Ecole Supérieure Privée des Technologies et de l'Ingénierie

Spécialité : SSIR-D

Rapport de Projet CyberOps

Conception et Mise en place d'une Solution SOC Open Source

Réalisée par : Wissem nasri

Encadré par : Tarek Hdiji

Année Universitaire : 2024 – 2025

Table des matières

Introduction Générale 1
Chapitre 1 : Cadre général du projet 2
1 Introduction 2
2 Méthodologie du Travail 2
2.1 Méthode Agile 2
2.2 Méthode Classique ou Traditionnelle 3
2.3 Méthode Adoptée 3
3 Cahier des Charges 4
3.1 Contexte du Projet 4
3.2 Spécifications des Besoins 4
3.2.1 Besoins Fonctionnels 5
3.2.2 Besoins Non Fonctionnels 5
4 Infrastructure Réseau 5
4.1 Détails sur la connectivité des dispositifs 6
4.2 Architecture du Réseau 7
5 Conclusion 8
Chapitre 2 : Les Menaces et Solutions 9
1 Introduction 9
2 Menaces potentielles pour l'architecture 9
3 Sécurité Défensive 10
4 Fondations d'un Security Operation Center 10
4.1 Les Ressources Humaines 11
4.2 Les Processus 12
4.3 Technologies et Outils 12
4.4 Rôles et Responsabilités 13
5 Conclusion 14
Chapitre3 : conception de la solution 15
1 Introduction : 15
2 L'utilisation des solutions open-source 15
2.1 Le rôle de l'open-source en cybersécurité 15
2.2 Avantages des outils open-source 15

2.3 Outils SOC open-source courants 16
2.4 Exigences et problèmes de compatibilité des outils open-source dans un SOC 16
2.4.1 Exigences techniques 16
2.4.2 Problèmes de compatibilité 17
3 Outils et composants open-source pour la solution SOC 17
3.1 Wazuh: Endpoint Detection and Response 18
3.1.1 Fonctionnalités de Wazuh 19
3.1.2 Cas d'utilisation de Wazuh 19
3.1.3 Architecture et composants de Wazuh 20
3.1.4 L'utilité de Wazuh dans la solution SOC 20
3.2 TheHive 5 : Réponse aux incidents et gestion des cas 21
3.2.1 Fonctionnalités et cas d'utilisation de TheHive 5 21
3.2.2 Architecture et composants de TheHive 22
3.2.3 L'utilité de TheHive dans la solution SOCaaC 23
3.3 Cortex : Orchestration, Automatisation et Réponse en Sécurité 23
3.3.1 Fonctionnalités de Cortex 24
3.3.2 Cas d'Utilisation de Cortex 24
3.3.3 Architecture et Composants de Cortex 25
3.3.4 L'Utilité de Cortex dans la Solution SOC 26
3.4 Plateforme de Partage d'Informations sur les Menaces 26
3.4.1 Fonctionnalités de MISP 27
3.4.2 Cas d'Utilisation de MISP 27
3.4.3 Architecture et Composants de MISP 28
3.4.4 L'Utilité de MISP dans la Solution SOC 29
4 Conclusion 29
Chapitre4 : implimentation de la solution 31
1 Introduction 31
2 Aperçu de l'architecture SOC 31
2.1 Architecture de haut niveau 31
2.1.1 Wazuh 32
2.1.2 TheHive 33

2.1.3 Cortex 33
2.1.4 MISP 33
2.2 Interaction des Composants et Flux de Données 33
2.2.1 Flux de Détection et de Création d'Alerte 34
2.2.2 Flux d'Extraction et d'Enrichissement des Observables 35
2.2.3 Flux d'Analyse et de Mitigation des Menaces 36
2.3 Stratégie de Déploiement du SOC 36
2.3.1 Deployement de l'environement 38
2.4 Stratégie d'Implémentation des Composants SOC 38
2.4.1 Présentation des Services Déployés via Docker Compose 39
2.4.2 Installation wazuh 44
2.5 Vision globale de la solution 45
3 Intégration 46
3.1 Intégration entre The Hive et cortex 46
3.2 Intégration entre The Hive et MISP 47
3.3 intégration entre wazuh et the Hive 49
3.3.1 Création du script d'intégration personnalisé 49
3.3.2 Création d'un script Bash 49
3.3.3 Activation l'intégration dans la configuration de Wazuh 50
4 Test et réalisation 51
4.1 Test de fonctionnement 51
4.1.1 Domain 51
4.1.2 Hash 54
4.2 Test brute force attack 56
5 Conclusion 59

Table des figures

Figure 1:processus agile 3
Figure 2:modéle en cascade 3
Figure 3:methode scrum 4
Figure 4:configuration des resaux virtuels 7
Figure 5:architecture existant 8
Figure 6:composants du soc 11
Figure 7:roles et responsbilité de soc 13
Figure 8:wazuh logo 18
Figure 9:les composants de wazuh 20
Figure 10:The Hive logo 21
Figure 11:architecture et composants TheHive 22
Figure 12:logo cortex 23
Figure 13:architecture et composants du cortex 25
Figure 14:Logo MISP 26
Figure 15:Architecture et Composants de MISP 28
Figure 16:architecture de SOC 32
Figure 17:Flux de detection et de creation d'Alerte 34
Figure 18:Flux d'Extraction et d'Enrichissement des Observables 35
Figure 19:Flux d'Analyse et de Mitigation des Menaces 36
Figure 20:Configuration du Service TheHive 39
Figure 21:Configuration du Service Cassandra 40
Figure 22: Configuration du Service Elasticsearch 40
Figure 23:Configuration du Service MinIO 41
Figure 24:Configuration du Service Cortex 41
Figure 25 : Configuration du Service MISP 42
Figure 26 : Configuration de la Base de Données MySQL pour MISP 43
Figure 27 : Configuration du Service Redis 43
Figure 28: Configuration des Modules MISP 44
Figure 29:instalation wazuh 44
Figure 30:interface utilisateur web intègre a Kibana 45
Figure 31:architecture complète 46
Figure 32:Créez un utilisateur et Générez une clé API 46
Figure 33:Configuration dans The Hive 47
Figure 34:Générez une clé API MISP 47
Figure 35:Ajoutez la configuration de MISP dans le fichier application.conf de The Hive 48
Figure 36: intégration effectué de cortex et misp 48
Figure 37:integration misp et cortex 49
Figure 38:un script Python nommé custom-w2thive.py 49

Figure 39:script bash 49
Figure 40:API Key the Hive 50
Figure 41:Activation l'intégration dans la configuration de Wazuh..... 51
Figure 42:exemple d'un Domain enlevé depuis misp (IOCs) 52
Figure 43:creation d'un observation de type domain 52
Figure 44:analyse des IOCs existant..... 53
Figure 45:investigation dans les IOCs existant +resulatat+rapport..... 53
Figure 46:ajout d'une observation de type hash (wannacry) 54
Figure 47:resultat de l'analyse via VIRUSTOTAL 54
Figure 48:rapport généré 55
Figure 49:username+ip address de l'agent 56
Figure 50:lancement de l'attaque..... 56
Figure 51:detection par wazuh..... 57
Figure 52:détection par the Hive 57
Figure 53:information sur l'attaquant 58
Figure 54:creation d'une case et ajout de l'adresse de l'acteur malveillant..... 58
Figure 55:exportation de case vers misp..... 59

Introduction Générale

Dans un environnement numérique en constante évolution, les organisations font face à une multitude de menaces qui compromettent la sécurité de leurs systèmes d'information. La protection des données et des ressources sensibles est devenue un impératif pour maintenir la confiance des clients et partenaires, tout en assurant la conformité avec les réglementations de sécurité. Pour répondre à ces défis, de nombreuses entreprises choisissent de mettre en place un Security Operations Center (SOC), un centre dédié à la surveillance en temps réel, à la détection des incidents et à la gestion des réponses aux menaces.

TEK-UP, une organisation proactive dans la gestion de ses infrastructures informatiques, a reconnu l'importance de la cybersécurité et a décidé d'intégrer un SOC dans son architecture réseau existante. L'objectif est de centraliser la surveillance des événements de sécurité, d'améliorer la réactivité face aux cyberattaques et de garantir une gestion plus efficace des incidents de sécurité. Ce projet s'inscrit dans une démarche de renforcement de la résilience de l'infrastructure informatique de l'organisation, en anticipant les menaces tout en optimisant les ressources disponibles pour leur gestion.

Le SOC sera mis en place dans le cadre de l'architecture réseau actuelle de TEK-UP, composée de plusieurs segments distincts tels que le WAN, le LAN, le LAN2 et la DMZ, chacun ayant un rôle spécifique dans la gestion des flux de données et la protection du système global. L'intégration du SOC, via des outils tels que le SIEM (Security Information and Event Management) et le SOAR (Security Orchestration, Automation, and Response), permettra de centraliser les alertes de sécurité, d'assurer une analyse continue des incidents et de faciliter la gestion des réponses automatisées face aux menaces détectées.

Ce projet vise non seulement à renforcer la sécurité de l'infrastructure, mais aussi à offrir une plus grande visibilité sur les opérations réseau, à améliorer la gestion des risques et à augmenter l'efficacité de la réponse aux incidents, tout en garantissant une protection proactive contre les cybermenaces.

Chapitre 1 : Cadre général du projet

1 Introduction

Ce chapitre présente le cadre général du projet, incluant la méthodologie à suivre tout au long du travail, le cahier des charges (expliquant ses objectifs et sa pertinence), ainsi qu'une description de l'infrastructure existante.

2 Méthodologie du Travail

Un projet informatique, quelle que soit sa taille et la portée de ses objectifs, nécessite la mise en place d'un plan organisationnel tout au long de son cycle de vie. La modélisation consiste à créer une présentation simplifiée d'un projet. Grâce au modèle, il est possible de représenter un projet, un concept et d'identifier son niveau de complexité. Parmi les méthodes les plus connues, nous citons la méthode Agile et la méthode classique.

2.1 Méthode Agile

La méthode Agile est plus efficace et moins rigide que la méthode classique. Ce type de méthodologie permet une grande flexibilité et une meilleure visibilité dans la gestion du projet, ce qui permet à l'équipe d'être plus réactive aux besoins du client. Ainsi, le projet est partagé en mini-projets, nécessitant la validation du client avant de passer à la phase suivante, tout en prenant en considération l'évolution de ses besoins. Cette méthode est particulièrement adaptée aux grands projets au sein des grandes entreprises, offrant une meilleure adaptabilité, visibilité et gestion des risques. Les objectifs dépendent des besoins du client. Le processus de l'approche Agile est illustré par la figure 1.

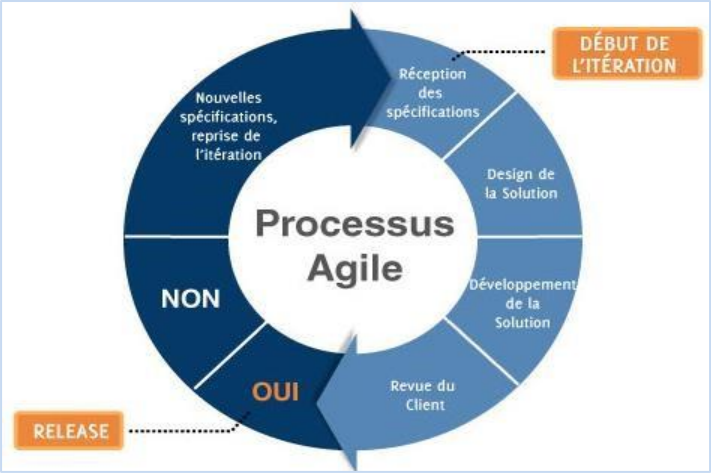

Figure 1:processus agile

2.2 Méthode Classique ou Traditionnelle

Le modèle en cascade est un modèle de développement séquentiel dans lequel les exigences doivent être claires avant de passer à la phase suivante de conception. Chaque produit de travail ou activité doit être terminé avant de passer à la phase suivante, et les tests sont effectués à la fin de chaque phase, ce qui contribue à maintenir la qualité du projet. Ce modèle comprend six phases. La figure suivante montre les différentes phases de l'approche traditionnelle.

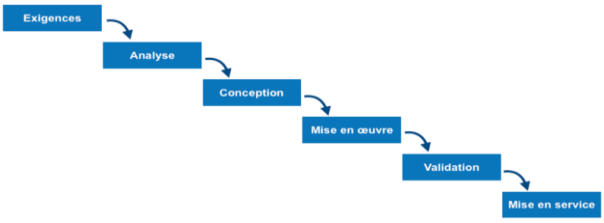

Figure 2:modéle en cascade

2.3 Méthode Adoptée

Étant donné que notre projet nécessite des tests après chaque tâche et peut être divisé en sous-projets, nous avons choisi d'adopter la méthode Scrum de l'approche Agile. Scrum est aujourd'hui l'approche la plus populaire en raison de plusieurs avantages, tels que l'amélioration de l'exécution, la qualité, des délais fixes et la livraison de logiciels fonctionnels. La figure 3 présente un aperçu général sur le cycle de vie de l'approche Scrum.

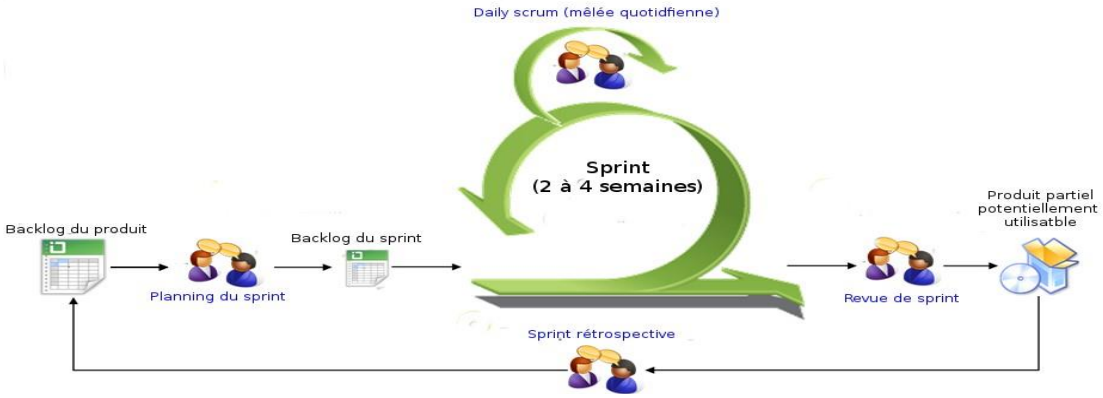

Figure 3:methode scrum

Durant notre projet, cette méthode nous permet d'accepter les changements, même tard dans le développement, tout en maintenant un rythme durable.

3 Cahier des Charges

Le cahier des charges définit de manière précise les besoins et les objectifs du projet, en s'assurant que l'implémentation du Security Operations Center (SOC) respecte les exigences fonctionnelles, techniques et sécuritaires. Il est structuré autour de deux grandes parties : le contexte du projet et les spécifications des besoins, ces dernières étant subdivisées en besoins fonctionnels et besoins non fonctionnels.

3.1 Contexte du Projet

Le projet d'implémentation d'un SOC dans l'infrastructure de TEK-UP vise à renforcer la sécurité des systèmes d'information, notamment en centralisant la gestion des incidents de sécurité et en améliorant la visibilité des événements de sécurité au sein du réseau de l'entreprise. TEK-UP dispose déjà d'une architecture réseau robuste, comprenant pfSense (en tant que pare-feu et routeur) avec plusieurs interfaces (WAN, LAN, LAN2, DMZ). Le SOC sera intégré dans la DMZ, où il pourra surveiller les événements de sécurité en provenance de toutes les interfaces réseau tout en étant isolé du réseau interne pour minimiser les risques.

L'objectif est de déployer une solution SOC qui permette une surveillance continue, une détection rapide des intrusions, une gestion automatisée des incidents et des réponses en temps réel. Le SOC devra également faciliter l'archivage et l'analyse post-incident pour assurer une amélioration continue des processus de sécurité.

3.2 Spécifications des Besoins

Les spécifications des besoins sont divisées en deux grandes catégories : les besoins fonctionnels, qui décrivent les fonctionnalités attendues du SOC, et les besoins non

fonctionnels, qui traitent des aspects de performance, de sécurité et de maintenance du système.

3.2.1 Besoins Fonctionnels

- Collecte et centralisation des logs : Intégration avec les dispositifs de sécurité pour collecter et analyser les logs en temps réel.
- Analyse des événements : Corrélation des événements pour détecter les intrusions et anomalies.
- Gestion des incidents : Notification et classification des incidents avec une réponse automatisée.
- Tableaux de bord et rapports : Fourniture de rapports détaillés et d'une vue d'ensemble des événements de sécurité.
- Analyse forensique et archivage : Conservation sécurisée des logs pour des analyses post-incident.

3.2.2 Besoins Non Fonctionnels

- Scalabilité : Capacité à évoluer avec la croissance de l'infrastructure.
- Haute disponibilité : Redondance et tolérance aux pannes pour garantir une disponibilité continue.
- Sécurité du SOC : Contrôles d'accès stricts et protection des données sensibles.
- Performance : Réactivité en temps réel pour la gestion des incidents.
- Interopérabilité : Compatibilité avec les systèmes existants, notamment pfSense et autres outils de sécurité.

4 Infrastructure Réseau

L'architecture réseau actuelle de TEK-UP repose sur un système structuré et sécurisé pour assurer la communication entre les différentes zones et garantir la sécurité des données. Cette architecture est gérée par un pare-feu pfSense qui dispose de quatre interfaces : WAN (connexion à Internet), LAN (serveurs et clients virtuels), LAN2 (zone dédiée à la Red Team pour les tests d'intrusion), et DMZ (zone de services exposés). L'infrastructure utilise des réseaux virtuels pour interconnecter efficacement ces différentes zones et assurer la gestion fine des flux de données.

4.1 Détails sur la connectivité des dispositifs

Tous les dispositifs de l'infrastructure sont connectés via des réseaux virtuels (VMnet), permettant une gestion fluide et isolée des différentes interfaces. Ces réseaux virtuels facilitent la communication entre les zones tout en garantissant une isolation adéquate des différentes parties du réseau.

- VMnet8 : 192.168.122.0 – Utilisé comme interface WAN, cette plage d'adresses permet la connexion externe à Internet, gérée par pfSense pour contrôler l'accès à l'extérieur et filtrer le trafic entrant et sortant.

- VMnet5 : 10.0.2.0 – Affecté à la DMZ, ce réseau héberge les services accessibles depuis l'extérieur(apache,dns...).

- VMnet10 : 192.168.153.0 – Utilisé pour le LAN virtuel, cette interface regroupe les serveurs et les clients internes. Elle permet de garantir une communication interne sécurisée entre les ressources nécessaires au bon fonctionnement de l'organisation.

- VMnet4 : 192.168.118.0 – Dédicacé au LAN2, cette interface est réservée à la Red Team, qui l'utilise pour effectuer des tests d'intrusion. Elle est isolée des autres zones du réseau pour éviter toute interférence avec les opérations internes tout en permettant des simulations d'attaques sur le réseau.

