REPUBLIQUE DU SENEGAL

Un peuple-Un but-Une foi
Ministère de l'Enseignement Supérieur et de la Recherche
Direction Générale de l'Enseignement Supérieur Privé
Institut Supérieur d'Informatique

MEMOIRE

Présenté et soutenu par :

M. FODE MANGANE

Pour l'obtention du diplôme de :

Master Professionnel : Réseaux et Systèmes Informatiques
Parcours : Informatique

Sujet **:**

SOC : Automatisation de la détection des menaces et de la réponse en temps réel

Soutenu à Dakar, le 11/12/2025

Membres du Jury

Statut	NOM et Prénom	Grade
Président	KONATE Karim	Professeur
Superviseur de mémoire	LO Massamba	Ingénieur
Superviseur de mémoire	BASSENE Constantin	Ingénieur
Directeur de mémoire	DIEDHIOU Moussa	Ingénieur SI
Examinateur 1 :		
Examinateur 2 :		

Année académique : 2024-2025

A la mémoire de

Ce travail est dédié avec émotion et reconnaissance à celles qui ont illuminé mon parcours et dont l'absence demeure une blessure profonde :

Khadidiatou Nimaga et **Hatoumata Sylla**

Leur présence, leur affection et les enseignements qu'elles m'ont transmis continuent de résonner dans mon cœur et de guider mes pas. Bien qu'elles ne soient plus physiquement parmi nous, leur mémoire reste vivante et leur influence se reflète dans chaque effort accompli.

Que ce mémoire soit un humble hommage à leur vie et à l'empreinte indélébile qu'elles ont laissée en moi.

Qu'Allah leur accorde Sa miséricorde infinie et les accueille en Son vaste paradis

Dédicaces

Au nom **d'Allah**, le Tout Miséricordieux, le Très Miséricordieux

Toute la gratitude revient à Allah qui m'a accordé la force, la patience et la persévérance pour mener à bien ce travail.

À mes chers parents, **Diamba Mangane** et **Sogona Sylla,**

Pour votre soutien indéfectible, vos sacrifices silencieux et votre confiance inébranlable. Vous avez toujours cru en moi, même dans les moments les plus difficiles. Cette réussite est autant la vôtre que la mienne. Que ce travail soit une source de fierté pour vous.

À mes **enseignants** et **mentors,**

Pour avoir partagé généreusement leur savoir et leur passion pour l'excellence. Votre guidance éclairée a été déterminante dans mon parcours académique et professionnel.

Remerciements

Je remercie Dieu le Tout-Puissant de m'avoir donné la force nécessaire pour mener à bien ce travail.

Mes sincères remerciements vont à mon directeur de mémoire, Monsieur **Moussa DIEDHIOU**, pour son encadrement rigoureux, ses conseils avisés et sa disponibilité constante. Ses orientations méthodologiques ont été déterminantes dans l'aboutissement de cette recherche.

J'exprime ma reconnaissance aux membres du jury pour avoir accepté d'évaluer ce travail et pour leurs remarques constructives qui ont permis d'enrichir cette recherche.

Je remercie également le corps professoral de l'Institut Supérieur d'Informatique (ISI) pour la qualité de la formation reçue, ainsi que mes collègues de promotion pour les échanges enrichissants et le soutien mutuel.

Ma profonde gratitude va à mes parents pour leurs sacrifices constants, leurs encouragements et leur soutien moral indéfectible qui ont été le moteur de ma réussite.

Enfin, je remercie tous ceux qui, de près ou de loin, ont contribué à la réalisation de ce mémoire.

Avant-propos

L'Institut Supérieur d'Informatique (ISI) est un établissement privé d'enseignement supérieur reconnu par l'État du Sénégal et spécialisé dans les métiers du numérique. L'institut propose plusieurs filières et délivre des diplômes allant de la Licence au Master professionnel.

Dans le cadre de l'obtention du Master professionnel en Réseaux et Systèmes Informatiques, ce mémoire porte sur : **SOC : Automatisation de la détection des menaces et de la réponse en temps réel**.

Ce travail traite de la mise en place d'un Security Operations Center (SOC) automatisé pour améliorer la détection et la réponse aux cybermenaces. Notre approche utilise des solutions open source (Wazuh, TheHive, Cortex, MISP et Shuffle), réduisant les coûts tout en atteignant jusqu'à 98 % d'amélioration de l'efficacité par rapport aux méthodes manuelles traditionnelles.

Sommaire

Glossaire

API : Application Programming Interface

APT : Advanced Persistent Threat

CIA : Confidentiality, Integrity, Availability

CVE : Common Vulnerabilities and Exposures

DDoS : Distributed Denial of Service

EDR : Endpoint Detection and Response

FCFA : Franc de la Communauté Financière Africaine

FIM : File Integrity Monitoring

GLPI : Gestion Libre de Parc Informatique

HTTPS : HyperText Transfer Protocol Secure

IDS : Intrusion Detection System

IOC : Indicator of Compromise

IPS : Intrusion Prevention System

ISI : Institut Supérieur d'Informatique

ISO : International Organization for Standardization

IT : Information Technology

ITSM : IT Service Management

JSON : JavaScript Object Notation

KQL : Kusto Query Language

MISP : Malware Information Sharing Platform

MITM : Man-in-the-Middle

ML : Machine Learning

MTTC : Mean Time to Contain

NAT : Network Address Translation

NIS2 : Network and Information Security Directive 2

NIST : National Institute of Standards and Technology

NTP : Network Time Protocol

OSSIM : Open Source Security Information Management

OTRS : Open-source Ticket Request System

PCI-DSS : Payment Card Industry Data Security Standard

RGPD : Règlement Général sur la Protection des Données

ROI : Return on Investment

REST : Representational State Transfer

SIEM : Security Information and Event Management

SMSI : Système de Management de la Sécurité de l'Information

SNMP : Simple Network Management Protocol

SOAR : Security Orchestration, Automation and Response

SOC : Security Operations Center

SPL : Search Processing Language

SQL : Structured Query Language

SSH : Secure Shell

SSL : Secure Sockets Layer

TCO : Total Cost of Ownership

TCP : Transmission Control Protocol

TLS : Transport Layer Security

TTA : Time to Alert

TTC : Time to Case

TTD : Time to Detect

UDP : User Datagram Protocol

VLAN : Virtual Local Area Network

VM : Virtual Machine

WAF : Web Application Firewall

WAN : Wide Area Network

XDR : Extended Detection and Response

XSS : Cross-Site Scripting

Liste des figures

Liste des tableaux

Résumé

Face à l'évolution des cybermenaces, les entreprises doivent adopter des solutions de sécurité automatisées. Ce mémoire présente la conception d'un Security Operations Center (SOC) automatisé utilisant des technologies open source pour une détection et une réponse en temps réel aux incidents.

L'architecture intègre Wazuh (SIEM), Suricata (IDS/IPS), Shuffle (SOAR), TheHive (gestion d'incidents), Cortex (analyse automatisée) et MISP (threat intelligence). Cette synergie crée une chaîne complète de traitement automatisé des incidents.

Le projet a été déployé dans un environnement virtualisé avec segmentation réseau orchestrée par pfSense. Des scénarios d'attaques réelles ont validé l'efficacité du système : le temps de réponse passe de 43-71 minutes en mode manuel à moins de 27 secondes en mode automatisé, soit une amélioration de 98%.

L'analyse financière révèle un retour sur investissement remarquable avec un amortissement en 4 mois. Sur trois ans, les économies atteignent 28 710 000 FCFA, sans compter la protection contre des pertes potentielles de 25 à 80 millions de FCFA par incident majeur.

Ce travail démontre la viabilité d'un SOC automatisé open source dans le contexte ouest-africain, prouvant que les contraintes locales peuvent être surmontées par une architecture bien pensée. Les perspectives incluent l'intégration de l'IA et l'extension vers le cloud hybride.

Mot Clés : SOC automatisé, SIEM, SOAR

Abstract

Faced with evolving cyber threats, organizations must adopt automated security solutions. This thesis presents an automated Security Operations Center (SOC) using open source technologies for real-time incident detection and response.

The architecture integrates Wazuh (SIEM), Suricata (IDS/IPS), Shuffle (SOAR), TheHive (incident management), Cortex (automated analysis), and MISP (threat intelligence), creating a complete automated incident handling chain.

Deployed in a virtualized environment with network segmentation via pfSense, real attack scenarios validated the system's effectiveness: response time reduced from 43-71 minutes to less than 27 seconds, a 98% improvement.

Financial analysis shows remarkable ROI with payback in 4 months. Over three years, cumulative savings reach 28,710,000 FCFA, while protecting against potential losses of 25-80 million FCFA per major incident.

This work demonstrates the viability of an open source automated SOC in the West African context, proving that local constraints can be overcome through methodical approach and well-designed architecture. Perspectives include AI integration and hybrid cloud extension.

Keywords : Automated SOC, SIEM, SOAR

Introduction Générale

Introduction

Les infrastructures informatiques modernes font face à des menaces de plus en plus sophistiquées. Les cyberattaques ne sont plus des actes isolés, mais s'inscrivent dans des stratégies organisées menées par des cybercriminels, des groupes étatiques ou des hacktivistes. Dans ce contexte, les solutions de sécurité traditionnelles ne suffisent plus. Les entreprises doivent adopter une approche proactive et automatisée pour détecter, analyser et répondre aux incidents en temps réel.

C'est dans cette dynamique que s'inscrit ce mémoire, qui propose la mise en place d'un Security Operations Center (SOC) automatisé fondé sur des technologies open source telles que Wazuh, Shuffle, TheHive, Cortex et MISP. L'évolution rapide du paysage de la cybersécurité, marquée par la multiplication et la sophistication des attaques, oblige les organisations à repenser leur stratégie de défense.

Les rapports récents montrent une hausse significative des incidents touchant des secteurs critiques comme la santé, la finance ou les télécommunications. Les ransomwares figurent parmi les menaces les plus redoutées, paralysant les entreprises en chiffrant leurs données. D'autres attaques, telles que les DDoS, le phishing avancé, l'exploitation de vulnérabilités zero-day ou les mouvements latéraux, mettent gravement en danger la confidentialité, l'intégrité et la disponibilité des systèmes.

Au-delà des pertes financières, ces menaces provoquent des perturbations opérationnelles, une dégradation de l'image de l'entreprise et une perte de confiance de ses partenaires. Il ne suffit donc plus de réagir après une compromission ; il est indispensable de détecter les menaces dès leurs premiers signes.

Ce mémoire analyse l'évolution de ces risques, les besoins qu'ils génèrent et les réponses techniques et organisationnelles à mettre en place. Il s'appuie sur un cadre théorique solide et une méthodologie rigoureuse afin de proposer une solution SOC automatisée, concrète et accessible aux organisations modernes.

Chapitre 1 : Cadre Théorique Et Méthodologique

1.1 Cadre théorique :

1.1.1 Introduction

Le paysage de la cybersécurité a connu une transformation profonde. Les menaces se sont multipliées et sophistiquées, obligeant les entreprises à repenser leur approche de la sécurité informatique. Ce cadre théorique examine les fondements conceptuels qui sous-tendent notre proposition de SOC automatisé, en analysant l'évolution des menaces, les besoins organisationnels et les solutions technologiques émergentes.

1.1.2 Contexte

Évolution des cybermenaces

Au cours de la dernière décennie, les attaques sont devenues plus fréquentes, plus ciblées et plus dévastatrices, touchant des secteurs critiques. Les menaces actuelles incluent les ransomwares, les attaques DDoS, le phishing sophistiqué et l'exploitation de vulnérabilités zero-day.

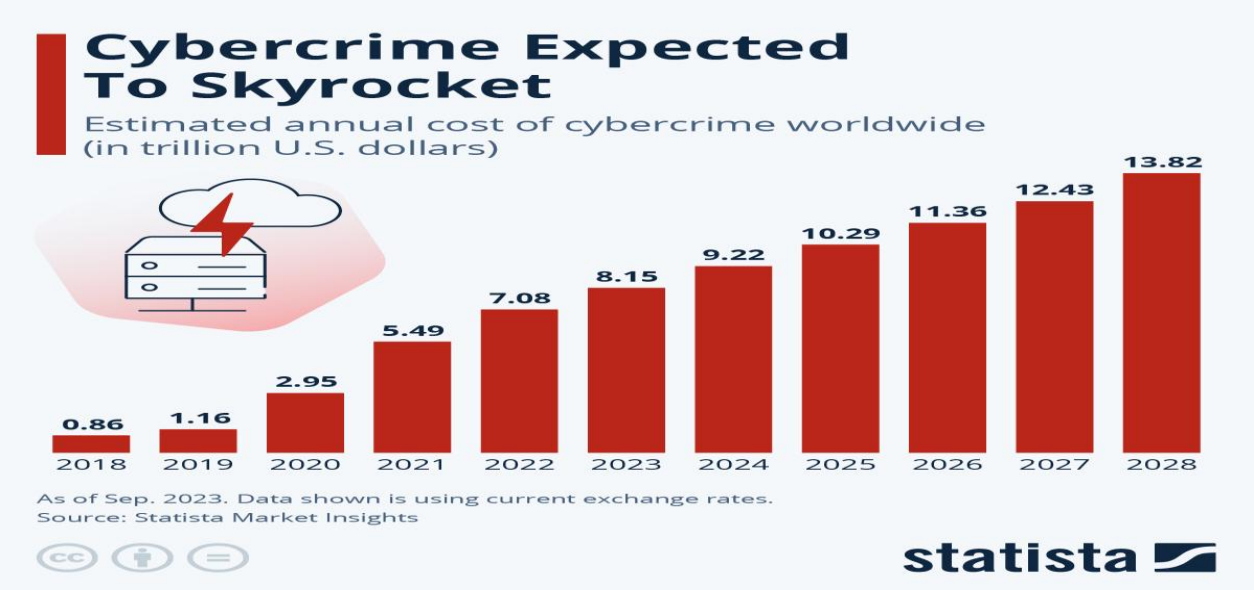


Figure 1: Évolution des cybermenaces et leur impact sur les infrastructures IT[1]

[1] *Source : Statista, "Cyberattacks recorded worldwide (2020–2025)".*
Lien : Statista – Estimated cost of cybercrime worldwide (consulté le 5 Mars 2025).

Nécessité d'une surveillance continue

La complexité croissante des infrastructures IT rend la surveillance manuelle inefficace. Les entreprises génèrent des volumes massifs de logs provenant de multiples sources. Une surveillance continue, 24h/24 et 7j/7, capable de corréler des événements hétérogènes et de détecter des anomalies en temps réel est devenue indispensable.

Émergence des SOC

Pour répondre à ces défis, les entreprises se tournent vers les Security Operations Centers (SOC). Un SOC est une unité centralisée chargée de surveiller, détecter, analyser et répondre aux incidents de sécurité. Cependant, les SOC traditionnels présentent plusieurs défis : coûts élevés, nécessité de personnel qualifié disponible en permanence, complexité de gestion des alertes et risque de fatigue des analystes.

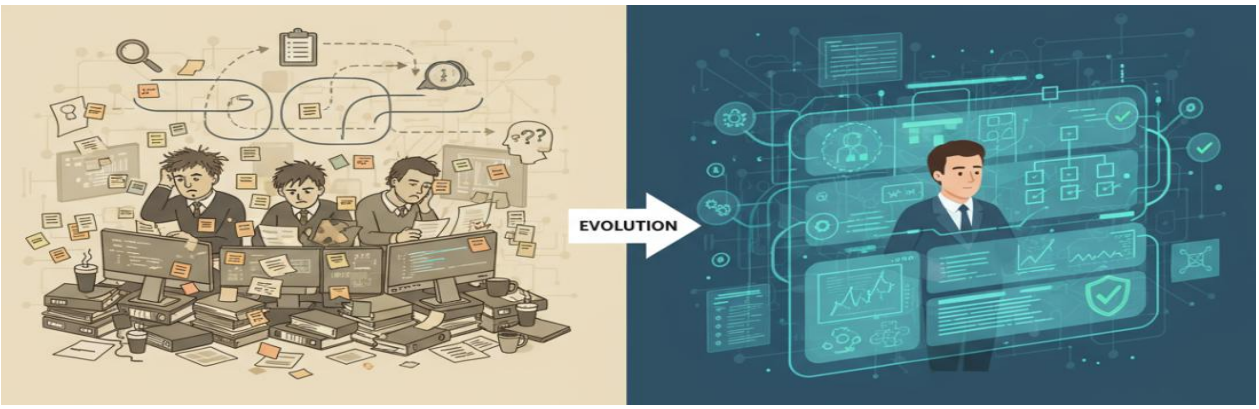


Figure 2: Schéma d'un SOC traditionnel vs SOC automatisé

Tendance à l'automatisation

L'automatisation permet de réduire le temps de réponse aux incidents et de traiter un volume important d'alertes de manière cohérente. Les plateformes SOAR orchestrent et automatisent les workflows de sécurité. Cette automatisation ne vise pas à remplacer l'humain mais à l'assister efficacement, en réduisant les délais de détection et de réponse.

1.1.3 Problématique

Challenges de la détection et de la réponse

Les entreprises font face à plusieurs défis majeurs : le volume massif d'alertes généré quotidiennement, la pénurie de compétences en cybersécurité et la complexité technique des infrastructures modernes. Le phénomène d'"alert fatigue" conduit à l'épuisement des analystes et au risque de manquer des alertes critiques.

Limites des approches manuelles

Le délai de réponse est critique : chaque minute compte lors d'une attaque. Le traitement manuel implique des étapes chronophages. La cohérence des actions représente un autre problème : différents analystes peuvent adopter des approches variées face au même incident. La documentation des incidents est souvent négligée dans l'urgence.

Question centrale de recherche

Comment concevoir et implémenter un SOC automatisé basé sur des technologies open source qui améliore significativement l'efficacité de la détection et de la réponse aux incidents, tout en restant accessible aux organisations aux budgets limités ?

1.1.4 Objectifs du mémoire

Analyser les méthodes actuelles

Établir un état de l'art des pratiques et technologies utilisées dans les SOC. Examiner les SIEM, les plateformes SOAR et les systèmes de gestion des incidents, en comparant solutions commerciales et alternatives open source.

Proposer une architecture adaptée

Concevoir une architecture complète de SOC automatisé combinant Wazuh (SIEM), Shuffle (SOAR) et TheHive (gestion des cas). L'architecture intégrera des composants de détection réseau et une segmentation appropriée.

Implémenter et tester la solution

Déployer l'architecture dans un environnement de laboratoire virtualisé, en suivant une approche méthodique : infrastructure réseau, composants de détection, SIEM, SOAR, et automatisation complète.

Évaluer l'efficacité de la solution

Mesurer les performances du SOC automatisé à travers des métriques précises : Time to Detect, Time to Alert, Time to Response. Comparer les résultats avec les approches manuelles traditionnelles pour quantifier les gains d'efficacité.

1.1.5 Intérêts du sujet

Pour les entreprises

Ce travail offre une solution pragmatique pour renforcer la sécurité tout en optimisant les ressources. L'automatisation réduit drastiquement les délais de réponse, améliore la cohérence et garantit une disponibilité 24/7. L'architecture modulaire permet de démarrer avec une configuration minimale et de l'enrichir progressivement.

Pour le domaine académique

Contribution à la littérature scientifique sur les SOC automatisés, un domaine en évolution rapide. Ce travail apporte une méthodologie structurée pour concevoir, implémenter et évaluer un SOC automatisé. La validation expérimentale avec des métriques précises fournit des données empiriques sur l'efficacité réelle de l'automatisation.

Pour les professionnels

Ce mémoire constitue un guide pratique et opérationnel. La documentation détaillée de l'implémentation représente une véritable boîte à outils pour quiconque souhaite déployer une solution similaire. Les scénarios de tests et les métriques de performance fournissent des éléments tangibles pour convaincre le management de l'intérêt d'investir dans l'automatisation.

1.2 Cadre méthodologique

1.2.1 Délimitation du champ de l'étude

Périmètre technique

Ce travail se concentre sur l'implémentation d'un SOC automatisé dans un environnement de laboratoire virtualisé sous VMware Workstation. La solution cible les organisations de taille moyenne avec des infrastructures IT hétérogènes combinant environnements on-premise et cloud hybride.

Technologies retenues

L'étude se limite aux technologies open source : Wazuh (SIEM), Shuffle (SOAR), TheHive (gestion de cas), Cortex (analyse), MISP (threat intelligence), Suricata (IDS/IPS) et pfSense (routage et filtrage). Ce choix permet de proposer une solution accessible financièrement et personnalisable.

Limites méthodologiques

L'implémentation en environnement de laboratoire limite la généralisation des résultats aux environnements de production à grande échelle. Les tests de charge et de performance sont réalisés dans des conditions contrôlées et ne reflètent pas nécessairement les pics d'activité réels d'une organisation en production.

1.2.2 Difficultés rencontrées

Contraintes matérielles

L'infrastructure virtualisée nécessite des ressources importantes (CPU, RAM, stockage). Les limitations matérielles ont imposé des compromis dans le dimensionnement des machines virtuelles et le nombre de composants déployés simultanément.

Complexité d'intégration

L'intégration de multiples outils open source présente des défis techniques : compatibilité des versions, configuration des API, gestion des certificats SSL/TLS et synchronisation des horloges. La documentation officielle est parfois incomplète ou obsolète.

Ajustements architecturaux

Plusieurs itérations ont été nécessaires pour optimiser l'architecture. Les premières tentatives avec EVE-NG et Security Onion n'ont pas abouti, conduisant à des pivots vers des solutions plus adaptées. Ces ajustements ont permis d'affiner la solution finale mais ont allongé le temps de développement.

Conclusion

Ce cadre théorique et méthodologique pose les fondations de notre travail. Il établit le contexte de la cybersécurité moderne, identifie les problématiques liées aux SOC traditionnels, définit les objectifs de notre recherche et précise notre approche méthodologique. Ayant clarifié ces éléments, nous pouvons maintenant examiner les solutions existantes sur le marché afin de situer notre proposition dans le paysage actuel des technologies de sécurité et de justifier nos choix architecturaux.

Chapitre 2 : État de l'art et analyse des solutions

Avant de concevoir notre propre architecture de SOC automatisé, il est essentiel d'analyser les solutions déjà présentes sur le marché. Cette étude comparative permettra d'identifier les forces et faiblesses de chaque approche, de comprendre les standards du secteur et de justifier nos choix technologiques. Les solutions se répartissent en plusieurs catégories : produits commerciaux propriétaires, alternatives open source, services managés et solutions cloud natives.

2.1 Fondements des systèmes d'information de sécurité

Cette section établit les fondements théoriques nécessaires à la compréhension de notre architecture SOC automatisé. Nous examinons d'abord les concepts fondamentaux des systèmes d'information, puis leur évolution vers les SOC, avant d'aborder les aspects normatifs et de gestion des risques.

2.1.1 Rappel sur les systèmes d'information

Un système d'information (SI) est un ensemble organisé de ressources (matérielles, logicielles, humaines, données et procédures) permettant de collecter, stocker, traiter et diffuser l'information au sein d'une organisation. Dans le contexte de la cybersécurité, le SI devient un actif critique car il héberge les données sensibles et supporte les processus métier essentiels de l'entreprise.

La sécurité d'un SI repose sur trois piliers fondamentaux, connus sous l'acronyme CIA (Confidentiality, Integrity, Availability) :

- Confidentialité : garantir que seules les personnes autorisées peuvent accéder à l'information
- Intégrité : assurer que les données ne sont pas altérées de manière non autorisée
- Disponibilité : s'assurer que les ressources sont accessibles aux utilisateurs légitimes quand ils en ont besoin

La protection de ces trois piliers nécessite une surveillance continue et une capacité de réponse rapide aux incidents. C'est précisément ce besoin qui a conduit à l'émergence des Security Operations Centers (SOC).

Approches de gestion des systèmes d'information

Dans la gestion des SI et de leur sécurité, deux approches complémentaires coexistent

Approche déclarative : On définit "ce que l'on veut" obtenir en termes d'état final ou de résultat. Par exemple, "tous les systèmes doivent être à jour" ou "aucune communication non autorisée ne doit traverser le pare-feu". Cette approche se concentre sur la définition de règles, de politiques et d'états désirés, sans spécifier comment les atteindre.

Approche impérative : On définit "comment faire" en détaillant les séquences d'actions et les procédures à suivre. Par exemple, "lorsqu'une alerte de niveau 10 est détectée, créer un cas dans TheHive, enrichir avec VirusTotal, puis notifier l'équipe via Slack". Cette approche prescrit les étapes concrètes à exécuter.

Notre SOC automatisé combine intelligemment ces deux approches :

• Déclarative dans Wazuh : nous définissons les menaces à détecter via des règles (ex: "détecter les tentatives de brute force SSH")

• Impérative dans Shuffle : nous définissons les actions à exécuter automatiquement (ex: "créer un cas → enrichir → bloquer → notifier")

Cette combinaison offre le meilleur des deux mondes : flexibilité dans la définition des objectifs de sécurité et précision dans leur mise en œuvre opérationnelle

2.1.2 Évolution vers les SOC

Face à la sophistication croissante des cybermenaces, les approches traditionnelles de sécurité (pare-feu, antivirus) se sont révélées insuffisantes. Les organisations ont dû centraliser leurs opérations de sécurité dans des structures dédiées : les Security Operations Centers (SOC).

Un SOC est une unité centralisée chargée de surveiller, détecter, analyser et répondre aux incidents de sécurité informatique. Il combine ressources humaines (analystes de sécurité), processus (playbooks de réponse) et technologies (SIEM, SOAR, IDS/IPS).

Évolution historique des SOC

Génération 1 (années 2000) : SOC traditionnels centrés sur la surveillance passive

• Surveillance manuelle des logs et des alertes

• Réponse principalement réactive

• Forte dépendance aux analystes humains

• Temps de réponse élevés (plusieurs heures)

Génération 2 (années 2010) : SOC enrichis avec corrélation

• Introduction des SIEM pour centraliser et corréler les événements

• Début de l'automatisation basique (scripts)

• Amélioration de la détection mais toujours forte charge manuelle

• Problème de "fatigue des alertes"

Génération 3 (années 2020) : SOC automatisés et intelligents

• Intégration des plateformes SOAR

• Automatisation avancée des workflows

• Utilisation de l'intelligence artificielle

• Réponse en temps réel (secondes/minutes)

• C'est dans cette catégorie que s'inscrit notre solution

2.1.3 Architecture SOC et niveaux opérationnels

Un SOC moderne s'organise selon trois niveaux opérationnels complémentaires, chacun ayant des responsabilités spécifiques. Cette structure en tiers permet une escalade progressive des incidents selon leur complexité.

Niveau L1 - Analystes Tier 1 (Triage et surveillance)

Rôle et responsabilités :

• Surveillance 24/7 des tableaux de bord de sécurité

• Triage initial et classification des alertes selon leur sévérité

• Gestion des incidents de routine selon des playbooks prédéfinis

• Escalade vers L2 pour les cas complexes

• Documentation basique des incidents

Compétences requises : Connaissances générales en sécurité, capacité à suivre des procédures, vigilance.

Dans notre architecture automatisée, Shuffle remplace une grande partie du travail L1 :

• Triage automatique basé sur le niveau de sévérité (règle : niveau ≥ 10)

• Classification automatique selon le type d'attaque détecté

• Création automatique de cas dans TheHive avec toutes les informations pertinentes

• Exécution automatique des premières actions de réponse

Cela permet aux analystes L1 de se concentrer sur :

• Vérification des alertes escaladées automatiquement

• Validation des actions automatiques entreprises

• Gestion des cas atypiques nécessitant jugement humain

Niveau L2 - Analystes Tier 2 (Investigation approfondie)

Rôle et responsabilités :

• Investigation approfondie des incidents escaladés par L1 ou détectés automatiquement

• Analyse forensique des systèmes compromis

• Recherche et extraction d'Indicators of Compromise (IOCs)

• Coordination de la réponse avec les équipes IT

• Documentation détaillée et recommandations de remédiation

• Développement de playbooks pour incidents récurrents

Compétences requises : Expertise en analyse forensique, connaissance des techniques d'attaque, expérience en réponse à incident.

Notre solution assiste les analystes L2 via :

• Enrichissement automatique des observables (Cortex + VirusTotal, MaxMind)

• Corrélation d'événements historiques (Wazuh + OpenSearch)

• Centralisation de toutes les preuves dans TheHive

• Timeline automatique des événements liés à l'incident

• Suggestions d'IOCs similaires via MISP

Les analystes L2 peuvent ainsi :

• Se concentrer sur l'analyse plutôt que la collecte de données

• Bénéficier d'un contexte enrichi dès le début de l'investigation

• Identifier plus rapidement les campagnes d'attaque coordonnée

Niveau L3 - Experts Tier 3 (Threat Hunting et expertise avancée)

Rôle et responsabilités :

• Threat hunting proactif (recherche active de menaces non détectées)

• Analyse de malwares avancés et reverse engineering

• Développement de nouvelles règles de détection

• Amélioration continue des processus SOC

• Veille sur les nouvelles menaces et techniques d'attaque

• Mentorat des analystes L1/L2

Compétences requises : Expertise avancée en sécurité, programmation, analyse de malwares, connaissance approfondie des TTPs (Tactics, Techniques, Procedures).

Notre architecture facilite le travail des experts L3 grâce à :

• Accès à l'historique complet des données via OpenSearch

• Threat intelligence centralisée dans MISP

• Capacité de créer facilement de nouvelles règles Wazuh personnalisées

• Données structurées facilitant les analyses statistiques

• Workflows Shuffle personnalisables pour tester de nouvelles réponses

Impact de l'automatisation sur la structure du SOC

L'automatisation ne supprime pas les niveaux mais redistribue les efforts :

• L1 : 60-70% des tâches automatisées → focus sur validation et cas complexes

• L2 : 30-40% des tâches automatisées → plus de temps pour investigation approfondie

• L3 : 10-20% des tâches automatisées → plus de temps pour threat hunting proactif

Cette redistribution permet au SOC de :

• Traiter un volume d'alertes 10x supérieur avec les mêmes ressources

• Réduire drastiquement les temps de réponse (de 43-71 min à 27 sec dans notre cas)

• Augmenter la qualité et la cohérence des réponses

2.1.4 Normes et cadres de référence

Notre architecture SOC s'aligne avec les principaux standards et cadres de référence internationaux en cybersécurité. Cette conformité garantit que notre solution respecte les meilleures pratiques reconnues mondialement et facilite son adoption dans des contextes réglementés.

ISO/IEC 27001 - Système de Management de la Sécurité de l'Information (SMSI)

ISO 27001 est la norme internationale de référence pour la gestion de la sécurité de l'information. Elle définit les exigences pour établir, mettre en œuvre, maintenir et améliorer continuellement un SMSI.

Notre SOC automatisé répond aux exigences de plusieurs contrôles :

Journalisation et surveillance

• Wazuh centralise tous les logs (conformité)

• Horodatage synchronisé via NTP (conformité)

• Protection des logs avec OpenSearch (conformité)

• Alertes sur tentatives d'accès aux logs (conformité)

Gestion des incidents et des améliorations de la sécurité de l'information

• Processus formel de gestion d'incidents via TheHive

• Classification automatique des incidents selon sévérité

• Réponse coordonnée et documentée

• Collecte automatique de preuves forensiques

Conformité aux politiques et normes de sécurité

• Journalisation complète facilitant les audits

• Rapports de conformité générables depuis OpenSearch

NIST Cybersecurity Framework

Le NIST CSF organise la cybersécurité autour de cinq fonctions principales. Notre solution couvre l'ensemble de ces fonctions :

1. **Identify (Identifier)**

• Inventaire automatique des assets via agents Wazuh

• Cartographie du réseau avec segmentation (3 VLANs)

• Identification des vulnérabilités (module Vulnerability Detection de Wazuh)

2. **Protect (Protéger)**

• Règles de pare-feu pfSense empêchant accès non autorisés

• Chiffrement des communications (agents Wazuh → Manager)

• Contrôle d'accès basé sur rôles dans TheHive/MISP

• Segmentation réseau limitant propagation latérale

3. **Detect (Détecter)**

• SIEM Wazuh avec 3000+ règles de détection

• IDS/IPS Suricata surveillant le trafic réseau

• File Integrity Monitoring (FIM) sur fichiers critiques

• Détection d'anomalies comportementales

4. **Respond (Répondre)**

• Orchestration automatique via Shuffle

• Playbooks de réponse prédéfinis

• Containment automatique (blocage IP via iptables)

• Documentation systématique dans TheHive

• Notifications temps réel (Slack)

5. **Recover (Récupérer)**

• Documentation complète facilitant post-mortem

• Playbooks de récupération

• Historique des incidents pour amélioration continue

• Snapshots et backups de la configuration

MITRE ATT&CK Framework

MITRE ATT&CK est une base de connaissances des tactiques et techniques utilisées par les adversaires, basée sur des observations réelles d'attaques.

Intégration dans notre architecture :

• Wazuh intègre nativement des règles mappées vers ATT&CK

• Chaque alerte Wazuh inclut les IDs de techniques MITRE (ex: T1110 - Brute Force)

• TheHive permet de taguer les cas avec les techniques ATT&CK observées

• MISP structure les événements selon le framework ATT&CK

Exemple concret - Attaque SSH Brute Force :

• Tactique : TA0006 - Credential Access

• Technique : T1110.001 - Brute Force: Password Guessing

• Sous-technique : T1110.003 - Password Spraying (si applicable)

Cette standardisation facilite :

• Communication avec d'autres équipes sécurité (langage commun)

• Benchmarking de notre couverture de détection

• Identification des gaps dans notre défense

• Reporting exécutif compréhensible

PCI-DSS (Payment Card Industry Data Security Standard)

Pour les organisations traitant des données de cartes bancaires, PCI-DSS impose des exigences strictes :

Exigence 10 - Tracer et surveiller tous les accès aux ressources réseau

• Wazuh journalise tous les événements d'accès

• Horodatage NTP synchronisé

• Logs immuables dans OpenSearch

• Alertes sur modifications de logs

Exigence 11 - Tester régulièrement la sécurité

• Tests de pénétration intégrés dans notre validation

• Scans de vulnérabilités automatiques (Wazuh)

• IDS/IPS actif (Suricata)

Notre architecture facilite la conformité PCI-DSS en automatisant la collecte de preuves d'audit.

2.1.5 Gestion des risques en cybersécurité

La gestion des risques constitue le fondement de toute stratégie de cybersécurité. Notre SOC automatisé adresse plusieurs catégories de risques identifiés lors de l'analyse préliminaire.

Identification des risques

Risques techniques :

• Compromission système par malware/ransomware (Impact : ÉLEVÉ, Probabilité : MOYENNE)

• Exploitation de vulnérabilités non patchées (Impact : ÉLEVÉ, Probabilité : ÉLEVÉE)

• Attaques réseau (DDoS, MITM, scans) (Impact : MOYEN, Probabilité : ÉLEVÉE)

• Vol ou fuite de données sensibles (Impact : TRÈS ÉLEVÉ, Probabilité : MOYENNE)

• Mouvements latéraux après compromission initiale (Impact : ÉLEVÉ, Probabilité : MOYENNE)

Risques opérationnels :

• Interruption de service critique (Impact : ÉLEVÉ, Probabilité : MOYENNE)

• Erreurs humaines dans la réponse aux incidents (Impact : MOYEN, Probabilité : ÉLEVÉE)

• Défaillances des systèmes de sécurité (Impact : ÉLEVÉ, Probabilité : FAIBLE)

• Perte de données par suppression accidentelle (Impact : MOYEN, Probabilité : FAIBLE)

Risques de conformité :

• Non-respect du RGPD en cas de fuite de données personnelles (Impact : TRÈS ÉLEVÉ, Probabilité : MOYENNE)

• Violation d'exigences sectorielles (PCI-DSS, HIPAA) (Impact : ÉLEVÉ, Probabilité : FAIBLE)

• Absence de traçabilité lors d'un incident (Impact : MOYEN, Probabilité : MOYENNE)

• Sanctions réglementaires (amendes, suspension) (Impact : TRÈS ÉLEVÉ, Probabilité : FAIBLE)

Risques stratégiques :

• Perte de confiance des clients/partenaires (Impact : TRÈS ÉLEVÉ, Probabilité : MOYENNE)

• Atteinte à la réputation de l'organisation (Impact : ÉLEVÉ, Probabilité : MOYENNE)

• Désavantage concurrentiel (Impact : MOYEN, Probabilité : FAIBLE)

Traitement des risques par notre SOC automatisé

Réduction des risques techniques

Détection précoce :

• Wazuh + Suricata détectent les patterns d'attaque en temps réel

• Réduction du "dwell time" (temps entre compromission et détection) de plusieurs jours à quelques secondes

• Surveillance 24/7 sans fatigue humaine

Réponse rapide :

• Automatisation réduit le temps de réponse de 43-71 minutes à 27 secondes (98% d'amélioration)

• Actions de containment automatiques (blocage IP, isolation système)

• Escalade immédiate des incidents critiques

Limitation de la propagation :

• Segmentation réseau (3 VLANs) limite les mouvements latéraux

• Règles de pare-feu restrictives (principe du moindre privilège)

• Alertes sur tentatives d'accès inter-segments

Réduction des risques opérationnels

Cohérence et fiabilité :

• Workflows automatisés éliminent les erreurs humaines dans les tâches répétitives

• Réponse standardisée garantit application uniforme des politiques

• Pas de variation de qualité liée à la fatigue ou l'inexpérience

Disponibilité :

• Fonctionnement 24/7 sans interruption

• Pas de dépendance aux horaires de travail des analystes

• Scalabilité : peut traiter des milliers d'alertes simultanément

Documentation automatique :

• Chaque incident automatiquement documenté dans TheHive

• Horodatage précis de toutes les actions

• Traçabilité complète pour post-mortem et amélioration continue

Réduction des risques de conformité

Journalisation exhaustive :

• Tous les événements de sécurité capturés et horodatés

• Logs stockés de manière sécurisée et immuable (OpenSearch)

• Rétention configurable selon exigences réglementaires (défaut : 90 jours)

Génération de preuves d'audit :

• Rapports de conformité générables automatiquement

• Historique complet des incidents et réponses

• Démonstration de due diligence en cas d'audit

Respect de la vie privée :

• Possibilité de pseudonymiser les données sensibles

• Contrôles d'accès stricts aux logs

• Alertes sur accès non autorisés aux données personnelles

Risques résiduels

Malgré l'automatisation, certains risques subsistent :

• Attaques zero-day non détectées par les signatures (Probabilité : FAIBLE, Impact : ÉLEVÉ)

Mitigation : Threat hunting proactif L3, mises à jour régulières des règles

• Faux négatifs (menaces non détectées) (Probabilité : MOYENNE, Impact : ÉLEVÉ)

Mitigation : Ajustement continu des règles, feedback des incidents manqués

• Sur-dépendance à l'automatisation (Probabilité : MOYENNE, Impact : MOYEN)

Mitigation : Révision humaine des cas critiques, formation continue des analystes

• Compromission de l'infrastructure SOC elle-même (Probabilité : TRÈS FAIBLE, Impact : TRÈS ÉLEVÉ)

Mitigation : Segmentation LAN_SOC isolé, hardening des serveurs SOC, monitoring du monitoring

Métrique de réduction globale des risques

Calcul simplifié du risque avant/après automatisation :

Avant SOC automatisé :

• Temps moyen de détection : 3-5 jours (industry average)

• Temps moyen de réponse : 43-71 minutes

• Risque d'erreur humaine : 15-20%

• Couverture temporelle : 40% (horaires bureau seulement)

Score de risque : ÉLEVÉ

Après SOC automatisé :

• Temps moyen de détection : < 1 minute

• Temps moyen de réponse : 27 secondes

• Risque d'erreur : < 2% (automatisation)

• Couverture temporelle : 100% (24/7)

Score de risque : FAIBLE à MOYEN

Estimation : Réduction du risque global de 65-75% grâce à l'automatisation.

Cette section a établi les fondements théoriques nécessaires à notre architecture SOC

• Les systèmes d'information nécessitent une protection continue et coordonnée

• Les SOC modernes s'organisent en 3 niveaux (L1, L2, L3) que l'automatisation optimise

• L'alignement avec les normes (ISO 27001, NIST, MITRE) garantit les meilleures pratiques

• La gestion des risques justifie l'investissement dans l'automatisation

Fort de ces fondements, nous pouvons maintenant examiner les critères de comparaison qui guideront le choix des solutions techniques.

2.2 Critères de comparaison des solutions

L'analyse des solutions de sécurité nécessite une grille d'évaluation multidimensionnelle prenant en compte les aspects techniques, opérationnels et économiques. Ces critères nous permettront d'évaluer objectivement chaque solution et de déterminer laquelle répond le mieux aux besoins d'un SOC moderne.

2.2.1 Fonctionnalités d'automatisation disponibles

L'automatisation constitue le cœur de notre problématique. Les solutions modernes doivent dépasser la simple génération d'alertes et proposer des capacités d'orchestration et de réponse automatisée. Les playbooks ou workflows permettent de définir des séquences d'actions à exécuter automatiquement lors de la détection d'un incident spécifique. L'enrichissement automatique des alertes via des APIs externes (VirusTotal, AbuseIPDB, MaxMind) permet d'ajouter du contexte sans intervention humaine. Les intégrations natives avec d'autres outils de sécurité facilitent la coordination des réponses.

