégrée à un SIEM tel que Wazuh afin de produire des alertes exploitables et contextualisées ?
4. Dans quelles conditions l'approche permet-elle d'améliorer la détection en temps réel de l'exfiltration de données sensibles vers des fichiers externes ou des canaux réseau ?

1.3 CONTRIBUTION DE LA RECHERCHE

Ce travail de recherche apporte plusieurs contributions majeures dans le domaine de la sécurité des données personnelles, en particulier pour la détection des fuites de données sensibles dans les environnements reposant sur des bases de données relationnelles. Notre approche établit un lien inédit entre les techniques de l'analyse dynamique de contamination et les systèmes opérationnels de gestion des événements de sécurité (SIEM). Les principales contributions sont les suivantes :

• **Développement d'un moteur de l'analyse dynamique de contamination opérationnel pour bases relationnelles :** Nous concevons et mettons en œuvre un moteur de l'analyse dynamique de contamination capable de suivre, en temps réel, la propagation des valeurs sensibles au sein d'une base de données relationnelle. Contrairement aux systèmes de prévention des fuites (DLP) fondés sur des motifs textuels, notre solution associe à chaque valeur sensible un marqueur de provenance décrivant son origine (table, colonne, identifiant, session SQL, utilisateur, requête d'extraction, etc.). Ce suivi permet d'identifier avec précision l'origine d'une fuite et de comprendre son cheminement. À notre connaissance, il s'agit d'un des premiers systèmes de l'analyse dynamique de contamination véritablement opérationnels appliqués à un contexte de base de données.

• **Intégration native de l'analyse dynamique de contamination dans le SIEM Wazuh avec génération d'alertes enrichies :** Nous proposons la première intégration directe entre un moteur d'analyse dynamique de contamination et Wazuh. Notre système génère automatiquement des alertes JSON enrichies, incluant la provenance, le vecteur de fuite, le niveau de sensibilité et les métadonnées associées. Des règles personnalisées permettent à Wazuh de classifier ces alertes

et d'offrir une visualisation claire dans Kibana. Cette intégration transforme Wazuh en un SIEM capable de détecter des fuites basées non pas sur la syntaxe du contenu, mais sur la provenance réelle des données une fonctionnalité absente des installations Wazuh traditionnelles.

• **Réduction significative des faux positifs grâce à l'exploitation de la provenance :** Les techniques actuelles de détection fondées sur les expressions régulières ne distinguent pas deux valeurs textuellement identiques mais provenant de contextes différents. Notre approche résout cette ambiguïté en considérant qu'une valeur n'est sensible que si elle provient d'une source sensible. Ainsi, une même chaîne de caractères peut être classée différemment selon son origine. Dans nos expériences, l'approche par expressions régulières produit 397 faux positifs, tandis que notre système d'analyse dynamique de contamination n'en produit aucun. Cette contribution répond directement à l'un des défis les plus critiques des systèmes de détection : la maîtrise des faux positifs.

• **Détection multi-canal des fuites via exports SQL, mouvements de fichiers et requêtes HTTP :** 3 Les fuites de données réelles se produisent rarement via un unique canal. Notre architecture unifie trois vecteurs principaux : (1) les exports SQL via "`SELECT ... INTO OUTFILE`", (2) les transferts et manipulations de fichiers sensibles dans les répertoires surveillés désignés par la base de donnée, et (3) l'exfiltration via des requêtes HTTP, détectée à partir des journaux Apache. En reliant les marqueurs de provenance à ces différents canaux, le système offre une couverture plus complète et plus réaliste des scénarios de fuite.

• **Validation expérimentale dans des scénarios contrôlés.**

L'approche proposée est évaluée à travers un ensemble de scénarios reproduisant différents contextes d'exfiltration de données, notamment via des requêtes SQL, des écritures de fichiers et des échanges HTTP. Le protocole expérimental repose sur des situations où des données

identiques sont présentes dans des sources sensibles et non-sensibles, afin de mettre en évidence les limites des approches fondées uniquement sur le contenu. L'évaluation s'appuie sur des métriques standards telles que la précision, le rappel, le taux de faux positifs et le score F1. Les résultats obtenus montrent que les approches traditionnelles, notamment celles basées sur des expressions régulières et d'apprentissage automatique, génèrent un nombre important de faux positifs dans des contextes d'ambiguïté de provenance. À l'inverse, l'approche proposée, fondée sur l'analyse de la provenance, permet de distinguer correctement les données sensibles des données non-sensibles, y compris lorsque celles-ci sont identiques sur le plan syntaxique. Cette validation expérimentale met ainsi en évidence l'apport de la prise en compte de la provenance pour améliorer la précision de la détection et réduire significativement les faux positifs dans des environnements réalistes.

1.4 APPROCHE DE LA RECHERCHE

Ensuite, chaque événement suspect généré lors de la surveillance est enrichi avec des informations de provenance et transformé en alerte structurée. Ces alertes sont directement intégrées au SIEM Wazuh, qui les interprète à l'aide de règles personnalisées permettant de classifier les incidents selon leur criticité. L'objectif est d'offrir aux analystes SOC une visibilité exhaustive et contextualisée sur les incidents impliquant des données sensibles. Enfin, nous comparons notre approche aux méthodes traditionnelles de détection, notamment celles fondées sur des expressions régulières et la correspondance syntaxique utilisées dans de nombreuses solutions. Cette comparaison nous permet d'évaluer l'efficacité du système en termes de réduction des faux positifs, de précision de détection et de robustesse dans un environnement réel. Les

performances observées sont ensuite discutées afin de mettre en évidence les avantages et limites de notre approche basée sur la provenance, par rapport aux travaux antérieurs.

1.5 STRUCTURE DU DOCUMENT

Le reste de ce mémoire est organisé en plusieurs chapitres principaux afin de présenter, analyser et évaluer l'approche proposée pour la détection des fuites de données sensibles au sein des environnements reposant sur des bases de données relationnelles.

• **Chapitre 2 : Concepts fondamentaux.** Ce chapitre introduit les notions essentielles liées à la protection des données sensibles, à l'analyse dynamique de contamination, aux systèmes SIEM, ainsi qu'à la gestion de la provenance des données. Il décrit également les différents vecteurs d'exfiltration considérés dans ce mémoire (SQL, fichiers, HTTP).

• **Chapitre 3 : ÉTUDE DE L'ÉTAT DE L'ART.** Ce chapitre analyse les approches existantes dans la détection de fuites de données, incluant les méthodes basées sur des expressions régulières, les solutions DLP traditionnelles, les approches par apprentissage automatique, ainsi que les premières tentatives d'application de l'analyse dynamaique de contamination dans la cybersécurité. Il permet de situer notre recherche par rapport à l'état de l'art.

• **Chapitre 4 : Méthodologie.** Ce chapitre présente en détail la méthodologie adoptée pour cette recherche. Il décrit le processus complet allant de l'extraction des données sensibles à la construction des marqueurs de provenance, au suivi dynamique, à la surveillance multi-canal, et enfin à l'intégration dans le SIEM Wazuh. Il introduit aussi les scénarios d'exfiltration utilisés pour l'évaluation.

• **Chapitre 5 : Résultats.** Ce chapitre expose les résultats expérimentaux obtenus, notamment la comparaison entre notre approche d'analyse dynamique de contamination et les méthodes fondées sur les expressions régulières. Il analyse la précision, le taux de faux positifs, la robustesse de l'approche ainsi que la qualité des alertes générées.

• **Chapitre 6 : Discussion et limites.** Ce chapitre discute les implications des résultats, les avantages et les limites de notre approche, ainsi que les conditions dans lesquelles elle peut être déployée efficacement dans un environnement réel.

• **Conclusion et pistes de recherche future.** Ce chapitre conclut le mémoire en résumant les contributions majeures de ce travail. Il propose également des orientations possibles pour améliorer l'analyse de provenance, étendre la détection à de nouveaux vecteurs d'exfiltration, ou intégrer des mécanismes d'apprentissage automatique pour renforcer la précision du système.

CHAPITRE II

CONCEPTES FONDAMENTAUX

Ce chapitre présente les concepts fondamentaux nécessaires à la compréhension de l'approche proposée dans ce mémoire. Il introduit d'abord les enjeux liés à la protection des données sensibles ainsi que les principales problématiques associées aux fuites de données dans les environnements organisationnels modernes. Le chapitre aborde ensuite les notions essentielles relatives à l'analyse dynamique de contamination (Dynamic Taint Analysis), aux vecteurs d'exfiltration des données, aux systèmes de gestion des événements de sécurité (Security Information and Event Management - SIEM) ainsi qu'à la gestion de la provenance des données. L'objectif est de fournir le cadre théorique et technique nécessaire pour comprendre les mécanismes de détection proposés dans ce mémoire ainsi que les limites des approches traditionnelles fondées uniquement sur l'analyse du contenu des données.

2.1 PROTECTION DES DONNÉES SENSIBLES

La protection des données sensibles constitue aujourd'hui un enjeu central pour les organisations, tant du point de vue légal qu'opérationnel. Avec la multiplication des cyberattaques, des compromissions internes et des fuites accidentelles, les entreprises doivent garantir que les informations critiques telles que les numéros d'assurance sociale, les données biométriques, les

informations financières ou médicales soient traitées conformément aux normes réglementaires. Selon (ENTERPRISE, 2020), la gestion des données sensibles nécessitent non seulement des contrôles d'accès stricts, mais également des mécanismes permettant de détecter toute tentative d'exfiltration ou d'usage non autorisé. De nombreuses études, dont celle de (Roman et al., 2020), montrent que les fuites de données surviennent fréquemment à cause d'erreurs humaines ou de manipulations internes non contrôlées, ce qui justifie la mise en place de systèmes de surveillance automatisés capables d'intervenir en temps réel.

Les réglementations telles que le RGPD en Europe, la HIPAA aux États-Unis et la Loi 25 au Québec renforcent par ailleurs les exigences en matière de protection des données personnelles et imposent aux organisations de documenter la provenance et l'usage des données sensibles (Voigt et Von dem Bussche, 2017). Dans ce contexte, les systèmes traditionnels basés sur des règles statiques atteignent rapidement leurs limites, ce qui ouvre la voie à des approches plus dynamiques et contextuelles, comme la Dynamic Taint Analysis étudiée dans ce travail.

Figure 2.1 : Relation entre le risque lié à la protection de la vie privée et le risque organisationnel (ENTERPRISE)

Les données personnelles constituent un pilier essentiel au bon fonctionnement des organisations. Une simple négligence dans leur gestion peut entraîner des conséquences graves, telles que la perte de confiance des clients, des atteintes à la réputation ou encore des sanctions légales. Ces

risques, souvent sous-estimés, illustrent l'importance d'une approche rigoureuse en matière de protection de l'information, comme le résume la figure 2.1.

2.2 L'ANALYSE DYNAMIQUE DE CONTAMINATION

L'analyse dynamique de contamination est une technique d'analyse dynamique largement utilisée dans la recherche en sécurité informatique pour suivre la propagation de données marquées au sein d'un programme ou d'un système (Schwartz et al., 2010). À l'analyse dynamique de contamination a été conçue pour détecter des attaques de corruption mémoire, telles que les débordements de tampon, en marquant certaines données d'entrée et en observant leur parcours pendant l'exécution (Newsome et Song, 2005). L'idée générale consiste à attacher un « taint » (ou marqueur) à des données considérées comme sensibles, puis à propager ces marqueurs lors des opérations effectuées sur ces données.

Dans notre recherche, nous adaptons l'analyse dynamique de contamination à un environnement non traditionnel : les bases de données relationnelles. Très peu de travaux ont appliqué l'analyse dynamique de contamination à la surveillance de données SQL, car la plupart des méthodes se concentrent sur les flux mémoire ou les langages interprétés. Les approches existantes, comme celles de (Liu et al., 2015), proposent une propagation de taint dans les systèmes distribués, mais ne considèrent pas explicitement la provenance des données dans des bases relationnelles. Notre adaptation permet d'associer à chaque valeur sensible un marqueur contenant sa table d'origine, sa colonne, son identifiant unique, la session SQL ayant exécuté la requête et le contexte temporel. Cette adaptation de l'analyse dynamique de contamination représente une innovation clé dans la détection de fuites de données en environnement organisationnel.

2.3 VECTEURS D'EXFILTRATION DES DONNÉES

L'exfiltration de données sensibles correspond au processus par lequel des informations confidentielles quittent leur environnement de stockage ou de traitement légitime pour être transférées vers un espace non autorisé. Contrairement aux attaques purement intrusives, les fuites de données exploitent fréquemment des mécanismes légitimes du système d'information, ce qui complique leur détection. Ces mécanismes peuvent être. utilisés aussi bien par des acteurs malveillants que dans des contextes d'erreurs humaines ou de mauvaises pratiques.

Dans les environnements organisationnels, les vecteurs d'exfiltration sont étroitement. liés aux fonctionnalités offertes par les systèmes applicatifs et les infrastructures logicielles. Les bases de données, les applications web et les services réseau constituent autant de points de sortie potentiels pour les données sensibles. De nombreuses études montrent que les fuites de données ne résultent pas nécessairement d'une compromission directe du système, mais plutôt d'un usage détourné ou non contrôlé de fonctionnalités autorisées (Roman et al.,2018). Cette caractéristique rend les approches de détection fondées uniquement sur la surveillance des accès ou sur des règles statiques insuffisantes.

2.3.1 EXPORTATION DE DONNÉES À PARTIR DES BASES DE DONNÉES RELATIONNELLES

Les systèmes de gestion de bases de données relationnelles intègrent nativement des mécanismes d'exportation permettant de transférer les résultats de requêtes SQL vers des fichiers externes. Dans MySQL, par exemple, la commande "`SELECT ... INTO OUTFILE`" permet d'écrire directement le contenu d'une requête dans un fichier stocké sur le système hôte. Ces

fonctionnalités sont largement utilisées dans des contextes légitimes, tels que la génération de rapports, la sauvegarde de données ou l'échange d'informations entre systèmes. Cependant, ces mécanismes constituent également un vecteur d'exfiltration privilégié lorsqu'ils sont utilisés sans contrôle strict. Une seule requête peut suffire à exporter un volume important de données sensibles, telles que des identifiants personnels, des numéros financiers ou des informations médicales. Les journaux générés par le système de fichiers ou par le moteur de base de données indique généralement la création ou la modification d'un fichier, mais ne permettent pas de déterminer si les données exportées sont sensibles ni d'identifier leur table ou colonne d'origine. Cette absence de contextualisation pose un problème majeur pour les systèmes de surveillance traditionnels. En l'absence d'informations sur la provenance des données, il devient impossible de distinguer un export légitime de données non sensibles d'une fuite critique impliquant des informations confidentielles. Cette limitation justifie la nécessité d'approches capables de suivre les données à un niveau plus fin, en associant chaque valeur exportée à son d'origine.

2.3.2 EXFILTRATION DE DONNÉES VIA LES APPLICATIONS WEB ET LE PROTOCOLE HTTP

Les applications web représentent un autre vecteur majeur d'exfiltration de données sensibles. Elles constituent une interface directe entre les bases de données et des clients distants, qu'il s'agisse d'utilisateurs humains ou de systèmes automatisés. Dans ce contexte, les données extraites des bases relationnelles peuvent être transmises via des réponses HTTP, des fichiers téléchargeables ou des flux de données retournés par des interfaces de programmation applicative (API). Le protocole HTTP, par sa nature stateless et largement autorisée dans les réseaux organisationnels, facilite le transfert de données sans attirer l'attention des mécanismes

de sécurité traditionnels. Les serveurs web génèrent des journaux détaillant les requêtes reçues et les réponses envoyées, tels que les fichiers `access.log` et `error.log`. Néanmoins, ces journaux se limitent généralement à des informations syntaxiques, comme les adresses IP, les URL ou les codes de réponse, sans fournir de détails sur le contenu réel des données transmises. Cette limitation entraîne une ambiguïté importante dans la détection des fuites. Une valeur apparaissant dans une réponse HTTP peut correspondre à une donnée hautement sensible ou à une information publique, sans qu'il soit possible de les différencier à partir des seuls journaux réseau. Les approches fondées sur des expressions régulières ou des signatures prédéfinies tentent de pallier ce problème, mais elles restent vulnérables aux faux positifs et aux contournements, notamment lorsque les données sont formatées, fragmentées ou encodées.

2.3.3 VECTEURS D'ÉXFILTRATION SECONDAIRES

Au-delà des exportations de fichiers et des communications HTTP, d'autres vecteurs d'exfiltration peuvent être observés dans les environnements organisationnels. Parmi ceux-ci figurent l'envoi de données par courrier électronique, la synchronisation avec des services de stockage infonuagique, ou encore l'utilisation de supports de stockage amovibles. Ces vecteurs exploitent également des fonctionnalités légitimes du système et peuvent être difficiles à détecter sans une instrumentation spécifique. Toutefois, l'analyse exhaustive de ces vecteurs nécessite des sources de données supplémentaires, telles que les journaux de messagerie, les API des services infonuagiques ou les événements matériels, ce qui dépasse le périmètre de ce mémoire. Leur mention permet néanmoins de situer le travail présenté dans un contexte plus large de protection des données.

2.3.4 PÉRIMÈTRE DE L'ÉTUDE ET JUSTIFICATION DU CHOIX DES VECTEURS

Dans ce mémoire, nous concentrons notre analyse sur les vecteurs d'exfiltration observables et instrumentables à partir de journaux système et applicatifs existants, à savoir l'exportation de données vers des fichiers et les communications HTTP. Ce choix est motivé par leur prévalence dans les environnements organisationnels et par leur pertinence pour l'étude de la Dynamic Taint Analysis appliquée aux bases de données relationnelles. En se limitant à ces vecteurs, il devient possible d'évaluer de manière précise la capacité d'un système fondé sur la provenance des données à distinguer les fuites réelles de données sensibles des opérations légitimes. Cette délimitation du périmètre permet également d'assurer la reproductibilité des expérimentations et de faciliter l'intégration des résultats au sein d'un SIEM, ouvrant ainsi la voie à une détection de fuites de données plus contextuelle et plus fiable.

2.4 SYSTÈMES DE GESTION DES ÉVÉNEMENTS DE SÉCURITÉ (SIEM)

Les systèmes de gestion des événements et informations de sécurité (SIEM) jouent un rôle crucial dans la surveillance des infrastructures informatiques modernes. Ils collectent, centralisent et analysent divers journaux afin d'identifier des comportements potentiellement malveillants (González-Granadillo et al., 2021),(Sashwin et al., 2024). Historiquement, les SIEM ont été conçus pour détecter des tentatives d'intrusion, des anomalies réseau ou des erreurs de configuration. Cependant, comme le souligne (Khraisat et al.,2019), leur capacité à détecter des fuites de données dépend fortement de la qualité des règles configurées et du contexte fourni par les événements. Dans ce mémoire, nous exploitons Wazuh, un SIEM open-source largement utilisé pour la surveillance, la gestion de conformité et la détection

d'intrusions. Toutefois, Wazuh comme la plupart des SIEM ne dispose d'aucun mécanisme intégré permettant de déterminer si les données exfiltrées sont réellement sensibles. Les SIEM se basent sur des logs, mais pas sur la provenance des données. Notre approche vise précisément à combler cette lacune en intégrant la Dynamic Taint Analysis dans le pipeline du SIEM.