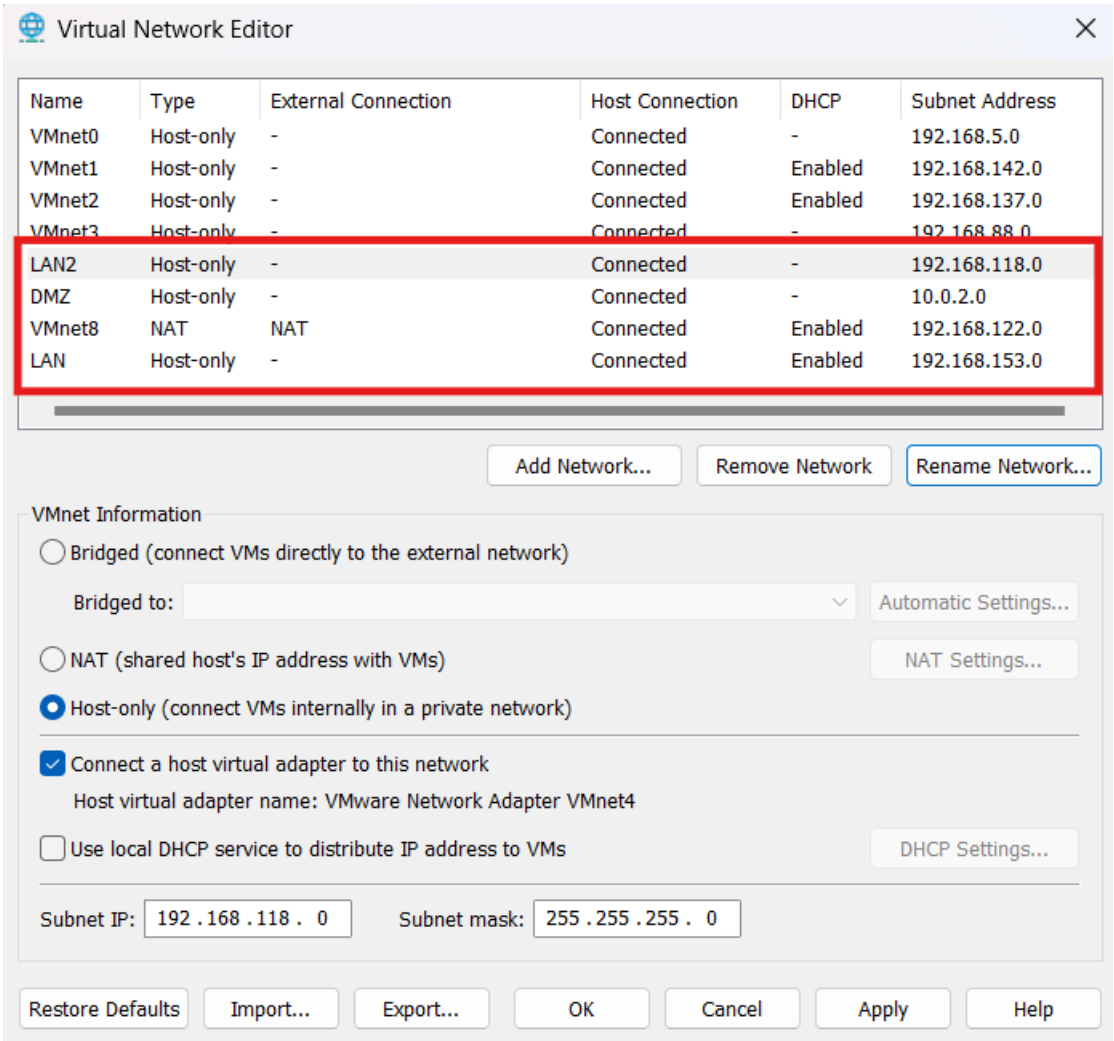

Figure 4:configuration des resaux virtuels

4.2 Architecture du Réseau

Voici une illustration de l'architecture réseau actuelle de TEK-UP, montrant les différentes zones et interfaces du réseau :

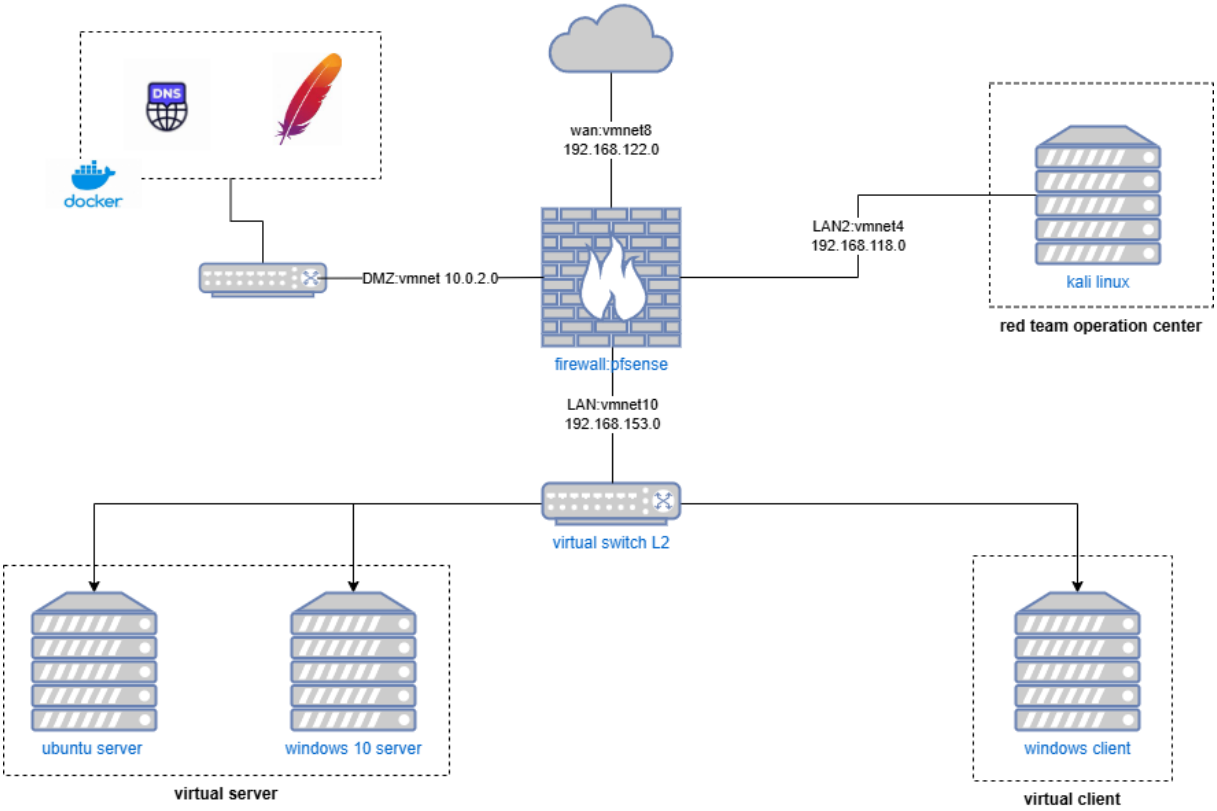

Figure 5:architecture existant

5 Conclusion

En conclusion, ce chapitre a permis de poser les bases du projet en présentant la méthodologie à suivre, en détaillant le cahier des charges et son rôle essentiel dans la définition des objectifs, ainsi qu'en fournissant une description de l'infrastructure actuelle. Dans le chapitre suivant, nous aborderons les menaces pesant sur cette architecture et nous discuterons des mesures à mettre en place pour résoudre cette problématique.

Chapitre 2 : Les Menaces et Solutions

1 Introduction

L'architecture réseau actuelle de TEK-UP, incluant un pare-feu pfSense avec quatre interfaces, expose certains points vulnérables qui nécessitent une surveillance et une gestion centralisée des événements de sécurité. Dans ce chapitre, nous allons analyser les menaces potentielles qui pèsent sur l'architecture, puis présenter des solutions adaptées, notamment à travers l'intégration d'un SOC (Security Operations Center) .

2 Menaces potentielles pour l'architecture

Dans le contexte de l'architecture existante, les principales menaces auxquelles le réseau est exposé incluent :

Menaces liées à pfSense (interface WAN) :

Le pare-feu pfSense est la première ligne de défense, mais il peut être ciblé par des attaques telles que des scans de ports, des attaques par déni de service (DDoS), ou des tentatives de contourner les règles de filtrage via des vulnérabilités non corrigées.

Menaces sur le LAN (serveurs et clients virtuels) :

Le LAN interne est exposé à des attaques telles que les intrusions internes, les exploits des systèmes non patchés, et les attaques de type man-in-the-middle si le trafic n'est pas suffisamment sécurisé.

Menaces sur le DMZ (zone des services exposés) :

Les serveurs dans la DMZ, comme les serveurs web ou les serveurs de messagerie, sont plus vulnérables aux attaques externes, notamment les injections SQL, les attaques XSS (Cross-Site Scripting), ou encore les attaques par force brute sur les applications accessibles publiquement.

Menaces sur LAN2 (Red Team) :

Cette zone dédiée à la Red Team pour les tests d'intrusion est un point critique car elle peut être utilisée pour simuler des attaques sur le réseau. Les vulnérabilités de cette zone doivent être gérées efficacement pour éviter qu'elles ne soient exploitées par des attaquants externes.

Menaces externes (Internet) :

Les connexions externes (via le WAN) sont exposées à des menaces provenant de l'Internet, telles que des malwares, des botnets, ou des attaques DDoS visant à perturber les services.

L’analyse des menaces présentes dans l’architecture existante souligne le besoin d'une solution de sécurité renforcée. L’implémentation d’un SOC (Security Operations Center) dans la DMZ devient ainsi indispensable pour assurer une surveillance continue et une réponse rapide aux incidents de sécurité. Ces solutions permettront une gestion optimale des menaces et l'automatisation des réponses aux incidents, consolidant ainsi une posture de cybersécurité proactive et solide.

3 Sécurité Défensive

La sécurité défensive est une approche proactive visant à protéger les actifs numériques, les systèmes et les réseaux contre les menaces et attaques potentielles. Elle utilise un large éventail de stratégies, de technologies et de pratiques conçues pour détecter, prévenir et atténuer les risques pour l'infrastructure informatique d'une organisation.

En intégrant des mesures de sécurité défensive telles que les systèmes de détection d'intrusion, les contrôles d'accès et la formation à la sensibilisation à la sécurité, les organisations renforcent leurs défenses contre les menaces potentielles. Ces mesures sont surveillées et gérées de près par les Security Operations Centers (SOC), qui jouent un rôle essentiel dans la détection, l'analyse et la réponse aux incidents de sécurité en temps réel.

Les SOCs exploitent des technologies avancées, des renseignements sur les menaces et l'automatisation de la sécurité pour renforcer les capacités défensives, assurant ainsi une posture proactive face aux menaces cybernétiques en constante évolution. Cette intégration harmonieuse de la sécurité défensive avec les opérations SOC forme une base solide pour une posture de cybersécurité efficace au sein des organisations.

4 Fondations d'un Security Operation Center

Un Security Operation Center, ou SOC, est une unité organisationnelle ou commerciale centrale qui opère pour gérer et améliorer la posture de sécurité globale d'une organisation. Sa fonction principale est de détecter, analyser et répondre aux événements de cybersécurité, y compris les menaces et incidents, en mobilisant des ressources humaines, des processus et des technologies. Les équipes sont responsables de la gestion de l'infrastructure de sécurité ainsi que de la configuration et du déploiement de diverses solutions, outils et produits de sécurité.

Figure 6:composants du soc

4.1 Les Ressources Humaines

L'élément humain dans les SOCs englobe une gamme de rôles, chacun apportant une expertise et une perspective unique dans l'effort de défense collectif. Du chasseur de menaces qui analyse les données réseau à l'investigateur en criminalistique qui examine les attaques, chaque individu joue un rôle essentiel dans le renforcement de la sécurité de l'organisation.

Voici quelques rôles critiques :

- **Analystes de Sécurité** : Professionnels formés pour surveiller les outils de sécurité, analyser les alertes, enquêter sur les menaces et coordonner la réponse aux incidents. Leur mission inclut l'analyse des données de sécurité pour identifier les vulnérabilités, l'enquête sur les alertes pour en déterminer la cause, et la recommandation de solutions pour atténuer les risques.
- **Spécialistes du Renseignement sur les Menaces** : Ces spécialistes recueillent des informations sur les cybermenaces. Leur mission est de rechercher les menaces émergentes, analyser les renseignements pour comprendre les motivations et capacités des attaquants, et fournir des informations pour orienter les efforts de sécurité.
- **Répondants aux Incidents** : Ces professionnels gèrent les violations de sécurité et cyberattaques, avec pour mission de contenir les incidents pour minimiser les

dommages, enquêter pour en déterminer l'ampleur et la cause, et diriger les efforts de restauration des systèmes affectés.

- **Ingénieurs de Sécurité** : Ils maintiennent et optimisent les technologies et l'infrastructure de sécurité du SOC. Leur mission est de concevoir et implémenter des contrôles de sécurité, configurer les pare-feux et autres outils de sécurité, et renforcer les systèmes pour les rendre plus résistants aux attaques.

4.2 Les Processus

Au-delà des compétences des individus, le SOC repose sur divers processus pour orchestrer une réponse efficace à chaque menace. Ces processus incluent :

- **Classification et Triage des Événements** : Ce processus permet de trier les alertes pour identifier les problèmes les plus urgents et prioriser les ressources en conséquence.
- **Remédiation et Récupération** : Consiste à désinfecter les systèmes, restaurer les données et renforcer les défenses, en priorisant les actifs critiques pour minimiser l'impact sur les activités.
- **Amélioration Continue** : Les processus SOC sont revus et mis à jour régulièrement pour contrer les nouvelles menaces. Des sessions de formation et des simulations sont organisées pour garantir la préparation de l'équipe.
- **Rapport** : La transparence étant essentielle en cybersécurité, les rapports SOC documentent toutes les activités au sein du réseau, essentiels pour les audits et la conformité.

4.3 Technologies et Outils

Dans un SOC, des processus bien définis sont essentiels pour une détection et une réponse aux menaces efficaces. Cependant, ces processus atteignent leur plein potentiel lorsqu'ils sont associés aux bons outils, tels que :

- **Gestion de l'Information et des Événements de Sécurité (SIEM)** : Agrège et analyse les journaux des divers outils de sécurité, offrant visibilité en temps réel et capacités de détection des menaces.
- **Systèmes de Détection et de Prévention des Intrusions (IDS/IPS)** : Surveillent le trafic réseau pour détecter et bloquer les activités malveillantes en temps réel.
- **Détection et Réponse aux Points de Terminaison (EDR)** : Surveille les points d'extrémité pour détecter les signes de malware et autres menaces, permettant une réponse rapide.

- **Orchestration, Automatisation et Réponse de Sécurité (SOAR)** : Automatise les tâches et flux de travail pour améliorer l'efficacité et les temps de réponse grâce à des actions automatisées.

4.4 Rôles et Responsabilités

Le SOC sert de noyau d'intelligence et de système nerveux central pour la défense en cybersécurité de l'organisation. Composé d'une équipe spécialisée et d'une infrastructure dédiée, il a pour responsabilité de surveiller, détecter, analyser et répondre aux menaces et incidents de sécurité. La structure du SOC est généralement organisée en trois niveaux, avec un SOC Manager à sa tête.

Figure 7:roles et responsbilité de soc

- **Niveau 1 : Spécialiste du Triage** : Responsable de la collecte de données brutes et de la révision des alarmes. Le spécialiste doit confirmer et enrichir les alertes avec des données pertinentes pour éviter la fatigue d'alerte.
- **Niveau 2 : Répondant aux Incidents** : Examine les incidents de sécurité prioritaires, analyse les indicateurs de compromission, et détermine la portée de l'attaque.
- **Niveau 3 : Chasseur de Menaces** : Ces analystes expérimentés gèrent les incidents majeurs, supervisent les évaluations de vulnérabilité, et identifient de manière proactive les menaces et vulnérabilités potentielles.
- **SOC Manager** : Supervise l'équipe des opérations de sécurité, apporte un soutien technique si nécessaire, et veille à la gestion de l'équipe, aux aspects

financiers du SOC, aux audits de sécurité, et communique avec les hauts responsables de la sécurité.

5 Conclusion

En conclusion, nous avons abordé les menaces potentielles pour l'architecture existante et ses limites en matière de sécurité. Nous avons également jeté les bases pour comprendre les composantes essentielles d'un SOC bien structuré. Le chapitre suivant s'appuiera sur ces fondations en vous guidant à travers le processus de conception et d'implémentation d'une solution SOC.Nous y explorerons l'utilité des outils open-source pour un SOC, justifierons le choix des composants, approfondirons l'architecture d'un SOC sécurisé et évolutif, et fournirons des indications sur les stratégies de déploiement.

Chapitre3 : conception de la solution

1 Introduction :

Dans ce deuxième chapitre, nous abordons les bases de la conception et de la mise en œuvre d'une solution de Security Operation Center en utilisant des outils open source. Nous explorerons l'état de l'art, définirons chaque composant et son rôle, examinerons l'architecture d'un SOC sécurisé et évolutif, et proposerons des stratégies de déploiement.

2 L'utilisation des solutions open-source

Dans le paysage de la cybersécurité en constante évolution, rester en avance sur les menaces demeure un défi de taille. Les solutions de sécurité traditionnelles sont souvent coûteuses, ce qui laisse de nombreuses organisations, en particulier les petites et moyennes entreprises (PME), vulnérables aux cyberattaques. Cependant, les solutions open-source représentent une alternative convaincante, offrant des options économiques et personnalisables qui permettent aux organisations de renforcer leurs défenses en matière de sécurité.

2.1 Le rôle de l'open-source en cybersécurité

L'adoption des pratiques open-source a révolutionné divers domaines technologiques, le développement logiciel étant un exemple marquant. Dans le domaine des Centres d'Opérations de Sécurité (SOC), l'évolution a été différente. Bien que de nombreux analystes en sécurité ne disposent pas des compétences en programmation nécessaires pour développer leur propre infrastructure, les outils open-source offrent une solution viable. En tirant parti des outils de sécurité open-source, les organisations peuvent accéder à une multitude de ressources communautaires, facilitant ainsi la mise en place de solutions de sécurité robustes.

2.2 Avantages des outils open-source

Aujourd'hui, la fréquence et la sophistication des cyberattaques continuent d'augmenter. Les organisations de toutes tailles sont confrontées à la menace de violations de données, rendant les mesures de sécurité résilientes indispensables. Les outils open-source offrent plusieurs avantages :

• **Rentabilité** : Les solutions open-source éliminent les frais de licence élevés, permettant aux organisations d'allouer les ressources de manière plus efficace.

• **Personnalisation** : Les organisations peuvent adapter les outils open-source pour répondre à leurs besoins de sécurité uniques et à leur environnement d'infrastructure.

• **Transparence** : La nature open-source de ces outils favorise la transparence, permettant l'examen par la communauté et des efforts d'amélioration collaborative.

• **Support communautaire** : La communauté dynamique qui entoure les projets open-source fournit un partage de connaissances et un soutien précieux, renforçant ainsi les capacités de sécurité.

2.3 Outils SOC open-source courants

Une large gamme d'outils open-source couvre divers aspects des opérations d'un SOC, incluant :

• **Agrégation, recherche et visualisation**

• **Gestion des problèmes et des incidents**

• **Évaluation des vulnérabilités**

• **Tests de pénétration**

• **Surveillance du réseau**

• **Détection et réponse des points de terminaison (EDR)**

Ces outils permettent aux organisations de mettre en place des cadres de sécurité robustes, mais leur mise en œuvre et leur maintenance nécessitent une expertise technique.