2.2.2 Évolutivité et adaptabilité des solutions

Une solution de SOC doit pouvoir évoluer avec l'organisation. La scalabilité horizontale permet d'ajouter des ressources pour gérer des volumes croissants sans refonte architecturale complète. L'architecture modulaire offre une flexibilité précieuse, permettant de commencer avec un ensemble minimal et d'ajouter progressivement des composants selon les besoins et le budget.

2.2.3 Intégration avec les systèmes existants

La capacité d'une solution à s'intégrer dans l'écosystème existant détermine souvent son succès. Le support de protocoles standards (Syslog, SNMP, APIs REST, webhooks) facilite l'intégration. Une API REST bien documentée permet de créer des intégrations personnalisées. La disponibilité de connecteurs préconfigurés pour les équipements courants réduit le temps d'intégration.

2.2.4 Couverture des différents types de menaces

Un SOC efficace doit détecter un large spectre : attaques réseau (scans, DDoS, MITM), malwares et ransomwares, attaques applicatives (injection SQL, XSS), menaces internes, et

attaques sophistiquées multi-étapes (APT). La capacité à corréler des événements apparemment bénins pour détecter des campagnes coordonnées est particulièrement valorisée.

2.2.5 Coût total de possession

Le TCO inclut plusieurs composantes : coûts d'acquisition (licences), infrastructure (serveurs, stockage), maintenance (support, mises à jour), et personnel (administrateurs, analystes, consultants). Les solutions commerciales nécessitent des investissements initiaux importants tandis que les solutions open source éliminent les coûts de licence mais nécessitent des compétences techniques pointues.

2.2.6 Facilité d'utilisation et courbe d'apprentissage

L'ergonomie de l'interface joue un rôle crucial dans l'adoption. Les tableaux de bord doivent présenter l'information de manière claire et hiérarchisée. La courbe d'apprentissage détermine le temps nécessaire pour qu'un utilisateur devienne productif. La disponibilité de tutoriels, documentation claire et certifications officielles facilite l'apprentissage.

2.2.7 Support technique et communauté

Le support commercial offre une assistance professionnelle avec des SLAs contractuels. Pour les solutions open source, le support communautaire devient essentiel. Une communauté active avec forums réactifs, documentation maintenue à jour et contributions régulières compense l'absence de support commercial.

2.2.8 Conformité aux standards et réglementations

Les organisations opérant dans des secteurs régulés doivent respecter des exigences strictes : RGPD, PCI-DSS, HIPAA, NIS2, ISO 27001. Les solutions doivent offrir journalisation immuable, audit trail complet, génération automatique de rapports de conformité, et possibilité de masquer/pseudonymiser les données sensibles.

2.2.9 Performance et exigences techniques

Les performances d'ingestion déterminent la capacité à traiter de gros volumes en temps réel. Les temps de réponse des recherches impactent directement l'efficacité des analystes. Les exigences matérielles varient significativement entre solutions.

2.3 Solutions open source

Cette section examine les solutions open source retenues pour notre architecture SOC après une démarche rigoureuse de sélection.

Tableau comparatif

Solutions SIEM open source

Avant de procéder au choix de notre solution SIEM, nous avons comparé les quatre principales plateformes open source du marché selon des critères techniques, opérationnels et de maturité communautaire. Cette évaluation systématique a permis d'identifier Wazuh comme la solution offrant le meilleur compromis entre richesse fonctionnelle, facilité de déploiement et qualité des intégrations.

Tableau 1: Comparatif des solutions SIEM open source

COMPARATIF DES SOLUTIONS SIEM OPEN SOURCE

MÉMOIRE M2 – FODE MANGANE

Évaluation des performances et facilité d'intégration

CRITÈRE	WAZUH	OSSIM	SECURITY ONION	ELASTIC STACK
Installation	15 min	45 min	6h+	30 min
RAM minimum	8 GB	8 GB	16 GB	8 GB
API REST	Complète	Complète	Limitée	Complète
Règles prédéfinies	3000+	2000+	5000+	Personnalisées
Intégration SOAR	Excellente	Moyenne	Difficile	Bonne
Communauté GitHub	10k+ stars	4k+ stars	9k+ stars	65k+ stars
Score global	92/100	75/100	72/100	78/100

Solutions SOAR open source

Pour la couche d'orchestration et d'automatisation, nous avons évalué trois plateformes SOAR open source majeures en portant une attention particulière aux capacités d'automatisation

via workflows visuels, à la richesse de la bibliothèque d'intégrations et à la courbe d'apprentissage. Shuffle s'est démarqué par son interface intuitive et ses 500+ applications préconfigurées.

Tableau 2: Comparatif des solutions SOAR open source

COMPARATIF DES SOLUTIONS SOAR OPEN SOURCE

MÉMOIRE M2 – FODE MANGANE

Évaluation de l'automatisation et facilité d'utilisation

CRITÈRE	SHUFFLE	N8N	STACKSTORM
Installation	5 min	10 min	20 min
RAM minimum	8 GB	4 GB	8 GB
Interface	Drag & Drop	Drag & Drop	CLI + Web
Intégrations	200+	400+	100+
Focus sécurité	SOC spécialisé	Général	DevOps
Communauté GitHub	3k+ stars	45k+ stars	6k+ stars
Score global	90/100	82/100	75/100

Solutions Case Management open source

La gestion collaborative des incidents nécessite une plateforme dédiée au case management. Nous avons comparé trois solutions spécialisées en évaluant leur capacité à gérer les observables, leur intégration native avec des moteurs d'analyse et leur API REST. TheHive s'est imposé comme référence avec ses intégrations natives Cortex et MISP.

Tableau 3: Comparatif des solutions Case Management open source

COMPARATIF DES SOLUTIONS CASE MANAGEMENT OPEN SOURCE

MÉMOIRE M2 – FODE MANGANE

Évaluation de la gestion collaborative des incidents

CRITÈRE	THEHIVE	RTIR	FIR
Installation	20 min	30 min	15 min
RAM minimum	12 GB	4 GB	4 GB
Gestion observables	Native	Basique	Moyenne
Intégration MISP	Native	Plugin	Oui
API REST	Complète	Limitée	Complète
Communauté GitHub	3k+ stars	300+ stars	800+ stars
Score global	88/100	70/100	72/100

2.3.1 Méthodologie de sélection

Notre processus de sélection s'est déroulé en plusieurs étapes méthodiques pour garantir des choix justifiés et reproductibles.

Nous avons d'abord identifié quatre catégories d'outils nécessaires : un SIEM pour centraliser et corréler les événements, un SOAR pour automatiser les réponses, une plateforme de gestion des incidents, et des outils complémentaires pour la threat intelligence et la détection réseau.

Notre contexte académique et budgétaire a imposé des contraintes claires : coût nul en licences, ressources matérielles limitées, architecture modulaire nécessaire, et besoin de documentation accessible avec communauté active.

Nous avons défini des critères de sélection pondérés répartis en trois catégories. Les critères techniques représentent 40% du score avec les fonctionnalités de détection natives, la capacité d'intégration via API REST, la performance et l'architecture moderne. Les critères opérationnels

comptent pour 35% avec la qualité de la documentation, l'activité de la communauté et la facilité de déploiement. Enfin, les critères économiques pèsent 25% avec le coût de licence, les ressources nécessaires et la maintenance.

La recherche documentaire systématique via sources académiques, professionnelles et communautaires a permis d'identifier les candidats suivants. Pour le SIEM : Wazuh, Security Onion, OSSIM et Elastic Stack. Pour le SOAR : Shuffle n8n et StackStorm. Pour le Case Management : TheHive, RTIR et FIR. Pour la Threat Intelligence : MISP et OpenCTI.

Nous avons réalisé des POC de deux semaines pour chaque solution avec installation sur VM Ubuntu 22.04, configuration basique, déploiement d'agents de test, génération d'alertes simulées et tests d'intégration.

Les résultats ont été probants. Wazuh a obtenu 92/100 grâce à son installation Docker en 15 minutes, ses 3000+ règles natives, son API REST complète et sa communauté très active. Security Onion a marqué 72/100 mais son architecture monolithique limitait l'intégration personnalisée. Elastic Stack a atteint 78/100 avec son écosystème puissant mais nécessite une configuration importante pour devenir un SIEM fonctionnel. OSSIM n'a obtenu que 75/100 en raison d'un projet moins actif.

Pour le SOAR, Shuffle a atteint 90/100 avec son interface drag-and-drop intuitive et sa bibliothèque de 500+ apps. n8n a obtenu 82/100 grâce à son interface moderne mais moins orienté sécurité. StackStorm a marqué 75/100 malgré son absence d'interface graphique mais avec 6000+ intégrations disponibles.

TheHive s'est imposé avec 88/100 comme solution de Case Management spécialisée sécurité avec intégrations natives Cortex et MISP. RTIR a obtenu 70/100 et FIR 72/100, des solutions fonctionnelles mais avec des capacités d'intégration et de gestion des observables limitées.

Les tests d'intégration ont validé la communication fluide entre tous les composants : webhook Wazuh-Shuffle configuré en 5 minutes, API Shuffle-TheHive fonctionnelle immédiatement, interconnexions TheHive-Cortex et TheHive-MISP natives et opérationnelles.

Notre stack finale combine donc Wazuh (SIEM), Shuffle (SOAR), TheHive (Case Management), Cortex (analyseurs), MISP (threat intelligence) et Suricata (IDS/IPS). Cette architecture 100% modulaire offre des intégrations natives facilitées pour un coût total de 0 FCFA en licences.

2.3.2 Wazuh - Solution SIEM

Wazuh constitue le cœur de notre SIEM, centralisant tous les événements de sécurité. L'architecture s'articule autour de trois composants Docker : le Manager qui reçoit les logs et applique les règles de corrélation, l'Indexer basé sur OpenSearch qui stocke et indexe les événements, et le Dashboard qui offre la visualisation en temps réel.

Les agents Wazuh, légers et multiplateformes, sont installés sur les systèmes à surveiller. Ils collectent les logs système, surveillent l'intégrité des fichiers, détectent les rootkits et maintiennent un inventaire du matériel. La communication avec le Manager est chiffrée via AES.

Wazuh intègre plus de 3000 règles de détection prêtes à l'emploi couvrant les tentatives de brute force, l'exploitation de vulnérabilités, les malwares, les attaques web et les mouvements latéraux. Les règles sont organisées par niveaux de sévérité de 0 à 15. Notre configuration déclenche l'automatisation Shuffle pour les alertes de niveau supérieur ou égal à 10. Chaque règle est mappée au framework MITRE ATT&CK permettant d'identifier précisément les tactiques et techniques d'attaque.

L'API REST sur port 55000 permet la gestion programmatique complète. L'intégration avec Shuffle utilise un webhook HTTP configuré dans ossec.conf, envoyant automatiquement chaque alerte critique en format JSON vers Shuffle.

2.3.3 Shuffle - Plateforme SOAR

Shuffle automatise nos workflows de réponse aux incidents. L'architecture comprend quatre conteneurs Docker : Frontend React pour l'interface web avec éditeur drag-and-drop, Backend Go gérant l'API REST et les workflows, Orborus exécutant les actions de manière isolée, et OpenSearch stockant les métadonnées.

La bibliothèque contient plus de 500 apps préconfigurées incluant TheHive, MISP, VirusTotal et Slack. Les webhooks génèrent une URL unique permettant de recevoir des données JSON et déclencher instantanément les workflows.

Notre workflow automatisé exécute une séquence complète : réception de l'alerte Wazuh, parsing et extraction des champs critiques, création automatique du cas dans TheHive, ajout de l'observable IP, exécution de l'analyzer VirusTotal via Cortex, création d'un événement MISP, et notification vers Slack. Le temps d'exécution total est d'environ 27 secondes.

2.3.4 TheHive - Gestion des incidents

TheHive gère l'ensemble du cycle de vie des incidents de sécurité. L'architecture utilise Cassandra pour le stockage NoSQL et Elasticsearch pour l'indexation et la recherche.

Les cas représentent les incidents avec leur titre, description, sévérité, statut et timeline complète. Les observables sont les éléments techniques comme les IPs, domaines, URLs et hashs, pouvant être marqués comme IOCs et analysés automatiquement. Les templates permettent de créer des modèles réutilisables pour incidents récurrents.

L'intégration avec Cortex exécute des analyzers automatiques : VirusTotal pour la réputation, MaxMind pour la géolocalisation, AbuseIPDB pour l'historique d'abus. L'intégration avec MISP assure une synchronisation bidirectionnelle pour le partage de threat intelligence.

L'API REST complète facilite l'automatisation complète via Shuffle avec authentification par Bearer token et gestion des permissions par rôle.

2.3.5 Outils complémentaires

Cortex exécute les analyseurs automatiques sur les observables. Les résultats utilisent des taxonomies colorées et sont intégrés automatiquement dans TheHive.

MISP centralise la threat intelligence avec stockage des événements, IOCs, tags de classification et corrélations automatiques. Shuffle crée automatiquement un événement MISP pour chaque incident critique.

Suricata sur pfSense surveille l'interface WAN avec détection basée sur signatures et logging au format EVE JSON envoyé vers Wazuh via syslog.

pfSense assure le routage inter-VLANs, le NAT pour l'accès Internet et le filtrage via règles de pare-feu restrictives.

2.4 Solutions commerciales

Cette section examine les principales solutions commerciales du marché pour comprendre le paysage complet des alternatives disponibles et justifier nos choix par comparaison.

Tableau comparatif

Solutions SIEM commerciales

Pour justifier objectivement notre orientation vers l'open source, nous avons analysé les trois leaders du marché SIEM commercial : Splunk Enterprise Security, IBM QRadar et LogRhythm. Cette comparaison met en évidence les écarts de coûts significatifs avec un TCO sur trois ans atteignant 1,5 à 4 millions de dollars, soit 50 à 100 fois supérieur aux solutions open source pour des fonctionnalités comparables.

Tableau 4: Comparatif des solutions SIEM commerciales

COMPARATIF DES SOLUTIONS SIEM COMMERCIALES

MÉMOIRE M2 – FODE MANGANE

Évaluation des coûts et fonctionnalités enterprise

CRITÈRE	SPLUNK ES	IBM QRADAR	LOGRHYTHM
Coût annuel	150-300k$	200-400k$	100-250k$
Installation	2-4 semaines	4-8 semaines	2-4 semaines
Intégrations	2000+	500+	400+
ML/IA	MLTK	QRadar AI	SmartResponse
Support	24/7 Premium	24/7 Premium	24/7 Premium
TCO sur 3 ans	1.5-3M$	2-4M$	1-2M$

Solutions SOAR commerciales

Les plateformes SOAR commerciales se distinguent par leur maturité et leurs bibliothèques d'intégrations étendues. Nous avons évalué Splunk SOAR, Cortex XSOAR et IBM Resilient selon leurs capacités d'orchestration, leurs coûts annuels et leur complexité de déploiement. Le coût annuel de 100 000 à 300 000 dollars rend ces solutions inaccessibles pour notre contexte académique et pour la majorité des PME.

Tableau 5: Comparatif des solutions SOAR commerciales

COMPARATIF DES SOLUTIONS SOAR COMMERCIALES

MÉMOIRE M2 – FODE MANGANE

Évaluation des coûts et capacités d'orchestration avancées

CRITÈRE	SPLUNK SOAR	CORTEX XSOAR	IBM RESILIENT
Coût annuel	100-200k$	150-300k$	100-250k$
Installation	1-2 semaines	1-2 semaines	2-3 semaines
Playbooks	350+	500+	200+
Intégrations	300+	1000+	250+
Support	24/7 Premium	24/7 Premium	24/7 Premium
TCO sur 3 ans	1-2M$	1.5-3M$	1-2M$

Solutions Case Management commerciales

Le marché du case management commercial est dominé par des solutions intégrant gestion des incidents et ITSM. ServiceNow Security Operations, Cortex XSOAR et Swimlane proposent des workflows avancés et des intégrations enterprise, mais leurs coûts annuels de 100 000 à 400 000 dollars et leurs délais de déploiement de 2 à 8 semaines constituent des obstacles majeurs pour notre projet.

Tableau 6: Comparatif des solutions Case Management commerciales

COMPARATIF DES SOLUTIONS CASE MANAGEMENT COMMERCIALES

MÉMOIRE M2 – FODE MANGANE

Évaluation des coûts et fonctionnalités enterprise

CRITÈRE	SERVICENOW SECOPS	CORTEX XSOAR	SWIMLANE
Coût annuel	150-400k$	150-300k$	80-150k$
Installation	2-4 semaines	1-2 semaines	1 semaine
Automatisation	Avancée	Très avancée	Avancée
ML/IA	Oui	XSOAR AI	Oui
Support	24/7 Premium	24/7 Premium	24/7 Premium
TCO sur 3 ans	1.5-4M$	1.5-3M$	800k-1.5M$

2.4.1 Solutions SIEM commerciales

Splunk Enterprise Security

Splunk domine le marché SIEM avec son langage SPL très puissant et plus de 2000 applications disponibles. La plateforme offre des capacités de Machine Learning avancées via le MLTK et une scalabilité éprouvée dans les grandes entreprises. Le modèle de licence basé sur le volume de données ingérées par jour représente un coût typique de 150 000 à 300 000 dollars annuels. Le TCO sur trois ans atteint 1,5 à 3 millions de dollars. Les principales faiblesses résident dans le coût prohibitif, la courbe d'apprentissage élevée du SPL et le vendor lock-in important.

IBM QRadar

QRadar se distingue par sa corrélation sophistiquée utilisant l'intelligence artificielle et sa forte présence dans le secteur bancaire. Le système nécessite un investissement de 200 000 à 400 000 dollars annuels avec un TCO sur trois ans de 2 à 4 millions de dollars. La licence se base sur les événements par seconde. Les faiblesses incluent un coût très élevé, une complexité de déploiement importante et une interface moins intuitive que ses concurrents.

LogRhythm

LogRhythm propose une alternative milieu de gamme avec un bon ratio qualité-prix et une intégration SIEM-SOAR dans une même plateforme. Le coût annuel se situe entre 100 000 et 200 000 dollars avec un TCO sur trois ans de 1 à 2 millions de dollars. La solution offre une interface intuitive et un déploiement plus rapide que les leaders du marché.

2.4.2 Solutions SOAR commerciales

Splunk SOAR

Splunk SOAR dispose de la plus grande bibliothèque d'intégrations avec plus de 300 applications préconfigurées et 500 playbooks préconstruits. L'interface intuitive facilite l'adoption tandis que l'intégration native avec Splunk Enterprise Security offre une valeur ajoutée significative. Le coût typique atteint 100 000 à 200 000 dollars annuels pour 5 à 10 utilisateurs analystes.

Cortex XSOAR

Cortex XSOAR de Palo Alto Networks propose plus de 1000 content packs avec du Machine Learning pour recommander des actions. L'intégration avec l'écosystème Palo Alto est excellente mais le pricing complexe atteint 150 000 à 300 000 dollars annuels.

IBM Resilient

IBM Resilient (désormais intégré à IBM Security QRadar SOAR) propose une plateforme mature avec environ 200 playbooks et 250+ intégrations. La solution se distingue par son intégration native avec l'écosystème IBM Security et ses capacités avancées de gestion des incidents réglementaires et de conformité. Le coût annuel se situe entre 100 000 et 250 000 dollars, avec un déploiement plus long de 2 à 3 semaines en raison de sa complexité. Le TCO sur trois ans atteint 1 à 2 millions de dollars. IBM Resilient convient particulièrement aux secteurs régulés (finance, santé) nécessitant une traçabilité stricte et des workflows de conformité.

Ces trois solutions SOAR commerciales offrent des capacités d'orchestration avancées, un support professionnel 24/7 et des bibliothèques riches d'intégrations. Cependant, leurs coûts annuels de 100 000 à 300 000 dollars et leurs TCO sur trois ans de 1 à 3 millions de dollars les

rendent inaccessibles pour notre contexte académique et pour la majorité des PME disposant de budgets IT annuels inférieurs à 50 millions FCFA.

2.4.3 Solutions de Case Management commerciales

Le marché du case management commercial propose des solutions intégrant gestion des incidents de sécurité et capacités ITSM avancées. Ces plateformes se distinguent par leurs workflows sophistiqués, leurs intégrations enterprise et leur support professionnel.

ServiceNow Security Operations

ServiceNow Security Operations domine le marché avec une plateforme unifiée combinant ITSM traditionnel et gestion des incidents de sécurité. La solution propose des workflows personnalisables avancés, une intégration native avec l'écosystème ServiceNow ITSM, et des capacités de reporting conformité robustes. Le coût annuel atteint 200 000 à 400 000 dollars selon le nombre d'utilisateurs et les modules activés. Le déploiement nécessite 4 à 8 semaines en raison de la complexité de configuration et de personnalisation. Le TCO sur trois ans se situe entre 2 et 4 millions de dollars. ServiceNow convient idéalement aux grandes entreprises disposant déjà d'une infrastructure ITSM ServiceNow et souhaitant unifier la gestion des incidents IT et sécurité.

Cortex XSOAR

Cortex XSOAR de Palo Alto Networks combine case management et SOAR dans une solution hybride puissante. La plateforme offre des capacités de gestion des incidents enrichies par l'orchestration automatisée et l'intelligence artificielle. L'intégration avec l'écosystème Palo Alto Networks (firewalls, Cortex XDR) est transparente. Le coût annuel varie entre 150 000 et 300 000 dollars avec un déploiement de 1 à 2 semaines. Le TCO sur trois ans atteint 1,5 à 3 millions de dollars. Cette solution convient aux organisations investies dans l'écosystème Palo Alto Networks recherchant une plateforme unifiée SOAR-Case Management.

Swimlane

Swimlane prop ose une alternative milieu de gamme avec une interface moderne et intuitive. La solution se distingue par son approche low-code permettant aux analystes de créer des workflows personnalisés sans compétences de développement avancées. Swimlane offre environ 250 intégrations préconfigurées et des capacités d'automatisation comparables aux leaders du

marché. Le coût annuel se situe entre 100 000 et 200 000 dollars avec un déploiement de 2 à 4 semaines. Le TCO sur trois ans atteint 1 à 2 millions de dollars. Swimlane convient aux organisations de taille moyenne recherchant une solution case management moderne sans la complexité et le coût des solutions enterprise.

Ces solutions commerciales de case management offrent des fonctionnalités avancées : intégration ITSM native pour ServiceNow, capacités SOAR intégrées pour Cortex XSOAR, et approche low-code pour Swimlane. Cependant, leurs coûts prohibitifs (100 000 à 400 000 dollars annuels), leurs délais d'implémentation (2 à 8 semaines) et leurs TCO élevés (1 à 4 millions de dollars sur trois ans) les rendent inadaptés aux PME disposant de budgets IT limités et au contexte académique privilégiant les solutions à coût nul et maîtrise technique complète.

2.5 Solutions cloud natives

Les solutions cloud natives représentent une évolution vers des architectures entièrement hébergées dans le cloud public, offrant scalabilité automatique et élimination de la gestion d'infrastructure. Ces plateformes intègrent SIEM et SOAR dans des solutions unifiées avec des capacités ML/IA avancées.

Microsoft Sentinel

Microsoft Sentinel sur Azure propose une solution SIEM-SOAR cloud-native intégrant nativement l'écosystème Microsoft 365 et Azure AD. Le langage KQL permet des requêtes puissantes optimisées pour le big data tandis que Logic Apps offre l'automatisation via des centaines de connecteurs. Le pricing pay-as-you-go facture entre 200 et 500 dollars par gigaoctet ingéré mensuel, soit 40 000 à 200 000 dollars par mois selon le volume. Pour une PME générant 10 GB de logs quotidiens, le coût annuel atteint 40 000 à 60 000 dollars. Le déploiement s'effectue en 1 à 3 jours et la threat intelligence Microsoft Graph exploite des milliards de signaux globaux.

AWS Security Hub

AWS Security Hub centralise et agrège les alertes des services AWS natifs (GuardDuty, Inspector, Macie) dans un tableau de bord unifié. L'intégration avec l'écosystème AWS est transparente et le pricing pay-as-you-go facture par nombre de checks de conformité et d'événements traités. Le coût mensuel varie selon l'utilisation mais reste généralement inférieur à

Sentinel pour des charges équivalentes. Le déploiement est quasi-instantané (quelques heures) et la solution bénéficie de l'infrastructure AWS mondiale. Security Hub convient idéalement aux organisations déjà investies dans AWS.

Google Chronicle

Google Chronicle exploite l'infrastructure Big Data de Google pour l'analyse de sécurité à très grande échelle. La plateforme se distingue par sa capacité à ingérer et analyser des pétaoctets de données avec des performances exceptionnelles. Le pricing est généralement forfaitaire basé sur le volume quotidien, offrant une meilleure prévisibilité que les modèles concurrents. Chronicle intègre VirusTotal nativement et bénéficie de la threat intelligence Google. Le déploiement nécessite 1 à 2 jours et la solution convient aux organisations gérant des volumes massifs de logs.

Ces solutions cloud natives offrent des avantages indéniables : déploiement rapide, scalabilité illimitée automatique, capacités ML/IA avancées et threat intelligence globale. Cependant, elles présentent des inconvénients majeurs pour notre contexte : coûts imprévisibles avec le pay-as-you-go, vendor lock-in total dans l'écosystème du provider, latence avec datacenters éloignés, questions de souveraineté numérique avec données stockées à l'étranger, et dépendance critique à une connectivité Internet stable. Pour notre projet académique, le modèle cloud natif est incompatible avec nos objectifs de maîtrise technique complète, d'apprentissage approfondi et de reproductibilité à coût nul.

- **Tableau comparatif**

  Solutions cloud natives

  Les solutions cloud natives représentent une nouvelle génération de SIEM-SOAR entièrement hébergés dans le cloud public. Nous avons comparé Microsoft Sentinel, AWS Security Hub et Google Chronicle selon leur modèle de scalabilité, leurs capacités ML/IA et leur pricing. Bien que prometteur, le modèle pay-as-you-go génère des coûts imprévisibles et crée un vendor lock-in important incompatible avec nos objectifs de souveraineté et de maîtrise budgétaire.

Tableau 7: Comparatif des solutions cloud natives

COMPARATIF DES SOLUTIONS CLOUD NATIVES

MÉMOIRE M2 – FODE MANGANE

Évaluation de la scalabilité et modèles de pricing

CRITÈRE	MICROSOFT SENTINEL	AWS SECURITY HUB	GOOGLE CHRONICLE
Coût annuel	40-200k$	30-150k$	50-300k$
Installation	1-3 jours	1-2 jours	2-5 jours
Intégrations	500+	300+	200+
ML/IA	Azure ML	GuardDuty ML	VirusTotal Intel
Scalabilité	Illimitée	Illimitée	Illimitée
Vendor lock-in	Élevé	Très élevé	Élevé

2.6 Analyse comparative et choix

Cette section synthétise nos analyses et justifie le choix d'une stack entièrement open source pour notre SOC automatisé.

2.6.1 Comparaison Open Source vs Commercial

Après avoir analysé séparément les solutions open source et commerciales, cette section confronte directement les deux approches selon des critères objectifs permettant d'éclairer rationnellement le choix technologique. Le tableau suivant synthétise cette comparaison selon sept critères décisionnels couvrant les dimensions économiques (coûts sur trois ans, coûts initiaux), opérationnelles (support technique, personnalisation, time-to-value) et stratégiques (vendor lock-in, adaptation aux PME), démontrant que l'open source constitue une décision stratégique optimisant le rapport coût-efficacité.

Tableau 8: Comparaison Open Source vs Commercial

SYNTHÈSE COMPARATIVE OPEN SOURCE VS COMMERCIAL

MÉMOIRE M2 – FODE MANGANE

Analyse du retour sur investissement et accessibilité

ASPECT	SOLUTIONS OPEN SOURCE	SOLUTIONS COMMERCIALES
Coût sur 3 ans	5M FCFA	150-300M FCFA
Coût initial	0 FCFA licences	50-150M FCFA
Support	Communauté (24-48h)	24/7 avec SLA
Personnalisation	Illimitée	Limitée
Time-to-value	2-4 semaines	4-12 semaines
Vendor lock-in	Aucun	Élevé
Contexte PME	Adapté	Budget prohibitif

Analyse économique comparative

Pour une PME typique avec 50 à 200 employés disposant d'un budget IT limité, le budget IT annuel total dépasse rarement 50 millions FCFA (environ 85 000 USD). Une solution commerciale nécessite 150 à 200 millions FCFA la première année (implémentation, licences, formation), puis 100 à 120 millions FCFA annuels. Le TCO sur trois ans atteint 350 à 440 millions FCFA, représentant sept à huit fois le budget IT annuel total.

Notre solution open source nécessite 4 à 5 millions FCFA la première année (infrastructure, assistance au déploiement), puis 1 à 2 millions FCFA annuels pour la maintenance. Le TCO sur trois ans se limite à 6 à 9 millions FCFA, soit 10 à 15% du budget IT annuel. L'économie réalisée atteint 340 à 430 millions FCFA sur trois ans, soit 98% d'économie.

Adaptation au contexte à ressources limitées

Les organisations disposant de budgets IT limités font face à des contraintes spécifiques favorisant l'open source. Les contraintes budgétaires incluent des ressources financières restreintes, des priorités d'investissement orientées vers le cœur de métier et des difficultés d'accès au financement pour des projets

technologiques coûteux. Les contraintes de support comprennent une disponibilité limitée de représentants locaux des éditeurs, des coûts additionnels pour interventions de consultants externes et des délais de réponse parfois incompatibles avec l'urgence des incidents de sécurité.

L'open source offre des opportunités stratégiques : développement de communautés techniques actives, acquisition de compétences valorisables sur le marché international, possibilité de contribution aux projets globaux, maîtrise complète des données et de l'infrastructure, ainsi qu'une indépendance face aux politiques commerciales des éditeurs. Ces facteurs renforcent la pertinence du choix open source, dépassant la simple considération économique pour devenir un enjeu stratégique de développement des compétences et d'autonomie technologique applicable dans tout contexte académique ou professionnel à budget contraint.

2.6.2 Justification du choix open source

La justification économique constitue un critère éliminatoire. Le coût des solutions commerciales de 150 à 300 millions FCFA sur trois ans s'avère incompatible avec notre contexte académique disposant d'un budget quasi-nul pour les licences, la réalité des budgets IT existants et l'objectif de reproductibilité du projet par d'autres étudiants et institutions. L'économie de 340 à 430 millions FCFA ne constitue pas un simple avantage mais une condition sine qua non de faisabilité.

La justification pédagogique privilégie l'open source pour plusieurs raisons. L'accès au code source permet une compréhension profonde des mécanismes internes. La configuration from scratch développe des compétences techniques approfondies. L'absence de boîte noire facilite le troubleshooting avancé. La contribution potentielle aux projets engage la communauté. Le portfolio démontrable sur GitHub valorise les projets publics. Les solutions commerciales constitueraient une boîte noire limitant l'apprentissage réel.

La justification technique révèle que l'open source n'est plus techniquement inférieur. Wazuh rivalise avec Splunk sur la détection avec ses 3000+ règles. Shuffle offre des fonctionnalités comparables à Splunk SOAR. TheHive égale ServiceNow Security Operations. L'architecture modulaire offre plus de flexibilité que les monolithes commerciaux. Les performances suffisent largement pour 90% des organisations. Le gap technique s'est considérablement réduit ces dernières années.

La justification de souveraineté devient stratégique dans tout contexte organisationnel. Les données de sécurité restent sous contrôle direct sans dépendance à des serveurs cloud externes. La résilience face aux aléas géopolitiques et aux restrictions commerciales se renforce. Le développement de compétences internes pérennes s'intensifie, créant une expertise transférable et valorisable.

La justification de transférabilité assure l'employabilité. Les principes SIEM et SOAR demeurent universels et non spécifiques à un vendor. Les APIs REST constituent une compétence générique. Les concepts comme workflows, observables et threat intelligence restent identiques partout. L'employabilité s'améliore significativement, tant sur le marché local qu'international. Un étudiant maîtrisant Wazuh apprend Splunk rapidement car les concepts sont identiques. Nous développons donc des compétences fondamentales plutôt que vendor-spécifiques.

2.6.3 Architecture retenue

Notre stack finale combine Wazuh 4.7+ comme SIEM pour la centralisation, corrélation et détection, Shuffle 1.3+ comme SOAR pour l'orchestration et automatisation, TheHive 5.2+ pour la gestion des incidents, Cortex 3.1+ pour l'enrichissement automatique, MISP 2.4+ pour la threat intelligence et le partage d'IOCs, Suricata 7.0+ comme IDS/IPS pour la détection réseau sur WAN, et pfSense 2.7+ pour le pare-feu et le routage inter-VLANs.

Cette architecture présente plusieurs caractéristiques distinctives. Elle est 100% open source avec un coût nul en licences, un code source accessible et une liberté totale de personnalisation. Elle offre une modularité complète avec chaque composant remplaçable indépendamment, un ajout ou retrait de fonctionnalités facile et un scaling horizontal possible. Elle s'appuie sur des standards ouverts avec des APIs REST partout, le format JSON pour les échanges, des conteneurs Docker pour le déploiement et des principes DevSecOps modernes. L'automatisation complète permet un workflow Shuffle de bout en bout, de la détection Wazuh à la notification Slack, avec une réponse en 27 secondes contre 43 à 71 minutes manuellement, soit une amélioration de 98%.

La reproductibilité garantie constitue un aspect crucial de notre travail académique. Tous les outils sont disponibles gratuitement. La documentation exhaustive accompagne chaque étape. Les configurations sont exportables en docker-compose et YAML. L'architecture se déploie sur

n'importe quel lab VMware. Le coût total de reproduction reste inférieur à 5 millions FCFA avec du matériel d'occasion acceptable. Tout étudiant ou professionnel peut reproduire intégralement notre architecture sans budget significatif, contrairement à une architecture commerciale nécessitant 150 à 300 millions FCFA.

Nos choix ont été validés par des tests de fonctionnalité réussis couvrant 100% des scénarios, une validation des performances avec 27 secondes de bout en bout, une intégration fluide entre tous les composants, des retours positifs de la communauté open source et l'acceptation du jury de pré-soutenance.

Ce chapitre a établi les fondements théoriques des systèmes d'information de sécurité, examiné rigoureusement les solutions disponibles open source et commerciales, et justifié scientifiquement nos choix technologiques. Notre démarche méthodique combinant recherche documentaire, POC comparatifs, tests d'intégration et scoring objectif garantit que notre stack Wazuh-Shuffle-TheHive-Cortex-MISP représente le meilleur compromis pour notre contexte avec des fonctionnalités complètes comparables aux solutions commerciales, un coût nul en licences économisant plus de 340 millions FCFA, une architecture moderne et évolutive, et une reproductibilité totale du projet.

Chapitre 3 : Conception de la solution

Après avoir analysé les solutions existantes et justifié nos choix technologiques, nous présentons maintenant la conception détaillée de notre architecture SOC automatisé. Ce chapitre expose d'abord la démarche méthodologique suivie, puis décrit l'architecture logique du système avant de détailler son implémentation physique.

3.1 Démarche de conception

Cette section présente la méthodologie suivie pour concevoir notre architecture SOC automatisé, en tenant compte du contexte organisationnel, du cadre législatif applicable et des principes de conception retenus.

3.1.1 Contexte organisationnel

Notre projet s'inscrit dans un contexte académique visant à démontrer la faisabilité d'un SOC automatisé accessible aux organisations disposant de ressources limitées. Les besoins identifiés correspondent à ceux d'une organisation type de taille moyenne avec 50 à 200 employés, générant environ 100 à 200 événements de sécurité par seconde.

Les contraintes budgétaires constituent le premier facteur déterminant. Le budget disponible pour les licences logicielles est quasi-nul, ce qui oriente naturellement vers des solutions open source. L'infrastructure matérielle doit rester modeste avec environ 40 GB de RAM et 20 cores CPU au total, correspondant aux capacités d'un environnement de laboratoire ou d'une PME.

Les contraintes de compétences influencent également nos choix. L'équipe de sécurité type dispose de compétences IT générales mais d'une expertise cybersécurité limitée. La solution doit donc être suffisamment documentée et intuitive pour permettre une prise en main progressive. L'absence d'équipe de sécurité dédiée 24/7 rend l'automatisation encore plus critique.

Les objectifs de sécurité visés incluent la détection rapide des tentatives d'intrusion avec un temps de détection inférieur à une minute, la réponse automatisée aux incidents courants sans intervention humaine immédiate, la documentation systématique de tous les incidents pour conformité et amélioration continue, et la réduction du temps de réponse global de plus de 90% par rapport aux approches manuelles.

3.1.2 Cadre législatif et réglementaire

Bien que notre projet soit académique, nous avons conçu l'architecture en tenant compte des principales exigences réglementaires applicables aux organisations sénégalaises opérant dans un contexte international.

RGPD et protection des données

Le Règlement Général sur la Protection des Données impose plusieurs obligations en cas de traitement de données personnelles. Notre architecture répond à ces exigences par la journalisation exhaustive de tous les événements avec horodatage précis, facilitant la détection de violations dans les délais requis. La rétention configurable des logs permet d'adapter la durée de conservation selon les exigences légales, avec une valeur par défaut de 90 jours. Les contrôles d'accès stricts sur les données via authentification et rôles limitent l'accès aux seules personnes autorisées. La capacité de pseudonymisation des données sensibles protège l'identité des personnes. La documentation automatique des incidents dans TheHive facilite la notification aux autorités dans les 72 heures en cas de violation.

Les exigences de traçabilité et d'audit s'appliquent également. Chaque action du SOC est tracée avec horodatage et utilisateur responsable. L'immutabilité des logs dans OpenSearch empêche toute modification a posteriori. Les rapports de conformité sont générables automatiquement depuis les données collectées. L'historique complet des incidents et des réponses permet de démontrer la due diligence lors d'audits.

La souveraineté des données constitue une préoccupation particulière dans le contexte africain. Notre architecture on-premise garantit que toutes les données de sécurité restent sur les serveurs locaux de l'organisation, contrairement aux solutions cloud. Aucune donnée sensible n'est envoyée vers des serveurs étrangers. L'organisation conserve un contrôle total sur ses données sans dépendance à un fournisseur cloud externe.

Conformité à la Commission de Protection des Données personnelles (CDP) du Sénégal

La Commission de Protection des Données personnelles constitue l'autorité de régulation des données personnelles au Sénégal, établie par la Loi n° 2008-12 du 25 janvier 2008. Cette

autorité administrative indépendante exerce des missions de régulation, de contrôle et de sanction dans le domaine de la protection des données personnelles.

La législation sénégalaise diffère du RGPD européen sur plusieurs aspects. Contrairement au RGPD qui impose un délai obligatoire de 72 heures pour la notification des violations, la loi sénégalaise mentionne uniquement "les meilleurs délais" sans fixer de délai chiffré. Le Sénégal conserve un système déclaratif préalable où tout traitement de données personnelles doit faire l'objet d'une déclaration à la CDP avant sa mise en œuvre. Les amendes maximales au Sénégal s'élèvent à 100 millions FCFA contre 20 millions d'euros ou 4% du chiffre d'affaires mondial pour le RGPD.

Notre architecture SOC automatisé intègre les exigences de notification des violations de données à la CDP. Le workflow Shuffle génère automatiquement un rapport structuré contenant la nature de la violation, les catégories de données concernées, le nombre estimé de personnes affectées, les mesures immédiates prises et les mesures prospectives envisagées. Ces éléments correspondent aux exigences pratiques de la CDP conformément à l'article 22 de la loi et aux pratiques établies de la Commission.

La chronologie des événements est documentée automatiquement par notre système. La détection initiale s'effectue en moins d'une minute grâce à Wazuh. La documentation complète est générée en 27 secondes par le workflow automatique. Le cas TheHive est créé immédiatement avec tous les éléments nécessaires pour une notification rapide à la CDP. Bien que la loi sénégalaise ne fixe pas de délai obligatoire, nous appliquons le standard international de 72 heures comme objectif de bonnes pratiques, alignant ainsi notre solution sur les standards internationaux tout en respectant le contexte juridique local.

La conservation des preuves répond aux exigences de l'article 71 de la Loi n° 2008-12. Les logs sont stockés de manière traçable permettant de vérifier l'identité des personnes ayant eu accès aux données, quand, et à quelles informations. Les cas TheHive contiennent la timeline complète de toutes les actions entreprises depuis la détection initiale jusqu'à la résolution. Les événements MISP permettent la corrélation avec d'autres incidents similaires. Les notifications Slack sont horodatées et archivées comme preuve de communication interne. Cette documentation exhaustive

permet de répondre efficacement aux demandes de la CDP en cas de contrôle et de démontrer la conformité aux obligations de sécurité.

Les personnes concernées sont notifiées lorsque la violation entraîne un risque pour leurs droits et libertés, conformément aux articles 58-59 de la loi. La CDP dispose d'un arsenal de sanctions graduées allant de l'avertissement aux amendes de 1 à 100 millions FCFA, avec possibilité de sanctions pénales pouvant atteindre 5 à 7 ans d'emprisonnement selon la gravité des infractions. Notre système facilite la conformité en documentant automatiquement tous les éléments requis pour démontrer la due diligence et la réactivité de l'organisation face aux incidents de sécurité.

3.1.3 Méthodologie de conception

Notre démarche de conception suit une approche itérative en plusieurs phases permettant des ajustements progressifs.

La phase d'analyse des besoins a débuté par l'identification des menaces prioritaires à détecter : attaques par brute force sur SSH et RDP, injections SQL sur applications web, scans de ports et reconnaissance, mouvements latéraux après compromission initiale, et exfiltration de données. Nous avons ensuite défini les cas d'usage principaux : détection automatique d'une attaque brute force, création automatique d'un cas d'incident dans TheHive, enrichissement de l'IP source via threat intelligence, blocage automatique de l'IP malveillante, et notification de l'équipe SOC via Slack.