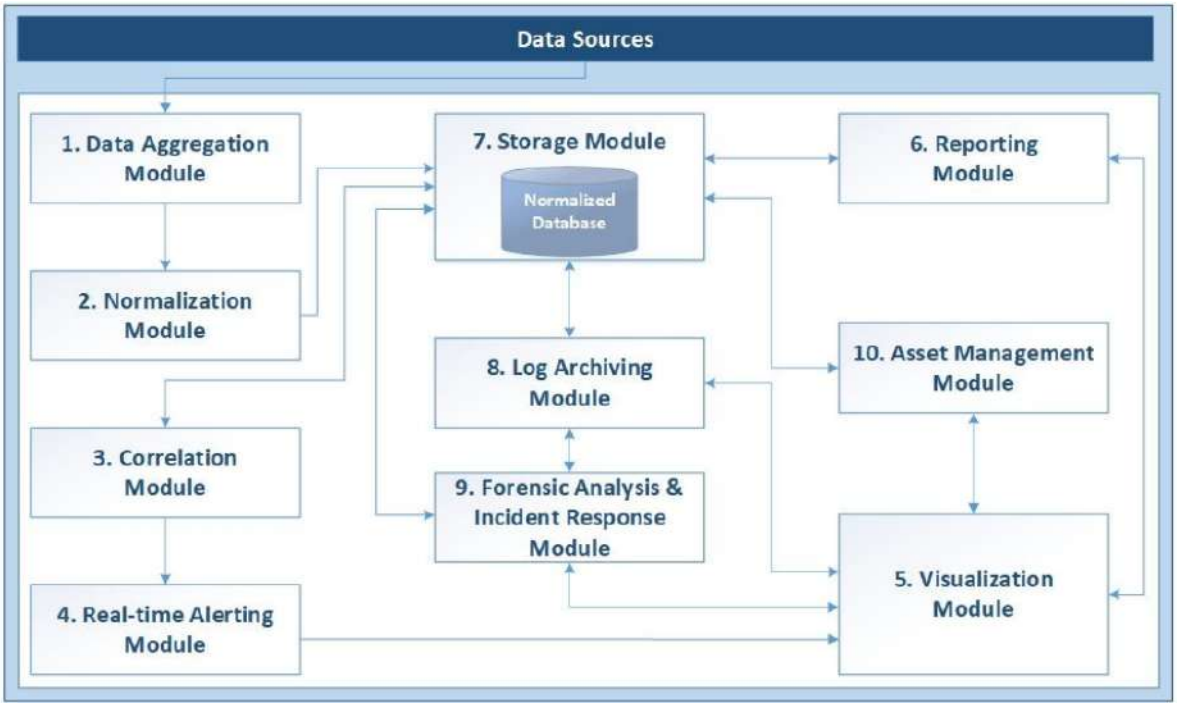

Figure 2.2 : Architecture du système SIEM (Sheeraz et al.)

2.5 GESTION DE LA PROVENANCE DES DONNÉES

La gestion de la provenance des données (data provenance) désigne l'ensemble des mécanismes permettant de capturer, structurer et exploiter des métadonnées décrivant l'origine d'une donnée, les transformations qu'elle a subies et les dépendances qui relient ses différentes versions au sein d'un système. L'objectif principal de la provenance est de fournir une traçabilité explicite des données afin de répondre à des questions telles que : d'où provient une valeur,

quelles opérations l'ont produite ou modifiée, dans quel contexte elle a été utilisée et quels processus ou utilisateurs y ont accédé. Les travaux fondateurs de (Buneman et al., 2001) ont établi des notions structurantes, notamment la provenance (pourquoi une donnée apparaît dans un résultat) et *where* (d'où provient cette donnée dans l'exécution d'une requête), qui constituent encore aujourd'hui des références pour la modélisation de la provenance dans les bases de données. Au-delà des bases de données, la provenance s'est imposée comme un concept transversal dans les systèmes distribués et les environnements de calcul complexes. Elle est utilisée pour améliorer la reproductibilité des traitements, faciliter l'audit et diagnostiquer des comportements inattendus dans les chaînes de traitement (Freire et al., 2008). Les synthèses récentes montrent que les approches de provenance peuvent être classées selon la granularité des informations capturées (niveau requête, tuple, attribut, valeur), leur mode de capture (instrumentation, journalisation, reconstruction) et leur finalité (explication, optimisation, conformité) (Glavic et Dittrich,2015). Cette catégorisation est particulièrement utile pour positionner notre approche, car la détection de fuites de données exige une provenance plus fine et orientée sécurité. Dans le contexte de la protection des données sensibles, la gestion de la provenance constitue un levier essentiel pour réduire l'ambiguïté des détections. Les approches traditionnelles de prévention des fuites (*Data Leakage Prevention*, DLP) s'appuient souvent sur l'analyse du contenu ou sur des règles statiques, ce qui conduit à de nombreux faux positifs dès qu'une valeur ressemble à un motif sensible sans en être réellement une. Les études de référence sur les systèmes DLP soulignent que l'absence de contexte et de traçabilité sur l'origine des données est l'une des causes majeures de ces faux positifs (Shabtai et al., Roman et al., 2012, 2018). En pratique, une même chaîne de caractères peut être sensible dans une base de production (ex. identifiant personnel réel) et non sensible dans une base de test, ou encore être générée artificiellement par une application. La provenance permet

de lever cette ambiguïté en attachant la valeur à son contexte d'apparition. Les bases de données relationnelles jouent un rôle central dans ces problématiques, puisqu'elles concentrent de grandes quantités de données personnelles et constituent un point de sortie naturel via des mécanismes d'export (ex. `"SELECT ... INTO OUTFILE"`) ou via des applications web interrogeant le SGBD. Néanmoins, les mécanismes de journalisation et de contrôle d'accès usuels ne permettent pas de suivre précisément les flux d'information, notamment au niveau des valeurs individuelles. Les travaux sur le contrôle de flux d'information (*Information Flow Control*) dans les bases relationnelles montrent l'importance de relier les opérations SQL à des politiques de confidentialité et à des mécanismes de suivi des données, afin d'éviter les divulgations indirectes (Liu et al., 2015). Ces travaux motivent l'usage d'une provenance riche et fine pour analyser la circulation des données dans des environnements SQL.

Dans ce mémoire, nous exploitons la provenance comme un composant opérationnel de détection, et non uniquement comme un mécanisme explicatif. La provenance est couplée à l'analyse dynamique de contamination afin de permettre un suivi dynamique des données sensibles. Le principe consiste à associer à chaque donnée sensible un marqueur enrichi contenant des informations telles que : la table et la colonne d'origine, un identifiant unique de la ligne (ou une clé logique), la session SQL ou l'entité ayant déclenché l'accès, ainsi que des informations temporelles permettant de relier la donnée à une séquence d'événements. L'intérêt de cette approche est de conserver, lors des transformations (jointures, projections, agrégations) et lors des sorties (export fichier, réponse HTTP), un lien traçable vers la source de sensibilité. Cette stratégie est cohérente avec les principes de la Dynamic Taint Analysis, qui vise à marquer des données d'entrée et à propager ces marqueurs au fil de l'exécution pour détecter les sorties non autorisées. Les travaux fondateurs de (Newsome et Song, 2005) montrent que l'amalyse dynamique de contamination permet d'identifier des flux d'information conduisant

à des expositions indésirables, tandis que (Schwartz et al., 2010) en propose une synthèse détaillée, incluant les forces et limites (sur-taint, sous-taint, granularité) des mécanismes de propagation. Dans notre cas, l'intégration de la provenance au marqueur de taint vise à augmenter la précision des détections en explicitant le contexte d'origine de chaque valeur et en facilitant l'interprétation des alertes. En somme, la gestion de la provenance des données permet de passer d'une détection fondée sur des motifs (contenu) à une détection fondée sur le contexte (origine). Elle constitue une brique fondamentale pour construire une détection de fuites de données plus fiable et mieux explicable, en particulier lorsqu'elle est combinée à une analyse dynamique des flux et à une exploitation des événements dans un SIEM.

CHAPITRE III

ÉTUDE DE L'ÉTAT DE L'ART

La croissance des volumes de données traitées par les systèmes d'information a considérablement accru les risques de fuite de données sensibles, qu'elles soient causées par des erreurs humaines ou des activités malveillantes. Cette situation nécessite des mécanismes capables de détecter les données sensibles et d'assurer leur suivi durant leur propagation.

Ce chapitre présente une revue des travaux existants sur la détection des fuites de données sensibles et l'analyse des flux d'information, tout en mettant en évidence leurs limites en matière de traçabilité et de réduction des faux positifs.

3.1 DÉTECTION FONDÉE SUR LES SIGNATURES ET LES EXPRESSIONS RÉGULIÈRES

Les approches fondées sur les signatures, en particulier les expressions régulières et le filtrage par mots-clés, constituent historiquement le socle des solutions commerciales de prévention des fuites de données (Data Loss Prevention, DLP (Brindha et Shaji, 2015),(Kaur et al.,2017). Leur principe repose sur l'identification de motifs syntaxiques prédéfinis correspondant à des formats réputés sensibles, tels que les numéros de cartes de crédit, les numéros d'assurance sociale

ou les identifiants bancaires. Ces mécanismes sont souvent complétés par des heuristiques simples, par exemple des règles de longueur, de préfixe ou de fréquence d'apparition. Ces approches présentent plusieurs avantages notables. Elles sont généralement peu coûteuses en termes de calcul, faciles à déployer et efficaces dans des environnements structurés où les données sensibles suivent des formats bien définis. Pour cette raison, elles sont largement adoptées dans les solutions industrielles de sécurité et intégrées dans de nombreux outils DLP pour la détection rapide de données sensibles dans les courriels, les fichiers ou les flux réseau.

Cependant, ces techniques souffrent d'une limitation fondamentale : elles analysent exclusivement la forme de surface des données. En l'absence de toute information contextuelle ou de provenance, elles sont incapables de distinguer une valeur réellement sensible d'un identifiant non-critique partageant le même format syntaxique. Cette limitation devient particulièrement problématique dans les environnements modernes, où les données peuvent être transformées, fragmentées ou encodées, rendant les motifs difficiles à reconnaître. De plus, les approches basées sur les expressions régulières sont de plus en plus considérées comme insuffisantes face à la complexité croissante des systèmes d'information. Des études récentes montrent que les solutions DLP traditionnelles fondées sur des règles statiques et des correspondances syntaxiques ne sont plus adaptées aux environnements distribués et aux architectures cloud, en raison du volume élevé de données et du taux important de faux positifs qu'elles génèrent. Cette incapacité, à capturer la dynamique réelle des flux de données limite leur efficacité dans des contextes opérationnels réalistes. Par ailleurs, la conception et la maintenance des expressions régulières représentent un défi non-négligeable. Des travaux récents ont montré que les développeurs rencontrent des difficultés importantes dans la création, la validation et l'évolution des expressions régulières, ce qui peut introduire des erreurs ou des comportements imprévus dans les systèmes de détection. (Michael et al., 2019). Ces limitations techniques

renforcent la fragilité des approches fondées exclusivement sur des règles syntaxiques. En conséquence, cette confusion conduit mécaniquement à un taux élevé de faux positifs, réduisant la valeur opérationnelle des alertes et augmentant la charge cognitive des analystes de sécurité. Ce phénomène est largement documenté dans la littérature, notamment dans l'étude de Meli et al. (Meli et al., 2019), qui met en évidence les limites structurelles des méthodes purement syntaxiques face à des jeux de données ambigus. Ces constats soulignent la nécessité de dépasser les approches fondées uniquement sur le contenu et d'intégrer des dimensions supplémentaires, telles que le contexte d'utilisation et la provenance des données, afin d'améliorer la précision et la pertinence des systèmes de détection de fuites de données.

3.2 EMPREINTES NUMÉRIQUES

Les techniques d'empreintes numériques visent à dépasser les limites des signatures exactes en construisant des représentations approximatives de documents ou de corpus confidentiels à l'aide de n-grammes, de k-skip-n-grammes ou d'empreintes floues (Shapira et al., 2013)(Shu et al., 2015). Ces méthodes permettent de détecter des contenus partiellement modifiés, réordonnés ou reformulés, ce qui les rend plus robustes face à certaines transformations superficielles. Le principe fondamental du fingerprinting consiste à extraire une signature compacte (empreinte) représentant les caractéristiques essentielles d'un contenu, puis à comparer cette signature avec celles de données sortantes afin de détecter des fuites potentielles. Contrairement aux approches fondées sur des expressions régulières, ces techniques sont capables de capturer des similarités approximatives, même lorsque les données ont subi des modifications légères. Des travaux récents montrent que les approches basées sur les n-grammes ou les séquences permettent de représenter efficacement des structures complexes

et de détecter des correspondances partielles dans des flux de données hétérogènes. Par ailleurs, des recherches récentes (Kayalvizhi et al., 2026) (Zheng et al.,2025) en cybersécurité confirment que les techniques de fingerprinting restent largement utilisées dans les systèmes de détection de fuites de données, notamment pour analyser des contenus textuels ou des flux réseau. Elles sont souvent intégrées dans des architectures DLP modernes afin d'améliorer la détection de contenus transformés ou obfusqués, ce qui constitue une amélioration significative par rapport aux approches purement syntaxiques. Toutefois, le fingerprinting reste intrinsèquement dépendant d'un jeu de données confidentiel préenregistré servant de référence. Il mesure une similarité entre contenus, sans être capable d'identifier l'origine effective d'une valeur. Ainsi, lorsque deux sources indépendantes contiennent légitimement des données identiques par exemple une base sensible et une base de test ou publique le détecteur peut reconnaître la valeur, mais il lui est impossible de déterminer laquelle des deux sources en est à l'origine. De plus, plusieurs travaux récents soulignent que les techniques de fingerprinting présentent des limites importantes face aux transformations avancées des données. En particulier, les modifications profondes du contenu, telles que la réécriture sémantique ou la génération automatique de données, peuvent réduire significativement la capacité de détection des empreintes, ce qui limite leur efficacité dans des environnements modernes et dynamiques. Cette incapacité, à raisonner sur la provenance constitue une limite majeure. En effet, le fingerprinting traite la sensibilité comme une propriété du contenu, et non comme une propriété de son origine. Par conséquent, il ne permet pas de distinguer deux occurrences identiques d'une même valeur issues de contextes différents. Cette limitation est particulièrement critique dans les environnements organisationnels où des données similaires peuvent coexister dans plusieurs bases avec des niveaux de sensibilité distincts. En résumé, bien que les techniques d'empreintes numériques offrent une meilleure robustesse que les approches basées sur des signatures exactes, elles

restent insuffisantes pour capturer la dimension contextuelle et la provenance des données. Cela justifie le besoin de méthodes complémentaires capables d'intégrer explicitement l'origine des données dans le processus de détection des fuites.

3.3 APPROCHES FONDÉES SUR L'APPRENTISSAGE AUTOMATIQUE

Les approches basées sur l'apprentissage automatique cherchent à classifier les données sensibles à partir de caractéristiques sémantiques ou contextuelles extraites du contenu (Alneyadi et al., 2015)(Lu et al., 2018). Dans des environnements riches en langage naturel, tels que des courriels ou des documents textuels, ces modèles peuvent exploiter des indices lexicaux, syntaxiques ou statistiques pour inférer la sensibilité d'une information. Avec l'évolution récente des techniques d'apprentissage profond, plusieurs travaux ont démontré l'efficacité des modèles de machine learning pour la détection de fuites de données et d'anomalies dans des systèmes complexes. Par exemple, des études récentes montrent que les approches basées sur le deep learning permettent d'identifier des comportements anormaux et des fuites potentielles dans des environnements distribués et à grande échelle (Syarova et al., 2024)(Yeboah et al., 2026). Ces modèles sont capables de traiter de grands volumes de données et d'extraire automatiquement des représentations pertinentes.

Cependant, dans de nombreux scénarios opérationnels, les fuites de données impliquent des identifiants isolés exportés via des requêtes SQL, des fichiers générés automatiquement, des paramètres HTTP ou des journaux applicatifs. Ces contextes offrent peu, voire aucun, signal linguistique exploitable par les modèles. En pratique, un même identifiant est présenté au modèle sous une forme identique, qu'il provienne d'une source sensible ou non. Par ailleurs, des travaux récents ont mis en évidence une limitation importante des approches

d'apprentissage automatique liée au phénomène de data leakage dans les modèles eux-mêmes. Ce phénomène se produit lorsque des informations non disponibles en conditions réelles sont utilisées lors de l'entraînement, ce qui conduit à des performances artificiellement élevées et à une mauvaise capacité de généralisation (Syarova et al., 2024)(Alturayeif et Hassine, 2025). Des études montrent que ce problème peut significativement biaiser les résultats expérimentaux et compromettre la fiabilité des modèles en production. De plus, plusieurs recherches soulignent que les modèles d'apprentissage automatique apprennent principalement des corrélations statistiques plutôt que des relations causales ou de provenance. Ainsi, bien qu'ils soient efficaces pour capturer des similarités sémantiques ou comportementales, ils ne sont pas capables d'inférer l'origine réelle des données. Cette limitation est particulièrement critique dans des environnements où des valeurs identiques peuvent exister dans plusieurs bases avec des niveaux de sensibilité différents. Enfin, bien que certaines approches hybrides combinent apprentissage automatique et détection d'anomalies pour améliorer les performances, elles nécessitent généralement de grandes quantités de données annotées et des ressources de calcul importantes (Alturayeif et Hassine, 2025)(Joeres et al., 2025). Cela limite leur déploiement dans des environnements réels, notamment dans des systèmes nécessitant une détection en temps réel. Ainsi, malgré leurs performances, les approches basées sur l'apprentissage automatique restent limitées par leur incapacité à intégrer explicitement la provenance des données. Cela justifie le besoin de méthodes complémentaires capables de relier chaque donnée à son origine afin d'améliorer la précision et la fiabilité des systèmes de détection de fuites.

3.4 ANALYSE STATIQUE

L'analyse statique par contamination examine le code source ou intermédiaire des applications afin d'identifier des chemins potentiels de propagation entre des sources et des puits sensibles (Xiaohui et al., 2014)(Yang et al., 2015). Cette approche est efficace pour détecter des fuites exprimables dans la structure interne d'un programme, notamment dans les graphes de contrôle et de flux de données. Les techniques modernes d'analyse statique, en particulier celles basées sur le static taint analysis, permettent d'analyser les flux d'information sans exécuter le programme. Elles reposent sur la construction de graphes de dépendances et l'analyse des relations entre variables afin d'identifier les chemins potentiels de propagation des données sensibles. Des travaux récents montrent que l'analyse statique constitue une technique fondamentale pour la détection de vulnérabilités dans les logiciels modernes, notamment dans les systèmes embarqués et les applications critiques. Par ailleurs, plusieurs études récentes proposent des améliorations significatives des techniques de taint analysis statique afin de mieux capturer les flux complexes introduits par les frameworks modernes. Par exemple, (Zhao et al., 2024) proposent une approche basée sur les relations sémantiques entre le code et les données pour améliorer la précision de la détection des vulnérabilités. De même, (Huang et al., 2026) introduisent le système SemTaint, qui permet une analyse plus scalable et précise en intégrant des modèles sémantiques des frameworks modernes Cependant, malgré ces avancées, plusieurs limitations importantes subsistent. Tout d'abord, l'analyse statique repose sur l'hypothèse d'une visibilité complète du code source. Or, dans les environnements réels, cette hypothèse est rarement vérifiée, notamment dans les architectures distribuées ou les systèmes intégrant des composants externes. Ensuite, des travaux récents soulignent l'existence d'un compromis fondamental entre précision et passage à l'échelle precision-scalability trade-off dans les

techniques d'analyse statique. Afin de couvrir l'ensemble des chemins d'exécution possibles, ces approches tendent à surapproximer les flux de données, ce qui entraîne un nombre important de faux positifs. Par ailleurs, l'analyse statique ne permet pas d'observer les flux de données qui se produisent en dehors des frontières de l'application analysée. Les exportations de bases de données, les écritures de fichiers, les interactions avec des services externes ou les journaux HTTP échappent largement à ce modèle. Cette limitation est particulièrement critique dans les environnements modernes où les données circulent entre plusieurs systèmes hétérogènes. De plus, des études récentes montrent que les techniques d'analyse statique peuvent souffrir d'imprécisions liées à la modélisation incomplète des frameworks et à la complexité des applications modernes, ce qui peut conduire à des phénomènes de sur-taint ou de sous-taint et réduire la fiabilité des résultats. Enfin, l'analyse statique ne dispose d'aucun mécanisme intrinsèque pour distinguer plusieurs instances d'exécution ou plusieurs bases de données contenant des valeurs identiques, en l'absence d'un modèle explicite de provenance à l'échelle des données concrètes. Elle raisonne sur des chemins potentiels dans le code, mais ne capture pas le contexte réel d'exécution ni l'origine effective des données. Ainsi, bien que l'analyse statique constitue un outil puissant pour la détection précoce de vulnérabilités, ses limitations en termes de contexte, de précision et de prise en compte de la provenance restreignent son efficacité dans les détections de fuites de données dans des environnements distribués.