2.4 Exigences et problèmes de compatibilité des outils open-source dans un SOC

Bien que les outils open-source offrent des avantages significatifs, ils présentent également certains défis et considérations, notamment en termes d'exigences et de compatibilité.

2.4.1 Exigences techniques

La mise en œuvre des outils open-source dans un SOC nécessite généralement des prérequis techniques spécifiques :

• **Infrastructure** : Une infrastructure matérielle et réseau adéquate pour soutenir le déploiement et le fonctionnement des outils de sécurité open-source.

• **Expertise technique** : Un personnel qualifié ayant les connaissances nécessaires pour installer, configurer et maintenir les solutions open-source.

• **Formation** : Une formation continue pour se maintenir à jour avec les derniers développements et les meilleures pratiques en matière d'outils de sécurité open-source.

2.4.2 Problèmes de compatibilité

La compatibilité peut être une préoccupation majeure lors de l'intégration des outils open-source dans un SOC :

• **Interopérabilité** : Assurer que les différents outils open-source peuvent communiquer efficacement et travailler ensemble au sein de l'environnement SOC.

• **Verrouillage fournisseur** : Bien que les outils open-source atténuent le verrouillage fournisseur, les dépendances vis-à-vis de technologies ou de versions spécifiques peuvent créer des scénarios de verrouillage indirect.

• **Intégration avec les systèmes existants** : La compatibilité avec l'infrastructure de sécurité existante et les autres systèmes d'entreprise doit être évaluée pour garantir une intégration fluide.

• **Mises à jour régulières et patchs** : Maintenir les outils à jour avec les derniers patchs de sécurité et mises à jour pour atténuer les vulnérabilités et les problèmes de compatibilité.

En abordant ces exigences et problèmes de compatibilité, les organisations peuvent tirer parti des outils open-source et réduire les risques, rendant ainsi leur Centre d'Opérations de Sécurité plus efficace et résilient.

3 Outils et composants open-source pour la solution SOC

Le Security Operations Center(SoC)est une solution complète, prête à être déployée, conçue pour rationaliser et améliorer les opérations de cybersécurité. Au cœur de SOC se trouve l'utilisation stratégique des outils open-source, qui sont essentiels pour créer un cadre de sécurité efficace, personnalisable et robuste. Le choix de ces outils est crucial, car ils offrent la possibilité de s'adapter aux besoins spécifiques en matière de sécurité.

La solution SOC comprend :

• Plateforme de détection et de réponse des points de terminaison (EDR) : Surveille et répond aux menaces sur les points de terminaison individuels.

• Plateforme de gestion des informations et des événements de sécurité (SIEM) : Agrège et analyse les données des événements de sécurité provenant de l'ensemble de l'organisation.

• Plateforme de gestion des incidents et des cas : Gère et coordonne les réponses aux incidents de sécurité et les cas.

• Plateforme d'orchestration, d'automatisation et de réponse en matière de sécurité (SOAR) : Automatise les opérations de sécurité, la réponse aux incidents et la gestion des menaces.

• Plateforme d'automatisation des flux de travail : Rationalise et automatise les différentes tâches et processus de sécurité.

• Plateforme de renseignement sur les menaces cybernétiques : Collecte et analyse les données sur les menaces pour fournir des informations exploitables.

• Solution VPN virtuelle hébergée dans le cloud : Sécurise les connexions à distance au réseau de l'organisation.

3.1 Wazuh: Endpoint Detection and Response

Figure 8:wazuh logo

Wazuh offre une surveillance de la sécurité et une protection des actifs informatiques grâce à ses capacités de gestion des informations et des événements de sécurité (SIEM) et de détection et de réponse étendues (XDR). Les cas d'utilisation de Wazuh sont conçus pour protéger les actifs numériques et améliorer la posture de cybersécurité d'une organisation.

Ces cas d'utilisation incluent la surveillance de l'intégrité des fichiers (File Integrity Monitoring, FIM) pour garantir l'intégrité des fichiers critiques, l'évaluation de la configuration de sécurité (Security Configuration Assessment, SCA) pour renforcer les configurations système contre les menaces potentielles, la détection des vulnérabilités pour identifier les faiblesses avant qu'elles ne soient exploitées, et bien d'autres. Explorez les cas d'utilisation et les capacités ci-dessous.

3.1.1 Fonctionnalités de Wazuh

Il existe plusieurs raisons justifiant le choix de Wazuh par rapport à d'autres EDR qui pourraient potentiellement remplir le même rôle, mais voici les caractéristiques qui en font un bon choix :

• **Détection des menaces en temps réel** : Chaque serveur Wazuh est directement connecté à la base de données MITRE ATTACK, fournissant des mises à jour en temps réel des menaces identifiées au sein de la communauté d'utilisateurs de Wazuh.

• **Combinaison des fonctions XDR et SIEM :** La combinaison des fonctions XDR et SIEM fait de Wazuh une solution complète pour la gestion proactive des menaces de sécurité informatique.

• **Sécurité complète des points de terminaison :** Les agents Wazuh sont disponibles pour une gamme de systèmes d'exploitation de points de terminaison, y compris Microsoft Windows, Apple MacOS, Linux, Solaris.

• **Options de déploiement** : Pour mieux correspondre à l'environnement moderne basé sur des microservices, Wazuh propose plusieurs options de conteneurs pour le déploiement. Des conteneurs Wazuh Kubernetes, Puppet, Ansible et Docker sont disponibles

3.1.2 Cas d'utilisation de Wazuh

Wazuh propose une variété de cas d'utilisation qui peuvent améliorer la posture de sécurité d'une organisation. Voici quelques cas d'utilisation intéressants :

• **Évaluation de la configuration** : Wazuh surveille les paramètres de configuration des points de terminaison pour garantir leur conformité avec différents standards et politiques de sécurité.

• **Détection de logiciels malveillants** : Wazuh détecte les activités malveillantes et les indicateurs de compromission sur différents points de terminaison à la suite d'infections par des malwares ou d'attaques informatiques.

• **Surveillance de l'intégrité des fichiers** : Wazuh surveille le système de fichiers, identifiant les modifications dans le contenu et les autorisations des fichiers système.

• **Détection des vulnérabilités** : Les agents Wazuh récupèrent la liste des logiciels installés sur le système pour identifier les logiciels vulnérables connus.

• **Analyse des données de journaux** : Les agents Wazuh collectent les journaux du système d'exploitation et des applications, puis les utilisent pour des analyses basées sur des règles et leur stockage. Les règles de Wazuh détectent les erreurs d'application ou de système, les mauvaises configurations, les activités malveillantes et les violations des politiques.

3.1.3 Architecture et composants de Wazuh

Au cœur de Wazuh, on trouve trois composants fondamentaux. Ces trois composants travaillent ensemble pour corréler, détecter et atténuer les menaces afin de protéger les actifs numériques.

• **Tableau de bord Wazuh** : Le composant Tableau de bord Wazuh est utilisé pour gérer la configuration de Wazuh, surveiller et visualiser les menaces.

• **Serveur Wazuh** : Le composant Serveur Wazuh analyse les données reçues des agents, les traite et recherche des indicateurs de compromission à l'aide de renseignements sur les menaces.

• **Indexeur Wazuh** : L'Indexeur Wazuh est responsable du stockage et de l'indexation des alertes générées par le serveur Wazuh.

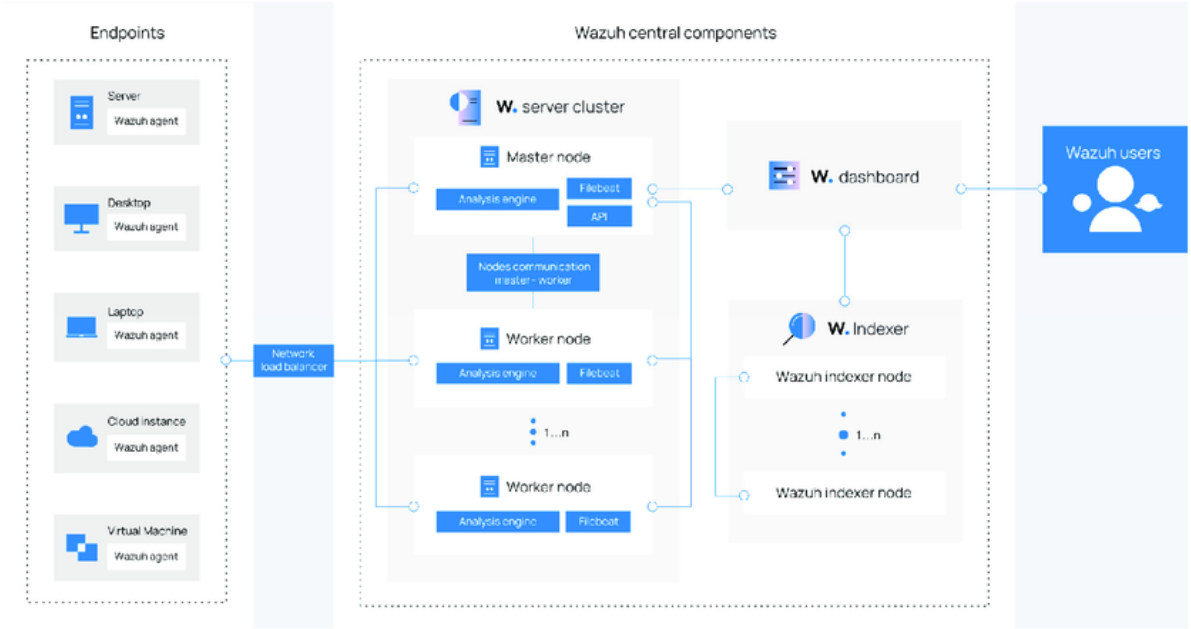

Figure 9:les composants de wazuh

Le diagramme ci-dessus représente les composants de Wazuh et le flux de données réel, depuis la collecte des données des points de terminaison jusqu'à leur transmission, leur indexation, leur analyse et la détection des menaces.

3.1.4 L'utilité de Wazuh dans la solution SOC

Je suis le développement du projet Wazuh depuis environ deux ans maintenant, et je peux affirmer qu'il a un bel avenir. Jusqu'à présent, il y a plus de 150 accords de partenariat [14] entre Wazuh et diverses entreprises liées à l'informatique ou à la cybersécurité.

Pour la solution SOC, Wazuh sera utilisé comme principal outil de détection et de réponse des points de terminaison (EDR). Des agents seront déployés sur chaque point

de terminaison pour collecter les journaux et les alertes, qui seront envoyées à Wazuh. Ces alertes seront ensuite traitées et, si nécessaire, utilisées pour générer des alertes et des cas critiques dans TheHive.

3.2 TheHive 5 : Réponse aux incidents et gestion des cas

Figure 10:The Hive logo

TheHive est une plateforme open-source de réponse aux incidents de sécurité (IR) et de gestion des cas (CM) qui facilite la collaboration et optimise le processus de gestion des incidents grâce à une gestion des cas hautement flexible.

3.2.1 Fonctionnalités et cas d'utilisation de TheHive 5

TheHive offre un ensemble robuste de fonctionnalités spécifiquement conçues pour simplifier les flux de travail de réponse aux incidents. Ces fonctionnalités améliorent la collaboration entre les professionnels de la sécurité et les aident à enquêter efficacement et à atténuer les menaces de sécurité.

Collaboration en temps réel : Plusieurs analystes peuvent collaborer simultanément avec des mises à jour en temps réel sur les cas, les tâches, les observables et les indicateurs de compromission (IOCs).

Gestion efficace des tâches : Attribution efficace des tâches, avec des aperçus et des importations provenant de diverses sources telles que les rapports par e-mail, les fournisseurs de CTI (Cyber Threat Intelligence) et les SIEM.

Gestion des preuves : Les analystes de sécurité peuvent suivre l'avancement des cas, joindre des preuves, ajouter des étiquettes et importer des archives ZIP contenant des données suspectes en toute sécurité.

Gestion des observables : Ajouter et gérer facilement des observables tels que des adresses IP, des hachages et des domaines pour une réponse plus rapide et une meilleure efficacité.

Intégration avec les plateformes de Cyber Threat Intelligence : Compatible avec plusieurs plateformes CTI comme Cortex, permettant d'accélérer les enquêtes, de contenir les menaces et d'enrichir les informations sur les menaces.

3.2.2 Architecture et composants de TheHive

Comme Wazuh, TheHive repose sur trois composants principaux : l'application TheHive, la base de données et le moteur d'indexation, et enfin un stockage de fichiers (local ou cloud).

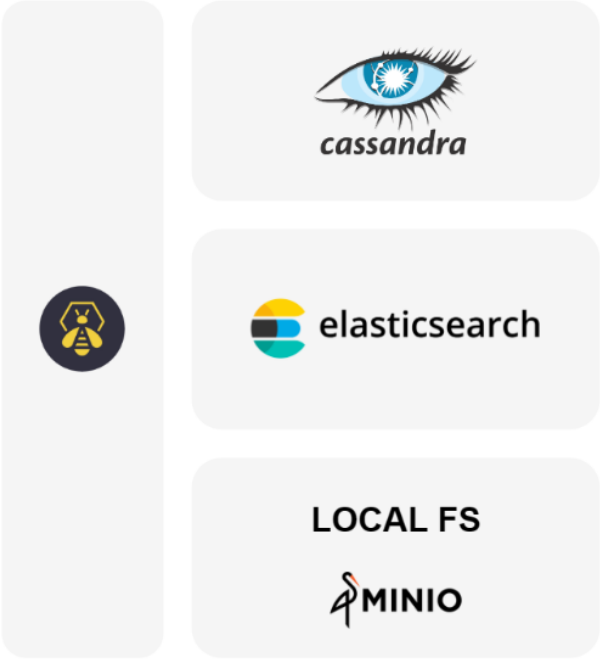

Figure 11:architecture et composants TheHive

Comme le montre l'illustration ci-dessus, l'architecture de TheHive est modulaire, ce qui permet à chaque composant d'être déployé en tant que nœud unique ou dans une configuration clusterisée. Cette flexibilité permet de créer des architectures clusterisées complexes pour des performances optimales et une scalabilité accrue.

Apache Cassandra : Cassandra est une base de données NoSQL hautement fiable, connue pour sa capacité à gérer de grands volumes de données. Cela garantit que TheHive peut gérer et stocker efficacement d'énormes quantités de données de sécurité tout en maintenant des temps d'accès rapides.

Elasticsearch : Elasticsearch est un moteur de recherche et d'analyse open-source très évolutif, capable de traiter des requêtes de recherche complexes et des analyses de données en temps réel. TheHive peut rapidement indexer et rechercher de grands ensembles de données, fournissant ainsi aux analystes de sécurité des informations rapides et précises nécessaires pour une réponse efficace aux incidents et pour la chasse aux menaces.

Solution de stockage de fichiers : TheHive 5 offre des options flexibles pour le stockage de fichiers.

> **Stockage local de fichiers** : Pour les installations locales, TheHive peut utiliser le système de fichiers local du serveur hébergeant l'application. Cette simplicité assure une gestion facile et un accès direct aux fichiers.
>
> **Stockage de fichiers basé sur le cloud** : Pour des installations plus complexes, TheHive 5 prend en charge des solutions de stockage cloud comme MINIO, un système de stockage objet distribué haute performance qui permet l'intégration avec les services de stockage cloud existants.

Ces composants garantissent collectivement que TheHive 5 fournit une plateforme de gestion des incidents et des cas efficace et évolutive, capable de répondre aux exigences des environnements de cybersécurité modernes.

3.2.3 L'utilité de TheHive dans la solution SOCaaC

Lorsqu'on cherche une plateforme de gestion des incidents et des cas, de nombreux outils sont disponibles. Certains outils offrent même plus de fonctionnalités que TheHive, mais dans ce projet, l'accent a été mis sur l'automatisation et l'interopérabilité entre les différents outils.

TheHive prend en charge de nombreuses intégrations prêtes à l'emploi, comme Cortex. Il s'agit d'une solution idéale où vous pouvez avoir vos cas et alertes et obtenir des rapports d'analyse prêts sur divers observables tels que des adresses IP, des URL, des hachages et des domaines, générés par Cortex.

TheHive contiendra les alertes collectées à partir de Wazuh et les classera en fonction de leur gravité. Ainsi, lorsqu'une alerte critique survient, un nouveau cas sera automatiquement créé.

3.3 Cortex : Orchestration, Automatisation et Réponse en Sécurité

Figure 12:logo cortex

Cortex est un outil puissant, open-source, conçu pour améliorer et automatiser l'analyse des observables et des indicateurs de compromission (IOCs). Il s'intègre de manière transparente avec TheHive pour fournir une intelligence sur les menaces et faciliter la gestion des incidents.

3.3.1 Fonctionnalités de Cortex

Ce qui distingue Cortex, c'est sa capacité à s'intégrer avec TheHive, où il peut également utiliser les répondants de Cortex pour effectuer des actions spécifiques sur les alertes, les cas, les tâches et les observables collectés au cours de l'investigation : envoyer un e-mail aux analystes SOC, bloquer une adresse IP, notifier les membres de l'équipe qu'une alerte est critique et nécessite une intervention urgente, et bien plus encore.

Efficacité : Cortex automatise les tâches d'analyse répétitives et chronophages, permettant aux analystes de se concentrer sur les aspects plus critiques de la réponse aux incidents.

Enrichissement : Cortex fournit des données enrichies et du contexte, améliorant la qualité et la rapidité de la détection et de l'analyse des menaces.

Automatisation : Cortex facilite les réponses automatisées aux incidents de sécurité, réduisant ainsi le temps de confinement et de remédiation.

3.3.2 Cas d'Utilisation de Cortex

Grâce à son efficacité, son enrichissement et son automatisation, Cortex dispose de nombreux cas d'utilisation qui aident principalement à réduire le temps d'investigation en automatisant de nombreuses tâches répétitives que les analystes SOC doivent effectuer manuellement.

Analyse Automatisée des Menaces : Utilisez Cortex pour automatiser l'analyse des observables, réduisant ainsi l'effort manuel requis de la part des analystes en sécurité. Rassemblez rapidement des informations et du contexte sur les menaces potentielles grâce à divers analyseurs intégrés.

Réponse aux Incidents : Utilisez Cortex pour effectuer des actions de réponse automatisées, comme bloquer des adresses IP ou isoler des systèmes compromis, afin de contenir les menaces. Il peut également être intégré à TheHive pour enrichir les données des incidents et accélérer le processus d'investigation.

Enrichissement de l'Intelligence sur les Menaces : Utilisez Cortex pour enrichir les données d'intelligence sur les menaces en utilisant plusieurs sources et analyseurs afin d'améliorer la qualité et la pertinence des flux et rapports d'intelligence sur les menaces.

Gestion des Vulnérabilités : Utilisez Cortex pour analyser les vulnérabilités et évaluer leur impact à l'aide d'analyseurs dédiés, en automatisant le processus d'identification et de hiérarchisation des vulnérabilités au sein de l'environnement de l'organisation.

Automatisation de la Sécurité et Intégration : Utilisez et intégrez Cortex avec les outils de sécurité existants via son API pour automatiser les tâches répétitives et chronophages en matière de sécurité, améliorant ainsi l'efficacité opérationnelle globale en réduisant la nécessité d'interventions manuelles dans les processus routiniers.

3.3.3 Architecture et Composants de Cortex

Cortex repose sur trois composants principaux : le backend avec l'API, Elasticsearch comme moteur de base de données et d'indexation, et enfin le système de stockage de fichiers.