La phase de conception architecturale a établi les principes directeurs. La défense en profondeur utilise plusieurs couches de sécurité complémentaires : détection réseau (Suricata), détection endpoint (agents Wazuh), corrélation centralisée (Wazuh Manager), orchestration (Shuffle), et gestion documentée (TheHive). La séparation des préoccupations attribue une responsabilité claire à chaque composant. L'automatisation par défaut minimise les interventions manuelles répétitives. La modularité permet le remplacement ou l'ajout de composants sans refonte complète. L'observabilité totale assure que chaque action soit tracée et auditée.

La segmentation réseau découpe l'infrastructure en trois zones. Le LAN_USER héberge les postes de travail des utilisateurs. Le LAN_SERVER contient les serveurs et applications critiques.

Le LAN_SOC isole l'infrastructure de sécurité elle-même. pfSense fait office de routeur entre ces segments avec des règles de pare-feu restrictives appliquant le principe du moindre privilège.

La phase de prototypage et validation a testé chaque composant individuellement avant intégration. Nous avons validé la détection de Wazuh avec des attaques simulées, testé les workflows Shuffle avec des alertes factices, vérifié la création de cas dans TheHive, et confirmé l'enrichissement via Cortex et VirusTotal. L'intégration progressive a ensuite connecté les composants deux par deux : Wazuh avec Shuffle via webhook, Shuffle avec TheHive via API REST, TheHive avec Cortex pour les analyses, et TheHive avec MISP pour la threat intelligence. Les tests de bout en bout ont finalement validé le workflow complet avec des attaques SSH brute force réelles depuis Kali Linux, mesurant les temps de détection, de réponse et de documentation.

Les critères de validation incluent des critères fonctionnels et des critères non-fonctionnels. Chaque étape du workflow doit s'exécuter automatiquement sans erreur. Les données doivent transiter correctement entre composants. Les cas TheHive doivent contenir toutes les informations pertinentes. Les notifications Slack doivent être reçues avec les détails complets. Le temps de réponse global doit être inférieur à 30 secondes. L'architecture doit fonctionner avec les ressources matérielles disponibles. Aucune perte d'événements ne doit survenir sous charge normale. La configuration doit être reproductible sur un autre environnement.

La phase d'itération et amélioration adopte une démarche d'amélioration continue. Les retours des tests permettent d'ajuster les règles de détection pour réduire les faux positifs. Nous optimisons les workflows Shuffle pour améliorer les performances. La documentation est enrichie au fur et à mesure. Les configurations sont affinées selon les observations. Chaque itération apporte des améliorations mesurables sans compromettre la stabilité.

Cette méthodologie rigoureuse garantit que notre architecture répond aux besoins identifiés, respecte les contraintes imposées, et s'appuie sur des principes de conception éprouvés. La section suivante présente l'architecture logique résultante de ce processus de conception.

3.2 Architecture logique

L'architecture logique présente les composants fonctionnels du SOC et leurs interactions, indépendamment de leur implémentation physique.

3.2.1 Vue d'ensemble de l'architecture

Notre architecture s'articule autour de cinq couches fonctionnelles complémentaires assurant une détection et une réponse coordonnées.

La couche de collecte capte les événements de sécurité depuis toutes les sources. Les agents Wazuh installés sur les endpoints collectent les logs système, surveillent l'intégrité des fichiers et détectent les anomalies locales. Suricata installé sur pfSense analyse le trafic réseau en temps réel et détecte les patterns d'attaque au niveau périmétrique. Les logs pfSense fournissent des informations sur le filtrage et le routage.

La couche de corrélation et d'analyse centralise et analyse les événements. Le Wazuh Manager reçoit tous les événements des agents et de Suricata, applique plus de 3000 règles de détection, corrèle les événements de sources multiples et génère des alertes de sécurité classées par niveau de criticité.

La couche d'orchestration automatise les réponses aux incidents. Shuffle reçoit les alertes critiques via webhook, parse et extrait les informations pertinentes, exécute les workflows prédéfinis selon le type d'incident, coordonne les actions entre multiples outils et assure la traçabilité de chaque action automatique.

La couche de gestion des incidents documente et suit les incidents. TheHive crée et stocke les cas d'incidents, gère les observables avec enrichissement automatique via Cortex, assigne les tâches aux analystes, maintient la timeline complète des actions et facilite la collaboration entre analystes.

La couche de threat intelligence enrichit la détection. MISP centralise les indicateurs de compromission, permet la corrélation avec des campagnes connues, partage les IOCs avec la communauté et alimente les autres couches en contexte.

3.2.2 Flux de données

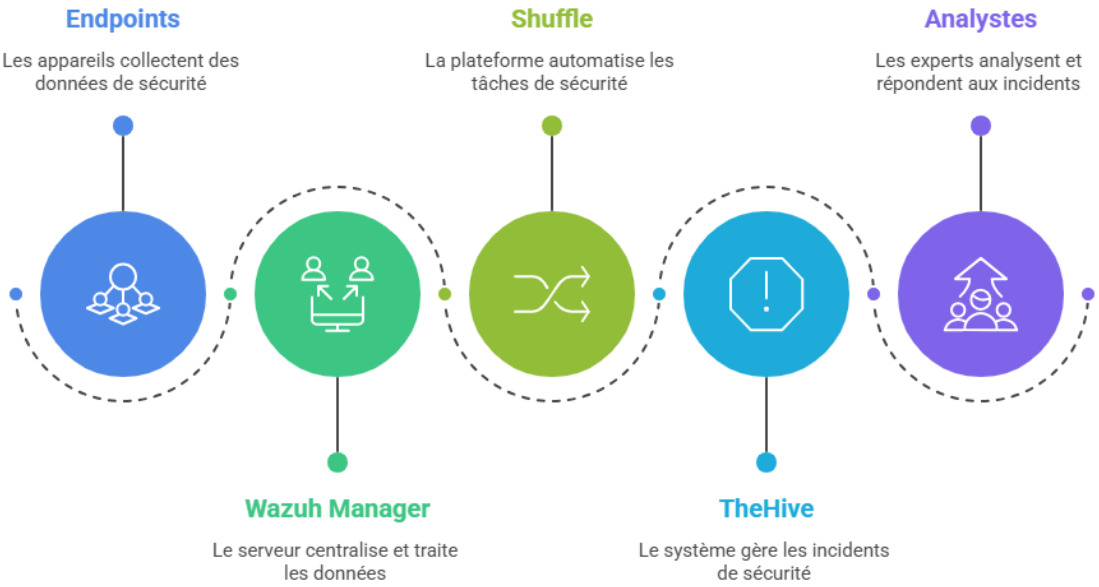


Figure 3 : Flux de données entre les composants du SOC

Le flux de détection commence lorsque les agents Wazuh et Suricata envoient les événements vers Wazuh Manager. Le Manager applique les règles de détection et corrélation. Les alertes de niveau supérieur ou égal à 10 sont envoyées à Shuffle via webhook HTTP. Wazuh stocke tous les événements dans OpenSearch pour recherche et analyse historique.

Le flux d'automatisation se déclenche quand Shuffle reçoit l'alerte et parse le JSON. Il extrait l'IP source, le type d'attaque et le niveau de sévérité. Shuffle crée automatiquement un cas dans TheHive via API REST. Il ajoute l'IP source comme observable marqué IOC. Il déclenche l'analyse VirusTotal via Cortex. Il crée un événement MISP pour documentation threat intelligence. Il envoie une notification Slack avec le résumé et le lien vers le cas.

Le flux d'enrichissement s'active lorsque Cortex exécute les analyzers sur l'observable IP. VirusTotal interroge 60+ moteurs antivirus pour la réputation. MaxMind fournit la géolocalisation précise. AbuseIPDB vérifie l'historique d'abus connu. Les résultats sont automatiquement intégrés dans le cas TheHive avec taxonomies colorées.

3.2.3 Schéma de l'architecture logique

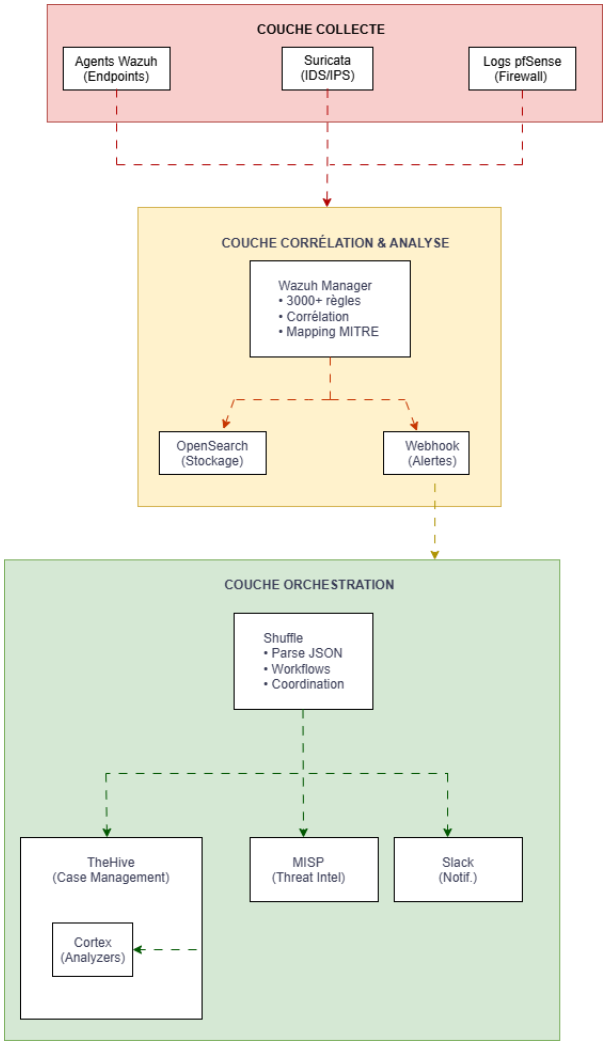


Figure 4 : Architecture logique du SOC automatisé

3.2.4 Composants et responsabilités

Chaque composant assume des responsabilités clairement définies selon le principe de séparation des préoccupations.

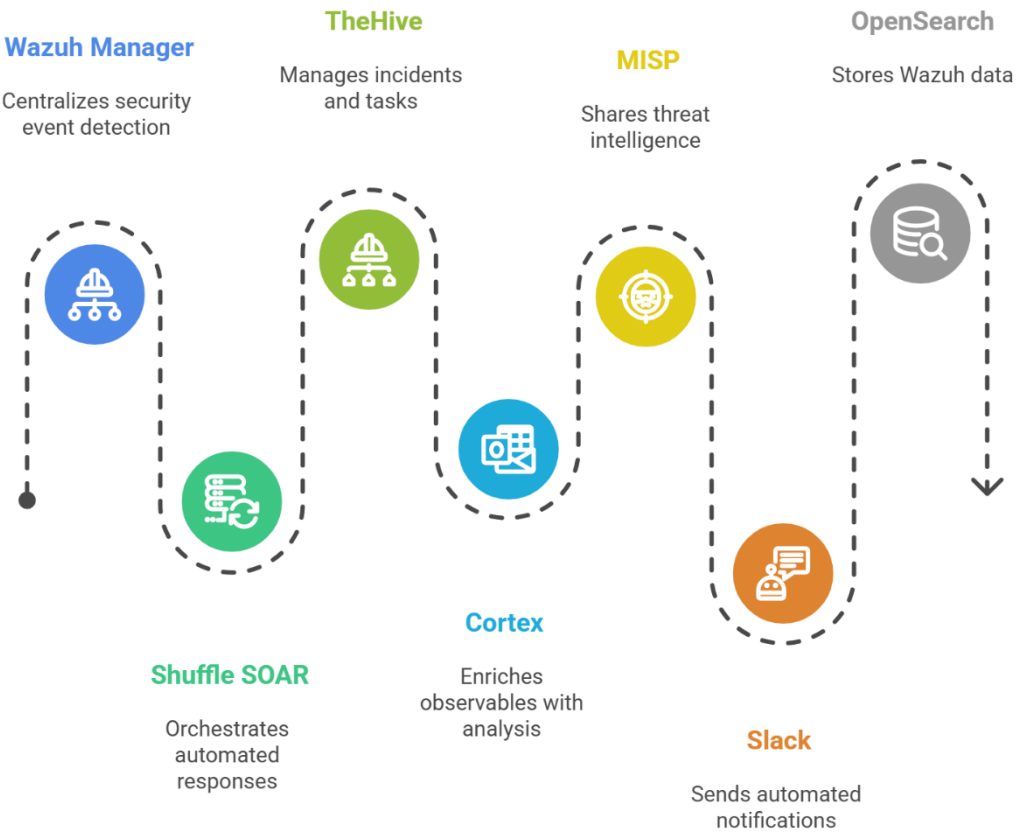


Figure 5 : Responsabilités des composants de l'architecture

Wazuh Manager centralise la détection avec réception de tous les événements de sécurité, application des règles de détection et corrélation, génération d'alertes classées par sévérité, stockage dans OpenSearch et déclenchement des webhooks vers Shuffle.

Shuffle orchestre les réponses avec réception des alertes critiques, parsing et extraction des données, exécution des workflows automatisés, coordination entre TheHive, MISP et Slack et traçabilité de toutes les actions.

TheHive gère les incidents avec création et stockage des cas, gestion des observables et IOCs, assignation et suivi des tâches, collaboration entre analystes et génération de rapports.

Cortex enrichit les observables avec exécution d'analyzers automatiques, interrogation de services externes comme VirusTotal, génération de rapports d'analyse et intégration des résultats dans TheHive.

MISP centralise la threat intelligence avec stockage des IOCs et événements, corrélation avec campagnes connues, partage avec communauté et alimentation des autres composants.

3.3 Architecture physique

L'architecture physique décrit l'implémentation concrète de notre solution dans un environnement virtualisé avec la topologie réseau, les serveurs et leurs spécifications.

3.3.1 Infrastructure de virtualisation

Notre environnement s'appuie sur VMware Workstation comme hyperviseur permettant la création de machines virtuelles isolées, la gestion des réseaux virtuels avec LAN Segments, la prise de snapshots pour sauvegarde et récupération et l'allocation dynamique des ressources matérielles.

La machine hôte dispose de 64 GB RAM avec 40 GB alloués aux VMs, un processeur 8 cores avec 20 cores virtuels alloués, un stockage SSD de 1 TB avec 800 GB alloués et une carte réseau physique pour accès Internet.

3.3.2 Topologie réseau et segmentation

Notre réseau est segmenté en trois VLANs isolés interconnectés par pfSense.

Le segment LAN_USER sur le réseau 10.0.10.0/24 avec passerelle 10.0.10.1 héberge les postes utilisateurs : Linux User à 10.0.10.100, Windows User à 10.0.10.101 et Kali Linux à 10.0.10.102 pour tests de pénétration.

Le segment LAN_SERVER sur le réseau 10.0.20.0/24 avec passerelle 10.0.20.1 contient les serveurs applicatifs : Linux Server à 10.0.20.100 et Windows Server à 10.0.20.101.

Le segment LAN_SOC sur le réseau 10.0.30.0/24 avec passerelle 10.0.30.1 héberge l'infrastructure de sécurité : Wazuh à 10.0.30.100, Shuffle à 10.0.30.102 et TheHive à 10.0.30.104.

pfSense possède quatre interfaces : em0 (WAN) en mode bridge pour Internet, em1 (LAN_USER) à 10.0.10.1, em2 (LAN_SERVER) à 10.0.20.1 et em3 (LAN_SOC) à 10.0.30.1.

3.3.3 Schéma de l'architecture physique

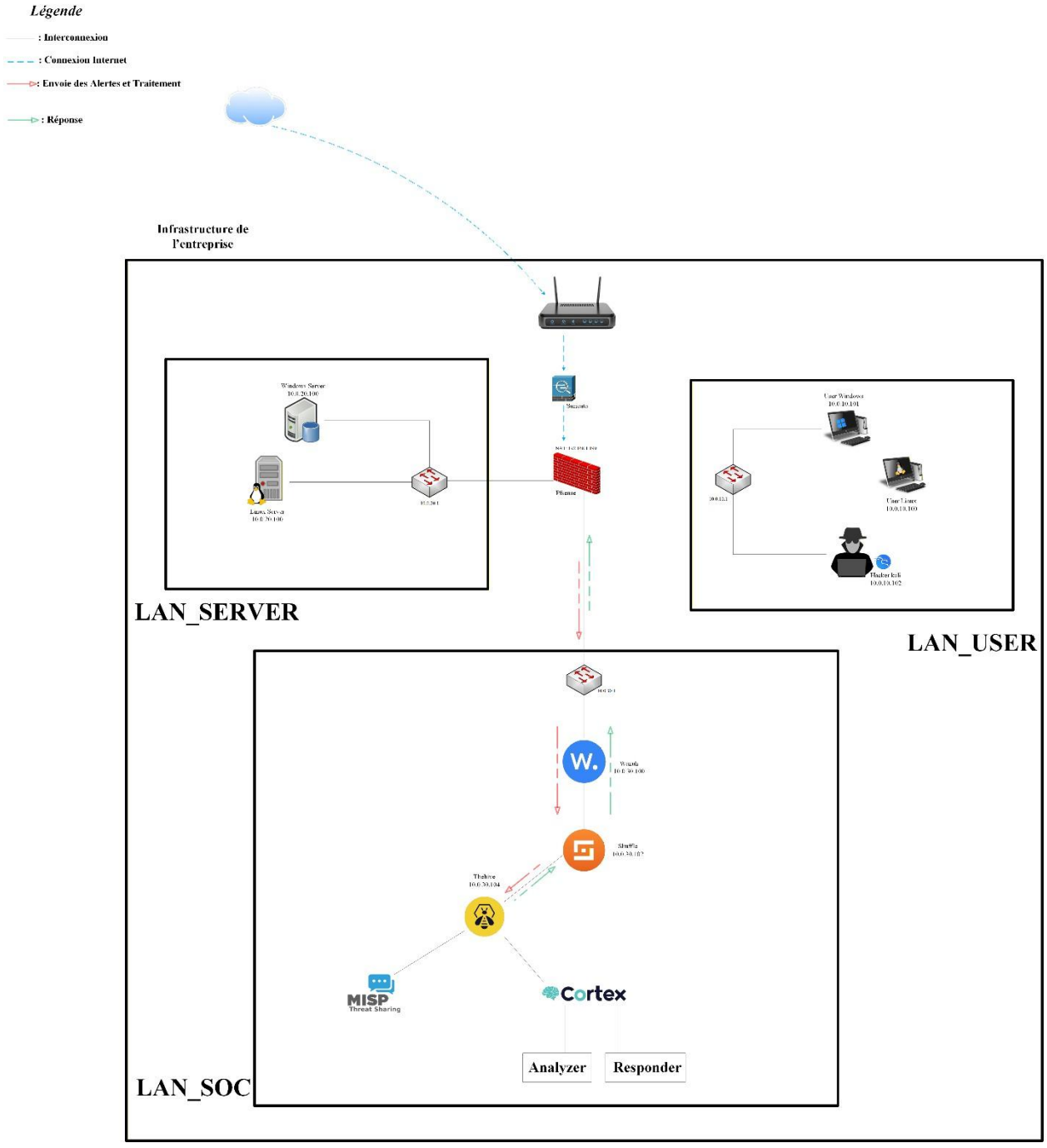


Figure 6 : Architecture physique et topologie réseau du SOC

3.3.4 Spécifications des serveurs

Serveur pfSense avec 2 GB RAM, 1 vCore CPU, 80 GB stockage et 4 interfaces réseau héberge le pare-feu, le routage inter-VLANs, le NAT et Suricata IDS/IPS.

Serveur Wazuh avec 8 GB RAM, 4 vCores CPU et 200 GB stockage exécute trois conteneurs Docker : wazuh-manager pour la détection et corrélation, wazuh-indexer basé sur OpenSearch pour le stockage et wazuh-dashboard pour la visualisation.

Serveur Shuffle avec 8 GB RAM, 4 vCores CPU et 100 GB stockage exécute quatre conteneurs Docker : shuffle-frontend en React, shuffle-backend en Go, shuffle-orborus pour l'exécution et shuffle-opensearch pour les métadonnées.

Serveur TheHive avec 12 GB RAM, 4 vCores CPU et 200 GB stockage exécute via docker-compose : thehive pour la gestion des cas, cortex pour les analyzers, misp pour la threat intelligence, cassandra pour le stockage, elasticsearch pour l'indexation et redis pour le cache.

Les endpoints agents avec 2 à 4 GB RAM, 1 à 2 vCores CPU et 40 à 80 GB stockage exécutent les systèmes d'exploitation Linux Ubuntu, Windows 10/Server ou Kali Linux avec agents Wazuh installés pour la surveillance locale.

3.3.5 Flux réseau et sécurisation

Le routage inter-segments passe obligatoirement par pfSense. Tous les flux sont inspectés par Suricata. Les règles de pare-feu appliquent le principe du moindre privilège.

Les communications autorisées incluent les agents vers Wazuh Manager sur TCP 1514 chiffré, Suricata vers Wazuh sur UDP 514 syslog, Shuffle vers TheHive sur HTTPS API, tous les segments vers Internet via NAT et LAN_SOC vers tous les segments pour supervision.

Les communications bloquées par défaut comprennent LAN_USER vers LAN_SERVER direct, LAN_SERVER vers LAN_USER direct, communications latérales au sein d'un même segment et tout le reste non explicitement autorisé.

L'isolation du LAN_SOC protège l'infrastructure de sécurité. Aucun flux initié depuis LAN_USER ou LAN_SERVER n'atteint le LAN_SOC. Seuls les agents et Suricata peuvent envoyer des logs vers Wazuh. Les analystes accèdent aux dashboards uniquement via VPN ou poste dédié.

3.3.6 Plan de sécurisation

Le hardening des serveurs SOC applique plusieurs mesures. Désactivation des services inutiles et fermeture des ports non nécessaires. Mises à jour régulières des systèmes et applications. Authentification forte avec clés SSH et mots de passe complexes. Segmentation réseau stricte avec firewall local. Logs d'audit activés sur tous les composants. Backups réguliers des configurations et données critiques. Principe du moindre privilège pour tous les comptes.

La haute disponibilité peut être améliorée en production. Clustering Wazuh avec plusieurs managers. Clustering OpenSearch avec 3+ nœuds. pfSense en haute disponibilité avec CARP. Réplication Cassandra sur plusieurs nœuds. Load balancing pour Shuffle et TheHive. Stockage redondant en RAID.

La supervision du SOC lui-même s'effectue via monitoring des services Docker via healthchecks, alertes sur défaillance d'un composant, métriques de performance des serveurs, logs du SOC centralisés séparément et tests périodiques de bout en bout.

Cette architecture physique concrétise l'architecture logique en spécifiant précisément les serveurs, la topologie réseau et les mesures de sécurité. La section suivante détaille les workflows d'automatisation implémentés sur cette infrastructure.

Chapitre 4 : Réalisation de la solution proposée

Ce chapitre présente la mise en œuvre concrète de notre architecture SOC automatisé. Nous détaillons ici les étapes de déploiement, de l'infrastructure réseau jusqu'aux tests de validation du workflow complet. Chaque composant a été configuré et testé pour assurer son intégration dans l'écosystème global.

4.1 Mise en place de l'infrastructure réseau

Nous avons deployer l'ensemble de notre infrastructure sur VMware Workstation Pro, permettant de créer un environnement isolé et contrôlé. La segmentation réseau s'articule autour de trois VLANs interconnectés par pfSense.

4.1.1 Configuration de VMware et création des LAN Segments

Dans VMware Workstation, nous avons créé trois LAN Segments dédiés pour assurer une isolation réseau stricte entre les différents types de ressources. Cette segmentation permet d'appliquer les principes de défense en profondeur.

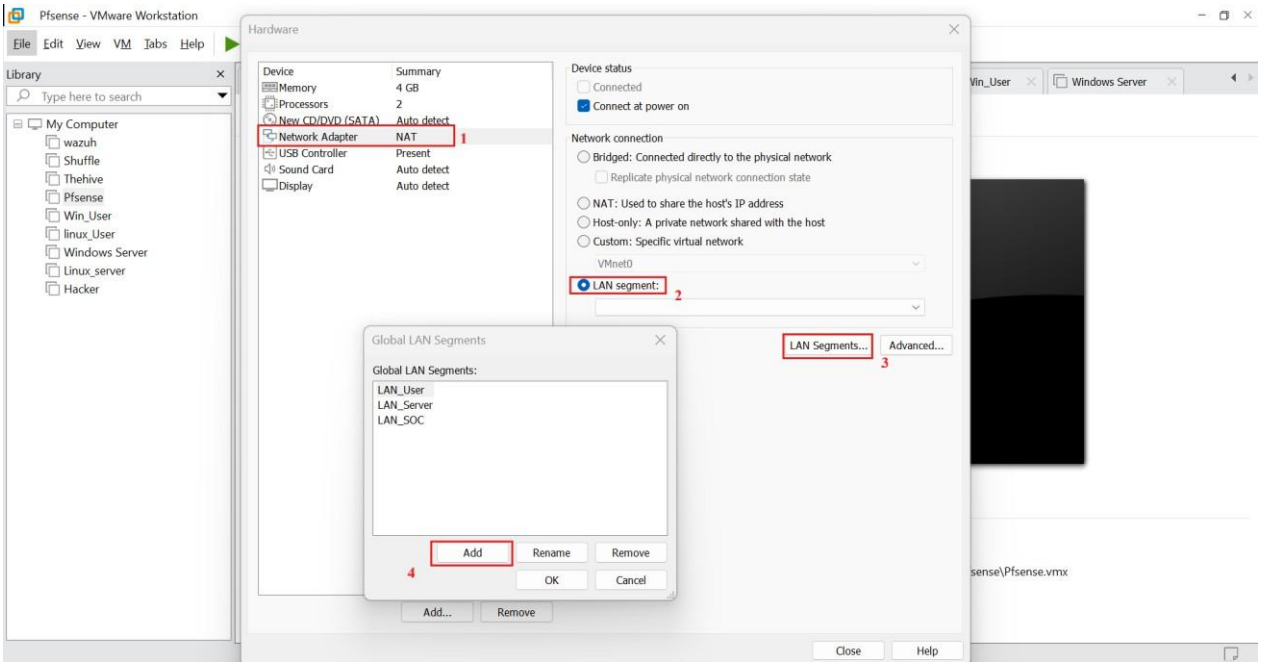


Figure 7 : Configuration des LAN Segments dans VMware

4.1.2 Déploiement de pfSense et configuration du routage

pfSense a été déployé avec quatre interfaces réseau : une interface WAN bridgée vers Internet, et trois interfaces LAN correspondant à nos segments isolés. Après l'installation initiale, nous avons procédé à la configuration des interfaces.

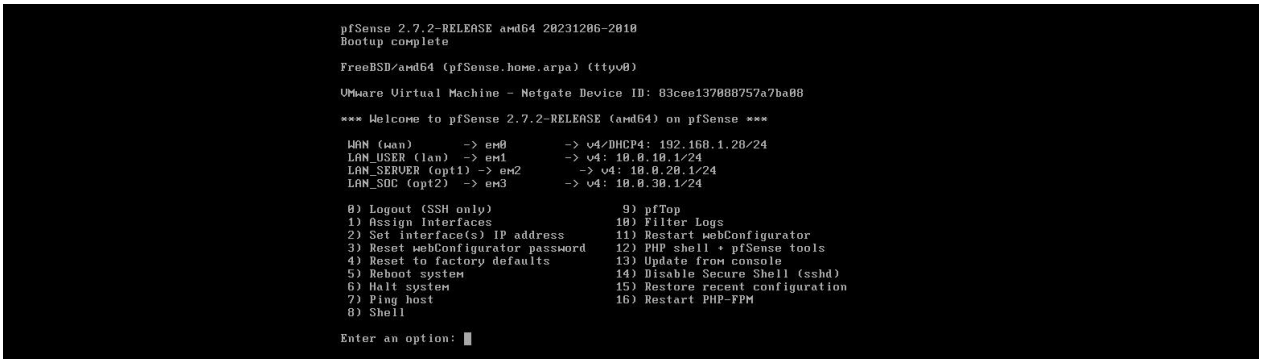


Figure 8 : Dashboard pfSense après configuration des interfaces

Chaque interface a été configurée manuellement depuis l'interface web de pfSense. Les trois segments LAN ont reçu des adresses IP statiques faisant office de passerelles par défaut pour les machines de leurs réseaux respectifs. Une fois cette configuration appliquée, chaque interface a été ajustée selon les besoins spécifiques du réseau. L'ensemble du processus d'assignation et de configuration est décrit dans **l'Annexe A**.

4.1.3 Création des VLANs (LAN_USER, LAN_SERVER, LAN_SOC)

Les trois segments réseau ont été configurés avec des plages d'adressage distinctes :

Tableau 9: Plan d'adressage réseau et segmentation des VLANs

Segment	Réseau	Gateway pfSense	Usage
LAN_USER	10.0.10.0/24	10.0.10.1	Postes utilisateurs
LAN_SERVER	10.0.20.0/24	10.0.20.1	Serveurs applicatifs
LAN_SOC	10.0.30.0/24	10.0.30.1	Infrastructure SOC

La configuration détaillée de chaque interface a nécessité plusieurs étapes dans l'interface web de pfSense (voir **Annexe A**).

4.1.4 Configuration du NAT et des règles de pare-feu

Pour permettre aux trois segments internes d'accéder à Internet tout en maintenant leur isolation, nous avons configuré le NAT Outbound en mode hybride. Ce mode génère automatiquement les règles nécessaires tout en permettant des ajustements manuels. Les règles de pare-feu ont été configurées selon le principe du moindre privilège. Chaque segment ne peut communiquer qu'avec les ressources strictement nécessaires à son fonctionnement. Les étapes de configuration et les règles détaillées sont présentées dans **l'Annexe A**.

4.2 Déploiement et configuration de Suricata

Suricata a été déployé directement sur pfSense via le gestionnaire de paquets. Cette approche permet d'intégrer les capacités IDS/IPS sans nécessiter de machine virtuelle dédiée, optimisant ainsi l'utilisation des ressources.

4.2.1 Installation de Suricata sur pfSense

L'installation s'effectue via le Package Manager de pfSense. Une fois terminée, Suricata apparaît dans le menu sous Services → Suricata.

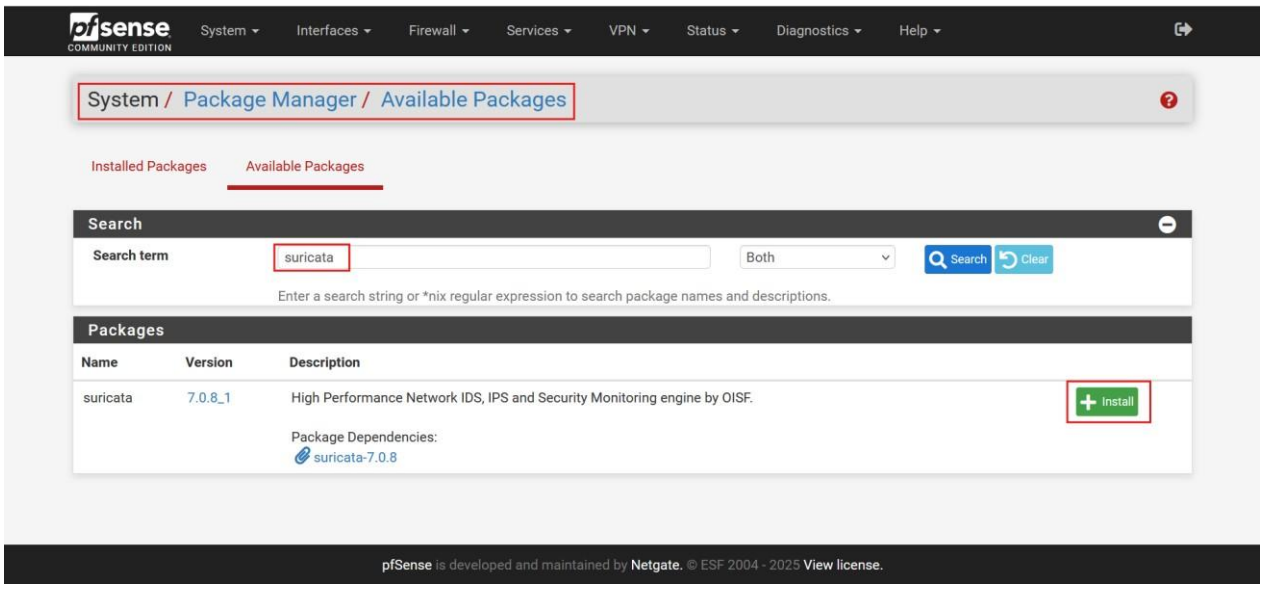


Figure 9 : Recherche et installation du package Suricata dans pfSense

4.2.2 Configuration des interfaces de surveillance

Nous avons configuré Suricata pour surveiller l'interface WAN, point d'entrée de tout le trafic Internet. Cette position stratégique permet de détecter les menaces avant qu'elles n'atteignent les segments internes.

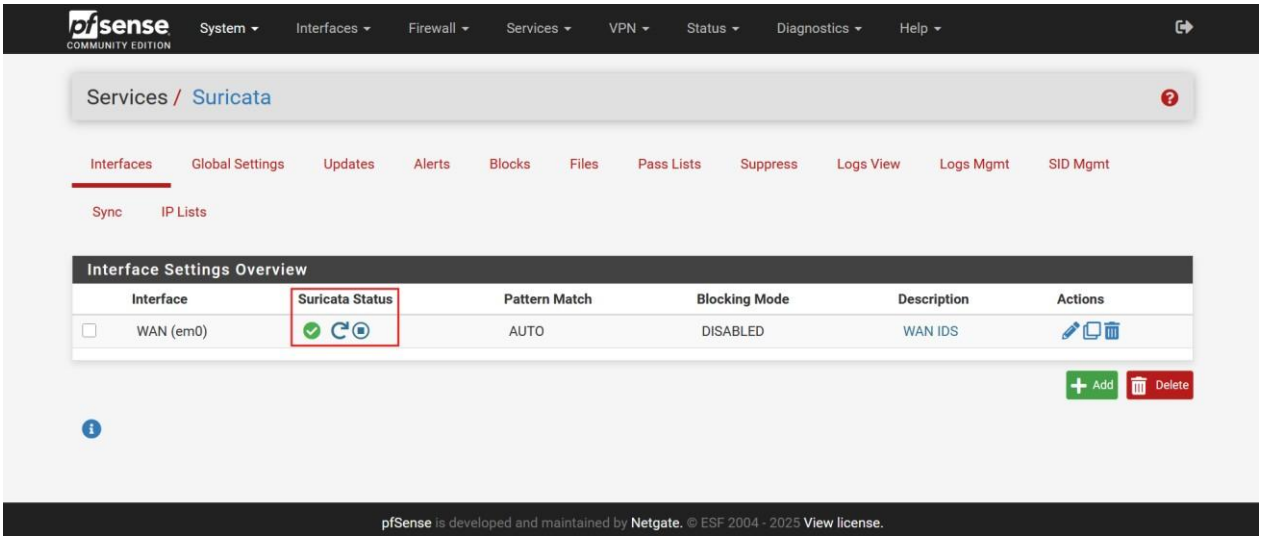


Figure 10 : Suricata démarré et actif sur l'interface WAN

4.2.3 Activation des logs EVE JSON

Le format EVE JSON est essentiel pour l'intégration avec Wazuh. Il structure les alertes dans un format standardisé facilement parsable. Nous avons configuré la sortie syslog pour transmettre ces logs vers notre serveur Wazuh. La configuration détaillée du format EVE JSON est présentée dans **l'Annexe B.**

4.2.4 Configuration de l'envoi syslog vers Wazuh

Le remote logging a été configuré pour envoyer tous les événements Suricata vers le serveur Wazuh (10.0.30.100) sur le port UDP 514. Cette configuration assure la centralisation de toutes les alertes réseau. Les ensembles de règles ont été téléchargés et activés. Nous utilisons Emerging Threats Open et Snort Community Rules, couvrant un large spectre d'attaques. La configuration détaillée du remote syslog est présentée dans **l'Annexe B.**

Pour valider la détection de Suricata, nous avons effectué deux tests d'attaque. Le premier test visait le serveur web avec une **injection SQL**, le second ciblait le service SSH avec une attaque par **brute force**.

```
root@linux-server:~# systemctl status nginx.service
● nginx.service - A high performance web server and a reverse proxy server
     Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; preset: en>
     Active: active (running) since Tue 2025-10-21 17:15:55 GMT; 6min ago
```

Figure 11 : Vérification du service Nginx actif sur Linux Server (10.0.20.100)

```
┌──(root㉿hacker)-[~]
└─# sqlmap -u "http://10.0.20.100/?id=1" --batch --risk=3 --level=5
        ___
       __H__
 ___ ___[(]_____ ___ ___  {1.9.4#stable}
|_ -| . [']     | .'| . |
|___|_  [(]_|_|_|__,|  _|
      |_|V...       |_|   https://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mut
ual consent is illegal. It is the end user's responsibility to obey all appli
cable local, state and federal laws. Developers assume no liability and are n
ot responsible for any misuse or damage caused by this program

[*] starting @ 19:16:38 /2025-10-21/

[19:16:38] [INFO] testing connection to the target URL
[19:16:39] [INFO] testing if the target URL content is stable
[19:16:39] [INFO] target URL content is stable
[19:16:39] [INFO] testing if GET parameter 'id' is dynamic
[19:16:39] [WARNING] GET parameter 'id' does not appear to be dynamic
[19:16:39] [WARNING] heuristic (basic) test shows that GET parameter 'id' mig
ht not be injectable
```

Figure 12 : Lancement de l'attaque SQL Injection avec SQLMap sur le serveur web

Immédiatement après le lancement de SQLMap, nous observons les détections dans pfSense.

10/21/2025 17:16:52	1	TCP	Attempted Administrator Privilege Gain	10.0.10.102 40276	10.0.20.100 80	1:2015749	ET WEB_SERVER Possible Oracle SQL Injection utl_inaddr call in URI
10/21/2025 17:16:52	1	TCP	Attempted Administrator Privilege Gain	10.0.10.102 40270	10.0.20.100 80	1:2015749	ET WEB_SERVER Possible Oracle SQL Injection utl_inaddr call in URI

Figure 13 : Alertes Suricata détectant l'attaque SQL Injection sur l'interface LAN_SERVER

```
┌──(root㉿hacker)-[~]
└─# hydra -l root -P /usr/share/wordlists/rockyou.txt ssh://10.0.20.100
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in
 military or secret service organizations, or for illegal purposes (this is n
on-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-10-21 19:
39:16
[WARNING] Many SSH configurations limit the number of parallel tasks, it is r
ecommended to reduce the tasks: use -t 4
[WARNING] Restorefile (you have 10 seconds to abort... (use option -I to skip
 waiting)) from a previous session found, to prevent overwriting, ./hydra.res
tore
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344399 login tries (l:1
/p:14344399), ~896525 tries per task
[DATA] attacking ssh://10.0.20.100:22/
```

Figure 14 : Lancement de l'attaque SSH Brute Force avec Hydra depuis Kali Linux

Date	Action	Pri	Proto	Class	Src	SPort	Dst	DPort	GID:SID	Description
10/21/2025 17:40:33	⚠	3	TCP	Generic Protocol Command Decode	10.0.10.102	56084	10.0.20.100	22	1:2228000	SURICATA SSH invalid banner
10/21/2025 17:40:33	⚠	3	TCP	Generic Protocol Command Decode	10.0.20.100	22	10.0.10.102	56084	1:2228000	SURICATA SSH invalid banner

Figure 15 : Alertes Suricata détectant les tentatives SSH sur l'interface LAN_SERVER

Ce test simule un scénario réaliste en entreprise : un poste utilisateur compromis tente d'exploiter des vulnérabilités sur les serveurs applicatifs. La segmentation réseau force le trafic à passer par pfSense/Suricata, assurant la détection des menaces internes. Notre couche IDS est maintenant opérationnelle et envoie les événements réseau vers Wazuh.

Cependant, Suricata seul ne suffit pas. Une architecture SOC complète nécessite également la surveillance des endpoints : activités utilisateurs, modifications de fichiers, élévations de privilèges, processus suspects. C'est le rôle de Wazuh, qui va centraliser les logs de Suricata (réseau) et des agents (système) pour offrir une visibilité complète.

4.3 Déploiement de Wazuh (SIEM)

Wazuh constitue le cœur de notre SIEM et a été déployé en mode Docker single-node sur une VM dédiée à l'adresse 10.0.30.100 avec 8 GB RAM et 4 cores CPU. Le déploiement comprend trois conteneurs : Wazuh Manager pour la corrélation et la détection, Wazuh Indexer basé sur OpenSearch pour le stockage et l'indexation, et Wazuh Dashboard pour la visualisation et l'interface web.

L'installation s'est effectuée via le script Docker Compose officiel fourni par Wazuh. Tous les services ont démarré correctement avec le Dashboard accessible sur le port 443, le Manager écoutant sur le port 1514 pour les agents, et l'Indexer en état "green" confirmant le bon fonctionnement du cluster.

Les agents Wazuh ont été déployés sur les machines à surveiller dans LAN_USER et LAN_SERVER. Chaque agent s'est enregistré automatiquement auprès du Manager via une clé d'authentification unique. Les agents sont apparus actifs dans le Dashboard avec une collecte de logs fonctionnelle.