3.5 ANALYSE DYNAMIQUE DE CONTAMINATION TRADITIONNELLE

L'analyse dynamique de contamination, également appelée suivi dynamique des flux d'information, consiste à surveiller l'exécution d'un programme afin de suivre la propagation de données marquées comme sensibles (Alashjee et al.,2019 (Ahmed et al., 2023) (Zhu et al.,

2011). Les valeurs introduites depuis des sources considérées comme sensibles ou non-fiables, sont marquées comme contaminées, puis ces marqueurs sont propagés au fil de l'exécution des instructions, à travers les registres et la mémoire. L'analyse dynamique de comtamination est aujourd'hui largement utilisée dans le domaine de la sécurité logicielle pour détecter des vulnérabilités telles que les injections SQL, les attaques XSS ou les fuites de données. Elle permet de suivre précisément les flux d'information à l'exécution et d'identifier les chemins menant à des points sensibles (sinks). Des travaux récents, tels que ViaLin, ont montré que les approches modernes de DTA peuvent reconstruire des chemins de propagation précis tout en réduisant le coût en performance (Ahmed et al., 2023). Par ailleurs, des recherches récentes (Wu et Bartel 2025, Chew et al. 2025**?** , Alecci et al. 2025) ont étendu l'analyse dynamique à des systèmes complexes et distribués. Par exemple, (Cela et al., 2024) proposent une approche d'analyse dynamique de bout en bout appliquée à des applications réelles comme WhatsApp, permettant de suivre les flux de données à travers plusieurs composants client et serveur (Cela et al., 2024). Ces travaux montrent l'intérêt croissant de la DTA pour l'analyse des systèmes modernes. Cependant, malgré ces avancées, plusieurs limitations importantes subsistent. Tout d'abord, la DTA introduit un coût d'exécution significatif en raison de l'instrumentation nécessaire pour suivre les flux de données à un niveau fin. Des travaux récents montrent que même les approches optimisées restent coûteuses en termes de performance, ce qui limite leur utilisation en production. Ensuite, la l'analyse dynamique contamination repose généralement sur une sémantique binaire de contamination (contaminé / non contaminé) et ne conserve pas d'information détaillée sur l'origine précise des données. Bien que certaines approches récentes tentent d'améliorer la précision du suivi en introduisant des mécanismes de propagation plus fins, elles restent limitées dans leur capacité à représenter explicitement la provenance des données. De plus, les systèmes classiques d'analyse dynamique opèrent à l'intérieur d'un

programme instrumenté et ne sont pas conçus pour suivre des flux traversant plusieurs systèmes hétérogènes, tels que des bases de données, des systèmes de fichiers, des serveurs web ou des plateformes distribuées. Cette limitation est particulièrement critique dans les environnements modernes où les données circulent entre plusieurs composants indépendants.

Enfin, des travaux récents ont également montré que les mécanismes de l'analyse dynamique contamination peuvent être contournés ou perturbés par certaines vulnérabilités, notamment via des attaques ciblant les métadonnées de taint, ce qui compromet la fiabilité des résultats (Wu et Bartel,2025). Ainsi, bien que l'analyse dynamique de contamination constitue une technique puissante pour suivre les flux d'information à l'exécution, ses limitations en termes de performance, de scalabilité et d'absence de provenance fine restreignent son efficacité dans les environnements distribués et orientés données. Ces limites motivent le besoin de nouvelles approches capables de capturer explicitement l'origine des données à travers plusieurs systèmes.

3.6 DISCUSSION

Le Tableau 3.1 synthétise les limitations structurelles communes aux principales approches de prévention des fuites de données, qu'elles soient fondées sur l'analyse du contenu ou centrées sur l'exécution des programmes. Bien que ces techniques diffèrent par leurs fondements méthodologiques allant des signatures syntaxiques aux modèles statistiques, sémantiques ou comportementaux elles partagent une hypothèse implicite forte : la sensibilité d'une information peut être déduite à partir des propriétés intrinsèques de la donnée observée. En pratique, ces approches exploitent des caractéristiques telles que la forme syntaxique d'une valeur, sa similarité avec des contenus préenregistrés, ou encore les indices sémantiques présents dans son environnement immédiat. Or, plusieurs travaux ont montré que ces propriétés sont

insuffisantes pour caractériser de manière fiable la nature sensible d'une information dans des systèmes complexes et dynamiques. En particulier, les approches basées sur le contenu ignorent un facteur déterminant, à savoir l'de la donnée et les transformations successives qu'elle subit au cours de son cycle de vie (Shabtai et al., 2012)(Shabtai et al., 2012). L'absence de prise en compte de la provenance empêche de distinguer des valeurs identiques provenant de sources différentes, ce qui entraîne de nombreux faux positifs dans les systèmes DLP traditionnels (Meli et al., 2019). Les travaux sur l'analyse dynamique des flux d'information montrent également que l'absence de provenance limite l'interprétation correcte des flux détectés.

Par ailleurs, ces approches se révèlent particulièrement vulnérables aux transformations intentionnelles ou accidentelles des données. Des travaux récents en Dynamic Taint Analysis montrent que les phénomènes de sur-taint et de sous-taint peuvent altérer la précision des analyses, en propageant incorrectement ou en omettant certaines informations sensibles. De même, les transformations telles que l'encodage, la fragmentation ou la réécriture des données rendent les méthodes fondées sur le contenu encore plus fragiles. En outre, dans les environnements modernes distribués, les données circulent entre plusieurs systèmes hétérogènes (bases de données, applications web, systèmes de fichiers, services cloud). Or, comme le montrent plusieurs travaux récents, les approches classiques d'analyse (statique ou dynamique) sont généralement limitées à un périmètre applicatif restreint et ne permettent pas de reconstruire des flux inter-systèmes complets :contentReference[oaicite :2]index=2. Cette limitation empêche une compréhension globale du cycle de vie des données. Ces constats mettent en évidence une limite conceptuelle majeure des approches existantes : tant que la sensibilité est traitée comme une propriété du contenu plutôt que comme une propriété de la provenance, les mécanismes de détection resteront intrinsèquement ambigus. Cette observation est cohérente avec les travaux récents sur les graphes de provenance, qui montrent que la

traçabilité des transformations est essentielle pour interpréter correctement les flux de données (Freire et al., 2008)(Glavic et Dittrich 2007). Cette observation constitue le point de départ de notre approche, qui vise à replacer la provenance des données au cœur du processus de détection des fuites, afin de permettre une distinction fiable entre données sensibles et non sensibles dans des environnements hétérogènes et dynamiques.

En réponse à ces limites, notre travail s'intéresse à l'utilisation de l'analyse dynamique de contamination comme mécanisme de suivi explicite de la provenance des données sensibles dans des environnements distribués et hétérogènes. Contrairement aux approches traditionnelles centrées uniquement sur le contenu observé, notre approche cherche à relier chaque donnée détectée à son origine effective, en exploitant les traces produites par différents composants du système, notamment les bases de données, les fichiers exportés, les journaux applicatifs et les échanges HTTP. Cette orientation soulève plusieurs enjeux scientifiques et techniques. Tout d'abord, il devient nécessaire de déterminer comment une approche de l'analyse dynamique de contamination peut être adaptée afin d'assurer un suivi fiable de la provenance des données à travers plusieurs systèmes interconnectés. Ensuite, l'intégration explicite de la provenance doit permettre d'évaluer dans quelle mesure il est possible de réduire les faux positifs générés par les approches classiques de détection fondées sur les signatures, le machine learning ou les SIEM traditionnels. Par ailleurs, l'intégration d'un mécanisme de l'analyse dynamique de contamination dans un SIEM tel que Wazuh soulève la question de la génération d'alertes contextualisées, interprétables et directement exploitables par les analystes de sécurité. Enfin, l'efficacité de l'approche doit être étudiée dans des scénarios de détection en temps réel, notamment lors de l'exfiltration de données sensibles vers des fichiers externes ou des canaux réseau comme HTTP. Ces questions constituent le fondement de notre démarche de recherche et motivent le développement de l'approche proposée dans les chapitres suivants.

Tableau 3.1 : Comparaison des approches de détection de fuites de données selon la prise en compte de la provenance

Approche	Basée sur le contenu	Sensible à la provenance	Limitation principale
Expressions régulières / Signatures	Oui	Non	Incapable de distinguer des valeurs syntaxiquement identiques provenant de sources ayant des niveaux de sensibilité différents.
Empreintes numériques (Fingerprinting)	Oui	Non	Repose uniquement sur la similarité du contenu ; échoue lorsque plusieurs sources contiennent des valeurs identiques.
Apprentissage automatique	Oui (sémantique)	Non	Incapable de classifier correctement des identifiants isolés ou d'inférer l'origine réelle des données ; absence de provenance explicite.
Analyse statique	Partiellement	Non	Ne permet pas d'observer les flux réels (exports, journaux, HTTP) et ne capture pas le contexte réel d'exécution.
Analyse dynamique de contamination traditionnelle	Partiellement (niveau binaire)	Non	Suit la propagation des données sans conserver une provenance fine ; limité aux systèmes instrumentés.
Notre approche (fondée sur la provenance)	Partiellement	**Oui**	Suit dynamiquement la propagation des données tout en conservant leur provenance à travers des systèmes hétérogènes, permettant de distinguer des valeurs identiques selon leur origine.

Le Tablea u 3.1 met en évidence les différences fondamentales entre les principales approches de détection de fuites de données selon leur capacité à prendre en compte la provenance des informations. Les approches fondées sur les expressions régulières, les signatures et les empreintes numériques reposent principalement sur l'analyse du contenu observé. Bien qu'elles soient efficaces pour identifier des motifs ou des similarités, elles demeurent incapables de distinguer des valeurs identiques provenant de sources différentes, ce qui entraîne un nombre important de faux positifs dans les environnements réels. Les approches basées sur l'apprentissage automatique exploitent des caractéristiques sémantiques ou statistiques afin d'améliorer la détection. Toutefois, elles restent dépendantes des propriétés intrinsèques des données et ne disposent pas d'un mécanisme explicite permettant d'identifier l'origine réelle des informations traitées. Elles rencontrent ainsi des difficultés lorsqu'un même identifiant apparaît dans plusieurs contextes avec des niveaux de sensibilité différents. L'analyse statique et l'analyse dynamique de contamination permettent quant à elles d'observer les flux d'information

à l'intérieur des programmes. Cependant, l'analyse statique demeure limitée à une vision théorique des chemins d'exécution et ne permet pas de capturer les flux réels produits lors de l'exécution du système. De son côté, l'analyse dynamique de contamination traditionnelle suit la propagation des données sensibles à l'exécution, mais conserve généralement une vision binaire de la contamination sans intégrer une provenance détaillée des données à travers plusieurs systèmes hétérogènes. À l'inverse, notre approche repose sur une analyse dynamique de contamination fondée sur la provenance. Elle permet non seulement de suivre les flux de données sensibles, mais également de conserver l'origine exacte des informations tout au long de leur propagation entre différents composants du système, notamment les bases de données, les fichiers exportés et les échanges réseau. Cette prise en compte explicite de la provenance permet de distinguer des valeurs identiques selon leur origine et contribue ainsi à réduire significativement les faux positifs générés par les approches traditionnelles.

CHAPITRE IV

MÉTHODOLOGIE

Ce chapitre présente la méthodologie adoptée pour la conception et l'implémentation du système proposé dans le cadre de ce mémoire. À partir des limites identifiées dans les travaux connexes, notamment, l'absence de prise en compte explicite de la provenance dans les approches traditionnelles de prévention des fuites de données, notre démarche vise à développer un mécanisme de détection capable de suivre dynamiquement l'origine et la propagation des données sensibles dans des environnements hétérogènes. L'objectif de cette méthodologie est de concevoir une approche opérationnelle permettant de détecter les exfiltrations de données sensibles tout en réduisant les faux positifs générés par les méthodes fondées uniquement sur le contenu. Pour atteindre cet objectif, le système proposé combine des mécanismes d'analyse dynamique de contamination des techniques de corrélation de journaux et une intégration avec le SIEM Wazuh afin de produire des alertes contextualisées et exploitables en temps réel. Ce chapitre décrit ainsi l'architecture générale du système, les mécanismes de gestion de la provenance, la construction du *taint_pool*, les méthodes de surveillance des exportations SQL et des communications HTTP, ainsi que le processus de génération et d'intégration des alertes dans l'infrastructure SIEM. Il présente également les choix de conception retenus afin d'assurer une approche non-intrusive, compatible avec des environnements réels de production.

4.1 DÉVELOPPEMENT DU SYSTÈME D'ANALYSE DYNAMIQUE DE CONTAMINATION FONDÉ SUR LA PROVENANCE

Dans cette section, nous détaillons les objectifs ainsi que les fondements de l'implémentation du système proposé. Nous y présentons les principes de conception adoptés et les motivations qui ont guidé le développement de l'approche basée sur la provenance.

4.1.1 OBJECTIFS ET FONDEMENTS DE L'IMPLÉMENTATION

L'implémentation présentée dans ce mémoire a pour objectif de matérialiser un système opérationnel de l'analyse dynamique de contamination fondée sur la provenance, conçu pour la détection fiable des fuites de données sensibles dans des environnements informatiques modernes. Les systèmes actuels de prévention des fuites de données reposent majoritairement sur des techniques d'analyse du contenu, telles que les expressions régulières, le fingerprinting ou l'apprentissage automatique. Ces approches traitent implicitement la sensibilité comme une propriété intrinsèque de la donnée, indépendamment de son origine réelle. Or, comme démontré dans les chapitres précédents, cette hypothèse conduit à des ambiguïtés fondamentales lorsque des valeurs identiques circulent simultanément dans des contextes sensibles et non sensibles. L'approche retenue dans cette implémentation repose sur le principe selon lequel la sensibilité d'une donnée ne peut être déterminée de manière fiable qu'en tenant compte de sa provenance, c'est-à-dire de la source exacte à partir de laquelle elle a été produite, ainsi que du chemin qu'elle emprunte avant de quitter le système. L'objectif principal est donc de suivre dynamiquement la propagation de valeurs concrètes issues de bases de données déclarées sensibles et de détecter leur exfiltration effective vers des canaux de sortie tels

que des fichiers ou des transmissions réseau. Un objectif complémentaire consiste à réduire significativement les faux positifs, afin d'améliorer la valeur opérationnelle des alertes produites par le système. Un autre fondement essentiel de cette implémentation est son caractère non intrusif. Contrairement aux systèmes classiques d'analyse dynamique de contamination, qui nécessitent une instrumentation fine du code source ou du binaire, le système proposé observe exclusivement des canaux déjà disponibles dans l'environnement, notamment les journaux de requêtes SQL et les journaux HTTP. Ce choix permet une intégration réaliste dans des environnements de production existants. Enfin, l'intégration avec le SIEM Wazuh inscrit cette implémentation dans une chaîne complète de supervision de sécurité, allant de la détection à la visualisation et à la corrélation des incidents.

4.1.2 ARCHITECTURE GÉNÉRALE ET POSITIONNEMENT DU MOTEUR DU SYSTÈME

L'architecture du système repose sur un moteur d'analyse développé en Python, conçu comme un composant externe d'observation et de corrélation. Ce moteur n'intervient pas directement dans l'exécution des applications ni dans le fonctionnement interne du serveur de bases de données, mais se positionne comme un observateur privilégié des flux générés par ces composants. Il interagit avec plusieurs sources d'information hétérogènes, notamment les bases de données MySQL, les journaux de requêtes SQL, les journaux de serveurs web Apache et l'infrastructure SIEM Wazuh. Le rôle du moteur du système d'analyse de contamination est de collecter, corréler et interpréter ces informations afin de reconstruire des flux de données logiques depuis leur origine jusqu'à leur point de sortie. Cette architecture découplée permet de déléguer au SIEM les tâches de stockage, d'indexation et de visualisation, tandis que la

logique de détection fondée sur la provenance est réalisée en amont. Ce découplage améliore la modularité du système et facilite son extension à d'autres canaux d'exfiltration ou à d'autres infrastructures sans remettre en cause l'architecture globale. Figure

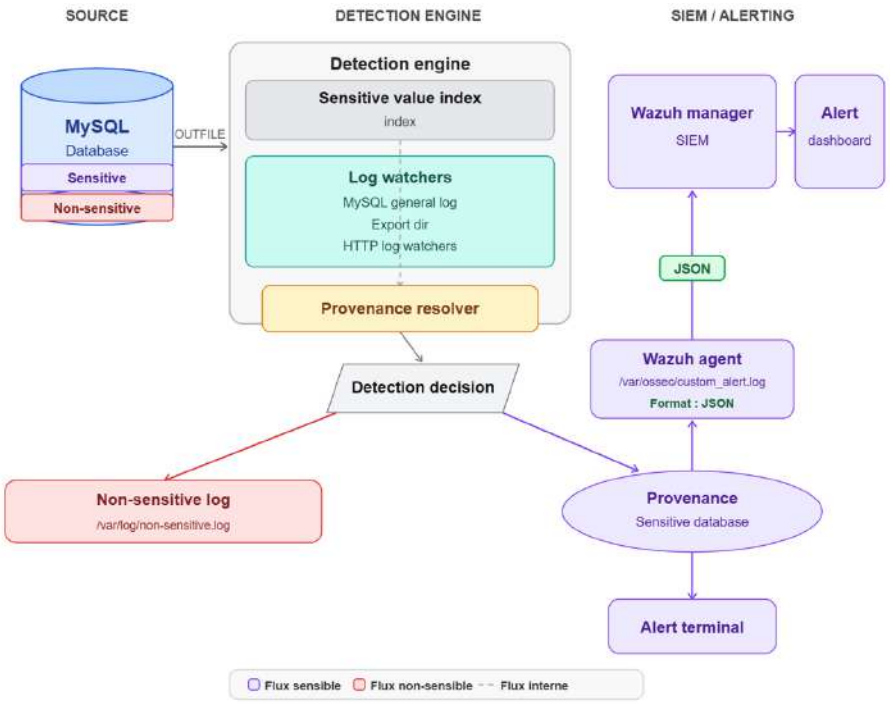

Figure 4.1 : Architecture du l'analyse dynamique de contamination sur la provenance, intégrée au SIEM Wazuh

4.1.3 DÉCLARATION EXPLICITE DES BASES DE DONNÉES ET RÔLE DE LA CONFIGURATION

La table de configuration dta_config constitue un pilier central de l'implémentation, car elle formalise explicitement la notion de sensibilité au niveau des sources de données. Cette table permet de déclarer quelles bases de données doivent être considérées comme sensibles et lesquelles ne le sont pas, indépendamment du contenu qu'elles hébergent. Ce choix de conception répond directement à un problème largement documenté dans la littérature :

des valeurs textuellement identiques peuvent correspondre à des informations hautement confidentielles dans un contexte donné, et à des données parfaitement bénignes dans un autre. En s'appuyant sur une table de configuration explicite, le système évite toute inférence implicite ou heuristique sur la sensibilité des données. La décision est externalisée et contrôlée par l'administrateur, ce qui améliore la transparence et la robustesse du système. Cette approche facilite également l'évolution du périmètre de surveillance, puisqu'il suffit d'ajouter ou de modifier une entrée dans dta_config pour prendre en compte une nouvelle base sensibl6e. Ainsi, cette table joue à la fois un rôle fonctionnel pour le moteur et un rôle explicatif pour la compréhension globale du système.

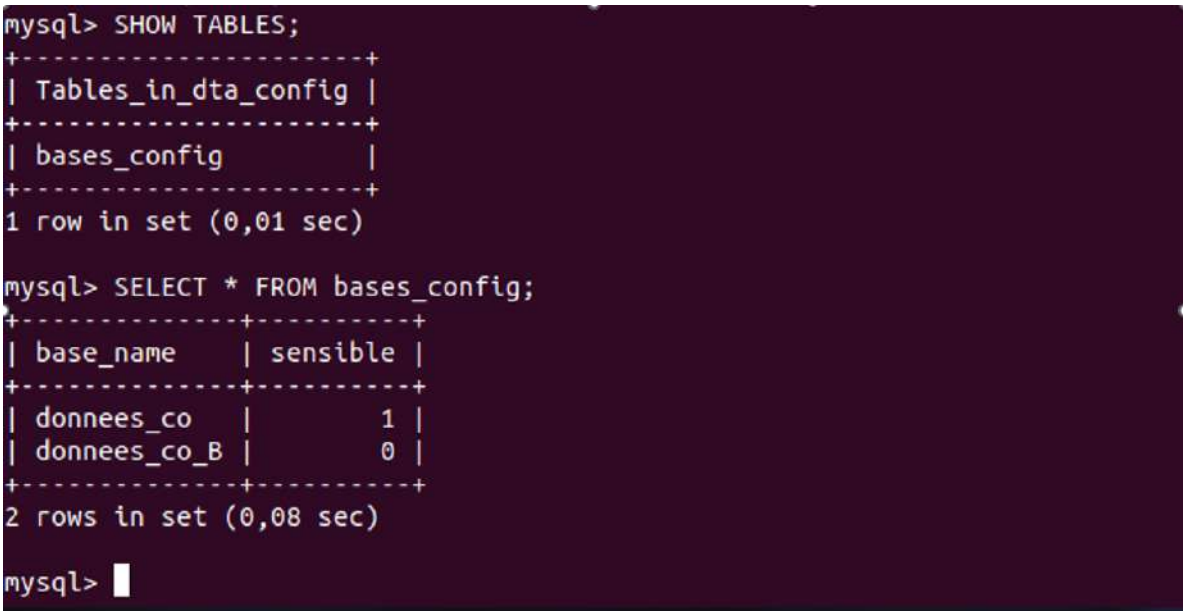

Figure 4.2 : Déclaration des bases de données sensibles et non sensibles

Cette figure 4.2 présente le contenu de la table `bases_config`, utilisée pour la configuration du système de détection. Cette table permet de définir explicitement le niveau de sensibilité des bases de données surveillées. On observe que la base `donnees_co` est marquée comme sensible (valeur 1), tandis que la base `donnees_co_B` est considérée comme non sensible

(valeur 0). Cette distinction constitue un élément fondamental du système, car elle permet au moteur d'analyse de la contamination de déterminer la provenance des données et d'adapter le processus de détection en conséquence.