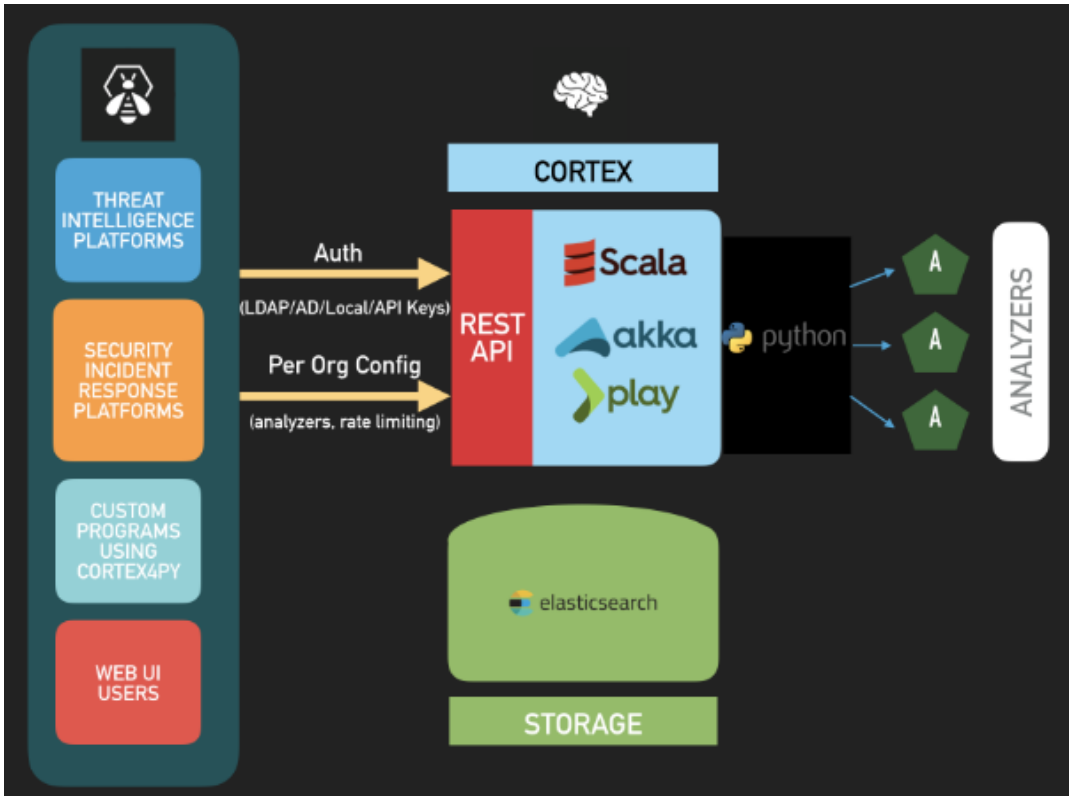

Figure 13:architecture et composants du cortex

Comme montré dans la figure ci-dessus, Cortex est écrit en Scala. Le frontend utilise AngularJS. Son API REST lui permet d'être évolutif horizontalement, et les analyseurs fournis sont écrits en Python, bien que des analyseurs supplémentaires puissent être écrits en Python ou tout autre langage compatible avec Linux.

Plateforme Cortex : La plateforme Cortex est composée du frontend utilisant AngularJS, tandis que le backend gère l'API et reçoit les données provenant de différentes sources telles que TheHive ou d'autres plateformes CTI, voire des applications personnalisées utilisant l'API avec des requêtes HTTP.

Elasticsearch : Elasticsearch est un moteur de recherche et d'analyse open-source hautement évolutif, capable de gérer des requêtes complexes et d'analyser des données en temps réel. Comme TheHive, Cortex utilise Elasticsearch pour interroger et indexer rapidement les données.

Solution de Stockage de Fichiers : Cortex utilise le système de fichiers comme solution de stockage, mais propose également deux types de déploiement pour les analyseurs et répondants (également appelés Neurons).

Exécution des Neurons Localement : Lorsque les Neurons sont exécutés localement, ils sont installés directement sur le système Cortex. Cependant, cette approche présente des inconvénients, car elle nécessite des applications et des bibliothèques spécifiques, ce qui peut entraîner des conflits.

Exécution des Neurons dans des Conteneurs (Recommandé) : Les images Docker des Neurons éliminent les préoccupations liées aux applications ou bibliothèques nécessaires, simplifiant leur utilisation car Cortex les télécharge et les exécute dans des conteneurs isolés, assurant ainsi l'intégrité et la sécurité.

En tirant parti de ces fonctionnalités et cas d'utilisation clés, la plateforme SOAR Cortex de StrangeBee aide les organisations à améliorer leurs opérations de cybersécurité, à automatiser l'analyse et la réponse aux menaces, et à améliorer l'efficacité et l'efficience globales de leurs équipes de sécurité.

3.3.4 L'Utilité de Cortex dans la Solution SOC

Cortex est l'un des meilleurs outils SOAR qui propose une intégration prête à l'emploi avec TheHive, conçu spécifiquement pour être utilisé avec TheHive.

Dans ce contexte, Cortex sera utilisé pour automatiser l'analyse des observables présents dans les alertes de TheHive. Il analysera automatiquement chaque observable et ajoutera ensuite le rapport d'analyse à l'alerte correspondante. L'objectif est de gagner du temps sur l'analyse initiale, permettant ainsi aux analystes d'investigation de se concentrer sur des tâches plus critiques.

3.4 Plateforme de Partage d'Informations sur les Menaces

Figure 14:Logo MISP

MISP (Malware Information Sharing Platform & Threat Sharing) est une plateforme open-source dédiée au partage, à la collecte et à l'analyse des informations sur les menaces. Elle est conçue pour faciliter l'échange d'indicateurs de compromission (IOCs), d'analyses de menaces et d'autres données de sécurité entre les organisations, afin d'améliorer la détection et la réponse aux incidents de sécurité.

3.4.1 Fonctionnalités de MISP

MISP offre une gamme de fonctionnalités permettant de centraliser et d'organiser les informations relatives aux menaces de manière structurée et sécurisée, tout en facilitant leur partage avec d'autres entités de sécurité.

Partage de données : MISP facilite le partage d'indicateurs de compromission (IOCs) entre les organisations et les communautés de sécurité pour améliorer la détection proactive des menaces.

Enrichissement des données : MISP permet d'enrichir les données d'intelligence sur les menaces en les croisant avec d'autres sources d'information, ce qui améliore la qualité des alertes et de l'analyse.

Automatisation : MISP permet l'automatisation de la collecte et du traitement des données de sécurité en intégrant des flux de travail d'analyse d'IOCs et d'enrichissement des informations.

Flexibilité : MISP est hautement configurable et peut être utilisé pour différents types d'échanges de données, comme les alertes de sécurité, les listes de signatures malveillantes, les informations sur les vulnérabilités, etc.

3.4.2 Cas d'Utilisation de MISP

Grâce à sa capacité de centraliser et de partager des informations sur les menaces, MISP propose plusieurs cas d'utilisation permettant aux équipes de sécurité de mieux se préparer face aux attaques et d'améliorer leurs processus de détection et de réponse

Partage d'IOCs : MISP permet aux organisations de partager des indicateurs de compromission (IOCs) en temps réel, ce qui facilite l'identification rapide des menaces et la mise en place de mesures de réponse adéquates.

Enrichissement des alertes de sécurité : MISP permet d'enrichir les alertes de sécurité avec des informations supplémentaires, comme des contextes de menace, des vecteurs d'attaque et des informations sur les acteurs malveillants, ce qui permet aux analystes SOC d'avoir une vue plus complète des incidents.

Gestion des incidents de sécurité : En intégrant MISP à d'autres outils de gestion des incidents (comme TheHive), les données recueillies sur les menaces peuvent être utilisées pour alimenter les processus de réponse aux incidents, en automatisant la création d'alertes et en accélérant la gestion des tickets.

Analyse des tendances de menaces : Grâce à la collecte et l'analyse centralisée des données sur les menaces, MISP permet aux équipes de sécurité de suivre l'évolution des tendances et des comportements des attaquants, offrant ainsi une meilleure préparation aux menaces émergentes.

Détection des attaques ciblées : MISP permet de détecter les attaques ciblées en croisant des indicateurs provenant de différentes sources et en identifiant des patterns communs.

3.4.3 Architecture et Composants de MISP

MISP repose sur une architecture modulaire, avec plusieurs composants pour gérer l'ingestion des données, leur analyse, et leur partage avec d'autres systèmes.

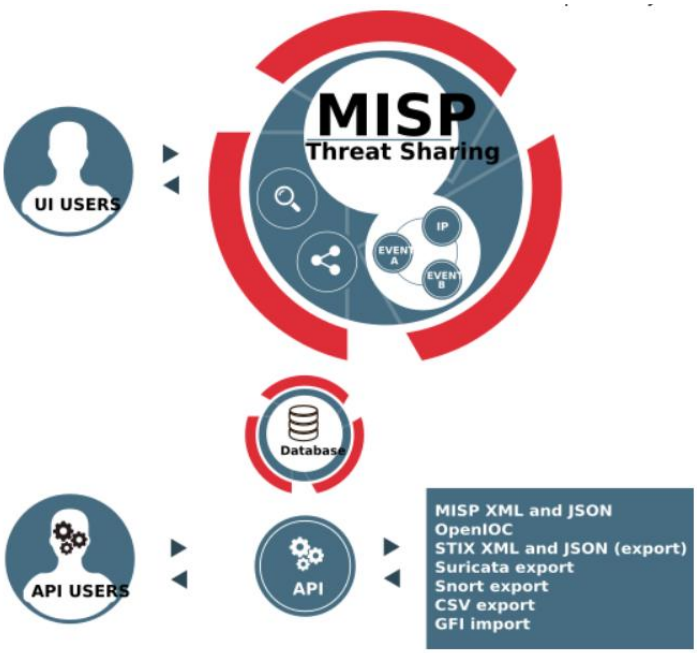

Figure 15:Architecture et Composants de MISP

Plateforme MISP : La plateforme MISP est composée du frontend (interface utilisateur) et du backend, qui gère la base de données, les flux de travail et l'intégration avec d'autres systèmes via des API RESTful.

Base de données : MISP utilise une base de données MySQL ou MariaDB pour stocker les données relatives aux menaces et aux incidents. Les IOCs, les événements de sécurité et les attributs associés sont organisés de manière relationnelle pour faciliter l'analyse.

API RESTful : MISP offre une API RESTful permettant d'intégrer des flux de données provenant de sources externes, comme des outils de sécurité, des systèmes de

renseignement sur les menaces (Threat Intelligence), et des plateformes de gestion des incidents.

Module d'Enrichissement : MISP inclut des modules d'enrichissement des IOCs permettant de croiser les informations collectées avec des sources externes telles que des bases de données de vulnérabilités, des renseignements sur des malwares ou des acteurs de menaces.

Gestion des Utilisateurs et des Permissions : MISP permet de gérer les utilisateurs et de définir des rôles et des permissions pour restreindre l'accès aux données sensibles et contrôler les actions que chaque utilisateur peut effectuer sur la plateforme.

3.4.4 L'Utilité de MISP dans la Solution SOC

Dans le cadre d'une solution SOC (SOC), MISP joue un rôle crucial en fournissant une base solide pour le partage et l'enrichissement des informations sur les menaces. En intégrant MISP avec d'autres outils de gestion des incidents comme TheHive et des systèmes de détection d'intrusion comme Wazuh, il devient possible de centraliser et d'enrichir les données relatives aux menaces, ce qui améliore l'efficacité des analystes SOC dans leurs investigations

Dans ce contexte, MISP permet :

Centralisation des informations sur les menaces : En agrégeant les IOCs de différentes sources et en les analysant, MISP fournit une vue d'ensemble des menaces.

Partage d'informations en temps réel : Les données partagées entre les organisations, par exemple les indicateurs de compromission, permettent une réaction rapide et une meilleure anticipation des attaques.

Intégration avec d'autres outils de sécurité : MISP peut être intégré avec TheHive pour enrichir automatiquement les alertes et accélérer la gestion des incidents. Il peut aussi alimenter Wazuh pour la détection d'attaques en se basant sur des IOCs partagés.

En résumé, MISP permet d'améliorer la visibilité des menaces, d'automatiser l'enrichissement des alertes et d'optimiser la réponse aux incidents, en offrant une plateforme robuste pour le partage et l'analyse des données de sécurité.

4 Conclusion

Dans ce chapitre, nous avons exploré l'utilisation de solutions open-source dans le cadre d'une solution SOC (SOC). Nous avons examiné les composants clés de cette solution SOC, notamment Wazuh pour la détection et la réponse aux incidents sur les endpoints, TheHive pour la gestion des incidents et des cas, Cortex pour l'orchestration, l'automatisation et la réponse en matière de sécurité, ainsi que MISP. Chaque composant a été analysé en termes de fonctionnalités, de cas d'utilisation,

d'architecture et de son utilité au sein de la solution SOC, mettant en évidence une approche globale pour améliorer les opérations de sécurité grâce aux technologies open-source.

Dans le prochain chapitre, nous entrerons dans les détails de la mise en œuvre complète de la solution SOC et examinerons comment l'installation et la configuration de chaque composant se déroulent.

Chapitre4 : implimentation de la solution

1 Introduction

Dans ce quatrième chapitre, nous traiterons de la mise en œuvre de la solution SOC. Nous débuterons par une présentation de la solution SOC et de ses différents composants, en détaillant le processus de déploiement et d'installation des outils utilisés. Ensuite, nous décrirons en détail l'intégration des solutions de sécurité déployées, à savoir Wazuh, The Hive, MISP et Cortex. Nous mettrons en avant les configurations établies et les interactions mises en place pour garantir une synergie optimale entre ces outils. Enfin, nous réaliserons une série de tests visant à valider cette intégration, en vérifiant le bon fonctionnement des flux de données, la communication entre les composants et l'efficacité des fonctionnalités offertes.

2 Aperçu de l'architecture SOC

SOC est une solution de cybersécurité robuste et automatisée qui garantit une collecte de données sécurisée, une détection efficace des menaces et une gestion structurée des incidents, offrant ainsi une infrastructure réseau évolutive et résiliente.

Comme discuté dans les chapitres précédents, la solution SOCaaC est conçue pour automatiser les tâches suivantes :

• Collecter les journaux provenant de divers points de terminaison afin de générer des alertes et détecter d'éventuelles anomalies.

• Automatiser la création d'alertes et de cas pour les incidents critiques sur une plateforme dédiée à la gestion des incidents et des cas.

• Exécuter des analyseurs sur les observables des alertes.

• Intégrer les résultats de tous les analyseurs dans les alertes pour assister les analystes de sécurité en automatisant l'analyse des menaces.

• Rester à jour grâce à une plateforme dédiée à l'intelligence sur les menaces informatiques.

• Faciliter l'ajustement fin et les efforts de détection.

2.1 Architecture de haut niveau

Comme le montre la figure ci-dessous, l'architecture intègre plusieurs outils open-source pour fournir un cadre de cybersécurité automatisé, évolutif et sécurisé.

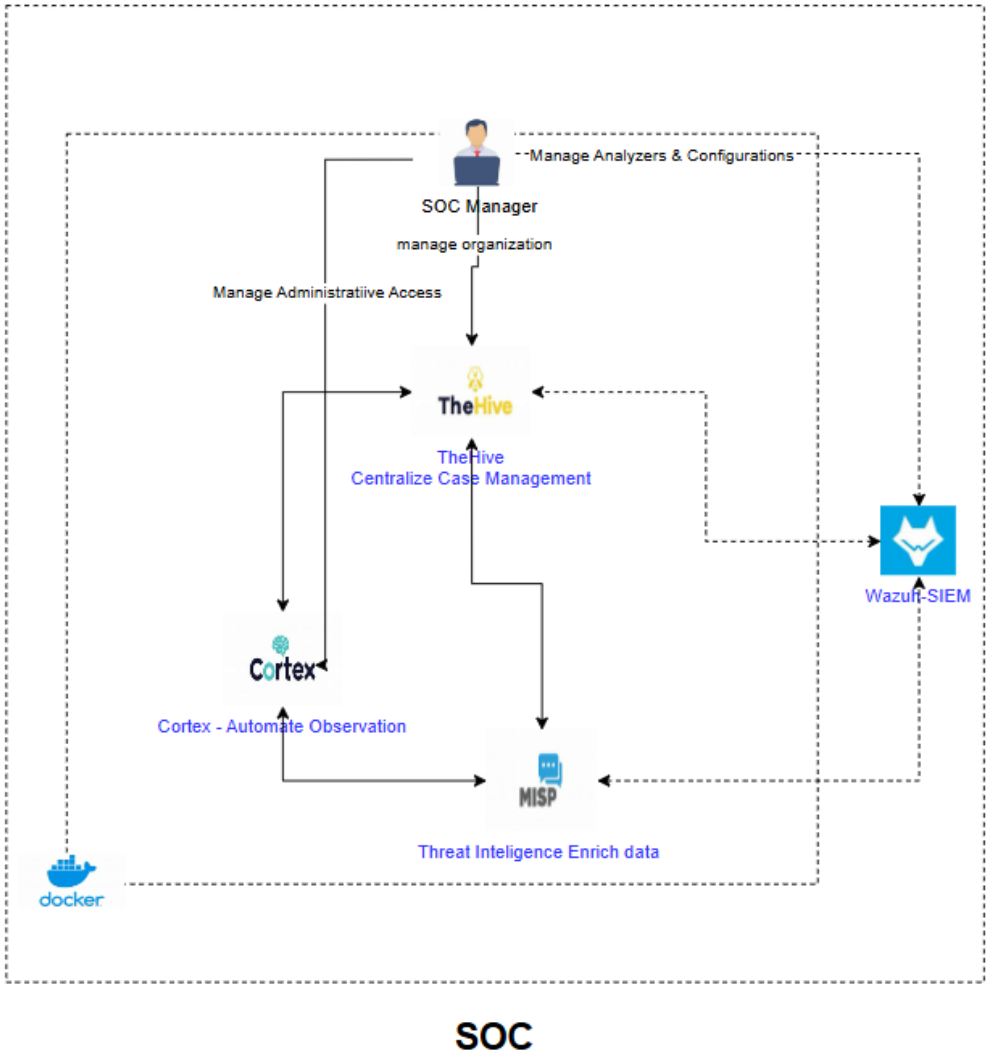

Figure 16:architecture de SOC

Cette architecture sera déployée dans la zone DMZ, où se trouvent des services tels qu'Apache et DNS, afin de renforcer la sécurité de l'infrastructure. Elle sera configurée dans un environnement Docker Compose, garantissant ainsi une gestion centralisée et une évolutivité optimale de l'ensemble des composants

2.1.1 Wazuh

Rôle : Détection et Réponse aux Points d'Extrémité (EDR)

Fonctionnalités : Wazuh collecte les journaux et les alertes provenant de divers points d'extrémité, détecte les menaces et offre une visibilité sur les événements de sécurité au niveau des terminaux.

2.1.2 TheHive

Rôle : Gestion des Incidents et des Cas de Sécurité

Fonctionnalités : TheHive permet de gérer les incidents et les cas de sécurité, facilitant ainsi le travail des analystes en cybersécurité pour enquêter et répondre efficacement aux menaces. Il regroupe les mêmes alertes provenant de Wazuh tout en permettant une réponse collaborative aux incidents.