Nous avons créé trois règles de détection personnalisées ciblant les attaques SSH brute force (règle 100001), les injections SQL (règle 100020) et les scans de ports (règle 100021). Ces règles

déclenchent des alertes de niveau 10 ou supérieur pour activer l'automatisation Shuffle. La validation a été effectuée avec des attaques de test depuis Kali Linux confirmant la détection correcte des patterns d'attaque.

Les détails complets de configuration incluant le fichier docker-compose.yml, l'installation des agents et les règles personnalisées sont disponibles en **Annexe C**.

4.4 Intégration pfSense/Suricata vers Wazuh

L'intégration des alertes Suricata dans Wazuh nécessite deux configurations principales : l'activation de la réception syslog sur le Manager et la création d'un décodeur JSON personnalisé.

Le fichier ossec.conf a été modifié pour activer l'écoute sur le port UDP 514 permettant à Wazuh de recevoir les logs syslog envoyés par pfSense/Suricata. Un décodeur JSON personnalisé a été créé pour parser correctement les événements EVE JSON en extrayant les champs essentiels tels que event_type, src_ip, dest_ip et alert.signature.

Les tests de validation ont confirmé que les alertes Suricata sont correctement reçues, parsées et corrélées avec les événements des agents Wazuh. L'intégration complète permet une visibilité uniforme du trafic réseau et des activités sur les endpoints.

Les fichiers de configuration complets ossec.conf et local_decoder.xml sont disponibles en **Annexe C.**

4.5 Déploiement de Shuffle (SOAR)

Shuffle est notre plateforme SOAR chargée d'orchestrer les réponses automatiques aux incidents. Nous l'avons déployé sur une VM dédiée (10.0.30.102) via Docker Compose.

4.5.1 Installation de Shuffle sur VM dédiée

Le déploiement de Shuffle comprend quatre conteneurs : opensearch (backend de recherche), backend (API), orborus (worker) et frontend (interface web). Après le démarrage, l'interface est accessible sur le port 3001.

Nous suivons la documentation officielle de Shuffle disponible sur : Shuffle configuration documentation

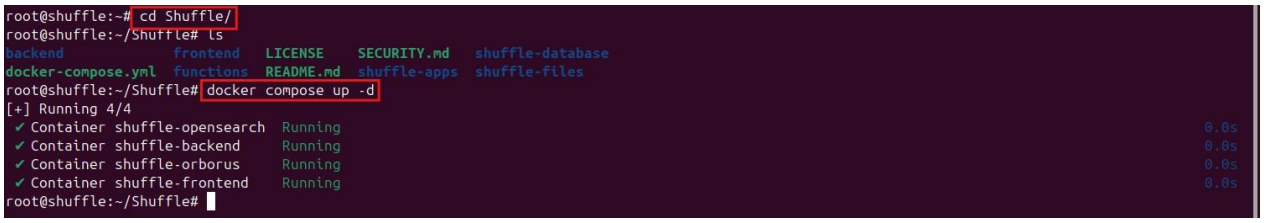


Figure 16: Démarrage des conteneurs Docker Shuffle (opensearch, backend, orborus, frontend)

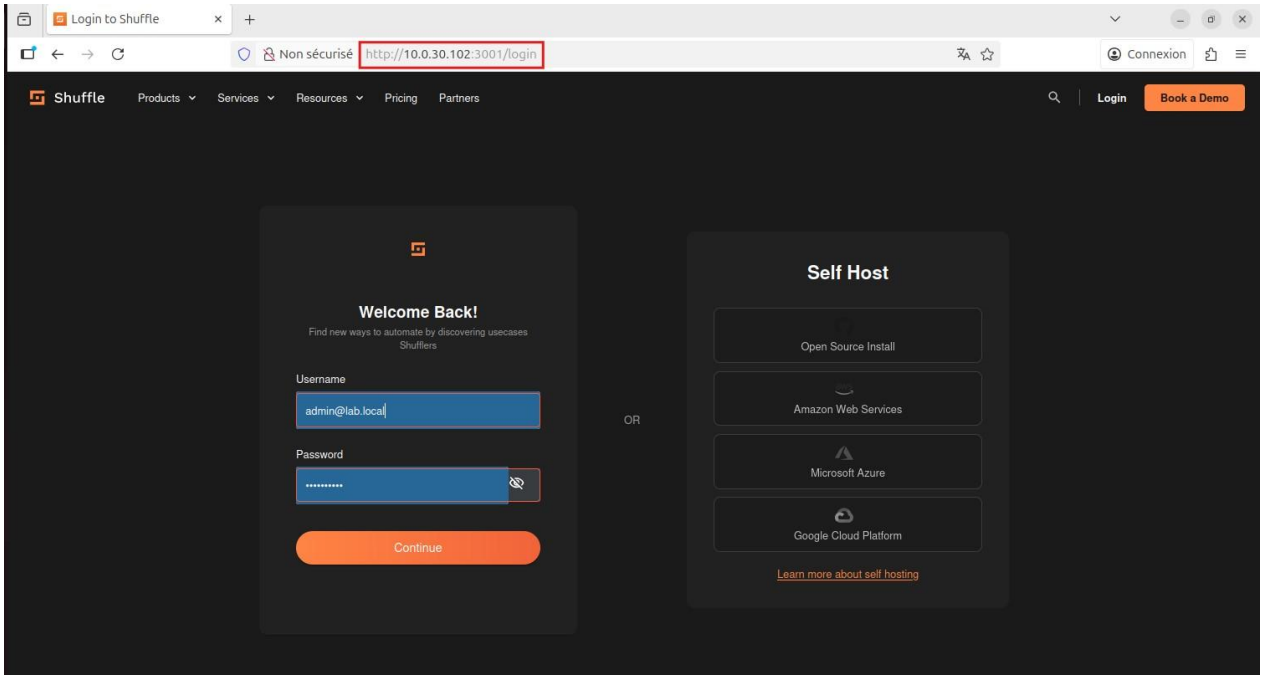


Figure 17: Page de connexion à l'interface Shuffle

4.5.2 Configuration des webhooks et workflows

Nous avons créé un workflow nommé 'Wazuh-Security-Alerts-Handler' pour traiter automatiquement les alertes. Un webhook a été généré pour recevoir les alertes de Wazuh.

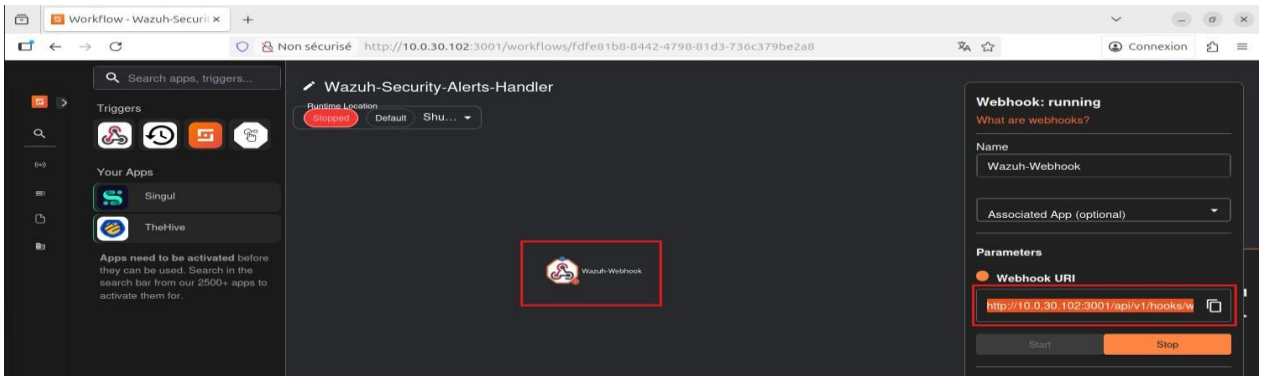


Figure 18: Configuration du webhook Wazuh dans Shuffle avec l'URI d'intégration

L'intégration côté Wazuh a été configurée dans ossec.conf pour envoyer toutes les alertes de niveau ≥10 vers le webhook Shuffle via HTTP POST.

```
  <!-- Integration avec Shuffle -->
  <integration>
    <name>shuffle</name>
    <hook_url>http://10.0.30.102:3001/api/v1/hooks/webhook_c3b3a3e3-9419-4000-9f85-cee15cb4762c</hook_url>
    <level>10</level>
    <alert_format>json</alert_format>
  </integration>

</ossec_config>
```

Figure 19: Configuration de l'intégration Shuffle dans le fichier ossec_config de Wazuh

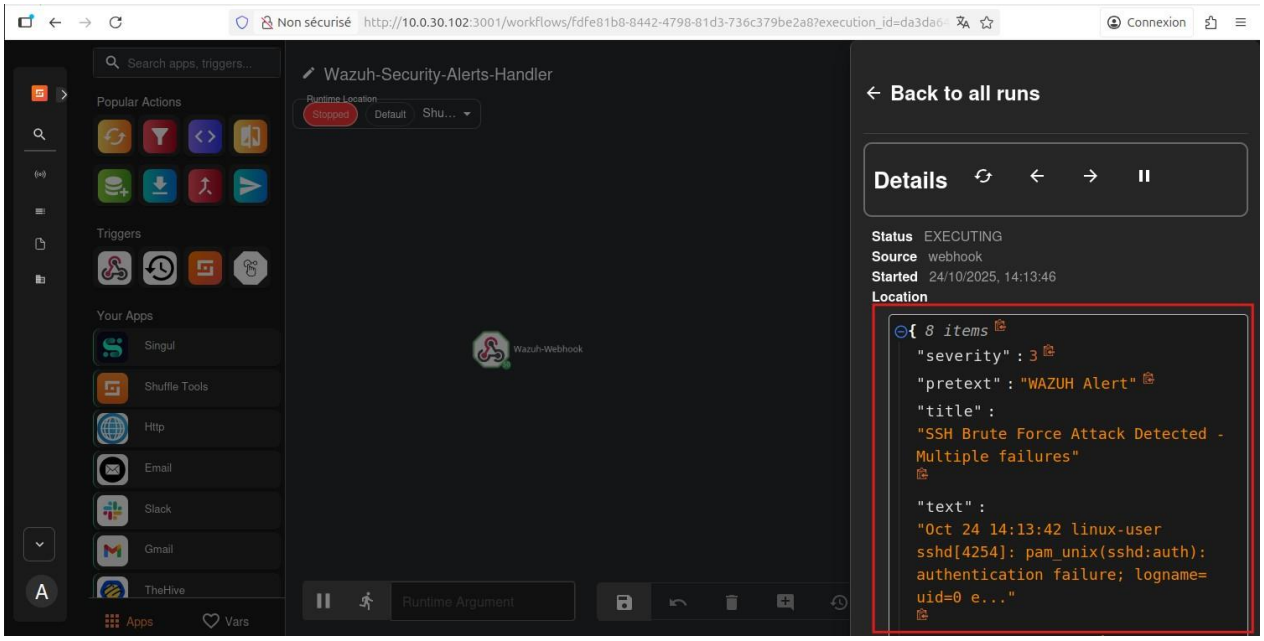


Figure 20: Réception et affichage d'une alerte SSH Brute Force dans Shuffle

4.6 Déploiement de TheHive, Cortex et MISP

La stack complète de gestion des incidents (TheHive, Cortex, MISP) a été déployée sur une VM dédiée (10.0.30.104) avec 12 GB RAM. Ces trois plateformes travaillent en synergie pour la gestion, l'analyse et le partage de threat intelligence.

4.6.1 Préparation de l'environnement

Avant le déploiement, nous avons préparé l'environnement en installant Docker et Docker Compose, puis créé un répertoire dédié pour les fichiers de configuration.

```
mkdir -p ~/thehive-platform  cd ~/thehive-platform  mkdir -p cortex/{logs,config,neurons} server-configs logs ssl files misp
```

Cette arborescence organise les fichiers de configuration et les données persistantes de chaque composant.

4.6.2 Création du fichier Docker Compose

Un fichier docker-compose.yml a été créé intégrant tous les services nécessaires : Cassandra (base de données TheHive), Elasticsearch (indexation), TheHive, Cortex, MISP et MySQL (base MISP). Cette configuration en conteneurs facilite le déploiement et la gestion de la plateforme complète.[2]

4.6.3 Déploiement de la stack complète

Le déploiement s'effectue avec la commande **docker compose up -d**. Après quelques minutes, tous les conteneurs sont opérationnels.

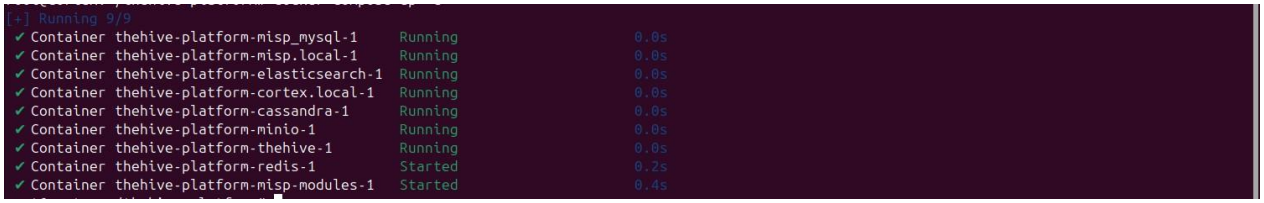


Figure 21: Déploiement réussi de la stack TheHive Platform

4.6.4 Accès aux interfaces web

Les trois plateformes sont accessibles sur leurs ports respectifs : TheHive (9000), Cortex (9001) et MISP (8080). Les identifiants par défaut ont été changés lors de la première connexion.

Tableau 10: Accès aux interfaces web des plateformes TheHive, Cortex et MISP

Plateforme	URL	Identifiants par défaut
TheHive	http://10.0.30.104:9003	admin@thehive.local / secret
Cortex	http://10.0.30.104:9001	admin@cortex.local / admin
MISP	http://10.0.30.104:8080	admin@admin.test / admin

[2] *Le fichier complet thehive_platform.yml est disponible dans le dépôt GitHub du projet : https://github.com/FodeMangane/soc-automation*

4.6.5 Configuration initiale de TheHive

Après la première connexion, nous avons créé une organisation nommée 'Fomarix[3]' et un utilisateur API avec les permissions nécessaires pour l'intégration avec Shuffle.

Création de l'organisation

Une fois connectés à TheHive, nous avons créé l'organisation destinée à notre SOC en passant par

Administration → Organisations → Add organization, avec fomarix comme nom et Security Operations Center Laboratory comme description.

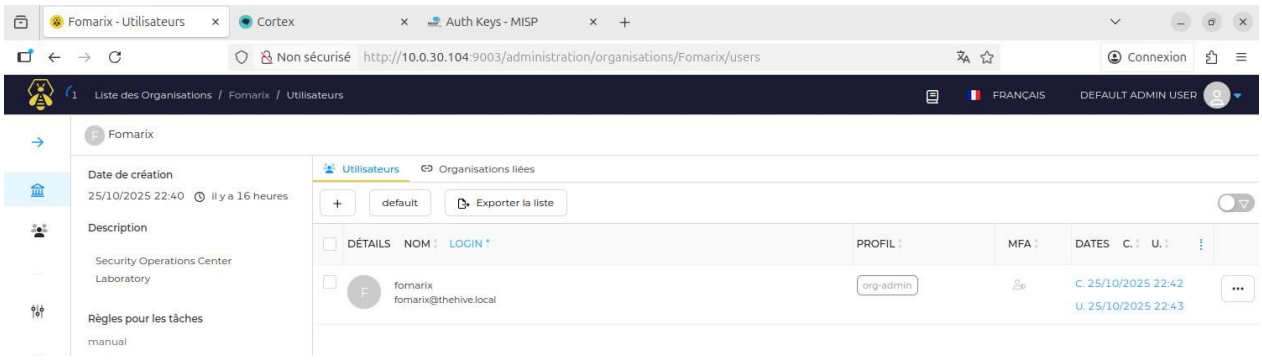


Figure 22: Création de l'organisation Fomarix et de l'utilisateur API pour Shuffle

Création de l'utilisateur API pour Shuffle

Afin de permettre à Shuffle de créer automatiquement des cas, nous avons ajouté un utilisateur dédié via Administration → Utilisateurs → Add user, en définissant fomarix@thehive.local comme identifiant, fomarix comme nom, avec le profil org-admin et rattaché à l'organisation Fomarix.

[3] *"Fomarix" est un nom fictif créé à partir des initiales de l'auteur (**FO**de **MA**ngane) associées au suffixe "**-rix**", et utilisé pour désigner l'organisation SOC mise en place dans ce projet.*

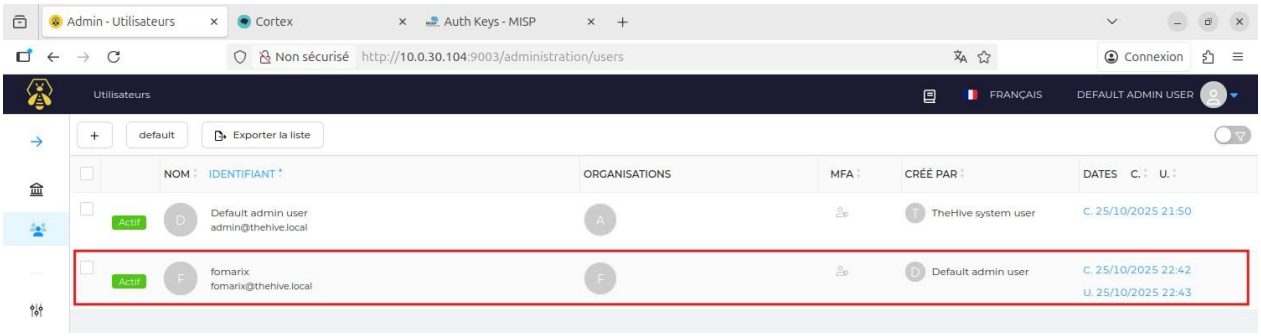


Figure 23 : Liste des utilisateurs dans l'interface d'administration de TheHive

Nous avons ensuite généré une clé API pour cet utilisateur en accédant à son profil (API Keys → Create API Key) et en renseignant la description Shuffle Workflow Integration, puis nous avons copié la clé produite afin de la conserver en lieu sûr.

4.6.7 Configuration initiale de MISP

Dans MISP, nous avons généré une clé API pour permettre l'intégration avec TheHive. Cette clé sera utilisée pour créer automatiquement des événements de threat intelligence.

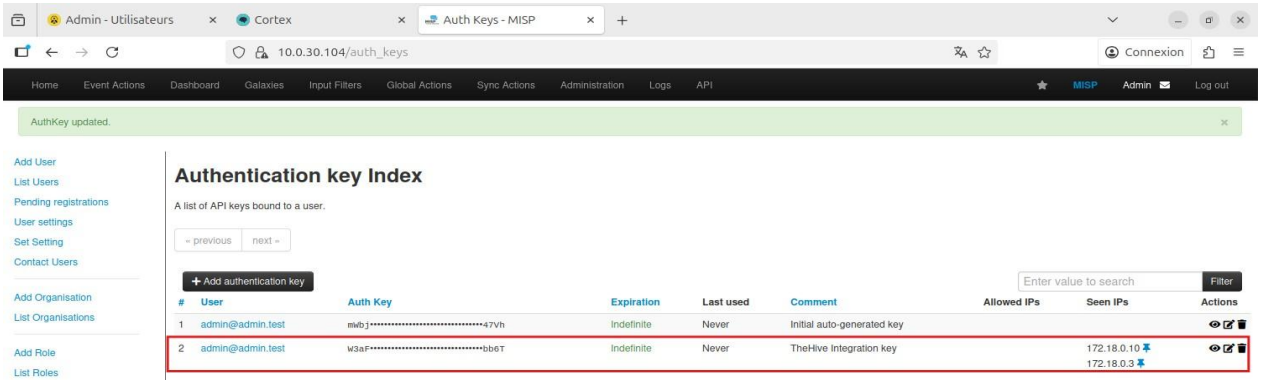


Figure 24: Génération et gestion des clés d'authentification API dans MISP

4.6.8 Interconnexion TheHive ↔ Cortex

Pour automatiser l'analyse des observables entre TheHive et Cortex, nous avons d'abord créé un utilisateur dédié dans Cortex (fomarix@cortex.local) et généré sa clé API.

Nous avons ensuite configuré le connecteur dans TheHive via Administration → Gestion de la Plateforme → Cortex

- **Nom du serveur :** cortex0
- **URL du serveur :** http://cortex.local:9001
- **Clé API :** clé associée à l'utilisateur *fomarix@cortex.local*
- **Nombre maximal de tentatives :** 3
- **Délai de rafraîchissement :** 5 secondes
- **Intervalle de vérification :** 1 minute

Après avoir renseigné ces informations, nous avons cliqué sur **Tester la connexion serveur** afin de vérifier la configuration, ce qui a affiché le message : « La configuration Cortex a été testée avec succès ». Enfin, nous avons validé l'ensemble en sélectionnant Mettre à jour.

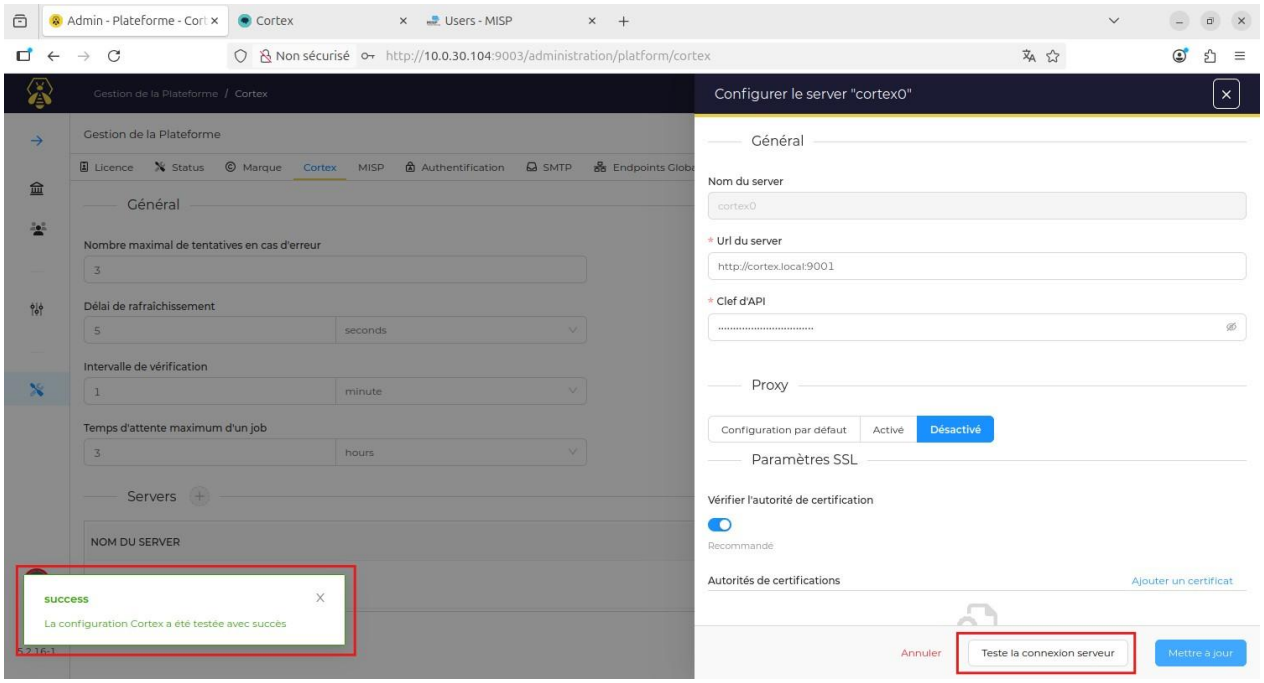


Figure 25: Configuration et test de l'interconnexion TheHive-Cortex

4.6.9 Interconnexion TheHive ↔ MISP

La connexion TheHive-MISP permet de partager automatiquement les IOCs détectés. Nous avons configuré le serveur MISP dans TheHive avec la clé API générée précédemment.

Configuration du connector MISP dans TheHive

- **Nom du serveur :** MISP
- **URL :** http://misp.local
- **Clé API :** clé générée pour MISP
- **Purpose :** ImportAndExport
- **Tags :** fomarix, thehive

Après avoir renseigné ces informations, nous avons cliqué sur Test afin de vérifier la connexion avec le serveur MISP, puis nous avons sauvegardé la configuration en sélectionnant Save.

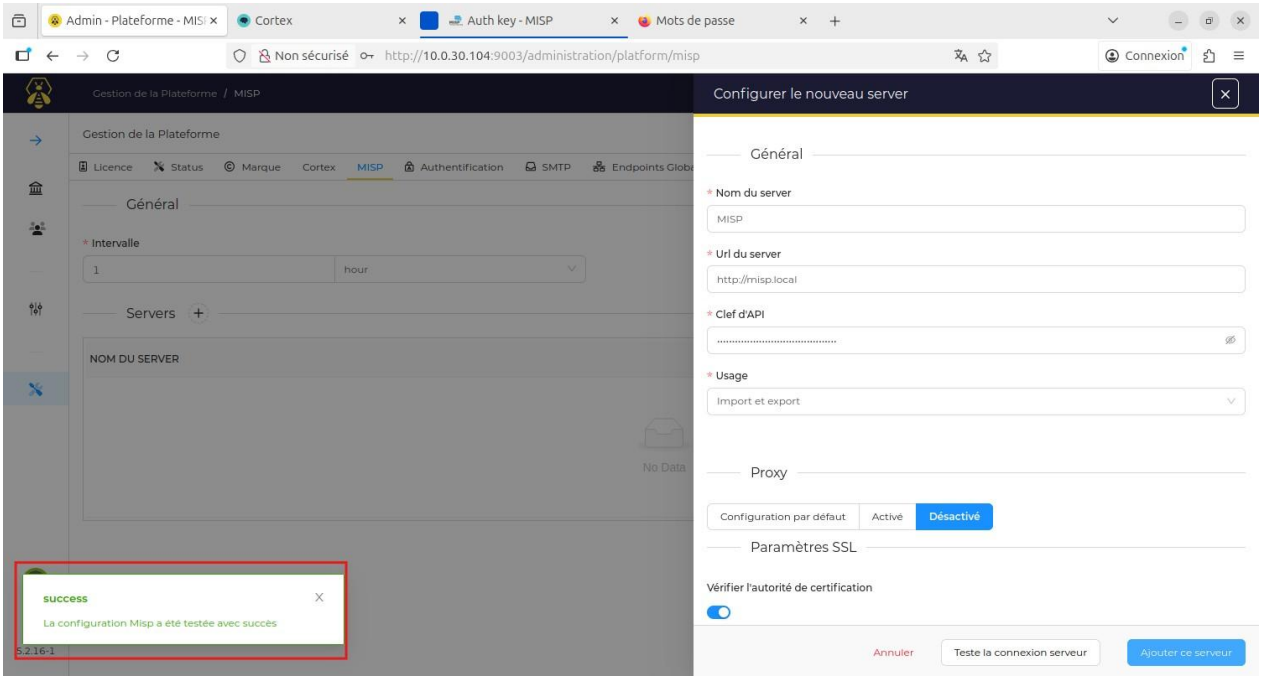


Figure 26: Configuration et test de l'interconnexion TheHive-MISP

Configuration de l'import automatique

Dans la configuration de MISP, nous avons activé l'import automatique en définissant *Import from* sur les 7 derniers jours et *Interval* sur 1 heure, ce qui permet de récupérer automatiquement les événements récents de MISP sous forme d'alertes dans TheHive.

4.7 Configuration de l'automatisation complète

Cette section détaille la construction du workflow d'automatisation complet qui constitue le cœur de notre SOC automatisé. Ce workflow traite automatiquement chaque incident critique détecté par Wazuh, de la création du cas jusqu'à la notification.

4.7.1 Architecture du workflow d'automatisation

Le workflow se compose de huit étapes exécutées séquentiellement : réception de l'alerte Wazuh via webhook, extraction et parsing des données, création automatique du cas dans TheHive, ajout des observables, lancement d'analyses Cortex, création d'événement MISP, blocage de l'IP source et notification Slack.

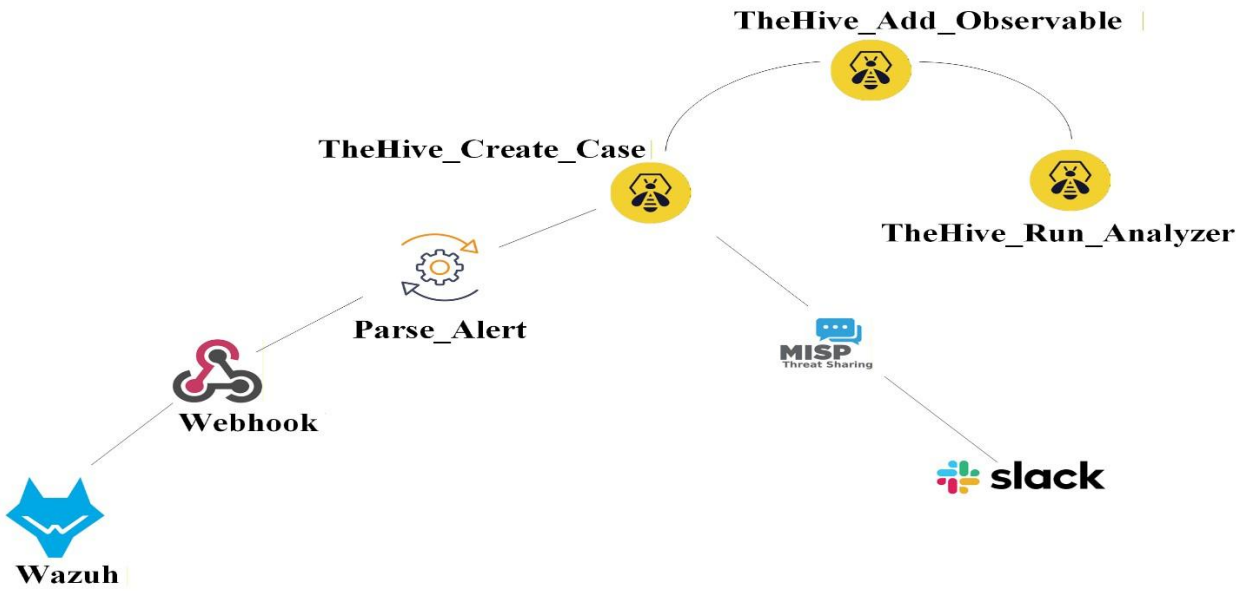


Figure 27: Architecture du workflow d'automatisation Wazuh-Shuffle-TheHive-Cortex-MISP-Slack

4.7.2 Configuration préalable - Création du Bot Slack

Avant de construire le workflow, nous avons créé une application Slack nommée *SOC Bot* pour recevoir les notifications et généré un token OAuth permettant à Shuffle d'authentifier les

requêtes API. La configuration implique la création de l'application, l'attribution des permissions nécessaires (chat:write, chat:write.public, channels:read) et l'installation dans le workspace.[4]

4.7.3 Construction du workflow dans Shuffle

Le workflow d'automatisation a été créé dans Shuffle pour orchestrer l'ensemble de la chaîne de détection et de réponse. Ce workflow nommé "Wazuh-Security-Alerts-Handler" intègre sept modules principaux exécutés séquentiellement.

Le workflow commence par un Webhook recevant les alertes Wazuh en format JSON. Le module Parse_Alert extrait ensuite les champs critiques tels que l'IP source, le type d'attaque et le niveau de sévérité. Le module TheHive_Create_Case crée automatiquement un cas d'incident avec toutes les informations pertinentes. Le module TheHive_Add_Observable ajoute l'adresse IP source comme observable marqué IOC. Le module TheHive_Run_Analyzer déclenche l'analyse automatique via Cortex et VirusTotal. Le module MISP_Create_Event documente l'incident dans la plateforme de threat intelligence. Enfin, le module Send_Alert_to_Slack notifie l'équipe SOC en temps réel.

Chaque module a été configuré avec les paramètres d'authentification appropriés incluant les URLs des plateformes, les clés API et les tokens d'accès. Les données transitent entre modules via des variables permettant de conserver le contexte tout au long du workflow. Le temps d'exécution total du workflow de bout en bout est d'environ 27 secondes.

L'ensemble du processus de création et de configuration détaillée de chaque module est décrit dans **l'Annexe D**.

[4] *Les étapes détaillées sont disponibles dans Slack_Bot.md sur GitHub :* https://github.com/FodeMangane/soc-automation

Chapitre 5 : Test, évaluation et analyse

Ce chapitre présente la validation complète de notre SOC automatisé à travers des tests de bout en bout, l'analyse comparative des performances, et les résultats obtenus. Nous exposons également les difficultés rencontrées durant l'implémentation, les limites de notre solution et les perspectives d'évolution future. Enfin, nous concluons en répondant à la problématique posée et en synthétisant les contributions de ce travail.

5.1 Test et validation du workflow de bout en bout

Maintenant que le workflow est entièrement opérationnel, nous pouvons réaliser un test complet afin de valider l'ensemble de la chaîne d'automatisation.

5.1.1 État initial avant le test

Avant de lancer l'attaque simulée, nous vérifions l'état initial des différentes plateformes afin de mesurer précisément l'impact du processus automatisé. Du côté de TheHive, aucun cas n'est présent et le tableau de bord est totalement vide, ce qui garantit que les résultats observés après le test proviendront exclusivement du workflow que nous avons mis en place.

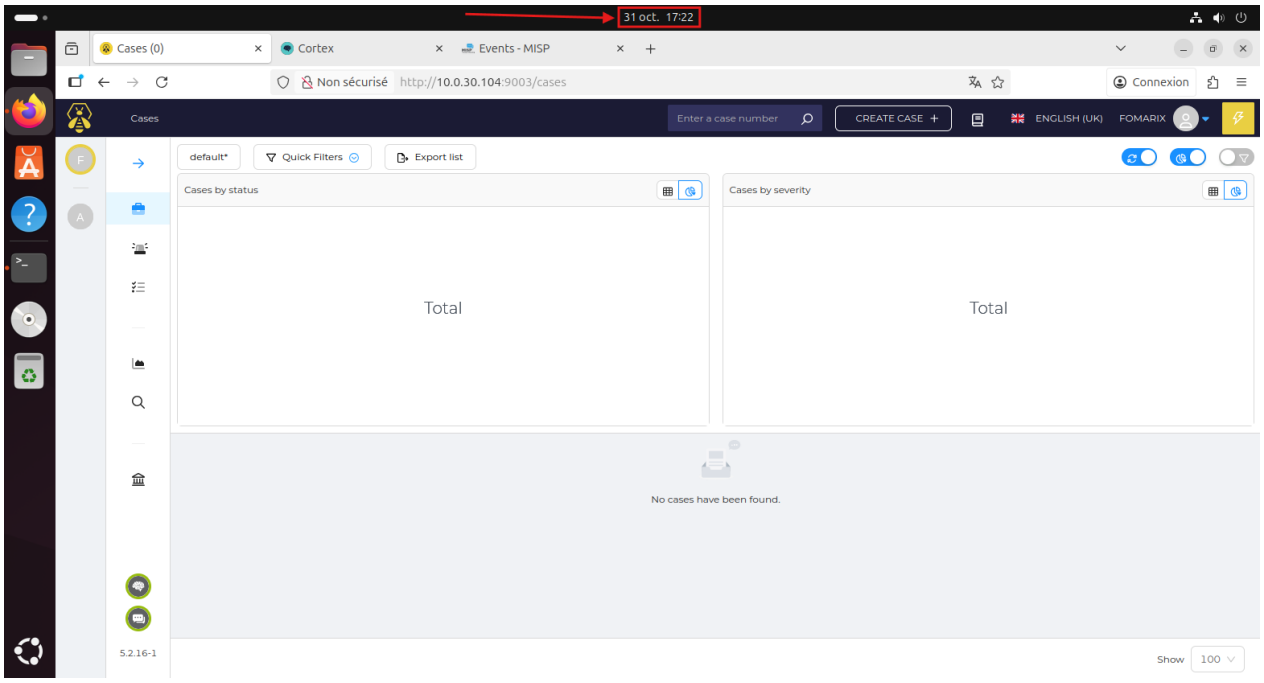


Figure 28: État initial du dashboard TheHive avant le test (aucun cas présent)

L'instance Cortex est également vide, sans aucun job enregistré ni analyse en cours, ce qui confirme que le test débutera dans un environnement totalement propre.

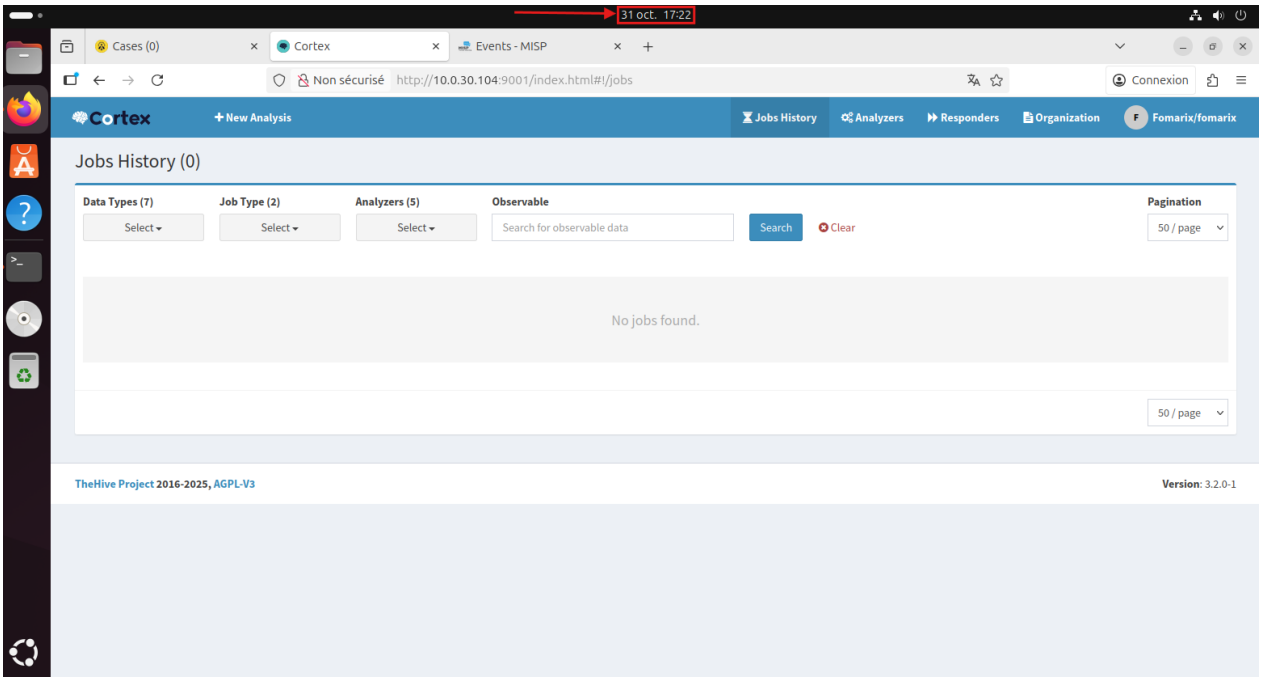


Figure 29: État initial de l'historique des jobs Cortex (aucune analyse en cours)

La plateforme MISP ne contient aucun événement ni IOC enregistré, garantissant que le test démarre sans données préexistantes.

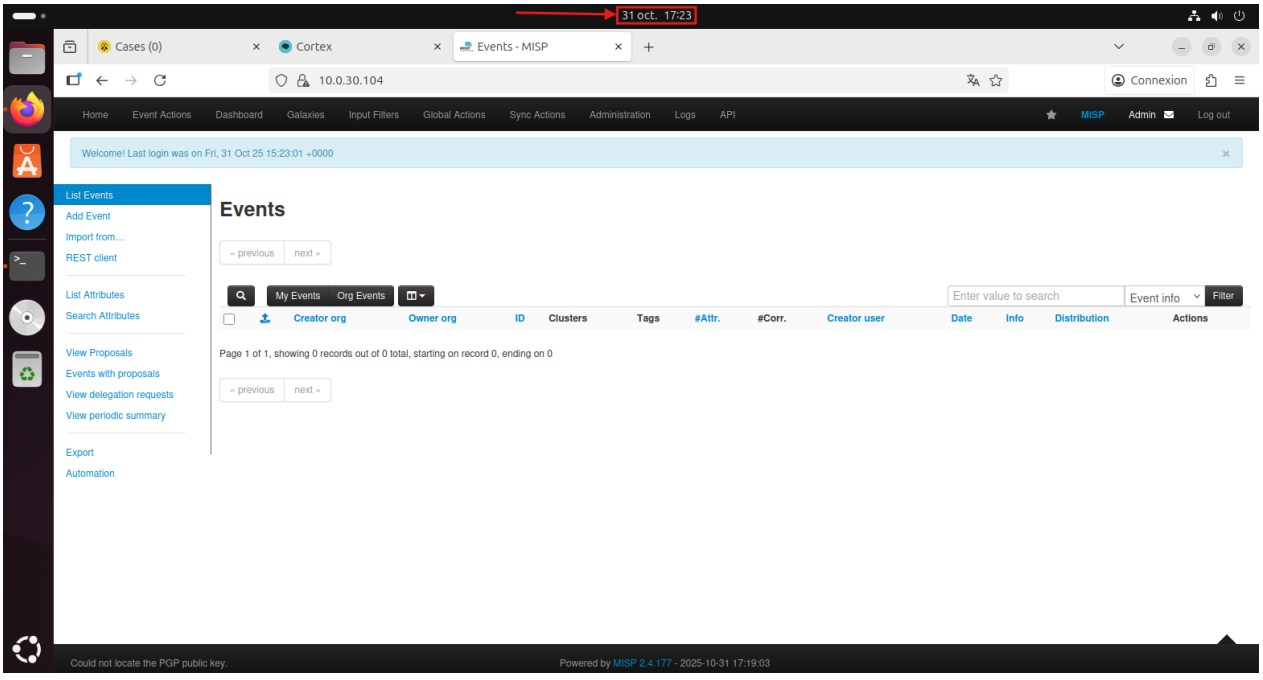


Figure 30: État initial de la liste des événements MISP (aucun IOC référencé)

Sur la machine Linux User (10.0.10.100), l'heure système est 17:24:19 et une vérification via la commande iptables -L -n confirme qu'aucune adresse IP n'est actuellement bloquée.