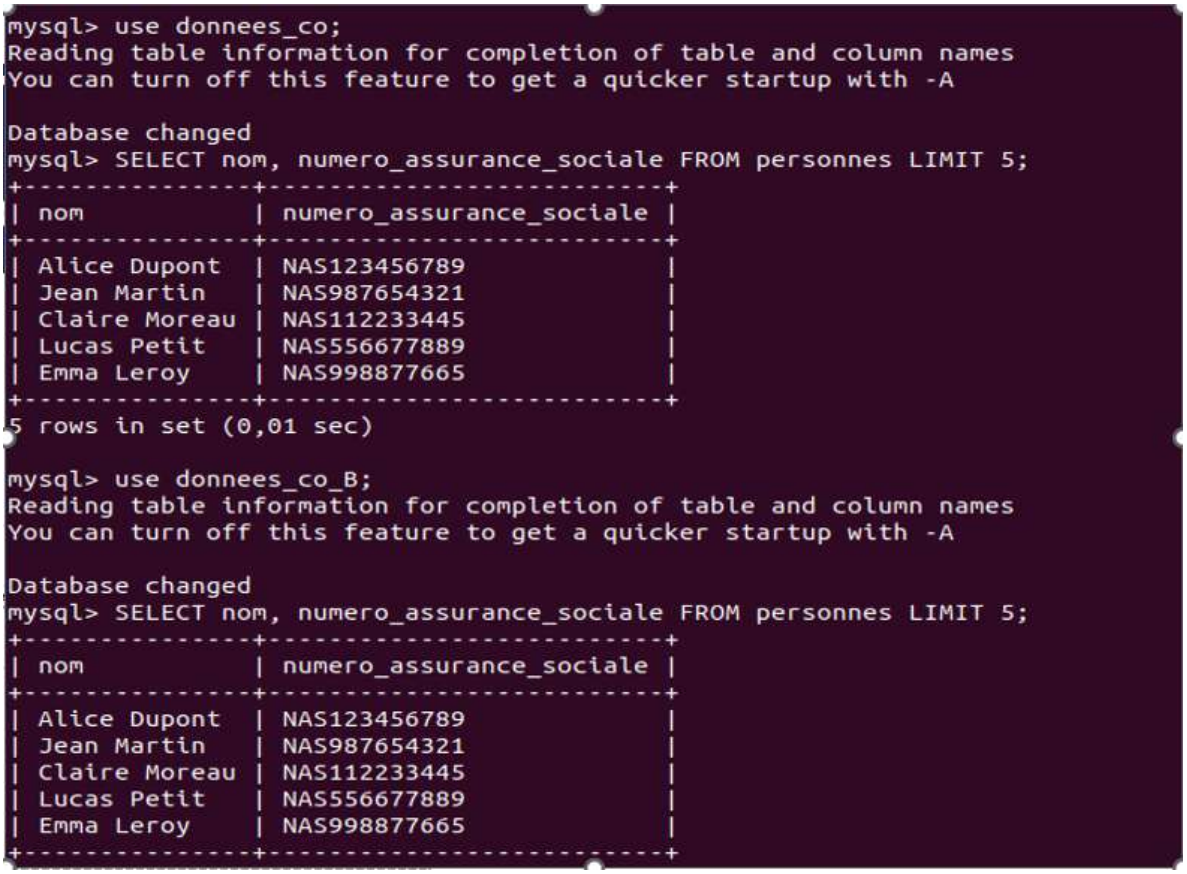

Figure 4.3 : Duplication de données identiques dans les deux bases de données distinctes

La figure 4.3 illustre l'accès aux données dans les bases `donnees_co` et `donnees_co_B` à travers des requêtes SQL de type SELECT. On observe que les deux bases contiennent des données identiques, notamment des noms et des numéros d'assurance sociale (NAS). Toutefois, elles diffèrent par leur niveau de sensibilité, la base `donnees_co` étant déclarée sensible, tandis que `donnees_co_B` est considérée comme non sensible. Cette duplication met en évidence une problématique centrale des approches classiques de détection des fuites de données, qui

ne peuvent pas distinguer des valeurs identiques selon leur contexte d'origine. Cette ambiguïté est levée dans notre approche grâce à la prise en compte de la provenance des données.

4.1.4 CHARGEMENT DES DONNÉES SENSIBLES ET CONSTRUCTION DU TAINT_POOL

Le moteur de l'analyse dynamiquue de comtamination initialise son fonctionnement en chargeant l'ensemble des valeurs sensibles issues des bases déclarées dans dta_config. Ces valeurs sont stockées en mémoire dans une structure appelée taint_pool, qui associe chaque valeur concrète à une étiquette de provenance indiquant précisément sa base d'origine. Cette structure constitue le référentiel central permettant au moteur de déterminer si une valeur observée lors d'une exportation correspond à une donnée sensible.

Contrairement aux approches traditionnelles, qui propagent des marqueurs de contamination au niveau des octets ou des registres mémoire, cette implémentation opère à un niveau plus abstrait, en suivant des valeurs logiques telles qu'elles apparaissent dans les bases de données et les flux applicatifs. Ce choix permet de suivre des données à travers des systèmes hétérogènes sans instrumentation lourde. Afin de maintenir la cohérence du taint_pool, un mécanisme de rafraîchissement périodique est mis en œuvre, permettant d'intégrer les nouvelles valeurs sensibles ajoutées aux bases surveillées et d'éviter les omissions susceptibles

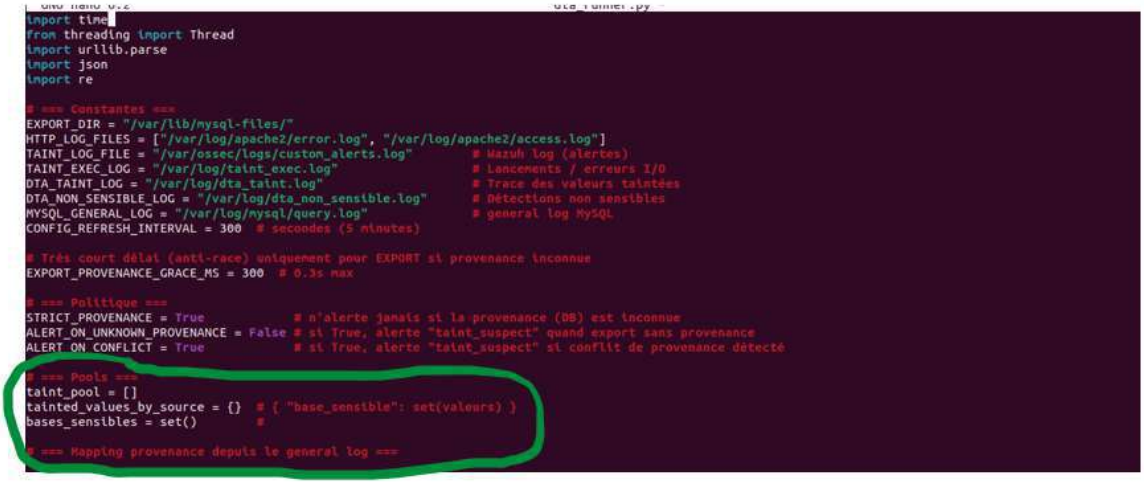

Figure 4.4 : Construction du TAINT_POOL

La figure 4.5 présente l'initialisation des structures de données utilisées dans notre module de suivi de contamination. On y observe notamment la mise en place des taint pools, qui associent les valeurs sensibles à leur source d'origine, ainsi que les dictionnaires de suivi par base sensible. Ces structures permettent de conserver une trace de la provenance des données et de faciliter la corrélation entre les événements observés dans les journaux, tout en supportant la détection de conflits ou d'ambiguïtés liées à des valeurs identiques issues de sources différentes.

```python
# === Chargement des données sensibles ===
def charger_donnees_sensibles(base, user="Amadou", password="Admin@2025", table="personnes"):
    base_norm = str(base).strip().lower()
    if base_norm not in bases_sensibles:
        print(f"{base} ignorée (non marquée sensible)")
        return
    try:
        conn = mysql.connector.connect(host="localhost", user=user, password=password, database=base)
        cursor = conn.cursor()
        cursor.execute(f"SELECT nom, numero_assurance_sociale FROM {table}")
        for nom, nas in cursor.fetchall():
            TaintedData(nom, label="nom", source_bdd=base_norm)
            TaintedData(nas, label="nas", source_bdd=base_norm)
        cursor.close()
        conn.close()
        print(f"Données chargées depuis {base_norm}")
    except Exception as e:
        print(f"Erreur chargement {base} : {e}")
```

Figure 4.5 : Chargement des données par le Taint_Pool

La figure 4.5 illustre la mis en place de la fonction charger_donnees_sensibles, qui constitue le point d'entrée du mécanisme de marquage (tainting) dans notre système de détection. Cette fonction se connecte à une base de données MySQL préalablement identifiée comme sensible (vérification via bases_sensibles), puis extrait les attributs critiques — en l'occurrence le nom et le numéro d'assurance sociale (NAS) depuis la table personnes. Chaque valeur récupérée est immédiatement encapsulée dans un objet "TaintedData", accompagné d'un label sémantique ("nom", "nas") et de l'origine de la donnée (source_bdd). C'est cette encapsulation qui alimente le Taint_Pool et permet, en aval, de tracer la propagation des informations sensibles à travers les flux applicatifs.

4.1.5 SURVEILLANCE DES EXPORTATIONS SQL ET GESTION TEMPORELLE

La détection des exportations SQL repose sur l'analyse continue du MySQL General Query Log. Lorsqu'une requête de type "`SELECT ... INTO OUTFILE`" est détectée, le moteur enregistre le contexte d'exécution, notamment la base active et l'horodatage de la requête. Cette information est stockée dans une structure interne qui associe temporairement un fichier exporté à sa base d'origine, permettant ainsi d'attribuer correctement la provenance des données lors de l'analyse du contenu du fichier.

La gestion des délais et de la concurrence constitue un aspect critique de l'implémentation. Dans un environnement réel, plusieurs requêtes peuvent être exécutées simultanément et la création effective des fichiers peut être légèrement différée. Pour pallier ces contraintes, le moteur implémente un mécanisme de synchronisation temporelle reposant sur des fenêtres d'attente et des tentatives successives de résolution. Chaque association est assortie d'une durée de vie limitée afin de maîtriser l'usage mémoire et d'éviter des attributions erronées

sur des événements anciens. Cette logique permet de réduire significativement les cas de provenance inconnue.

```python
# === Surveillance des fichiers exports ===
def surveiller_exports():
    print(f"Surveillance export : {EXPORT_DIR}")
    already_checked = {}
    while True:
        try:
            fichiers = os.listdir(EXPORT_DIR)
        except Exception:
            fichiers = []
        # purge mappings expirés
        now = datetime.now()
        for k, (dbp, ts) in list(outfile_provenance.items()):
            if now - ts > OUTFILE_PROV_TTL:
                del outfile_provenance[k]

        for fichier in fichiers:
            chemin = os.path.join(EXPORT_DIR, fichier)
            if not os.path.isfile(chemin):
                continue
            try:
                with open(chemin, "r") as f:
                    f.seek(already_checked.get(chemin, 0))
                    lignes = f.readlines()
                    already_checked[chemin] = f.tell()
            except Exception:
                continue
            for ligne in lignes:
                detecter_fuite(ligne.strip(), "EXPORT", fichier)
        time.sleep(2)
```

Figure 4.6 : Analyse continue des fichiers d'export de données

La fonction surveiller_exports, chargée de la surveillance continue du répertoire d'export (EXPORT_DIR) afin de détecter toute fuite de données sensibles. Implémentée sous forme de boucle infinie, cette fonction parcourt périodiquement les fichiers présents dans le répertoire cible et analyse uniquement les nouvelles lignes ajoutées depuis la dernière inspection, grâce au dictionnaire already_checked qui mémorise la position de lecture (offset) de chaque fichier. Chaque ligne nouvellement écrite est ensuite transmise à la fonction detecter_fuite, qui confronte son contenu au Taint Pool pour identifier la présence éventuelle de données marquées. Un mécanisme de purge des mappings de provenance expirés (OUTFILE_PROV_TTL) est également intégré afin de maîtriser l'empreinte mémoire du système sur la durée.

```
# === Surveillance HTTP ===
def surveiller_http_logs():
    for log_file in HTTP_LOG_FILES:
        Thread(target=surveiller_http_log, args=(log_file,), daemon=True).start()

def surveiller_http_log(log_file):
    print(f"Surveillance HTTP : {log_file}")
    if not os.path.exists(log_file):
        print(f"{log_file} introuvable")
        return
    with open(log_file, "r") as f:
        f.seek(0, os.SEEK_END)
        while True:
            ligne = f.readline()
            if not ligne:
                time.sleep(1)
                continue
            ligne_decodee = urllib.parse.unquote(ligne.strip())
            detecter_fuite(ligne_decodee, "HTTP", log_file)
```

Figure 4.7 : Concéption de la surveillance des accès HTTP

Dans cotre conception du module surveillance des accès HTTP, troisième pilier de notre dispositif de détection après le chargement du Taint_Pool et la surveillance des exports. Ce module se distingue par trois caractéristiques techniques notables. Premièrement, l'architecture adoptée repose sur un modèle multi-thread. La fonction surveiller_http_logs crée un thread dédié pour chaque fichier de log déclaré dans HTTP_LOG_FILES, ce qui permet la surveillance simultanée de plusieurs sources (par exemple le journal d'accès Apache, celui de Nginx, ou des logs applicatifs spécifiques) sans qu'un fichier volumineux ne bloque l'analyse des autres. L'attribut "daemon=True" garantit par ailleurs que ces threads s'arrêtent automatiquement à la fermeture du processus principal, évitant ainsi tout processus orphelin. Deuxièmement, la fonction surveiller_http_log met en œuvre un mécanisme de lecture en flux continu (tail-like). L'instruction f.seek(0, os.SEEK_END) positionne le curseur de lecture en fin de fichier dès l'ouverture : seules les requêtes HTTP nouvellement enregistrées sont donc analysées, ce qui évite de retraiter l'historique du journal. La boucle infinie associée à time.sleep(1) en cas de fichier vide assure une réactivité quasi instantanée tout en limitant la charge processeur. Une vérification préalable de l'existence du fichier (os.path.exists) prévient par ailleurs les erreurs liées à des journaux non disponibles au démarrage. Troisièmement, l'élément le plus significatif

sur le plan sécuritaire chaque ligne est soumise à un décodage URL via urllib.parse.unquote avant d'être analysée. Cette étape est cruciale : les requêtes HTTP encodent fréquemment les caractères spéciaux selon la norme percent-encoding. Sans ce décodage, un attaquant pourrait aisément contourner la détection en encodant les données sensibles exfiltrées dans l'URL ou le corps de la requête. La normalisation préalable garantit ainsi que la confrontation au Taint Pool s'effectue sur la valeur sémantique réelle et non sur sa représentation transportée. Enfin, l'appel detecter_fuite(ligne_decodee, "HTTP", log_file) boucle l'analyse en signalant explicitement le canal HTTP comme sink, ce qui permet, en cas d'alerte, de distinguer une fuite par exfiltration réseau d'une fuite par fichier exporté localement et d'adapter la réponse en conséquence

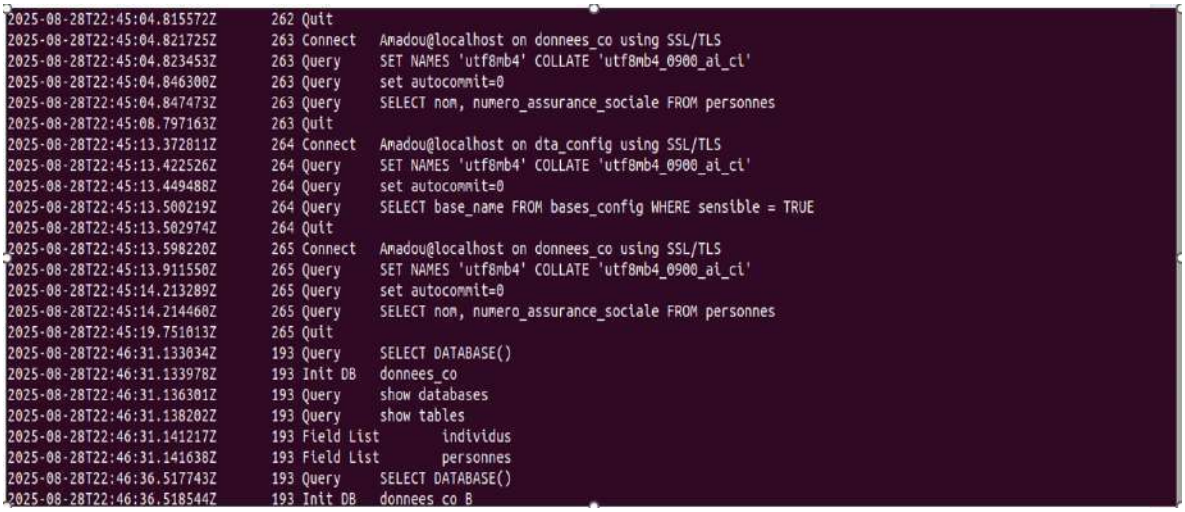

Figure 4.8 : MySQL General Query Log

4.1.6 ANALYSE DES EXFILTRATIONS VIA HTTP ET RÉSOLUTION DES AMBIGUÏTÉS

En complément des exportations vers des fichiers, le moteur d'analyse dynamique de contamination surveille les exfiltrations réseau en analysant les journaux Apache. Les requêtes HTTP sont reconstruites afin d'identifier les paramètres pertinents, notamment ceux indiquant la base

de données interrogée. Lorsqu'une valeur présente dans le taint_pool est détectée dans une réponse HTTP, le moteur tente d'en déduire la provenance à partir des informations disponibles dans la requête et des journaux.

Les informations fournies par les journaux réseau peuvent toutefois être incomplètes ou ambiguës. Dans ces situations, plusieurs méthodes de résolution peuvent conduire à des résultats divergents. Le moteur détecte alors explicitement un conflit de provenance et marque l'événement comme suspect. Ce mécanisme permet de conserver une trace des situations ambiguës sans produire systématiquement des alertes critiques, offrant ainsi un compromis entre prudence analytique et réduction des faux positifs.

4.1.7 GÉNÉRATION DES ALERTES ET INTÉGRATION DANS LE SIEM WAZUH

Lorsqu'une fuite de données est confirmée, le moteur d'analyse dynamique de contamination génère un événement structuré au format JSON. Ce format inclut non seulement la valeur détectée et le canal d'exfiltration, mais également des informations détaillées sur la provenance, la méthode de résolution et l'existence éventuelle de conflits. Ces événements sont écrits dans un répertoire surveillé par un agent Wazuh, qui se charge de les transmettre au Wazuh Manager. Des règles de corrélation personnalisées permettent ensuite de classifier ces événements en fonction de leur gravité et de leur type. Les fuites confirmées sont distinguées des cas suspects, ce qui permet aux analystes de prioriser les incidents et d'éviter une surcharge informationnelle. Grâce à cette intégration, le SIEM ne se contente plus de corréler des événements bruts, mais devient capable d'exploiter une information enrichie sur la provenance des données, renforçant ainsi la pertinence et la fiabilité des alertes. Cette implémentation démontre qu'il est possible de concevoir un système d'analyse dynamique de contamination fondé sur la provenance, capable

de fonctionner en temps réel dans un environnement réel sans instrumentation intrusive. En s'appuyant sur des mécanismes de surveillance passive, de résolution explicite de provenance et d'intégration avec un SIEM, le système dépasse les limites structurelles des approches traditionnelles de prévention des fuites de données. Il permet notamment de distinguer des valeurs identiques issues de sources de sensibilité différente, tout en offrant une traçabilité complète des flux de données sensibles.