2.1.3 Cortex

Rôle : Orchestration, Automatisation et Réponse en Cybersécurité (SOAR)

Fonctionnalités : Cortex exécute des analyseurs sur des observables, automatise l'enrichissement des renseignements sur les menaces et coordonne des réponses automatisées aux incidents de sécurité.

2.1.4 MISP

Rôle : Plateforme de Partage de Renseignements sur les Menaces (Threat Intelligence Sharing)

Fonctionnalités : MISP (Malware Information Sharing Platform) facilite le partage, la collecte et l'analyse de renseignements sur les menaces. Il permet aux organisations de collaborer en échangeant des informations structurées sur les indicateurs de compromission (IOC) et d'autres données de sécurité, contribuant ainsi à une détection et une prévention plus efficaces des menaces.

2.2 Interaction des Composants et Flux de Données

Dans l'architecture SOC (Security Operations Center), les flux de données et les interactions entre les composants sont conçus pour garantir une intégration fluide et une automatisation efficace. Cette section explique le flux de données entre tous les composants et décrit leur fonctionnement ensemble pour détecter et atténuer une cyberattaque.

2.2.1 Flux de Détection et de Création d'Alerte

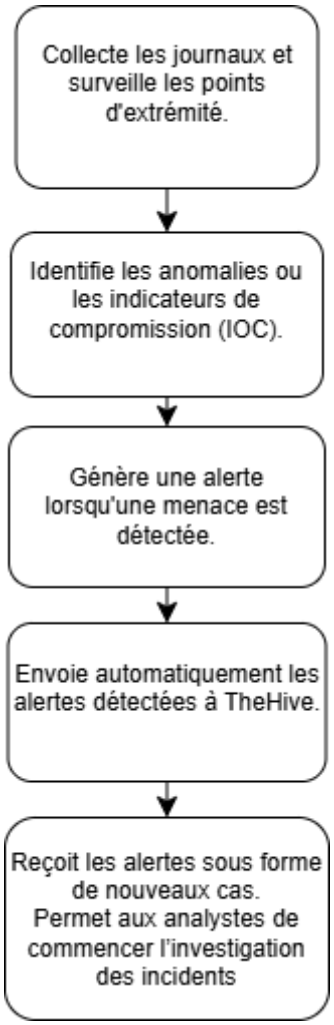

Figure 17:Flux de detection et de creation d'Alerte

Wazuh collecte les journaux et surveille les points d'extrémité pour détecter les anomalies ou les indicateurs de compromission (IOC). Lorsqu'une menace est identifiée, Wazuh génère une alerte et l'envoie automatiquement à TheHive. Ce dernier crée un nouveau cas pour chaque alerte reçue, permettant aux analystes de commencer l'investigation et de prendre les mesures nécessaires.

2.2.2 Flux d'Extraction et d'Enrichissement des Observables

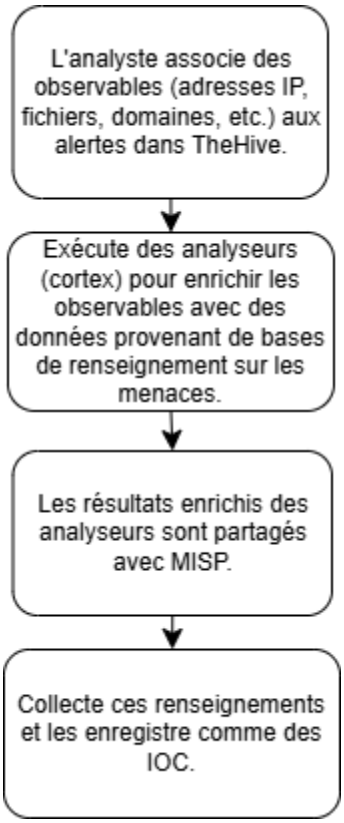

Figure 18:Flux d'Extraction et d'Enrichissement des Observables

Dans TheHive, l'analyste associe des observables (adresses IP, fichiers, domaines, etc.) aux alertes pour mener l'investigation. Ces observables sont ensuite envoyés à Cortex, qui les enrichit avec des données provenant de diverses bases de renseignement sur les menaces. Les résultats enrichis sont partagés avec MISP, où les IOC (indicateurs de compromission) sont collectés et centralisés.

2.2.3 Flux d'Analyse et de Mitigation des Menaces

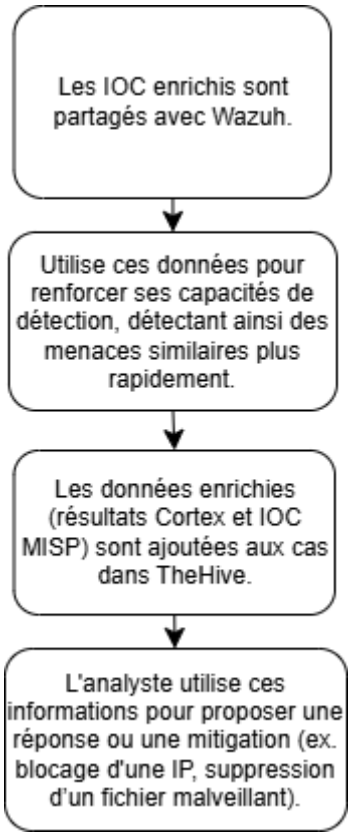

Figure 19:Flux d'Analyse et de Mitigation des Menaces

Les IOC enrichis sont également partagés avec Wazuh, qui utilise ces informations pour améliorer ses capacités de détection, permettant ainsi d'identifier plus rapidement des menaces similaires. Les données enrichies de Cortex et MISP sont ensuite renvoyées à TheHive, où l'analyste les utilise pour proposer des réponses ou des actions de mitigation, comme le blocage d'adresses IP ou la suppression de fichiers malveillants.

2.3 Stratégie de Déploiement du SOC

La stratégie de déploiement du SO repose sur deux approches complémentaires : l'utilisation de Docker Compose pour l'orchestration des services et le déploiement de Wazuh sur une machine virtuelle Ubuntu. Chacune de ces approches présente des avantages spécifiques.

Docker Compose pour l'orchestration des services : Docker Compose permet de gérer plusieurs services en les regroupant dans un fichier de configuration unique. Cela facilite la mise en place, la gestion et la mise à jour des différents composants du SOCaaC, tels que Elasticsearch, Cortex, TheHive et MISP. Les principaux avantages de cette approche incluent :

Scalabilité : Il est facile d'ajouter ou de supprimer des services et d'adapter les ressources en fonction des besoins, ce qui permet une gestion dynamique de l'infrastructure.

Portabilité : Les services peuvent être déployés de manière identique sur différents environnements, qu'il s'agisse de serveurs locaux ou de cloud, garantissant une homogénéité de la configuration.

Isolation des services : Chaque service fonctionne dans son propre conteneur, ce qui réduit les risques de conflits de configuration et simplifie la gestion des dépendances.

Automatisation : Le fichier docker-compose.yml permet de déployer et configurer automatiquement les services avec une seule commande, ce qui simplifie les processus d'installation et de maintenance.

Wazuh sur une machine virtuelle Ubuntu : Le déploiement de Wazuh sur une machine virtuelle Ubuntu permet de bénéficier de la stabilité et de la sécurité d'un environnement dédié à la surveillance de la sécurité. Les avantages de cette approche incluent :

Séparation des rôles : En déployant Wazuh sur une machine virtuelle distincte, on isole les tâches de surveillance et de détection des menaces des autres services, ce qui améliore la sécurité et la gestion des ressources.

Contrôle total sur l'environnement : L'utilisation d'une machine virtuelle permet de configurer précisément les ressources allouées à Wazuh (mémoire, CPU, stockage), garantissant ainsi une performance optimale pour la détection des incidents.

Sécurité renforcée : Ubuntu, en tant que système d'exploitation réputé pour sa stabilité et sa sécurité, offre un environnement solide pour déployer Wazuh, avec des mises à jour régulières et une communauté active pour soutenir la sécurité de la machine.

Flexibilité de gestion : L'utilisation d'une machine virtuelle permet de gérer facilement l'environnement d'exploitation de Wazuh, tout en étant capable de l'intégrer dans des infrastructures plus larges, comme des environnements hybrides ou sur site.

Ainsi, cette combinaison permet d'optimiser la gestion et la sécurité du SOC en tirant parti des avantages de l'orchestration avec Docker Compose pour les services et d'un environnement dédié et sécurisé pour Wazuh sur Ubuntu.

2.3.1 Deployement de l’environement

Lors de la planification du déploiement de la solution SOC, il est essentiel de prendre en compte l'environnement dans lequel elle fonctionnera. L'environnement de déploiement peut varier considérablement en fonction des besoins et des ressources de l'organisation, incluant des configurations sur site, dans le cloud ou hybrides.

2.3.1.1 Déploiement sur site (On-Premises)

Le SOC est déployé sur les ressources internes de l'organisation, offrant ainsi un contrôle total sur le matériel et la sécurité des données. Dans cette approche, l'infrastructure physique est gérée en interne, mais l'utilisation d'un hyperviseur tel que VMware permet de virtualiser plusieurs machines, optimisant ainsi l'allocation des ressources et la gestion des systèmes. Cette solution combine les avantages du contrôle physique direct tout en bénéficiant de la flexibilité et de la simplicité de la gestion virtuelle des machines.

2.3.1.2 Déploiement dans le cloud

Le SOC utilise des services cloud pour offrir une scalabilité rapide, avec une gestion partagée de la sécurité entre l'organisation et le fournisseur de cloud

2.3.1.3 Déploiement hybride

Cette approche combine des environnements sur site et cloud, permettant d'optimiser la flexibilité et d'intégrer des systèmes hérités.

Dans ce contexte, nous nous concentrons sur le déploiement du SOC dans un environnement virtualisé, à l'aide d'un hyperviseur VMware. Cette configuration permet de déployer et de gérer plusieurs machines virtuelles sur une infrastructure physique interne, tout en assurant une allocation dynamique des ressources et une gestion centralisée et simplifiée des systèmes.

2.4 Stratégie d'Implémentation des Composants SOC

Lors du développement de la solution SOC, j'ai automatisé l'installation et le déploiement de chaque composant en créant un fichier .yml destiné à Docker Compose. Ce fichier permet de déployer tous les services nécessaires dans des conteneurs, tandis que l'installation de Wazuh est effectuée sur une machine virtuelle Ubuntu séparée. Ainsi, l'implémentation du SOC repose sur un fichier .yml unique pour orchestrer les services Docker, tandis que Wazuh est déployé et configuré indépendamment sur la machine virtuelle. Examinons maintenant la manière dont cette architecture fonctionne.

2.4.1 Présentation des Services Déployés via Docker Compose

2.4.1.1 1. TheHive

Nature de l'installation : Le service TheHive utilise l'image Docker officielle de TheHive (version 5.2) pour déployer l'application. Il dépend de plusieurs services, tels que Cassandra, Elasticsearch, MinIO et Cortex. Cette approche garantit que tous les services nécessaires à TheHive sont lancés de manière cohérente et intégrée. Le service est configuré pour redémarrer automatiquement sauf en cas d'arrêt explicite.

```
thehive:
  image: strangebee/thehive:5.2
  restart: unless-stopped
  depends_on:
    - cassandra
    - elasticsearch
    - minio
    - cortex.local
  mem_limit: 1500m
  ports:
    - "0.0.0.0:9000:9000"
  environment:
    - JVM_OPTS="-Xms1024M -Xmx1024M"
  command:
    - --secret
    - "lab123456789"
    - "--cql-hostnames"
    - "cassandra"
    - "--index-backend"
    - "elasticsearch"
    - "--es-hostnames"
    - "elasticsearch"
    - "--s3-endpoint"
    - "http://minio:9002"
    - "--s3-access-key"
    - "minioadmin"
    - "--s3-secret-key"
    - "minioadmin"
    - "--s3-use-path-access-style"
    - "--cortex-port"
    - "9001"
    - "--cortex-keys"
    - "WaNKzlS5Ec11ZvkftxhywkaUDtyNkW5q"
  volumes:
    - ./thehive/conf/application.conf:/etc/thehive/application.conf
  networks:
    - SOC_NET
```

Figure 20:Configuration du Service TheHive

2.4.1.2 Cassandra

Nature de l'installation : Cassandra est utilisé comme base de données NoSQL pour stocker les données de TheHive. Le service est configuré avec l'image officielle de

Cassandra (version 4) et est déployé dans un conteneur Docker. Comme TheHive, Cassandra est configuré pour redémarrer automatiquement en cas de problème. Les données sont conservées dans un volume persistant pour éviter la perte de données lors des redémarrages.

```
cassandra:
  image: 'cassandra:4'
  restart: unless-stopped
  ports:
    - "0.0.0.0:9042:9042"
  environment:
    - CASSANDRA_CLUSTER_NAME=TheHive
  volumes:
    - cassandradata:/var/lib/cassandra
  networks:
    - SOC_NET
```

Figure 21:Configuration du Service Cassandra

2.4.1.3 Elasticsearch

Nature de l'installation : Elasticsearch est utilisé comme moteur de recherche pour indexer et rechercher des données dans TheHive. Le service utilise l'image officielle d'Elasticsearch et est configuré pour fonctionner en mode single-node. Elasticsearch est également configuré pour ne pas activer la sécurité, ce qui est courant dans un environnement de développement.

```
elasticsearch:
  image: docker.elastic.co/elasticsearch/elasticsearch:7.17.9
  restart: unless-stopped
  mem_limit: 512m
  ports:
    - "0.0.0.0:9200:9200"
  environment:
    - discovery.type=single-node
    - xpack.security.enabled=false
    - cluster.name=hive
    - http.host=0.0.0.0
    - "ES_JAVA_OPTS=-Xms256m -Xmx256m"
  volumes:
    - elasticsearchdata:/usr/share/elasticsearch/data
  networks:
    - SOC_NET
```

Figure 22: Configuration du Service Elasticsearch

2.4.1.4 MinIO

Nature de l'installation : MinIO est utilisé pour fournir un stockage S3 compatible à TheHive. Ce service déploie MinIO en tant que serveur de stockage d'objets, avec une interface d'administration accessible via le port 9002. Les données sont stockées dans

un volume persistant pour assurer la persistance entre les redémarrages des conteneurs.

```
minio:
  image: quay.io/minio/minio
  restart: unless-stopped
  command: ["minio", "server", "/data", "--console-address", ":9002"]
  environment:
    - MINIO_ROOT_USER=minioadmin
    - MINIO_ROOT_PASSWORD=minioadmin
  ports:
    - "0.0.0.0:9002:9002"
  volumes:
    - "miniodata:/data"
  networks:
    - SOC_NET
```

Figure 23:Configuration du Service MinIO

2.4.1.5 Cortex

Nature de l'installation : Cortex est un moteur d'analyse d'incidents qui fonctionne en complément de TheHive. Le service est configuré pour utiliser l'image officielle de Cortex et est connecté à Elasticsearch pour la gestion des résultats d'analyse. Le volume Docker permet de lier les fichiers de configuration et les analyseurs nécessaires au bon fonctionnement de Cortex.

```
cortex.local:
  image: thehiveproject/cortex:latest
  restart: unless-stopped
  environment:
    - job_directory=/tmp/cortex-jobs
    - docker_job_directory=/tmp/cortex-jobs
  volumes:
    - /var/run/docker.sock:/var/run/docker.sock
    - /tmp/cortex-jobs:/tmp/cortex-jobs
    - ./cortex/logs:/var/log/cortex
    - ./cortex/conf/application.conf:/cortex/application.conf
    - ./Cortex-Analyzers:/opt/cortex/Cortex-Analyzers
  depends_on:
    - elasticsearch
  ports:
    - "0.0.0.0:9001:9001"
  networks:
    - SOC_NET
```

Figure 24:Configuration du Service Cortex

2.4.1.6 MISP

Nature de l'installation : MISP (Malware Information Sharing Platform) est une plateforme de partage d'indicateurs de compromission (IOC) et d'informations sur les menaces. Le service utilise une image Docker pour MISP et dépend d'une base de données MySQL, avec les configurations nécessaires stockées dans des volumes externes pour la persistance des données.

```yaml
misp.local:
  image: coolacid/misp-docker:core-latest
  restart: unless-stopped
  depends_on:
    - misp_mysql
  ports:
    - "0.0.0.0:80:80"
    - "0.0.0.0:443:443"
  volumes:
    - "./server-configs/:/var/www/MISP/app/Config/"
    - "./logs/:/var/www/MISP/app/tmp/logs/"
    - "./files/:/var/www/MISP/app/files"
    - "./ssl/:/etc/nginx/certs"
  environment:
    - MYSQL_HOST=misp_mysql
    - MYSQL_DATABASE=mispdb
    - MYSQL_USER=mispuser
    - MYSQL_PASSWORD=misppass
    - MISP_ADMIN_EMAIL=mispadmin@lab.local
    - MISP_ADMIN_PASSPHRASE=mispadminpass
    - MISP_BASEURL=localhost
    - TIMEZONE=Europe/London
    - "INIT=true"
    - "CRON_USER_ID=1"
    - "REDIS_FQDN=redis"
    - "HOSTNAME=https://192.168.122.155"
  networks:
    - SOC_NET
```

Figure 25 : Configuration du Service MISP

2.4.1.7 MISP MySQL

Nature de l'installation : MySQL est utilisé pour gérer la base de données de MISP. Le service est configuré pour utiliser l'image officielle de MySQL, avec des variables d'environnement définissant le nom de la base de données et les identifiants d'utilisateur.

```yaml
misp_mysql:
  image: mysql/mysql-server:5.7
  restart: unless-stopped
  volumes:
    - mispsqldata:/var/lib/mysql
  environment:
    - MYSQL_DATABASE=mispdb
    - MYSQL_USER=mispuser
    - MYSQL_PASSWORD=misppass
    - MYSQL_ROOT_PASSWORD=mispass
  networks:
    - SOC_NET

redis:
  image: redis:latest
  networks:
    - SOC_NET
```

Figure 26 : Configuration de la Base de Données MySQL pour MISP

2.4.1.8 Redis

Nature de l'installation : Redis est utilisé pour la gestion des sessions et la mise en cache dans le contexte de MISP. Ce service utilise l'image officielle de Redis et est connecté aux autres services dans le réseau SOC_NET.

```yaml
redis:
  image: redis:latest
  networks:
    - SOC_NET
```

Figure 27 : Configuration du Service Redis

2.4.1.9 MISP Modules

Nature de l'installation : MISP Modules est un composant qui permet d'étendre les fonctionnalités de MISP avec des modules d'analyse et d'intégration externes. Ce service est lié à Redis et à MySQL, et est configuré pour dépendre de ces services.

```
misp-modules:
  image: coolacid/misp-docker:modules-latest
  environment:
    - "REDIS_BACKEND=redis"
  depends_on:
    - redis
    - misp_mysql
  networks:
    - SOC_NET
```

Figure 28: Configuration des Modules MISP

2.4.2 Installation wazuh

L'installation de Wazuh sur Ubuntu a été réalisée en utilisant un script automatisé. Ce script, accessible directement depuis le site officiel de Wazuh, permet de simplifier l'installation en configurant automatiquement tous les composants nécessaires au bon fonctionnement de la plateforme.

```
nasri@nasri-virtual-machine:~$ curl -sO https://packages.wazuh.com/4.7/wazuh-ins
tall.sh && sudo bash ./wazuh-install.sh -a.
```

Figure 29:instalation wazuh

Ce script :

-Télécharge les fichiers nécessaires depuis les dépôts officiels de Wazuh.