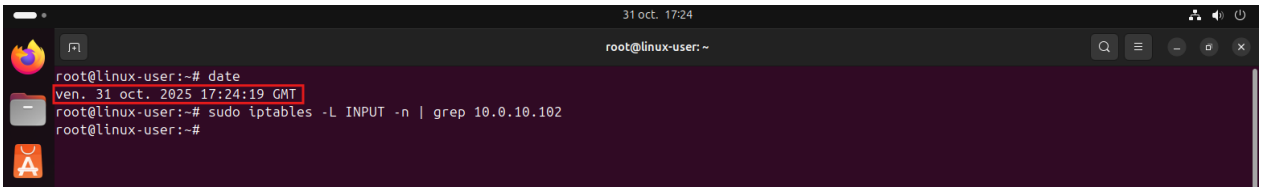


Figure 31: Vérification de l'état initial de la machine Linux User sans IP bloquée

5.1.2 Lancement de l'attaque SSH Brute Force

Depuis la machine Kali Linux (10.0.10.102), nous lançons une attaque SSH brute force ciblant Linux User (10.0.10.100).

Heure de lancement de l'attaque : **17:25:49**

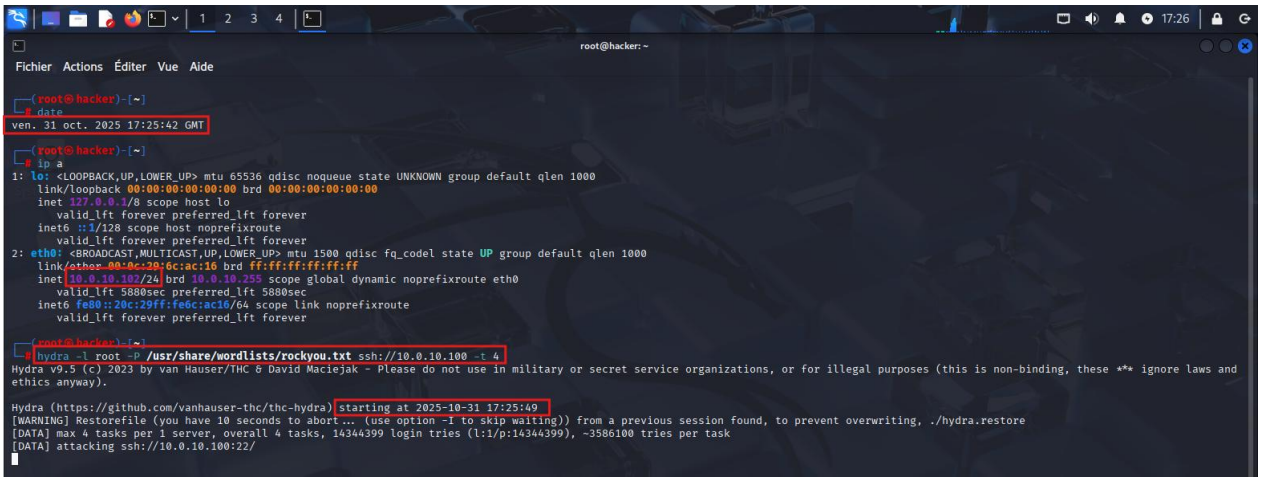


Figure 32: Lancement de l'attaque SSH Brute Force depuis Kali Linux (10.0.10.102)

Dès la détection de l'attaque brute force (règle 100001, 5763 ou 2502), le mécanisme Wazuh **Active Response** se déclenche automatiquement et procède au blocage de l'adresse IP source sur le pare-feu de la machine Linux User.

L'attaque a été lancée à **17:25:49** et Wazuh a détecté puis bloqué l'adresse malveillante à **17:26:11**, soit **22 secondes** après le début de l'attaque. Une vérification sur la machine Linux User via la commande **iptables -L -n** confirme effectivement que le blocage automatique a bien été appliqué.

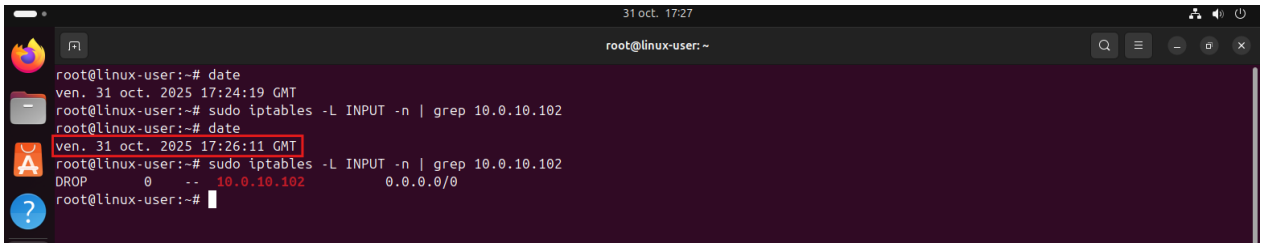


Figure 33: Détection et blocage automatique de l'IP attaquante par Wazuh via iptables

L'adresse IP de l'attaquant est maintenant isolée et ne peut plus communiquer avec la machine. Le délai de 22 secondes correspond uniquement au temps humain nécessaire pour ouvrir le terminal et vérifier l'état d'iptables ; le blocage, lui, a été appliqué instantanément dès le déclenchement de l'Active Response par Wazuh.

Au même moment, l'alerte est transmise automatiquement à Shuffle via le webhook, déclenché à 17:26:08.

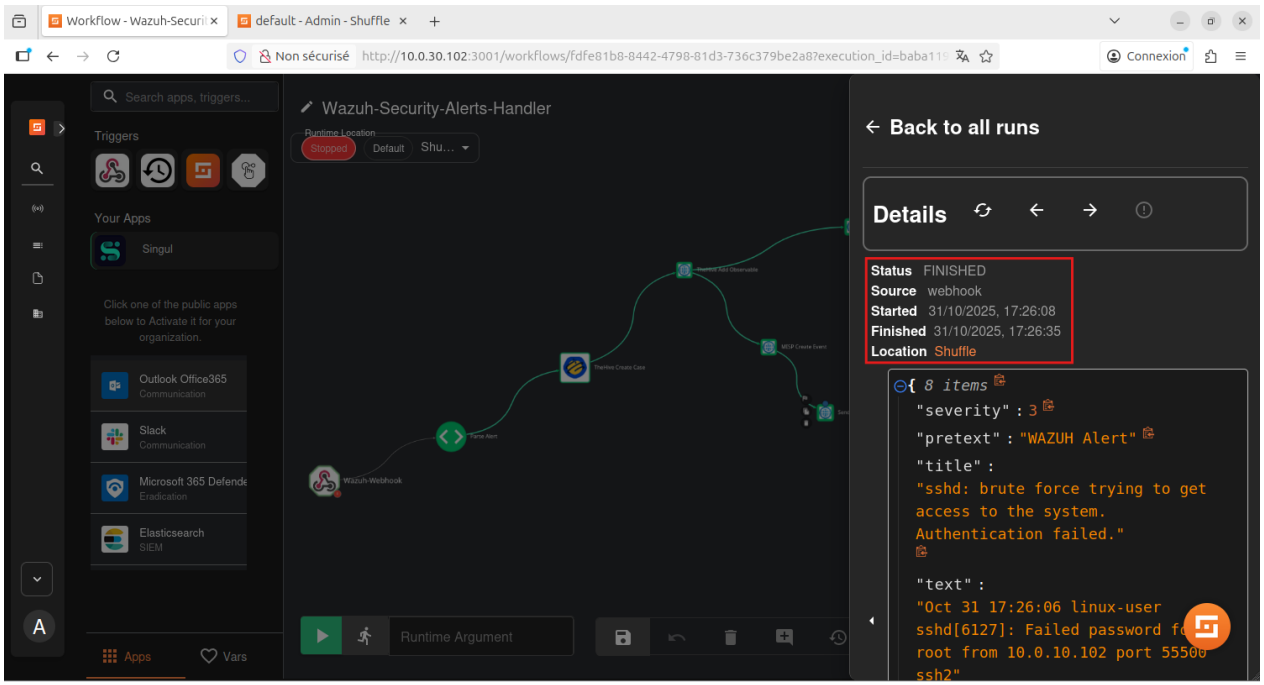


Figure 34: Déclenchement du workflow Shuffle suite à l'alerte Wazuh (17:26:08)

5.1.3 Détails de l'exécution du workflow

Nous avons accédé aux détails de l'exécution dans Shuffle afin d'analyser chaque étape du workflow. Le statut global indique que le workflow s'est terminé avec succès (FINISHED). Il a démarré le 31/10/2025 à 17:26:08 et s'est achevé à 17:26:35, soit une durée totale de **27 secondes**.

Pour la première action, Wazuh_Webhook, le statut est SUCCESS et les données reçues correspondent à l'alerte Wazuh, avec la règle **rule_id 5763**, l'IP source **10.0.10.102** et l'agent linux-user.

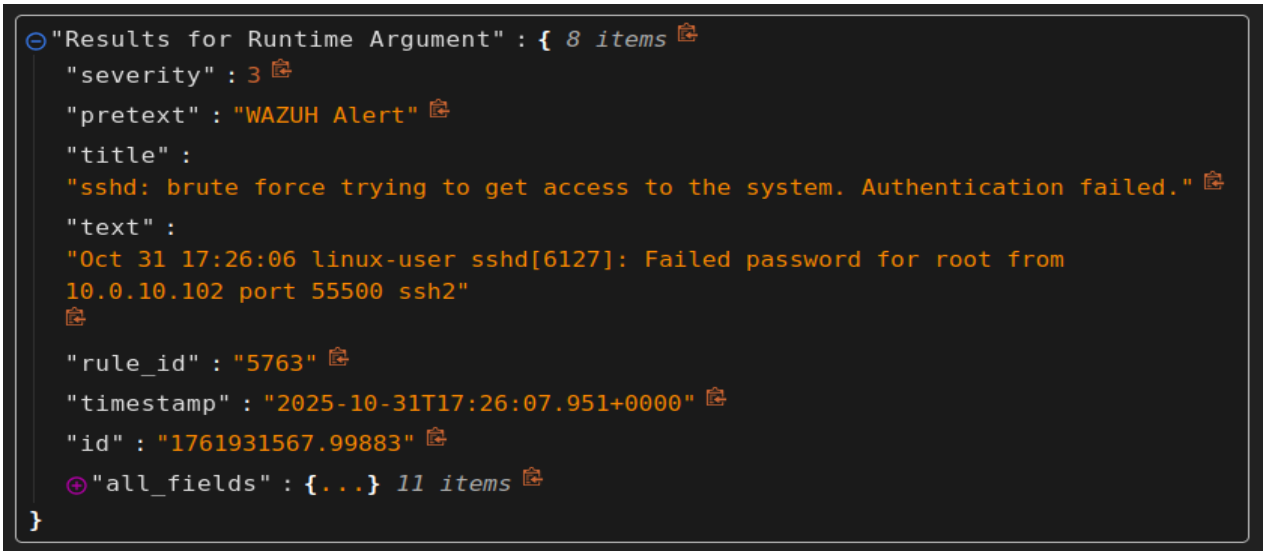


Figure 35: Réception de l'alerte SSH Brute Force dans le Runtime Argument de Shuffle

L'action Parse_Alert a également été exécutée avec succès (SUCCESS). Elle a permis d'extraire et de structurer les informations importantes de l'alerte Wazuh, facilitant ainsi leur utilisation dans les étapes suivantes du workflow.

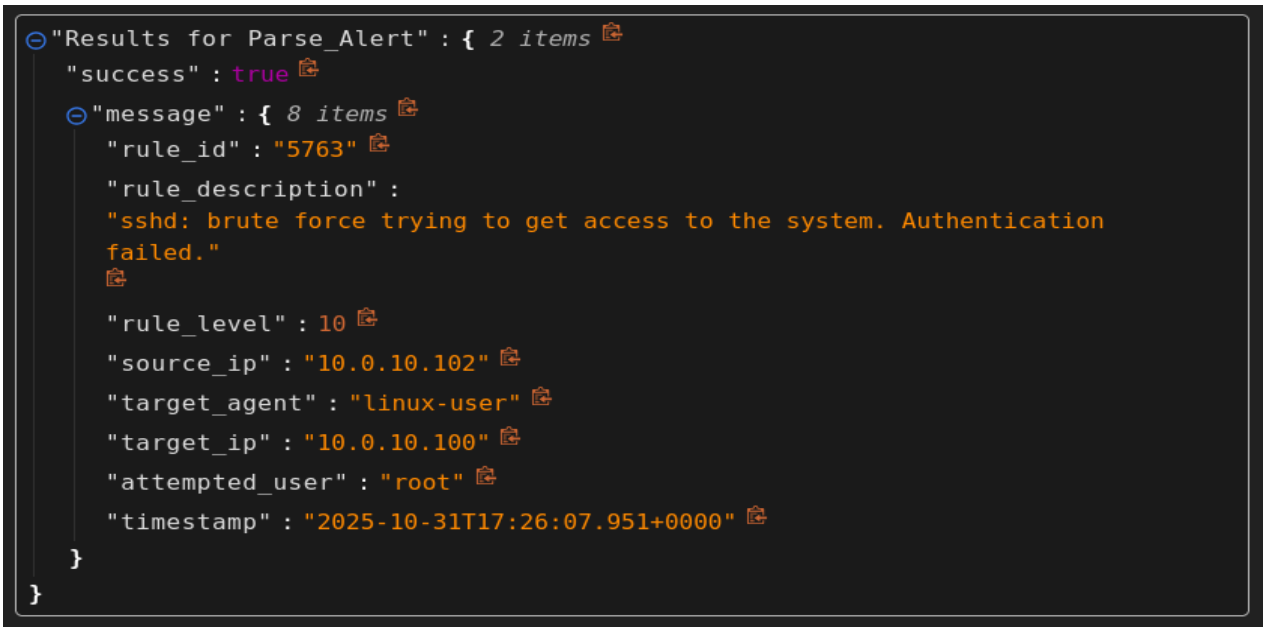


Figure 36: Résultat de l'extraction des données par le module Parse_Alert

L’action TheHive_Create_Case s’est conclue avec succès, renvoyant le statut 201 Created. Elle a créé le cas n°1, identifié par l’ID **~122884200**, avec pour titre **« SSH Brute Force Attack »**, ce qui confirme la bonne insertion des informations issues de Parse_Alert dans TheHive.

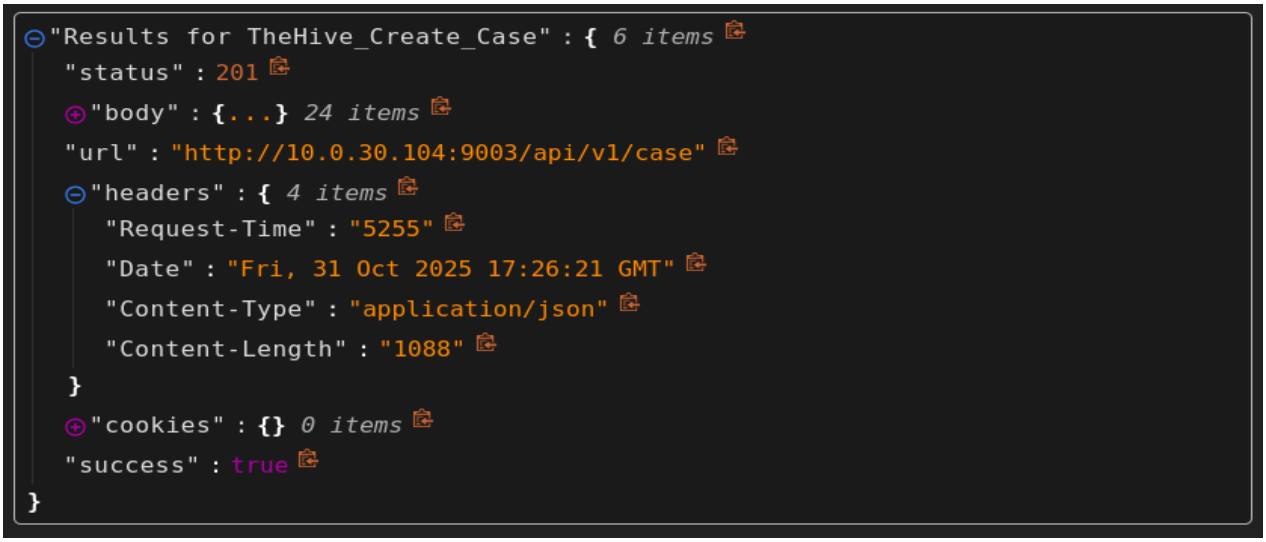


Figure 37: Résultat de la création automatique du cas dans TheHive (status 201)

Après la création du cas dans TheHive, l’action TheHive_Add_Observable a été exécutée avec succès, renvoyant le statut 201 Created. Elle a ajouté l’adresse IP **10.0.10.102** comme observable de type ip dans le cas nouvellement créé, avec l’identifiant **~122888296**, et l’a marquée comme Indicator of Compromise (IOC). Cette étape permet de lier automatiquement l’IP attaquante au cas, facilitant ainsi les analyses et actions suivantes dans le workflow.

Figure 38: Résultat de l'ajout de l'observable IP dans TheHive (status 201)

Ensuite, l'action TheHive_Run_Analyzer a été lancée avec succès, renvoyant le statut 201 Created. Elle a créé un job d'analyse identifié par **~163848192** et a exécuté l'analyseur **VirusTotal** avec l'ID **0f12d003ba1b420c3353458c97d97b1c**, permettant ainsi d'enrichir automatiquement l'observable ajouté avec des informations de threat intelligence.

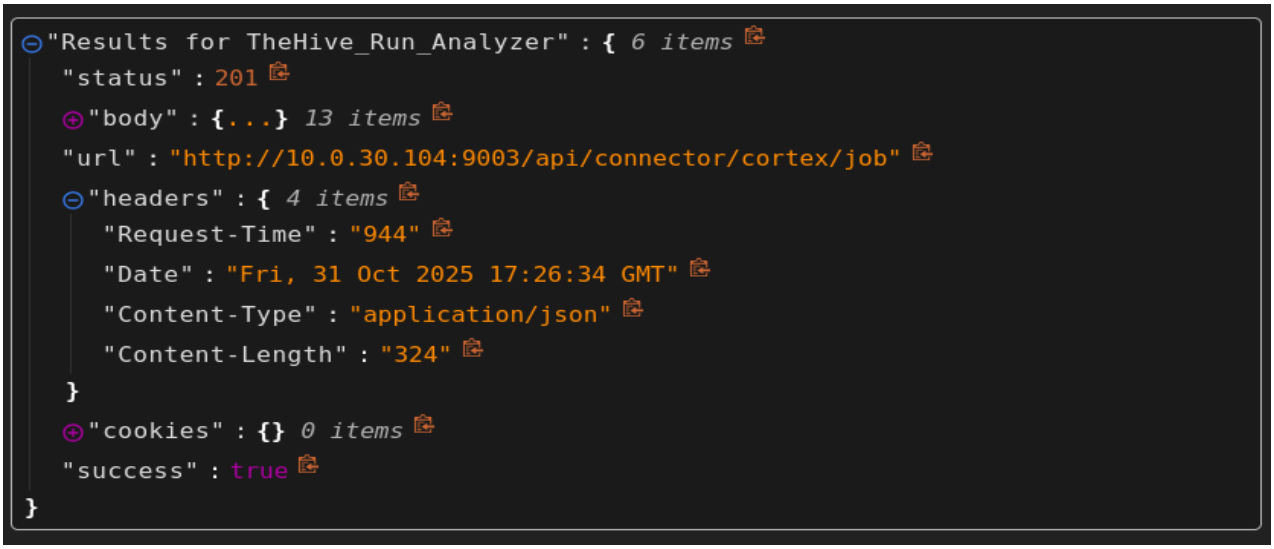


Figure 39: Résultat du lancement de l'analyzer Cortex (status 201)

Puis, MISP_Create_Event a été exécutée avec succès et a renvoyé **200 OK** : un événement a été créé (ID **1**) avec pour info "SSH Brute Force from 10.0.10.102 - Rule 5763", publié automatiquement, contenant **1 attribut** (l'IP) et les **7 tags** configurés (dont tlp:amber, type:OSINT, ssh-brute-force, mitre-attack:T1110.001).

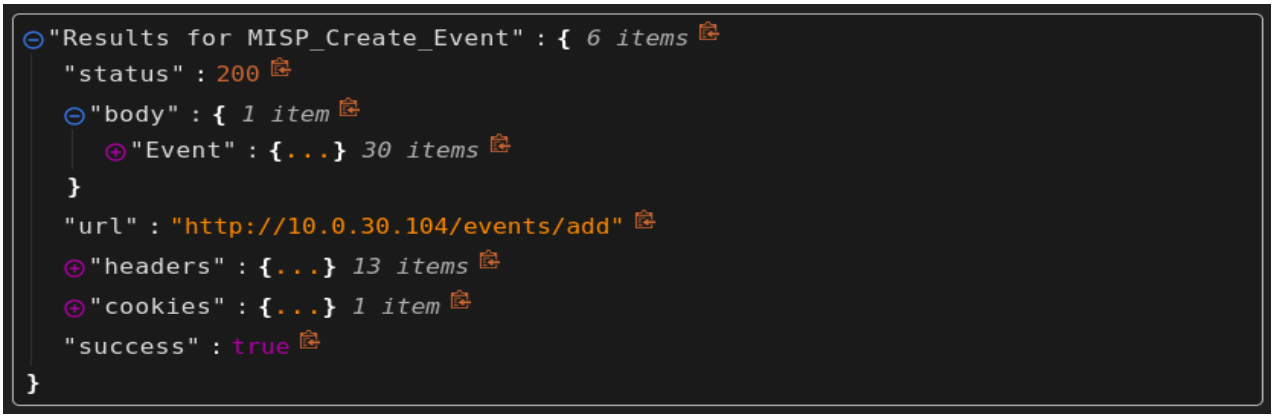


Figure 40: Résultat de la création de l'événement dans MISP (status 200)

Ensuite, la notification Slack a été envoyée : l'action Send_Alert_to_Slack a retourné **200 OK**, le message a été posté dans le canal **#soc-alerts** (ID **C09P52Q30DB**) et le serveur Slack a

renvoyé le timestamp du message **1761931595.546069**, confirmant la bonne diffusion de l'alerte à l'équipe SOC.

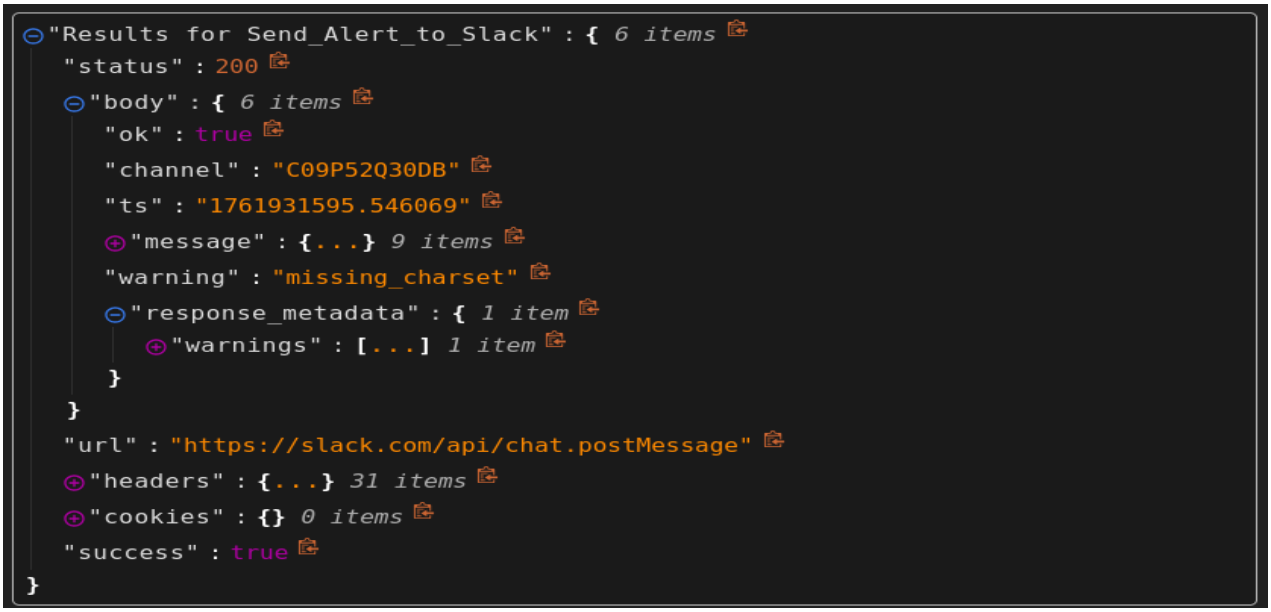


Figure 41: Résultat de l'envoi de la notification Slack au canal #soc-alerts (status 200)

5.1.4 Validation des résultats

Nous avons consulté l'interface TheHive pour vérifier le cas créé automatiquement. Le cas apparaît sous le numéro **#1**, intitulé **« SSH Brute Force Attack »** (ID **~122884200**), de sévérité **HIGH (3)**, au statut **New**, avec les marquages **TLP : AMBER (2)** et **PAP : AMBER (2)**. Il est étiqueté *wazuh*, *Bruteforce*, assigné à *fomarix@thehive.local* et a été créé le **31/10/2025 à 17:26:17**.

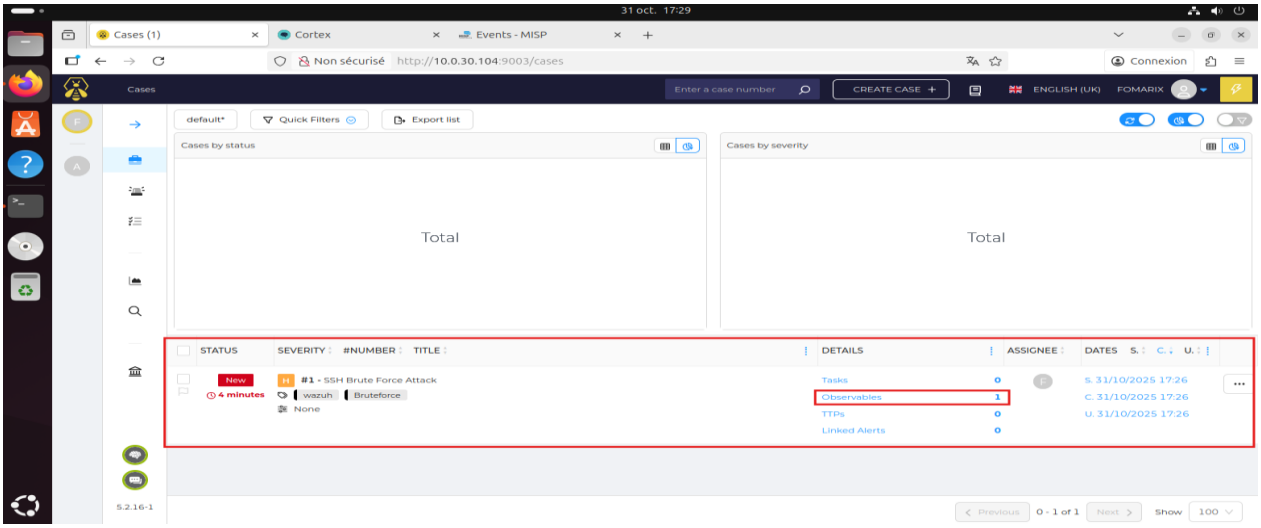


Figure 42: Cas automatiquement créé dans TheHive (#1 - SSH Brute Force Attack)

Le cas contient toutes les informations extraites par le workflow Shuffle dans la description, incluant les détails de la règle Wazuh, les informations de l'attaque (IP source, cible, utilisateur tenté), et la liste des actions automatiques déjà effectuées.

Ensuite, dans l'onglet **Observables** du cas, l'adresse IP source a bien été ajoutée automatiquement : il s'agit de l'observable **~122888296**, de type ip, valeur **10[.]0[.]10[.]102** (notation défanged pour sécurité), marquée **IOC : Oui**, avec les marquages **TLP : AMBER (2)** et **PAP : AMBER (2)**, les tags wazuh et ssh-bruteforce, et le message "IP source attaque SSH". Cet observable a été créé le **31/10/2025 à 17:26:29**.

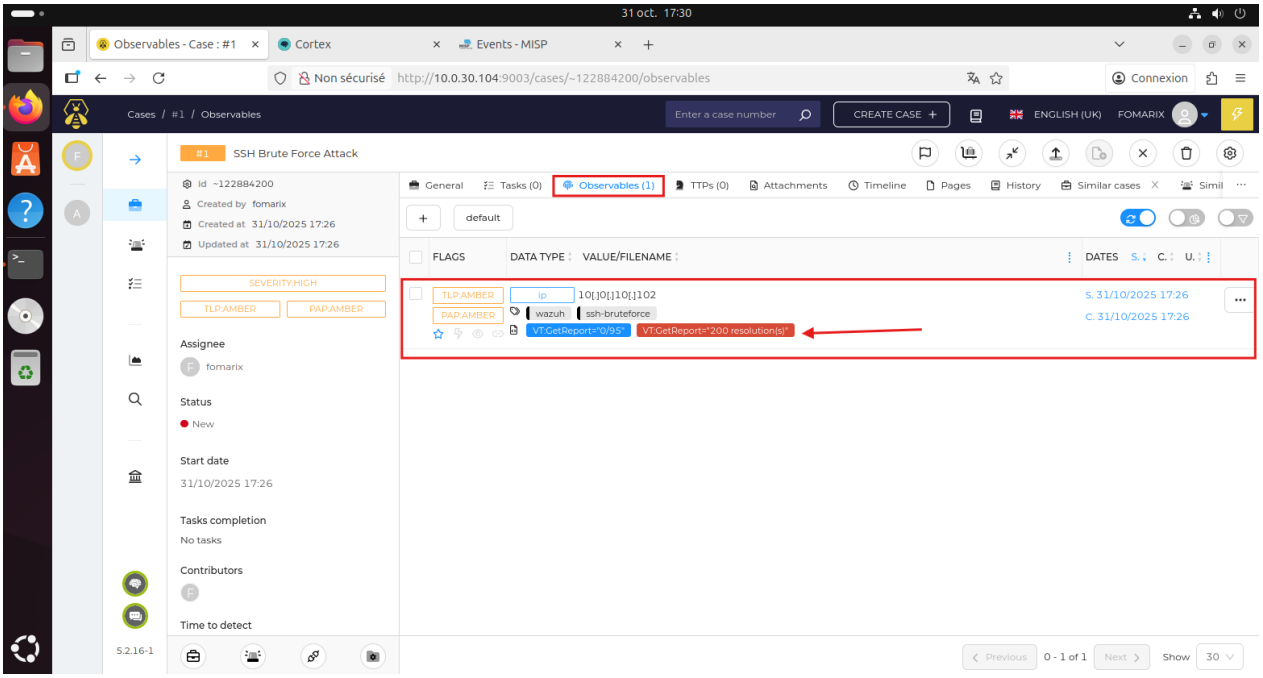


Figure 43: Observable IP source (10.0.10.102) automatiquement ajouté dans l'onglet Observables

Quelques secondes après, l'analyse VirusTotal lancée automatiquement par le workflow a remonté ses résultats dans TheHive. Il suffit de cliquer sur l'observable et d'accéder à l'onglet Reports pour consulter les informations enrichies.

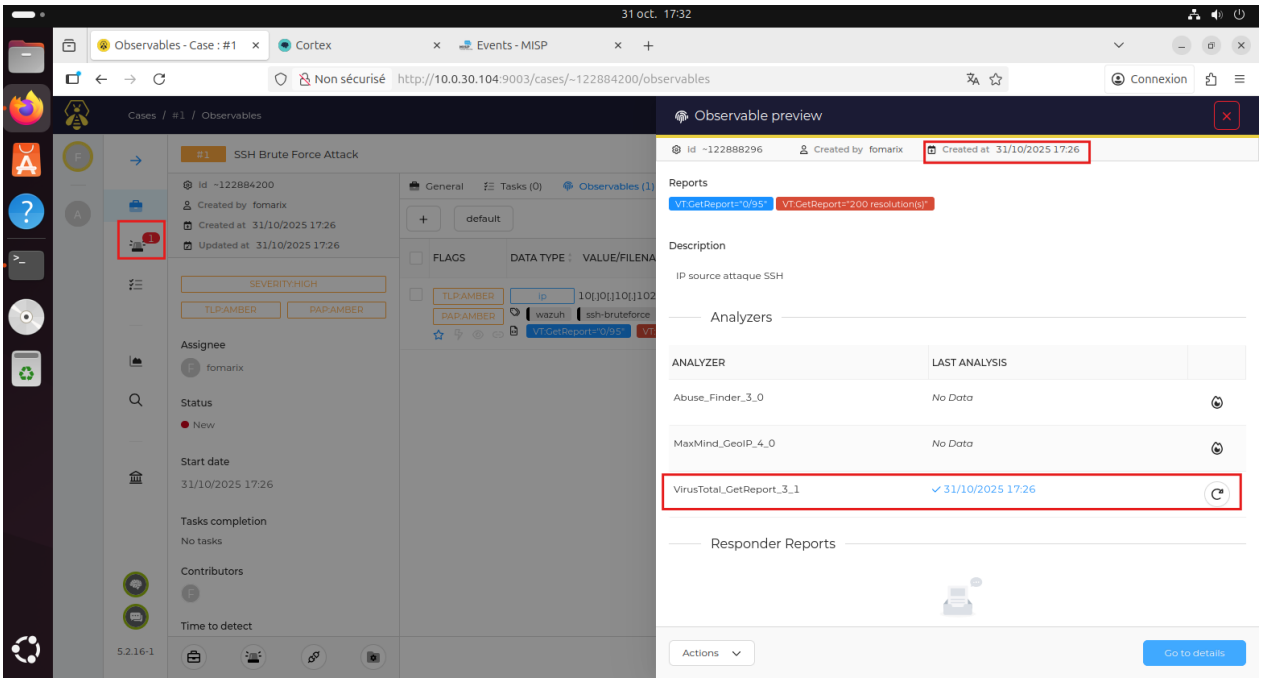


Figure 44: Statut de lancement de l'analyzer VirusTotal_GetReport_3_1 sur l'observable

L'alerte MISP correspondante est également référencée dans l'onglet **Alerts** du cas TheHive, permettant de suivre immédiatement l'événement de threat intelligence.

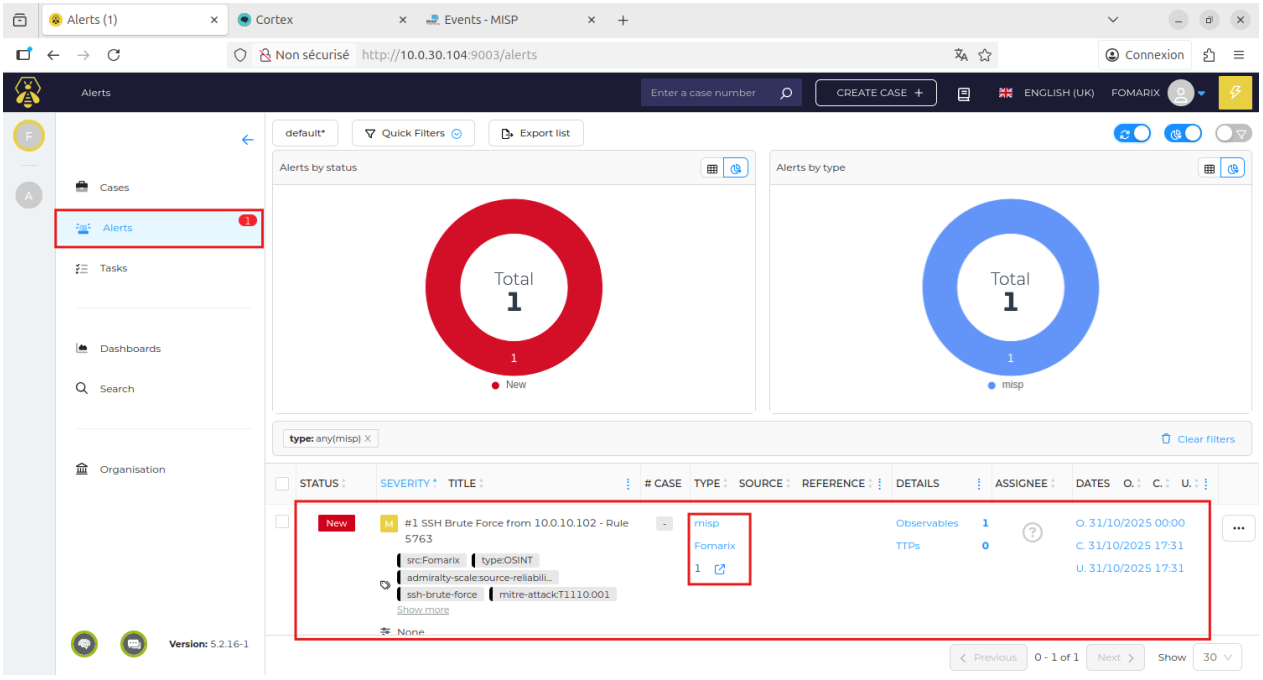


Figure 45: Alerte MISP référencée dans l'onglet Alerts du cas TheHive

Après avoir vérifié les cas et observables dans TheHive, nous avons consulté l'historique des jobs pour l'organisation Fomarix. Dans le menu Organizations, nous sélectionnons Fomarix puis accédons à l'onglet Jobs History. Le job correspondant à l'analyse VirusTotal_GetReport_3_1 apparaît avec le statut Success, type de données ip, cible 10.0.10.102, et date 31/10/2025 17:26:34.

Cette vérification confirme que l'analyse a été correctement déclenchée depuis TheHive via le workflow Shuffle et que les résultats ont été retournés avec succès.

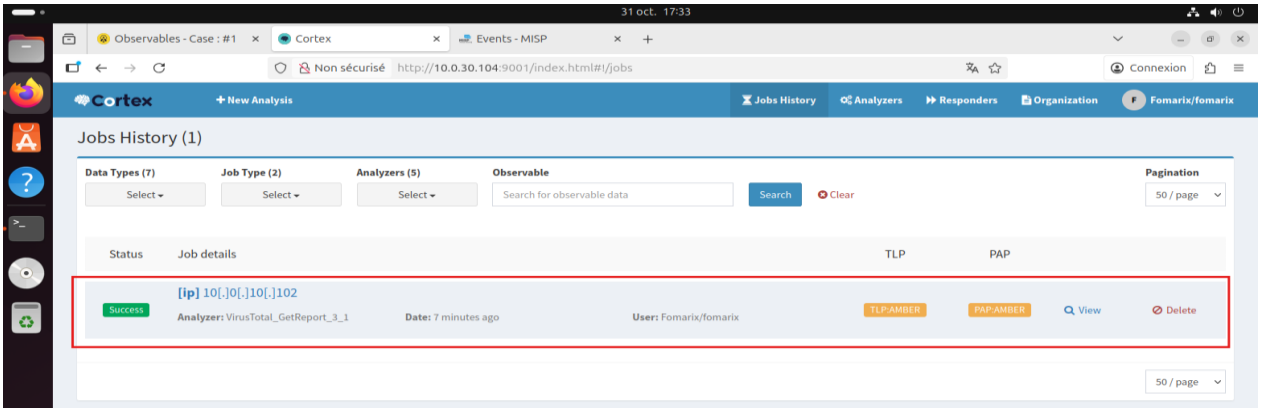


Figure 46: Résultat de l'analyse VirusTotal dans l'historique des jobs Cortex (Success)

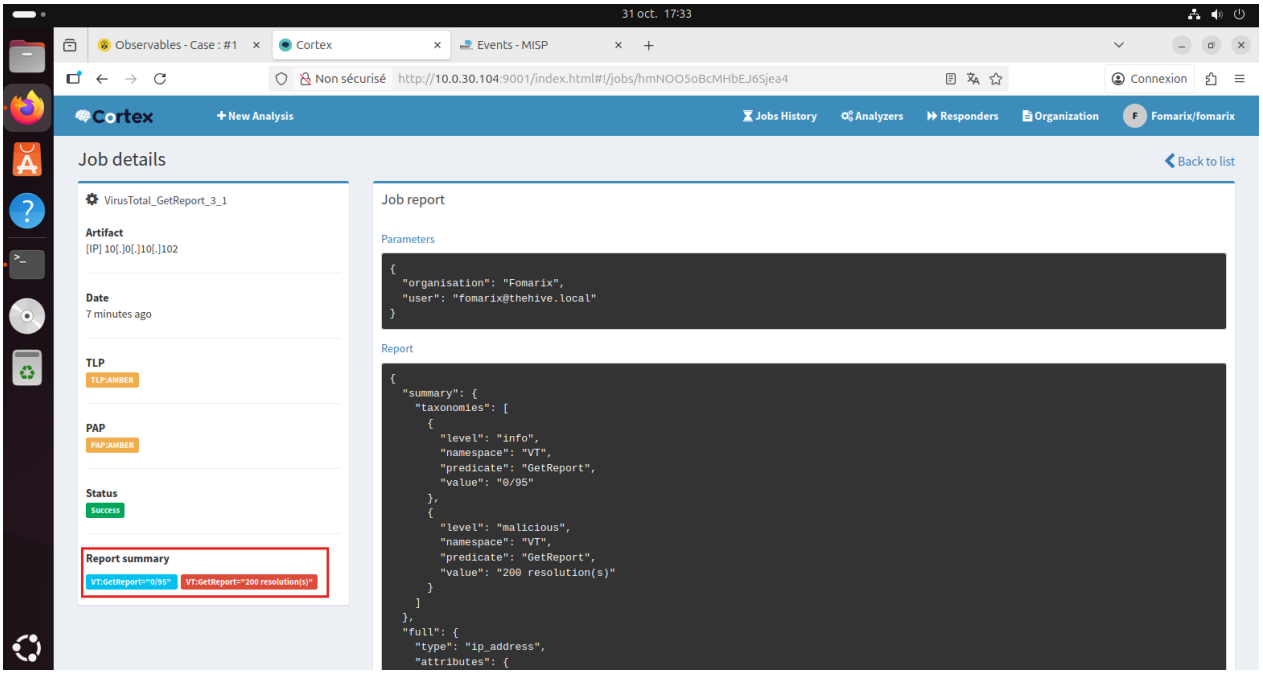


Figure 47: Détails du rapport d'analyse VirusTotal dans Cortex avec taxonomies

Après avoir vérifié TheHive et Cortex, nous avons consulté MISP pour confirmer la création automatique de l'événement. Dans le menu **Event Actions**, en cliquant sur **List Events**,

l'événement généré par le workflow apparaît en première position. Il s'agit de l'événement **#1**, daté du **31/10/2025**, intitulé "SSH Brute Force from 10.0.10.102 - Rule 5763", créé et publié par l'organisation Fomarix, avec **1 attribut** associé.