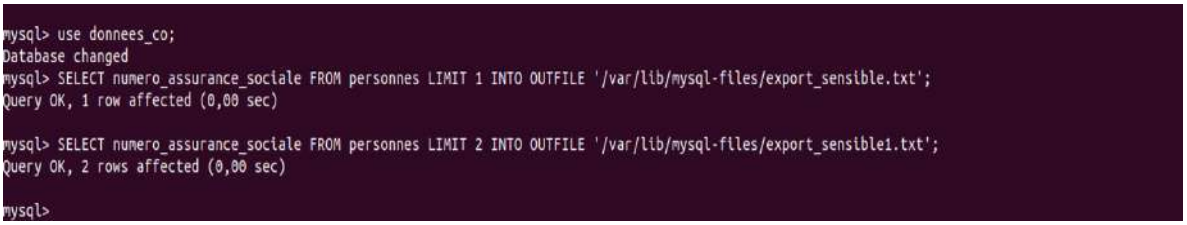

Figure 4.9 : Exportation de données sensibles depuis une base sensible (donnees_co)

La figure 4.9 illustre un scénario d'exportation de données sensibles à partir d'une base de données relationnelle MySQL. Dans cet exemple, l'utilisateur se connecte à la base de données donnees_co, identifiée comme contenant des informations sensibles. Deux requêtes SQL sont ensuite exécutées à l'aide de la commande "SELECT ... INTO OUTFILE". La première requête extrait une valeur du champ numero_assurance_sociale depuis la table personnes et l'exporte dans un fichier externe nommé export_sensible.txt. La seconde requête effectue une opération similaire, mais exporte cette fois deux enregistrements vers un fichier export_sensible1.txt. Ces opérations sont confirmées par le système MySQL, qui indique que les requêtes ont été exécutées avec succès (Query OK), avec respectivement une et deux lignes affectées. Cela signifie que les données ont effectivement été écrites dans les fichiers situés dans le répertoire /var/lib/mysql-files/. Ce type de commande constitue un vecteur d'exfiltration critique, car il permet de transférer directement des données sensibles depuis la base vers le système de fichiers, sans mécanisme de contrôle explicite sur la nature des données exportées. Du point

de vue des systèmes traditionnels, ces opérations apparaissent comme des actions légitimes, puisqu'elles utilisent des fonctionnalités natives du SGBD. Dans le cadre de notre approche, ces requêtes représentent un point clé pour l'analyse de la provenance. Les valeurs extraites sont marquées comme sensibles dès leur lecture dans la base de données, puis leur propagation est suivie jusqu'au fichier de sortie. Cela permet de détecter précisément que les données exportées proviennent d'une source sensible (donnees_co), et de déclencher une alerte lors de leur exfiltration. Ainsi, cette figure met en évidence la nécessité d'un mécanisme capable non seulement de détecter la présence de données sensibles, mais également de suivre leur origine et leur trajectoire dans le système, ce que les approches classiques basées uniquement sur le contenu ne permettent pas de réaliser.

Figure 4.10 : Exportation de données sensibles depuis une base sensible (donnees_co)

Comme illustré dans la figure, ci-dessus, l'utilisateur interagit avec le système de gestion de base de données MySQL en se connectant à la base données_co, identifiée comme contenant des informations sensibles. Une fois la base sélectionnée, deux requêtes SQL sont exécutées afin d'extraire des données depuis la table personnes. La première requête utilise la commande "SELECT ... INTO OUTFILE" pour extraire une valeur du champ numero_assurance_sociale et l'écrire dans un fichier externe nommé export_sensible.txt. La seconde requête effectue une opération similaire, mais exporte cette fois deux enregistrements vers un autre fichier

export_sensible1.txt. Le système MySQL confirme l'exécution de ces requêtes en indiquant respectivement « 1 row affected » et « 2 rows affected », ce qui signifie que les données ont bien été extraites et stockées dans les fichiers situés dans le répertoire /var/lib/mysql-files/. D'un point de vue fonctionnel, ces commandes sont légitimes et font partie des fonctionnalités standard offertes par MySQL pour permettre l'exportation de données. Toutefois, du point de vue de la sécurité, elles représentent un vecteur critique d'exfiltration. En effet, elles permettent de transférer directement des données sensibles depuis la base de données vers le système de fichiers, sans mécanisme intrinsèque de vérification de la sensibilité des données ni de contrôle sur leur destination. Cette situation illustre une limite importante des approches traditionnelles de détection de fuites de données. Les systèmes basés uniquement sur l'analyse du contenu ne peuvent pas distinguer si les données exportées proviennent d'une source sensible ou non, en particulier lorsque des valeurs identiques existent dans différents contextes. Dans le cadre de notre approche, ces requêtes constituent le point de départ du suivi de la provenance. Les données extraites sont marquées comme sensibles dès leur lecture dans la base donnees_co. Ce marquage est ensuite propagé dynamiquement à travers les différentes étapes du système,notamment lors de l'écriture dans les fichiers de sortie. Ainsi, même si les données sont transformées ou déplacées, leur origine reste traçable. Cette capacité, à suivre l'origine des données permet de détecter précisément que les fichiers générés contiennent des informations issues d'une source sensible, et donc de qualifier ces opérations comme des fuites potentielles. Elle constitue un avantage majeur par rapport aux approches classiques, en permettant de lever l'ambiguïté liée à l'analyse purement syntaxique ou statistique des données. Ainsi, cette figure illustre non seulement un scénario réaliste d'exfiltration de données, mais également l'intérêt d'une approche fondée sur la provenance pour assurer une détection fiable et précise des fuites de données dans des environnements complexes.

Figure 4.11 : Visualisation des logs générés par le système de Dynamic Taint Analysis sur les données sensible de donnees_co

On'a les journaux générés par le système de détection lors de l'analyse des opérations d'exportation de données. Ces journaux sont extraits du fichier /var/log/dta_non_sensible.log et sont structurés sous forme d'objets JSON, chaque entrée correspondant à un événement de détection. Chaque ligne contient plusieurs attributs décrivant le contexte de l'événement. Le champ timestamp indique le moment précis de la détection, tandis que valeur_detectee correspond à la donnée identifiée dans les flux, ici des identifiants de type numéro d'assurance sociale (par exemple NAS123456789. Le champ canal précise le vecteur d'exfiltration, qui correspond dans ce cas à une opération d'export EXPORT. Le champ fichier indique le nom du fichier dans lequel les données ont été écrites par exemple export_sensible.txt ou export_non_sensible.txt. L'élément clé de ces journaux réside dans les champs liés à la provenance. Le champ provenance_method indique la source utilisée pour déterminer l'origine des données, tandis que provenance_db identifie la base de données d'origine lorsque cette information est disponible. Dans certains cas, la provenance est indiquée comme unknown, ce qui signifie que le système n'a pas pu établir de lien direct entre la donnée observée et une source précise à partir des journaux disponibles. À l'inverse, lorsque la provenance est déterminée (par exemple via mysql_general_log, la base de données donnees_co_b est

correctement identifiée comme source. Le champ provenance_conflict indique si une ambiguïté a été détectée dans l'attribution de la provenance. Une valeur false signifie qu'aucun conflit n'a été identifié, ce qui confirme la cohérence de la corrélation entre les événements. Ces journaux mettent en évidence deux comportements distincts. Dans certains cas, des valeurs identiques (par exemple NAS123456789) apparaissent dans des fichiers différents, incluant à la fois des fichiers issus de sources sensibles et non sensibles. Les approches traditionnelles basées sur le contenu auraient déclenché des alertes identiques pour ces occurrences. Cependant, grâce à l'intégration de la provenance, notre système est capable de différencier ces événements en fonction de leur origine réelle. Lorsque la provenance est correctement identifiée, le système peut déterminer si la donnée provient d'une base sensible et qualifier l'événement comme une fuite potentielle. À l'inverse, lorsque la donnée est associée à une source non sensible ou qu'aucune provenance critique n'est identifiée, l'événement est enregistré sans générer d'alerte critique. Ainsi, cette figure illustre le fonctionnement interne du système de corrélation basé sur la provenance. Elle démontre la capacité du système à analyser des flux hétérogènes, à relier des événements provenant de différentes sources (base de données, fichiers, journaux) et à fournir une interprétation contextuelle des données détectées. Cette capacité constitue un élément clé pour réduire les faux positifs et améliorer la précision globale de la détection des fuites de données sensibles dans les systemes.

Figure 4.12 : Journal JSON des alertes de fuite de données sensibles générées par Wazuh

lors de la phase de validation expérimentale de notre dispositif. Chaque ligne correspond à une alerte structurée comportant un horodatage précis (timestamp), un niveau de criticité (level : 10), une description explicite (« Détection de fuite de données sensibles ») ainsi qu'un identifiant unique de règle (id : 100100) correspondant à la règle personnalisée injectée dans la configuration de Wazuh. La concentration temporelle des événements plusieurs dizaines d'alertes générées en moins de deux secondes (entre 18 :19 :05.202 et 18 :19 :05.294) atteste de la réactivité en temps réel du système. Les dernières lignes, de niveau inférieur, illustrent par contraste le bruit de fond système habituel (sessions PAM, arrêts de service), confirmant que la règle personnalisée discrimine efficacement les événements de fuite parmi le flux global de logs supervisés. l'appel detecter_fuite(ligne_decodee, "HTTP", log_file) boucle l'analyse en signalant explicitement le canal HTTP comme sink, ce qui permet, en cas d'alerte, de distinguer une fuite par exfiltration réseau d'une fuite par fichier exporté localement et d'adapter la réponse en conséquence

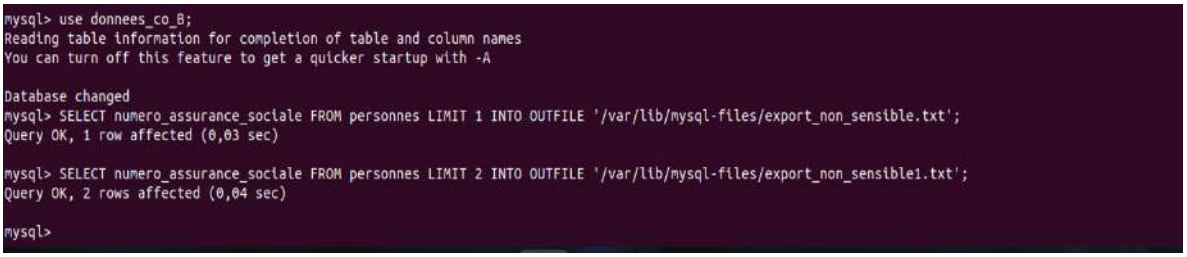

Figure 4.13 : Exportation de données sensibles depuis une base sensible (donnees_co_B)

près avoir sélectionné la base donnees_co_B préalablement déclarée comme sensible dans la configuration du Taint Pool deux requêtes SQL successives utilisent la clause INTO OUTFILE pour exporter le contenu de la colonne numero_assurance_sociale (donnée personnelle protégée) vers des fichiers texte stockés dans le répertoire /var/lib/mysql-files/. Ce répertoire correspond à l'un des emplacements surveillés en continu par la fonction surveiller_exports. Les confirmations Query OK retournées par MySQL attestent que les fichiers ont effectivement été créés, déclenchant ainsi en aval la chaîne de détection présentée précédemment.

Figure 4.14 : Exportation de données sensibles depuis une base non sensible (donnees_co_B)

La figure 4.14 présente un extrait des messages générés par le système lors de l'analyse des opérations d'exportation de données, mettant en évidence le processus de décision basé sur la provenance. Les premières lignes indiquent une étape de corrélation, matérialisée par le message Mapping OUTFILE, qui associe un fichier de sortie (par exemple export_non_sensible.txt à une base de données source (donnees_co_b). Cette étape est essentielle, car elle permet d'établir un lien explicite entre une donnée observée dans un fichier et son origine dans le système. Les lignes suivantes correspondent aux résultats de la détection. Chaque entrée est structurée

sous forme d'un objet JSON contenant plusieurs informations : un horodatage (timestamp), la valeur détectée (valeur_detectee), le canal utilisé (canal, ici EXPORT), ainsi que le fichier de destination. Ces éléments décrivent précisément le contexte dans lequel la donnée a été observée. L'élément central de cette figure réside dans la classification des événements comme « Détection non sensible ». Bien que des valeurs correspondant à des identifiants sensibles (par exemple NAS123456789) soient détectées, le système ne déclenche pas d'alerte critique. Cette décision s'explique par le fait que la provenance associée à ces données indique une source non sensible (donnees_co_b) ou une provenance non déterminée unknown. Cette distinction met en évidence un avantage majeur de l'approche proposée. Contrairement aux méthodes traditionnelles basées uniquement sur le contenu, qui auraient considéré ces valeurs comme sensibles en raison de leur format, notre système prend en compte l'origine réelle des données. Ainsi, une même valeur peut être classée différemment selon sa provenance. On observe également que certains événements présentent une provenance partiellement connue (via mysql_general_log) ou inconnue. Malgré cela, l'absence de correspondance avec une source sensible empêche la génération de fausses alertes. Le champ provenance_conflict est positionné à false, ce qui indique qu'aucune ambiguïté n'a été détectée dans l'attribution de la provenance. Cette figure illustre donc le mécanisme de décision du système, qui repose sur une corrélation entre les journaux et une analyse de la provenance pour qualifier les événements. Elle démontre que l'approche permet non seulement de détecter des données sensibles, mais également de contextualiser leur origine afin de réduire les faux positifs et d'améliorer la précision globale de la détection. Ainsi, même en présence de données identiques à celles issues de sources sensibles, le système est capable de distinguer les cas bénins des cas critiques, ce qui constitue un apport significatif par rapport aux approches traditionnelles

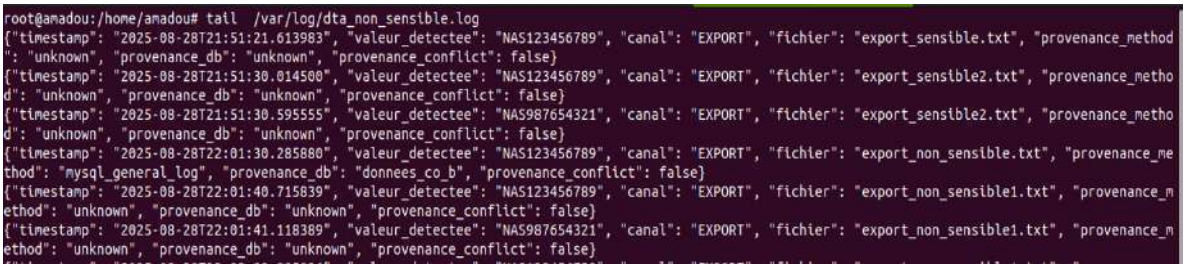

Figure 4.15 : Visualisation des logs générés par le système de Dynamic Taint Analysis sur les données non sensible de donnees_co_B)

La figure 4.15 présente un extrait du fichier /var/log/dta_non_sensible.log, journal natif produit par notre moteur d'analyse dynamique de contamination lors de la détection effective de fuites de données sensibles. Chaque entrée, formatée en JSON, documente un événement de détection avec un ensemble structuré de métadonnées : l'horodatage précis (timestamp), la valeur sensible identifiée (valeur_detectee), le canal d'exfiltration concerné (canal : "EXPORT"), le fichier de destination (fichier), ainsi que des informations de provenance retraçant l'origine de la donnée fuitée (provenance_method, provenance_db, provenance_conflict). Cette journalisation atteste que le dispositif a correctement détecté l'exportation des numéros d'assurance sociale (NAS123456789, NAS987654321) issus de la base donnees_co_b vers les fichiers générés par les requêtes INTO OUTFILE analysées précédemment.

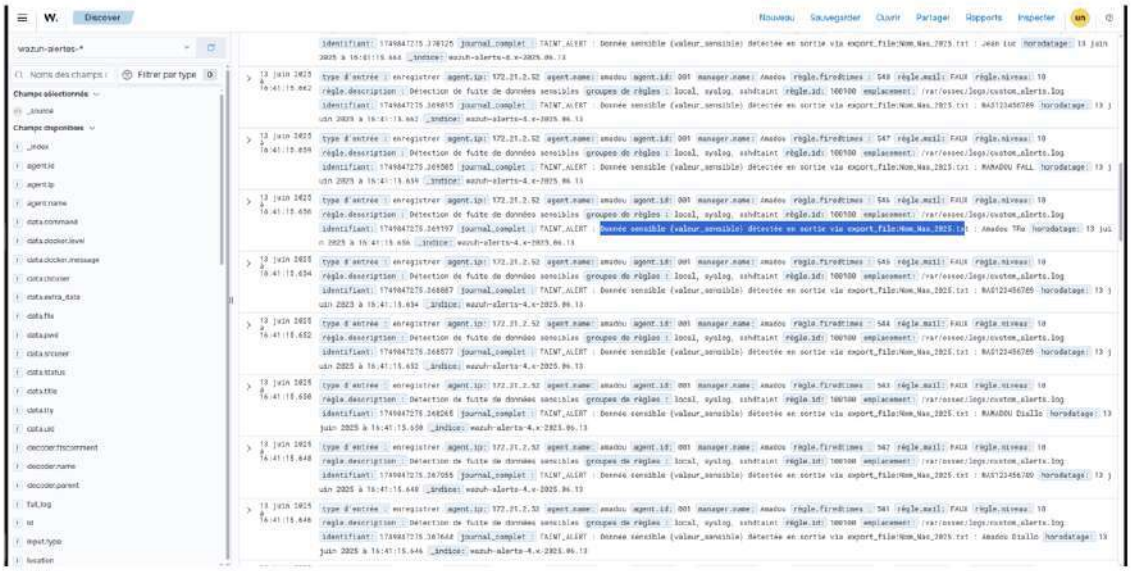

Figure 4.16 : Visualisation des alertes de fuite de données sensibles dans le tableau de bord Wazuh

La figure 4.16 présente le tableau de bord Discover de l'interface web Wazuh, donnant accès à l'index wazuh-alertes et offrant une vue agrégée des alertes générées par notre dispositif de détection de fuites. Chaque ligne correspond à une alerte enrichie produite par la règle personnalisée id : 100100 (« Détection de fuite de données sensibles ») de niveau 10, déclenchée par le décodeur sshdtaint à partir des entrées du journal /var/ossec/logs/custom_alerts.log. Les enregistrements affichés documentent précisément l'exfiltration de numéros d'assurance sociale (NAS123456789) associés à des identités nominales (Amadou Diallo, Mamadou Fall, Jean Luc. . .) vers le fichier Nom_Nas_2025.txt, démontrant ainsi la capacité du système à corréler des données structurées multi-champs et à les exposer dans une interface exploitable

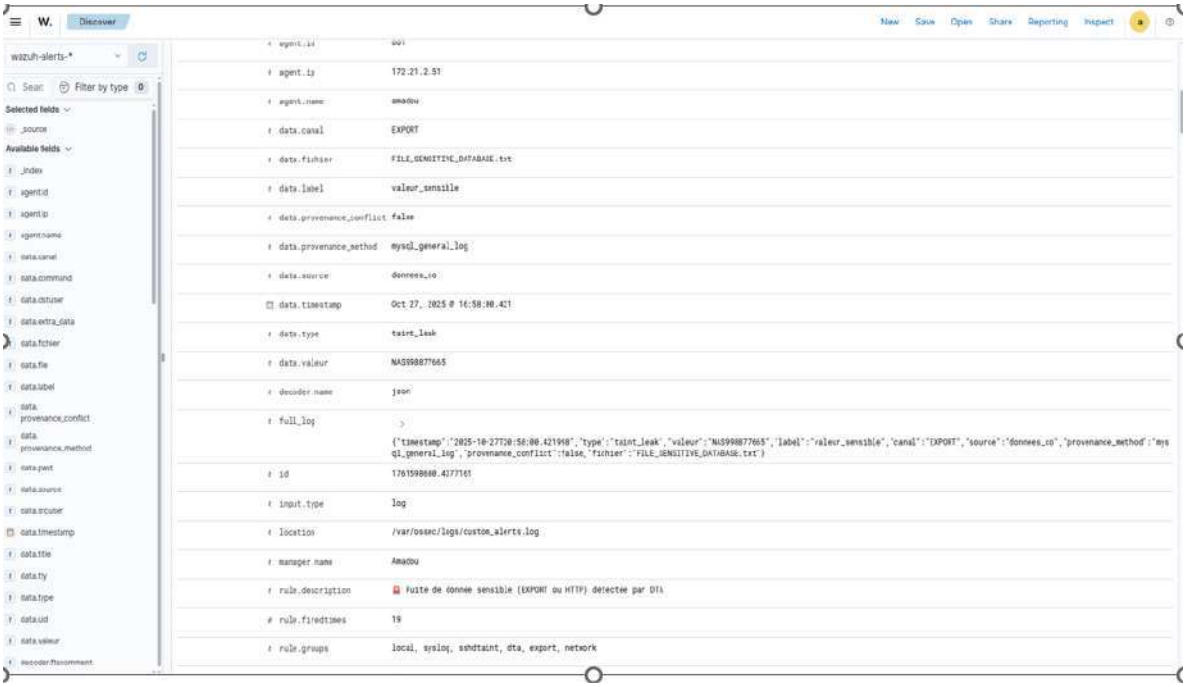

Figure 4.17 : Affichage des alertes de fuite de données sensibles exportées vers un fichier texte ou HTTP dans le tableau de bord Wazuh