-Configure et démarre les services principaux (Wazuh Manager, Elasticsearch, Filebeat, Kibana).

-Automatises les étapes complexes pour une configuration standard prête à l'emploi.

Une fois l'installation terminée, Wazuh est accessible via l'interface utilisateur web intégrée à Kibana, permettant une gestion centralisée des données de sécurité.

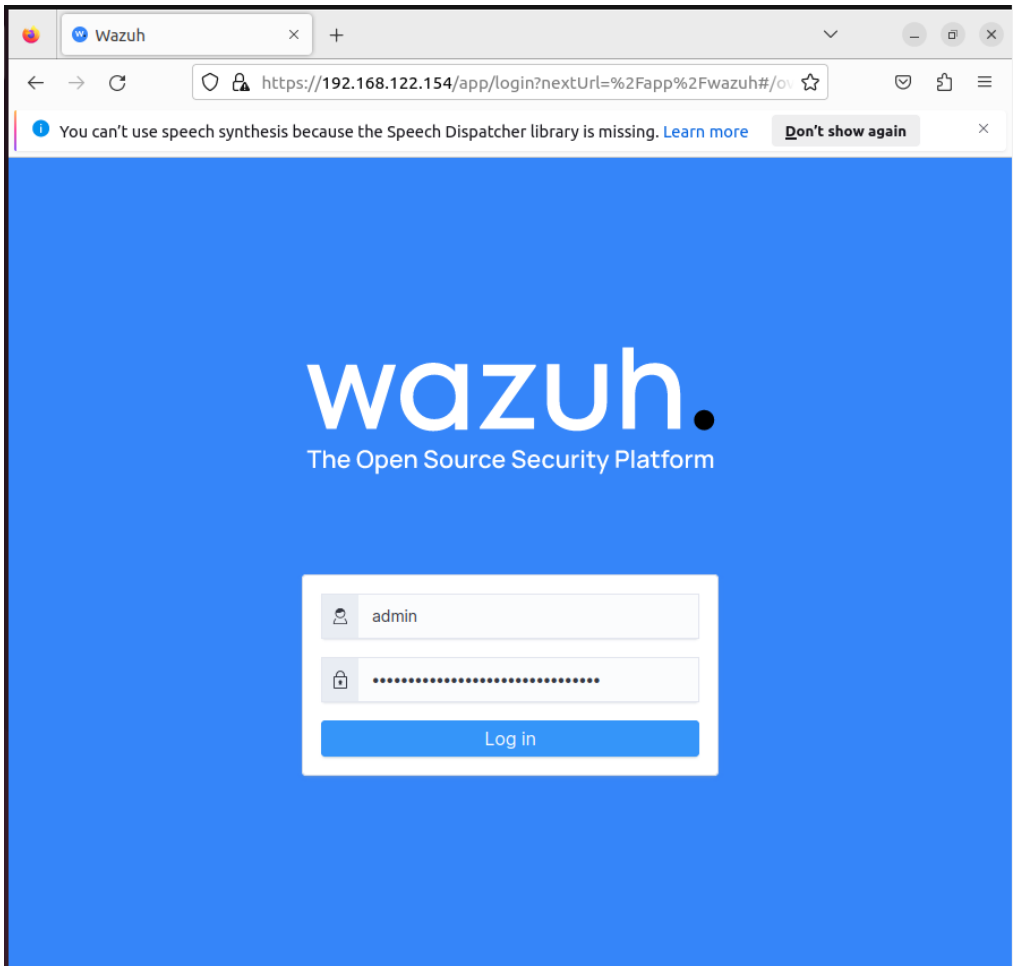

Figure 30:interface utilisateur web intègre a Kibana

2.5 Vision globale de la solution

L'architecture du SOC est placée dans la zone DMZ pour permettre une surveillance centralisée des flux de données en provenance des différentes interfaces (WAN, LAN, et LAN2) tout en maintenant une séparation entre le réseau interne et les connexions externes. Dans cet emplacement, le SOC peut ainsi analyser le trafic entrant et sortant sans compromettre la sécurité du réseau interne. Cette position stratégique permet également au SOC de gérer efficacement les événements de sécurité et de répondre rapidement aux incidents, en surveillant particulièrement les interactions avec les services exposés dans la DMZ, tels que les serveurs Apache et DNS, qui sont souvent des points sensibles d'accès extérieur.

Ainsi, la présence du SOC dans la DMZ renforce la sécurité globale du réseau en offrant une protection proactive et une capacité de réponse rapide aux menaces potentielles, tout en garantissant un isolement des composants critiques du réseau interne.

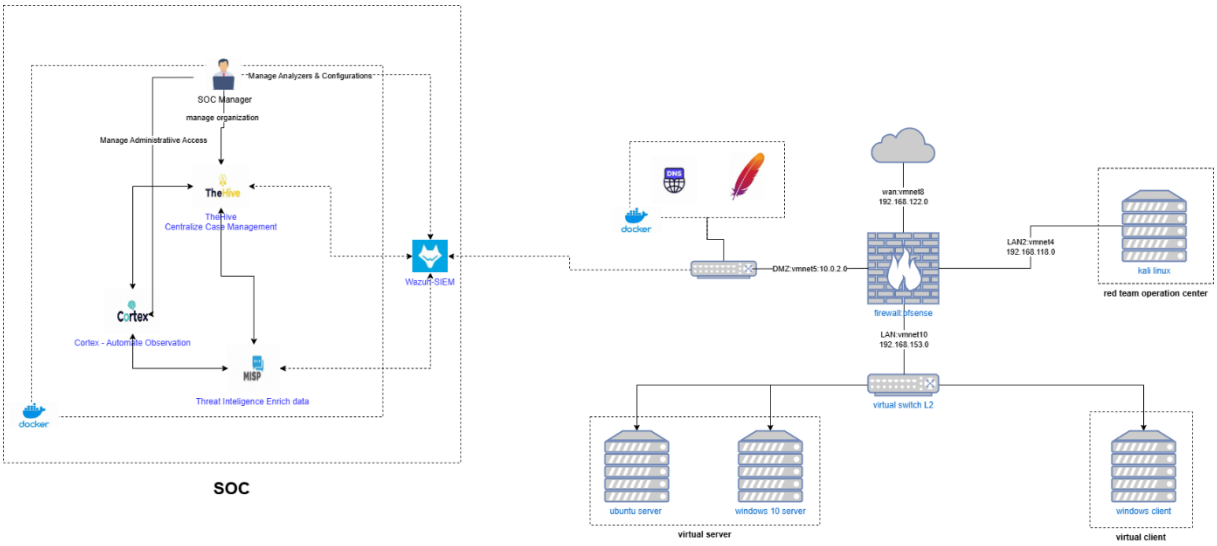

Figure 31:architecture complète

3 Intégration

3.1 Intégration entre The Hive et cortex

The Hive utilise Cortex comme moteur d'analyse pour automatiser les tâches et enrichir les alertes.

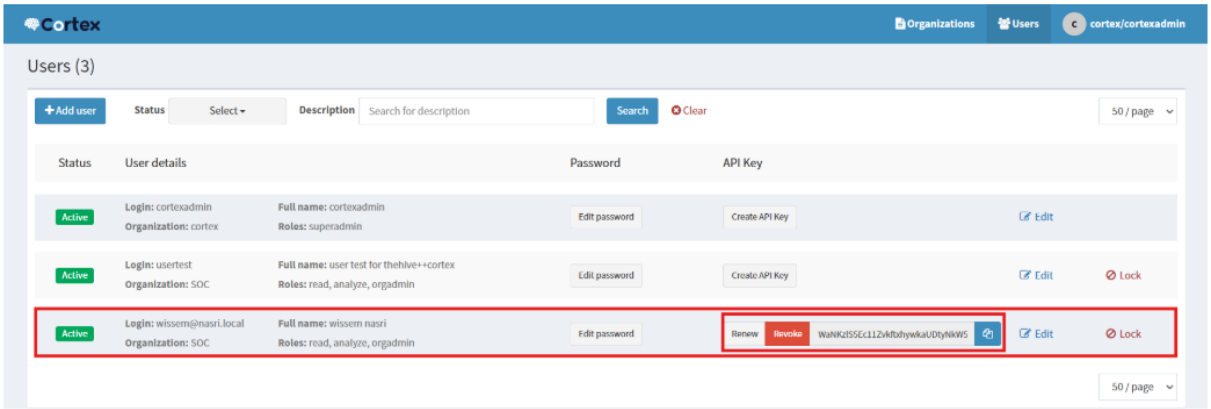

Figure 32:Créez un utilisateur et Générez une clé API

```
services:
  thehive:
    image: strangebee/thehive:5.2
    restart: unless-stopped
    depends_on:
      - cassandra
      - elasticsearch
      - minio
      - cortex.local
    mem_limit: 1500m
    ports:
      - "0.0.0.0:9000:9000"
    environment:
      - JVM_OPTS="-Xms1024M -Xmx1024M"
    command:
      - --secret
      - "lab123456789"
      - "--cql-hostnames"
      - "cassandra"
      - "--index-backend"
      - "elasticsearch"
      - "--es-hostnames"
      - "elasticsearch"
      - "--s3-endpoint"
      - "http://minio:9002"
      - "--s3-access-key"
      - "minioadmin"
      - "--s3-secret-key"
      - "minioadmin"
      - "--s3-use-path-access-style"
      - "--cortex-port"
      - "9001"
      - "--cortex-keys"
      - "WaNKzlS5Ec11ZvkftxhywkaUDtyNkW5q"
    volumes:
      - ./thehive/conf/application.conf:/etc/thehive/application.conf
```

Figure 33:Configuration dans The Hive

3.2 Intégration entre The Hive et MISP

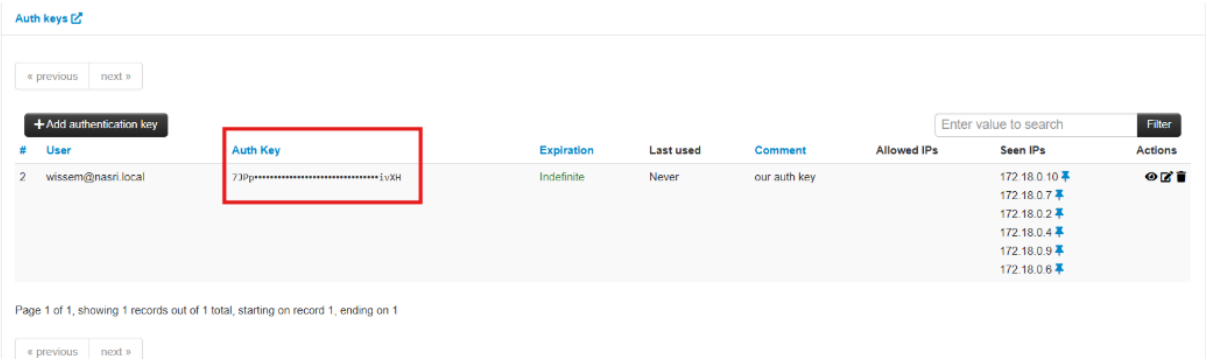

Figure 34:Générez une clé API MISP

```
root@WissemNasri-SOAR: /home/nasri/docker-compose/thehive/conf
  GNU nano 6.2                          application.conf
play.modules.enabled += org.thp.thehive.connector.misp.MispModule
misp {
  interval: 1 hour
  servers: [
    {
      name = "MISP"
      url = "https://misp.local"
      auth {
        type = key
        key = "7JPpwxLr3VgRcslkXNROzOeOM2QzT8XLQ5xJivXH" #your API Key here
      }
      tags = ["tag1", "tag2", "tag3"]
      caseTemplate = "misp"
      includedTheHiveOrganisations = ["SOC"]
    }
  ]
}
```

Figure 35:Ajoutez la configuration de MISP dans le fichier application.conf de The Hive

integration effectué:

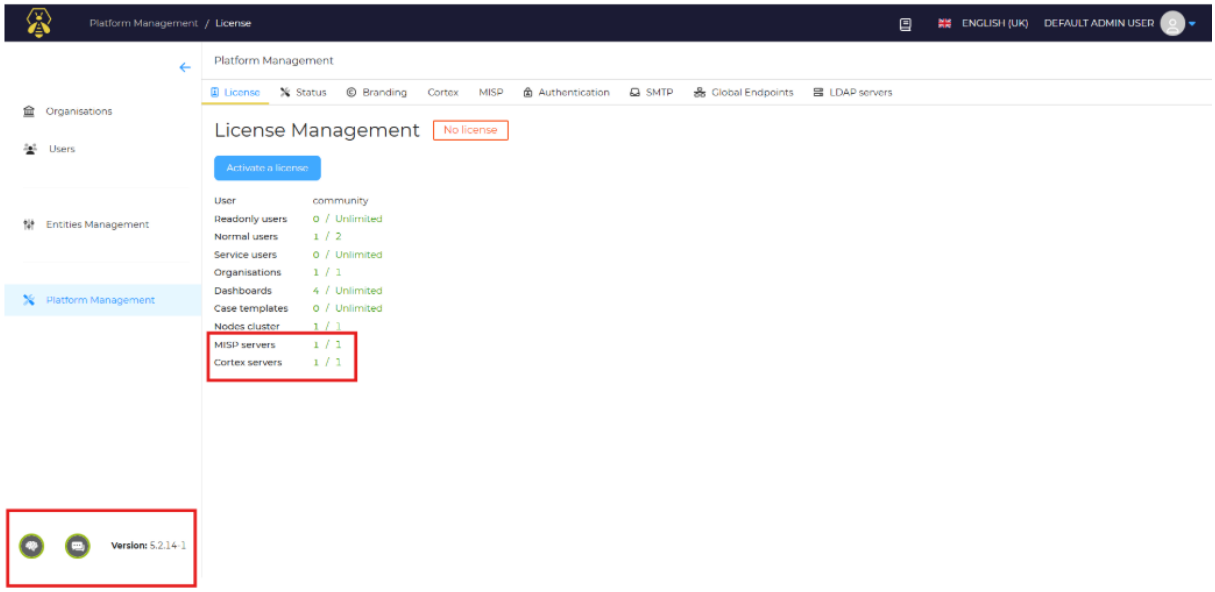

Figure 36: intégration effectué de cortex et misp

Configuration des Analyseurs MISP dans Cortex pour l'Organisation SOC

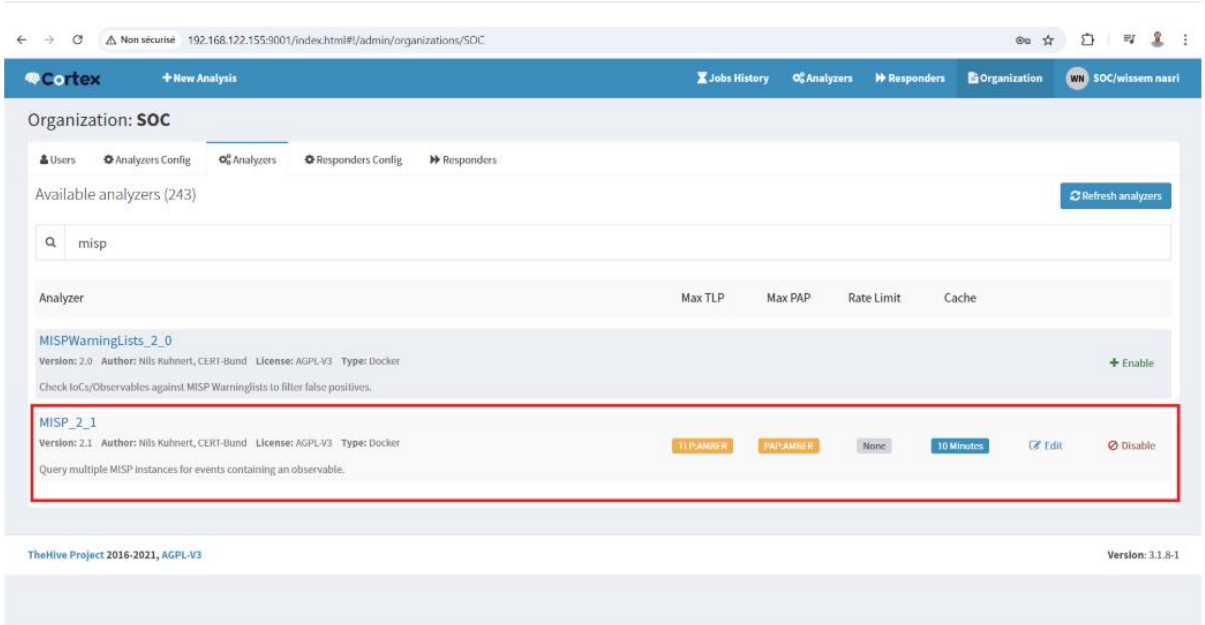

Figure 37:integration misp et cortex

3.3 intégration entre wazuh et the Hive

3.3.1 Création du script d'intégration personnalisé

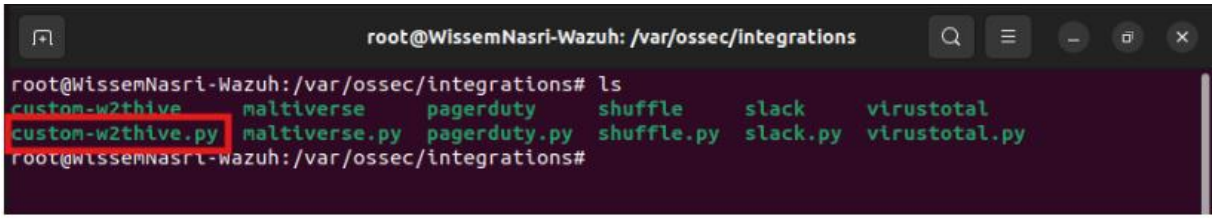

Figure 38:un script Python nommé custom-w2thive.py

Ce script contient une variable appelée lvl_threshold, définie par défaut à 0. Cela signifie que toutes les alertes générées par Wazuh seront envoyées à The Hive. Attention : Si vous surveillez de nombreux agents, cela pourrait générer un grand volume d'alertes. Vous pouvez ajuster cette valeur pour ne transmettre que les alertes les plus critiques (le seuil va de 0 à 15, selon les classifications des règles Wazuh, disponibles dans le manuel).

3.3.2 Création d'un script Bash

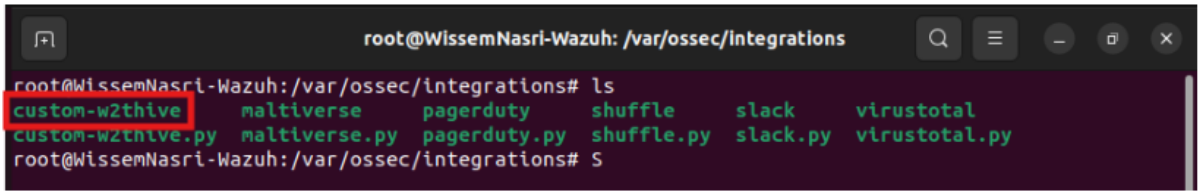

Figure 39:script bash

Ce script servira à exécuter correctement le fichier Python custom-w2thive.py que vous avez créé.