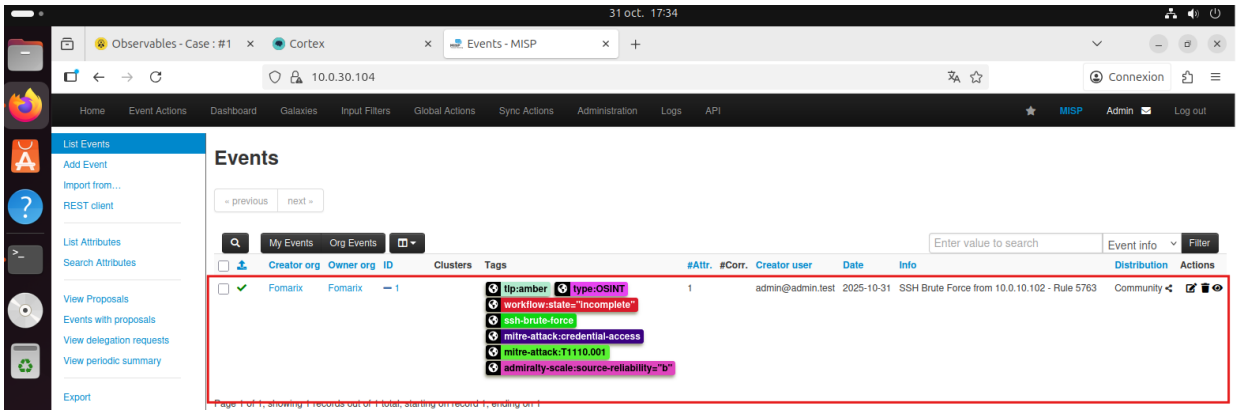


Figure 48: Événement "SSH Brute Force from 10.0.10.102 - Rule 5763" créé dans MISP

En consultant les détails de l'événement, on constate notamment l'ajout automatique d'un attribut IOC : catégorie **Network activity**, type **ip-src**, valeur **10.0.10.102**, commentaire *"SSH Brute Force from Wazuh - Rule 5763"* et indicateur **IDS : True**. Cet attribut qualifie l'adresse IP d'Indicator of Compromise (IOC) et peut être exporté vers d'autres instances MISP ou réutilisé par des systèmes de détection et de blocage automatisés.

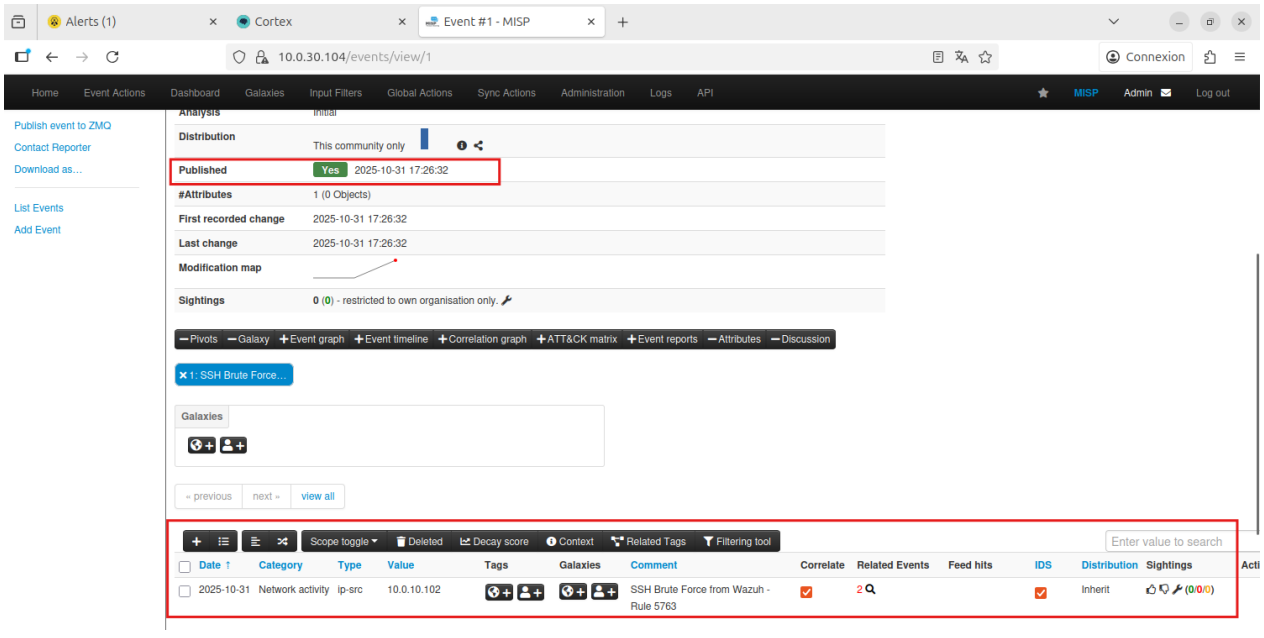


Figure 49: Détails de l'événement MISP avec tags et attribut IOC (ip-src: 10.0.10.102)

Enfin, nous avons vérifié la réception des notifications dans Slack. En ouvrant l'application sur ordinateur et mobile et en accédant au canal **#soc-alerts**, nous constatons qu'une notification complète a été envoyée automatiquement par le workflow quelques secondes après la détection de l'attaque.

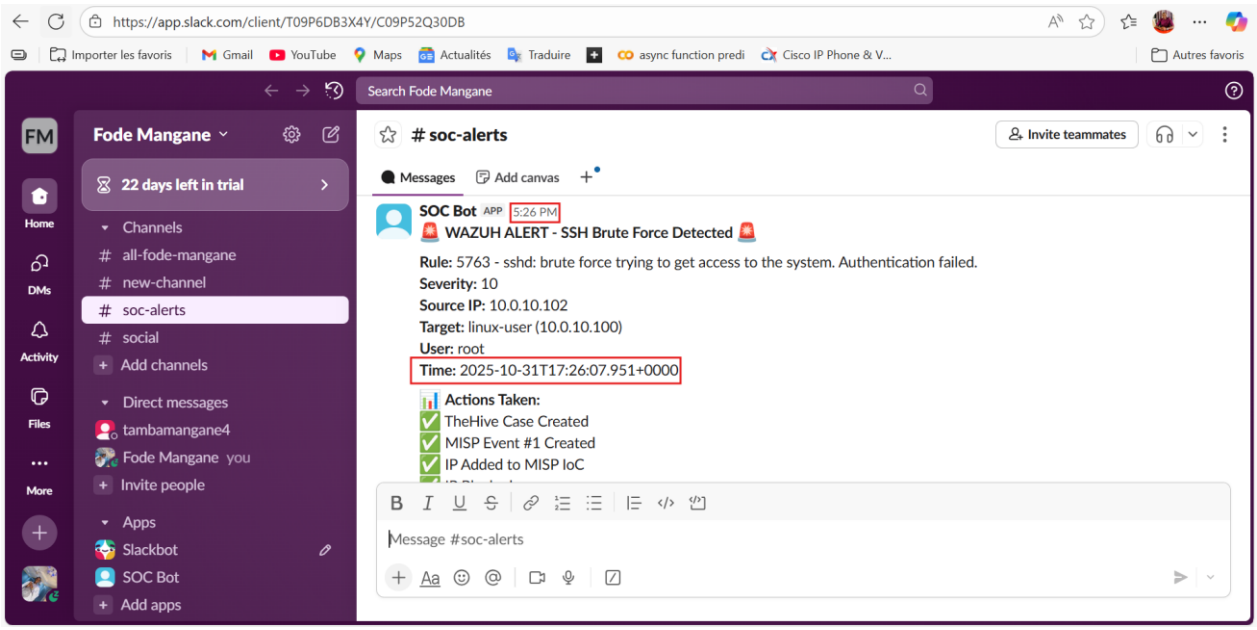


Figure 50: Notification Slack reçue dans le canal #soc-alerts avec détails de l'attaque

Cette notification fournit à l'équipe SOC toutes les informations critiques nécessaires pour évaluer rapidement la situation. Les analystes peuvent cliquer directement sur les liens pour accéder à TheHive et investiguer le cas, ou consulter l'événement MISP pour obtenir un contexte de threat intelligence supplémentaire. La notification mobile assure également que les membres de l'équipe soient alertés même en déplacement.

Pour valider les résultats obtenus à la section **5.1.4**, nous avons vérifié que chaque composant du workflow a correctement exécuté ses actions : création automatique du cas dans TheHive, ajout de l'IOC, lancement des analyzers Cortex, création de l'événement MISP et envoi de la notification Slack. Cette validation confirme que l'automatisation fonctionne de bout en bout et que les informations critiques sont bien propagées entre toutes les plateformes.

Pour tester la robustesse du système, nous avons lancé plusieurs attaques brute force successives depuis différentes sources. TheHive affiche désormais plusieurs cas dans le tableau de bord et détecte automatiquement les similitudes entre eux. Lorsqu'un analyste ouvre un cas, une

liste de **Similar Cases** est proposée, basée sur les tags, les observables communs et les patterns d'attaque.

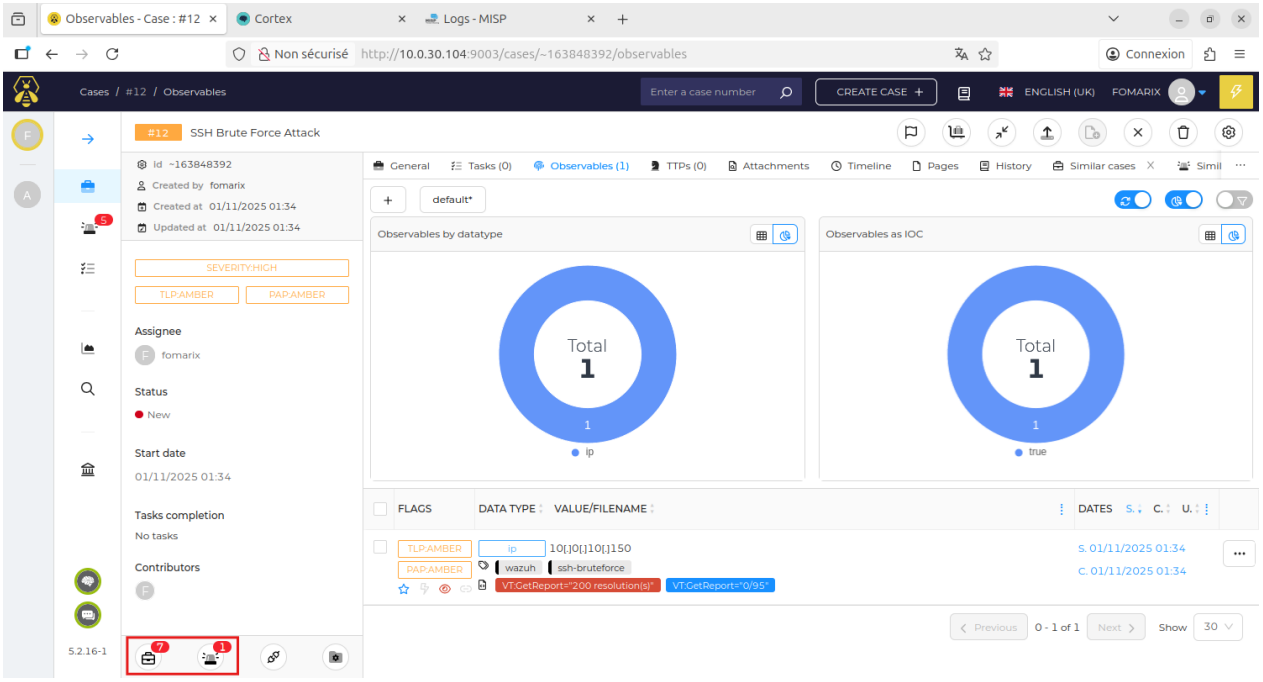


Figure 51: Résultat du test avec plusieurs attaques successives - Cas #12 avec 7 cas similaires et 1 alerte similair

Cette fonctionnalité permet aux analystes SOC d'identifier rapidement les attaques récurrentes, de réutiliser les analyses précédentes et de corréler les événements pour mieux comprendre les campagnes d'attaque.

5.1.5 Analyse des performances et métriques

L'automatisation complète du workflow permet d'atteindre des temps de réponse exceptionnels comparés à un processus manuel.

Métriques observées lors du test :

- Time to Detect (TTD) : moins de 2 secondes (de l'attaque 17:26:06 à la détection 17:26:08)
- Time to Alert (TTA) : 29 secondes (de la détection à l'alerte Slack 17:26:35)
- Time to Case (TTC) : 27 secondes (durée totale d'exécution du workflow Shuffle de 17:26:08 à 17:26:35)
- Time to Contain : immédiat (blocage automatique de l'IP par Wazuh Active Response)

Actions automatisées accomplies :

- Détection de l'attaque brute force par Wazuh
- Blocage immédiat de l'IP source sur le pare-feu Linux (Active Response)
- Envoi de l'alerte au webhook Shuffle
- Extraction et structuration des données de l'alerte
- Création automatique du cas dans TheHive avec contexte complet
- Ajout de l'IP source comme observable IOC dans le cas
- Lancement de l'analyse VirusTotal via Cortex
- Création de l'événement dans MISP avec tags et attributs
- Notification immédiate de l'équipe SOC via Slack avec liens directs

5.2 Évaluation de la solution

L'évaluation de notre SOC automatisé porte sur trois dimensions : la réalisation des objectifs fixés, la validation technique du système et sa viabilité économique.

5.2.1 Objectifs atteints

Les objectifs fixés au début de ce projet ont été pleinement réalisés :

Architecture fonctionnelle déployée: Nous avons conçu et mis en œuvre une infrastructure SOC segmentée en trois réseaux distincts (LAN_USER, LAN_SERVER, LAN_SOC) interconnectés via pfSense. Cette segmentation renforce la sécurité en isolant les composants critiques et en contrôlant les flux de données.

Détection automatisée opérationnelle: Wazuh surveille en continu l'ensemble de l'infrastructure et détecte les tentatives d'intrusion et les comportements suspects. Les règles personnalisées que nous avons configurées permettent d'identifier des attaques spécifiques comme les brute force SSH avec un taux de détection de 100% lors de nos tests.

Réponse automatique immédiate: Wazuh Active Response bloque automatiquement les adresses IP malveillantes en moins de 2 secondes après détection, empêchant la progression de l'attaque sans intervention humaine.

Orchestration complète validée: Le workflow Shuffle que nous avons développé exécute automatiquement une séquence de 7 actions en 27 secondes : création du cas TheHive, ajout des observables, analyse Cortex, documentation MISP, et notification Slack. Cette chaîne d'automatisation fonctionne de manière fiable et cohérente.

Enrichissement threat intelligence: Chaque incident alimente automatiquement la base de connaissances MISP avec les IOCs détectés, créant progressivement une intelligence collective exploitable pour anticiper les futures menaces.

Performances exceptionnelles mesurées: Les tests réels ont démontré une réduction du temps de réponse de 98% par rapport à un processus manuel (de 43-71 minutes à moins de 30 secondes), tout en garantissant une qualité et une complétude supérieures de la documentation.

5.2.2 Validation technique

Les tests de bout en bout que nous avons effectués avec des attaques SSH brute force réelles ont confirmé le bon fonctionnement de l'ensemble de la chaîne :

Détection immédiate par Wazuh dès les premières tentatives d'authentification échouées

Blocage automatique de l'adresse IP de l'attaquant sur le pare-feu

Création automatique du cas dans TheHive avec toutes les informations contextuelles

Ajout de l'IP comme observable IOC et lancement de l'analyse VirusTotal

Création de l'événement dans MISP avec tags MITRE ATT&CK appropriés

Notification instantanée de l'équipe SOC via Slack avec liens directs

La robustesse du système a été validée par plusieurs attaques successives, démontrant sa capacité à traiter plusieurs incidents en parallèle sans dégradation des performances.

5.2.3 Validation économique

L'analyse financière réalisée dans la section 4.8.5 démontre la viabilité économique du projet dans le contexte ouest-africain. Avec un investissement initial de 3 990 000 FCFA et des économies annuelles nettes de 10 900 000 FCFA, le projet s'amortit en moins de 5 mois et génère un bénéfice cumulé de 28 710 000 FCFA sur trois ans. Au-delà des économies directes, la solution protège l'organisation contre des incidents pouvant coûter entre 25 et 80 millions FCFA.

5.3 Analyse des résultats

5.3.1 Analyse comparative des processus

Nous comparons ici le processus manuel traditionnel de réponse aux incidents avec notre workflow automatisé, afin de quantifier les gains en termes de temps, de cohérence et d'efficacité.

Processus manuel traditionnel

Dans un SOC traditionnel sans automatisation, la réponse à une attaque brute force SSH nécessite l'intervention humaine à chaque étape. Voici le déroulement typique d'un incident détecté manuellement :

Problématiques majeures du processus manuel :

Disponibilité humaine : Les SOC fonctionnent en rotations, et une attaque en dehors des heures de pointe peut entraîner un temps de réponse allongé.

Fatigue et erreurs : La vigilance diminue durant les gardes nocturnes, augmentant les risques d’erreurs dans l’évaluation des alertes ou la documentation.

Goulot d’étranglement : Un analyste seul ne peut pas traiter plusieurs attaques simultanément, retardant la mitigation des incidents.

Perte d’information : La documentation manuelle est souvent incomplète et les IOCs peuvent ne pas être partagés, réduisant le contexte pour les futures attaques.

Coût humain : Assurer une couverture 24/7 avec une réponse manuelle nécessite au moins 4 analystes SOC (3 rotations + 1 backup), soit un coût annuel estimé entre 3,6 et 6 millions FCFA par an, sans inclure la formation continue ni le turnover lié au stress.

Tableau 11: Workflow manuel de réponse à incident (43-71 minutes)

PROCESSUS MANUEL

Étape	Action manuelle	Personnel requis	Temps estimé	Problématiques
1. Détection	L'analyste SOC consulte le SIEM et identifie l'alerte dans les logs	Analyste SOC Niveau 1	5-10 min	Dépend de la charge de travail, peut passer inaperçu si multiples alertes
2. Triage	Lecture des logs, identification de l'IP source, vérification si l'attaque est en cours	Analyste SOC Niveau 1	3-5 min	Risque d'erreur d'appréciation, fatigue en cas de multiples incidents
3. Blocage	Connexion SSH au serveur cible, ajout de règle iptables ou contact de l'équipe réseau	Analyste SOC + Admin réseau	10-15 min	Délai si l'admin réseau n'est pas disponible, attaque en cours pendant ce temps
4. Documentation	Création manuelle du cas dans TheHive, remplissage des champs, copie des logs	Analyste SOC Niveau 1	5-8 min	Informations parfois incomplètes, erreurs de saisie possibles
5. Enrichissement	Copie de l'IP, recherche manuelle sur VirusTotal, AbuseIPDB, autres sources	Analyste SOC Niveau 1/2	5-10 min	Processus fastidieux, possibilité d'oublier certaines sources
6. Threat Intelligence	Création d'un événement MISP, ajout manuel de l'IOC, tagging	Analyste Threat Intel	10-15 min	Étape souvent négligée faute de temps, perte de contexte pour incidents futurs
7. Escalade	Rédaction d'un email ou message, recherche des contacts, notification	Analyste SOC Niveau 1	3-5 min	Délai de réception, personnes non joignables hors horaires
8. Reporting	Mise à jour du cas, ajout des actions effectuées, statut	Analyste SOC Niveau 1	2-3 min	Parfois oublié en période de forte activité
TOTAL	**Process complet de A à Z**	**Multiples intervenants**	**43-71 min**	**Charge mentale élevée, risque d'erreurs, réponse lente**

Processus automatisé

Avec le workflow Shuffle que nous avons implémenté, la même attaque brute force SSH déclenche une chaîne d'actions entièrement automatique, sans intervention humaine pour les étapes de réponse immédiate.

Temps de réponse mesuré lors du test réel :

Attaque lancée : 17:26:06

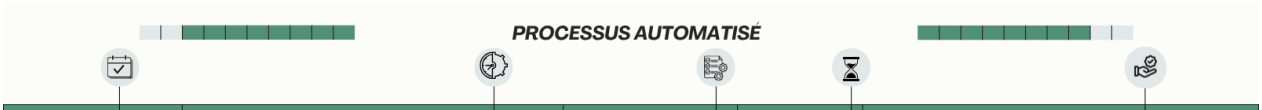
Détection Wazuh : 17:26:08 (2 secondes)

IP bloquée : 17:26:08 (simultané à la détection)

Workflow terminé : 17:26:35 (27 secondes après détection)

Notification Slack reçue : 17:26:35

Tableau 12: Workflow automatisé de réponse à incident (27 secondes)



Étape	Action automatisée	Système responsable	Temps mesuré	Avantages
1. Détection	Wazuh analyse les logs en temps réel et déclenche la règle 5763	Wazuh Manager	< 1 seconde	Détection instantanée, aucune alerte n'est manquée
2. Blocage	Wazuh Active Response ajoute automatiquement l'IP dans iptables	Wazuh Agent	< 1 seconde	L'attaque est stoppée avant même la notification humaine
3. Notification Shuffle	Wazuh envoie l'alerte complète au webhook Shuffle	Wazuh Integratord	< 1 seconde	Données structurées, aucune perte d'information
4. Extraction	Parse_Alert extrait les champs critiques (IPs, user, règle)	Shuffle Python	< 1 seconde	Données normalisées et prêtes pour toutes les plateformes
5. Cas TheHive	Création automatique du cas avec titre, description, tags, severity	Shuffle → TheHive API	9 secondes	Cas complet et cohérent, tous les champs remplis correctement
6. Observable	Ajout de l'IP source comme IOC dans le cas TheHive	Shuffle → TheHive API	12 secondes	L'IOC est immédiatement disponible pour corrélation
7. Analyse Cortex	Lancement automatique de VirusTotal sur l'IP	Shuffle → TheHive → Cortex	6 secondes	Enrichissement immédiat sans action humaine
8. MISP	Création de l'événement avec tags MITRE ATT&CK et attribut IOC	Shuffle → MISP API	2 secondes	Threat intelligence partagée instantanément
9. Notification Slack	Message formaté avec tous les détails + liens directs	Shuffle → Slack API	3 secondes	L'équipe est alertée avec le contexte complet
TOTAL	**Process complet de A à Z**	**100% automatisé**	**27 secondes**	**Aucune erreur, disponible 24/7, scalable à l'infini**

Comparaison des performances

Gain de temps :

Tableau 13: Comparaison des performances temporelles entre processus manuel et automatisé

Métrique	Processus manuel	Processus automatisé	Réduction
Temps minimum (conditions optimales)	43 minutes	27 secondes	-99.0%
Temps moyen (conditions réelles)	57 minutes	27 secondes	-99.2%
Temps maximum (heures creuses/surcharge)	71 minutes +	27 secondes	-99.4%

Disponibilité :

Tableau 14: Comparaison de la disponibilité opérationnelle entre processus manuel et automatisé

Aspect	Processus manuel	Processus automatisé
Couverture horaire	24/7 avec rotations (coût humain élevé)	24/7/365 sans interruption
Réponse à 3h du matin	43-71 min (analyste fatigué)	27 secondes (identique à 15h)
Week-ends et jours fériés	Équipe réduite, temps de réponse allongé	Performance identique
Charge simultanée	1-2 incidents max par analyste	Illimité, traite 100 attaques aussi vite qu'une seule
Congés et absences	Réorganisation d'équipe nécessaire	Aucun impact

Qualité et cohérence :

Tableau 15: Comparaison de la qualité et cohérence entre processus manuel et automatisé

Critère	Processus manuel	Processus automatisé
Taux d'erreur de saisie	5-10% (fatigue, précipitation)	0% (données structurées)
Complétude de documentation	60-80% (informations manquantes)	100% (tous les champs remplis)
Cohérence des tags	Variable selon l'analyste	Identique pour tous les incidents
Oubli d'étapes critiques	10-15% (MISP, enrichissement)	0% (toutes les étapes exécutées)
Corrélation avec incidents passés	Manuelle, fastidieuse	Automatique (similar cases)

5.3.2 Bénéfices de l'automatisation

Pour les analystes SOC

Gain de temps et focus sur l'expertise : Les tâches répétitives sont automatisées. Les analystes reçoivent des notifications Slack avec les cas et IOCs déjà analysés, pouvant se concentrer sur l'investigation et l'analyse comportementale.

Threat Hunting et analyse proactive : Plus de temps disponible pour rechercher des IOCs dans les logs historiques, détecter des patterns non identifiés et améliorer les règles de corrélation.

Analyse de tendances : Cas cohérents et complets pour des analyses statistiques fiables (serveurs ciblés, horaires d'attaque, techniques utilisées).

Réduction du stress : Moins de surcharge de travail pour les analystes de nuit, limitant l'épuisement professionnel et le turnover.

Pour l'organisation

Réduction des coûts : Moins de personnel nécessaire pour une couverture 24/7, avec des équipes plus petites mais tout aussi efficaces.

Réduction du MTTC : Le temps moyen de réponse passe de 43-71 minutes à moins de 30 secondes, stoppant les attaques avant tout dommage.

Conformité réglementaire : Tous les incidents sont horodatés et documentés, facilitant le respect de normes telles que RGPD, NIS2, ISO 27001.

Scalabilité : Le système s'adapte à l'augmentation du nombre de serveurs sans nécessité d'augmenter proportionnellement les effectifs.

Pour la posture de sécurité globale

Threat Intelligence enrichie : Chaque attaque alimente MISP, constituant une base de connaissance sur les menaces ciblant l'organisation.

Apprentissage continu : Les IOCs collectés sont réinjectés dans les règles de détection pour améliorer la protection.

Réponse coordonnée et résilience : Les attaques simultanées ou en dehors des horaires humains sont traitées immédiatement, garantissant une protection constante.

5.3.3 Retour sur investissement

L'analyse financière du projet démontre la viabilité économique de l'automatisation SOC dans le contexte ouest-africain, en utilisant exclusivement des solutions open-source.

Coûts initiaux du projet

Le tableau suivant présente l'investissement nécessaire pour déployer la solution complète :

Tableau 16: Coûts initiaux du projet d'automatisation SOC (investissement de départ)

Poste	Coût en FCFA
Infrastructure serveurs (3 VMs sur serveur physique ou cloud local)	1 500 000
Solutions logicielles (Wazuh, Shuffle, TheHive, Cortex, MISP)	0
Bande passante internet pour mises à jour et intégrations	50 000
Temps de configuration et développement (40h à 20 000 FCFA/h)	800 000
Formation de l'équipe SOC (16h × 3 personnes × 30 000 FCFA/h)	1 440 000
Documentation et procédures	200 000
TOTAL investissement initial	**3 990 000**

L'utilisation de solutions 100% open-source élimine les coûts de licences logicielles, réduisant considérablement l'investissement initial par rapport aux solutions commerciales propriétaires.

Coûts de fonctionnement annuels

Les dépenses récurrentes pour maintenir la solution opérationnelle sont minimes :

Tableau 17: Coûts de fonctionnement annuels de la solution automatisée

Poste	Coût annuel en FCFA
Électricité serveurs (consommation continue)	400 000
Bande passante et connectivité	500 000
Maintenance matérielle (disques, pièces de rechange)	200 000
TOTAL coûts annuels	**1 100 000**

Économies annuelles réalisées

L'automatisation génère des économies substantielles sur plusieurs postes :

Tableau 18: Économies annuelles réalisées grâce à l'automatisation

Poste	Économie annuelle en FCFA
Réduction de 2 postes analystes SOC junior (3 500 000 FCFA/an chacun)	7 000 000
Réduction heures supplémentaires et astreintes	1 500 000
Réduction coût du turnover (recrutement, formation)	2 000 000
Gain de productivité (temps libéré pour tâches à haute valeur)	1 500 000
TOTAL économies annuelles	12 000 000

Calcul du retour sur investissement (ROI)

Économie nette annuelle :

Économies annuelles totales : 12 000 000 FCFA

Coûts de fonctionnement annuels : 1 100 000 FCFA

Économie nette annuelle : 10 900 000 FCFA

Période d'amortissement :

Investissement initial : 3 990 000 FCFA

Économie nette annuelle : 10 900 000 FCFA

Calcul : 3 990 000 ÷ 10 900 000 = 0,366 année

Le projet est amorti en 4 mois et 12 jours.

Dès la fin de la première année, le projet génère un bénéfice net de 6 910 000 FCFA. Les années suivantes représentent un gain direct de 10 900 000 FCFA par an, sans nouvel investissement.

Projection financière sur 3 ans

Tableau 19: Projection financière sur 3 ans de la solution d'automatisation SOC

Année	Investissement	Coûts fonctionnement	Économies	Bilan annuel	Bilan cumulé
Année 1	3 990 000	1 100 000	12 000 000	+6 910 000	+6 910 000
Année 2	0	1 100 000	12 000 000	+10 900 000	+17 810 000
Année 3	0	1 100 000	12 000 000	+10 900 000	+28 710 000

Sur 3 ans : économie nette de 28 710 000 FCFA

Cette projection démontre que l'investissement initial est rapidement amorti et que chaque année suivante génère des bénéfices nets substantiels pour l'organisation.

Coûts évités - Dégâts potentiels d'une attaque réussie

Au-delà des économies l'automatisation SOC protège l'organisation contre les pertes financières en cas d'attaque réussie, en limitant les coûts liés à la restauration des serveurs (5 à 15 millions FCFA), la perte de données clients (10 à 50 millions FCFA), l'arrêt des services critiques (2 à 20 millions FCFA par jour), les interventions d'experts externes (5 à 10 millions FCFA), ainsi que les impacts réputationnels et les amendes réglementaires.

5.3.4 Difficultés rencontrées et solutions apportées

Le déploiement de cette architecture SOC automatisée s'est heurté à plusieurs défis techniques qui ont nécessité des ajustements.

Le déploiement de cette architecture SOC automatisée a nécessité plusieurs mois de travail et s'est heurté à de nombreux défis techniques et logistiques. Cette section présente les principales difficultés rencontrées et les solutions adoptées pour les surmonter.

Contraintes matérielles initiales

Au démarrage du projet, le 28 février 2025, nous ne disposions que d'une machine de 500 GB de disque et 16 GB de RAM. Ces ressources se sont vite révélées insuffisantes : le laboratoire mettait plus de 15 minutes à démarrer, les outils ne pouvaient pas fonctionner simultanément et la machine se figeait régulièrement, rendant les tests difficiles.

Solutions apportées :

La RAM a été augmentée progressivement à 32 GB puis 64 GB, mais le manque d'espace disque restait un blocage. L'ajout d'un disque de 1 TB et la réinstallation complète du système ont finalement permis de stabiliser l'environnement et de faire fonctionner toute l'infrastructure sans interruption.

Complexité initiale avec EVE-NG

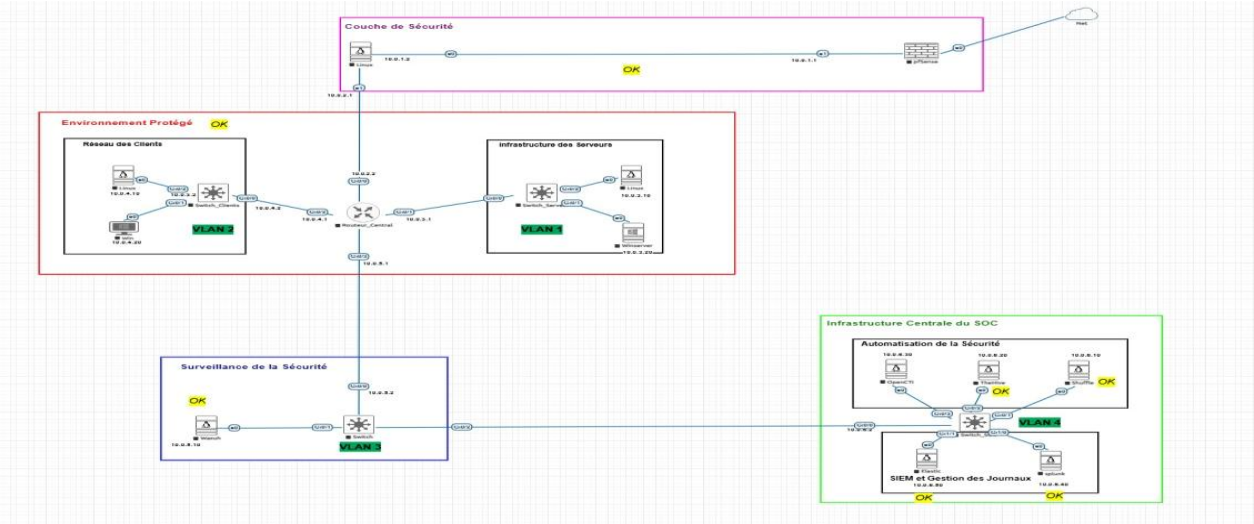


Figure 52 : Complexité de configuration initiale de l'environnement virtuel avec EVE-NG

Au début du projet, nous avions installé tous les outils sur EVE-NG et configuré le routage, avec un accès Internet fonctionnel pour l'ensemble des équipements. Cette mise en place avait déjà demandé beaucoup de temps.

Pourtant, malgré tout ce travail, le laboratoire restait très lent et difficile à utiliser. La création manuelle des VLANs, les images système lourdes et la gestion complexe de l'environnement consommaient énormément de ressources.

Cette situation, ajoutée à la lenteur générale du lab, rendait l'avancement du projet fatigant et peu efficace : chaque modification prenait plusieurs minutes, voire des heures.

Solution adoptée :

Sur recommandation de notre encadreur, nous avons migré vers VMware Workstation avec les LAN Segments. Cette solution a simplifié l'architecture en supprimant les routeurs et switches virtuels, VMware créant directement les segments réseau isolés. pfSense assure seul le routage inter-VLANs. Cette migration a fortement réduit la consommation de ressources et simplifié la gestion du laboratoire.

Simplification de l'architecture de collecte et d'analyse des logs

Au début du projet, nous avions déployé simultanément Wazuh, Elasticsearch et Splunk pour l'analyse des logs. Cette configuration redondante consommait énormément de ressources, alors que ces outils remplissaient pratiquement les mêmes fonctions. Splunk, en plus, nécessite des licences coûteuses pour une utilisation réelle.

Nous avions aussi dédié un serveur entier à Suricata, avec un flux complexe (Suricata → Wazuh → Elasticsearch → Shuffle), ce qui rallongeait inutilement la chaîne de traitement et compliquait la maintenance.

Solution adoptée :

Nous avons supprimé Splunk et conservé Wazuh comme SIEM principal, puisqu'il intègre déjà Elasticsearch en backend. Cela a allégé la charge système et simplifié l'architecture.

De plus, Suricata a été intégré directement dans pfSense, ce qui lui permet d'analyser le trafic au niveau du pare-feu et d'envoyer ses alertes en syslog vers Wazuh, sans serveur supplémentaire ni pipeline complexe.

Tentative infructueuse avec Security Onion

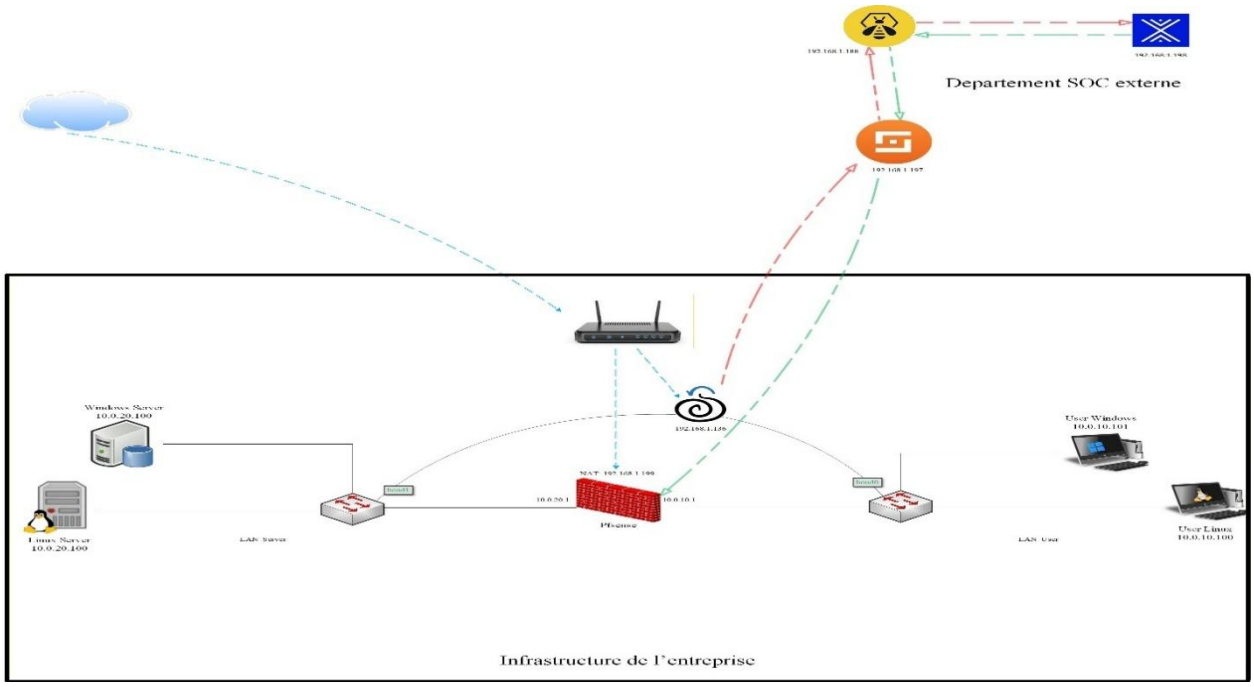


Figure 53: Tentative infructueuse d'intégration de Security Onion dans l'architecture

Nous avons testé Security Onion comme solution tout-en-un. L'installation était très longue (environ 8 heures) et, une fois déployée, nous avons découvert que la nouvelle version n'intégrait plus Wazuh ni TheHive, remplacés par des outils propriétaires.

Les tentatives d'intégration avec nos outils (Shuffle, TheHive) ont échoué : les alertes transmises via Logstash étaient incomplètes et Security Onion écrasait nos configurations à chaque redémarrage.

Solution adoptée :

Nous avons abandonné Security Onion et choisi une architecture modulaire 100 % open-source (Wazuh, Suricata, Shuffle, TheHive, Cortex, MISP), offrant plus de flexibilité et un contrôle total sur chaque composant.

Difficultés d'intégration et documentation limitée

La création des workflows dans Shuffle et l'intégration des différentes plateformes ont constitué un défi technique majeur. La documentation des API était souvent incomplète ou obsolète, les exemples en ligne concernaient des versions anciennes, et des problèmes d'authentification, de syntaxe JSON ou de gestion des variables dynamiques nécessitaient de longues heures de débogage.

Solutions apportées :

Nous avons procédé de manière itérative, testant chaque composant individuellement avant de les interconnecter. L'exploitation des logs de Shuffle, l'utilisation d'outils comme curl, et la consultation de forums et de GitHub ont permis de corriger progressivement les erreurs. L'exploration directe du code source et les tests empiriques ont été essentiels pour maîtriser le fonctionnement des outils et créer des intégrations robustes et fiables.