La figure 4.16 présente une visualisation des alertes de sécurité dans l'interface Discover de Wazuh, illustrant l'intégration du système proposé dans un environnement SIEM réel. Cette interface permet d'explorer les événements collectés et indexés, en offrant une vue structurée des alertes sous forme de journaux enrichis. Chaque entrée correspond à un événement détecté et contient un ensemble d'attributs décrivant le contexte de la détection. On observe notamment les champs relatifs à l'agent (agent.ip, agent.name, agent.id), qui permettent d'identifier la machine à l'origine de l'événement. Les informations liées aux règlesWazuh sont également présentes, telles que rule.id, rule.level et rule.description. Dans ce cas, les alertes associées aux fuites de données sensibles sont caractérisées par une description explicite (« Détection de fuite de données sensibles ») et un niveau de sévérité élevé level : 10, reflétant leur criticité. Le champ journal_complet constitue un élément central de cette visualisation. Il contient un message enrichi indiquant que des données sensibles valeur_sensible ont

été détectées lors d'une opération d'exportation vers un fichier spécifique (par exemple export_file :Nom_Nas_2025.txt. Cette information résulte directement du mécanisme de corrélation et d'analyse de la provenance mis en oeuvre dans notre système. La répétition des alertes sur une courte période temporelle met en évidence une activité intensive d'exfiltration, cohérente avec le scénario expérimental. Contrairement aux systèmes traditionnels, ces alertes ne sont générées que lorsque la provenance des données confirme leur origine sensible, ce qui permet de limiter les faux positifs. Par ailleurs, l'interface Discover permet de filtrer et d'analyser les événements selon différents critères (champs, types, périodes), facilitant ainsi le travail des analystes de sécurité. Cette capacité d'exploration est essentielle pour comprendre le contexte des incidents et prioriser les réponses. Ainsi, cette figure démontre non seulement la capacité du système à détecter des fuites de données sensibles, mais également son intégration efficace dans une plateforme SIEM. Elle met en évidence la production d'alertes structurées, exploitables en temps réel, et enrichies par des informations de contexte issues de l'analyse de la provenance. Cette combinaison permet d'améliorer significativement la précision des détections et de réduire la charge d'analyse pour les équipes de sécurité.

Ainsi, cette méthodologie vise non seulement à concevoir un système capable de suivre dynamiquement la propagation des données sensibles, mais également à démontrer l'apport de la provenance dans l'amélioration de la détection des fuites de données dans des environnements informatiques modernes. L'approche proposée cherche à dépasser les limites des mécanismes traditionnels de prévention des fuites de données, qui reposent principalement sur l'analyse du contenu et qui considèrent implicitement la sensibilité comme une propriété intrinsèque de la donnée. Comme démontrée dans les chapitres précédents, cette hypothèse devient insuffisante dans des environnements distribués où des valeurs identiques peuvent circuler simultanément dans des contextes sensibles et non-sensibles.

Dans ce contexte, la méthodologie développée dans ce mémoire s'appuie sur une approche d'analyse dynamique de contamination fondée sur la provenance, permettant de relier chaque donnée détectée à sa source réelle ainsi qu'aux différents événements ayant contribué à sa propagation. L'objectif est de construire un mécanisme de suivi capable d'observer les flux de données à travers plusieurs composants hétérogènes du système, notamment les bases de données relationnelles, les systèmes de fichiers, les journaux applicatifs et les communications réseau. Cette approche vise ainsi à fournir une vision cohérente et contextualisée du cycle de vie des données sensibles, depuis leur extraction jusqu'à leur éventuelle exfiltration. Plus précisément, cette méthodologie cherche à répondre à plusieurs questions fondamentales liées à l'utilisation de l'analyse dynamique de contamination dans des environnements réels. La première concerne la capacité de l'analyse dynamique de contamination à suivre de manière fiable la provenance des données sensibles dans des systèmes modernes, caractérisés par la multiplicité des composants logiciels et la diversité des canaux de communication. Contrairement aux approches classiques d'analyse dynamique de contamination limitées à un programme instrumenté, l'approche proposée cherche à reconstruire des flux inter-systèmes en corrélant plusieurs sources de journaux et d'événements.

La seconde question porte sur l'impact de l'intégration de la provenance dans la réduction des faux positifs générés par les approches traditionnelles de détection. Les méthodes fondées sur les expressions régulières, le fingerprinting, l'apprentissage automatique ou les SIEM classiques détectent généralement des motifs ou des similarités sans pouvoir déterminer l'origine réelle des données observées. Cette limitation conduit à des situations ambiguës où des données non sensibles sont incorrectement classées comme critiques simplement parce qu'elles possèdent le même format ou la même valeur qu'une donnée sensible. L'approche proposée cherche à montrer que la prise en compte explicite de la provenance permet de lever

cette ambiguïté en distinguant des valeurs identiques selon leur contexte d'origine. La troisième question concerne l'intégration de l'analyse dynamique de contamination dans une plateforme SIEM telle que Wazuh. Dans les systèmes de supervision traditionnels, les alertes sont souvent produites à partir d'événements bruts dépourvus de contexte sur l'origine réelle des données. Cette méthodologie vise donc à enrichir les alertes générées avec des informations détaillées sur la provenance, les canaux d'exfiltration, les mécanismes de corrélation utilisés ainsi que les éventuels conflits de provenance détectés. L'objectif est de produire des alertes plus précises, plus contextualisées et directement exploitables par les analystes de sécurité. Enfin, cette méthodologie permet également d'étudier les conditions dans lesquelles l'approche proposée améliore la détection en temps réel des exfiltrations de données sensibles. Une attention particulière est portée aux scénarios d'exportation de données vers des fichiers externes via des commandes SQL telles que "`SELECT ... INTO OUTFILE`", ainsi qu'aux exfiltrations transitant par des canaux réseau HTTP. L'analyse de ces scénarios permet d'évaluer la capacité du système à suivre dynamiquement les flux de données sensibles dans des contextes réalistes, tout en maintenant une approche non-intrusive compatible avec des environnements de production. L'ensemble de cette méthodologie constitue ainsi le fondement expérimental et architectural du système développé dans ce mémoire. Les sections suivantes présentent en détail l'architecture du moteur d'analyse dynamique de contamination fondé sur la provenance, les mécanismes de construction du *taint_pool*, les stratégies de corrélation des journaux, ainsi que l'intégration des alertes dans l'infrastructure SIEM Wazuh.

CHAPITRE V

RESULTATS

Ce chapitre présente les résultats expérimentaux obtenus dans le cadre de l'évaluation du système de Dynamic Taint Analysis fondé sur la provenance proposé dans ce mémoire. L'objectif principal de cette évaluation est de mesurer l'efficacité de l'approche dans des scénarios réalistes de détection de fuites de données sensibles, tout en analysant sa capacité à réduire les faux positifs générés par les méthodes traditionnelles fondées uniquement sur le contenu. Comme démontré dans les chapitres précédents, les approches classiques de prévention des fuites de données, telles que les expressions régulières, le fingerprinting, les modèles d'apprentissage automatique ou les mécanismes de détection intégrés aux SIEM traditionnels, présentent des limitations importantes lorsqu'elles sont confrontées à des environnements distribués et hétérogènes. En particulier, ces approches rencontrent des difficultés majeures lorsque des valeurs identiques apparaissent simultanément dans des sources de sensibilité différente, puisqu'elles ne disposent d'aucun mécanisme explicite permettant d'identifier l'origine réelle des données. Dans ce contexte, l'évaluation expérimentale vise à analyser dans quelle mesure l'intégration de la provenance permet d'améliorer la fiabilité globale de la détection. Une attention particulière est portée à la capacité du système à distinguer des données identiques selon leur origine, à suivre dynamiquement leur propagation à travers

plusieurs composants du système et à générer des alertes contextualisées dans l'infrastructure SIEM Wazuh. Les expériences réalisées permettent également d'étudier le comportement du système dans des conditions réalistes comprenant des exportations SQL, des exfiltrations via HTTP, des événements concurrents ainsi que du trafic non pertinent.

Ce chapitre présente ainsi une comparaison détaillée entre l'approche proposée et plusieurs méthodes représentatives de l'état de l'art. Les résultats obtenus sont analysés selon plusieurs critères, notamment la précision, le rappel, le score F1, la génération de faux positifs, la scalabilité, la latence de détection et la robustesse du système dans des environnements bruités. L'objectif est de démontrer que la prise en compte explicite de la provenance constitue un élément déterminant pour améliorer la détection des fuites de données sensibles dans des environnements modernes.

5.1 RÉSULTATS EXPÉRIMENTAUX

Afin de valider expérimentalement l'approche proposée, plusieurs scénarios de détection ont été mis en œuvre dans un environnement reproduisant des conditions réalistes d'exfiltration de données sensibles. Ces expérimentations ont pour objectif d'évaluer la capacité du système à suivre la provenance des données, à distinguer des valeurs identiques selon leur origine et à produire des alertes fiables dans un contexte de supervision de sécurité intégré au SIEM Wazuh. L'évaluation repose sur une comparaison entre notre approche d'analyse dynamique de contamination fondée sur la provenance et plusieurs approches représentatives de l'état de l'art en prévention des fuites de données. Les expériences réalisées portent notamment sur l'analyse de la précision de détection, la réduction des faux positifs, la robustesse face aux environnements bruités ainsi que les performances du système en termes de latence et

de scalabilité. Les sections suivantes présentent en détail les objectifs de cette évaluation expérimentale, les scénarios étudiés, les approches comparées ainsi que les résultats obtenus.

5.1.1 OBJECTIF DE L'ÉVALUATION EXPÉRIMENTALE ET COMPARAISON AVEC DES APPROCHES EXISTANTES

L'objectif de cette évaluation expérimentale est de mesurer l'efficacité de l'approche proposée, fondée sur une analyse dynamique de contamination sensible à la provenance, et de la comparer à plusieurs approches représentatives de l'état de l'art en détection de fuites de données, notamment les méthodes basées sur des expressions régulières, les techniques de fingerprinting ainsi que les approches fondées sur l'apprentissage automatique. Ces approches reposent généralement sur une hypothèse implicite commune : la sensibilité d'une donnée peut être déterminée à partir de ses propriétés intrinsèques, telles que sa structure syntaxique, sa similarité avec des données connues ou encore son contexte sémantique. Toutefois, plusieurs travaux récents ont montré que ces hypothèses peuvent conduire à des erreurs importantes, en particulier dans les systèmes de sécurité, où les données sont souvent ambiguës, dynamiques et distribuées. Afin d'évaluer les limites de ces approches, nous avons conçu un scénario expérimental volontairement contraignant, inspiré de situations réelles observées dans des environnements organisationnels. Dans ce scénario, deux bases de données distinctes contiennent des valeurs strictement identiques, mais diffèrent quant à leur niveau de sensibilité : l'une est déclarée sensible, tandis que l'autre est explicitement considérée comme non sensible. Un total de 800 opérations d'exportation est réalisé, réparti équitablement entre les deux bases, avec 400 exportations issues de la base sensible et 400 exportations issues de la base non sensible. Ce scénario constitue un cas limite particulièrement pertinent, car il met en évidence une

ambiguïté fondamentale : du point de vue du contenu, les données sont strictement identiques, ce qui empêche toute approche basée uniquement sur le contenu de distinguer une fuite réelle d'un transfert légitime. Cette situation reflète des cas réels où des données identiques peuvent coexister dans plusieurs bases avec des niveaux de sensibilité différents. Dans ce contexte, les approches traditionnelles fondées sur le contenu détectent correctement les motifs correspondant à des données sensibles lors des exportations depuis la base sensible. Cependant, elles déclenchent également des alertes pour les exportations issues de la base non sensible. Ce comportement s'explique par leur incapacité à intégrer la notion de provenance dans le processus de décision. En conséquence, elles produisent un nombre élevé de faux positifs, ce qui réduit considérablement leur utilité opérationnelle. Des études ont montré que ce phénomène peut représenter une part très importante des alertes générées, ce qui entraîne une surcharge pour les analystes et diminue l'efficacité globale des systèmes de détection. De plus, les approches basées sur l'apprentissage automatique, bien qu'elles permettent de capturer des patterns plus complexes, présentent également des limitations importantes. Elles reposent sur des corrélations statistiques et nécessitent des données d'entraînement représentatives. Or, dans des scénarios comme celui étudié, où les données sont identiques, mais diffèrent uniquement par leur origine, ces modèles ne disposent d'aucune information discriminante exploitable. Par ailleurs, des travaux récents ont montré que ces modèles peuvent être sensibles à des phénomènes de fuite de données, ce qui peut conduire à une surestimation de leurs performances et à une mauvaise généralisation en conditions réelles.

À l'inverse, l'approche proposée repose sur une modélisation explicite de la provenance des données. Une valeur n'est considérée comme sensible que si elle provient d'une source identifiée comme sensible, et cette information est propagée dynamiquement à travers les différentes transformations et canaux d'exfiltration. Ainsi, deux valeurs strictement identiques

peuvent être distinguées en fonction de leur origine, ce qui permet de lever l'ambiguïté inhérente aux approches traditionnelles.

Les résultats expérimentaux montrent que cette différence conceptuelle se traduit par des gains significatifs en termes de performance. En particulier, notre approche permet de réduire drastiquement le taux de faux positifs tout en maintenant un niveau élevé de détection des fuites réelles. Cette amélioration est essentielle dans un contexte opérationnel, où la réduction des faux positifs est un facteur clé pour assurer la viabilité des systèmes de détection. Enfin, ces résultats mettent en évidence une limite fondamentale des approches existantes : tant que la sensibilité est traitée comme une propriété du contenu, les mécanismes de détection restent intrinsèquement ambigus. À l'inverse, la prise en compte de la provenance permet d'introduire une dimension contextuelle essentielle, rendant possible une détection plus fiable et plus précise des fuites de données dans des environnements complexes et hétérogènes.

5.1.2 ANALYSE DE LA PRÉCISION ET ABSENCE DE FAUX POSITIFS

Les résultats quantitatifs obtenus sont présentés dans le Le Tableau **??**, qui compare les performances de plusieurs approches représentatives de l'état de l'art avec celles de notre approche fondée sur la provenance. Les méthodes évaluées incluent des approches basées sur des expressions régulières, des systèmes de reconnaissance d'entités nommées (NER), ainsi que des modèles d'apprentissage profond orientés détection d'anomalies. Dans l'ensemble, ces approches parviennent à détecter les données issues de la base sensible, ce qui se traduit par des niveaux de rappel relativement élevés. Par exemple, le modèle de deep learning atteint un rappel de 0,92, tandis que les approches NER et regex obtiennent respectivement des rappels de 0,58 et 0,70. Ces résultats confirment que les méthodes basées sur le contenu sont efficaces

pour identifier des motifs connus ou des comportements atypiques lorsqu'ils sont présents dans les données. Cependant, ces performances en rappel s'accompagnent d'une dégradation significative de la précision. En effet, ces approches détectent également les valeurs issues de la base non sensible, car elles ne disposent d'aucun mécanisme permettant de distinguer l'origine des données. Cette limitation se traduit par des précisions relativement faibles : 0,52 pour les expressions régulières, 0,55 pour les approches NER et 0,60 pour le modèle de deep learning. Ce comportement s'explique par le fait que ces approches considèrent la sensibilité comme une propriété intrinsèque du contenu. Ainsi, toute valeur correspondant à un motif, à une entité ou à un profil statistique donné est systématiquement classée comme sensible, indépendamment du contexte dans lequel elle apparaît. Dans notre scénario expérimental, caractérisé par la présence de valeurs strictement identiques dans des bases de sensibilité différente, cette hypothèse conduit à une ambiguïté structurelle. À l'inverse, l'approche proposée introduit un critère de décision fondamentalement différent, basé sur la provenance des données. Une valeur n'est considérée comme sensible que si elle est associée à une source identifiée comme telle, information obtenue par corrélation entre différents journaux systèmes. Cette modélisation permet d'associer chaque donnée observée à son origine réelle. Les résultats montrent que cette différence conceptuelle se traduit par une amélioration significative des performances. Notre approche atteint une précision de 1.0, un rappel de 1.0 et un score F1 de 1.0. Elle est également la seule à ne générer aucune alerte pour les données issues de la base non sensible.

L'absence de faux positifs constitue un résultat central de cette évaluation. Elle démontre que la prise en compte explicite de la provenance permet de lever une ambiguïté fondamentale que les approches basées uniquement sur le contenu ne peuvent résoudre. En distinguant des valeurs identiques selon leur origine, notre approche améliore non seulement la précision, mais également la fiabilité globale du système. D'un point de vue opérationnel, cette amélioration

est essentielle. Les faux positifs représentent un défi majeur pour les systèmes de détection de fuites, car ils augmentent la charge cognitive des analystes et réduisent la confiance accordée aux alertes générées. En éliminant ces faux positifs, l'approche proposée rend le système plus exploitable dans des environnements réels. Enfin, ces résultats mettent en évidence un point fondamental : l'amélioration observée ne résulte pas d'un raffinement des techniques de classification, mais d'un changement de paradigme. En définissant la sensibilité comme une propriété dépendant de la provenance plutôt que du contenu, notre approche permet de résoudre un problème intrinsèquement ambigu dans les systèmes traditionnel.

5.1.3 SYNTHÈSE DES RÉSULTATS

La synthèse des résultats expérimentaux met en évidence l'apport fondamental de l'approche proposée par rapport aux méthodes traditionnelles de détection de fuites de données. Les chiffres obtenus montrent que les expressions régulières, bien qu'efficaces pour détecter des motifs sensibles dans des scénarios simples, présentent une limitation structurelle dès lors que des valeurs identiques apparaissent dans des sources sensibles et non-sensibles. Dans le scénario évalué, cette limitation se traduit par une explosion des faux positifs et une précision divisée par deux, malgré un rappel élevé. À l'inverse, l'analyse dynamique de contamination sensible à la provenance permet de maintenir simultanément une détection exhaustive et une précision parfaite. En associant chaque valeur détectée à une provenance explicite et en utilisant cette information comme critère principal de décision. Les résultats montrent ainsi que la sensibilité doit être considérée comme une propriété contextuelle dépendant de l'origine et de la propagation des données, et non comme une simple caractéristique syntaxique. En conclusion, les résultats expérimentaux confirment que l'approche proposée surpasse significativement les

approches (Meli et al.), (Ji et al.) et (Martino et al.). dans les scénarios où la provenance joue un rôle déterminant. Ils démontrent que l'intégration de l'analyse dynamique de contamination avec une gestion explicite de la provenance et une infrastructure SIEM permet d'obtenir une détection à la fois précise, interprétable et opérationnellement viable. Ces résultats valident l'hypothèse centrale de ce mémoire et justifient l'adoption d'une approche orientée provenance pour la détection fiable des fuites de données sensibles.