3.3.3 Activation l'intégration dans la configuration de Wazuh

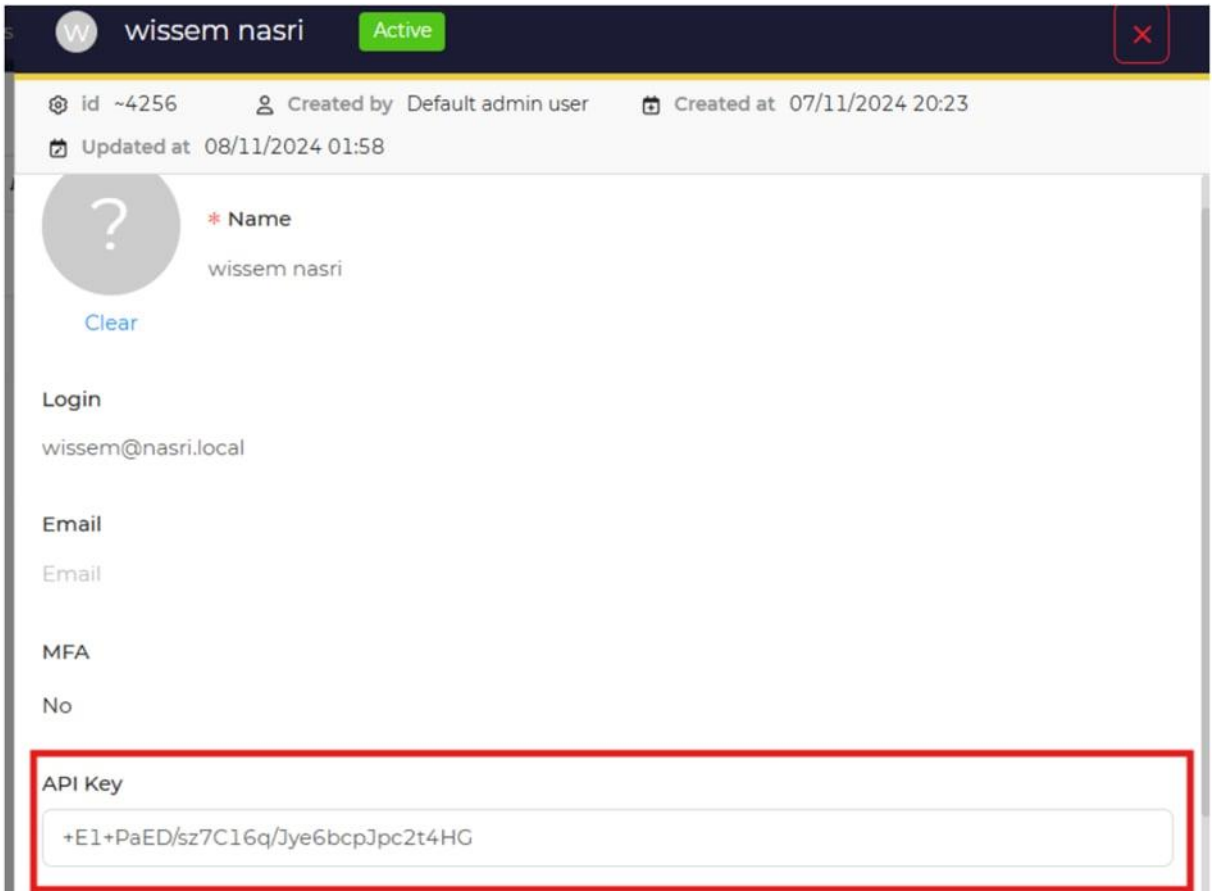

Figure 40:API Key the Hive

```
root@WissemNasri-Wazuh: /var/ossec/etc
GNU nano 6.2                                ossec.conf *

<!--
  Wazuh - Manager - Default configuration for ubuntu 22.04
  More info at: https://documentation.wazuh.com
  Mailing list: https://groups.google.com/forum/#!forum/wazuh
-->

<ossec_config>
  <global>
    <jsonout_output>yes</jsonout_output>
    <alerts_log>yes</alerts_log>
    <logall>no</logall>
    <logall_json>no</logall_json>
    <email_notification>no</email_notification>
    <smtp_server>smtp.example.wazuh.com</smtp_server>
    <email_from>wazuh@example.wazuh.com</email_from>
    <email_to>recipient@example.wazuh.com</email_to>
    <email_maxperhour>12</email_maxperhour>
    <email_log_source>alerts.log</email_log_source>
    <agents_disconnection_time>10m</agents_disconnection_time>
    <agents_disconnection_alert_time>0</agents_disconnection_alert_time>
  </global>
  <integration>
    <name>custom-w2thive</name>
    <hook_url>http://192.168.122.155:9000</hook_url>
    <api_key>+E1+PaED/sz7C16q/Jye6bcpJpc2t4HG</api_key>
    <alert_format>json</alert_format>
  </integration>
```

Figure 41:Activation l'intégration dans la configuration de Wazuh

En résumé, cette procédure permet d'intégrer Wazuh et The Hive afin que les alertes de sécurité soient automatiquement transmises à The Hive pour une gestion centralisée.

4 Test et réalisation

4.1 Test de fonctionnement

4.1.1 Domain

Enlevé un domain depuis cette evenement a titre d'exemple **AA24-241A Iran-based Cyber Actors Enabling Ransomware Attacks on**

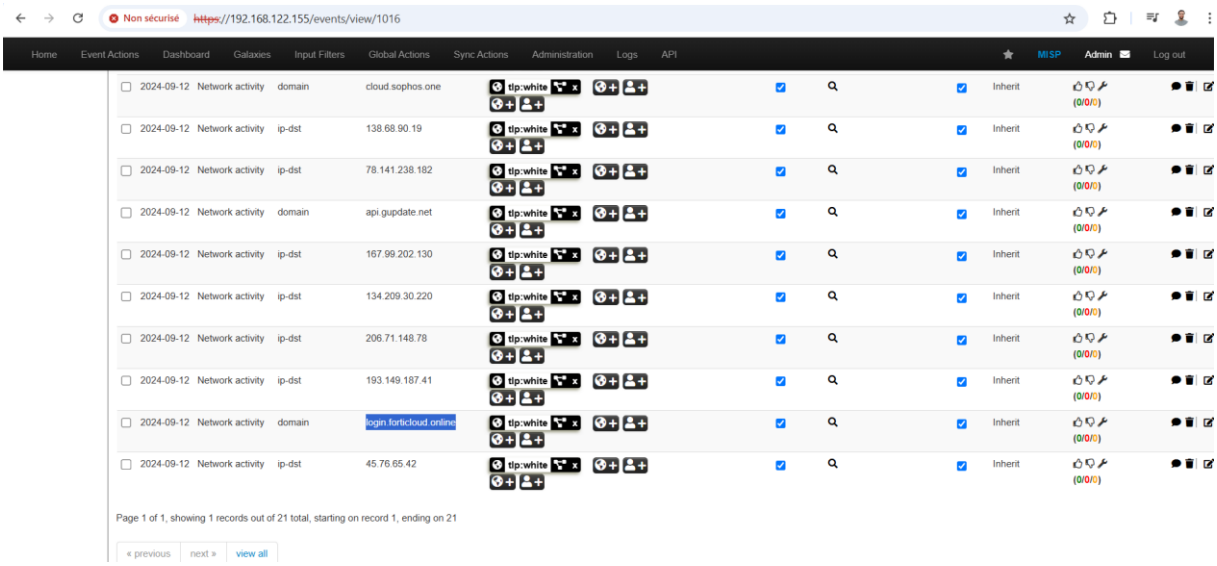

Figure 42:exemple d'un Domain enlevé depuis misp (IOCs)

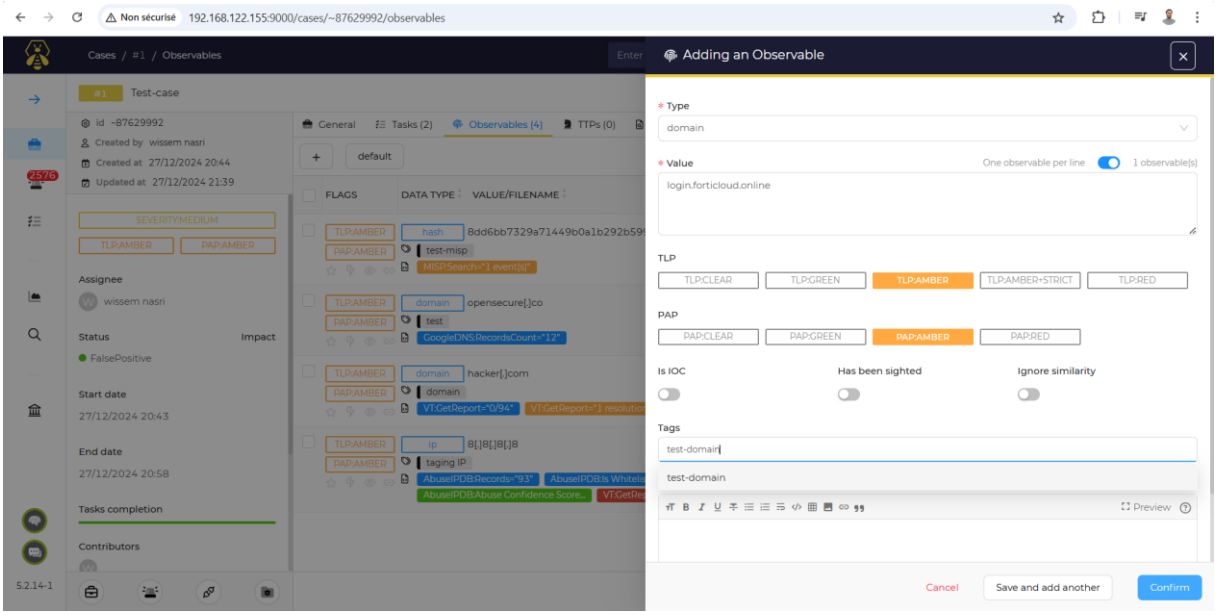

Figure 43:creation d'un observation de type domain

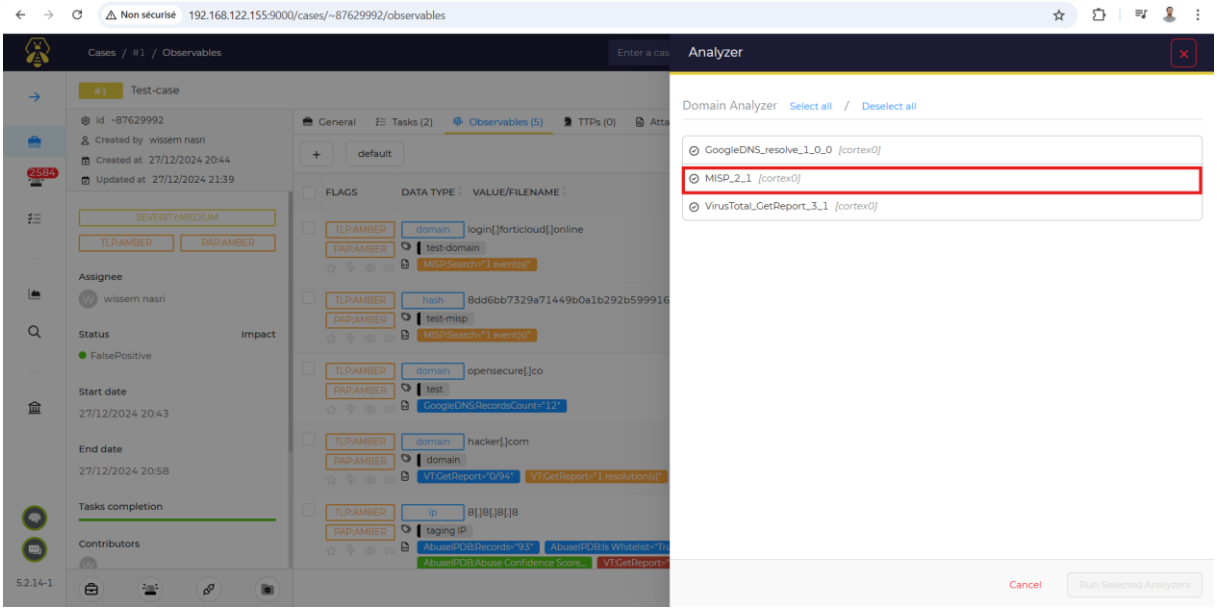

Figure 44:analyse des IOCs existant

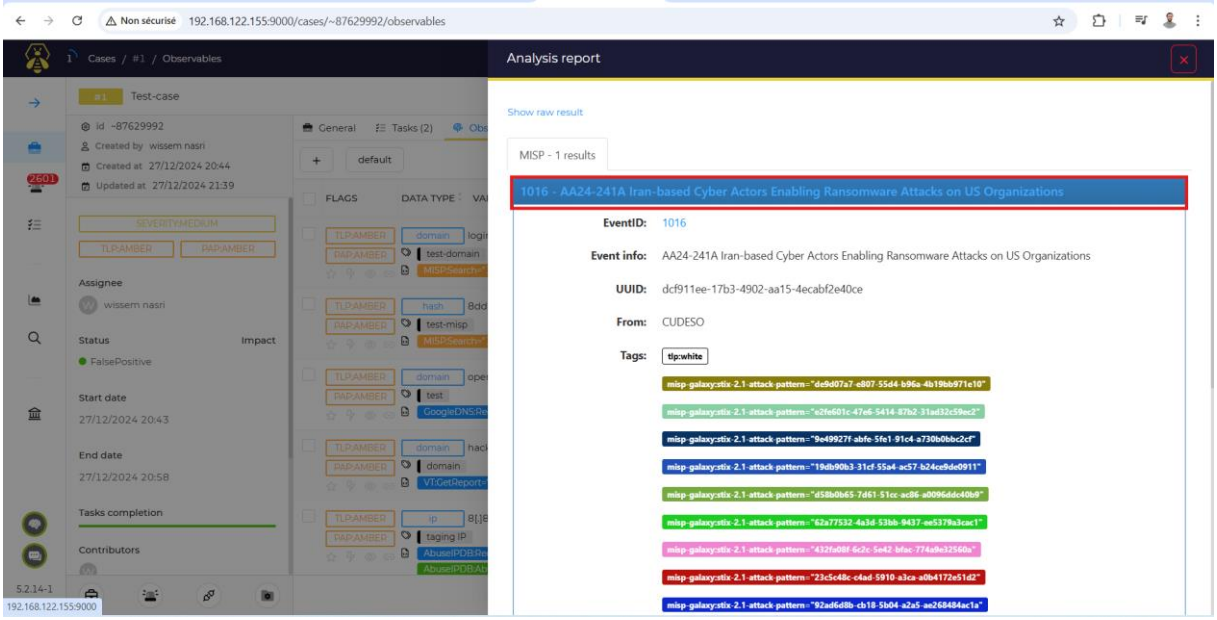

Figure 45:investigation dans les IOCs existant +resulatat+rapport

4.1.2 Hash

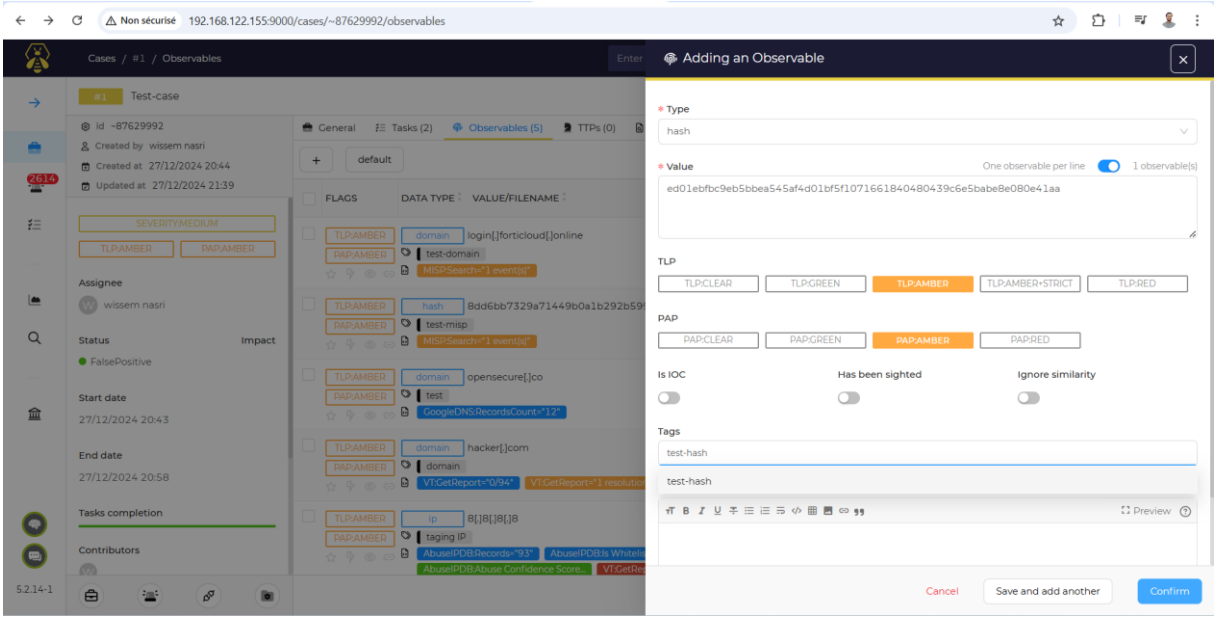

Figure 46:ajout d'une observation de type hash (wannacry)