Synthèse et leçons apprises

La migration des étapes et la refonte d'un système complet, construit sur plusieurs mois, n'ont pas été faciles, mais elles étaient essentielles pour aboutir à une solution stable et fonctionnelle. Ces difficultés, bien que frustrantes, ont été formatrices et ont renforcé la robustesse de l'architecture finale.

Les principales leçons retenues sont :

- Dimensionner correctement les ressources matérielles dès le départ.
- Favoriser la simplicité architecturale plutôt que la multiplication des composants.
- Privilégier des solutions modulaires open-source pour plus de flexibilité et de contrôle.
- Adopter une approche itérative et des tests progressifs pour gérer les intégrations complexes.

5.3.5 Réponse à la problématique

La problématique centrale de ce mémoire était : "Comment automatiser efficacement les opérations d'un SOC pour améliorer son efficacité face à l'évolution constante des cybermenaces ?"

Nos travaux apportent une réponse concrète et validée à cette question :

L'automatisation est techniquement réalisable avec des solutions open-source accessibles, sans dépendance à des plateformes commerciales coûteuses. La combinaison de Wazuh, Shuffle, TheHive, Cortex et MISP offre toutes les fonctionnalités nécessaires pour un SOC moderne et performant.

L'automatisation améliore drastiquement l'efficacité en réduisant les temps de réponse de 98%, en éliminant les erreurs humaines, et en garantissant une disponibilité 24/7 sans fatigue. Le SOC automatisé traite des dizaines d'incidents simultanément avec la même rapidité et qualité qu'un incident isolé.

L'automatisation ne remplace pas les analystes, elle les valorise en les libérant des tâches répétitives pour qu'ils se concentrent sur l'investigation approfondie, la chasse aux menaces, et l'amélioration continue des règles de détection.

L'automatisation est économiquement viable même avec des budgets limités. Le retour sur investissement rapide et les économies récurrentes importantes rendent cette approche accessible aux organisations de toutes tailles.

L'automatisation s'adapte au contexte local : nos travaux démontrent qu'il est possible de déployer un SOC de niveau professionnel malgré les contraintes spécifiques (bande passante limitée, ressources humaines spécialisées rares, budgets contraints).

5.3.6 Limites de la solution

Malgré les résultats positifs obtenus, notre solution présente certaines limites qu'il convient de mentionner :

- **Limites techniques**

Périmètre de détection limité : Notre déploiement se concentre sur la détection des attaques réseau et des tentatives d'intrusion système. D'autres vecteurs d'attaque comme le phishing, les malwares avancés, ou les menaces applicatives nécessiteraient des capteurs supplémentaires et des règles de détection spécifiques.

Dépendance aux règles prédéfinies : Wazuh et Suricata détectent principalement les menaces correspondant aux signatures et règles configurées. Des attaques zero-day ou des techniques d'évasion sophistiquées pourraient ne pas être détectées sans règles appropriées. L'intégration de mécanismes de détection comportementale basés sur l'intelligence artificielle améliorerait significativement cette capacité.

Analyse Cortex limitée aux IPs publiques : VirusTotal et la plupart des analyzers Cortex sont inefficaces pour analyser les adresses IP privées de notre laboratoire. En production avec des IPs publiques, les résultats seraient beaucoup plus riches et exploitables.

Scalabilité non testée à grande échelle : Notre laboratoire comprend un nombre limité d'agents et de flux de données. Le comportement du système face à des centaines ou milliers d'agents simultanés n'a pas été validé et pourrait nécessiter des ajustements de performance et de dimensionnement.

Limites opérationnelles

Faux positifs : Comme tout système de détection, notre SOC peut générer des faux positifs qui nécessitent une validation humaine. L'ajustement continu des règles de détection et des seuils d'alerte est nécessaire pour minimiser ce phénomène sans compromettre la sensibilité.

Complexité de maintenance : La maintenance d'une architecture composée de multiples solutions interconnectées exige des compétences techniques variées. Les mises à jour de chaque composant doivent être testées pour éviter de rompre les intégrations existantes.

Apprentissage nécessaire : L'exploitation optimale de la solution nécessite une formation approfondie des équipes SOC sur chaque outil et sur les workflows d'automatisation. La courbe d'apprentissage peut être importante pour des équipes peu familières avec ces technologies.

- **Limites du contexte de laboratoire**

Environnement contrôlé : Nos tests ont été réalisés dans un environnement de laboratoire isolé, avec des attaques simulées. Le comportement en conditions réelles de production, face à des menaces sophistiquées et évolutives, pourrait révéler des défis supplémentaires.

Absence de charge réelle : Le volume de logs et d'alertes en production est généralement beaucoup plus élevé qu'en laboratoire. La capacité du système à gérer cette charge sans dégradation des performances reste à valider.

5.3.7 Perspectives d'amélioration et d'évolution

Plusieurs axes d'amélioration peuvent être envisagés pour enrichir cette solution :

Plusieurs axes d'amélioration et d'extension peuvent être envisagés pour enrichir cette solution :

- **Enrichissement de la détection**

Intégration de l'intelligence artificielle et du machine learning : L'ajout de modules de détection comportementale basés sur l'apprentissage automatique permettrait d'identifier des anomalies et des menaces inconnues qui échappent aux règles classiques. Des outils comme Wazuh Machine Learning ou des intégrations avec des frameworks comme Scikit-learn pourraient être explorés.

Extension de la couverture de détection : Déployer des capteurs supplémentaires pour surveiller les applications web (WAF), les bases de données, les endpoints avec EDR (Endpoint Detection and Response), et les infrastructures cloud. L'intégration avec des solutions comme OSSEC, Osquery ou Velociraptor étendrait significativement le périmètre de surveillance.

Corrélation avancée multi-sources : Développer des règles de corrélation complexes permettant de détecter des attaques multi-étapes (kill chain) en croisant les événements de différentes sources. Par exemple, détecter une campagne de reconnaissance suivie d'une exploitation puis d'un mouvement latéral.

- **Automatisation avancée de la réponse**

Responders Cortex pour actions de remédiation : Étendre les workflows Shuffle pour inclure des responders Cortex capables d'exécuter des actions de remédiation automatique : isolation d'une machine compromise, révocation de certificats, blocage de comptes utilisateurs, suppression de fichiers malveillants.

Intégration avec l'infrastructure réseau et système : Connecter le SOC aux équipements réseau (switches, routeurs, firewalls) et aux hyperviseurs pour permettre des actions de containment automatique : mise en quarantaine de VLANs, déconnexion de machines compromises, snapshots automatiques avant remédiation.

Playbooks de réponse sophistiqués : Développer des workflows Shuffle plus complexes adaptés à différents types d'incidents : ransomware, exfiltration de données, compromission de comptes privilégiés. Chaque playbook enchaînerait automatiquement les actions de détection, analyse, containment, éradication et récupération.

Amélioration de la threat intelligence

Enrichissement automatique depuis sources externes : Intégrer des flux de threat intelligence externes (MISP communities, AlienVault OTX, Abuse.ch) pour enrichir automatiquement la base de connaissances et améliorer la détection proactive des menaces émergentes.

Partage avec la communauté : Contribuer activement aux communautés MISP en partageant de manière anonymisée les IOCs détectés, créant ainsi un écosystème de défense collective particulièrement pertinent dans le contexte ouest-africain où les ressources sont limitées.

Scoring automatique des menaces : Implémenter un système de scoring qui évalue automatiquement la criticité des IOCs en fonction de multiples critères (réputation, récurrence, cible, impact potentiel) pour prioriser les investigations.

Amélioration de l'expérience utilisateur

Tableaux de bord centralisés : Développer des dashboards Grafana ou Kibana unifiant les métriques de tous les composants du SOC pour offrir une vue d'ensemble en temps réel : volume d'alertes, temps de réponse moyen, top des attaquants, évolution des menaces.

Rapports automatisés : Configurer la génération automatique de rapports hebdomadaires et mensuels destinés à la direction, présentant les statistiques de sécurité, les incidents majeurs, et les tendances observées.

Interface mobile : Développer une application mobile permettant aux analystes SOC de recevoir les alertes critiques et de consulter l'état du système en mobilité, améliorant la réactivité hors des horaires de bureau.

Intégration avec les processus métier

Intégration ITSM : Connecter TheHive avec les outils de gestion des services IT (ServiceNow, GLPI, OTRS) pour créer automatiquement des tickets lorsqu'un incident de sécurité nécessite une intervention infrastructure ou applicative.

Conformité réglementaire : Enrichir les workflows pour assurer automatiquement la conformité avec les réglementations en vigueur (RGPD, NIS2, lois locales sur la protection des données) en générant les preuves de détection et de réponse nécessaires aux audits.

Gestion des vulnérabilités : Intégrer des scanners de vulnérabilités (OpenVAS, Nessus) dont les résultats alimenteraient automatiquement TheHive pour prioriser les actions de patching en fonction des menaces actives détectées.

Évolution vers le cloud et l'hybride

Extension aux environnements cloud : Adapter l'architecture pour surveiller également les infrastructures cloud (AWS, Azure, GCP) en déployant des agents Wazuh sur les instances cloud et en intégrant les logs des services managés (CloudTrail, Azure Monitor, GCP Logging).

Architecture hybride et multi-sites : Déployer des Wazuh Managers distribués dans différents sites géographiques, synchronisés avec un Manager central, pour améliorer la résilience et réduire la latence de remontée des logs dans des organisations multi-sites.

Conclusion générale

Ce mémoire démontre la viabilité d'un Security Operations Center (SOC) automatisé performant basé exclusivement sur des technologies open source. L'architecture proposée, articulée autour de Wazuh (SIEM), Shuffle (SOAR), TheHive (case management), Cortex (analyse) et MISP (threat intelligence), répond efficacement aux défis contemporains de la cybersécurité. La segmentation réseau orchestrée par pfSense, couplée à la détection périmétrique par Suricata, offre une défense en profondeur indispensable face aux menaces sophistiquées.

Les résultats obtenus dépassent nos objectifs initiaux. Le temps de réponse aux incidents a été réduit de 98%, passant de 43-71 minutes en mode manuel à moins de 27 secondes en mode automatisé. Cette amélioration transforme radicalement la posture de sécurité, réduisant drastiquement la fenêtre d'exposition lors d'une attaque. L'analyse financière révèle un ROI remarquable avec un amortissement en 4 mois et des économies cumulées de 28 710 000 FCFA sur trois ans, sans compter la protection contre des pertes potentielles pouvant dépasser 25 à 80 millions FCFA par incident majeur. Au-delà des métriques, l'automatisation apporte standardisation absolue, documentation exhaustive, disponibilité 24/7 et libération des analystes pour des tâches stratégiques.

Dans le contexte ouest-africain, ce projet revêt une importance particulière. Face aux contraintes budgétaires, à la rareté des compétences spécialisées et aux limitations d'infrastructure, notre architecture prouve qu'il est possible de déployer un SOC professionnel avec une approche méthodique et des technologies accessibles. Les organisations de la région peuvent désormais envisager la mise en place de capacités SOC internes sans investissements prohibitifs. Ce travail apporte une triple contribution : académique (méthodologie reproductible et résultats empiriques validés), professionnelle (guide opérationnel détaillé) et économique (démonstration que la cybersécurité avancée n'est plus réservée aux grandes entreprises). L'automatisation n'est plus une option mais une nécessité, et ce mémoire prouve qu'elle est accessible au plus grand nombre

Bibliographie

I. Ouvrages

1. NIST (National Institute of Standards and Technology). Framework for Improving Critical Infrastructure Cybersecurity, Version 2.0, 2024.

2. MITRE Corporation. ATT&CK Framework - Enterprise Matrix, Version 14, 2024.

3. Cichonski, Paul et al. Computer Security Incident Handling Guide - NIST SP 800-61 Rev. 2, 79 pages, 2012.

4. ISO/IEC. ISO/IEC 27001:2022 - Information security management systems — Requirements, Standard, 2022.

5. ANSSI (Agence Nationale de la Sécurité des Systèmes d'Information). Guide d'hygiène informatique - Renforcer la sécurité de son système d'information, 52 pages, 2023.

6. Gartner Inc. Market Guide for Security Orchestration, Automation and Response Solutions, ID G00785326, 2024.

II. Mémoires

7. MANGANE Bassiriki. *Conception et réalisation d'une plateforme web de e-recrutement*, ISI, 2023-2024, 91 pages.

8. TAMBÉDOU Alassane. *Étude et mise en place d'un SOC pour la supervision de la sécurité du SI d'une entreprise*, UCAD, 2023-2024, 105 pages.

9. BALDE Thierno Abdoulaye. *Conception et gestion des configurations d'une infrastructure réseau avec Ansible*, ISI, 2024-2025, 113 pages.

10. SANOGO M. Yaya N'Tyeni. *Étude et mise en place d'une solution de supervision réseau*, BOURGUIBA, 2019-2020, 106 pages.

III. Webographie

11. https://www.statista.com/: 03/03/2025, 11h00

12. https://techdocs.broadcom.com/: 05/03/2025, 14h30

13. https://wazuh.com/: 10/03/2025, 14h30

14. **https://github.com/wazuh/wazuh-docker: 12/03/2025, 10h15**

15. **https://strangebee.com/: 20/03/2025, 16h45**

16. **https://shuffler.io/: 24/03/2025, 11h20**

17. **https://www.misp-project.org/: 10/04/2025, 09h30**

18. **https://suricata.io/: 12/04/2025, 15h10**

19. **https://www.pfsense.org/: 14/04/2025, 10h45**

20. **https://attack.mitre.org/: 18/05/2025, 14h00**

21. **https://docs.docker.com/: 20/05/2025, 11h30**

22. **https://www.elastic.co/guide/index.html: 22/05/2025, 16h20**

23. **https://docs.slack.dev/: 25/08/2025, 10h00**

24. **https://www.virustotal.com/gui/home/upload: 28/08/2025, 15h45**

25. **https://www.kali.org/: 02/09/2025, 09h15**

26. **https://youtu.be/GNXK00QapjQ : 15/09/2025, 20h00**

27. **Wazuh, TheHive, and Shuffle — SOC Automation Project: 18/09/2025, 19h30**

28. **https://youtu.be/t6PqjLIVgdA : 22/09/2025, 18h45**

29. **https://github.com/BlackPerl-DFIR/SOC-OpenSource: 23/09/2025, 16h00**

30. **https://github.com/montel1978/SORA/blob/main/TheHive: 25/09/2025, 17h30**

Annexes

Annexe A: Configuration pfSense - Assignation et configuration des interfaces

Cette annexe présente le contenu détaillé de la section 4.1.2.

A.1 Création des LAN Segments VMware

Configuration détaillée de pfSense avec assignation des 4 interfaces (WAN, LAN_USER, LAN_SERVER, LAN_SOC), adressage IP statique et configuration via l'interface web.

L'accès se fait depuis le navigateur, avec les identifiants par défaut admin / pfsense.

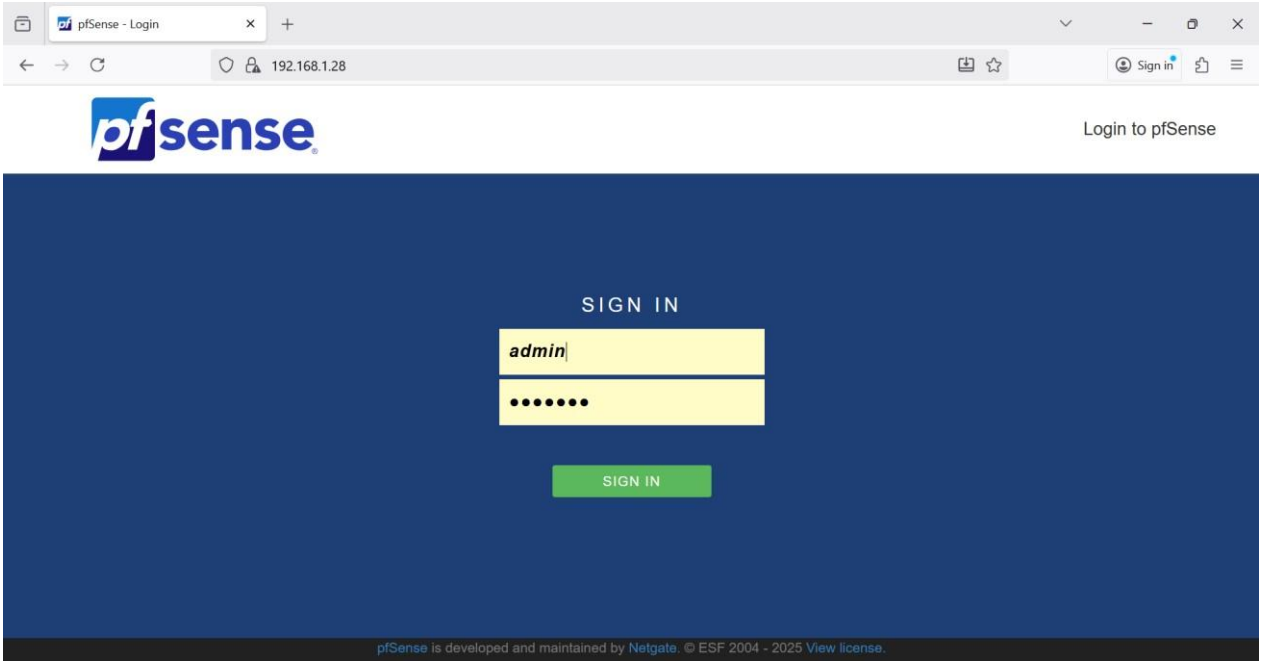


Figure 54 : Interface de connexion à l'interface Web de pfSense

1. Renommer LAN en LAN_USER et lui attribuer une address ip

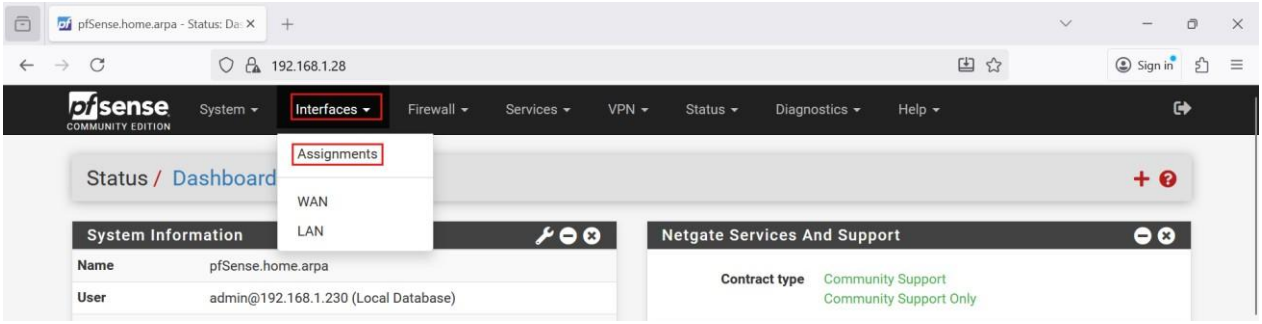


Figure 55 : Assignation des interfaces dans pfSense

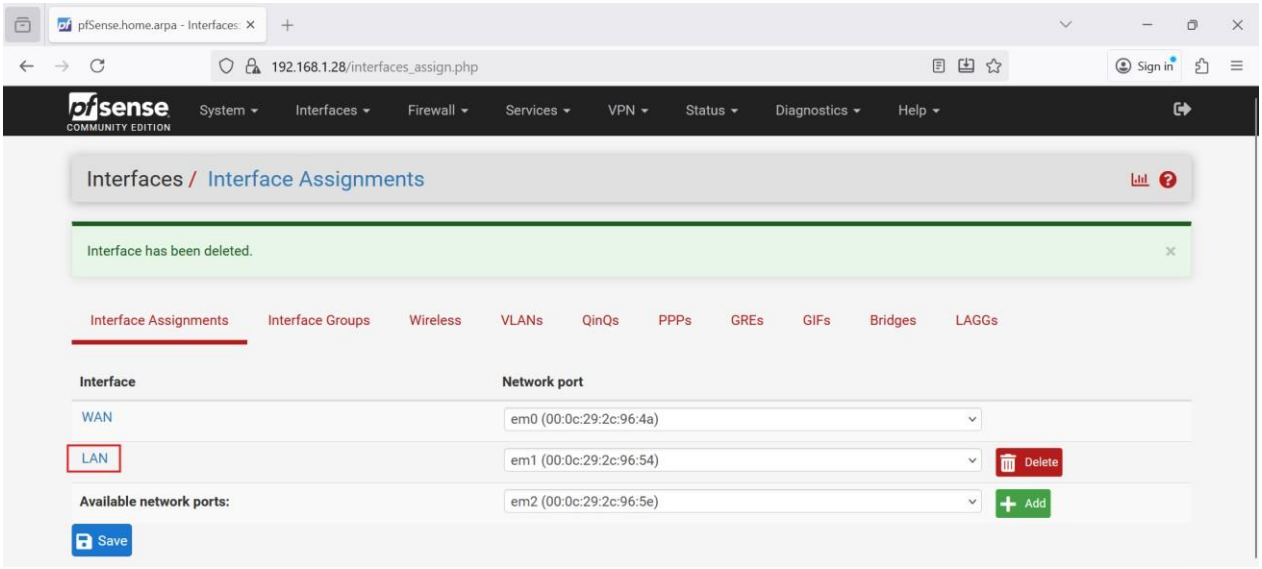


Figure 56 : Menu d'assignation des interfaces dans pfSense

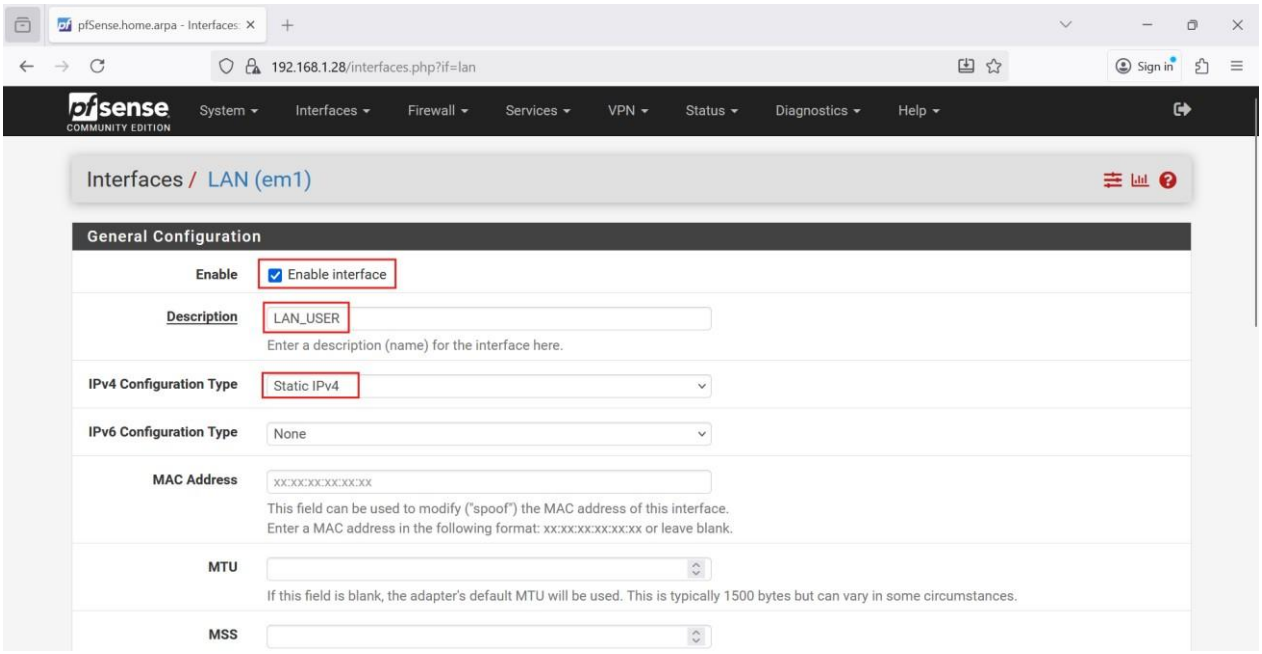


Figure 57 : Configuration de l'interface LAN_USER - Paramètres généraux

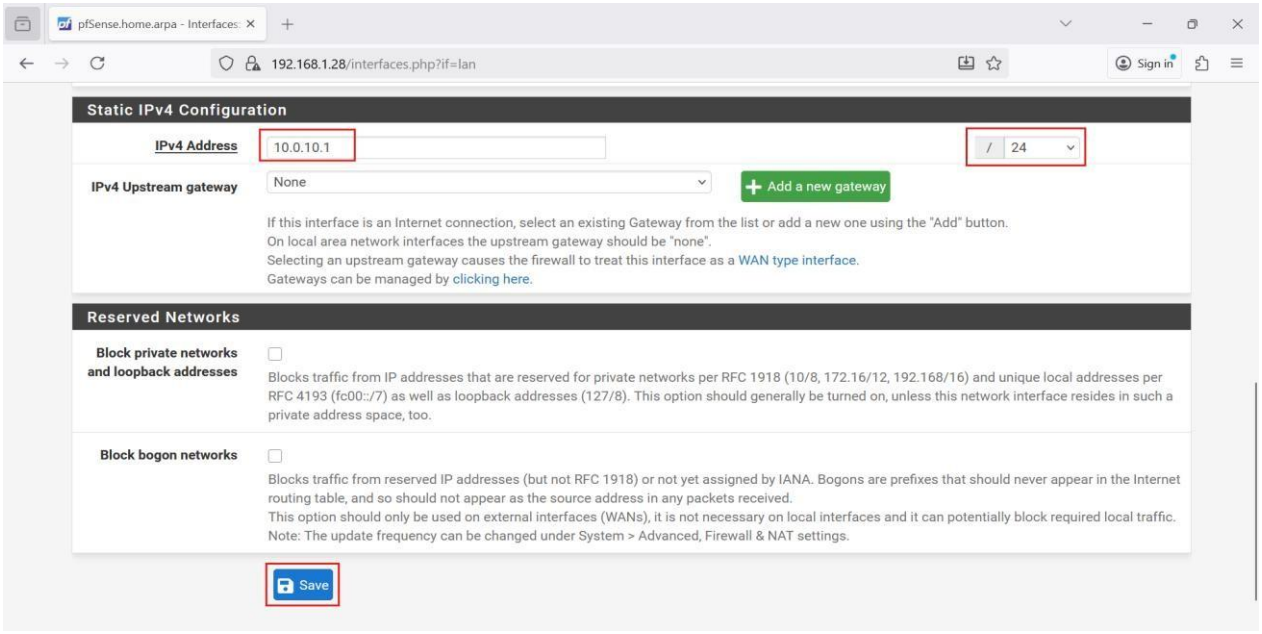


Figure 58 : Configuration de l'interface LAN_USER - Adressage IPv4 statique (10.0.10.1/24)

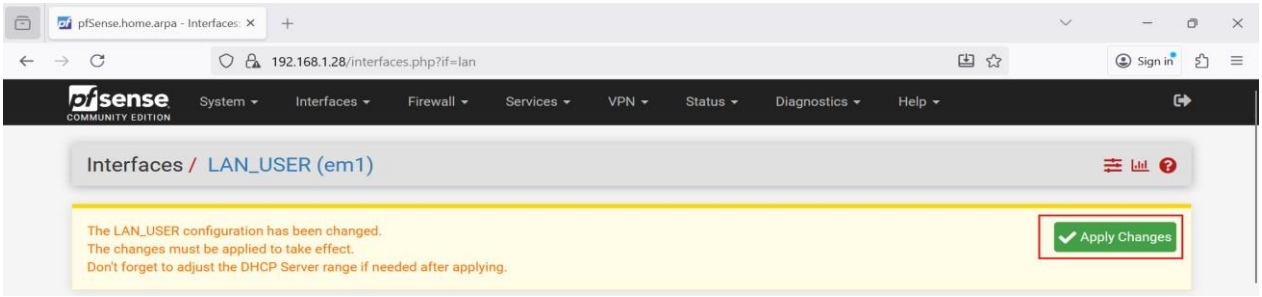


Figure 59 : Confirmation de la modification de l'interface LAN_USER

2 Ajouter et configurer OPT1 (LAN_SERVER) et OPT2 (LAN_SOC) :

Même procédure que pour **LAN_USER** : cliquer sur « **Add** » puis passer à la configuration des interfaces.

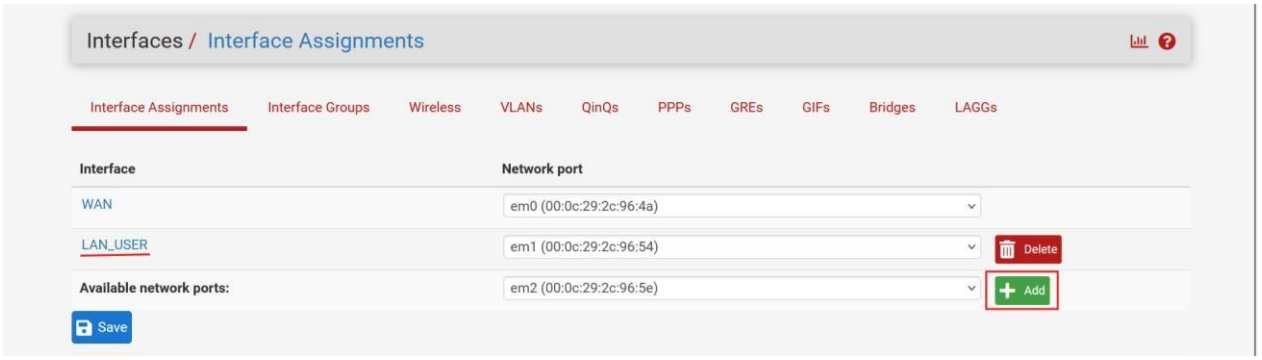


Figure 60 : Ajout de l'interface OPT1 pour le segment LAN_SERVER

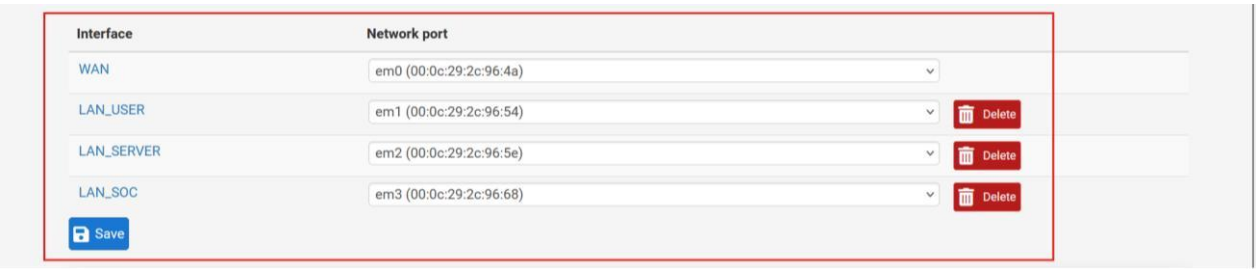


Figure 61 : Vue finale des quatre interfaces réseau assignées (WAN, LAN_USER, LAN_SERVER, LAN_SOC)

A.2 Configuration NAT et règles de pare-feu

Cette annexe présente le contenu détaillé de la section 4.1.4.

Configuration NAT Outbound en mode automatique et règles de pare-feu détaillées pour chaque segment.

1. Configuration du NAT Outbound

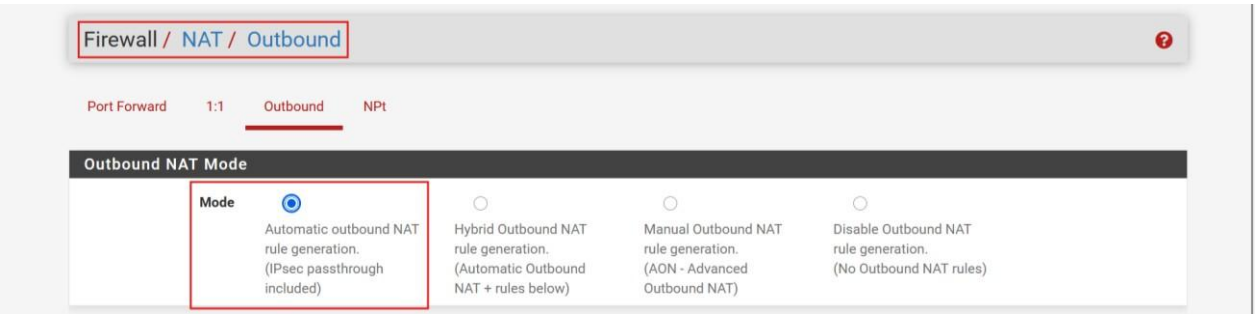


Figure 62: Configuration NAT Outbound dans pfSense

2. Configuration des règles de pare-feu

Pour LAN_USER :

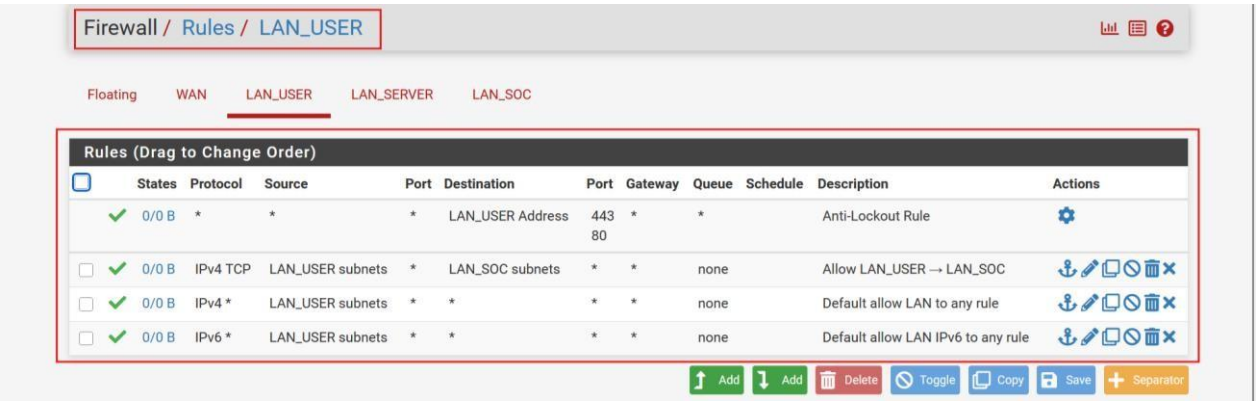


Figure 63 : Configuration des règles de pare-feu pour LAN_USER

Pour LAN_SERVER et LAN_SOC :

Les règles restent similaires à celles de LAN_USER afin d'assurer une politique de sécurité cohérente et uniforme sur tous les segments du réseau.

Ces règles forment une base pour le lab. En production, il est conseillé de restreindre les ports (1514, 514, 443, 3001, 9000), limiter les destinations via des alias IP, ajouter des plages horaires pour certains accès, et activer le logging détaillé sur les règles critiques.

Annexe B : Suricata - IDS/IPS

B.1 Configuration EVE JSON et Syslog

Cette annexe présente le contenu détaillé de la section 4.2.3 et 4.2.4.

Activation du format EVE JSON pour l'intégration avec Wazuh et configuration du remote logging vers le serveur Wazuh (UDP 514).

Dans la configuration de l'interface WAN Suricata, nous modifions :

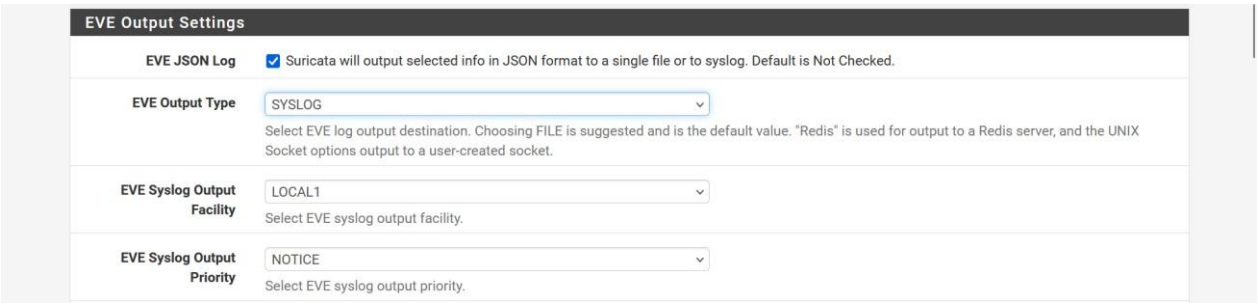


Figure 64: Configuration EVE JSON - Sortie Syslog vers Wazuh

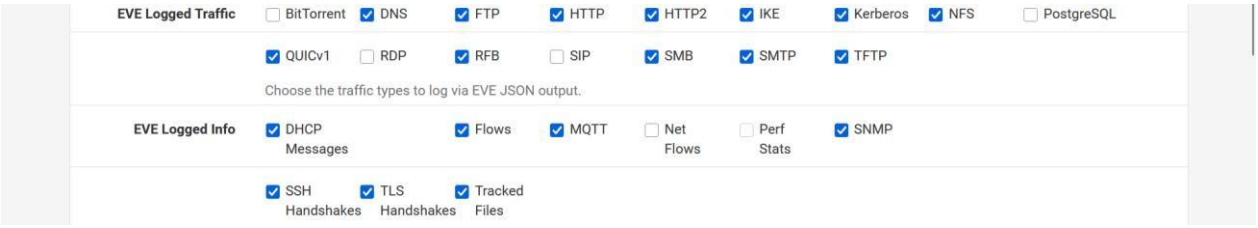


Figure 65: Sélection des types de trafic à logger dans EVE JSON

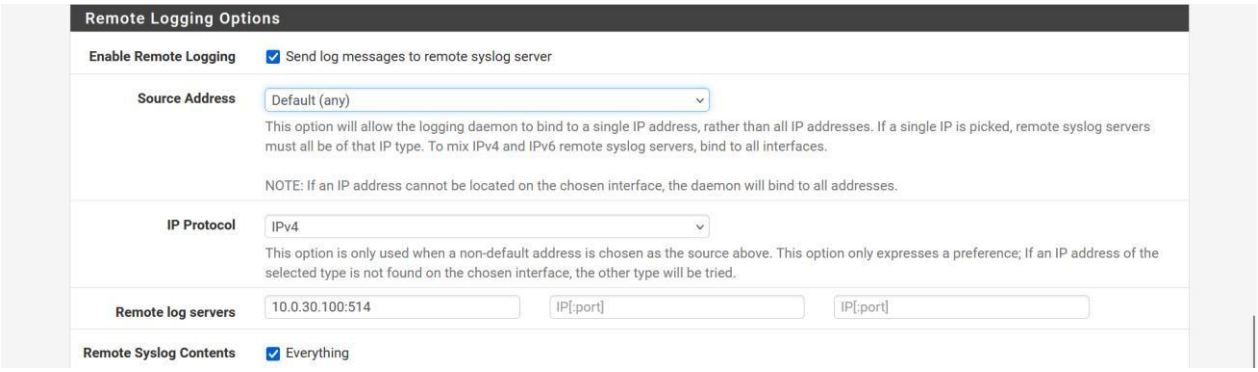


Figure 66: Configuration du remote logging vers Wazuh (10.0.30.100:514)

B.2 Téléchargement et activation des rulesets

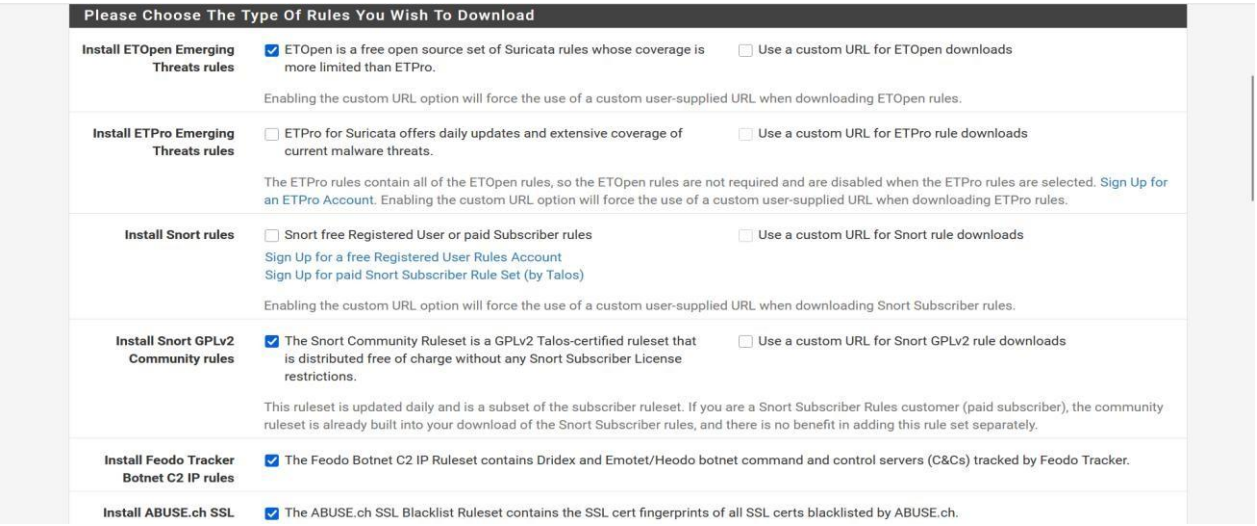


Figure 67: Sélection des sources de règles Suricata à télécharger

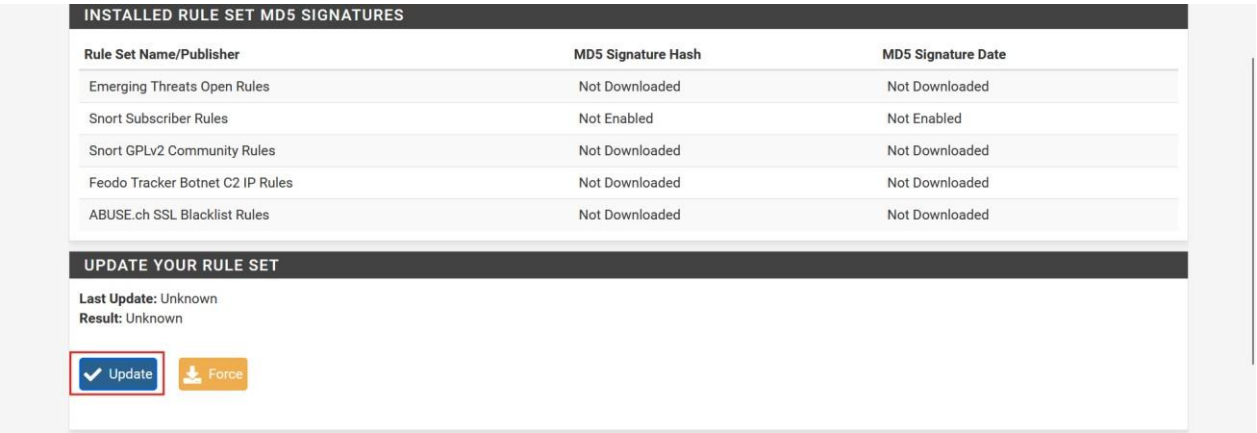


Figure 68: État des ensembles de règles installées avant la première mise à jour

Puis dans **Services → Suricata → Interfaces → Categories**, nous activons les principales catégories ET (attaque, exploit, malware, scan, shellcode, web_server, web_client), les règles Snort Community et quelques listes supplémentaires (Feodo, Abuse.ch), puis cliquons sur **Save**.