Méthode	**Type**	**Précision**	**Rappel**	**F1**	**Perte moyenne**	**Données sensibles**	**Données non sensibles**
Expressions régulières (Meli et al. [28])	Classification	0.52	0.70	0.60	N/A	Détectées	Détectées
Presidio / NER (Ji et al. [17])	Classification	0.55	0.58	0.56	N/A	Détectées	Détectées
Apprentissage profond (Martino et al. [27])	Basée sur les anomalies	0.60	0.92	0.72	6.49 / 5.94	Détectées	Détectées
Approche basée sur la provenance (proposée)	Classification	**1.0**	**1.0**	**1.0**	N/A	Détectées	**Non détectées**

Tableau 5.1 : Comparaison des approches de détection en présence d'ambiguïté de provenance

Le Tableau 5.1 confronte notre approche fondée sur la provenance à trois méthodes représentatives de l'état de l'art en détection de fuites de données sensibles : la détection par expressions régulières (Meli et al.), l'extraction par reconnaissance d'entités nommées (Presidio/NER, Ji et al.) et la détection par apprentissage profond (Martino et al.). L'évaluation s'appuie sur quatre indicateurs classiques précision, rappel, score F1 et perte moyenne auxquels s'ajoutent deux colonnes qualitatives caractérisant le comportement de chaque méthode face à des données sensibles et non-sensibles. Notre approche obtient un score parfait sur l'ensemble des métriques quantitatives (P = R = F1 = 1.0) et se distingue surtout par sa capacité à ne pas signaler les données non sensibles, là où les trois approches concurrentes produisent indistinctement des détections sur les deux catégories — révélant ainsi un taux important de faux positifs.

5.1.4 ANALYSE DE LA SCALABILITÉ ET DE LA ROBUSTESSE

Afin d'évaluer les performances du système, une analyse de la scalabilité et de la robustesse a été réalisée en faisant varier le volume de données sensibles chargées en mémoire. Cette évaluation permet de mesurer l'impact de l'augmentation du nombre de valeurs sur la consommation mémoire ainsi que sur le temps de chargement du système.

Taille	**Nombre de valeurs**	**Mémoire (MB)**	**Temps de chargement (s)**
10^3	1 000	0.05	0.0006
10^4	10 000	1.17	0.0072
10^5	100 000	9.98	0.0808

Tableau 5.2 : Scalabilité en termes d'utilisation mémoire et de temps de construction de l'index

Afin d'évaluer la capacité du système à fonctionner dans des environnements de différentes tailles, une analyse de scalabilité a été réalisée en faisant varier le nombre de valeurs sensibles indexées. Les résultats sont présentés dans le Tableau 5.2. Les résultats montrent que la consommation mémoire augmente de manière linéaire avec le nombre de valeurs indexées. Le système utilise environ 0,05 MB pour 10^3 valeurs et atteint près de 10 MB pour 10^5 valeurs. Cette croissance linéaire indique que la représentation en mémoire est efficace et prévisible, ce qui rend l'approche adaptée à des environnements de taille modérée. Par ailleurs, le temps de construction de l'index reste très faible dans toutes les configurations testées. Même pour 10^5 valeurs, le temps de chargement demeure inférieur à 0,1 seconde. Cette rapidité d'initialisation constitue un avantage important pour les systèmes nécessitant des redémarrages fréquents ou un déploiement dynamique. L'analyse des performances de détection, présentée dans le Tableau 5.3, met en évidence une augmentation significative de la latence avec la

taille des données. Le temps moyen de traitement par ligne passe de 0,18 ms pour 10^3 valeurs à 31,88 ms pour 10^5. Cette augmentation impacte directement le débit du système, qui passe de 5586 lignes par seconde à seulement 31 lignes par seconde. Cette dégradation des performances s'explique par la stratégie actuelle de correspondance, qui repose sur une comparaison séquentielle entre chaque entrée de log et l'ensemble des valeurs indexées. Par conséquent, le coût computationnel augmente linéairement avec la taille de l'index. Cependant, cette limitation relève principalement d'un choix d'implémentation plutôt que d'une contrainte intrinsèque de l'approche. L'utilisation de structures de données plus efficaces, telles que des tables de hachage ou des index optimisés, pourrait permettre d'améliorer significativement les performances. En complément, une évaluation de la robustesse a été réalisée en introduisant du bruit dans l'environnement expérimental, notamment sous forme de requêtes concurrentes, de trafic HTTP non sensible et de journaux système non pertinents. Cette configuration permet de simuler des conditions réalistes où plusieurs événements se produisent simultanément. Les résultats montrent que l'approche reste efficace dans ces conditions, grâce à son mécanisme de corrélation basé sur la provenance. Chaque événement est analysé indépendamment, ce qui permet au système de gérer des flux concurrents sans perte de précision. Ainsi, bien que des améliorations soient possibles en termes de performance à grande échelle, les résultats démontrent que l'approche proposée est à la fois robuste et applicable dans des environnements réels, en particulier dans des contextes distribués caractérisés par une forte hétérogénéité des sources de données.

Le Tableau 5.3 présente les performances de détection du système en termes de latence et de débit, en fonction du nombre de valeurs sensibles indexées.Les résultats mettent en évidence une augmentation significative de la latence lorsque la taille de l'index croît. Plus précisément, le temps moyen de traitement par ligne de log passe de 0,18 ms pour 10^3 valeurs à 31,88 ms

Taille	Temps de scan (s)	Latence (ms/ligne)	Débit (lignes/s)
10^3	1.43	0.18	5 586
10^4	13.35	1.67	599
10^5	255.03	31.88	31

Tableau 5.3 : Latence de détection et débit en fonction du nombre de valeurs indexées

pour 10^5. Cette évolution indique que le coût de traitement d'un événement augmente avec le volume de données à comparer, ce qui reflète directement l'impact de la taille de l'index sur les performances du système. Cette augmentation de la latence a une conséquence directe sur le débit global. En effet, le système est capable de traiter environ 5 586 lignes de log par seconde pour 10^3 valeurs, mais ce débit chute drastiquement à seulement 31 lignes par seconde pour 10^5. Cette diminution importante traduit une dégradation des performances lorsque le système est confronté à un grand nombre de valeurs sensibles, ce qui met en évidence une limite de scalabilité dans l'implémentation actuelle. Ce comportement s'explique par la stratégie de correspondance utilisée, qui repose sur une comparaison séquentielle entre chaque ligne de log et l'ensemble des valeurs sensibles indexées. Dans ce modèle, chaque nouvelle entrée doit être comparée à toutes les valeurs existantes, ce qui entraîne une complexité linéaire. Ainsi, plus le nombre de valeurs augmente, plus le temps de traitement s'allonge, ce qui impacte directement la latence et le débit. Il est toutefois important de souligner que cette limitation est principalement liée au choix d'implémentation et non à une contrainte intrinsèque de l'approche fondée sur la provenance. En effet, des structures de données plus efficaces, telles que des tables de hachage, des arbres de recherche ou des index optimisés, permettraient de réduire considérablement le temps de recherche et d'améliorer la scalabilité du système. Malgré cette dégradation des performances à grande échelle, les résultats montrent que le

système reste très efficace pour des volumes de données faibles à modérés, ce qui correspond à de nombreux cas d'usage réels. Par ailleurs, la précision de détection n'est pas affectée par cette augmentation de la taille des données, ce qui souligne la robustesse du mécanisme de détection basé sur la provenance. Afin de mieux refléter les conditions d'un environnement réel, une activité de fond a été introduite sous forme de requêtes concurrentes sur la base de données, de trafic HTTP non-sensible et de journaux système non-pertinents. Cette configuration permet d'évaluer le comportement du système en présence de bruit et d'événements simultanés, comme c'est le cas dans des systèmes de production. Les résultats obtenus montrent que le système conserve sa capacité de détection dans ces conditions, grâce à son mécanisme de traitement indépendant des événements. Chaque ligne de log est analysée de manière autonome, ce qui permet au système de gérer des flux concurrents sans interférence entre les événements. Bien que des tests de charge à grande échelle restent à explorer dans des travaux futurs, les résultats actuels démontrent que l'approche proposée est robuste et applicable dans des environnements réalistes. Ces observations confirment que la dégradation observée est principalement liée à des choix d'implémentation et non à une limite fondamentale du modèle basé sur la provenance.

CHAPITRE VI

DISCUSSION, LIMITES ET PERSPECTIVES

Ce chapitre présente une analyse critique des travaux réalisés dans le cadre de ce mémoire, en mettant en perspective les contributions de l'approche proposée ainsi que les limites observées lors de son implémentation et de son évaluation expérimentale. Après avoir présenté les fondements théoriques, l'architecture du système et les résultats obtenus, il devient essentiel de discuter de la portée réelle de l'approche dans le contexte des systèmes modernes de détection de fuites de données. L'objectif de ce chapitre est également d'examiner les implications méthodologiques et opérationnelles de l'intégration de la provenance dans une approche de Dynamic Taint Analysis. Cette discussion permet de mieux situer les apports du système proposé par rapport aux approches traditionnelles fondées uniquement sur le contenu, tout en mettant en évidence les défis associés à la corrélation de journaux, à la scalabilité du système et à son déploiement dans des environnements réels. Enfin, ce chapitre présente les principales limites de l'approche ainsi que plusieurs perspectives de recherche visant à améliorer les mécanismes de suivi de provenance, l'efficacité du système et l'intégration de nouvelles capacités de détection et de réponse dans des infrastructures de sécurité modernes.

6.1 DISCUSSION

Les travaux présentés dans ce mémoire s'inscrivent dans une problématique plus large liée à la détection fiable des fuites de données dans des environnements modernes caractérisés par leur complexité, leur distribution et leur hétérogénéité. L'analyse de l'état de l'art a montré que la majorité des approches existantes, qu'elles soient fondées sur des signatures, des techniques de similarité ou des modèles d'apprentissage automatique, reposent sur une hypothèse implicite commune : la sensibilité d'une donnée peut être déterminée à partir de ses propriétés intrinsèques. Cette hypothèse, bien qu'efficace dans des contextes simples, devient insuffisante dans des environnements réels où les données circulent entre plusieurs systèmes et peuvent exister simultanément dans des contextes de sensibilité différents. Dans ces situations, le contenu seul ne constitue plus un critère discriminant fiable, ce qui introduit une ambiguïté structurelle difficile à résoudre avec les approches traditionnelles. L'approche proposée dans ce mémoire s'inscrit dans une perspective différente en considérant que la sensibilité d'une donnée dépend avant tout de sa provenance. En s'appuyant sur la corrélation de journaux hétérogènes, elle permet d'inférer l'origine des données sans recourir à une instrumentation intrusive des systèmes. Ce choix constitue un compromis pertinent entre expressivité et déployabilité, en permettant une intégration dans des environnements existants tout en conservant une capacité d'analyse contextuelle. L'intégration du système au sein d'un SIEM, en l'occurrence Wazuh, illustre cette volonté de rapprocher les contributions théoriques des contraintes opérationnelles. Elle montre qu'il est possible de transformer un concept issu de la recherche en un mécanisme de détection concret, capable de produire des alertes exploitables dans un contexte de supervision de la sécurité. Dans le cadre de cette étude, certains choix expérimentaux ont été réalisés afin de structurer l'évaluation de manière contrôlée et représentative. Le choix de MySQL comme

système de gestion de base de donnée s'explique par son adoption large dans les environnements organisationnels et par la disponibilité de journaux de requêtes détaillés, essentiels pour tracer l'origine des données. Ces journaux permettent d'établir un lien explicite entre les opérations effectuées et les données manipulées, ce qui constitue un élément central pour l'inférence de la provenance. Le recours au numéro d'assurance sociale comme exemple de donnée sensible répond à un objectif de clarté et de représentativité. Ce type de donnée, fortement identifiable, permet d'illustrer efficacement les mécanismes de détection. Toutefois, l'approche proposée ne dépend pas de la nature spécifique des données, mais de leur association à une source identifiée comme sensible. Elle peut ainsi être appliquée à une grande variété de données, incluant des informations financières, médicales ou tout autre type de contenu nécessitant une protection. Par ailleurs, le scénario expérimental s'appuie sur deux bases de données distinctes, l'une contenant des données sensibles et l'autre des données non-sensibles. Ce cadre permet de mettre en évidence de manière explicite l'ambiguïté de provenance. Néanmoins, le modèle proposé est conçu pour gérer un nombre arbitraire de sources, chaque source étant associée à un niveau de sensibilité. L'extension à plusieurs bases de données ne modifie pas les principes de l'approche, mais augmente simplement la complexité du processus de corrélation. De la même manière, bien que l'étude se concentre sur des bases de données, l'approche n'est pas limitée à ce type de support. Elle repose sur l'analyse de journaux et la corrélation d'événements, ce qui la rend applicable à tout système capable de produire des traces exploitables, tels que les systèmes de fichiers, les services web ou d'autres formes de stockage. La base de données constitue ainsi un cas d'étude représentatif, mais non exclusif. Une analyse plus approfondie des performances met en évidence une distinction importante entre les limitations conceptuelles et les limitations liées à l'implémentation. Les contraintes observées en termes de performance à grande échelle résultent principalement des mécanismes

techniques utilisés, notamment la stratégie de correspondance séquentielle. Ces limitations peuvent être atténuées par l'utilisation de structures de données plus efficaces, sans remettre en cause les fondements de l'approche. Enfin, le choix d'un modèle de provenance basé sur la corrélation de journaux, plutôt que sur une représentation complète des flux internes, repose sur une abstraction volontaire. L'objectif n'est pas de capturer l'ensemble des transformations des données, mais de préserver une information essentielle : leur origine. Cette abstraction permet de distinguer des données identiques selon leur provenance, ce qui constitue le cœur du problème traité. Elle offre ainsi une solution pragmatique et adaptée aux environnements réels, où une instrumentation complète des systèmes n'est pas toujours envisageable. Ainsi, cette discussion met en évidence que la contribution principale de ce mémoire ne réside pas uniquement dans l'amélioration des performances de détection, mais dans une réorientation du problème vers une approche contextuelle fondée sur la provenance. Cette perspective permet de dépasser les limites des approches traditionnelles et ouvre la voie à des mécanismes de détection plus robustes, adaptés aux systèmes d'information contemporains.

6.2 LIMITES DE L'APPROCHE

Malgré les résultats encourageants obtenus, l'approche proposée présente un certain nombre de limites qui doivent être prises en considération. Tout d'abord, le système repose fortement sur la disponibilité et la qualité des journaux collectés. En particulier, la reconstruction de la provenance dépend de la présence de logs suffisamment riches, notamment les journaux de requêtes SQL permettant d'identifier l'origine des données extraites. En l'absence de ces informations, ou dans des environnements où les logs sont incomplets, agrégés ou désactivés pour des raisons de performance ou de confidentialité, la capacité du système à inférer

correctement la provenance peut être limitée. Ensuite, l'approche suppose une corrélation cohérente entre plusieurs sources de journaux hétérogènes (bases de données, système de fichiers, serveurs web). Dans des environnements réels, cette corrélation peut être affectée par des problèmes de synchronisation temporelle, de format des logs ou de perte d'événements, ce qui peut introduire des imprécisions dans l'analyse. Par ailleurs, l'évaluation expérimentale a été réalisée dans un environnement contrôlé et à une échelle limitée. Bien que les résultats obtenus soient significatifs, le système n'a pas encore été validé dans des environnements de production à grande échelle, caractérisés par des volumes de données élevés, une forte concurrence et une diversité importante de sources. L'absence de déploiement à grande échelle constitue ainsi une limite importante, notamment en ce qui concerne l'évaluation des performances et de la robustesse du système dans des conditions réelles. En outre, la stratégie actuelle de correspondance repose sur des mécanismes séquentiels, ce qui entraîne une dégradation des performances lorsque le nombre de valeurs sensibles augmente. Bien que cette limitation soit principalement liée à l'implémentation, elle peut constituer un frein dans des environnements nécessitant un traitement en temps réel à grande échelle. De plus, l'approche ne couvre pas l'ensemble des vecteurs d'exfiltration possibles. Certains canaux, tels que les communications chiffrées, les services infonuagiques ou les transferts via des applications externes, ne sont pas pris en compte dans le cadre actuel. L'extension du système à ces scénarios nécessiterait l'intégration de nouvelles sources de données et de mécanismes d'analyse supplémentaires. Enfin, l'approche repose sur une modélisation implicite de la provenance basée sur la corrélation d'événements, sans construction explicite de graphes de dépendance complets. Bien que ce choix permette une mise en œuvre non intrusive, il peut limiter la capacité à analyser des relations complexes ou des chaînes de propagation longues dans certains scénarios avancés. Ces limitations n'invalident pas l'approche proposée,

mais soulignent les défis associés à son déploiement dans des environnements réels. Elles constituent également des pistes d'amélioration pour des travaux futurs. Dans le cadre de cette étude, certains choix expérimentaux ont été réalisés afin de structurer l'évaluation de manière contrôlée et représentative. Le choix de MySQL comme système de gestion de base de données s'explique par son usage largement répandu dans les environnements organisationnels, ainsi que par la disponibilité de journaux de requêtes détaillés permettant de tracer précisément l'origine des données. Ce type de journalisation constitue un élément essentiel pour l'inférence de la provenance, car il permet d'établir un lien explicite entre les requêtes exécutées et les données exportées. Concernant la nature des données sensibles utilisées, le numéro d'assurance sociale a été retenu comme cas d'étude en raison de son caractère fortement identifiable et de son utilisation fréquente dans les travaux liés à la protection des données. Ce choix permet d'illustrer clairement les mécanismes de détection dans un contexte contrôlé. Toutefois, l'approche proposée ne dépend pas de la nature spécifique des données, mais uniquement de leur association à une source identifiée comme sensible. Par conséquent, elle peut être appliquée à toute autre forme de données sensibles, qu'il s'agisse d'identifiants, d'informations financières, de données médicales ou de tout autre contenu nécessitant une protection particulière. Par ailleurs, le scénario expérimental repose sur deux bases de données distinctes, l'une contenant des données sensibles et l'autre des données non sensibles. Ce choix vise à mettre en évidence de manière explicite l'ambiguïté de provenance dans un cadre simplifié. Néanmoins, l'approche proposée n'est pas limitée à ce cas particulier. Le modèle de corrélation et de propagation de la provenance est conçu pour gérer un nombre arbitraire de sources, chaque source étant associée à un niveau de sensibilité. Ainsi, l'extension à plusieurs bases de données ne modifie pas les principes fondamentaux du système, mais uniquement la complexité du processus de corrélation. De même, bien que l'étude se concentre principalement sur des bases de données,

l'approche n'est pas restreinte à ce type de support. Elle repose sur l'analyse de journaux et la corrélation d'événements, ce qui la rend applicable à tout système capable de produire des traces exploitables, tels que les systèmes de fichiers, les services web ou d'autres formes de stockage. Dans cette perspective, la base de données constitue un cas d'usage représentatif, mais non exclusif. Enfin, la question de la représentation des données et du suivi de leur propagation mérite d'être discutée. Dans l'approche proposée, les données manipulées sont considérées comme des instances observables issues de différentes sources, et la contamination est associée à leur origine plutôt qu'à leur structure interne. Ce choix permet de capturer les informations nécessaires à la distinction entre données sensibles et non sensibles, même lorsque celles-ci sont identiques sur le plan syntaxique. Ainsi, le mécanisme de suivi ne vise pas à représenter l'ensemble des transformations internes des données, mais à préserver une information essentielle : leur provenance. Cette abstraction, bien que simplifiée, s'avère suffisante pour répondre au problème ciblé, à savoir la détection fiable des fuites de données dans des environnements hétérogènes.

6.3 RÉPONSES AUX QUESTIONS DE RECHERCHE

Les travaux réalisés dans ce mémoire avaient pour objectif principal d'explorer le potentiel d'une approche de l'analyse dynamique de contamination fondée sur la provenance pour améliorer la détection des fuites de données sensibles dans des environnements modernes. À travers la conception du système, son implémentation et son évaluation expérimentale, plusieurs éléments de réponse ont pu être apportés aux questions de recherche formulées au début de ce mémoire. La première question portait sur la manière dont l'analyse dynamique de contamination peut être utilisée afin de suivre de façon fiable la provenance des données

sensibles dans un système informatique moderne. Les résultats obtenus montrent qu'il est possible d'utiliser les principes de l'analyse dynamique de contamination sans recourir à une instrumentation intrusive des applications. L'approche proposée repose sur une corrélation continue entre plusieurs sources de journaux hétérogènes, notamment les journaux de requêtes SQL, les journaux systèmes, les fichiers exportés et les traces HTTP. Cette corrélation permet d'associer dynamiquement chaque donnée observée à sa source d'origine et de suivre sa propagation à travers différents composants du système. Contrairement aux approches traditionnelles d'analyse dynamique de contamination limitées à un programme instrumenté, le système développé dans ce mémoire adopte une vision inter-systèmes, capables d'observer des flux de données traversant des environnements distincts. Les expérimentations montrent ainsi qu'une information de provenance cohérente peut être reconstruite à partir des événements observables produits par les systèmes existants.