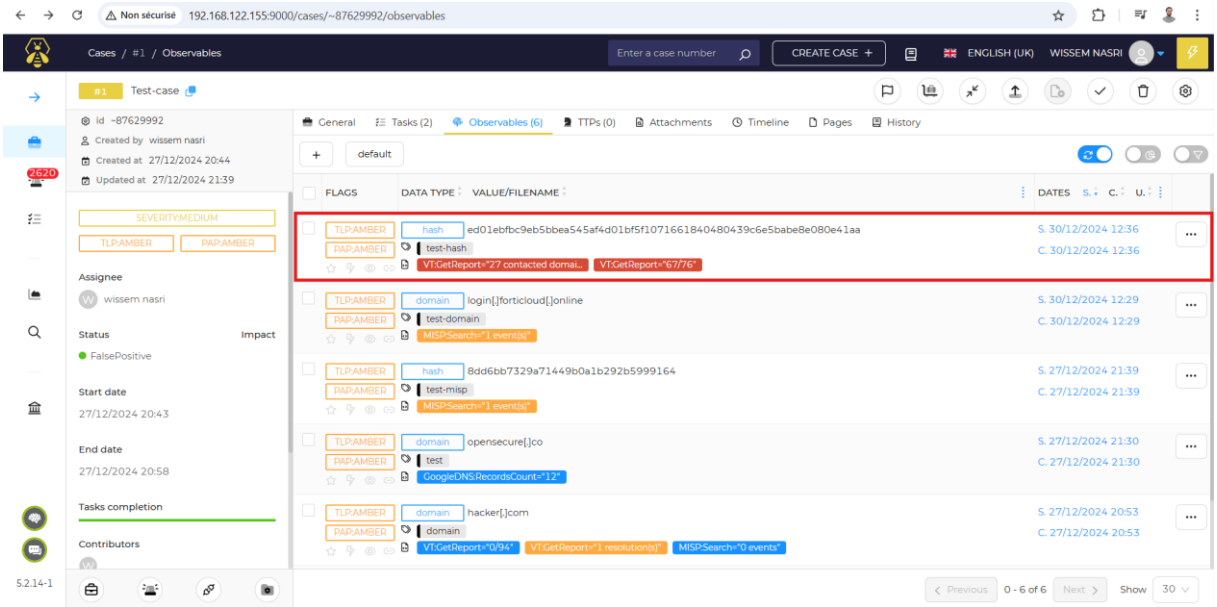

Figure 47:resultat de l'analyse via VIRUSTOTAL

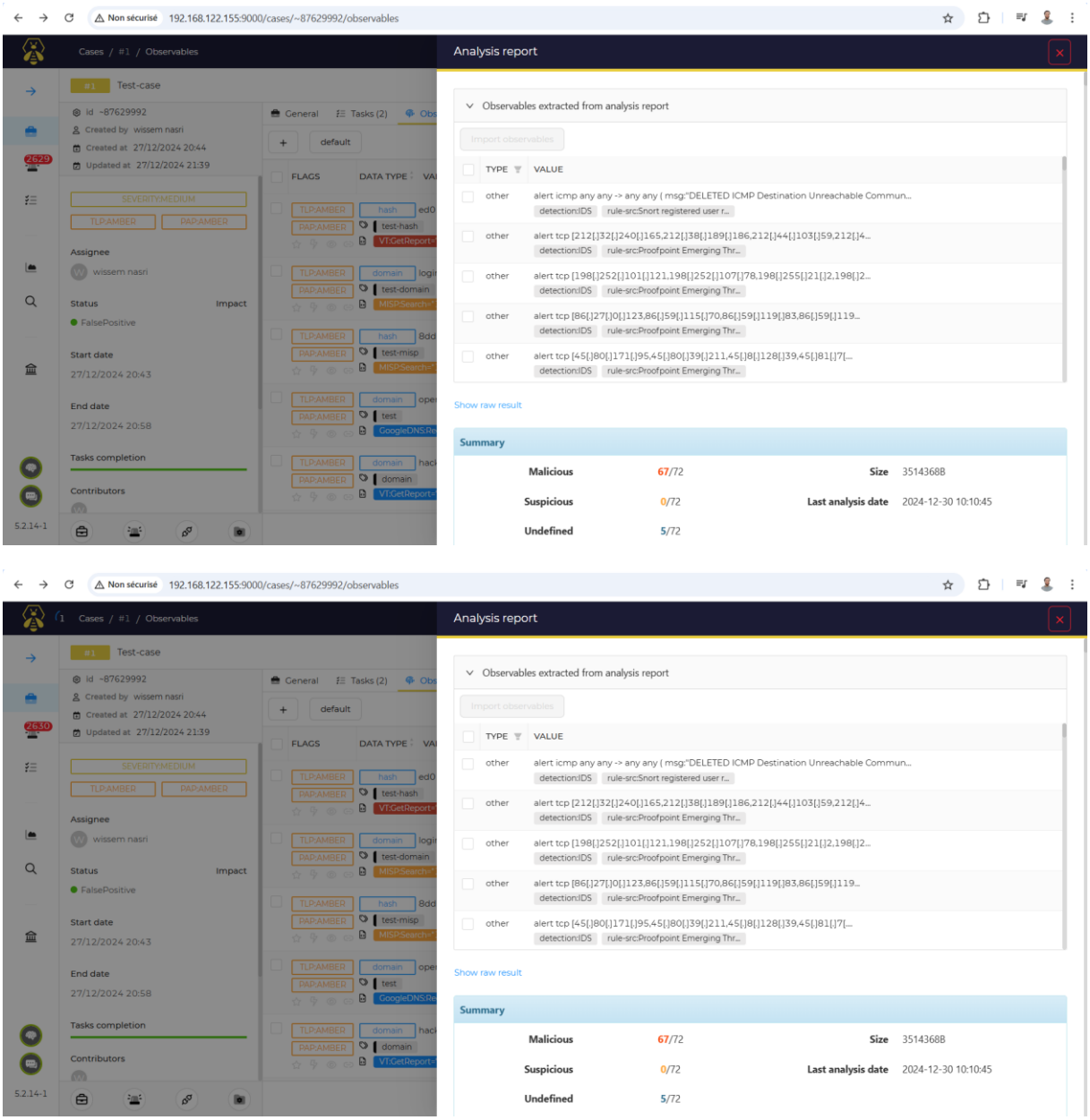

Figure 48:rapport généré

4.2 Test brute force attack

```
nasri@nasri-virtual-machine: ~
sri@nasri-virtual-machine:~$ whoami
asri
sri@nasri-virtual-machine:~$ ifconfig
ocker0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        inet 172.17.0.1  netmask 255.255.0.0  broadcast 172.17.255.255
        ether 02:42:a1:be:25:b2  txqueuelen 0  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

ns33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.122.198  netmask 255.255.255.0  broadcast 192.168.122.255
        inet6 fe80::7390:9a72:dc2e:4f47  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:db:42:b0  txqueuelen 1000  (Ethernet)
        RX packets 584862  bytes 827005973 (827.0 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 170343  bytes 14720617 (14.7 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

o: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 353  bytes 42579 (42.5 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 353  bytes 42579 (42.5 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

sri@nasri-virtual-machine:~$
```

Figure 49:username+ip address de l'agent

```
┌──(root㉿kali)-[~]
└─# hydra -l nasri -P password.txt 192.168.122.198 ssh

Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizatio
ns, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-12-30 12:30:08
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t
4
[DATA] max 1 task per 1 server, overall 1 task, 1 login try (l:1/p:1), ~1 try per task
[DATA] attacking ssh://192.168.122.198:22/
[22][ssh] host: 192.168.122.198   login: nasri   password: root
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-12-30 12:30:09

┌──(root㉿kali)-[~]
└─#
```

Figure 50:lancement de l'attaque

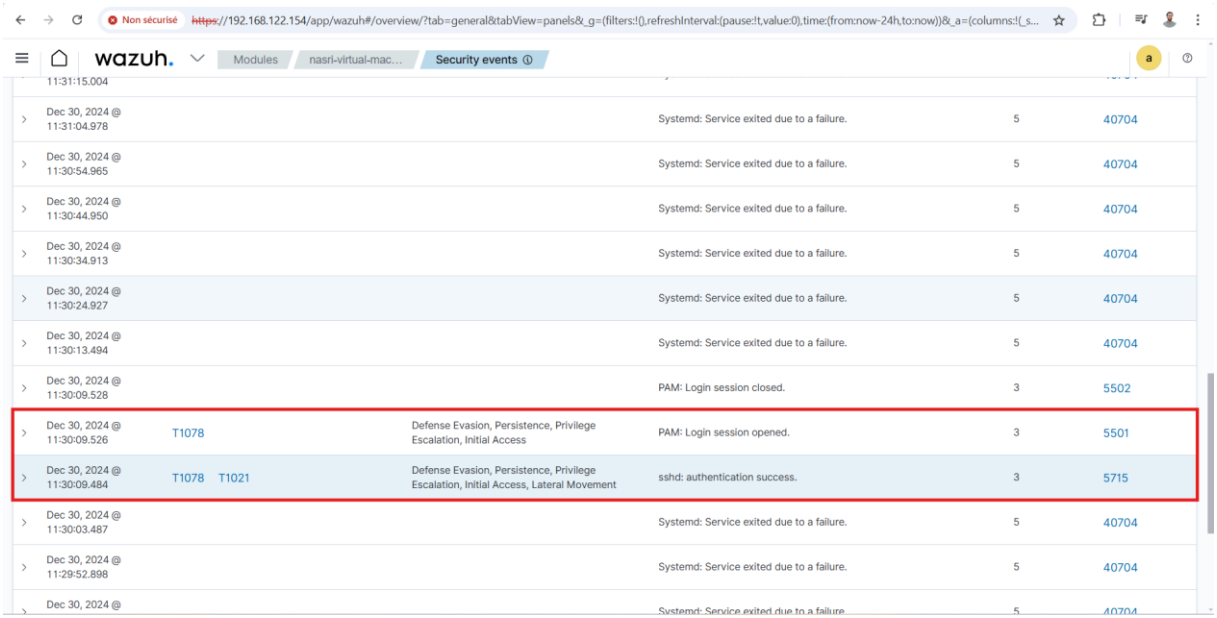

Figure 51:detection par wazuh

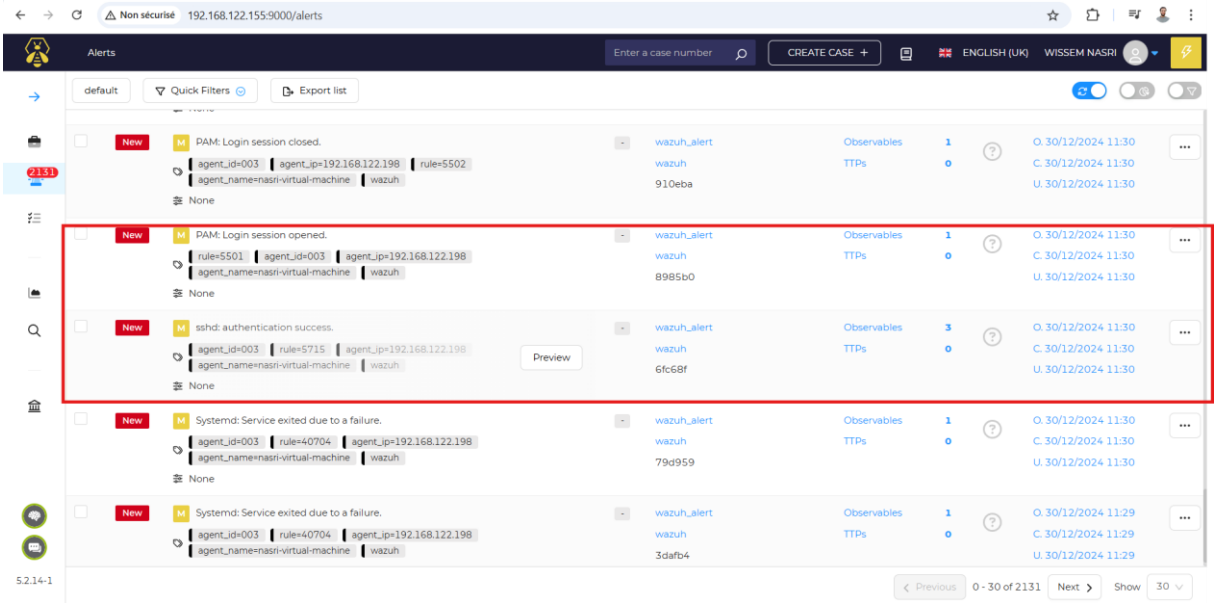

Figure 52:détection par the Hive

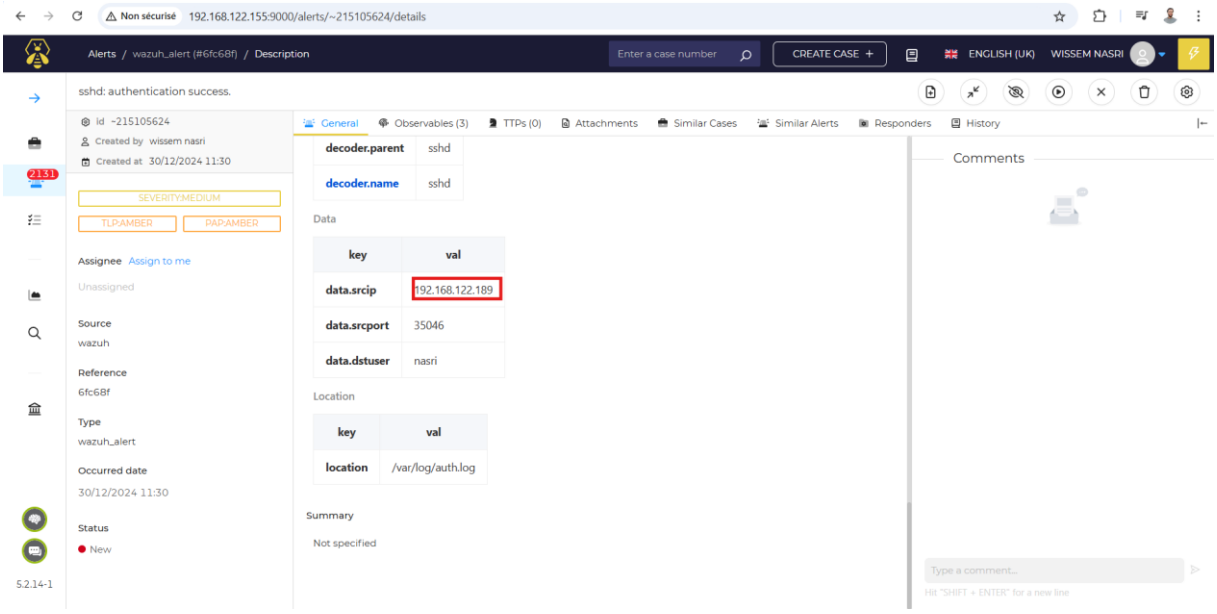

Figure 53:information sur l'attaquant

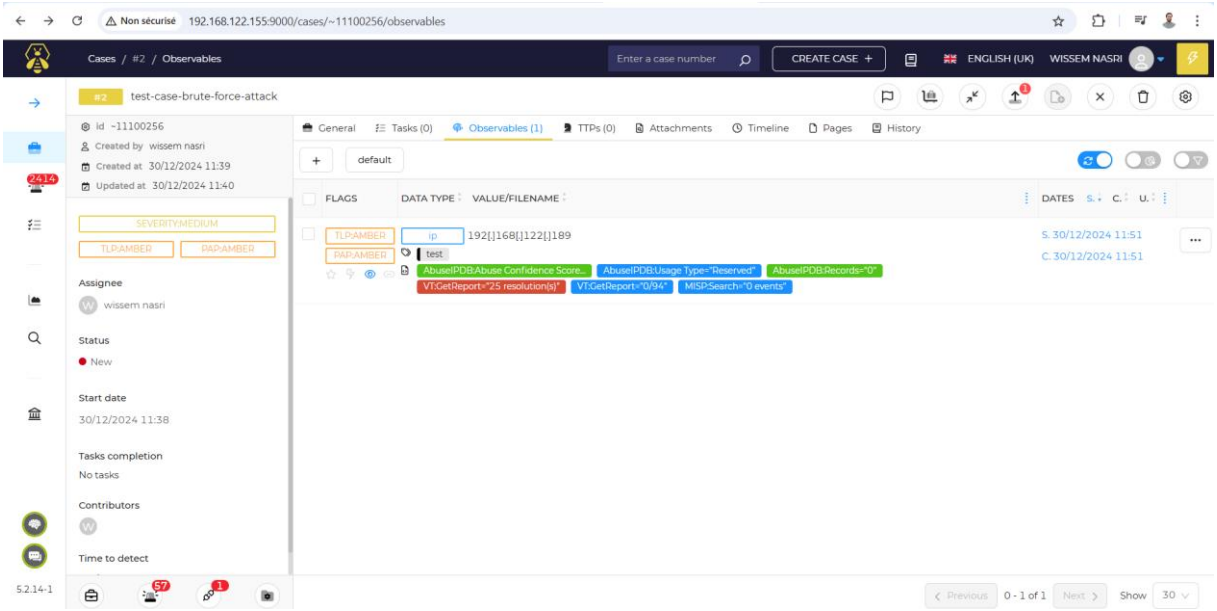

Figure 54:creation d'une case et ajout de l'adresse de l'acteur malveillant

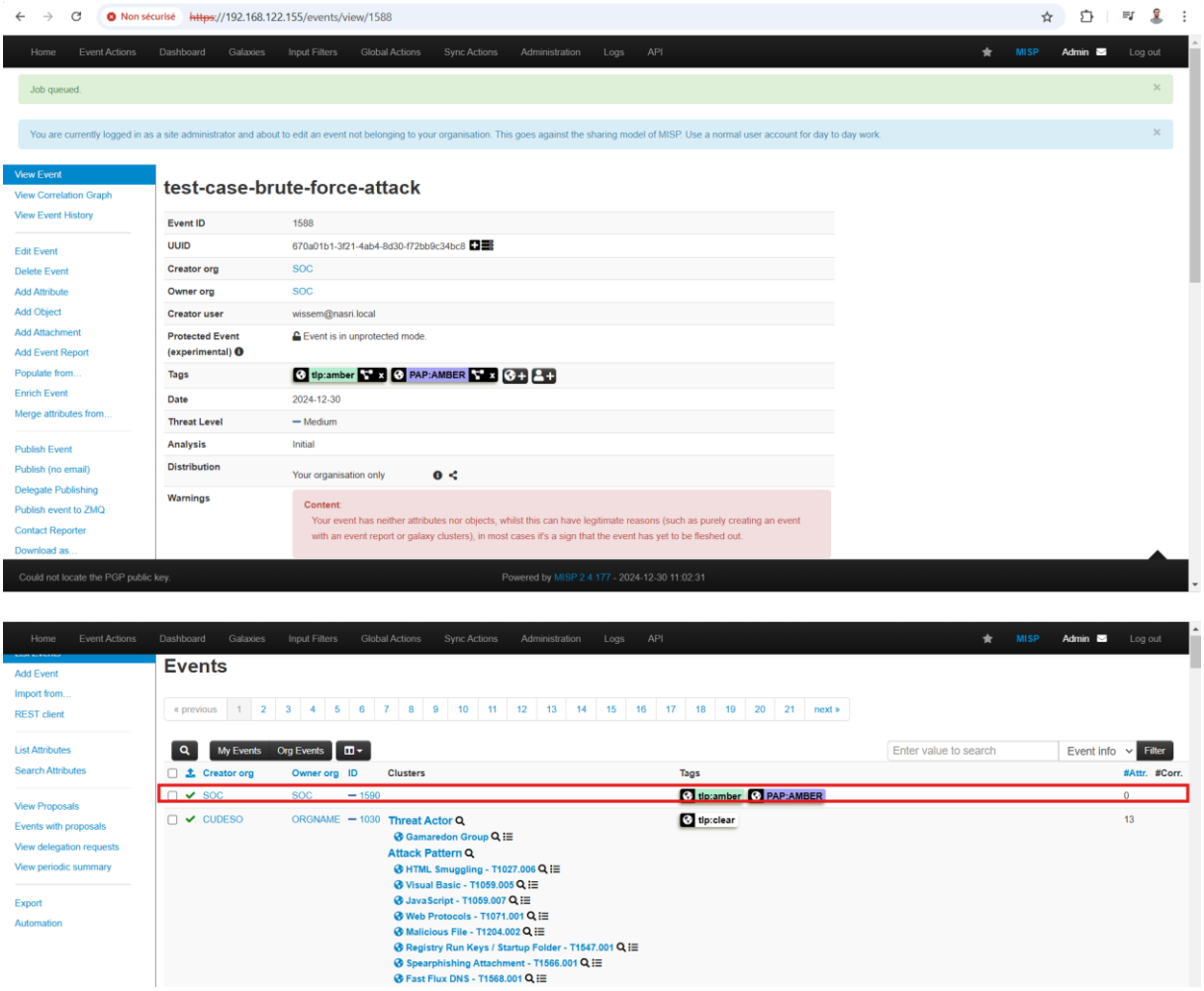

Figure 55:exportation de case vers misp

5 Conclusion

La mise en œuvre de la solution SOC illustre la puissance et l'efficacité de l'automatisation dans le déploiement d'un centre d'opérations de sécurité complet.

Chaque composant est installé et configuré avec une intervention humaine minimale, garantissant ainsi une cohérence accrue et réduisant les risques d'erreurs humaines.