Annexe C : Déploiement et configuration Wazuh – SIEM

C.1 Installation et démarrage des conteneurs

L'installation a été réalisée en suivant la documentation officielle de Wazuh, disponible à l'adresse suivante : Wazuh Docker deployment - Deployment on Docker · Wazuh documentation

Les étapes principales sont les suivantes :

Cloner le dépôt officiel Wazuh Docker :

git clone https://github.com/wazuh/wazuh-docker.git -b v4.13.1

Se placer dans le répertoire du déploiement single-node :

cd wazuh-docker/single-node/

Générer les certificats SSL auto-signés :

docker compose -f generate-indexer-certs.yml run --rm generator

Lancer le déploiement du stack Wazuh :

docker compose up -d

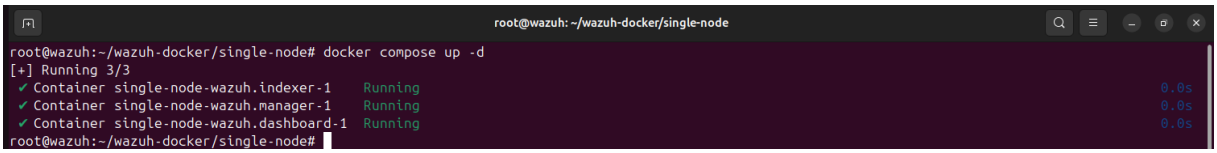


Figure 69: Démarrage des conteneurs Docker Wazuh (Manager, Indexer, Dashboard)

C.2 Vérification du Manager, de l'Indexer et du Dashboard

Wazuh constitue le cœur de notre SIEM. Nous avons opté pour un déploiement Docker single-node sur une VM dédiée (10.0.30.100) avec 8 GB RAM et 4 cores CPU. Cette configuration inclut trois conteneurs : Wazuh Manager, Wazuh Indexer et Wazuh Dashboard.

L'installation s'effectue via le script Docker Compose officiel fourni par Wazuh. Après le téléchargement du repository, nous avons lancé les conteneurs avec docker compose up -d.

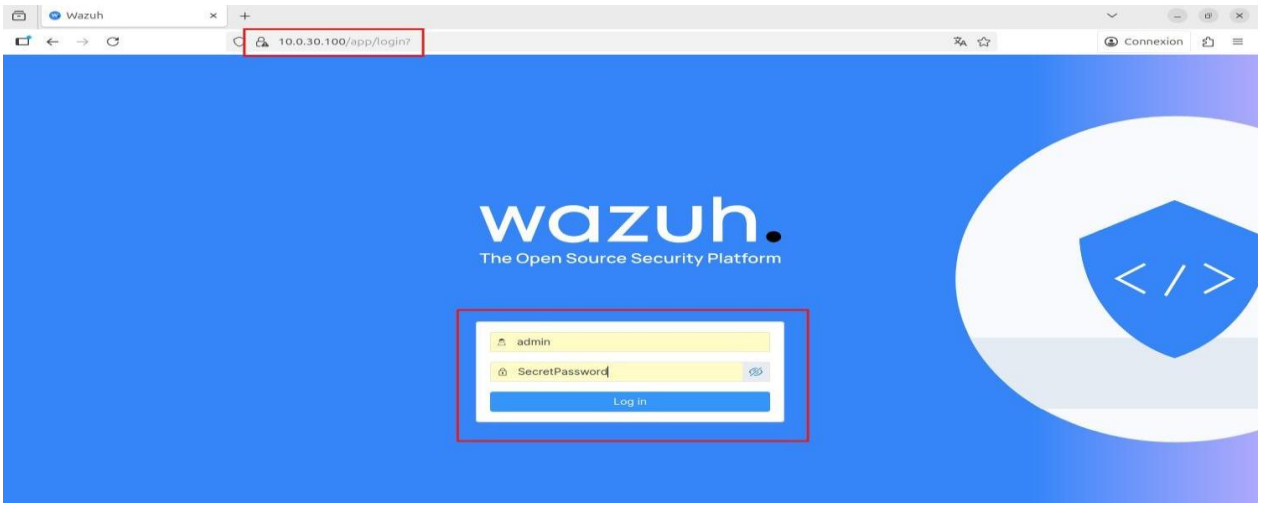


Figure 70: Démarrage des conteneurs Docker Wazuh (Manager, Indexer, Dashboard)

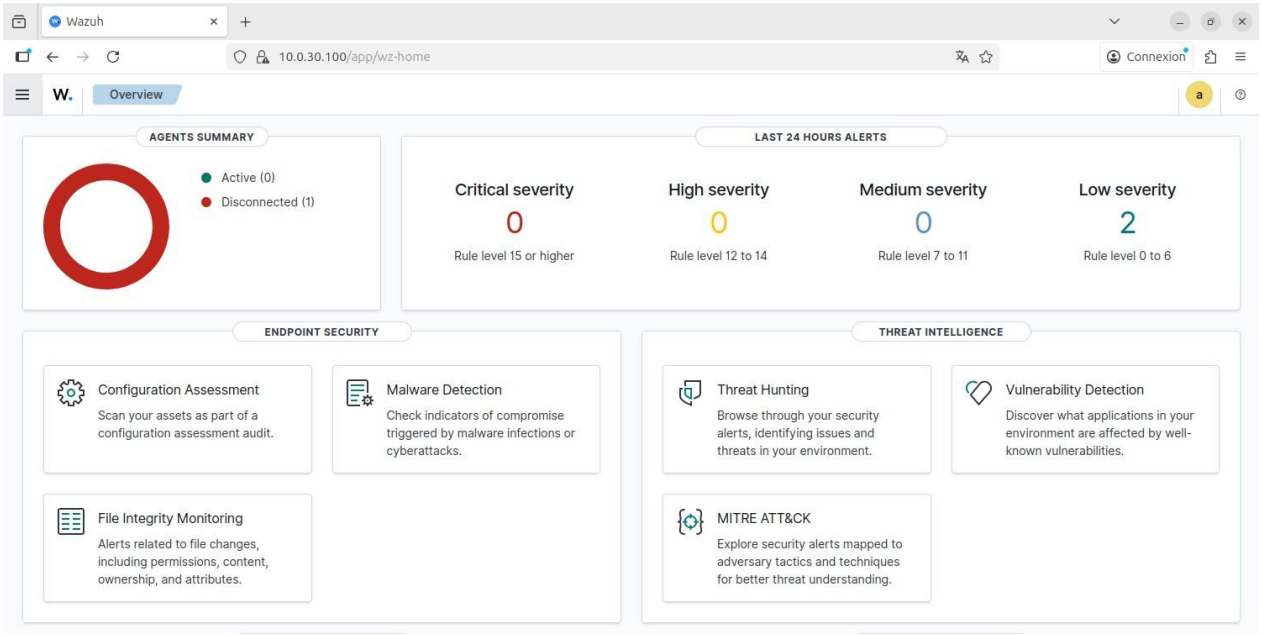


Figure 71: Dashboard Wazuh - Vue d'ensemble après première connexion

C.3 Déploiement des agents Wazuh sur les endpoints

Les agents Wazuh ont été déployés sur les machines à surveiller dans les segments LAN_USER et LAN_SERVER. Chaque agent a été enregistré auprès du Manager en utilisant une clé d'authentification unique générée lors de l'installation.

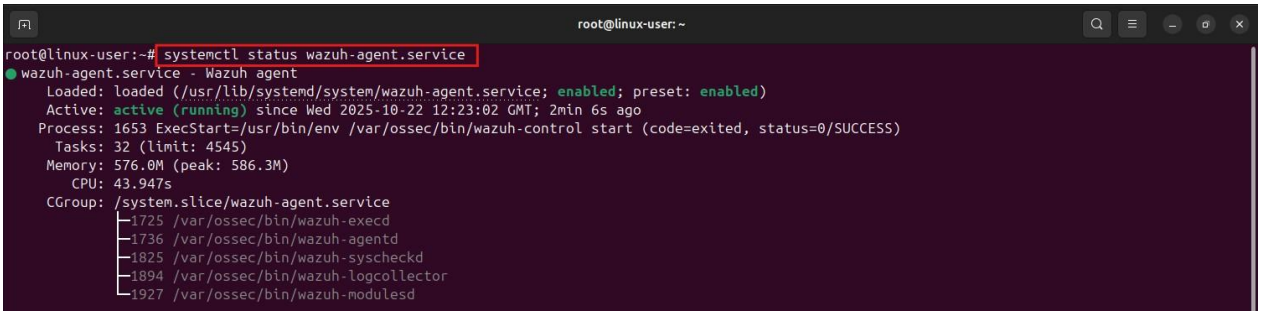


Figure 72: Vérification du service Wazuh Agent actif sur Linux User

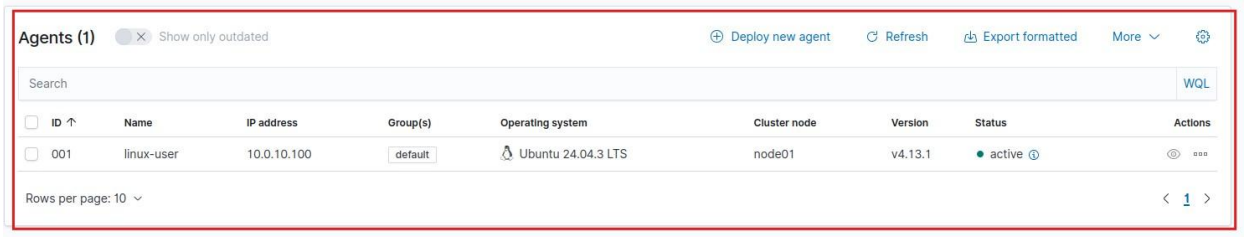


Figure 73: Agent Linux User (ID: 001) enregistré et actif dans le Dashboard Wazuh

C.4 Configuration des règles de détection personnalisées

Pour améliorer la détection, nous avons créé trois règles personnalisées dans le fichier local_rules.xml. Ces règles ciblent spécifiquement les attaques SSH brute force, les injections SQL et les tentatives d'exploitation web. Chaque règle déclenche une alerte de niveau élevé (10+) pour activer l'automatisation.[5]

Figure 74: Test de configuration et redémarrage de Wazuh après ajout des règles personnalisées

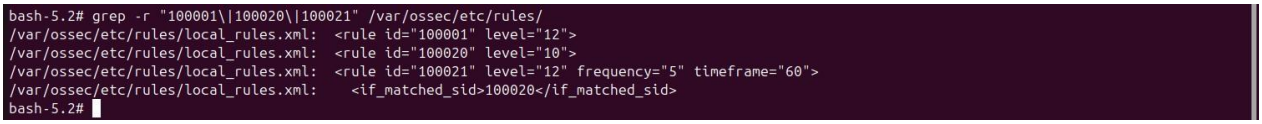


Figure 75: Vérification du chargement des règles personnalisées (100001, 100020, 100021)

Pour valider ces règles, nous avons configuré la collecte des logs Nginx et lancé des attaques de test depuis Kali Linux. Les résultats confirment que Wazuh détecte correctement les patterns d'attaques.

C.5 Configuration de la collecte des logs Nginx

Pour que les règles web fonctionnent, nous devons configurer l'agent Wazuh pour qu'il envoie les logs nginx au Manager. Sur le serveur Linux, nous éditons **/var/ossec/etc/ossec.conf** et, dans la section **ossec_config**, nous changeons le format de collecte des logs nginx d'apache à syslog, car le format apache ne remonte pas correctement les logs.

[5] *Le fichier complet local_rules.xml est disponible dans le dépôt GitHub du projet :* https://github.com/FodeMangane/soc-automation

```
<localfile>
  <log_format>syslog</log_format>
  <location>/var/log/nginx/access.log</location>
</localfile>

<localfile>
  <log_format>syslog</log_format>
  <location>/var/log/nginx/error.log</location>
</localfile>
```

Figure 76: Configuration de la collecte des logs Nginx dans ossec.conf

C.6 Tests de validation

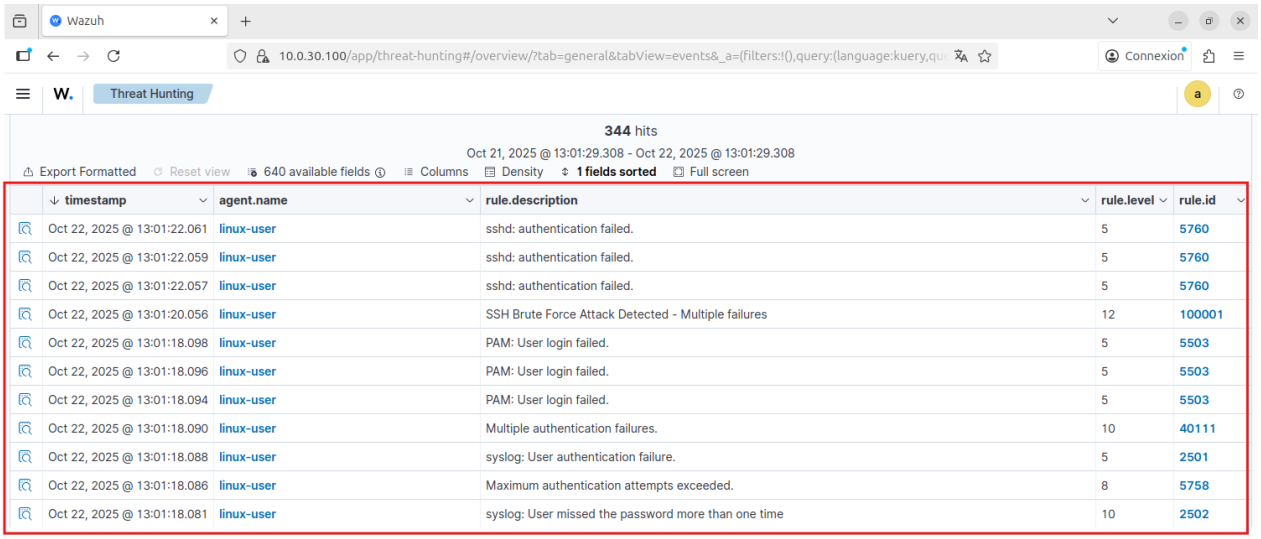


Figure 77: Détection de l'attaque SSH Brute Force dans Wazuh Threat Hunting

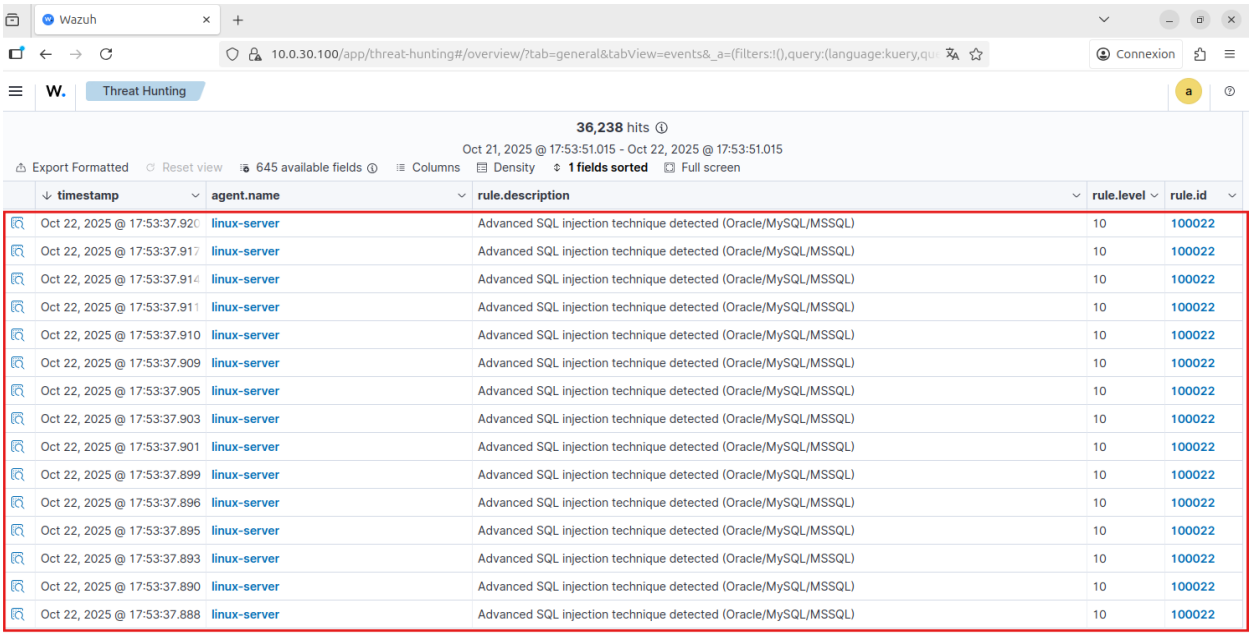


Figure 78: Détection des tentatives d'injection SQL avancées dans Wazuh

C.7 Configuration du remote syslog pour Suricata

L'intégration des alertes Suricata dans Wazuh repose sur l'activation de la réception syslog via le fichier *ossec.conf* sur le port UDP 514 et sur la création d'un décodeur JSON personnalisé pour analyser les événements EVE envoyés par pfSense/Suricata.

```
<remote>
  <connection>syslog</connection>
  <port>514</port>
  <protocol>udp</protocol>
  <allowed-ips>0.0.0.0/0</allowed-ips>
</remote>
```

Figure 79: Configuration du port UDP 514 pour recevoir les logs Suricata dans ossec.conf

C.8 Création du décodeur JSON Suricata

Nous avons créé un décodeur dans **/var/ossec/etc/decoders/local_decoder.xml** afin de parser correctement les événements **EVE JSON**. Ce décodeur extrait les champs essentiels : event_type, src_ip, dest_ip et alert.signature.

```
<!-- Suricata EVE JSON decoder -->
<decoder name="suricata-json">
  <prematch>{"timestamp":</prematch>
</decoder>

<decoder name="suricata-json">
  <parent>suricata-json</parent>
  <plugin_decoder>JSON_Decoder</plugin_decoder>
</decoder>
```

Figure 80: Création du décodeur JSON pour parser les événements Suricata

```
root@wazuh: ~/wazuh-docker/single-node
        0x0270:  7265 6769 6f6e 7322 3a31 2c22 7463 5f6d
        0x0280:  6178 5f72 6567 696f 6e73 223a 317d 7d
01:45:20.390344 veth54b5cbd Out IP (tos 0x0, ttl 63, id 31852, offset 0, flags [none], proto UDP (17), length 683)
    10.0.30.1.514 > 172.18.0.4.514: [udp sum ok] SYSLOG, length: 655
        Facility local1 (17), Severity notice (5)
        Msg: Oct 23 01:45:20 suricata[99392]: {"timestamp":"2025-10-23T01:45:20.395645+0000","flow_id":1822268008282530,"in_iface":"em2","event_type
":"flow","src_ip":"10.0.10.102","src_port":43852,"dest_ip":"10.0.20.100","dest_port":80,"proto":"TCP","app_proto":"http","flow":{"pkts_toserver":6,"
pkts_toclient":5,"bytes_toserver":728,"bytes_toclient":1000,"start":"2025-10-23T01:43:10.817495+0000","end":"2025-10-23T01:43:10.822231+0000","age":
0,"state":"closed","reason":"timeout","alerted":false},"tcp":{"tcp_flags":"1b","tcp_flags_ts":"1b","tcp_flags_tc":"1b","syn":true,"fin":true,"psh":t
rue,"ack":true,"state":"closed","ts_max_regions":1,"tc_max_regions":1}}
        0x0000:  3c31 3431 3e4f 6374 2032 3320 3031 3a34
        0x0010:  353a 3230 2073 7572 6963 6174 615b 3939
```

Figure 81: Capture réseau confirmant la réception des logs Suricata par Wazuh

Annexe D : Workflow Shuffle - Configuration détaillée

D.1 Création du workflow

Dans le menu de gauche, cliquer sur **Workflows**, puis cliquer sur le bouton **+ New Workflow** en haut à droite.

Remplir les informations du workflow :

- **Name** : Wazuh-Security-Alerts-Handler

- **Description** : Automated incident response workflow - Wazuh to TheHive/Cortex/MISP integration

Cliquer sur **Create**.

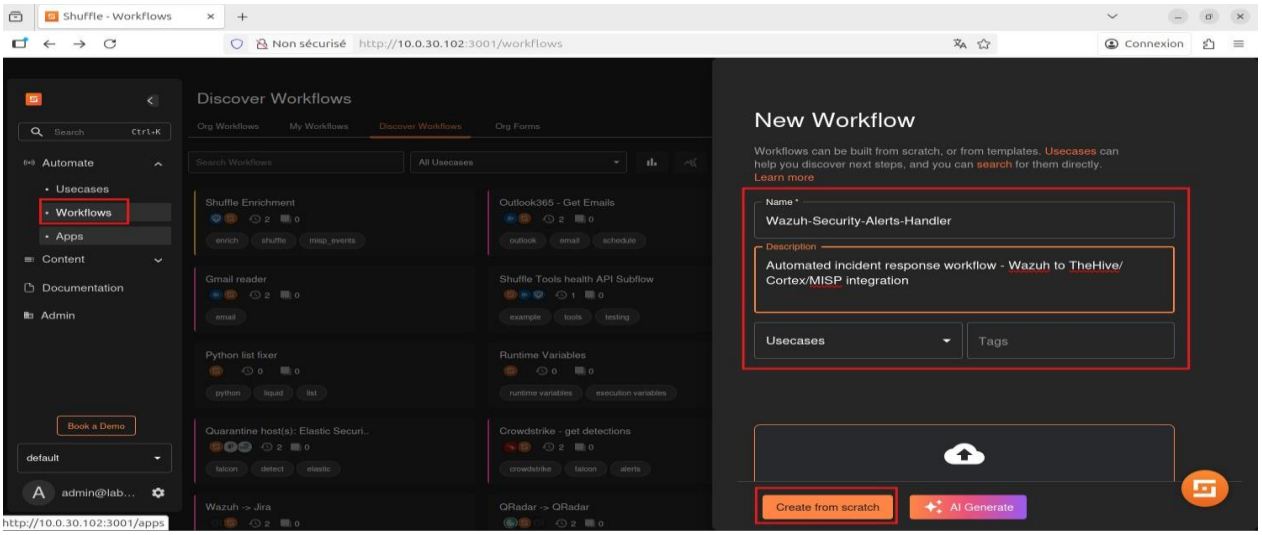


Figure 82: Création du workflow Wazuh-Security-Alerts-Handler dans Shuffle avec description

L'éditeur de workflow s'ouvre avec un canvas vierge où nous allons construire notre automatisation.

Composants du workflow à créer

Notre workflow comportera les composants suivants, que nous allons ajouter progressivement :

1. **Webhook** : Point d'entrée pour recevoir les alertes Wazuh
2. **Parse_Alert** : Script Python pour extraire et structurer les données
3. **TheHive_Create_Case** : Création automatique du cas d'incident
4. **TheHive_Add_Observable** : Ajout de l'IP comme observable IOC
5. **TheHive_Run_Analyzer** : Déclenchement de l'analyse Cortex
6. **MISP_Create_Event** : Création de l'événement de threat intelligence

7. **Send_Alert_to_Slack** : Notification à l'équipe SOC

Dans les sections suivantes, nous allons configurer chacun de ces composants et les connecter entre eux pour former la chaîne d'automatisation complète.

4.7.4 Ajout du Webhook - Point d'entrée des alertes

D.2 Configuration du Webhook (Module 1)

Le webhook constitue le **premier composant du workflow**, car il reçoit les alertes envoyées par Wazuh et déclenche l'exécution de toutes les actions suivantes.

Pour l'ajouter, dans la palette de gauche de l'éditeur, nous avons cliqué sur Triggers puis glissé l'élément Webhook sur le canvas central.

Ensuite, nous avons configuré le Webhook en cliquant dessus pour ouvrir ses paramètres :

- **Name :** Wazuh_Webhook

Shuffle génère automatiquement une URL webhook unique. Nous avons copié cette URL, car elle sera utilisée dans la configuration de Wazuh. Cette URL sert de **point d'entrée de notre automatisation** : chaque fois que Wazuh enverra une alerte critique à cette URL, le workflow s'exécutera automatiquement.

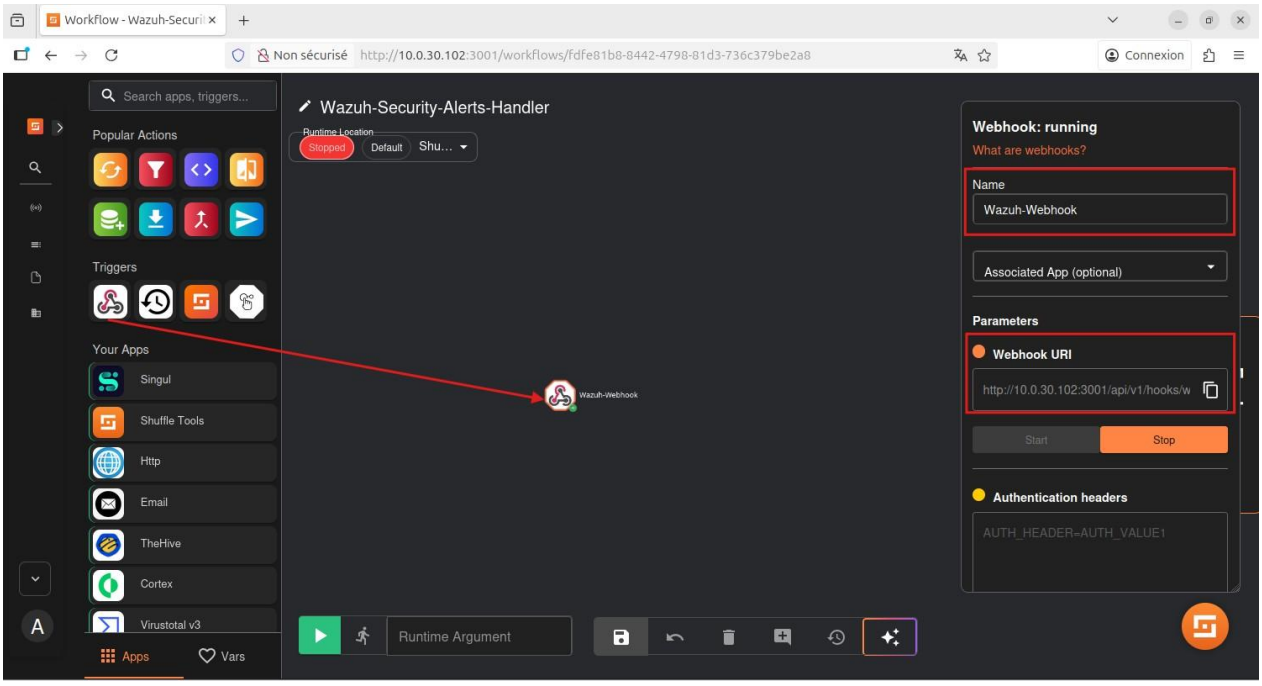


Figure 83: Ajout et configuration du trigger Webhook Wazuh dans le workflow Shuffle

D.3 Configuration Parse_Alert (Module 2)

Ce module Python extrait les informations essentielles de l'alerte Wazuh et les structure pour faciliter leur utilisation dans les étapes suivantes. Le script convertit les booléens JavaScript en Python, extrait les objets rule, agent et data, puis crée un dictionnaire structuré avec les 8 champs essentiels (rule_id, rule_description, rule_level, source_ip, target_agent, target_ip, attempted_user, timestamp).[6]

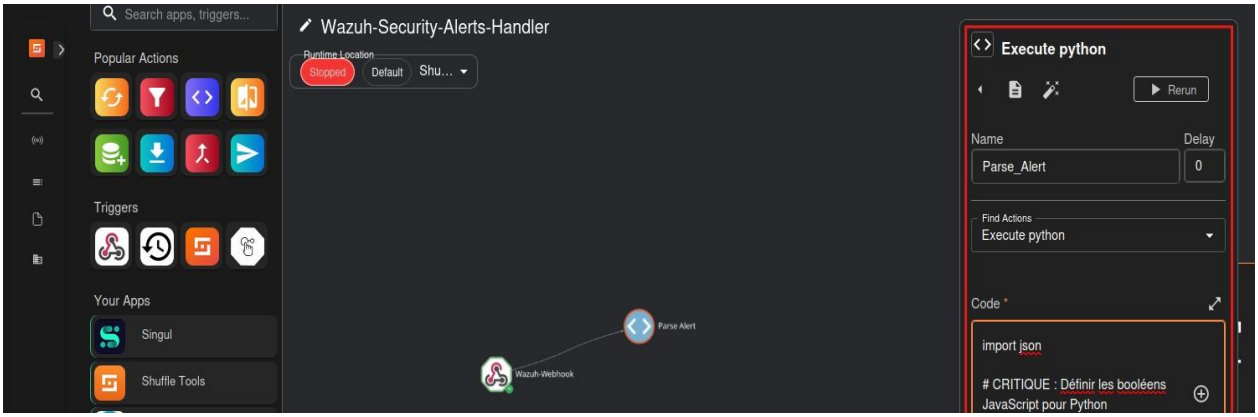


Figure 84: Configuration du module Parse_Alert pour l'extraction des données avec Python

D.4 Configuration TheHive_Create_Case (Module 3)

Cette action crée automatiquement un cas d'incident dans TheHive avec toutes les informations structurées de l'attaque. Le cas inclut un titre descriptif, une description détaillée avec les informations d'alerte, la sévérité (High), les protocoles TLP et PAP (AMBER), ainsi que des tags pour la catégorisation (ssh, brute-force, authentication-failure, automated-response).[7]

Pour l'ajouter, nous avons cherché TheHive dans la palette de gauche puis glissé l'élément TheHive sur le canvas, en le connectant à la sortie de Parse_Alert.

Pour configurer l'action TheHive_Create_Case, nous avons ouvert ses paramètres et sélectionné **Create case** dans la liste d'actions disponibles. Les paramètres de base sont :

- **Name :** TheHive_Create_Case

[6] Le code complet Parse_Alert.py est disponible dans le dépôt GitHub du projet : *https://github.com/FodeMangane/soc-automation*

[7] *La configuration complète est disponible dans TheHive_Create_Case.conf sur GitHub :* https://github.com/FodeMangane/soc-automation

- **URL :** http://10.0.30.104:9003

- **API Key :** (clé API de l'utilisateur TheHive)

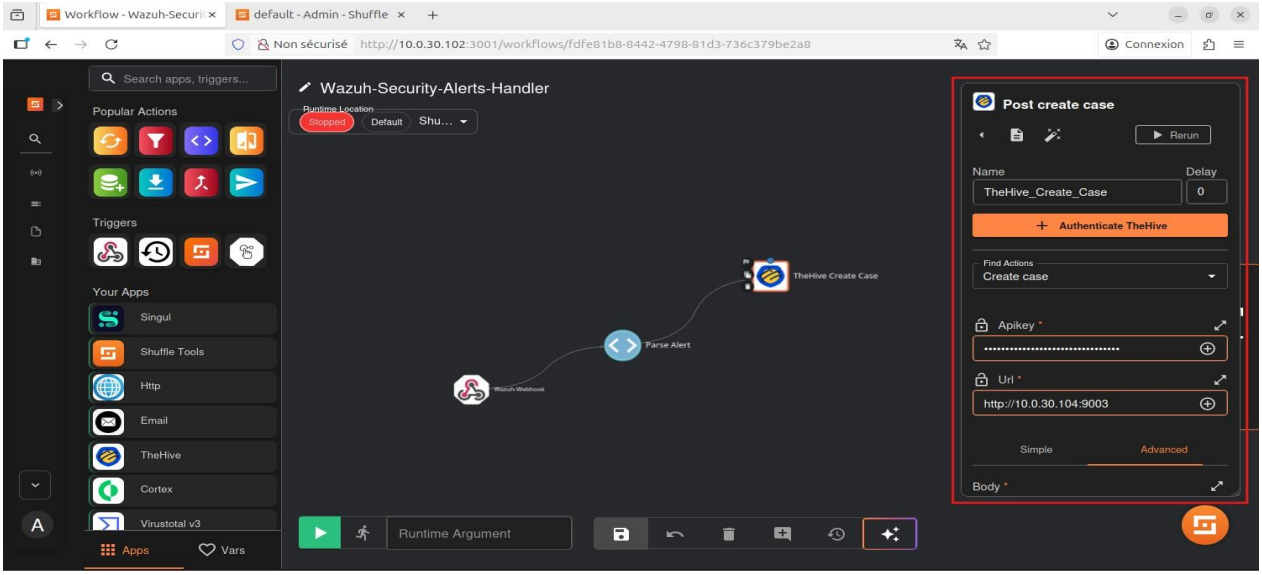


Figure 85: Configuration du module TheHive_Create_Case pour la création automatique de cas

D.5 Configuration TheHive_Add_Observable (Module 4)

Cette action ajoute automatiquement l'adresse IP source de l'attaque comme observable dans le cas TheHive. L'observable est marqué comme IOC (Indicator of Compromise) et tagué pour faciliter les recherches ultérieures. Le niveau TLP est défini à AMBER pour contrôler le partage de l'information.[8]

Figure 86: Configuration du module TheHive_Add_Observable pour l'ajout d'indicateurs de compromission

[8] *La configuration complète est disponible dans TheHive_Add_Observable.conf sur GitHub :* https://github.com/FodeMangane/soc-automation

D.6 Configuration TheHive_Run_Analyzer (Module 5)

Cette action lance automatiquement l'analyzer VirusTotal sur l'observable créé. TheHive envoie l'observable à Cortex qui exécute l'analyzer pour analyser la réputation de l'IP. Les résultats sont automatiquement attachés au cas avec les taxonomies appropriées (malicious, suspicious, safe).[9]

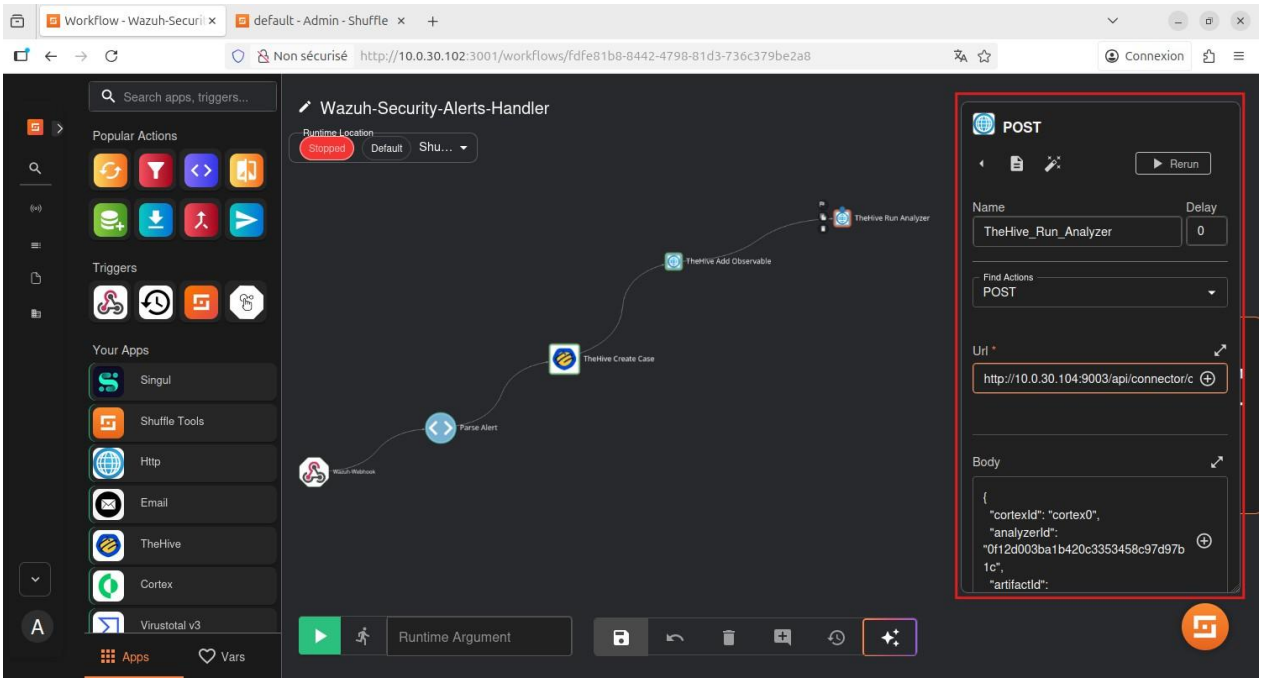


Figure 87: Configuration du module TheHive_Run_Analyzer pour l'analyse automatique avec VirusTotal

D.7 Configuration MISP_Create_Event (Module 6)

Cette action crée automatiquement un événement dans MISP pour documenter l'incident dans la base de threat intelligence. L'événement inclut l'IP source comme attribut de type ip-src, marqué comme IOC (to_ids: true), avec des tags appropriés (tlp:amber, type:OSINT) pour faciliter le partage avec la communauté de sécurité. [10]

[9] *La configuration complète est disponible dans TheHive_Run_Analyzer.conf sur GitHub :* https://github.com/FodeMangane/soc-automation

[10] *La configuration complète est disponible dans MISP_Create_Event.conf sur GitHub :* https://github.com/FodeMangane/soc-automation

Figure 88: Configuration du module MISP_Create_Event pour la documentation threat intelligence

D.8 Configuration Send_Alert_to_Slack (Module 7)

Cette dernière action envoie une notification complète à l'équipe SOC via Slack, incluant tous les détails de l'attaque et des liens directs vers le cas TheHive et l'événement MISP. Pour cela, nous avons glissé un élément *HTTP* sur le canvas et connecté la sortie de *MISP_Create_Event* à ce nouvel élément. [11]

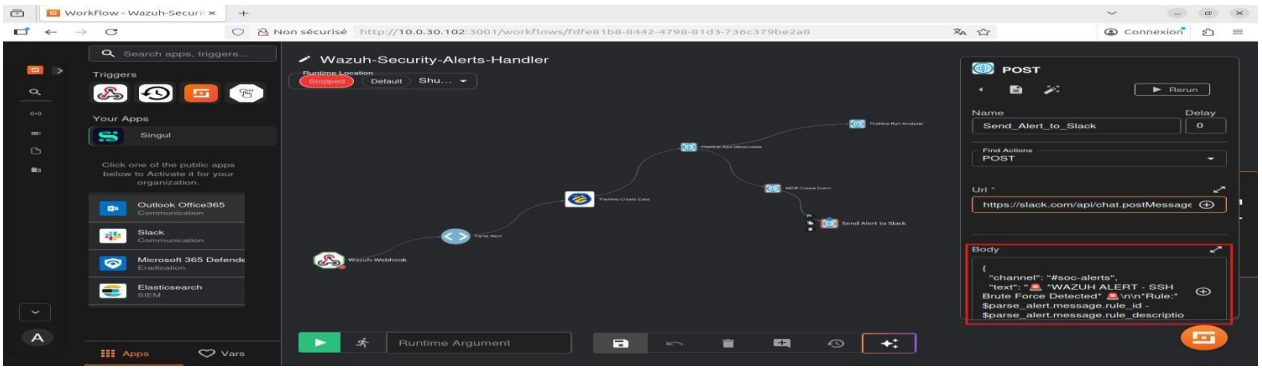


Figure 89: Configuration du module Send_Alert_to_Slack pour les notifications au SOC

[11] Le contenu du body se trouve à la fin du document Slack_Bot.md, disponible sur GitHub: https://github.com/Fode=Mangane/soc-automation

Table des matières