La seconde question concernait l'impact de la provenance sur la réduction des faux positifs générés par les approches classiques de détection. Les résultats expérimentaux mettent clairement en évidence que l'intégration explicite de la provenance constitue un facteur déterminant pour résoudre l'ambiguïté liée aux données identiques présentes dans plusieurs contextes. Les approches fondées uniquement sur le contenu, telles que les expressions régulières, les techniques de fingerprinting, les modèles d'apprentissage automatique ou les SIEM traditionnels, considèrent généralement la sensibilité comme une propriété intrinsèque de la donnée. Dans le scénario étudié, cette hypothèse conduit à la génération d'alertes pour des données provenant pourtant de sources non sensibles. À l'inverse, l'approche proposée associe chaque donnée détectée à son origine réelle avant de prendre une décision. Cette capacité, à distinguer des valeurs identiques selon leur provenance permet d'éliminer les ambiguïtés structurelles observées dans les approches traditionnelles. Les résultats obtenus montrent ainsi

une réduction significative des faux positifs, tout en conservant un haut niveau de détection des fuites réelles. La troisième question visait à déterminer comment une approche de l'analyse dynamique de contamination peut être intégrée à un SIEM tel que Wazuh afin de produire des alertes plus exploitables et contextualisées. Les travaux réalisés montrent qu'une telle intégration est non seulement possible, mais également particulièrement pertinente dans un contexte opérationnel. Le moteur de détection développé génère des événements enrichis contenant non seulement les données détectées et le canal d'exfiltration utilisé, mais également des informations détaillées sur la provenance, les mécanismes de corrélation et les éventuels conflits de provenance. Ces événements sont ensuite transmis au SIEM Wazuh sous forme de journaux structurés au format JSON. Grâce aux règles de corrélation définies dans Wazuh, les alertes produites peuvent être classifiées selon leur criticité et enrichies avec des informations de contexte facilitant leur interprétation aux analystes de sécurité. Cette approche permet ainsi de transformer un SIEM traditionnel, généralement limité à l'agrégation d'événements, en une plateforme capable d'exploiter une information contextuelle avancée sur l'origine des données.

Enfin, la dernière question concernait les conditions dans lesquelles l'approche proposée améliore la détection en temps réel des exfiltrations de données sensibles vers des fichiers externes ou des canaux réseau tel que HTTP. Les expérimentations menées montrent que cette amélioration devient particulièrement significative dans des environnements où plusieurs sources contiennent des données similaires ou identiques. Dans ces situations, les approches fondées uniquement sur le contenu deviennent incapables de différencier une fuite réelle d'une opération légitime. En s'appuyant sur une analyse continue des journaux SQL et HTTP, ainsi que sur un mécanisme de corrélation temporelle, le système proposé est capable de détecter des opérations d'exportation et des transmissions réseau tout en conservant l'information de provenance associée aux données concernées. Les résultats montrent également que l'approche

reste robuste en présence d'événements concurrents, de trafic non-pertinent et de journaux bruités. Bien que certaines limitations subsistent en termes de scalabilité à très grande échelle, les performances observées démontrent que l'approche est compatible avec une détection en temps réel dans des environnements de taille modérée à importante. Dans l'ensemble, les réponses apportées à ces questions confirment l'hypothèse centrale de ce mémoire : la provenance constitue une dimension essentielle pour améliorer la fiabilité de la détection des fuites de données sensibles. En replaçant l'origine des données au cœur du processus de décision, l'approche proposée permet de dépasser les limites structurelles des mécanismes traditionnels de prévention des fuites de données et ouvre la voie à des systèmes de détection plus précis, plus interprétables et mieux adaptés aux environnements informatiques contemporains.

6.4 PERSPECTIVES ET TRAVAUX FUTURS

Les travaux présentés dans ce mémoire ouvrent plusieurs perspectives visant à étendre et enrichir l'approche proposée, tant sur le plan méthodologique que sur le plan opérationnel. Une première direction concerne l'amélioration des mécanismes d'indexation et de recherche des données sensibles. L'utilisation de structures de données plus avancées, telles que des tables de hachage, des index inversés ou des techniques de filtrage probabiliste, permettrait de rendre le système plus efficace et mieux adapté à des environnements à grande échelle. Une autre perspective porte sur l'enrichissement des sources de journaux utilisées pour l'inférence de la provenance. L'intégration de nouvelles traces issues de différents systèmes d'information, notamment dans des environnements distribués ou infonuagiques, permettrait d'élargir le champ d'application de l'approche et de couvrir une plus grande diversité de scénarios. Par ailleurs, l'amélioration des mécanismes de corrélation constitue un axe important de développement.

L'introduction de modèles temporels plus élaborés ou de structures de type graphe permettrait de capturer des relations plus complexes entre les événements, et ainsi d'affiner la reconstruction de la provenance des données. Une autre piste consiste à combiner l'approche proposée avec des techniques complémentaires, notamment issues de l'apprentissage automatique. Ces méthodes pourraient être utilisées pour détecter des comportements anormaux ou des schémas d'exfiltration, en complément de l'analyse fondée sur la provenance. Enfin, l'intégration de cette approche dans des architectures de sécurité plus larges, incluant des mécanismes de réponse automatisée, représente une évolution naturelle. Cela permettrait de passer d'une logique de détection à une logique de prévention, en permettant au système d'intervenir automatiquement lors de l'identification d'un comportement suspect. Ces différentes perspectives montrent que l'approche proposée constitue une base évolutive, pouvant être enrichie et adaptée pour répondre aux exigences croissantes des systèmes d'information modernes.

CONCLUSION

Ce mémoire avait pour objectif de proposer une approche de détection des fuites de données sensibles fondée sur l'analyse dynamique de contamination et la prise en compte explicite de la provenance des données. Les travaux réalisés ont montré que les approches traditionnelles de prévention des fuites de données, qu'elles reposent sur des expressions régulières, des techniques de fingerprinting, des modèles d'apprentissage automatique ou des mécanismes classiques de SIEM, présentent une limite fondamentale : elles considèrent la sensibilité comme une propriété intrinsèque du contenu, sans être capables d'identifier l'origine réelle des données observées. Afin de répondre à cette limitation, nous avons proposé une approche de l'analyse dynamique de contamination fondée sur la provenance, capable de suivre dynamiquement la propagation de données sensibles à travers plusieurs composants hétérogènes d'un système informatique. L'approche développée repose sur la corrélation de journaux issus des bases de données, des systèmes de fichiers et des communications HTTP afin de reconstruire la provenance des données sans recourir à une instrumentation intrusive des applications. Cette stratégie permet d'associer chaque donnée détectée à sa source réelle et de distinguer des valeurs identiques selon leur contexte d'origine. Les résultats expérimentaux obtenus démontrent que l'intégration de la provenance améliore significativement la précision de la détection et permet

de réduire drastiquement les faux positifs générés par les approches basées uniquement sur le contenu. Dans les scénarios étudiés, le système proposé parvient à différencier des données syntaxiquement identiques provenant de bases de sensibilité différente, ce qui constitue une limitation majeure des méthodes traditionnelles. Les expérimentations montrent également que l'approche reste applicable dans des contextes réalistes impliquant des exportations SQL, des exfiltrations via HTTP ainsi qu'une intégration avec le SIEM Wazuh pour la génération d'alertes contextualisées et exploitables en temps réel. Au-delà des performances obtenues, ce travail met en évidence une évolution importante dans la manière d'aborder la détection des fuites de données. Les résultats soutiennent l'idée que la sensibilité d'une information ne peut pas être déterminée de manière fiable uniquement à partir de son contenu, mais qu'elle doit être interprétée à partir de son origine et de son contexte de propagation. Dans des environnements modernes où des valeurs identiques peuvent coexister dans plusieurs systèmes avec des niveaux de sensibilité différents, la provenance devient un élément essentiel pour lever les ambiguïtés et améliorer la fiabilité des mécanismes de détection. Enfin, plusieurs perspectives de recherche demeurent ouvertes. L'amélioration de la scalabilité du système, l'intégration de mécanismes de corrélation plus avancés, la prise en charge de nouveaux vecteurs d'exfiltration ainsi que l'automatisation des mécanismes de réponse constituent des prolongements naturels de ces travaux. Plus généralement, cette recherche ouvre la voie à des systèmes de protection des données davantage centrés sur la provenance et capables de répondre aux exigences croissantes des environnements informatiques contemporains.

BIBLIOGRAPHIE

[1] K. Ahmed, Y. Wang, M. Lis, et J. Rubin, “Vialin : Path-aware dynamic taint analysis for android,” dans *Proceedings of the 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, 2023, p. 1598–1610.

[2] A. M. Alashjee, S. Duraibi, et J. Song, “Dynamic taint analysis tools : A review.” *International Journal of Computer Science and Security (IJCSS)*, vol. 13, n°. 6, p. 231–244, 2019.

[3] M. Alecci, J. Samhi, M. Miltenberger, S. Arzt, T. F. Bissyandé, et J. Klein, “Damflow : Preventing a flood of irrelevant data flows in android apps,” *ACM Transactions on Software Engineering and Methodology*, 2025.

[4] S. Alneyadi, E. Sithirasenan, et V. Muthukkumarasamy, “Detecting data semantic : a data leakage prevention approach,” dans *2015 IEEE Trustcom/BigDataSE/ISPA*, vol. 1. IEEE, 2015, p. 910–917.

[5] N. Alturayeif et J. Hassine, “Data leakage detection in machine learning code : transfer learning, active learning, or low-shot prompting ?” *PeerJ Computer Science*, vol. 11, p. e2730, 2025.

[6] S. Boulesnane, L. Bouzidi, et C. Varinard, “Rgpd et e-administration : besoins, pratiques et défis,” 2020. [En ligne]. Disponible : https://api.semanticscholar.org/CorpusID:219419554

[7] T. Brindha et R. Shaji, “An analysis of data leakage and prevention techniques in cloud environment,” dans *2015 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT)*. IEEE, 2015, p. 350–355.

[8] P. Buneman, S. Khanna, et W.-C. Tan, “Why and where : A characterization of data provenance,” dans *Proceedings of the 8th International Conference on Database Theory (ICDT)*. Springer, 2001, p. 316–330.

[9] S. Cela, A. Ciancone, P. Gustafsson, Á. Hajdu, Y. Jia, T. Kapus, M. Koshtenko, W. Lewis, K. Mao, et D. Martac, "Automated end-to-end dynamic taint analysis for whatsapp," dans *Companion Proceedings of the 32nd ACM International Conference on the Foundations of Software Engineering*, 2024, p. 21–26.

[10] C. Chew, A. Parikh, S. Chatiwala, A. Chen, S. Mercado, et B. Liu, "Taintmonkey : Dynamic taint analysis of python web applications using monkey patching," dans *2025 IEEE MIT Undergraduate Research Technology Conference (URTC)*. IEEE, 2025, p. 1–5.

[11] I. P. T. ENTERPRISE, "Nist privacy framework : A tool for improving privacy through enterprise risk management," 2020.

[12] J. Freire, D. Koop, E. Santos, et C. T. Silva, "Provenance for computational tasks : A survey," *Computing in Science & Engineering*, vol. 10, n°. 3, p. 11–21, 2008.

[13] B. Glavic et K. Dittrich, "Data provenance : A categorization of existing approaches," dans *Datenbanksysteme in Business, Technologie und Web (BTW 2007)–12. Fachtagung des GI-Fachbereichs" Datenbanken und Informationssysteme"(DBIS)*. Gesellschaft für Informatik e. V., 2007, p. 227–241.

[14] B. Glavic et K. R. Dittrich, "Data provenance : A categorization of existing approaches," *IEEE Data Engineering Bulletin*, vol. 34, n°. 3, p. 5–14, 2015.

[15] G. González-Granadillo, S. González-Zarzosa, et R. Diaz, "Security information and event management (siem) : analysis, trends, and usage in critical infrastructures," *Sensors*, vol. 21, n°. 14, p. 4759, 2021.

[16] H. Huang, R. Yan, et J. Gao, "Semtaint : A scalable taint analysis approach for javaweb frameworks and composite containers," *Computers & Security*, p. 104821, 2026.

[17] Z. Ji, Y. Shen, J. Lin, et K. R. Koedinger, "Enhancing the de-identification of personally identifiable information in educational data," *arXiv preprint arXiv :2501.09765*, 2025.

[18] R. Joeres, D. B. Blumenthal, et O. V. Kalinina, "Data splitting to avoid information leakage with datasail," *Nature Communications*, vol. 16, n°. 1, p. 3337, 2025.

[19] K. Kaur, I. Gupta, et A. K. Singh, "Data leakage prevention : e-mail protection via gateway," dans *Journal of Physics : Conference Series*, vol. 933, n°. 1. IOP Publishing, 2017, p. 012013.

[20] P. Kayalvizhi, J. Jayaprakash, C. Dharanidharan, et P. Balasunil, "Ai-based document leak attribution system," dans *2026 World Conference on Computational Science and Technology (WcCST)*. IEEE, 2026, p. 742–746.

[21] A. Khraisat, I. Gondal, P. Vamplew, et J. Kamruzzaman, “Survey of intrusion detection systems : techniques, datasets and challenges,” *Cybersecurity*, vol. 2, n°. 1, p. 1–22, 2019.

[22] C. Liu et M. Babar, “Corporate cybersecurity risk and data breaches : A systematic review of empirical research,” *Australian Journal of Management*, 11 2024.

[23] J. Liu, A. C. Myers, et L. Zheng, “Fine-grained information flow control for relational databases,” dans *Proceedings of the ACM SIGMOD International Conference on Management of Data*. ACM, 2015, p. 1019–1034.

[24] X. M. Liu, “A risk-based approach to cybersecurity : a case study of financial messaging networks data breaches,” *The Coastal Business Journal*, vol. 18, n°. 1, p. 2, 2021.

[25] X. Liu, D. Yao, et E. Bertino, “Information flow analysis using context-sensitive dynamic taint analysis,” *IEEE Transactions on Dependable and Secure Computing*, vol. 12, n°. 5, p. 543–555, 2015.

[26] Y. Lu, X. Huang, Y. Ma, et M. Ma, “A weighted context graph model for fast data leak detection,” dans *2018 IEEE International Conference on Communications (ICC)*. IEEE, 2018, p. 1–6.

[27] G. Martino, F. A. Galatolo, M. G. Cimino, C. Callegari *et al.*, “Using deep learning with attention to detect data exfiltration by pos malware.” dans *ICEIS (1)*, 2023, p. 638–648.

[28] M. Meli, M. R. McNiece, et B. Reaves, “How bad can it git ? characterizing secret leakage in public github repositories.” dans *NDSS*, 2019.

[29] L. G. Michael, J. Donohue, J. C. Davis, D. Lee, et F. Servant, “Regexes are hard : Decision-making, difficulties, and risks in programming regular expressions,” dans *2019 34th IEEE/ACM International Conference on Automated Software Engineering (ASE)*. IEEE, 2019, p. 415–426.

[30] D. Muktadir-Al-Mukit et M. H. Ali, “The dynamics of stock market responses following the cyber-attacks news : Evidence from event study,” *Information Systems Frontiers*, p. 1–18, 2025.

[31] J. Newsome et D. Song, “Dynamic taint analysis for automatic detection, analysis, and signature generation of exploits on commodity software,” dans *Proceedings of the Network and Distributed System Security Symposium (NDSS)*. Internet Society, 2005.

[32] R. Roman, J. Lopez, et M. Mambo, “Mobile edge computing, fog and cloud computing : A survey on security and privacy issues,” *Future Generation Computer Systems*, vol. 78, p. 680–698, 2018.

[33] R. Roman, J. Zhou, et J. Lopez, “On the features and challenges of data leakage prevention systems,” *Computer Networks*, vol. 57, n°. 6, p. 1348–1364, 2018.

[34] K. Sashwin, K. Nithika, S. Priyadharshini, S. Maduvant *et al.*, “Analysis, trends, and utilization of security information and event management (siem) in critical infrastructures,” dans *2024 10th International Conference on Advanced Computing and Communication Systems (ICACCS)*, vol. 1. IEEE, 2024, p. 1980–1984.

[35] E. J. Schwartz, T. Avgerinos, et D. Brumley, “All you ever wanted to know about dynamic taint analysis and forward symbolic execution (but might have been afraid to ask),” dans *2010 IEEE Symposium on Security and Privacy*. IEEE, 2010, p. 317–331.

[36] ——, “All you ever wanted to know about dynamic taint analysis and forward symbolic execution (but might have been afraid to ask),” dans *Proceedings of the IEEE Symposium on Security and Privacy*. IEEE, 2010, p. 317–331.

[37] A. Shabtai, Y. Elovici, et L. Rokach, “A survey of data leakage detection and prevention solutions (2012),” 2012.

[38] ——, “Data leakage detection/prevention solutions,” dans *A Survey of Data Leakage Detection and Prevention Solutions*. Springer, 2012, p. 17–37.

[39] ——, “A survey of data leakage prevention systems,” *Security and Communication Networks*, vol. 5, n°. 6, p. 529–553, 2012.

[40] Y. Shapira, B. Shapira, et A. Shabtai, “Content-based data leakage detection using extended fingerprinting,” *arXiv preprint arXiv :1302.2028*, 2013.

[41] M. Sheeraz, M. A. Paracha, M. U. Haque, M. H. Durad, S. M. Mohsin, S. S. Band, et A. Mosavi, “Effective security monitoring using efficient siem architecture,” *Hum.-Centric Comput. Inf. Sci*, vol. 13, p. 1–18, 2023.

[42] X. Shu, D. Yao, et E. Bertino, “Privacy-preserving detection of sensitive data exposure,” *IEEE transactions on information forensics and security*, vol. 10, n°. 5, p. 1092–1103, 2015.

[43] A. Skatova, R. McDonald, S. Ma, et C. Maple, “Unpacking privacy : Valuation of personal data protection,” *Plos one*, vol. 18, n°. 5, p. e0284581, 2023.

[44] S. Syarova, S. Toleva-Stoimenova, A. Kirkov, S. Petkov, et K. Traykov, “Data leakage prevention and detection in digital configurations : survey,” dans *ENVIRONMENT. TECHNOLOGY. RESOURCES. Proceedings of the International Scientific and Practical Conference*, vol. 2, 2024, p. 253–258.

[45] P. Voigt et A. Von dem Bussche, *The EU General Data Protection Regulation (GDPR) : A Practical Guide*. Cham : Springer International Publishing, 2017.

[46] Y. Wu et A. Bartel, “Confusetaint : Exploiting vulnerabilities to bypass dynamic taint analysis,” dans *Proceedings of the 40th IEEE/ACM International Conference on Automated Software Engineering*, 2025.

[47] Y. Xiaohui, S. Yubo, C. Fei *et al.*, “Android’s sensitive data leakage detection based on api monitoring,” dans *International Conference on Cyberspace Technology (CCT 2014)*. IET, 2014, p. 1–4.

[48] T. Yang, H. Cui, S. Niu, et P. Zhang, “An analysis on sensitive data passive leakage in android applications,” dans *2015 IEEE 16th International Conference on Communication Technology (ICCT)*. IEEE, 2015, p. 125–131.

[49] P. N. Yeboah, A. Kayes, W. Rahayu, E. Pardede, et S. Mahbub, “A systematic literature survey of machine learning approaches to cyber data breach detection : Current research issues and future directions,” *Computer Networks*, p. 112179, 2026.

[50] J. Zhao, Y. Li, Y. Zou, Z. Liang, Y. Xiao, Y. Li, B. Peng, N. Zhong, X. Wang, W. Wang, et W. Huo, “Leveraging semantic relations in code and data to enhance taint analysis of embedded systems,” dans *33rd USENIX Security Symposium (USENIX Security 24)*. Philadelphia, PA : USENIX Association, Aug. 2024, p. 7067–7084. [En ligne]. Disponible : https://www.usenix.org/conference/usenixsecurity24/presentation/zhao

[51] Y. Zheng, J. Liu, K. Ren, et L. Xiong, “B2mark : A blind and buyer-traceable watermarking scheme for tabular datasets,” *Proceedings of the ACM on Management of Data*, vol. 3, n^{o}. 4, p. 1–26, 2025.

[52] D. Zhu, J. Jung, D. Song, T. Kohno, et D. Wetherall, “Tainteraser : Protecting sensitive data leaks using application-level taint tracking,” *ACM SIGOPS Operating Systems Review*, vol. 45, n^{o}. 1, p. 142–154, 2011.